



А. И. Солонина, Д. М. Клионский, Т. В. Меркучева, С. Н. Перов

Цифровая обработка сигналов и MATLAB



**А. И. Солонина, Д. М. Клионский,
Т. В. Меркучева, С. Н. Перов**

Цифровая обработка сигналов и MATLAB

Рекомендовано УМО по образованию в области
Инфокоммуникационных технологий и систем связи в качестве
учебного пособия для студентов высших учебных заведений,
обучающихся по направлению подготовки 210700 —
Инфокоммуникационные технологии и системы связи
квалификации (степени) «бакалавр» и квалификации (степени) «магистр»

Санкт-Петербург

«БХВ-Петербург»

2013

УДК 004.438
ББК 32.973.26-018.2
С60

Солонина, А. И.

С60 Цифровая обработка сигналов и MATLAB: учеб. пособие / А. И. Солонина, Д. М. Клионский, Т. В. Меркучева, С. Н. Перов. — СПб.: БХВ-Петербург, 2013. — 512 с.: ил. — (Учебная литература для вузов)

ISBN 978-5-9775-0919-0

Описываются базовые методы и алгоритмы цифровой обработки сигналов и средств их компьютерного моделирования в системе MATLAB. Даны основы алгоритмического языка MATLAB. Рассматриваются дискретные сигналы, линейные дискретные системы, дискретное преобразование Фурье с использованием алгоритмов БПФ, синтез и анализ КИХ- и БИХ-фильтров, в том числе с фиксированной точкой, спектральный анализ сигналов, многоскоростная обработка сигналов и адаптивная цифровая фильтрация. Технология обучения в процессе компьютерного моделирования на основе созданных авторами программ или графического интерфейса пользователя MATLAB расширяет теоретические знания и позволяет понять многие важные проблемы и аспекты практического применения методов и алгоритмов ЦОС. На прилагаемом CD хранятся обучающие программы и таблицы исходных данных.

*Для студентов, аспирантов и преподавателей вузов,
а также специалистов в области цифровой обработки сигналов*

УДК 004.438
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>
Фото	<i>Кирилла Сергеева</i>

РЕЦЕНЗЕНТЫ:

- Е. Б. Соловьева*, д-р техн. наук, завкафедрой теоретических основ электротехники
Санкт-Петербургского государственного электротехнического университета «ЛЭТИ»
- В. А. Варгаузин*, канд. техн. наук, доцент кафедры радиотехники и телекоммуникаций
Санкт-Петербургского государственного политехнического университета

Подписано в печать 30.04.13.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 41,28.
Тираж 700 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-0919-0

© Солонина А. И., Клионский Д. М., Меркучева Т. В.,
Перов С. Н., 2013
© Оформление, издательство "БХВ-Петербург", 2013

Оглавление

- Предисловие 9
- ЧАСТЬ I. ЗНАКОМСТВО С MATLAB 13
- Глава 1. Знакомство с MATLAB. Основные объекты языка MATLAB.....15
 - 1.1. Краткая теоретическая справка.....15
 - 1.1.1. Режим прямых вычислений.....16
 - 1.1.2. Базовые объекты языка MATLAB.....16
 - 1.1.3. Рабочая область памяти Workspace.....25
 - 1.1.4. Сохранение данных на диске 26
 - 1.2. Содержание лабораторной работы 26
 - 1.3. Задание на лабораторную работу.....26
 - 1.4. Задание на самостоятельную работу 31
 - 1.5. Отчет и контрольные вопросы.....32
 - 1.6. Литература.....33
- Глава 2. Операции с матрицами.....34
 - 2.1. Краткая теоретическая справка.....34
 - 2.1.1. Функции генерации типовых матриц.....36
 - 2.1.2. Преобразование матриц.....36
 - 2.1.3. Поэлементные операции с матрицами 37
 - 2.1.4. Операции с матрицами в задачах линейной алгебры.....38
 - 2.1.4.1. Арифметические операции с матрицами.....38
 - 2.1.4.2. Транспонирование и эрмитово сопряжение матриц 39
 - 2.1.4.3. Обращение матриц.....41
 - 2.1.4.4. Матричное деление 41
 - 2.1.5. Норма матрицы и вектора 42
 - 2.1.6. Операции с матрицами в задачах математической статистики.....43
 - 2.2. Содержание лабораторной работы 44
 - 2.3. Задание на лабораторную работу.....45
 - 2.4. Задание на самостоятельную работу 48
 - 2.5. Отчет и контрольные вопросы.....48
 - 2.6. Литература.....49
- Глава 3. Типы массивов.....50
 - 3.1. Краткая теоретическая справка.....50
 - 3.1.1. Матрицы числового и логического типов.....51
 - 3.1.2. Матрицы символьного типа.....51
 - 3.1.3. Структуры (массивы записей).....53
 - 3.1.4. Массивы ячеек.....55
 - 3.1.5. Определение типа массива 56
 - 3.2. Содержание лабораторной работы 56
 - 3.3. Задание на лабораторную работу.....56
 - 3.4. Задание на самостоятельную работу 58
 - 3.5. Отчет и контрольные вопросы.....59
 - 3.6. Литература.....59

Глава 4. Средства графики.....	60
4.1. Краткая теоретическая справка.....	60
4.1.1. Двумерные графики	61
4.1.2. Управление свойствами двумерных графиков	61
4.1.3. Трехмерные графики	64
4.1.4. Управление свойствами трехмерных графиков.....	65
4.2. Содержание лабораторной работы	66
4.3. Задание на лабораторную работу.....	66
4.4. Задание на самостоятельную работу	69
4.5. Отчет и контрольные вопросы.....	70
4.6. Литература.....	71
Глава 5. Режим программирования: script-файлы и function-файлы.....	72
5.1. Краткая теоретическая справка.....	72
5.1.1. Script-файлы	72
5.1.2. Function-файлы.....	73
5.1.3. Оформление и вывод листинга М-файлов	74
5.1.4. Ввод/вывод данных.....	75
5.1.5. Пауза и досрочное прерывание программы.....	76
5.1.6. Создание и хранение М-файлов.....	77
5.2. Содержание лабораторной работы	78
5.3. Задание на лабораторную работу.....	78
5.4. Задание на самостоятельную работу	80
5.5. Отчет и контрольные вопросы.....	81
5.6. Литература.....	82
Глава 6. Режим программирования: организация разветвлений и циклов	83
6.1. Краткая теоретическая справка.....	83
6.1.1. Операторы организации разветвлений.....	83
6.1.2. Операторы организации циклов	85
6.2. Содержание лабораторной работы	88
6.3. Задание на лабораторную работу.....	88
6.4. Задание на самостоятельную работу	90
6.5. Отчет и контрольные вопросы.....	91
6.6. Литература.....	91
ЧАСТЬ II. МОДЕЛИРОВАНИЕ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ	
В MATLAB.....	93
Глава 7. Дискретные сигналы	95
7.1. Краткая теоретическая справка.....	95
7.1.1. Детерминированные дискретные сигналы.....	96
7.1.2. Случайные дискретные сигналы.....	97
7.2. Содержание лабораторной работы	101
7.3. Задание на лабораторную работу.....	101
7.4. Типовой script-файл для выполнения лабораторной работы.....	107
7.5. Задание на самостоятельную работу	115
7.6. Отчет и контрольные вопросы.....	118
7.7. Литература.....	119
Глава 8. Линейные дискретные системы	120
8.1. Краткая теоретическая справка.....	120
8.1.1. Описание ЛДС во временной области.....	121
8.1.2. Описание ЛДС в z-области.....	123
8.1.3. Описание ЛДС в частотной области	125
8.1.4. Структуры звеньев 2-го порядка.....	127

8.2. Содержание лабораторной работы	130
8.3. Задание на лабораторную работу	130
8.4. Типовой script-файл для выполнения лабораторной работы	134
8.4.1. Используемые внешние функции	140
8.5. Задание на самостоятельную работу	140
8.6. Отчет и контрольные вопросы	141
8.7. Литература	142

Глава 9. Дискретное преобразование Фурье (часть 1) 143

9.1. Краткая теоретическая справка	143
9.1.1. Дискретное преобразование Фурье	143
9.1.2. Выделение дискретных гармоник полезного сигнала	145
9.1.3. Восстановление спектральной плотности	146
9.1.4. Восстановление аналогового сигнала	147
9.2. Содержание лабораторной работы	148
9.3. Задание на лабораторную работу	148
9.4. Типовой script-файл для выполнения лабораторной работы	153
9.4.1. Используемые внешние функции	162
9.5. Задание на самостоятельную работу	163
9.6. Отчет и контрольные вопросы	164
9.7. Литература	165

Глава 10. Дискретное преобразование Фурье (часть 2) 166

10.1. Краткая теоретическая справка	166
10.1.1. Растекание спектра	166
10.1.2. Улучшение различения дискретных гармоник с близко расположенными частотами	167
10.1.3. Вычисление линейных и круговых сверток с помощью ДПФ	167
10.1.4. Вычисление секционированных сверток с помощью ДПФ	169
10.2. Содержание лабораторной работы	169
10.3. Задание на лабораторную работу	169
10.4. Типовой script-файл для выполнения лабораторной работы	175
10.4.1. Используемые внешние функции	184
10.5. Задание на самостоятельную работу	185
10.6. Отчет и контрольные вопросы	186
10.7. Литература	186

Глава 11. Синтез КИХ-фильтров методом окон..... 187

11.1. Краткая теоретическая справка	187
11.1.1. Свойства КИХ-фильтров	188
11.1.2. Задание требований к АЧХ	189
11.1.3. Структуры КИХ-фильтров	193
11.1.4. Процедура синтеза КИХ-фильтров методом окон	195
11.1.5. Синтез КИХ-фильтров методом окон в MATLAB	196
11.2. Содержание лабораторной работы	197
11.3. Задание на лабораторную работу	198
11.4. Типовой script-файл для выполнения лабораторной работы	202
11.4.1. Синтез и анализ КИХ-фильтра ФНЧ	202
11.4.2. Синтез и анализ КИХ-фильтра ФВЧ	205
11.4.3. Синтез и анализ КИХ-фильтра ПФ	208
11.4.4. Синтез и анализ КИХ-фильтра РФ	210
11.4.5. Используемые внешние функции	213
11.5. Задание на самостоятельную работу	216
11.6. Отчет и контрольные вопросы	217
11.7. Литература	218

Глава 12. Синтез КИХ-фильтров методом наилучшей равномерной (чебышевской) аппроксимации	219
12.1. Краткая теоретическая справка.....	219
12.1.1. Процедура синтеза КИХ-фильтров методом чебышевской аппроксимации	219
12.1.2. Синтез КИХ-фильтров методом чебышевской аппроксимации в MATLAB	222
12.1.3. Описание требований к характеристике затухания в виде объекта <i>fdesign</i>	223
12.1.4. Синтез КИХ-фильтров в виде объектов <i>dfilt</i> на основе объектов <i>fdesign</i>	225
12.2. Содержание лабораторной работы.....	226
12.3. Задание на лабораторную работу.....	226
12.4. Типовой script-файл для выполнения лабораторной работы.....	232
12.4.1. Синтез и анализ КИХ-фильтра ФНЧ.....	232
12.4.2. Синтез и анализ КИХ-фильтра ФВЧ	235
12.4.3. Синтез и анализ КИХ-фильтра ПФ	239
12.4.4. Синтез и анализ КИХ-фильтра РФ	242
12.4.5. Используемые внешние функции	245
12.5. Задание на самостоятельную работу	246
12.6. Отчет и контрольные вопросы.....	247
12.7. Литература.....	248
 Глава 13. Синтез БИХ-фильтров методом билинейного Z-преобразования	249
13.1. Краткая теоретическая справка.....	249
13.1.1. Задание требований к характеристике затухания.....	249
13.1.2. Структуры БИХ-фильтров	250
13.1.3. Процедура синтеза БИХ-фильтров методом билинейного Z-преобразования	252
13.1.4. Синтез аналоговых фильтров в MATLAB	253
13.1.5. Синтез БИХ-фильтров методом билинейного Z-преобразования в MATLAB	253
13.1.6. Синтез БИХ-фильтров в виде объектов <i>dfilt</i> на основе объектов <i>fdesign</i>	255
13.1.7. Расстановка звеньев и масштабирование в каскадных структурах БИХ-фильтров	256
13.2. Содержание лабораторной работы.....	256
13.3. Задание на лабораторную работу.....	256
13.4. Типовой script-файл для выполнения лабораторной работы.....	261
13.4.1. Синтез и анализ БИХ-фильтра ФНЧ	261
13.4.2. Синтез и анализ БИХ-фильтра ФВЧ	266
13.4.3. Синтез и анализ БИХ-фильтра ПФ	270
13.4.4. Синтез и анализ БИХ-фильтра РФ	274
13.4.5. Используемые внешние функции	278
13.5. Задание на самостоятельную работу	279
13.6. Отчет и контрольные вопросы.....	280
13.7. Литература.....	280
 Глава 14. Синтез цифровых фильтров средствами GUI FDATool и FilterBuilder	281
14.1. Краткая теоретическая справка.....	281
14.1.1. Синтез цифровых фильтров в GUI FDATool	281
14.1.2. Экспорт из GUI FDATool в Workspace.....	283
14.1.3. Синтез цифровых фильтров в FilterBuilder GUI.....	284
14.2. Содержание лабораторной работы.....	287
14.3. Задание на лабораторную работу.....	287
14.4. Задание на самостоятельную работу	297
14.5. Отчет и контрольные вопросы.....	298
14.6. Литература.....	298
 Глава 15. Цифровые фильтры с фиксированной точкой	299
15.1. Краткая теоретическая справка.....	299
15.1.1. Эффекты квантования в структуре ЦФ с ФТ.....	300
15.1.2. Моделирование структуры исходного ЦФ в GUI FDATool	301

15.1.3. Моделирование структуры ЦФ с ФТ в GUI FDATool.....	301
15.1.3.1. Установка свойств ЦФ с ФТ на вкладке <i>Coefficients</i>	303
15.1.3.2. Установка свойств ЦФ с ФТ на вкладке <i>Input/Output</i>	305
15.1.3.3. Установка свойств ЦФ с ФТ на вкладке <i>Filter Internals</i>	309
15.1.4. Моделирование структуры ЦФ с ФТ в FilterBuilder GUI	313
15.2. Содержание лабораторной работы.....	316
15.3. Задание на лабораторную работу.....	317
15.4. Задание на самостоятельную работу	322
15.5. Отчет и контрольные вопросы.....	322
15.6. Литература.....	323
Глава 16. Спектральный анализ: непараметрические методы.....	324
16.1. Краткая теоретическая справка.....	324
16.1.1. Метод периодограмм.....	325
16.1.2. Основные показатели качества оценок СПМ	328
16.1.3. Метод периодограмм Даньелла	330
16.1.4. Метод периодограмм Бартлетта	332
16.1.5. Метод периодограмм Уэлча.....	333
16.1.6. Метод Блэкмана—Тьюки	334
16.1.7. Моделирование случайной последовательности с требуемой АКФ.....	337
16.1.8. Основные параметры окон	339
16.1.9. Спектрограмма.....	339
16.2. Содержание лабораторной работы.....	340
16.3. Задание на лабораторную работу.....	340
16.4. Типовой script-файл для выполнения лабораторной работы.....	346
16.5. Задание на самостоятельную работу	355
16.6. Отчет и контрольные вопросы.....	356
16.7. Литература.....	357
Глава 17. Спектральный анализ: параметрические методы.....	358
17.1. Краткая теоретическая справка.....	358
17.1.1. APCC-, AP- и CC-модели	359
17.1.2. Метод Юла—Уолкера (автокорреляционный) оценки параметров AP-модели	361
17.1.3. Методы оценки параметров AP-модели.....	364
17.1.4. Методы оценки СПМ.....	366
17.1.5. Оценка порядка AP-модели.....	367
17.1.6. Сравнение оценок СПМ с истинной СПМ.....	368
17.2. Содержание лабораторной работы.....	368
17.3. Задание на лабораторную работу.....	369
17.4. Типовой script-файл для выполнения лабораторной работы.....	373
17.5. Задание на самостоятельную работу	380
17.6. Отчет и контрольные вопросы.....	381
17.7. Литература.....	382
Глава 18. Спектральный анализ средствами GUI SPTool.....	383
18.1. Краткая теоретическая справка.....	383
18.1.1. Обращение к GUI SPTool	383
18.1.2. Сигналы: группа <i>Signals</i>	384
18.1.3. Моделирование системы цифровой фильтрации: группа <i>Filters</i>	387
18.1.4. Спектральный анализ: группа <i>Spectra</i>	389
18.1.5. Экспорт данных из GUI SPTool	391
18.2. Содержание лабораторной работы.....	392
18.3. Задание на лабораторную работу.....	392
18.4. Задание на самостоятельную работу	397
18.5. Отчет и контрольные вопросы.....	399
18.6. Литература.....	400

Глава 19. Многоскоростные системы ЦОС	401
19.1. Краткая теоретическая справка.....	401
19.1.1. Система однократной интерполяции	402
19.1.2. Система однократной децимации	406
19.1.3. Система однократной передискретизации.....	409
19.1.4. Полифазные структуры многоскоростных систем	410
19.2. Содержание лабораторной работы.....	415
19.3. Задание на лабораторную работу.....	415
19.4. Типовой script-файл для выполнения лабораторной работы.....	421
19.5. Задание на самостоятельную работу	431
19.6. Отчет и контрольные вопросы.....	433
19.7. Литература.....	434
 Глава 20. Моделирование полифазных структур многоскоростных систем средствами GUI FDATool и FilterBuilder	435
20.1. Краткая теоретическая справка.....	435
20.1.1. Моделирование полифазных структур в GUI FDATool.....	435
20.1.2. Моделирование полифазных структур в FilterBuilder GUI	440
20.1.3. Моделирование многоскоростных систем с полифазными структурами.....	441
20.2. Содержание лабораторной работы.....	441
20.3. Задание на лабораторную работу.....	441
20.4. Типовой script-файл для выполнения лабораторной работы.....	448
20.4.1. Система однократной интерполяции с полифазной структурой	449
20.4.2. Система однократной децимации с полифазной структурой.....	450
20.4.3. Система однократной передискретизации с полифазной структурой при повышении частоты дискретизации.....	452
20.4.4. Система однократной передискретизации с полифазной структурой при понижении частоты дискретизации	454
20.5. Задание на самостоятельную работу	456
20.6. Отчет и контрольные вопросы.....	456
20.7. Литература.....	457
 Глава 21. Адаптивные фильтры.....	458
21.1. Краткая теоретическая справка.....	458
21.1.1. Фильтр Винера	460
21.1.2. Алгоритм LMS	463
21.1.3. Алгоритм RLS	467
21.1.4. Применение адаптивных фильтров	472
21.1.4.1. Идентификация систем	472
21.1.4.2. Оценка импульсной характеристики неизвестной системы	474
21.1.4.3. Очистка сигнала от шума	475
21.1.4.4. Выравнивание частотной характеристики неизвестной системы	477
21.2.4.5. Оценка параметров линейного предсказания сигнала	481
21.2. Содержание лабораторной работы.....	483
21.3. Задание на лабораторную работу.....	483
21.4. Типовой script-файл для выполнения лабораторной работы.....	491
21.5. Задание на самостоятельную работу	501
21.6. Отчет и контрольные вопросы.....	503
21.7. Литература.....	504
 Список сокращений на английском языке	505
Список сокращений на русском языке	506
Предметный указатель.....	508

Предисловие

Революция в области персональных компьютеров и компьютерных технологий, продолжающаяся по сей день, стремительно распространяется на область образовательных технологий. Это коснулось и традиционного для технических вузов подхода "лекция и лаборатория". Стилистически новую образовательную технологию можно свести к простой замене союза "и" дефисом — "лекция-лаборатория". Смысл же от этого радикально меняется. Образно говоря, "сухая теория" превращается в "зеленеющее древо жизни", что в данном случае означает: эффективность обучения существенно повышается, если теория осваивается в процессе самостоятельного исследования, возможности которого стали доступны с появлением персональных компьютеров.

Изучение цифровой обработки сигналов (ЦОС) во многих вузах фактически уже использует технологию "лекция-лаборатория". Настоящее учебное пособие — попытка авторов внести свой вклад в ее развитие. Многие важные аспекты и проблемы практического применения методов и алгоритмов ЦОС могут быть целостно восприняты и осмыслены только в процессе самостоятельного исследования посредством компьютерного моделирования.

Прежде всего, необходимо выбрать компьютерную технологию для моделирования ЦОС, такую, которая будет востребована в профессиональной деятельности выпускника вуза.

В настоящее время к общепризнанным универсальным мировым стандартам в области компьютерных технологий относится программная среда (система) MATLAB, предназначенная для моделирования в самых разных областях науки и техники, в первую очередь, ЦОС. Она была создана в США компанией The MathWorks, Inc. Информация о MATLAB доступна на сайтах **www.mathworks.com**, **www.softline.ru**, **www.matlab.ru** и **www.exponenta.ru**.

Широкое распространение MATLAB обусловлено следующими основными достоинствами этой системы:

- ☐ алгоритмическим языком "сверхвысокого" уровня за счет матричной обработки данных;
- ☐ колоссальной библиотекой стандартных функций с возможностью ее расширения функциями, создаваемыми пользователем;

- огромным разнообразием графических средств;
- удобными средствами создания и отладки программ;
- широким набором программных средств общего (ядро MATLAB) и специального (пакеты расширения Toolbox) назначения;
- наличием разнообразных средств GUI (Graphical User Interface — графический интерфейс пользователя) без использования алгоритмического языка в явном виде;
- широким набором средств Simulink общего (ядро Simulink) и специального назначения для блочного моделирования динамических систем.

В качестве альтернативы MATLAB часто называют другой общепризнанный мировой стандарт в области компьютерных технологий — среду графического программирования LabVIEW с закрытым программным кодом и обширной библиотекой функциональных элементов, в том числе виртуальных измерительных приборов.

Действительно, широкий круг задач по моделированию ЦОС можно решать и в MATLAB, и в LabVIEW. Однако в вопросе о предпочтении, в первую очередь, необходимо учитывать назначение данных технологий, изначально задуманное разработчиками.

"Виртуальная лаборатория" LabVIEW ориентирована на разработку аппаратуры (hardware) и обеспечивает сопряжение "железа" (плат и модулей) с программной средой по стандартным интерфейсам и протоколам (TCP/IP, GPIB-488, RS-232 и др.). Это позволяет тестировать реальную систему или ее программную имитацию с помощью виртуальных приборов на реальных сигналах в условиях, приближенных к реальным.

"Матричная лаборатория" MATLAB, в первую очередь, ориентирована на создание программных продуктов (software) на основе математических моделей и содержит мощные средства для моделирования методов и алгоритмов ЦОС.

Поэтому технологии компьютерного моделирования в MATLAB и LabVIEW следует рассматривать, скорее, как дополняющие друг друга, а не конкурирующие. Так, при моделировании систем ЦОС с использованием сложных математических моделей — безусловное преимущество на стороне MATLAB, а при проверке функционирования данной системы на реальных сигналах — на стороне LabVIEW, интегрированной с программами (script-файлами) MATLAB. Моделирование в LabVIEW максимально приближено к физическому макетированию — необходимому этапу перед реализацией системы "в железе".

Для обеспечения базовой подготовки по ЦОС в новый учебный план бакалавров по направлениям "Инфокоммуникационные технологии и системы связи" (210700) и "Радиотехника" (210400) включена обязательная дисциплина "Цифровая обработка сигналов".

Современный уровень профессиональной подготовки бакалавров и магистров предполагает развитие обширной темы ЦОС в рамках вариативной части учебного плана в дисциплинах базового цикла и далее в дисциплинах профессионального цикла, связанных с различными приложениями методов ЦОС.

Данное учебное пособие разработано для поддержки дисциплин ЦОС и посвящено изучению базовой теории ЦОС в процессе моделирования в MATLAB и одновременно — освоению средств моделирования, программных и GUI, для разработки программных продуктов (software).

Назначение учебного пособия определило его структуру.

Книга включает 21 главу и тематически разделена на две части:

- часть I "Знакомство с MATLAB" (главы 1—6);
- часть II "Моделирование цифровой обработки сигналов в MATLAB" (главы 7—21).

Часть I, посвященная освоению базовых средств языка MATLAB (ядра MATLAB), включена для облегчения работы начинающих пользователей, а для тех, кто с ними знаком, может рассматриваться как справочный материал.

Часть II охватывает все основные разделы базовой теории и компьютерного моделирования ЦОС: дискретные сигналы и линейные дискретные системы (ЛДС); дискретное преобразование Фурье (ДПФ) с использованием быстрых алгоритмов (БПФ); синтез и анализ КИХ- и БИХ-фильтров, в том числе с фиксированной точкой; непараметрический и параметрический спектральный анализ сигналов; много-скоростная обработка сигналов; адаптивные фильтры.

В учебно-методических целях главы второй части учебного пособия структурированы в виде лабораторных работ с типовыми разделами, включая:

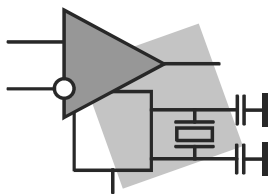
- цель работы и краткую теоретическую справку;
- исходные данные, пункты задания и вопросы по результатам исследования;
- типовой script-файл или описание GUI для выполнения задания;
- задание на самостоятельную работу по созданию собственных программ (function-файлов) или моделированию в GUI;
- контрольные вопросы.

На прилагаемом к учебному пособию компакт-диске хранятся все script-файлы и таблицы исходных данных с примерами их заполнения для первого варианта (всего 30 вариантов).

Для запуска script-файлов можно использовать версии MATLAB, начиная с R2009b.

Данная книга, в первую очередь, ориентирована на бакалавров, магистров, аспирантов и преподавателей вузов. Однако она может быть полезна для всех инженерно-технических специалистов, проявляющих интерес к области ЦОС.

Все предложения и замечания, которые будут приняты авторами с благодарностью, просим присылать в издательство "БХВ-Петербург" по электронному адресу: **mail@bhv.ru**.

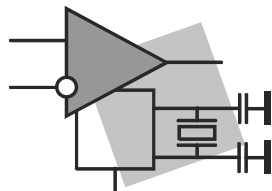


ЧАСТЬ I

Знакомство с MATLAB

- Глава 1.** Знакомство с MATLAB. Основные объекты языка MATLAB
- Глава 2.** Операции с матрицами
- Глава 3.** Типы массивов
- Глава 4.** Средства графики
- Глава 5.** Режим программирования: script-файлы и function-файлы
- Глава 6.** Режим программирования: организация разветвлений и циклов

ГЛАВА 1



Знакомство с MATLAB. Основные объекты языка MATLAB

Цель работы: познакомиться с назначением и интерфейсом системы MATLAB и овладеть начальными навыками работы в режиме прямых вычислений.

1.1. Краткая теоретическая справка

Система MATLAB — это интерактивная система, предназначенная для компьютерного моделирования практически в любой области науки и техники.

Интерфейс MATLAB образуют следующие окна.

- **Command Window** (Командное окно) — основное окно интерактивной системы MATLAB с активизированной командной строкой.

Из активизированной командной строки пользователь может возвращаться к предыдущим строкам с помощью клавиш $\langle \uparrow \rangle$ и $\langle \downarrow \rangle$.

Сеанс работы в окне **Command Window** до выхода из MATLAB называют *текущей сессией*.

- **Current Folder** (Текущая папка) — в этом окне выводится содержимое папки, имя которой отображается в раскрывающемся списке **Current Folder** на панели инструментов окна MATLAB.

В составе ранних версий MATLAB (до 2009 г.) содержалась автоматически создаваемая текущая папка со стандартным именем¹ *work*, предназначенная для хранения файлов и папок, создаваемых пользователем. В последующих версиях такая папка отсутствует. Для тех же целей предусмотрена папка MATLAB, автоматически создаваемая в папке **Мои документы** на рабочем столе.

Создание собственной папки в окне **Current Folder** выполняется с помощью контекстного меню по команде **New Folder** (Новая папка), и новой папке присваивается имя.

¹ Здесь и далее во избежание путаницы для папок и файлов MATLAB используется шрифт Courier New.

Сохранение пути к собственной папке в окне **Current Folder** выполняется по команде контекстного меню **Add to Path | Selected Folders** (Добавить к пути | Выделенные папки).

- ❑ **Workspace** (Рабочая область памяти) — в этом окне выводится список текущих переменных, сохраняемых в рабочей области памяти **Workspace** до выхода из **MATLAB**.
- ❑ **Command History** (История команд) — в этом окне выводится построчный список объектов языка **MATLAB**, вводимых в ходе текущей и предшествующих сессий. Двойным щелчком левой кнопки мыши можно дублировать любую строку из окна **Command History** в окно **Command Window**.

Пользователь может произвольно менять состав активных окон с помощью команд меню **Desktop** (Стол).

Система оперативной помощи **MATLAB** включает в себя:

- ❑ справочную систему в формате **HTML** (**HyperText Markup Language** — язык гипертекстовой разметки), обращение к которой производится по команде **Product Help** (Помощь по продукту) в меню **Help** окна **MATLAB**;
- ❑ команду:
`help <стандартное имя объекта языка MATLAB>`

1.1.1. Режим прямых вычислений

Режим прямых вычислений (называемый также командным режимом) означает, что вычисления выполняются без составления программы. Объекты языка **MATLAB** в ходе текущей сессии вводятся построчно в командной строке окна **Command Window** с соблюдением следующих правил:

- ❑ символ ";" (точка с запятой) в конце строки *блокирует автоматический вывод* результата;
- ❑ символ "..." (многоточие) в конце строки является признаком продолжения предыдущей строки;
- ❑ символ "%" (процент) в начале строки соответствует *комментарю*.

1.1.2. Базовые объекты языка MATLAB

К базовым объектам языка **MATLAB** относятся:

- ❑ команды;
- ❑ операторы;
- ❑ константы;
- ❑ переменные;
- ❑ функции;
- ❑ выражения.

Команда — это объект языка MATLAB со стандартным именем, предназначенный для взаимодействия с системой MATLAB и имеющий формат:

<команда> <содержательная часть>

где <команда> — стандартное имя команды; <содержательная часть> — уточняется для каждой конкретной команды и может отсутствовать.

В конце команды символ ";" не ставится.

Список команд общего назначения выводится по команде:

help general

Наиболее распространенные команды приведены в табл. 1.1. Другие будут рассматриваться по мере изложения материала.

Таблица 1.1. Команды

Команда	Назначение
clc	Очистка окна Command Window
clear	Удаление объектов из Workspace (без содержательной части — очистка Workspace)
format	Установка формата вывода данных (см. табл. 1.2)
help	Справка по стандартному объекту MATLAB
load	Загрузка файла с диска в Workspace (см. разд. 1.1.4)
save	Сохранение на диске объекта Workspace (см. разд. 1.1.4)
ver	Вывод информации об установленной версии MATLAB и пакетах расширения
what	Вывод содержимого папки (без содержательной части — текущей папки), например: what work\LAB\lab_01
which	Вывод пути для нахождения встроенной или внешней функции
who	Вывод содержимого Workspace
whos	Вывод содержимого Workspace с дополнительными сведениями

Оператор — это объект языка MATLAB со стандартным именем, предназначенный для разработки программ.

Простейшим оператором является оператор присваивания с форматом:

<имя переменной> = <выражение>

или

<выражение>

В последнем случае значение выражения присваивается переменной со стандартным именем ans.

Константа — это объект языка MATLAB, имеющий в процессе вычислений неизменное значение.

Различают следующие типы констант:

☐ численные, среди которых выделяют:

- целые;
- вещественные;
- комплексные;

☐ логические;

☐ символьные.

Целые и вещественные константы могут вводиться в обычной форме с разделением *точкой* целой и дробной частей:

```
>> 158;
>> -17.38;
```

или в форме E, которой соответствует представление числа в показательной форме:

$$\mu \cdot 10^p, \quad (1.1)$$

где μ — мантисса — вещественная константа; p — порядок — целая константа; 10 — основание, обозначаемое буквой e:

```
>> 0.157e-3;
>> 12.23e8;
```

Комплексные константы вводятся в алгебраической форме:

$$\xi + j\eta. \quad (1.2)$$

Мнимая единица вводится как i или j, но выводится всегда как i:

```
>> 5+3.7j
ans =
    5.0000 + 3.7000i
```

Возможен ввод с использованием символа умножения в мнимой части:

```
>> 5+3.7*j
ans =
    5.0000 + 3.7000i
```

Вещественная и/или мнимая части комплексного числа могут вводиться в форме E:

```
>> 5e-3+3.7e5j
ans =
    5.0000e-003 + 3.7000e+005i
```

Комплексно сопряженная константа вводится с помощью символа "'" (апостроф):

```
>> (5+3i) '
ans =
    5.0000 - 3.0000i
```

Вывод численных констант может производиться по умолчанию или в заданном формате с помощью команды:

```
format <вид формата>
```

где содержательная часть может отсутствовать.

Действие команды `format` сохраняется до ее отмены другой командой `format`.

Разновидности форматов можно вывести по команде:

```
help format
```

Наиболее распространенные форматы приведены в табл. 1.2.

Таблица 1.2. Форматы для вывода констант

Команда	Формат вывода
<code>format</code>	Формат, тождественный формату <code>format short</code>
<code>format short</code>	Формат с автоматическим выводом в обычной форме или нормализованной форме E с 4-мя значащими цифрами в дробной части мантиссы. Этот формат установлен по умолчанию
<code>format short e</code>	Короткий формат E с выводом в нормализованной форме E с 4-мя значащими цифрами в дробной части мантиссы
<code>format long</code>	Длинный формат с автоматическим выводом в обычной форме или нормализованной форме E с 15-ю значащими цифрами в дробной части мантиссы
<code>format long e</code>	Длинный формат с выводом в нормализованной форме E с 15-ю значащими цифрами в дробной части мантиссы

Форму E называют *нормализованной* (см. табл. 1.2), если целая часть мантиссы μ в (1.1) содержит одну отличную от нуля значащую цифру, а порядок p — три цифры.

Стандартные константы — это константы со стандартными именами. Их полный список может быть выведен по команде:

```
help elmat
```

Наиболее распространенные стандартные константы приведены в табл. 1.3.

Таблица 1.3. Стандартные константы

Стандартное имя константы	Назначение
<code>i</code> или <code>j</code>	Мнимая единица, соответствующая $\sqrt{-1}$: <code>i = sqrt(-1)</code>
<code>pi</code>	Число π

Таблица 1.3 (окончание)

Стандартное имя константы	Назначение
Inf (или inf)	Машинная бесконечность (число, большее максимально допустимого во внутренних вычислениях в MATLAB)
Nan	Не число (Not-a-number). Присваивается неопределенностям типа 0/0, inf/inf, 0·inf

Логические константы — это константы, принимающие значения 1 (`true` — истина) или 0 (`false` — ложь).

Символьные константы — это любые последовательности символов, заключенные в апострофы:

```
>> 'Sella'
ans =
Sella
```

Переменная — это объект языка MATLAB, который в процессе вычислений может менять свое значение.

Различают следующие типы переменных:

- простые переменные;
- массивы.

Переменные представляются своими именами (идентификаторами).

Имя переменной составляется из последовательности латинских букв, цифр и символа подчеркивания и начинается с буквы. В MATLAB *прописные и строчные буквы различаются*.

Массивом называют упорядоченную совокупность данных, объединенных одним именем.

Массив характеризуется:

- *размерностью*. Размерность массива равна количеству индексов k , которые указывают на упорядоченность данных в k -мерном пространстве.

Если данные упорядочены в строку (столбец), то их порядок следования указывается с помощью одного индекса, и массив называют одномерным или *вектором*.

Если данные упорядочены одновременно по строкам и по столбцам, то их порядок следования указывается с помощью двух индексов, и массив называют двумерным или *матрицей*.

Если данные упорядочены по матрицам, то их порядок следования указывается с помощью третьего индекса, и массив называют *трехмерным* и т. д.;

❑ *размером*. В матричной алгебре размер массива принято указывать произведением числа элементов по каждому индексу, а именно: $1 \times n$ — одномерный массив (вектор-строка); $m \times n$ — двумерный и т. д.

Матрицу называют *квадратной* порядка n , если число строк равно числу столбцов: $m = n$;

❑ *типом*. Тип массива определяется типом его элементов. Элементами числового массива являются численные константы. Основные типы массивов рассматриваются в гл. 3.

Особенностью MATLAB является то, что *тип переменной не объявляется, и любая переменная по умолчанию считается матрицей*¹.

В MATLAB *нижняя граница* индексов массива равна *единице*.

Матрица вводится построчно в квадратных скобках, элементы строки отделяются пробелом или запятой, а строки — точкой с запятой:

```
>> A = [1 2 3;5 6 7;8 9 7]
```

A =

1	2	3
5	6	7
8	9	7

Вектор (вектор-строка) размером $1 \times n$ вводится в квадратных скобках, а его элементы — через пробел или запятую:

```
>> A = [1 4 5 7 8]
```

A =

1	4	5	7	8

Вектор-столбец размером $m \times 1$ вводится в квадратных скобках, а его элементы — через точку с запятой:

```
>> A = [1;4;5]
```

A =

1
4
5

Скаляр размером 1×1 можно вводить без квадратных скобок:

```
>> b = 1.5e-3;
```

Простой переменной, таким образом, соответствует скаляр.

Функции в MATLAB представлены двумя разновидностями:

❑ *встроенные*;

❑ *внешние*.

Встроенная функция (по умолчанию функция) — это объект языка MATLAB со стандартным именем, предназначенный для выполнения действий с параметра-

¹ Отсюда и название MATLAB — MATrix LABoratory (Матричная лаборатория).

ми (аргументами), перечисленными через запятую и заключенными в круглые скобки.

Список основных элементарных математических функций, сгруппированных по назначению, представлен в табл. 1.4. Их полный список выводится по команде:

```
help elfun
```

Таблица 1.4. Элементарные математические функции

Тип функции	Функция	Назначение
Тригонометрическая	<code>sin (X)</code>	Синус — $\sin(x)$
	<code>cos (X)</code>	Косинус — $\cos(x)$
	<code>tan (X)</code>	Тангенс — $\operatorname{tg}(x)$
	<code>cot (X)</code>	Котангенс — $\operatorname{ctg}(x)$
Обратная тригонометрическая	<code>asin (X)</code>	Арксинус — $\arcsin(x)$
	<code>acos (X)</code>	Арккосинус — $\arccos(x)$
	<code>atan (X)</code>	Арктангенс — $\arctg(x)$
	<code>acot (X)</code>	Арккотангенс — $\operatorname{arcctg}(x)$
Экспоненциальная	<code>exp (X)</code>	Экспонента — e^x
	<code>pow2 (X)</code>	Возведение двойки в степень — 2^x
	<code>nextpow2 (X)</code>	Ближайшая степень двойки в сторону увеличения — $\operatorname{int}[\log_2(x)]$
Логарифмическая	<code>log (X)</code>	Натуральный логарифм — $\ln(x)$
	<code>log10 (X)</code>	Десятичный логарифм — $\lg(x)$
	<code>log2 (X)</code>	Логарифм по основанию 2 — $\log_2 x$
Корень квадратный	<code>sqrt (X)</code>	Корень квадратный $\sqrt{x}$
Число по модулю m	<code>mod (X,m)</code>	Число x по модулю m — $\operatorname{mod}_m x$
С комплексным аргументом	<code>abs (X)</code>	Модуль числа $ x $
	<code>angle (X)</code>	Аргумент числа x
	<code>complex (X1,X2)</code>	Запись комплексного числа по вещественной $X1$ и мнимой $X2$ частям
	<code>real (X)</code>	Выделение вещественной части — $\operatorname{Re}(x)$
	<code>imag (X)</code>	Выделение мнимой части — $\operatorname{Im}(x)$
	<code>conj (X)</code>	Комплексно сопряженное число

Таблица 1.4 (окончание)

Тип функции	Функция	Назначение
Округление	fix (X)	Округление в направлении нуля — усечение дробной части
	floor (X)	Округление в направлении $-\infty$ — округление до ближайшего целого в сторону уменьшения
	ceil (X)	Округление в направлении $+\infty$ — округление до ближайшего целого в сторону увеличения
	round (X)	Округление до ближайшего целого — при дробной части, равной 0.5, — в сторону увеличения модуля числа
	nearest (X)	Округление до ближайшего целого — при дробной части, равной 0.5, — в сторону увеличения
	convergent (X)	Округление до ближайшего целого — при дробной части, равной 0.5, — в сторону ближайшего четного числа

Список основных функций преобразования систем счисления представлен в табл. 1.5. Цифра 2 в имени этих функций соответствует английскому предлогу "to", переводимому как "в" или "к".

Таблица 1.5. Функции преобразования систем счисления

Функция	Назначение
dec2hex (X)	Преобразование десятичного целого в шестнадцатеричное. Десятичное число указывается в качестве аргумента, а шестнадцатеричное выводится без апострофов с использованием заглавных букв
dec2bin (X)	Преобразование десятичного целого в двоичное. Десятичное число указывается в качестве аргумента, а двоичное выводится без апострофов
bin2dec (X)	Преобразование двоичного целого в десятичное. Двоичное число указывается в качестве аргумента в апострофах, а десятичное выводится без апострофов
hex2dec (X)	Преобразование шестнадцатеричного целого в десятичное. Шестнадцатеричное число указывается в качестве аргумента в апострофах с использованием заглавных или строчных букв, а десятичное выводится без апострофов

Выражение — это объект языка MATLAB, представляющий собой имеющую смысл совокупность констант, переменных и функций, объединенных символами операций.

К основным типам выражений относятся арифметические и логические выражения.

Арифметическим выражением называют имеющую математический смысл совокупность констант, переменных и функций, объединенных символами (или функциями) арифметических операций:

```
>> x+sin(a)-sqrt(c+b);
```

Приоритет операций в арифметических выражениях устанавливается с помощью круглых скобок и старшинства операций внутри них, а именно: сначала вычисляются функции, затем возведение в степень, затем умножение и деление и в заключение — сложение и вычитание. Операции одного ранга выполняются слева направо.

Логическим выражением называют имеющую математический смысл совокупность арифметических выражений, объединенных символами (или функциями) операций отношения и логических операций:

```
>> (i==j) & ((a+b)>sqrt(c));
```

Простейшим логическим выражением является *отношение*. Результатом вычисления логического выражения будет логическая константа 1 (true) или 0 (false):

```
>> sin(3)<0.5  
ans =  
1
```

Приоритет операций в логических выражениях устанавливается с помощью круглых скобок и старшинства операций внутри них, а именно: сначала вычисляются арифметические выражения, затем выполняются операции отношения и в заключение — логические операции. Операции одного ранга выполняются слева направо.

Символ операции — это символическое обозначение операции с операндами или операндом (объектами, с которыми выполняется операция).

Функция операции — это эквивалентное обозначение символа операции в виде функции MATLAB.

Большинство символов операций дублируется эквивалентными функциями, однако некоторые операции обозначаются только символом, другие — только функцией.

Основные символы и дублирующие их функции операции, сгруппированные по назначению, представлены в табл. 1.6—1.8, где переменные x и y — числовые матрицы, а c — скаляр.

Полный список символов и функций операций выводится по команде:

```
help ops
```

Таблица 1.6. Символы и функции арифметических операций

Символ	Функция	Операция
+	<code>plus (X, Y)</code>	Сложение матричное и поэлементное
-	<code>minus (X, Y)</code>	Вычитание матричное и поэлементное
*	<code>mtimes (X, Y)</code>	Матричное умножение
.*	<code>times (X, Y)</code>	Поэлементное умножение
^	<code>mpower (X, c)</code>	Матричное возведение в целую степень
.^	<code>power (X, c)</code>	Поэлементное возведение в степень
\	<code>mldivide (X, Y)</code>	Левое матричное деление
/	<code>mrdivide (X, Y)</code>	Правое матричное деление
.\	<code>ldivide (X, Y)</code>	Левое поэлементное деление
./	<code>rdivide (X, Y)</code>	Правое поэлементное деление

Таблица 1.7. Символы и функции операций отношения

Символ	Функция	Операция
<code>==</code>	<code>eq (X, Y)</code>	Равно
<code>~=</code>	<code>ne (X, Y)</code>	Не равно
<code><</code>	<code>lt (X, Y)</code>	Меньше
<code>></code>	<code>gt (X, Y)</code>	Больше
<code><=</code>	<code>le (X, Y)</code>	Меньше либо равно
<code>>=</code>	<code>ge (X, Y)</code>	Больше либо равно

Таблица 1.8. Символы и функции логических операций

Символ	Функция	Операция
<code>&</code>	<code>and (X, Y)</code>	И (AND) — истина (<code>true</code> — логическая константа 1), если <i>оба</i> аргумента — истина
<code> </code>	<code>or (X, Y)</code>	ИЛИ (OR) — истина, если <i>хотя бы один</i> аргумент — истина
<code>~</code>	<code>not (X)</code>	НЕ (NOT) — ложь (<code>false</code> — логическая константа 0), если аргумент — истина, и наоборот

1.1.3. Рабочая область памяти **Workspace**

В MATLAB переменные текущей сессии хранятся в рабочей области памяти, называемой **Workspace**. Окно **Workspace**, открываемое по одноименной команде

в меню **Desktop**, содержит построчный список имен переменных (**Name**), каждую с ее символическим изображением и значением (**Value**) или размером и типом.

Двойной щелчок левой кнопки мыши на переменной в столбце **Name** или **Value** открывает окно **Variable Editor** (Редактор переменной), в котором наглядно отображается переменная и допускается ее редактирование.

1.1.4. Сохранение данных на диске

Для того чтобы в следующих сессиях воспользоваться данными текущей сессии, их можно сохранить на диске в файле с расширением `mat` по команде:

```
save <имя файла> <список переменных>
```

где:

`<имя файла>` — имя `mat`-файла; если оно не указано, то по умолчанию `mat`-файлу присваивается имя *первой* переменной из `<списка переменных>`, а сама первая переменная при этом не сохраняется; `<список переменных>` — список сохраняемых переменных, указываемых через пробел.

Данные — `mat`-файлы — по умолчанию сохраняются на диске в текущей папке. Например:

```
>> n = 1:100; x = sin(0.5*pi.*n); y = cos(0.5*pi.*n);  
>> save sigx n x y
```

Значения переменных `n`, `x`, `y` будут сохранены в файле `sigx.mat` в текущей папке.

По команде:

```
load <имя файла>
```

выполняется обратная процедура — загрузка данных (`mat`-файла) с диска в рабочее пространство памяти `Workspace`, например:

```
>> load sigx
```

Для систематизации сохраняемых файлов с различным назначением и расширением удобно создавать собственные папки.

1.2. Содержание лабораторной работы

Содержание работы связано с изучением режима прямых вычислений и базовых объектов языка MATLAB.

1.3. Задание на лабораторную работу

Задание на лабораторную работу включает в себя следующие пункты:

1. Запуск системы MATLAB и знакомство с ее интерфейсом.

Пояснить, какие окна образуют интерфейс MATLAB.

2. Знакомство со справочной системой MATLAB в формате HTML.

Пояснить, как обратиться к справочной системе.

3. Ввод комментария в окне **Command Window**.

Ввести наименование лабораторной работы.

Пояснить, какой символ используется для ввода комментария.

4. Знакомство с командами языка MATLAB.

Выполнить команду:

```
help general
```

Пояснить назначение и формат команды `help`.

О каких объектах языка MATLAB будет выдана справка?

5. Очистка окна **Command Window**.

Пояснить, какая команда используется.

6. Ввод вещественных констант.

Ввести следующие константы в обычной форме и форме E без символа ";" в конце строки:

0

0,000

0,814

−0,814

$8,14 \cdot 10^{-7}$

0,814578942

0,9999999999

0,0000814765178

8145,7

−8145,577777777

$0,814557 \cdot 10^5$

Пояснить:

- смысл символа ";" в конце строки;
- какой переменной присваиваются значения вводимых констант;
- в каком случае при вводе констант целесообразно использовать форму E;
- в каком формате выводятся константы по умолчанию;
- как вывести указанные константы с максимальным количеством значащих цифр в дробной части;
- какое количество значащих цифр в дробной части будет максимальным;

- какие форматы предусмотрены для вывода вещественных констант;
- какую форму E называют нормализованной.

7. Ввод комплексных констант.

Ввести следующие константы без символа ";" в конце строки:

0,057+0,5j

0,057+0,5i

1200000,5+56i

1200000,57857+56i

12,5+56i

12,5+0,000056i

-0,9999999i

0i

17+10⁻⁵i

15 · 10⁻⁵i

Пояснить:

- в какой форме вводятся комплексные константы;
- в какой форме вводятся их вещественные и мнимые части;
- в каком формате выводятся комплексные константы по умолчанию;
- какой формат целесообразно выбрать для вывода указанных констант;
- какая из констант списка будет воспринята как вещественная.

8. Ввод логических констант.

Ввести константы `true` и `false` без символа ";" в конце строки.

Пояснить, какие значения будут выведены и какой переменной присвоены.

9. Ввод символьных констант.

Ввести константы:

- ФИО;
- наименование лабораторной работы.

Пояснить, как вводятся и выводятся символьные константы.

10. Ввод векторов.

Ввести векторы — строки и столбцы — со следующими элементами:

-0,9; 125; 0; 5+3i; 12i;

-0,9; 125; 0; 5; 12;

1; 2; 4; 5; 12.

Пояснить:

- какие символы используются при вводе векторов;
- как в MATLAB воспринимаются скаляры и векторы.

11. Ввод матрицы.

Ввести матрицы 3×3 и 3×2 с произвольными элементами.

Пояснить, что называют размером и порядком матрицы.

12. Ввод переменных.

Присвоить произвольные значения простой переменной, вектору и матрице.

Пояснить, как выбираются имена переменных и как переменные воспринимаются в MATLAB.

13. Знакомство с особенностями ввода комплексных переменных.

Присвоить переменной i значение 5.

Присвоить переменной f значение комплексной константы $5+3i$, которую ввести двумя способами: без символа умножения в мнимой части; с символом умножения.

Пояснить:

- в каком из этих случаев и почему возникает ошибка;
- как предотвратить возникновение ошибок в подобных случаях.

14. Знакомство со стандартными функциями с комплексным аргументом.

Присвоить переменной произвольное комплексное значение.

Вычислить модуль, аргумент, вещественную и мнимую части переменной.

Присвоить другой переменной значение комплексно сопряженной константы.

Пояснить, какие стандартные функции для этого используются.

15. Ввод арифметических выражений.

Присвоить переменным a , b и c значения произвольных вещественных констант, не равных нулю.

Вычислить значения переменных d и e по следующим формулам:

$$d = a + b \sin(\pi/a + b/c - \cos \pi);$$

$$e = a^2 - \sqrt{|b|} + \sqrt[3]{c} + \frac{d + ac}{b}.$$

Пояснить:

- приоритет выполнения операций в арифметических выражениях;
- что является результатом вычисления арифметического выражения.

16. Ввод логических выражений.

Используя переменные предыдущего пункта, записать логическое выражение с использованием операций отношения и вычислить его значение.

Добавить в данное выражение логические операции и вычислить значение нового логического выражения.

Пояснить:

- приоритет выполнения операций в логических выражениях;
- что является результатом вычисления логического выражения.

17. Знакомство со стандартными переменными.

Ввести арифметические выражения, которым по умолчанию будут присвоены константы `Nan` и `Inf`.

Пояснить назначение данных констант.

18. Знакомство со стандартными функциями округления.

Выполнить следующие вычисления:

```
floor([8.2 8.5 8.7 -8.2 -8.5 -8.7])  
ceil([8.2 8.5 8.7 -8.2 -8.5 -8.7])  
convergent([8.2 8.5 8.7 -8.2 -8.5 -8.7])  
nearest([8.2 8.5 8.7 -8.2 -8.5 -8.7])  
round([8.2 8.5 8.7 -8.2 -8.5 -8.7])  
fix([8.2 8.5 8.7 -8.2 -8.5 -8.7])
```

Привести и пояснить полученные результаты.

19. Знакомство со стандартными функциями преобразования систем счисления.

Записать произвольное целое десятичное число и преобразовать его в шестнадцатеричное и двоичное.

Выполнить обратные преобразования.

Пояснить, какие стандартные функции использовались для преобразования.

20. Сохранение переменных на диске.

Присвоить переменным `A`, `B` и `C` произвольные значения и сохранить их в текущей папке в файле с произвольным именем.

Пояснить:

- какая команда используется для сохранения данных;
- как выбирается имя файла данных;
- какое расширение имеют файлы данных.

21. Знакомство с рабочим пространством памяти `Workspace`.

Выполнить следующие действия:

- очистить и проверить содержимое `Workspace`;
- загрузить сохраненный файл данных (см. п. 20) и вывести значение переменных `A`, `B`, `C` в окне **Command Window**;

- проверить содержимое Workspace;
- удалить из Workspace переменную *a* и проверить содержимое Workspace.

Пояснить назначение Workspace и выполняемые команды.

22. Завершение работы MATLAB.

1.4. Задание на самостоятельную работу

Самостоятельное задание рекомендуется для закрепления полученных знаний и включает в себя следующие пункты:

1С. Ввод вещественных констант.

Привести примеры ввода вещественных констант, для которых удобен обычный формат и формат E, а также тех, для которых, независимо от формы ввода, количество значащих цифр после запятой будет ограничено.

2С. Операции с комплексными константами.

Ввести вещественные константы:

i = 7; *j* = 5;

и определить, в каком из следующих случаев будут выведены комплексные константы:

$(5+7i) * (5+7*j)$

$(5+7*i) * (5+7*j)$

$(5+7i) * (5+7j)$

i = sqrt(-1); $(5+7*i) * (5+7j)$

j = sqrt(-1); $(5+7*i) * (5+7*j)$

3С. Вычисление арифметических выражений.

Присвоить простым переменным *a*, *b* и *c* произвольные значения и записать арифметические выражения для вычислений по следующим формулам:

$$c^{2+b} \frac{a+b}{a-b} + \sqrt{\frac{a-b}{a+b}} e^{\frac{a-b}{a+b}};$$

$$c^2 + b \frac{a+b}{a-b} \cdot \frac{\sqrt{a} - \sqrt[3]{b}}{a^2 + b^3};$$

$$\frac{a+b}{a-b} + c^{1/3} \sin \frac{a-b}{a+b}.$$

4С. Для четырех комбинаций логических констант *x* и *y* (00, 01, 10 и 11) вычислить значения логического выражения (составить таблицу истинности):

$$f = \overline{xy} \vee (\overline{x} \vee y \vee xy),$$

где символу " $\vee$ " соответствует логическая операция "ИЛИ".

5С. Операции со стандартными функциями.

Привести пример арифметического выражения с использованием стандартных функций, включая функции округления.

1.5. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит результаты выполнения каждого пункта задания, копируемые из окна **Command Window** (шрифт Courier New), и ответы на поставленные вопросы (шрифт Times New Roman).

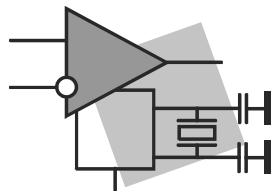
Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. Для чего предназначена система MATLAB?
2. Назовите окна интерфейса MATLAB и поясните их назначение.
3. Дайте определение следующим понятиям: текущая сессия, режим прямых вычислений.
4. Назовите базовые объекты языка MATLAB.
5. Дайте определение команды.
6. Дайте определение константы.
7. Какие типы констант используются в MATLAB?
8. Как вводятся комплексные константы?
9. Какие форматы вывода констант используются в MATLAB?
10. Дайте определение формы E и нормализованной формы E.
11. Какие константы называют стандартными?
12. Дайте определение переменной и поясните, с помощью какого оператора ей присваивается значение.
13. Дайте определение массива.
14. Чем характеризуется массив?
15. Дайте расшифровку названия "MATLAB" и поясните его смысл.
16. Как вектор и скаляр воспринимаются в MATLAB?
17. Чему равна нижняя граница индексов матрицы в MATLAB?
18. Как вводятся матрица, вектор и скаляр?
19. Чему соответствует простая переменная в MATLAB?
20. Дайте определение выражения в MATLAB.
21. Какие типы выражений используются в MATLAB?
22. Дайте определение арифметического и логического выражений.

1.6. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Главы 1—2.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Приложения 1—2.

ГЛАВА 2



Операции с матрицами

Цель работы: овладеть навыками матричной обработки данных в MATLAB.

2.1. Краткая теоретическая справка

Как уже говорилось ранее (см. разд. 1.1.2), в MATLAB любая переменная по умолчанию считается *матрицей*.

Матрица представляется своим *именем* (идентификатором) и характеризуется размером и типом.

Размер матрицы принято указывать произведением $m \times n$, где m , n — число строк и столбцов соответственно. Матрицу размером $n \times n$ называют *квадратной* порядка n . *Вектор* воспринимается как матрица размером $1 \times n$ (строка) или $m \times 1$ (столбец), а *скаляр* — как матрица размером 1×1 .

Хранение матриц в оперативной памяти организовано *по столбцам*.

Тип матрицы определяется типом ее элементов. В этой работе по умолчанию под матрицей будем подразумевать *числовую* матрицу.

Ввод матриц рассматривался в разд. 1.1.2.

Вектор, формирующий *регулярную сетку*, вводят в виде:

<начальное значение>: <шаг>: <конечное значение>

Шаг, равный единице, можно не указывать, условным признаком чего служат квадратные скобки.

Например:

```
>> y = 0:pi/4:pi
```

```
y =  
    0    0.7854    1.5708    2.3562    3.1416
```

```
>> x = 0:9
```

```
x =  
    0    1    2    3    4    5    6    7    8    9
```

Элементы матрицы могут быть представлены численными константами, простыми переменными, арифметическими выражениями и, в свою очередь, матрицами, например:

```
>> A = [5.3 sin(pi/4) 3+4i]
A =
    5.3000         0.7071    3.0000 + 4.0000i
>> a = [1 2;3 4], b=[4 5;6 7]
a =
     1     2
     3     4
b =
     4     5
     6     7
>> B = [a b]
B =
     1     2     4     5
     3     4     6     7
```

Обращение к элементу матрицы происходит по ее имени с указанием индексов строки и столбца в круглых скобках (нижняя граница индексов равна *единице*):

```
>> B(2,4)
ans =
     7
```

По обращению $B(i)$ матрица B воспринимается как *вектор*, элементы которого сформированы по столбцам:

```
>> B(5)
ans =
     4
```

Размер матрицы — число строк и столбцов — определяется с помощью функции:

size(x)

Длина вектора — число элементов строки (столбца) — определяется с помощью функции:

length(x)

Матрица нулевой размерности — пустая матрица — обозначается как $A = []$:

```
>> A = []; size(A)
ans =
     0     0
```

Имя пустой матрицы сохраняется в Workspace и в дальнейшем может использоваться для формирования матрицы любого размера.

2.1.1. Функции генерации типовых матриц

В MATLAB можно генерировать большое разнообразие типовых матриц с помощью встроенных функций, список которых может быть выведен по команде:

```
help elmat
```

Некоторые из них приведены в табл. 2.1.

Таблица 2.1. Функции генерирования типовых матриц

Функция	Типовая матрица
<code>zeros (M,N)</code>	Нулевая матрица $M \times N$
<code>ones (M,N)</code>	Матрица единиц $M \times N$
<code>eye (N)</code>	Единичная матрица порядка N
<code>rand (M,N)</code>	Матрица $M \times N$ случайных чисел в диапазоне от 0 до 1, распределенных по <i>равномерному</i> закону
<code>randn (M,N)</code>	Матрица $M \times N$ случайных чисел, распределенных по <i>нормальному</i> закону с математическим ожиданием, равным 0, и дисперсией, равной 1
<code>diag (V)</code>	Вектор из диагональных элементов квадратной матрицы V . Диагональная матрица — матрица, у которой все элементы равны нулю, кроме диагональных, равных вектору V
<code>toeplitz (r)</code>	Матрица Топлица — квадратная матрица с одинаковыми элементами на диагоналях, равными соответствующим элементам первого столбца r

2.1.2. Преобразование матриц

К операциям преобразования матриц относятся:

- выделение из матрицы вектора-столбца:

```
A(:,N)
```

где N — номер столбца;

- выделение из матрицы вектора-строки:

```
A(M,:)
```

где M — номер строки;

- выделение подматрицы с указанием граничных индексов:

```
A(M1:M2,N1:N2)
```

где $M1:M2$ — номера строк с $M1$ по $M2$ включительно; $N1:N2$ — номера столбцов с $N1$ по $N2$ включительно;

- выделение подматрицы с указанием начальных индексов:

A(M1:end;N1:end)

где M1:end — строки с M1 до последней включительно; N1:end — столбцы с N1 до последнего включительно;

- растягивание матрицы в вектор-столбец:

A(:)

- горизонтальная конкатенация (объединение) подматриц (по столбцам):

A = [A1,A2,A3,...]

где A1,A2,A3,... — объединяемые подматрицы с одинаковым числом строк;

- вертикальная конкатенация подматриц (по строкам):

A = [A1;A2;A3;...]

где A1;A2;A3;... — объединяемые подматрицы с одинаковым числом столбцов;

- копирование матрицы, выполняемое с помощью функции:

repmat (A,m,n)

где A — исходная матрица как элемент новой матрицы; m, n — число копий матрицы A по строкам и столбцам соответственно;

- копирование квадратных матриц, выполняемое с помощью функции:

repmat (A,n)

где A — исходная квадратная матрица как элемент новой квадратной матрицы; n — число копий матрицы A по строкам и столбцам.

2.1.3. Поэлементные операции с матрицами

К поэлементным операциям с матрицами относятся арифметические операции и вычисление элементарных функций, аргументы которых — матрицы.

Признаком поэлементных арифметических операций умножения, деления и возведения в степень является *точка* перед символом операции:

```
>> A = [1 2 3;2 -1 5;1 -1 -1], B = [-1 -2 -3;-4 -5 -6;-7 -8 -9]
```

A =

```
1     2     3
2    -1     5
1    -1    -1
```

B =

```
-1    -2    -3
-4    -5    -6
-7    -8    -9
```

```
>> A.*B
ans =
    -1    -4    -9
    -8     5   -30
    -7     8     9

>> A./sin(B)
ans =
   -1.1884   -2.1995  -21.2585
    2.6427   -1.0428   17.8945
   -1.5221    1.0108    2.4265
```

2.1.4. Операции с матрицами в задачах линейной алгебры

К простейшим операциям с матрицами в задачах линейной алгебры относятся:

- ☐ арифметические операции;
- ☐ транспонирование и эрмитово сопряжение;
- ☐ обращение;
- ☐ матричное деление.

2.1.4.1. Арифметические операции с матрицами

К арифметическим операциям с матрицами относятся:

- ☐ сложение и вычитание матриц одинакового размера.

Суммой (разностью) матриц **A** и **B** размером $m \times n$ называется матрица **C** того же размера с элементами, равными сумме (разности) соответствующих элементов матриц **A** и **B**. Для операций сложения и вычитания матриц справедливы обычные законы арифметики:

$$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A};$$

$$\mathbf{A} - \mathbf{B} = -\mathbf{B} + \mathbf{A}.$$

Пример сложения матриц:

```
>> A = [1 2 3; 2 -1 5; 1 -1 -1]; B = [-1 -2 -3; -4 -5 -6; -7 -8 -9];
>> C = A+B
C =
     0     0     0
    -2    -6    -1
    -6    -9   -10
```

- ☐ умножение матрицы на скаляр (число), эквивалентное операции поэлементного умножения на скаляр;
- ☐ умножение матрицы на матрицу.

Операция умножения возможна только в том случае, если число столбцов матрицы **A** равно числу строк матрицы **B**.

Произведением матрицы **A** размером $m \times n$ на матрицу **B** размером $n \times p$ называется матрица **C** размером $m \times p$, элемент i -й строки и k -го столбца которой равен сумме произведений соответственных элементов i -й строки матрицы **A** и k -го столбца матрицы **B**:

$$c_{ik} = \sum_{j=1}^n a_{ij} b_{jk}; \quad i = 1, 2, \dots, m; \quad k = 1, 2, \dots, p.$$

В общем случае умножение матриц не коммутативно:

$$\mathbf{AB} \neq \mathbf{BA}.$$

Пример умножения матриц:

```
>> D = [1 2 5 7; 3 8 0 3]
D =
     1     2     5     7
     3     8     0     3

>> size(D)
ans =
     2     4

>> F = [1 2 0; 3 8 5; 0 3 4; 9 7 1]
F =
     1     2     0
     3     8     5
     0     3     4
     9     7     1

>> size(F)
ans =
     4     3

>> D*F
ans =
    70    82    37
    54    91    43
```

- возведение квадратной матрицы в целую степень q , эквивалентное умножению матрицы саму на себя q раз:

```
>> A = [1 2; 4 5]
A =
     1     2
     4     5

>> B = A^3
B =
    57    78
   156   213
```

2.1.4.2. Транспонирование и эрмитово сопряжение матриц

Транспонирование матрицы — это операция замены каждой строки столбцом с тем же номером.

Эрмитово сопряжение матрицы — это операция транспонирования матрицы с одновременной заменой ее элементов на комплексно сопряженные.

Операции транспонирования и эрмитова сопряжения выполняются *с помощью одного и того же символа* "'" (апостроф). Результат зависит от исходной матрицы — является она вещественной или комплексной. В первом случае получим транспонированную, а во втором — эрмитово сопряженную матрицу:

```
>> A = [1 2 3;4 5 6;7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9

>> A'
ans =
     1     4     7
     2     5     8
     3     6     9

>> C = [3+2i 4-5i;7-5i 1+i;2+2i 1-8i]
C =
 3.0000 + 2.0000i  4.0000 - 5.0000i
 7.0000 - 5.0000i  1.0000 + 1.0000i
 2.0000 + 2.0000i  1.0000 - 8.0000i

>> C'
ans =
 3.0000 - 2.0000i  7.0000 + 5.0000i  2.0000 - 2.0000i
 4.0000 + 5.0000i  1.0000 - 1.0000i  1.0000 + 8.0000i
```

Транспонирование комплексной матрицы выполняется с помощью символа **позлементного транспонирования**:

```
>> C.'
ans =
 3.0000 + 2.0000i  7.0000 - 5.0000i  2.0000 + 2.0000i
 4.0000 - 5.0000i  1.0000 + 1.0000i  1.0000 - 8.0000i
```

Матрицу с комплексно сопряженными элементами можно получить путем транспонирования эрмитово сопряженной матрицы или с помощью функции `conj` (см. табл. 1.4):

```
>> (C').'
ans =
 3.0000 - 2.0000i  4.0000 + 5.0000i
 7.0000 + 5.0000i  1.0000 - 1.0000i
 2.0000 - 2.0000i  1.0000 + 8.0000i

>> conj(C)
ans =
 3.0000 - 2.0000i  4.0000 + 5.0000i
 7.0000 + 5.0000i  1.0000 - 1.0000i
 2.0000 - 2.0000i  1.0000 + 8.0000i
```

2.1.4.3. Обращение матриц

Матрица **B** называется *обратной* к матрице **A**, если произведение этих матриц дает единичную матрицу **I**:

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}.$$

Матрица, обратная к матрице **A**, обозначается как $\mathbf{A}^{-1}$.

Операция обращения возможна только для квадратных матриц с определителем (детерминантом), не равным нулю.

Определитель матрицы вычисляется с помощью функции:

`det (A)`

а обратная матрица — с помощью функции:

`inv (A)`

Например:

```
>> A = [1 2; 4 5]
A =
     1     2
     4     5
>> det (A)
ans =
    -3
>> inv (A)
ans =
   -1.6667    0.6667
    1.3333   -0.3333
```

2.1.4.4. Матричное деление

В списке символов арифметических операций содержатся два символа матричного деления с *квадратными* матрицами **A** и **B** порядка *n* (см. табл. 1.6):

- левое матричное деление — $\mathbf{A} \setminus \mathbf{B}$, эквивалентное алгебраической операции $\mathbf{A}^{-1}\mathbf{B}$, т. е. `inv (A) * B`;
- правое матричное деление — $\mathbf{A} / \mathbf{B}$, эквивалентное алгебраической операции $\mathbf{AB}^{-1}$, т. е. `A * inv (B)`.

Символ левого матричного деления "\" используют при решении систем линейных алгебраических уравнений (СЛАУ):

$$\mathbf{Ax} = \mathbf{b}, \quad (2.1)$$

где **A** — матрица коэффициентов при неизвестных; **b**, **x** — векторы-столбцы свободных членов и неизвестных соответственно.

Умножив обе части (2.1) на $\mathbf{A}^{-1}$ слева, получим решение системы в виде:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}, \quad (2.2)$$

что в MATLAB соответствует выполнению операций $\text{inv}(A) * B$, т. е. *левому* матричному делению:

```
>> X=A\B
```

Пример решения системы уравнений (2.1):

$$\begin{cases} x_1 + 5x_2 = 4; \\ -x_1 + 7x_2 = 8. \end{cases}$$

```
>> A = [1 5;-1 7], B = [4 8]
```

```
A =
```

```
     1     5
```

```
    -1     7
```

```
B =
```

```
     4     8
```

```
>> X = A\B'
```

```
X =
```

```
    -1
```

```
     1
```

Проверим правильность решения по (2.1) — получим вектор-столбец B:

```
>> A*X
```

```
ans =
```

```
     4
```

```
     8
```

Деление B/A будет ошибочным, т. к. эта операция соответствует BA^{-1} ($B * \text{inv}(A)$), а умножение матриц в общем случае *не коммутативно*: $A^{-1}B \neq BA^{-1}$ (см. разд. 2.1.4.1):

```
>> B/A
```

```
ans =
```

```
 3.0000   -1.0000
```

2.1.5. Норма матрицы и вектора

Норма матрицы (вектора) — это скаляр, с помощью которого оцениваются значения элементов матрицы (вектора).

Среди норм матрицы A и вектора X выделим следующие основные:

□ норма $\|A\|_1$, определяется как максимальная сумма модулей элементов в *столбце*:

$$\|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|. \quad (2.3)$$

Аналогичная норма для вектора равна сумме модулей элементов вектора:

$$\|x\|_1 = \sum_{i=1}^m |x_i|; \quad (2.4)$$

□ норма $\|\mathbf{A}\|_\infty$ определяется как максимальная сумма модулей элементов *в строке*:

$$\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^m |a_{ij}|. \quad (2.5)$$

Аналогичная норма для вектора равна максимальному элементу вектора:

$$\|\mathbf{x}\|_\infty = \max_i |x_i|; \quad (2.6)$$

□ норма $\|\mathbf{A}\|_2$ (евклидова норма) определяется как корень квадратный из суммы квадратов модулей всех элементов матрицы:

$$\|\mathbf{A}\|_2 = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}. \quad (2.7)$$

Аналогичная норма для вектора:

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^m |x_i|^2}. \quad (2.8)$$

Норма матрицы и вектора вычисляется с помощью функции:

`norm(A, p)`

где `p` — параметр, указывающий норму и принимающий значения: 1 — для $\|\mathbf{A}\|_1$; 2 — для $\|\mathbf{A}\|_2$ (по умолчанию); `inf` — для $\|\mathbf{A}\|_\infty$:

```
>> A = [1 2; 4 5]
A =
     1     2
     4     5
>> NORMS = [norm(A,1) norm(A) norm(A,inf)]
NORMS =
     7.0000     6.7678     9.0000
```

2.1.6. Операции с матрицами в задачах математической статистики

Для решения задач математической статистики предусмотрен набор встроенных функций, список которых может быть выведен по команде:

`help datafun`

Основные из них приведены в табл. 2.2.

Таблица 2.2. Функции математической статистики

Функция	Назначение
<code>max (A)</code>	Максимальные элементы столбца
<code>min (A)</code>	Минимальные элементы столбца
<code>sort (A)</code>	Сортировка элементов столбца по возрастанию
<code>sum (A)</code>	Сумма элементов столбца
<code>prod (A)</code>	Произведение элементов столбца
<code>mean (A)</code>	Математическое ожидание (среднее значение) элементов столбца
<code>std (A)</code>	Среднеквадратическое (стандартное) отклонение (СКО), вычисляемое по формуле: $\text{СКО} = \sqrt{\frac{(a_{1j} - \bar{a}_j)^2 + (a_{2j} - \bar{a}_j)^2 + \dots + (a_{mj} - \bar{a}_j)^2}{(m - 1)}},$ где: a_{ij}, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$ — элемент матрицы A; $\bar{a}_j$, $j = 1, 2, \dots, n$ — математическое ожидание (среднее значение) элементов j-го столбца; m — количество строк, n — количество столбцов
<code>std (A, 1)</code>	Среднеквадратическое (стандартное) отклонение (СКО), вычисляемое по формуле: $\text{СКО} = \sqrt{\frac{(a_{1j} - \bar{a}_j)^2 + (a_{2j} - \bar{a}_j)^2 + \dots + (a_{mj} - \bar{a}_j)^2}{m}}$
<code>var (A)</code>	Дисперсия элементов столбца, вычисляемая по формуле: $\sigma_j^2 = \frac{(a_{1j} - \bar{a}_j)^2 + (a_{2j} - \bar{a}_j)^2 + \dots + (a_{mj} - \bar{a}_j)^2}{(m - 1)}$
<code>var (A, 1)</code>	Дисперсия элементов столбца, вычисляемая по формуле: $\sigma_j^2 = \frac{(a_{1j} - \bar{a}_j)^2 + (a_{2j} - \bar{a}_j)^2 + \dots + (a_{mj} - \bar{a}_j)^2}{m}$

2.2. Содержание лабораторной работы

Содержание работы связано с изучением типовых операций с матрицами в MATLAB в режиме прямых вычислений.

2.3. Задание на лабораторную работу

Задание на лабораторную работу включает в себя следующие пункты:

1. Определение длины вектора и размера матрицы.

Ввести:

- произвольную матрицу A ;
- пустую матрицу Z ;
- вектор v в виде регулярной сетки с начальным элементом $-\pi$, конечным π и шагом $\pi/32$.

Определить размеры матриц и длину вектора.

Пояснить:

- что такое длина вектора и размер матрицы и как они определяются в MATLAB;
- с какой целью и как вводится пустая матрица и каков ее размер;
- как вводится регулярная сетка; в каком случае допускается не указывать шаг изменения значения переменной.

2. Генерирование типовых матриц.

Сгенерировать следующие квадратные матрицы 3-го порядка:

- нулевую матрицу C ;
- матрицу единиц D ;
- единичную матрицу $D1$;
- матрицу Теплица T с произвольными вещественными элементами первого столбца;
- матрицу E случайных чисел, распределенных по равномерному закону;
- матрицу F случайных чисел, распределенных по нормальному закону.

Пояснить:

- как выполняется генерация указанных матриц в MATLAB;
- что собой представляют матрицы: нулевая, единиц, единичная и Теплица.

3. Выделение элементов матрицы.

В матрице F (см. п. 2) выделить:

- второй элемент третьей строки;
- вектор диагональных элементов;
- первую строку;
- третий столбец;
- подматрицу с номерами строк 2:3 и номерами столбцов 1:3.

Пояснить, как происходит выделение подматриц.

4. Преобразование матриц.

Произвести с матрицей F (см. п. 2) следующие преобразования:

- выполнить горизонтальную конкатенацию матрицы F с матрицей C (см. п. 2);
- выполнить вертикальную конкатенацию матрицы F с матрицей D (см. п. 2);
- сформировать квадратную матрицу G 6-го порядка посредством копирования матрицы F .

Пояснить, как выполняются указанные преобразования.

5. Поэлементные операции с матрицами.

Для всех элементов матрицы F (см. п. 2) выполнить операцию возведения в квадрат и умножения на 2.

Пояснить, какие символы арифметических операций использованы.

6. Сложение и вычитание матриц.

Ввести квадратные матрицы A и B 3-го порядка с произвольными вещественными элементами.

Выполнить операции сложения и вычитания матриц A и B и присвоить результаты переменным $C1$ и $C2$.

Пояснить:

- что собой представляют переменные $C1$ и $C2$;
- является ли операция сложения (вычитания) матриц коммутативной.

7. Умножение матриц.

Ввести матрицы A и B с произвольными вещественными элементами. Размеры матриц выбрать так, чтобы для этих матриц была возможна операция умножения.

Выполнить умножение матрицы A на матрицу B и присвоить результат переменной C .

Пояснить:

- как должны быть согласованы размеры матриц A и B , чтобы для них была возможна операция умножения;
- что собой представляет переменная C ;
- является ли операция умножения матриц коммутативной.

8. Транспонирование и эрмитово сопряжение матриц.

Выполнить следующие операции:

- транспонировать матрицу F (см. п. 2);
- сформировать квадратную матрицу E 3-го порядка с произвольными комплексными элементами;
- транспонировать матрицу E ;

- сформировать матрицу R , эрмитово сопряженную с матрицей F ;
- сформировать матрицу $R1$ с комплексно сопряженными элементами относительно матрицы R .

Пояснить, как указанные операции выполняются в MATLAB.

9. Обращение матриц.

Выполнить следующие операции:

- вычислить определитель матрицы F (см. п. 2);
- сформировать матрицу $F1$, обратную к матрице F ;
- найти произведение матриц F и $F1$ и присвоить результат переменной $F2$.

Пояснить:

- для какой матрицы возможна операция обращения;
- какая функция служит для вычисления определителя матрицы;
- что собой представляет переменная $F2$ и с какой целью она вычислена.

10. Решить СЛАУ

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 14; \\ 2x_1 - x_2 - 5x_3 = -15; \\ x_1 - x_2 - x_3 = -4 \end{cases}$$

и проверить правильность решения.

Пояснить, какая операция матричного деления используется и почему.

11. Вычисление норм матрицы и вектора.

Для матрицы F (см. п. 2) вычислить нормы (2.3), (2.5) и (2.7).

Для вектора $x = \text{rand}(1, 100)$ вычислить нормы (2.4), (2.6) и (2.8).

Пояснить смысл указанных норм и способ их вычисления в MATLAB.

12. Операции с матрицами в задачах математической статистики.

Для матрицы F (см. п. 2) вычислить:

- максимальные и минимальные элементы столбцов;
- сумму и произведение элементов столбцов;
- средние значения элементов столбцов;
- СКО элементов столбцов с помощью функции `std(F, 1)`;
- дисперсию элементов столбцов с помощью функции `var(F, 1)`.

Пояснить:

- какие функции MATLAB использованы в каждом из этих случаев (кроме `std` и `var`);
- как проверить правильность результатов согласно определению среднего значения, СКО и дисперсии.

2.4. Задание на самостоятельную работу

Самостоятельное задание рекомендуется для закрепления полученных знаний и включает в себя следующие пункты:

1С. Операции с матрицами.

Привести пример арифметического выражения, в котором все переменные и результат вычисления — матрицы.

Вычислить статистические характеристики и нормы результирующей матрицы.

2С. Обращение, транспонирование и эрмитово сопряжение матрицы.

Выполнить для матрицы Теплица 5-го порядка.

3С. Решение СЛАУ:

$$\begin{cases} 5x_1 + 2x_2 - 7x_3 - 0,5x_4 + 9 = 0; \\ x_1 - 0,3x_2 + 9x_3 + 5 = 0; \\ 6x_1 + x_2 - 8x_3 - 19 = 0; \\ 3x_2 + 4x_3 - 2x_4 = 0 \end{cases}$$

с проверкой правильности решения.

2.5. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит результаты выполнения каждого пункта задания, копируемые из окна **Command Window** (шрифт Courier New), и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

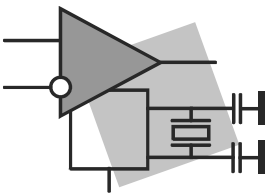
1. Дайте определение матрицы.
2. Что такое размер и порядок матрицы?
3. Как вектор и скаляр воспринимаются в MATLAB?
4. Чем определяется тип матрицы?
5. Как вводятся матрица, вектор и скаляр?
6. Чему равна нижняя граница индексов матрицы в MATLAB?
7. Как обратиться к элементу матрицы и вектора?
8. Что такое пустая матрица? С какой целью она вводится и каков ее размер?
9. Что такое регулярная сетка и как она вводится в MATLAB?
10. Как определить размер матрицы и длину вектора в MATLAB?
11. Как в MATLAB сформировать следующие матрицы: нулевую; единиц; единичную; случайных чисел, распределенных по равномерному и нормальному законам?

12. Как из матрицы выделить вектор-строку и вектор-столбец?
13. Как из матрицы выделить подматрицу с произвольными граничными индексами? С произвольными начальными индексами?
14. Как матрицу растянуть в вектор-столбец?
15. Как выполнить копирование матрицы?
16. Какие символы используются для поэлементных арифметических операций с матрицами?
17. Что такое транспонирование матрицы и как оно выполняется в MATLAB?
18. Что такое эрмитово сопряжение матрицы и как оно выполняется в MATLAB?
19. Дайте определение обратной матрицы и поясните, как она вычисляется в MATLAB.
20. Как в MATLAB вычислить определитель матрицы?
21. Для каких матриц допустима операция матричного умножения и какой символ операции используется?
22. Какие символы матричного деления используются в MATLAB и чем они отличаются? Какой из них используется при решении СЛАУ?
23. Как в MATLAB вычисляются различные нормы матрицы и вектора?
24. Как в MATLAB вычисляются среднее значение, дисперсия и СКО матрицы?

2.6. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 3.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Приложения 1—2.

ГЛАВА 3



Типы массивов

Цель работы: изучить основные типы массивов, используемых в MATLAB, и овладеть навыками их формирования.

3.1. Краткая теоретическая справка

В MATLAB тип массива (тип данных) определяется типом его элементов. По умолчанию мы имели дело с матрицами, элементы которых представлены константами в форме с плавающей точкой (ПТ) с двойной точностью. Такие массивы относятся к типу `double`.

Основные типы числовых и нечисловых массивов представлены в табл. 3.1.

Таблица 3.1. Типы массивов в MATLAB

Символическое обозначение типа массива	Тип массива	Функция преобразования типа
	Числовой:	
<code>double</code>	вещественный двойной точности	<code>double(X)</code>
<code>single</code>	вещественный одинарной точности	<code>single(X)</code>
<code>int8</code>	целый 8-битовый со знаком	<code>int8(X)</code>
<code>uint8</code>	целый 8-битовый без знака	<code>uint8(X)</code>
<code>int16</code>	целый 16-битовый со знаком	<code>int16(X)</code>
<code>uint16</code>	целый 16-битовый без знака	<code>uint16(X)</code>
<code>int32</code>	целый 32-битовый со знаком	<code>int32(X)</code>
<code>uint32</code>	целый 32-битовый без знака	<code>uint32(X)</code>
<code>int64</code>	целый 64-битовый со знаком	<code>int64(X)</code>
<code>uint64</code>	целый 64-битовый без знака	<code>uint64(X)</code>

Таблица 3.1 (окончание)

Символическое обозначение типа массива	Тип массива	Функция преобразования типа
logical	Логический	logical(X)
character (char)	Символьный	num2str(X)
structure (struct)	Структура (массив записей)	—
cell	Массив ячеек	—

3.1.1. Матрицы числового и логического типов

Преобразование матриц числового типа double в матрицы других числовых типов выполняется с помощью специальных встроенных функций (см. табл. 3.1).

При обработке матриц числового *целого* типа необходимо иметь в виду, что с ними запрещено выполнение некоторых арифметических операций и вычисление большинства встроенных функций, но разрешено выполнение операций отношения и логических операций:

```
>> A = [int8(159.7) int8(125.7) int8(-125.7)]
A =
    127    126   -126
>> B = [uint8(159.7) uint8(125.7) uint8(-125.7)]
B =
    160    126     0
>> C = [A<B; A==B; and(A,B)]
C =
     1     0     1
     0     1     0
     1     1     0
```

Элементами матрицы *логического* типа (logical array) являются логические константы, принимающие значения 1 (true — истина) или 0 (false — ложь), например, как в матрице C, или логические выражения (см. разд. 1.1.2):

```
>> x = [sin(3)<0.5  1;0 (sin(3)<0.1)&(cos(3)<0.2)]
x =
     1     1
     0     0
```

3.1.2. Матрицы символьного типа

Матрица символьного типа (char array) — это разновидность нечисловых матриц, элементами которой являются символьные константы (см. разд. 1.1.2).

Строки и столбцы матрицы символьного типа формируются по-разному, а именно:

- элементы *строки* представляют собой *слитную* запись, поэтому необходимые пробелы должен предусмотреть пользователь:

```
>> a = ['Alla ' 'Woman ' 'Russian']
a =
Alla Woman Russian
```

- элементы *столбца* должны содержать *одинаковое* число символов, при этом пробел считается символом:

```
>> a = ['Alla    ','Woman    ','Russian']
a =
Alla
Woman
Russian
```

Автоматическое добавление пробелов в элементах столбца с выравниванием по *левому* краю выполняется с помощью функции:

```
char('<char1>', '<char2>'...)
```

где *<char1>*, *<char2>*... — элементы столбца с произвольным количеством символов.

Одна функция `char` описывает один столбец матрицы с символьными константами *'<char1>'*, *'<char2>'*...:

```
>> a = char('Alla','Woman','Russian')
a =
Alla
Woman
Russian
```

Матрицу символьного типа удобно формировать по столбцам, используя для каждого из них свою функцию `char` и предусматривая необходимое количество пробелов для разделения столбцов:

```
>> x = [char('a','aa','aaa') char(' bb',' bbb',' b') ...
char(' cc',' ccc',' c')]
x =
a   bb  cc
aa  bbb ccc
aaa b   c
```

Преобразование матрицы численного или логического типа в матрицу символьного типа выполняется с помощью функции `num2str(x)`:

```
>> x = [5 7;-1 9]
x =
    5    7
   -1    9
```

```
>> y = num2str(x)
y =
5 7
-1 9
```

Здесь x — матрица типа `double`, а y — матрица символьного типа:

```
>> [size(x); size(y)]
ans =
2 2
2 5
```

3.1.3. Структуры (массивы записей)

Структура (массив записей — `struct array`) — это разновидность нечислового массива, предназначенного для описания M объектов N параметрами.

Для описания структуры потребуется ввести ряд новых терминов:

- ☐ *поле* (field) — имя параметра, описывающего объект: скаляра, вектора, матрицы¹ или нечислового массива;
- ☐ *число полей* равно количеству параметров N ;
- ☐ *значение поля* — значение параметра;
- ☐ *список полей* — список имен параметров;
- ☐ *запись* — список полей, одинаковый для всех M объектов;
- ☐ *число записей* равно количеству объектов M ;
- ☐ *значение записи* — список полей с их значениями для одного объекта.

Структура (массив записей) — это упорядоченная совокупность значений записей, объединенная одним именем.

Имя массива записей выбирается так же, как обычно для переменной (см. разд. 1.1.2), а *размер* равен числу записей M .

Значение каждой i -й записи формируется отдельно по каждому n -му полю:

<имя массива>(< i >) .<имя n -го поля> = <значение n -го поля>

Таким образом, для формирования M значений записей со списком из N полей требуется $M \times N$ операторов присваивания.

Сформируем массив записей (структуру) с именем `personal` для описания трех объектов ($M = 3$) — трех членов кафедры — четырьмя параметрами ($N = 4$).

Запись включает в себя следующий список полей:

- ☐ `surname` — фамилия — скаляр символьного типа;
- ☐ `data` — дата рождения (число, месяц, год) — вектор численного типа;

¹ Напомним (см. разд. 1.1.2), что по умолчанию в MATLAB любая переменная — матрица, а скаляр и вектор — ее частные случаи.

- `position` — должность — скаляр символьного типа;
- `phd` — наличие ученой степени — скаляр логического типа.

Сформируем значения массива записей `personal` по каждому полю:

```
>> personal(1).surname = 'Иванов';
>> personal(2).surname = 'Петров';
>> personal(3).surname = 'Сидоров';
>> personal(1).data = [1 2 1949];
>> personal(2).data = [5 7 1975];
>> personal(3).data = [5 8 1956];
>> personal(1).position = 'профессор';
>> personal(2).position = 'доцент';
>> personal(3).position = 'зав. лаб.';
>> personal(1).phd = true;
>> personal(2).phd = true;
>> personal(3).phd = false;
```

Список полей выводится по имени массива записей:

```
>> personal
personal =
1x3 struct array with fields:
    surname
    data
    position
    phd
```

Значение i -й записи выводится по имени массива записей с указанием индекса в круглых скобках. Например, значение 2-й записи:

```
>> personal(2)
ans =
    surname: 'Петров'
    data: [5 7 1975]
    position: 'доцент'
    phd: 1
```

Значение поля в i -й записи выводится по имени массива записей с указанием индекса в круглых скобках и имени поля. Например, значение поля `surname` первой записи:

```
>> personal(1).surname
ans =
Иванов
```

Значения поля во всех записях выводятся по имени массива с указанием имени поля. Например, поля `surname`:

```
>> personal.surname
ans =
Иванов
```

```
ans =
Петров
ans =
Сидоров
```

Удаление поля выполняется с помощью функции:

```
<имя массива> = rmfield(<имя массива>,'<имя поля>')
```

Например, удалим поле `data`:

```
>> personal = rmfield(personal,'data')
personal =
1x3 struct array with fields:
    surname
    position
    phd
```

3.1.4. Массивы ячеек

Массив ячеек (cell array) — это наиболее сложный тип массива, элементами которого являются ячейки, представляющие собой массивы любой размерности, любого размера и типа.

Элементы массива ячеек указываются в *фигурных* скобках.

Сформируем квадратную матрицу ячеек 3×3 :

```
>> A{1,1} = pi;
>> A{1,2} = [1 2 3;4 5 6];
>> A{1,3} = char('abs','angle');
>> A{2,1} = [ones(5,1)]';
>> A{2,2} = zeros(3,1);
>> A{2,3} = 'Alla';
>> A{3,1} = 7 ;
>> A{3,2} = rand(5,1);
>> A{3,3} = personal;
```

где `personal` — имя структуры, сформированной в разд. 3.1.3.

Вывод элементов массива ячеек выполняется по его имени с указанием индексов:

```
>> A{1,2}
ans =
     1     2     3
     4     5     6

>> A{3,3}
ans =
1x3 struct array with fields:
    surname
    age
    position
    phd
```

С элементами массива ячеек можно выполнять операции, разрешенные для данного типа массива с учетом согласования их размерностей и размеров, например:

```
>> B = sum(A{1,2})+A{1,1}
B =
    8.1416    10.1416    12.1416
```

Графическое представление матрицы ячеек создается с помощью функции:

```
cellplot(A, 'legend')
```

3.1.5. Определение типа массива

Для определения типа массива служит функция:

```
class(<имя массива>)
```

Например, для массива *A*, сформированного в разд. 3.1.4:

```
>> class(A)
ans =
cell
```

3.2. Содержание лабораторной работы

Содержание работы связано с изучением типов массивов в MATLAB в режиме прямых вычислений.

3.3. Задание на лабораторную работу

Задание на лабораторную работу включает в себя следующие пункты:

1. Знакомство с матрицами числового и логического типов.

Ввести матрицу *A* с элементами:

$$\mathbf{A} = \begin{bmatrix} 127,1 & -127,1 & 127,7 \\ -127,7 & 0 & 128,4 \\ -128,4 & 255,7 & 255,1 \end{bmatrix}$$

и выполнить с ней следующие действия:

- преобразовать в матрицу *B* в целых 8-битовых чисел со знаком;
- преобразовать в матрицу *C* в целых 8-битовых чисел без знака;
- преобразовать в матрицу *D* логического типа;
- определить тип матриц *A*, *B*, *C*, *D*.

Пояснить:

- как преобразовать матрицу *A* в матрицы *B*, *C* и *D*;
- по какому правилу формируются элементы матрицы *B*;

- по какому правилу формируются элементы матрицы C;
- по какому правилу формируются элементы матрицы D;
- как определяется тип матрицы.

2. Операции с матрицами числового типа.

Выполнить следующие операции:

- вычислить значения синуса всех элементов матриц A, B и C;
- вычислить суммы матриц A и B, B и C;
- проверить, являются ли элементы матрицы A числами, большими единицы, и определить вид и тип результата;
- выполнить логическую операцию "И" с матрицами B, C и определить вид и тип результата.

Сделать выводы по результатам выполнения операций.

3. Знакомство с матрицами символьного типа.

Выполнить следующие действия:

- сформировать трехэлементный вектор-строку X символьного типа с ФИО студента;
- сформировать трехэлементный вектор-столбец Y символьного типа с ФИО студента;
- сформировать матрицу F символьного типа 2×2 с элементами:

$$F = \begin{bmatrix} \text{КИХ} & R = 15 \\ \text{FIR} & \text{Order} = 15 \end{bmatrix};$$

- преобразовать матрицу A (см. п. 1) в матрицу G символьного типа;
- определить типы матриц A и G.

Пояснить:

- как обеспечить не слитный вывод элементов вектора-строки;
- как обеспечить автоматическое добавление пробелов при выводе элементов столбца;
- как преобразовать матрицу числового типа в матрицу символьного типа.

4. Знакомство с массивами записей (структурами).

Сформировать массив записей (структуру) с именем `Filter` для описания четырех фильтров.

Каждая запись должна содержать три поля со следующими именами и их значениями:

- `Type` (тип избирательности) — `lowpass`, `highpass`, `bandpass`, `stopband`;
- `Order` (порядок фильтра) — 10, 20, 30, 40;
- `Poles` (наличие полюсов) — `true`, `false`, `false`, `true`.

Выполнить следующие действия с массивом `Filter`:

- вывести список полей;
- вывести значение 1-й записи;
- вывести значения поля `Type` во всех записях;
- вывести значение поля `Type` в 3-й записи;
- удалить поле `Poles`.

Пояснить:

- с какой целью создается массив записей;
- что собой представляет запись и значение записи;
- каков размер массива записей `Filter`.

5. Знакомство с матрицами ячеек.

Создать матрицу ячеек s 3×3 , элементами которой являются сформированные ранее массивы:

$$S = \begin{bmatrix} A & B & C \\ D & F & G \\ X & Y & \text{Filter} \end{bmatrix}.$$

Выполнить следующие действия:

- последовательно вывести элементы матрицы `s` и определить их тип;
- вывести графическое представление матрицы `s`.

Пояснить:

- из каких элементов создается матрица ячеек;
- как эти элементы вводятся;
- как выводится графическое представление матрицы ячеек;
- что оно собой представляет.

3.4. Задание на самостоятельную работу

Самостоятельное задание рекомендуется для закрепления полученных знаний и включает в себя следующие пункты:

1С. Операции с элементами массива ячеек.

Сформировать массив ячеек `A`, рассмотренный в *разд. 3.1.4*, и привести пример арифметического выражения с элементами данного массива типа `double`.

2С. Операции с матрицами целого типа.

Привести пример выражения с матрицами целого типа.

3С. Операции с матрицами логического типа.

Привести пример логического выражения, в котором все переменные — матрицы.

4С. Операции с матрицами символьного типа.

Сформировать матрицу символьного типа размера 3×3 посредством преобразования числовой матрицы типа `double`.

5С. Операции с массивом записей.

Привести пример массива записей для описания трех объектов двумя параметрами, один из которых представлен вектором, а другой — матрицей.

3.5. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит результаты выполнения каждого пункта задания, копируемые из окна **Command Window** (шрифт Courier New), и ответы на поставленные вопросы (шрифт Times New Roman).

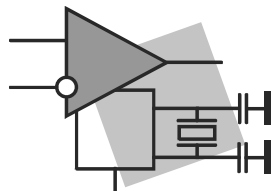
Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. К какому типу относятся числовые массивы по умолчанию?
2. Как определить тип массива?
3. Какие типы числовых массивов используются в MATLAB?
4. Как преобразовать матрицу типа `double` в матрицы целых чисел разрядности 8, 16, 32 и 64 со знаком и без знака?
5. Какие операции возможны с матрицами числового целого типа?
6. Как преобразовать числовую матрицу в матрицу логического типа?
7. Что собой представляет матрица логического типа?
8. Какие типы нечисловых массивов предусмотрены в MATLAB?
9. Что собой представляет матрица символьного типа?
10. Что собой представляет массив записей?
11. В каких случаях целесообразно создавать массив записей (структуру)?
12. Что собой представляет матрица ячеек?
13. В каких случаях целесообразно создавать матрицу ячеек?
14. С какой целью выводится графическое представление матрицы ячеек?

3.6. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 4.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Приложения 1—2.

ГЛАВА 4



Средства графики

Цель работы: изучить графический инструментарий MATLAB и овладеть навыками построения двумерных и трехмерных графиков.

4.1. Краткая теоретическая справка

Графический инструментарий MATLAB для построения и оформления *двумерных* и *трехмерных* графиков имеет свою специфику. Однако ряд следующих типовых положений является для них общим.

❑ Текущий график выводится в текущее графическое окно **Figure**, первый — в окно **Figure 1**. По умолчанию новый график выводится в *то же окно*, при этом предыдущий график автоматически удаляется.

❑ Вывод графиков *в отдельных* графических окнах с автоматически присваиваемыми номерами **Figure 1**, **Figure 2**, ... выполняется с помощью функции (без аргумента):

`figure`

которая ставится *перед* новой функцией построения графика.

❑ Вывод графика *в отдельном* графическом окне, имя которого присваивается пользователем, выполняется с помощью функции:

`figure('Name', '<Имя графика>', 'NumberTitle', 'off')`

❑ Вывод в текущее графическое окно **Figure** *нескольких графиков на одних координатных осях* выполняется по команде:

`hold on`

которая ставится *перед* новой функцией построения графика.

❑ Удаление из текущего графического окна **Figure** всех предыдущих графиков *перед выводом нового графика* выполняется по команде:

`hold off`

❑ Разбиение текущего графического окна **Figure** на отдельные поля для вывода *независимых графиков* выполняется с помощью функции:

`subplot(m,n,p)`

где $m \times n$ — размер *матрицы графического окна*: m строк и n столбцов; p — порядковый номер поля выводимого графика, считая по строкам слева направо.

Функция построения графика ставится после функции `subplot`.

- ☐ Средства оформления графиков, представленные в виде функций MATLAB (табл. 4.1), ставятся *после функции построения графика*. Во избежание ошибок `<текст>` рекомендуется вводить латинскими буквами.

Таблица 4.1. Функции оформления графиков

Функция	Назначение
<code>grid</code>	Нанесение координатной сетки с автоматическим выбором шага
<code>title('<текст>')</code>	Заголовок графика
<code>xlabel('<текст>')</code> <code>ylabel('<текст>')</code> <code>zlabel('<текст>')</code>	Обозначение осей графика x, y, z
<code>xlim([xmin xmax])</code> <code>ylim([ymin ymax])</code> <code>zlim([zmin zmax])</code>	Установка границ (двухэлементным вектором) по осям x, y, z при выводе графика
<code>legend('legend1', 'legend2', ...)</code>	Размещение легенды на автоматически выбранном месте. При выводе нескольких графиков на одних осях их легенда отображается в порядке вывода графиков

4.1.1. Двумерные графики

Система MATLAB предлагает большое разнообразие стандартных функций для построения двумерных графиков.

Полный список функций, используемых в двумерной графике, выводится по команде:

```
help graph2d
```

Основные из них с наиболее распространенными форматами приведены в табл. 4.2.

4.1.2. Управление свойствами двумерных графиков

Свойствами графика можно управлять с помощью `<параметров управления>` (см. табл. 4.2), которые условно можно разделить на две группы:

- ☐ `LineStyle` — свойства без стандартных имен;
- ☐ `PropertyName` — свойства со стандартными именами.

Таблица 4.2. Функции построения двумерных графиков

Имя функции	Назначение и формат
plot	Графики в линейном масштабе с линейной интерполяцией между соседними значениями: plot(x,y[,<параметры управления>]) где x, y — аргумент и функция (векторы или матрицы), согласованные по длине; <параметры управления> — необязательные параметры, управляющие свойствами графика (см. разд. 4.2.2). Квадратные скобки используются для условного обозначения необязательных параметров
semilogx	Графики в логарифмическом масштабе по оси абсцисс и линейном — по оси ординат. Формат подобен функции plot. Диапазон по оси абсцисс в логарифмическом масштабе можно задавать с помощью функции: logspace(d1,d2[,n]) где d1, d2 — начальное 10^{d1} и конечное 10^{d2} значения диапазона; n — количество точек в логарифмическом масштабе, по умолчанию равно 50
semilogy	Графики в линейном масштабе по оси абсцисс и логарифмическом — по оси ординат. Формат подобен функции plot
loglog	Графики в логарифмическом масштабе по осям абсцисс и ординат. Формат подобен функции plot
stem	Графики последовательностей чисел: stem(x,y,'fill'[,<параметры управления>]) где 'fill' — необязательный параметр, указывающий на закрашивание маркеров. Остальные входные параметры определяются как для функции plot
hist	Гистограммы: hist(y,x) где y, x — векторы одинаковой длины; гистограмма отображает число попаданий значений элементов вектора y в интервалы, центры которых заданы элементами вектора x. В отсутствии вектора x для значений элементов вектора y по умолчанию выбирается 10 интервалов, и гистограмма отображает число попаданий значений элементов вектора y в центры данных интервалов. Цвет столбцов выбирается с помощью функции colormap (см. разд. 4.1.4)

Параметры группы `LineStyle` определяют тип и цвет линии графика, а также вид маркеров. Значения параметров данной группы представлены в табл. 4.3. В функциях построения графиков значения параметров указываются в *апострофах* без разделяющих символов *в произвольном порядке*. Например:

```
stem(x,y,'-ms')
```

Если параметры не указаны, то они выбираются автоматически.

Таблица 4.3. Параметры группы `LineStyle`

Тип линии		Цвет линии или маркера		Вид маркера	
символ	линия	символ	цвет	символ	маркер
-	solid (непрерывная)	y	yellow (желтый)	.	point (точка)
:	dotted (пунктир, короткий штрих)	m	magenta (фиолетовый)	o	circle (кружок)
-.	dashdot (штрих-пунктир)	c	cyan (голубой)	x	x-mark (крестик)
--	dashed (пунктир, длинный штрих)	r	red (красный)	+	plus (плюс)
		g	green (зеленый)	*	star (звездочка)
		b	blue (синий)	s	square (квадрат)
		w	white (белый)		
		k	black (черный)		

Параметры группы `PropertyName` представлены четырьмя разновидностями со следующими стандартными именами, задаваемыми в апострофах:

- ❑ `LineWidth` — толщина линии в пунктах (1 пункт = 1/75 дюйма ≈ 0,34 мм), задаваемая цифрой без апострофов, по умолчанию равна 0,5;
- ❑ `MarkerEdgeColor` — цвет маркера, задаваемый значением соответствующего параметра из табл. 4.3 в апострофах;
- ❑ `MarkerFaceColor` — цвет закрашивания маркера (для замкнутых маркеров типа кружок, квадрат и т. п.), задаваемый значением соответствующего параметра из табл. 4.3 в апострофах;
- ❑ `MarkerSize` — размер маркера в пунктах, задаваемый цифрой без апострофов, по умолчанию равен 7.

Например:

```
stem(x,y,'MarkerSize',5,'MarkerEdgeColor','g','MarkerFaceColor',...  
'r','LineWidth',1)
```

4.1.3. Трехмерные графики

Трехмерная графика предназначена для построения в трехмерном пространстве графиков функций двух переменных (двух аргументов) $z(x, y)$.

Построение трехмерных графиков начинается с *формирования сетки* на плоскости xOy с помощью вспомогательных матриц **X** и **Y** по известным векторам **x** и **y** соответственно, где **X** — матрица, строки которой — копии вектора **x**, а **Y** — матрица, столбцы которой — копии вектора **y**. Матрицы **X** и **Y** должны иметь *одинаковые размеры*: количество строк каждой из них равно длине вектора **y**, а столбцов — длине вектора **x**.

Матрицы **X** и **Y** формируются с помощью функции:

```
[X,Y] = meshgrid(x,y)
```

Если векторы **x** и **y** одинаковы, то допускается короткий формат:

```
[X,Y] = meshgrid(x)
```

Полный список функций, используемых в трехмерной графике, выводится по команде:

```
help graph3d
```

Основные из них с наиболее распространенными форматами приведены в табл. 4.4.

Таблица 4.4. Функции построения трехмерных графиков

Имя функции	Назначение и формат
plot3	Трехмерные графики в виде двумерных линий: plot3(X,Y,Z[, <параметры управления>]) где X , Y — матрицы, формирующие сетку на плоскости xOy с помощью функции meshgrid ; Z — функция (вектор или матрица); <i><параметры управления></i> — необязательные параметры, управляющие свойствами графика (см. разд. 4.1.4). Квадратные скобки используются для условного обозначения необязательных параметров
mesh	Трехмерные сетчатые графики (с автоматическим нанесением координатных сеток). Формат подобен функции plot3
surf	Трехмерные сетчатые графики с окрашиванием поверхности (с автоматическим нанесением координатных сеток). Формат подобен функции plot3

Пример формирования матриц **X** и **Y** для сетки на плоскости xOy и построения трехмерных графиков с помощью функций **mesh** и **plot3**:

```
>> [X,Y] = meshgrid(-5:0.25:5);  
>> Z = X.^2+Y.^2;
```

```
>> mesh(X,Y,Z)
>> figure
>> plot3(X,Y,Z),grid
```

4.1.4. Управление свойствами трехмерных графиков

Свойствами трехмерного графика можно управлять с помощью <параметров управления> (см. табл. 4.4), рассмотренных в разд. 4.1.2. Для управления свойствами трехмерных графиков предусмотрен ряд дополнительных средств, из которых выделим следующие два:

- ❑ выбор палитры цветов;

Палитра цветов задается с помощью функции:

```
colormap('<символическое имя палитры>')
```

Символические имена основных палитр представлены в табл. 4.5; по умолчанию установлена палитра `hsv`.

Функция `colormap` может стоять до или после функции построения графика.

Восстановление палитры `hsv` выполняется с помощью функции:

```
colormap('default')
```

- ❑ вывод на поле графика шкалы цветов, устанавливающей соответствие со значениями функции, выполняется по команде:

```
colorbar
```

которая обязательно ставится последней.

Таблица 4.5. Стандартные палитры

Символическое имя	Палитра
bone	Серо-синяя
cool	Фиолетово-голубая
copper	Оттенки меди
flag	Чередование: красный, белый, синий, черный
gray	Оттенки серого
hot	Чередование: черный, красный, желтый, белый
hsv	Радуга
jet	Разновидность hsv
pink	Розовая
colorcube	Расширенная палитра hsv
autumn	Красно-желтая
spring	Желто-фиолетовая

Таблица 4.5 (окончание)

Символическое имя	Палитра
winter	Сине-зеленая
summer	Желто-зеленая
white	Белая (бесцветная)

4.2. Содержание лабораторной работы

Содержание работы связано с изучением инструментария MATLAB для построения, оформления и управления свойствами двумерных и трехмерных графиков.

4.3. Задание на лабораторную работу

Задание на лабораторную работу включает в себя следующие пункты:

1. Построение двумерного графика.

Для аргумента x , заданного на интервале

$$x \in [0; 8\pi] \text{ с шагом } \Delta x = \pi/8, \tag{4.1}$$

вычислить функцию

$$y_1 = \sin x \tag{4.2}$$

и вывести ее график в линейном масштабе с линейной интерполяцией между соседними значениями.

Выполнить следующие действия по оформлению графика:

- нанести координатную сетку;
- обозначить ось абсцисс.

Пояснить:

- какая функция используется для вывода графика;
- в какое окно выводится график;
- какие функции используются для нанесения координатной сетки и обозначения оси абсцисс.

2. Построение нескольких двумерных графиков на одних координатных осях.

В том же окне вывести графики функций:

$$y_2 = \frac{\sin x}{x}; \tag{4.3}$$

$$y_3 = 0,5 \cos x, \tag{4.4}$$

для которых аргумент x задан на интервале (4.1).

При выводе графиков выбрать различный цвет линий для функций y_1 , y_2 и y_3 .
Выполнить следующие действия по оформлению графика:

- обозначить ось ординат как `axis y`;
- ввести заголовок графика в виде `Functions y1 y2 y3`;
- разместить легенду для графиков функций: $y_1 — \sin(x)$; $y_2 — \sin(x)/x$; $y_3 — 0.5\cos(x)$.

Пояснить:

- какая команда обеспечивает вывод нескольких графиков на одних координатных осях;
- какая функция используется для вывода графиков;
- сохраняется ли координатная сетка и обозначение оси абсцисс при выводе следующих графиков в то же окно;
- какие функции используются для обозначения оси ординат, вывода заголовка и размещения легенды.

3. Построение независимых графиков в одном окне с его разбиением на отдельные поля.

В графическом окне с именем **Graph2D** вывести друг под другом графики функций y_1 (4.2), y_2 (4.3) и y_3 (4.4).

Выполнить следующие действия по оформлению графиков:

- нанести координатную сетку;
- обозначить оси абсцисс и ординат;
- ввести заголовки графиков.

Пояснить:

- как создается окно с заданным именем;
- какая функция позволяет строить несколько независимых графиков в одном графическом окне.

4. Построение графика последовательности чисел.

В окне **Sequence1** вывести график значений функции y_2 (4.3) с нанесением координатной сетки и без закрашивания маркеров.

В новом окне **Sequence2** вывести тот же график с нанесением координатной сетки, закрашиванием маркеров и следующей установкой параметров управления:

- толщина линий равна 2;
- размер маркеров равен 6;
- цвет маркеров отличается от цвета линий;
- цвет закрашивания маркеров отличается от цветов линий и маркеров.

Пояснить:

- какая функция используется для вывода графика последовательности чисел;
- какой параметр этой функции отвечает за закрашивание маркеров;
- как устанавливаются параметры управления.

5. Построение графиков в полулогарифмическом и логарифмическом масштабах.

По оси абсцисс x_1 задать диапазон значений $[1; 10^4]$ с помощью функции `logspace`.

Вычислить функцию

$$y_4 = \sqrt{x}. \quad (4.5)$$

В окне **Logarithms axes** вывести друг под другом графики функции y_4 (4.5) с нанесением координатной сетки и следующих масштабах по осям:

- логарифмическом — по оси абсцисс; линейном — по оси ординат;
- логарифмическом — по осям абсцисс и ординат.

Пояснить:

- как диапазон значений задается с помощью функции `logspace`;
- какая функция используется для вывода графика в логарифмическом масштабе по оси абсцисс;
- какая функция используется для вывода графика в логарифмическом масштабе по осям абсцисс и ординат.

6. Построение гистограмм.

В окне **Histogram** вывести гистограмму нормального белого шума (см. табл. 2.1) — вектора y_5 длиной 1000. Количество интервалов выбрать по умолчанию.

Пояснить:

- какая функция используется для построения гистограммы;
- что отображает гистограмма;
- как гистограмма связана с плотностью вероятности нормального белого шума.

7. Построение трехмерного графика.

Для аргументов x и y , заданных на одинаковых интервалах:

$$x \in [-\pi; \pi] \text{ с шагом } \Delta x = \pi/32,$$

$$y \in [-\pi; \pi] \text{ с шагом } \Delta y = \pi/32,$$

вычислить функцию

$$z = \sin x + \cos y,$$

и в окне **Graph3D** вывести ее сетчатый график с автоматическим нанесением координатных сеток.

Выполнить следующие действия по оформлению графика:

- выбрать фиолетово-голубую палитру;
- обозначить оси x , y , z ;
- вывести на поле графика шкалу цветов.

Пояснить:

- с чего начинается построение трехмерного графика; какая функция для этого используется;
- какая функция используется для вывода графика;
- какая функция используется для выбора палитры;
- какая команда используется для вывода шкалы цветов.

4.4. Задание на самостоятельную работу

Самостоятельное задание рекомендуется для закрепления полученных знаний и включает в себя следующие пункты:

1С. Построение двумерного графика.

Для аргумента x , заданного на интервале $x \in [-2\pi; 2\pi]$, вывести график функции

$$y = x + \sin x$$

с помощью функции `plot`, вывести заголовок и легенду, обозначить оси абсцисс и ординат.

2С. Построение двумерных графиков на одних координатных осях.

Для аргумента x , заданного на интервале $x \in [-2\pi; 2\pi]$, с помощью функции `plot` вывести графики функций, образующих систему уравнений:

$$\begin{cases} y = 5 \sin x; \\ y = 5x + 2. \end{cases}$$

Найти решение системы (приблизительное) по графику и проверить его методом подстановки в окне **Command Window**.

Для оформления графиков, включая вывод легенды, использовать программные средства.

3С. Построение двумерных графиков в одном графическом окне на отдельных полях.

Для аргумента x , заданного на интервале $x \in [-2\pi; 2\pi]$, с помощью функции `plot` на отдельных полях вывести графики функций:

$$y_1 = \sin x ;$$

$$y_2 = \sin |x| ;$$

$$y_3 = |\sin |x|| .$$

4С. Построение трехмерных графиков.

Привести примеры построения трехмерных графиков с помощью функций `plot3`, `mesh` и `surf` в отдельных графических окнах с оформлением, включая вывод шкалы цветов.

4.5. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит результаты выполнения каждого пункта задания, включая операции для вычисления функций и построения графиков, копируемые из окна **Command Window** (шрифт Courier New), созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**) и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

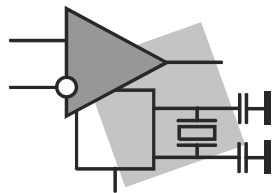
1. В какое графическое окно выводится график по умолчанию?
2. Как вывести график в новое графическое окно?
3. Как вывести несколько графиков на одних координатных осях?
4. Как удалить графики перед выводом нового графика в то же графическое окно?
5. Как вывести несколько независимых графиков в одном графическом окне с его разбиением на отдельные поля?
6. Какие средства оформления графиков используются в MATLAB?
7. Какие средства предусмотрены для установки типа, цвета и толщины линий?
8. Какие средства предусмотрены для установки вида, размера и цвета маркеров?
9. Какая функция используется для построения двумерных графиков в линейном масштабе с линейной интерполяцией между соседними значениями?
10. Какая функция используется для построения графика последовательности чисел?
11. Какие функции используются для построения графиков в полулогарифмическом и логарифмическом масштабах?
12. Какая функция используется для построения гистограмм?
13. В чем заключается подготовка перед построением трехмерного графика?
14. Какие функции используются для построения трехмерных графиков?

15. Как выбрать палитру цветов при построении трехмерного графика?
16. Как вывести шкалу цветов на поле трехмерного графика?

4.6. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 5.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Приложения 1—2.

ГЛАВА 5



Режим программирования: script-файлы и function-файлы

Цель работы: изучить программные средства MATLAB и овладеть навыками создания файлов-сценариев (script-файлов) и внешних функций (function-файлов).

5.1. Краткая теоретическая справка

Режим программирования предназначен для создания программ пользователя в среде MATLAB.

Все программы пользователя, создаваемые в MATLAB, сохраняются на диске и имеют расширение *m*, поэтому их называют *М-файлами*.

Различают *две* разновидности М-файлов:

- ☐ script-файл (файл-сценарий);
- ☐ function-файл (файл-функция).

В М-файлах, независимо от их вида, должны соблюдаться следующие правила языка MATLAB:

- ☐ переменные не объявляются и не описываются;
- ☐ не используются метки;
- ☐ отсутствует оператор безусловного перехода типа "go to" (т. к. нет меток);
- ☐ не фиксируется (оператором или служебным словом) конец программы.

5.1.1. Script-файлы

Script-файлом называют создаваемый пользователем М-файл, представляющий собой основную (управляющую) программу.

Термины "script-файл" и "программа" употребляют в тождественном смысле.

Программа состоит из операторов, записываемых построчно. По правилам хорошего стиля программирования рекомендуется:

- ☐ в начале программы ставить оператор-заголовок:

```
script
```

□ во избежание конфликта переменных в Workspace и для очистки экрана, после заголовка разместить команды `clc` и `clear`.

Имя script-файла выбирается по тем же правилам, что и имя переменной (см. разд. 1.1.2).

Пример простейшего script-файла `s1`:

```
script
clc
% Диапазон значений аргумента
x = 0:0.1:7;
% Вычисление значений синусоиды y
y = sin(x);
% Вычисление значений косинусоиды z
z = cos(x);
% Графики синусоиды y и косинусоиды z
subplot(2,1,1), plot(x,y,'--r'), grid
subplot(2,1,2), plot(x,z), grid
```

Обращение к script-файлу в режиме прямых вычислений осуществляется по его имени:

```
>> s1
```

После этого выполняются действия согласно программе с выводом результатов в окне **Command Window**.

Все переменные script-файла являются *глобальными*, т.е. они сохраняются в Workspace и доступны для использования в любых приложениях.

5.1.2. Function-файлы

Function-файлом называют создаваемый пользователем М-файл, представляющий собой внешнюю функцию (в отличие от встроенных функций MATLAB).

Термины "function-файл" и "внешняя функция" употребляют в тождественном смысле.

Описание function-файла начинается с оператора-заголовка `function`. Формат описания при нескольких выходных параметрах имеет вид:

```
function [Y1,Y2,...] = <имя функции>(X1,X2,...)
```

где *<имя функции>* — имя function-файла, выбираемое подобно имени переменной; *X1, X2, ...* — список *формальных* входных параметров; *Y1, Y2, ...* — список *формальных* выходных параметров.

При *одном* выходном параметре имеем короткий формат описания:

```
function Y = <имя функции>(X1,X2,...)
```

После заголовка следует *тело функции* — записанная построчно на языке MATLAB программа определения выходных параметров *Y1, Y2, ...* по входным — *X1, X2, ...*

Пример function-файла F1:

```
function [z,p] = F1(x,y)
% Вычисление суммы кубов z
z = x.^3+y.^3;
% Вычисление квадратного корня p
p = sqrt(abs(z));
```

Пример function-файла F2 с одним выходным параметром:

```
function z = F2(x,y)
% Вычисление суммы кубов z
z = x.^3+y.^3;
```

Обращение к внешней функции подобно обращению к встроенной функции MATLAB и при нескольких выходных параметрах имеет вид:

```
[Y1факт,Y2факт,...] = <имя функции>(X1факт,X2факт,...)
```

где X1факт, X2факт, ... — список *фактических* входных параметров; Y1факт, Y2факт, ... — список *фактических* выходных параметров.

Фактические значения входных параметров X1факт, X2факт, ... должны быть определены *перед* обращением к внешней функции.

Примеры обращения к function-файлу F1 с несколькими выходными параметрами:

```
>> [d,c] = F1(2,3);
>> a = 2; b = 3;
>> [d,c] = F1(a,b);
```

При одном выходном параметре допускается короткий формат обращения к внешней функции:

```
<имя функции>(X1факт,X2факт,...)
```

Примеры обращения к function-файлу F2 с одним выходным параметром:

```
>> a = 2; b = 3;
>> d = F2(a,b)+sin(7+F2(5,7));
```

Разделение параметров function-файлов на *формальные* и *фактические* обусловлено тем, что *формальные* параметры являются *локальными*, т. е. они (вместе с внутренними переменными function-файла) загружаются в Workspace на время вычисления внешней функции и удаляются из Workspace по завершении вычислений. *Фактические* же параметры сохраняются в Workspace.

5.1.3. Оформление и вывод листинга М-файлов

При оформлении М-файлов рекомендуется соблюдать следующие правила:

- ☐ включать комментарии, поясняющие назначение переменных, выполняемые действия и т. п.;
- ☐ во избежание выводов нежелательных промежуточных результатов ставить точку с запятой.

Вывод листинга М-файла в окне **Command Window** выполняется по команде:

```
type <имя М-файла>
```

5.1.4. Ввод/вывод данных

Ввод данных с клавиатуры организуется с помощью функции:

```
<имя переменной> = input('<текст>');
```

приостанавливающей выполнение программы для ввода данных с клавиатуры; точка с запятой в конце функции `input` блокирует автоматический вывод вводимых данных. После ввода и нажатия клавиши <Enter> автоматически продолжается выполнение программы:

```
>> w0 = input('w0 = ');
w0 =
```

С клавиатуры следует ввести значение `w0` и нажать клавишу <Enter>:

```
w0 = pi/16
```

Вывод данных в окно **Command Window** организуется следующим образом.

- Вывод значений переменной или текста выполняется с помощью соответствующей функции:

```
disp(<имя переменной>)
```

или

```
disp('<текст>')
```

Для вывода значений нескольких переменных или текстов на одной строке их следует представить в виде вектора:

```
>> x = 5; a = 3; b = 10;
>> disp([x a b])
      5      3     10
>> disp(['x      ' 'a      ' 'b'])
x      a      b
```

- Вывод символьных переменных в виде слитного текста с игнорированием в них пробелов справа или с их учетом выполняется с помощью соответствующей функции:

```
strcat('<текст1>', '<текст2>', ...)
```

или

```
strcat(['<текст1>' '<текст2>', ...])
```

Например:

```
>> strcat('hello      ', 'goodbye')
ans =
hellogoodbye
```



```
>> strcat(['hello      ' 'goodbye'])
ans =
hello      goodbye
```

Для вывода значения численной переменной одновременно с текстом удобно воспользоваться функцией `num2str` (см. разд. 3.1.2):

```
>> i = 5;
>> strcat([' Коэффициент ', num2str(i), '-го ВАРИАНТА'])
ans =
Коэффициент 5-го ВАРИАНТА
```

Вывод переменной `ans` можно блокировать с помощью функции `disp`:

```
>> i = 5;
>> disp(strcat([' Коэффициент ', num2str(i), '-го ВАРИАНТА']))
Коэффициент 5-го ВАРИАНТА
```

Подобный вывод удобно организовать в теле цикла с изменяющейся переменной цикла, о чем пойдет речь далее в разд. 6.1.2.

Функцию `strcat` можно использовать для вывода значения численной переменной в *заголовке графика* (см. табл. 4.1):

```
>> N = 3;
>> title(strcat(['Amplitude Spectrum N = ', num2str(N)]))
```

или в обозначении *осей графика*:

```
>> i = 7;
>> ylabel(strcat('S', num2str(i), '(f)'))
```

В М-файлах функция `disp` используется при выводе *комментариев и сообщений*:

```
>> disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
% Введите ИСХОДНЫЕ ДАННЫЕ
>> disp('% Вывод значения СКО')
% Вывод значения СКО
```

5.1.5. Пауза и досрочное прерывание программы

Приостановить процесс выполнения программы на неопределенное (до нажатия любой клавиши) время можно по команде:

```
pause
```

В режиме программирования команду `pause` необходимо ставить в тех случаях, когда в процессе выполнения программы последовательно выводятся разные графики в текущее графическое окно; в противном случае пользователю окажется доступным только один, последний, график. Команда `pause` ставится *перед* выводом следующего графика.

В том случае, если пользователь не предполагает следить за выполнением программы, и его интересует только результат, можно выводить графики в разные графические окна по команде `figure` без пауз.

Командой `pause` удобно воспользоваться в script- или function-файле перед выводом результатов, которому предшествует сообщение:

```
V = var(randn(1,1000));  
disp('% Для вывода ДИСПЕРСИИ ШУМА нажмите <ENTER>')  
pause  
disp([' V = ' num2str(V)])
```

в том числе при выводе графиков:

```
x = 0:pi/32:2*pi; y = sin(x);  
disp('% Для вывода ГРАФИКА СИГУСОИДЫ нажмите <ENTER>')  
pause  
plot(x,y), grid
```

Досрочное прерывание процесса выполнения программы в результате проверки тех или иных условий выполняется по команде:

return

Рекомендуется предусмотреть вывод сообщения о причине досрочного прерывания.

Для принудительного снятия script-файла с выполнения следует нажать комбинацию клавиш `<Ctrl>+<Break>`.

5.1.6. Создание и хранение М-файлов

Для создания М-файла и его сохранения в папке пользователя необходимо выполнить следующие действия:

1. В окне **MATLAB** выбрать в главном меню пункт **File | New** (Файл | Новый) и определить тип создаваемого М-файла.
2. В раскрывшемся окне **Editor** (Редактор) набрать текст М-файла построчно.
3. Для сохранения М-файла выбрать в главном меню команду **File | Save as** (Сохранить как).
4. В раскрывшемся окне **Save as** выбрать требуемую папку, присвоить имя новому М-файлу (без расширения) и нажать кнопку **Save** (Сохранить).

При открытом окне редактора после внесения изменений в М-файл необходимо его сохранить перед следующим запуском. Признаком несохраненного файла является символ "*" (звездочка) при его имени в окне редактора.

Создание новой папки выполняется с помощью контекстного меню в окне **Current Folder**.

Сохранение пути к требуемой папке выполняется по команде контекстного меню **Add to Path | Selected Folders** (Добавить путь | Выбранные папки). Сохранение пути к папке позволяет в текущей сессии запускать М-файл, не открывая данную папку.

При запуске М-файлов из текущей папки путь к ней можно не сохранять.

5.2. Содержание лабораторной работы

Содержание работы связано с изучением средств MATLAB для создания файлов-сценариев (script-файлов) и внешних функций (function-файлов).

5.3. Задание на лабораторную работу

Задание на лабораторную работу заключается в создании script- и function-файлов и их выполнении в режиме прямых вычислений и включает в себя следующие пункты:

1. Создание script-файла.

Создать script-файл, который начинается с оператора-заголовка, после чего выполняются следующие действия:

- очистка экрана;
- очистка Workspace;
- генерирование равномерного Y_{uniform} и нормального Y_{normal} белого шума длины N , равной 1000;
- вывод в графическом окне **White Uniform Noise** графика равномерного белого шума Y_{uniform} и гистограммы (друг под другом).

График шума вывести с помощью функции `plot` с нанесением координатной сетки и заголовком.

Гистограмму шума вывести с заголовком; количество интервалов выбрать по умолчанию;

- вывод в графическом окне **White Normal Noise** аналогичных графиков для нормального белого шума Y_{normal} .

Сохранить script-файл с именем `Noise_1`.

Запустить script-файл на выполнение.

Проверить содержимое Workspace после выполнения script-файла.

Пояснить:

- что такое script-файл;
- в каком окне создается script-файл;
- какие команды используются для очистки экрана и Workspace;
- как выбирается имя script-файла;
- какое расширение имеют script-файлы;
- как сохранить script-файл;
- как обратиться к script-файлу в режиме прямых вычислений;
- где хранятся переменные script-файла в процессе и по завершении его выполнения.

2. Добавление паузы и сообщения о выводе результатов.

В созданный script-файл `Noise_1` (см. п. 1) добавить:

- строки с сообщением о выводе графиков с текстом:

Для вывода графика и гистограммы РАВНОМЕРНОГО БЕЛОГО ШУМА нажмите <ENTER>

Для вывода графика и гистограммы НОРМАЛЬНОГО БЕЛОГО ШУМА нажмите <ENTER>

- паузу перед выводом каждого из графиков.

Пояснить, какие средства MATLAB для этого используются.

Сохранить script-файл с именем `Noise_2`.

Запустить script-файл на выполнение.

3. Ввод данных с клавиатуры.

В созданном script-файле `Noise_2` (см. п. 2) организовать ввод длины шума `N` с клавиатуры с сообщением о вводе.

Сохранить script-файл с именем `Noise_3`.

Запустить script-файл на выполнение.

Пояснить, как организуется ввод данных с клавиатуры.

4. Создание function-файла.

Создать function-файл `mean_var` для вычисления среднего значения `MEAN` и дисперсии `VAR` случайной последовательности `Y`.

В function-файл `mean_var` организовать вывод:

- символьной переменной `'Mean value = '` и численного значения переменной `MEAN`;
- символьной переменной `'Variance value = '` и численного значения переменной `VAR`.

Добавить в function-файл строки комментариев.

Вычислить среднее значение и дисперсию равномерного `Y_uniform` и нормально-го `Y_normal` белого шума длины 5000 с помощью созданного function-файла.

Проверить содержимое Workspace после выполнения function-файла.

Пояснить:

- что такое function-файл;
- каков формат function-файла;
- назначение формальных и фактических параметров function-файла;
- в каком окне создается function-файл;
- как сохранить function-файл;
- какое расширение имеют function-файлы;
- как обратиться к function-файлу для его выполнения;

- где хранятся переменные function-файла в процессе и по завершении его выполнения.

5. Использование function-файла в script-файле.

На основе script-файла `Noise_3` (см. п. 3) создать новый script-файл, в котором *после* вывода графиков вычислить среднее значение и дисперсию равномерного `Y_uniform` и нормального `Y_normal` белого шума с помощью внешней функции `mean_var`.

Добавить строки с сообщением о выводе результатов с текстом:

Вывод статистических характеристик РАВНОМЕРНОГО БЕЛОГО ШУМА

Вывод статистических характеристик НОРМАЛЬНОГО БЕЛОГО ШУМА

Сохранить script-файл с именем `Noise` в папке пользователя `My_Folder`.

Запустить script-файл на выполнение.

Проверить содержимое Workspace после выполнения script-файла.

Пояснить:

- как обратиться к function-файлу из script-файла;
- как сохранить путь к собственной папке перед запуском script-файла;
- какие переменные сохраняются в Workspace после выполнения script-файла.

5.4. Задание на самостоятельную работу

Самостоятельное задание рекомендуется для закрепления полученных знаний и включает в себя следующие пункты:

1С. Создание script-файла.

Создать script-файл для решения СЛАУ

$$\mathbf{Ax} = \mathbf{b}. \quad (5.1)$$

Организовать ввод с клавиатуры матрицы **A** и вектора **b**.

Значения элементов вектора **x** использовать в качестве коэффициентов a_i для вычисления значения многочлена

$$y = a_N x^N + a_{N-1} x^{N-1} + \dots + a_1 x + a_0 \quad (5.2)$$

с помощью функции:

`y = polyval(a, x)`

где **a** — вектор коэффициентов многочлена (5.2) в порядке убывания степеней; **x** и **y** — значения аргумента и многочлена (скаляры, векторы или матрицы).

Организовать ввод с клавиатуры значения (значений) аргумента **x**.

Перед вводом исходных данных и выводом результатов добавить строки соответствующих сообщений.

2С. Создание script-файла.

Создать script-файл для генерации "магической" матрицы **М** с помощью функции:

М = magic(n)

где **n** и **М** — порядок и имя матрицы.

Организовать ввод с клавиатуры порядка матрицы.

Вычислить определитель матрицы.

Вычислить суммы элементов столбцов, строк и главной диагонали матрицы.

Растянуть матрицу в вектор-столбец, выполнить сортировку его элементов по возрастанию и вычислить сумму элементов, деленную на порядок матрицы.

Перед вводом порядка матрицы и выводом результатов добавить строки соответствующих сообщений.

3С. Создание function-файла.

Создать function-файл для генерации двух матриц одинакового порядка:

- теплицевой матрицы **Т** с произвольными целыми значениями элементов первого столбца;
- "магической" матрицы **М**.

В качестве входных параметров выбрать порядок матриц и элементы первого столбца теплицевой матрицы.

4С. Создание script-файла с использованием function-файла.

Создать script-файл для решения СЛАУ (5.1).

В качестве матрицы коэффициентов **А** использовать теплицевую матрицу **Т**, а свободных членов **b** — "магическую" матрицу **М**.

Для генерации матриц использовать function-файл, созданный в п. 3С.

Вычислить определитель матрицы коэффициентов.

Определить количество одновременно решаемых СЛАУ.

Перед выводом результатов добавить строки соответствующих сообщений.

5.5. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит результаты выполнения каждого пункта задания, включая листинги М-файлов (шрифт Courier New), результаты их выполнения, копируемые из окна **Command Window** (шрифт Courier New), созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**) и ответы на поставленные вопросы (шрифт Times New Roman).

При защите лабораторной работы набор контрольных вопросов формируется из следующего списка:

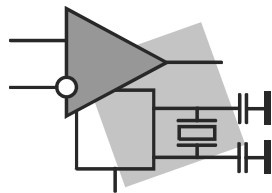
1. Для чего предназначен режим программирования?
2. Что такое М-файл?

3. Какие разновидности М-файлов создаются в режиме программирования?
4. Как вывести листинг М-файла?
5. Что такое script-файл и как к нему обратиться в режиме прямых вычислений?
6. Что такое function-файл и как к нему обратиться в режиме прямых вычислений и в script-файле?
7. Каков формат описания function-файла?
8. Какие переменные function-файла называют формальными и фактическими?
9. Какие переменные сохраняются в Workspace после выполнения script-файла?
10. Какие переменные сохраняются в Workspace после выполнения function-файла?
11. Какие переменные называют локальными и глобальными?
12. В каком окне создаются script- и function-файлы?
13. Как организовать ввод данных с клавиатуры в режиме программирования?
14. Как организовать вывод данных в окно **Command Window** в режиме программирования?
15. Как вывести на одной строке значение численной переменной одновременно с текстом?
16. В каких случаях целесообразно предусмотреть паузу?
17. Как сохранить М-файл в требуемой папке?
18. Как сохранить путь к данной папке?

5.6. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 7.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Приложения 1—2.

ГЛАВА 6



Режим программирования: организация разветвлений и циклов

Цель работы: изучить средства организации разветвлений и циклов в MATLAB и овладеть навыками их использования при разработке М-файлов.

6.1. Краткая теоретическая справка

Для организации разветвлений и циклов в М-файлах используются операторы языка MATLAB, рассматриваемые в следующих разделах.

6.1.1. Операторы организации разветвлений

Имеются две основные разновидности разветвлений, реализуемые двумя операторами MATLAB.

- Разветвление по условию выполняется с помощью оператора `if`, простейший формат которого с одним условием имеет вид:

```
if <условие>
    <фрагмент>
end
```

где <фрагмент> — фрагмент программы.

Действие оператора: если значение <условия> истинно (выполняется), то управление передается <фрагменту>, в противном случае управление передается части программы, следующей за `end`.

Условие представляет собой логическое выражение — простое, с одной операцией отношения (см. табл. 1.7), или более сложное, включающее логические операции (см. табл. 1.8).

Пример использования оператора `if` с простым условием:

```
if i==j
    a(i,j) = 1;
end
```


и с более сложным условием:

```
if (i==j) & ((i+j)>50)
    a(i,j) = 10;
end
```

Расширенный формат оператора if с одним условием имеет вид:

```
if <условие>
    <фрагмент1>
else
    <фрагмент2>
end
```

Действие оператора: если значение <условия> истинно, то управление передается <фрагменту1>, если значение <условия> ложно, то выполняется <фрагмент2>; после этого управление передается части программы, следующей за end.

Пример использования оператора if расширенного формата:

```
if i==j
    a(i,j) = 1;
else
    a(i,j) = -1;
end
```

Формат оператора if с несколькими условиями имеет вид:

```
if <условие1>
    <фрагмент1>
elseif <условие2>
    <фрагмент2>
...
elseif <условиеN-1>
    <фрагментN-1>
...
else
    <фрагментN>
end
```

Действие оператора: если значение <условия1> истинно, то управление передается <фрагменту1>, если значение <условия2> истинно, то управление передается <фрагменту2> и т. д. вплоть до <условияN-1>; если значения всех условий ложно, то управление передается <фрагментуN>; после этого оно передается части программы, следующей за end.

Пример использования оператора if с несколькими условиями:

```
if i>j
    a(i,j) = 1;
elseif i==j
    a(i,j) = -1;
```

```

else
    a(i,j) = 0;
end

```

- Разветвление в зависимости от значения выражения (арифметического, символического или логического) выполняется с помощью оператора `switch` следующего формата:

```

switch <выражение>
    case <значение1>
        <фрагмент1>
    case <значение2>
        <фрагмент2>
    ...
    otherwise
        <фрагментN>
end

```

Действие оператора: в зависимости от значения *<выражения>* управление передается соответствующему *<фрагменту>*; если значение выражения не равно ни одному из указанных, то управление передается *<фрагментуN>* (который может отсутствовать); после этого управление передается части программы, следующей за `end`.

Пример использования оператора `switch`:

```

x = [pi/6 pi/8 pi/16];
a = input('a = ');
b = input('b = ');
switch (a+b)
    case 0
        y = sin(x);
    case 1
        y = cos(x);
    otherwise
        y = tan(x)
end

```

6.1.2. Операторы организации циклов

Имеются две основные разновидности циклов, реализуемые двумя операторами MATLAB.

- Арифметический цикл с заранее известным (фиксированным) числом повторений организуется с помощью оператора `for` одного из следующих форматов:

- с простой переменной цикла:

```

for <переменная> = <нач. значение> : [<шаг> : ] <кон. значение>
    <тело цикла>
end

```

где *<переменная>* — имя простой переменной цикла; *<нач.значение>*, *<кон.значение>*, *<шаг>* — соответственно начальное и конечное значения переменной цикла и шаг ее изменения; если шаг равен 1, то его можно не указывать; *<тело цикла>* — повторяющийся фрагмент программы.

Действие оператора: при изменении значений *<переменной>* от *<нач.значения>* до *<кон.значения>* с заданным *<шагом>* повторяется *<тело цикла>*, каждый раз с новым значением *<переменной>*; после этого управление передается части программы, следующей за *end*.

Пример использования оператора *for* с простой переменной цикла (полужирным шрифтом выделены элементы, вычисляемые в цикле):

```
x = [2 3 5];
for i = 1:3
x(i) = i^2
end
x =
    1     3     5
x =
    1     4     5
x =
    1     4     9
```

- с переменной цикла — вектором:

```
for <переменная> = <вектор>
    <тело цикла>
end
```

Действие оператора: при изменении значений *<переменной>*, которой последовательно присваиваются значения элементов *<вектора>*, повторяется *<тело цикла>*, каждый раз с новым значением *<переменной>*; после этого управление передается части программы, следующей за *end*.

Пример использования оператора *for* с переменной цикла — вектором:

```
a = [-1 0 15];
for i = a
x = i+a
end
x =
   -2    -1    14
x =
   -1     0    15
x =
   14    15    30
```

- с переменной цикла — матрицей:

```
for <переменная> = <матрица>
    <тело цикла>
end
```

Действие оператора: при изменении значений *<переменной>*, которой последовательно присваиваются значения столбцов *<матрицы>*, повторяется *<тело цикла>*, каждый раз с новым значением *<переменной>*; после этого управление передается части программы, следующей за *end*.

Пример использования оператора *for* с переменной цикла — матрицей:

```
a = [1 2 3;4 5 6;7 8 9];
for i = a
x = i'
end
x =
     1     4     7
x =
     2     5     8
x =
     3     6     9
```

- Итерационный цикл с заранее неизвестным (не фиксированным) числом повторений организуется с помощью оператора *while* следующего формата:

```
while <условие>
    <тело цикла>
end
```

где *<условие>* — логическое выражение, в котором хотя бы одна из переменных встречается в *<теле цикла>*.

Действие оператора: *<тело цикла>* повторяется до тех пор, пока *<условие>* истинно, после чего управление передается части программы, следующей за *end*.

Пример использования оператора *while* для вычисления суммы геометрической прогрессии $s = \sum_{n=0}^{\infty} (-0,5)^n$ (*s*) с точностью до $\varepsilon = 10^{-4}$ (*e*) с выводом после выхода

из цикла значения суммы и погрешности ее вычисления (вектор [*s e*]):

```
n = 0; s0 = 0; e = 100;
while e>1e-4
s = s0+(-0.5).^n;
e = abs(s-s0);
s0 = s;
n = n+1;
end
[s e]
ans =
    0.6667    0.0001
```

Принудительный выход из цикла *for* или *while* реализуется оператором:

break

после которого управление передается части программы, следующей за *end*.

6.2. Содержание лабораторной работы

Содержание работы связано с изучением средств MATLAB для организации разветвлений и циклов при разработке script-файлов и function-файлов.

6.3. Задание на лабораторную работу

Задание на лабораторную работу включает в себя следующие пункты:

1. Организация разветвлений с одним условием.

Создать function-файл y_1 для вычисления функции

$$y_1(x) = \begin{cases} a \sin bx, & \text{если } a \neq 0 \text{ и } b \neq 0; \\ (a+2)x + b, & \text{иначе,} \end{cases} \quad (6.1)$$

где аргумент x задан на интервале $x \in [-4; 4]$ с шагом $\Delta x = 0,1$; a, b — произвольные вещественные константы (скаляры).

Вывести график функции $y_1(x)$.

Обратиться к function-файлу y_1 в режиме прямых вычислений для проверки разветвления по условию в (6.1).

Пояснить:

- какой оператор использован для организации разветвления;
- какие параметры function-файла y_1 являются входными и выходными.

2. Организация разветвлений с несколькими условиями.

Создать function-файл y_2 для вычисления функции

$$y_2(x) = \begin{cases} a \sin bx, & \text{если } a \neq 0 \text{ и } b \neq 0; \\ (a+2)x + b, & \text{если } a > -2 \text{ и } b > 0; \\ (2-a)x^2 + b, & \text{иначе,} \end{cases} \quad (6.2)$$

где аргумент x и скаляры a и b определены в п. 1.

Вывести график функции $y_2(x)$.

Обратиться к function-файлу y_2 в режиме прямых вычислений для проверки разветвления по условиям в (6.2).

Пояснить, какой оператор использован для организации разветвления.

3. Организация цикла с заранее известным числом повторений.

Создать function-файл `Fibonacci` для формирования ряда Фибоначчи — вектора F из M членов, где каждый следующий член равен сумме двух предыдущих:

$$F_i = F_{i-1} + F_{i-2}, \quad i = 3, 4, \dots, M. \quad (6.3)$$

Задать начальные значения $F_1 = 0$ и $F_2 = 1$.

Обратиться к function-файлу `Fibonacci` в режиме прямых вычислений для вывода ряда Фибоначчи.

Пояснить:

- какой оператор использован для организации цикла;
- какие параметры function-файла `Fibonacci` являются входными и выходными.

4. Организация цикла с заранее неизвестным числом повторений.

Создать function-файл `GeomProgression` для вычисления в цикле суммы бесконечной геометрической прогрессии:

$$S = \sum_{n=0}^{\infty} q^n \quad (6.4)$$

с заданной точностью ε и значением q , при котором выполняется условие абсолютной сходимости ряда.

После выхода из цикла вычислить точное значение суммы (6.4) по формуле:

$$S_{\text{true}} = \frac{1}{1-q},$$

и погрешность вычисления суммы:

$$\Delta S = |S - S_{\text{true}}|.$$

Вывести значения S , S_{true} , ε и ΔS .

Обратиться к function-файлу `GeomProgression` в режиме прямых вычислений для вывода требуемых значений.

Пояснить:

- какой оператор использован для организации цикла;
- какие параметры function-файла `GeomProgression` являются входными и выходными;
- как сопоставить выведенные значения.

5. Организация разветвления в зависимости от значения выражения.

Создать script-файл `DifferentFunctions` для выполнения одного из function-файлов: `y1`, `y2`, `Fibonacci` или `GeomProgression`, в зависимости от значения переменной `variant`.

В script-файле организовать:

- вывод сообщения о соответствии значения переменной `variant` function-файлу;
- ввод значения переменной `variant` с клавиатуры;

- разветвление в зависимости от значения переменной `variant` с выводом сообщения об исполняемом function-файле;
- вывод сообщения для непредусмотренного значения переменной `variant`, не соответствующего ни одному из function-файлов.

Обратиться к script-файлу `DifferentFunctions` в режиме прямых вычислений для проверки требуемого разветвления.

Пояснить:

- какой оператор использован для организации разветвления;
- что проверяется при организации разветвления;
- к какому типу данных может принадлежать переменная `variant`.

6.4. Задание на самостоятельную работу

Самостоятельное задание рекомендуется для закрепления полученных знаний и включает в себя следующие пункты:

1С. Организация разветвления по условию.

Создать function-файл для решения квадратного уравнения

$$ax^2 + bx + c = 0$$

двумя способами:

- используя известную алгебраическую формулу;
- с помощью функции вычисления корней многочлена производного порядка:

`x = roots(a)`

где `a` — вектор коэффициентов в порядке убывания степеней, `x` — корни многочлена (вектор).

В том случае, если корни оказались комплексно сопряженными, вычислить и вывести их модуль и аргументы.

2С. Организация разветвления в зависимости от значения выражения.

Создать function-файл для вычисления значения одной из следующих функций:

$$y(x) = \begin{cases} ax - b, & (a + b) = 0, 8; \\ ax^2 + b, & (a + b) = 2, 5; \\ \sin bx - a, & (a + b) = -3, 4; \\ \cos ax + b, & \text{иначе.} \end{cases}$$

Построить график функции $y(x)$ на выбранном интервале по оси x с помощью функции `plot`.

Обратиться к function-файлу в режиме прямых вычислений, задавая значения a и b , при которых будут выведены графики различных функций.

3С. Организация цикла с заранее известным числом повторений.

Создать function-файл для вычисления суммы конечного ряда при $0 < |x| < 1$:

$$S = \sum_{n=0}^{N-1} \frac{(-1)^n \sqrt{(n+1)x}}{n+1}.$$

4С. Организация цикла с заранее неизвестным числом повторений.

Создать function-файл для вычисления суммы бесконечного ряда с заданной точностью ε при $0 < |x| < 1$:

$$S = \sum_{n=0}^{\infty} \frac{(-1)^n \sqrt{(n+1)x}}{n+1}.$$

Определить количество циклов, требуемое для вычисления суммы при заданной точности.

6.5. Отчет и контрольные вопросы

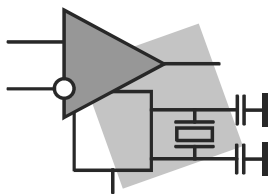
Отчет составляется в редакторе MS Word и содержит результаты выполнения каждого пункта задания, включая листинги М-файлов (шрифт Courier New), результаты их выполнения, копируемые из окна **Command Window** (шрифт Courier New), созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**) и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. Поясните назначение и формат оператора `if`.
2. Поясните назначение и формат оператора `switch`.
3. Поясните назначение и формат оператора `for`.
4. Поясните назначение и формат оператора `while`.
5. Как выполнить принудительный выход из цикла? Какой части программы передается управление в этом случае?

6.6. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 7.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Приложения 1—2.

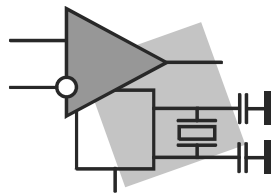


ЧАСТЬ II

Моделирование цифровой обработки сигналов в MATLAB

- Глава 7.** Дискретные сигналы
- Глава 8.** Линейные дискретные системы
- Глава 9.** Дискретное преобразование Фурье (часть 1)
- Глава 10.** Дискретное преобразование Фурье (часть 2)
- Глава 11.** Синтез КИХ-фильтров методом окон
- Глава 12.** Синтез КИХ-фильтров методом наилучшей равномерной (чебышевской) аппроксимации
- Глава 13.** Синтез БИХ-фильтров методом билинейного Z-преобразования
- Глава 14.** Синтез цифровых фильтров средствами GUI FDATool и FilterBuilder
- Глава 15.** Цифровые фильтры с фиксированной точкой
- Глава 16.** Спектральный анализ: непараметрические методы
- Глава 17.** Спектральный анализ: параметрические методы
- Глава 18.** Спектральный анализ средствами GUI SPTool
- Глава 19.** Многоскоростные системы ЦОС
- Глава 20.** Моделирование полифазных структур многоскоростных систем средствами GUI FDATool и FilterBuilder
- Глава 21.** Адаптивные фильтры

ГЛАВА 7



Дискретные сигналы

Цель работы: изучить математическое описание дискретных сигналов и овладеть программными средствами их моделирования в MATLAB.

7.1. Краткая теоретическая справка

В теории ЦОС принято разделять операции дискретизации по времени и квантования по уровню. Полагая операцию квантования отсутствующей, изучают дискретные сигналы и линейные дискретные системы (ЛДС), а затем, отдельно, — эффекты нелинейной операции квантования.

Дискретным называют сигнал, дискретный по времени и непрерывный по состоянию (уровню), который описывается последовательностью чисел *бесконечной* разрядности $x(nT)$ или $x(n)$, называемой коротко *последовательностью*. Значения nT , $n = 0, 1, \dots$, называют *дискретным временем*, где T — период дискретизации, а n — *дискретным нормированным временем*.

В теории ЦОС термины "*дискретный сигнал*" и "*последовательность*" употребляют в тождественном смысле.

Цифровым называют сигнал, дискретный по времени и квантованный по состоянию (уровню), который описывается последовательностью чисел *конечной* разрядности — *квантованной последовательностью* $\tilde{x}(nT)$ или $\tilde{x}(n)$.

При компьютерном моделировании под дискретным сигналом условно понимают последовательность чисел *максимально возможной* разрядности, а под цифровым — последовательность чисел *заданной* разрядности.

В MATLAB числа с максимальной разрядностью относятся к типу `double`¹, который выбирается по умолчанию (см. табл. 3.1).

¹ С плавающей точкой двойной точности (double-precision floating point).

7.1.1. Детерминированные дискретные сигналы

Детерминированным дискретным сигналом называют сигнал, значения которого в любой момент времени n (или nT) заранее известны или могут быть определены точно по заданной математической модели.

Детерминированный дискретный сигнал описывается последовательностью $x(nT)$ или $x(n)$, при этом термин "детерминированный" принято опускать.

Для детерминированного дискретного сигнала (последовательности) представляют интерес такие его характеристики, как среднее значение, энергия, средняя мощность, автокорреляционная и автоковариационная функции.

Средним значением последовательности называют сумму ее значений, отнесенную к длине.

Энергией последовательности называют сумму квадратов ее значений, а *средней мощностью* — энергию, отнесенную к длине последовательности.

В MATLAB среднее значение m вычисляется с помощью функции:

`m = mean(x)`

где x — вектор отсчетов последовательности.

Энергия E и средняя мощность P вычисляются согласно их определению:

`E = sum(x.^2)`

`P = sum(x.^2)/length(x)`

где `length(x)` — длина последовательности.

Автокорреляционная функция (АКФ)¹ $R_x(m)$ последовательности длины N позволяет оценить зависимость между ее отсчетами при различных сдвигах по времени m :

$$R_x(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x(n)x(n+m), \quad -(N-1) \leq m \leq (N-1). \quad (7.1)$$

Автоковариационная функция $r_x(m)$ позволяет оценить зависимость между отклонениями отсчетов последовательности от среднего значения μ_x при различных сдвигах по времени m :

$$r_x(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} [x(n) - \mu_x][x(n+m) - \mu_x], \quad -(N-1) \leq m \leq (N-1). \quad (7.2)$$

Согласно определению, $R_x(m)$ (7.1) и $r_x(m)$ (7.2) являются четными функциями длины $L = 2N - 1$, центрированными относительно $m = 0$:

$$R_x(m) = R_x(-m);$$

$$r_x(m) = r_x(-m).$$

¹ В англоязычной литературе — аббревиатура ACF (Autocorrelation Function).

При этом в точке $m = 0$ имеем:

$$R_x(0) = \frac{1}{N} \sum_{n=0}^{N-1} x^2(n) = P_{\text{ср } x} = \sigma_x^2 + \mu_x^2; \quad (7.3)$$

$$r_x(0) = \frac{1}{N} \sum_{n=0}^{N-1} [x(n) - \mu_x]^2 = \sigma_x^2, \quad (7.4)$$

где $P_{\text{ср } x}$ и σ_x^2 — средняя мощность и дисперсия последовательности $x(n)$.

Очевидно, что при $\mu_x = 0$ получаем равенства:

$$R_x(m) = r_x(m);$$

$$R_x(0) = r_x(0) = \sigma_x^2.$$

В MATLAB АКФ и автоковариационная функция рассчитываются с помощью функций (без учета множителя $1/N$):

R = xcorr(x)

r = xcov(x)

где x — вектор отсчетов исходной последовательности длины N ; **R** и **r** — векторы длины $L = 2N - 1$ значений АКФ $R_x(m)$ и автоковариационной функции $r_x(m)$, соответственно, центрированных относительно $m = N$:

$$R_x(N + m) = R_x(N - m), \quad m = 1, 2, \dots, N - 1; \quad (7.5)$$

$$r_x(N + m) = r_x(N - m), \quad m = 1, 2, \dots, N - 1. \quad (7.6)$$

При этом в точке $m = N$ имеем:

$$R_x(N) = P_{\text{ср } x} = \sigma_x^2 + \mu_x^2; \quad (7.7)$$

$$r_x(N) = \sigma_x^2. \quad (7.8)$$

Для вывода графика АКФ, центрированного относительно $m = 0$, следует выбрать интервал $m \in [-(N - 1); (N - 1)]$.

7.1.2. Случайные дискретные сигналы

Случайным (стохастическим) дискретным сигналом называют сигнал, значения которого в дискретные моменты времени n (или nT) заранее неизвестны и могут быть определены лишь с некоторой вероятностью.

Случайный дискретный сигнал описывается совокупностью случайных последовательностей $x_k(n)$, $k = 1, 2, \dots, K$, $n = 0, 1, \dots, (N - 1)$, и закономерностями, характеризующими свойства совокупности.

Описание случайного дискретного сигнала удобно представить в виде матрицы $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} x_1(0) & x_1(1) & \dots & x_1(n) & \dots & x_1(N-1) \\ x_2(0) & x_2(1) & \dots & x_2(n) & \dots & x_2(N-1) \\ \vdots & \vdots & & \vdots & & \vdots \\ x_K(0) & x_K(1) & \dots & x_K(n) & \dots & x_K(N-1) \end{bmatrix}.$$

Ансамблем реализаций называют совокупность случайных последовательностей $x_k(n)$ (строки матрицы $\mathbf{X}$), а *реализацией* — одну из последовательностей.

Любая реализация случайного сигнала представляет собой детерминированный сигнал.

В большинстве случаев в качестве закономерностей, характеризующих свойства дискретного случайного сигнала $\mathbf{X}$, ограничиваются одномерной и двумерной плотностями вероятности.

Одномерная плотность вероятности случайного дискретного сигнала $p(x, n)$, где x — значения случайного сигнала $\mathbf{X}$ в моменты времени n , позволяет посредством статистического усреднения¹ при достаточно большом K (теоретически $K \rightarrow \infty$) определить следующие *статистические характеристики* случайного дискретного сигнала:

- математическое ожидание $\mu_{\mathbf{X}}(n)$ — средние значения элементов столбца в моменты времени n , $n = 0, 1, \dots, (N-1)$;
- дисперсию $\sigma_{\mathbf{X}}^2(n)$ — средние значения квадратов разностей между элементами столбца и его средним значением $\mu_{\mathbf{X}}(n)$ в моменты времени n , $n = 0, 1, \dots, (N-1)$.

Двумерная плотность вероятности случайного дискретного сигнала $p(x_1, x_2, m, n)$, где x_1, x_2 — значения сигнала $\mathbf{X}$ в моменты времени m и n , позволяет посредством статистического усреднения определить дополнительные *статистические характеристики* случайного дискретного сигнала:

- АКФ $R_{\mathbf{X}}(m, n)$ — АКФ (7.1), где последовательности $x(n)$ соответствует усредненная по ансамблю последовательность $\mu_{\mathbf{X}}(n)$, $n = 0, 1, \dots, (N-1)$;
- автоковариационную функцию $r_{\mathbf{X}}(m, n)$ — автоковариационная функция (7.2), где значению $\mu_{\mathbf{X}}$ соответствует среднее значение $\mu_{\mathbf{X}}(n)$ — константа.

Случайный дискретный сигнал $\mathbf{X}$ называют *стационарным в широком смысле* (стационарным по Хинчину), если его одномерная плотность вероятности не зависит от времени n , а двумерная — зависит только от сдвига по времени m .

¹ Под *статистическим усреднением* понимают усреднение по ансамблю реализаций в фиксированный момент времени n .

Случайный дискретный сигнал X называют *стационарным в узком смысле* (строго стационарным), если сказанное справедливо для его любой n -мерной плотности вероятности [2].

Таким образом, сигнал, стационарный в узком смысле, всегда стационарен в широком смысле, но не наоборот.

По умолчанию под *стационарностью* случайного дискретного сигнала будем подразумевать его стационарность *в широком смысле*.

Следствием *стационарности* случайного дискретного сигнала будет *независимость от времени n* его статистических характеристик: математического ожидания μ_X и дисперсии σ_X^2 . При этом АКФ $R_X(m)$ и автоковариационная функция $r_X(m)$ будут зависеть только от сдвига по времени m .

Иными словами, статистические характеристики *стационарного* случайного дискретного сигнала обладают свойством *инвариантности*¹ *во времени*.

Соответственно, статистические характеристики *нестационарного* случайного дискретного сигнала будут *зависеть от времени n* (не обладают свойством инвариантности во времени).

В теории ЦОС понятие ансамбля реализаций широко используется как удобная математическая концепция при выводе многих соотношений. Однако на практике при обработке сигналов, как правило, доступна для наблюдения лишь одна реализация случайного дискретного сигнала.

Стационарный случайный дискретный сигнал называется *эргодическим*, если при определении его статистических характеристик усреднение *по ансамблю реализаций* эквивалентно *усреднению по времени* одной реализации, теоретически бесконечной длины $N \rightarrow \infty$.

Эргодический случайный дискретный сигнал — *случайная последовательность* $x(n)$ — описывается математическим ожиданием (средним значением) μ_x , дисперсией σ_x^2 , АКФ $R_x(m)$ и автоковариационной функцией $r_x(m)$. С их определением при $N \rightarrow \infty$ можно познакомиться в [2—3].

При *конечной* длине N последовательности говорят о вычислении их *оценок*:

$$\hat{\mu}_x = \frac{1}{N} \sum_{n=0}^{N-1} x(n);$$

$$\hat{\sigma}_x^2 = \frac{1}{N} \sum_{n=0}^{N-1} [x(n) - \hat{\mu}_x]^2.$$

Оценки АКФ $\hat{R}_x(m)$ и автоковариационной функции $\hat{r}_x(m)$ получают соответственно по формулам (7.1) и (7.2).

¹ Неизменности.

Очевидно, что статистические характеристики *эргодического* случайного дискретного сигнала, по определению стационарного, обладают свойством *инвариантности во времени*.

При обработке случайного сигнала в реальном времени его статистическая модель¹ может быть заранее не определена. В этом случае принято говорить о *текущих* оценках статистических характеристик $\mu_x(n)$, $\sigma_x^2(n)$, $R_x(m, n)$, $r_x(m, n)$ на интервале $[0; n]$.

Далее по умолчанию будем подразумевать *эргодические* случайные дискретные сигналы.

В MATLAB для вычисления *оценок* математического ожидания m и дисперсии D используются функции:

`m = mean(x)`

`D = var(x)`

где x — вектор отсчетов исходной последовательности длины N .

При моделировании методов и алгоритмов ЦОС часто используют случайные последовательности в виде белого шума. Две его широко применяемые разновидности генерируются в MATLAB (см. табл. 2.1):

- равномерный белый шум — последовательность случайных чисел из диапазона $[0; 1]$, распределенных по равномерному закону (математическое ожидание — 0,5 и дисперсия — $1/12$) — с помощью функции:

`x = rand(1, N)`

где x — вектор-строка отсчетов случайной последовательности длины N .

Автоковариационная функция данного равномерного белого шума при $N \rightarrow \infty$ имеет вид *цифрового единичного импульса*,

- нормальный белый шум — последовательность случайных чисел, распределенных по нормальному закону (математическое ожидание — 0 и дисперсия — 1) — с помощью функции:

`x = randn(1, N)`

АКФ данного нормального белого шума при $N \rightarrow \infty$ имеет вид *цифрового единичного импульса*.

Для моделирования нормального белого шума с заданными математическим ожиданием (средним значением) и дисперсией воспользуемся свойствами дисперсии $D\{X\}$ и математического ожидания $M\{X\}$ случайной величины X :

$$M\{X + C\} = M\{X\} + C;$$

$$D\{X + C\} = D\{X\} + D\{C\} = D\{X\};$$

¹ Стационарность/нестационарность, статистические характеристики.

$$M\{BX\} = BM\{X\};$$
$$D\{BX\} = B^2D\{X\},$$

где C, B — константы.

Таким образом, на основе случайной величины X с нулевым математическим ожиданием $M\{X\}=0$ и единичной дисперсией $D\{X\}=1$ можно получить случайную величину $\tilde{X}$:

$$\tilde{X} = BX + C \tag{7.9}$$

с математическим ожиданием $M\{\tilde{X}\}=C$ и дисперсией $D\{\tilde{X}\}=B^2$.

7.2. Содержание лабораторной работы

Содержание работы связано с моделированием детерминированных и случайных последовательностей, в том числе типовых последовательностей, и расчетом их характеристик программными средствами MATLAB.

7.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файла `lr_07`, который хранится на прилагаемом компакт-диске в папке `LAB_DSP\LAB_07`.

Перед выполнением работы необходимо сохранить путь к папке `LAB_07` по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 7.1 для номера бригады $N_{бр}$, где $N_{бр}=1, 2, \dots, 30$. Функция $N_{бр} \bmod M$ в записи исходных данных означает вычисление значения $N_{бр}$ по модулю M .

На прилагаемом компакт-диске в папке `Tables\Tables_07` хранятся табл. 7.1 исходных данных и пример ее заполнения для $N_{бр}=1$.

Таблица 7.1. Таблица исходных данных

Переменная	Назначение	Значение	Идентификатор
$N_{бр}$	Номер бригады	$N_{бр}$	Nb =
N	Длина последовательности	$N = 30 + N_{бр} \bmod 5$	N =
T	Период дискретизации	$T = 0,0005(1 + N_{бр} \bmod 3)$	T =
a	Основание экспоненты	$a = (-1)^{N_{бр}}(0,8 + 0,005N_{бр})$	a =

Таблица 7.1 (окончание)

Переменная	Назначение	Значение	Идентификатор
C	Амплитуда гармонического сигнала	$C = 1 + N_{\text{бп}} \bmod 5$	$C =$
$\hat{\omega}_0$ (рад)	Частота гармонического сигнала	$\hat{\omega}_0 = \pi / (6 + N_{\text{бп}} \bmod 5)$	$w0 =$
m	Задержка	$m = 5 + N_{\text{бп}} \bmod 5$	$m =$
U	Амплитуда импульса	$U = N_{\text{бп}}$	$U =$
n_0	Начальный момент импульса	$n_0 = N_{\text{бп}} \bmod 5 + 3$	$n0 =$
n_{imp}	Длина импульса	$n_{\text{imp}} = N_{\text{бп}} \bmod 5 + 5$	$n_{\text{imp}} =$
B_1, B_2, B_3	Амплитуды гармонических сигналов	$B_1 = 1, 5 + N_{\text{бп}} \bmod 5$ $B_2 = 5, 7 - N_{\text{бп}} \bmod 5$ $B_3 = 2, 2 + N_{\text{бп}} \bmod 5$	Вектор $B = [\dots]$
$\hat{\omega}_1, \hat{\omega}_2, \hat{\omega}_3$	Частоты гармонических сигналов	$\hat{\omega}_1 = \pi / (4 + N_{\text{бп}} \bmod 5)$ $\hat{\omega}_2 = \pi / (8 + N_{\text{бп}} \bmod 5)$ $\hat{\omega}_3 = \pi / (16 + N_{\text{бп}} \bmod 5)$	Вектор $w = [\dots]$
a_1, a_2, a_3	Коэффициенты линейной комбинации гармонических сигналов	$a_1 = 1, 5 - N_{\text{бп}} \bmod 5$ $a_2 = 0, 7 + N_{\text{бп}} \bmod 5$ $a_3 = 1, 4 + N_{\text{бп}} \bmod 5$	Вектор $A = [\dots]$
mean	Математическое ожидание	$\text{mean} = N_{\text{бп}} \bmod 5 + 3$	$\text{Mean} =$
var	Дисперсия	$\text{var} = N_{\text{бп}} \bmod 5 + 5$	$\text{Var} =$

Задание на лабораторную работу связано с моделированием и анализом последовательностей и включает в себя следующие пункты:

1. Цифровой единичный импульс $u_0(nT)$ (идентификатор $u0$):

$$u_0(nT) = \begin{cases} 1, & n = 0; \\ 0, & n \neq 0 \end{cases}$$

(7.10)

с выводом графиков на интервале дискретного времени nT (идентификатор nT):

$$nT \in [0; (N-1)T] \quad (7.11)$$

и дискретного нормированного времени n (идентификатор n):

$$n \in [0; (N-1)]. \quad (7.12)$$

Пояснить:

- взаимосвязь между дискретным и дискретным нормированным временем;
- различие между цифровым единичным импульсом и дельта-функцией.

2. Цифровой единичный скачок $u_1(nT)$ (идентификатор $u1$):

$$u_1(nT) = \begin{cases} 1, & n \geq 0; \\ 0, & n < 0 \end{cases} \quad (7.13)$$

с выводом графиков на интервалах времени (7.11) и (7.12).

Пояснить:

- соответствие между цифровым и аналоговым единичными скачками;
- чему равна частота дискретизации цифрового единичного скачка.

3. Дискретная экспонента $x_1(nT)$ (идентификатор $x1$):

$$x_1(nT) = \begin{cases} a^n, & n \geq 0; \\ 0, & n < 0 \end{cases} \quad (7.14)$$

с выводом графиков на интервалах времени (7.11) и (7.12).

Пояснить соответствие между дискретной и аналоговой экспонентами.

4. Дискретный комплексный гармонический сигнал $x_2(n)$ (идентификатор $x2$):

$$x_2(n) = Ce^{j\hat{\omega}_0 n} \quad (7.15)$$

с выводом графиков вещественной и мнимой частей на интервале времени (7.12).

Записать сигнал (7.15) в виде комбинации двух вещественных последовательностей.

5. Задержанные последовательности.

Вывести графики последовательностей (7.10), (7.13) и (7.14), задержанных на m отсчетов (идентификаторы $u0\_m$, $u1\_m$ и $x1\_m$), на интервале времени (7.12).

Записать формулы задержанных последовательностей.

6. Дискретный прямоугольный импульс $x_3(n)$:

$$x_3(n) = \begin{cases} U, & n_0 \leq n \leq (n_0 + n_{imp} - 1); \\ 0, & \text{иначе} \end{cases} \quad (7.16)$$

с выводом графика на интервале времени (7.12).

Выполнить моделирование импульса двумя способами:

- с помощью функции `rectpuls` — идентификатор `x3_1`;
- на основе цифрового единичного скачка — идентификатор `x3_2`.

Пояснить:

- формат функции `rectpuls` (познакомиться самостоятельно);
- как выполняется моделирование импульса в обоих случаях.

7. Дискретный треугольный импульс.

Вывести график дискретного треугольного импульса $x_4(n)$ (идентификатор `x4`), сформированного посредством свертки дискретного прямоугольного импульса $x_3(n)$ (7.16) с самим собой, на интервале времени, равном длине свертки L :

$$n \in [0; (L - 1)]. \quad (7.17)$$

Для вычисления свертки использовать функцию:

`conv(x, y)`

где x, y — сворачиваемые последовательности.

Привести аналитическую запись свертки. Определить теоретически и по графику длину свертки L и ширину треугольного импульса.

8. Линейная комбинация дискретных гармонических сигналов $x_5(n)$ (идентификатор `x5`):

$$x_5(n) = a_1 x_1(n) + a_2 x_2(n) + a_3 x_3(n), \quad (7.18)$$

где

$$x_i(n) = B_i \sin(\hat{\omega}_i n), \quad i = 1, 2, 3, \quad (7.19)$$

с выводом графиков последовательностей $x_i(n)$ и $x_5(n)$ на интервале времени

$$n \in [0; (5N - 1)]. \quad (7.20)$$

Вычислить среднее значение (идентификатор `mean_x5`), энергию (идентификатор `E`) и среднюю мощность (идентификатор `P`) последовательности (7.18).

Пояснить:

- операции при моделировании линейной комбинации сигналов (7.18);
- как определяют указанные характеристики.

9. Дискретный гармонический сигнал с экспоненциальной огибающей.

Вывести график дискретного сигнала $x_6(n)$ (идентификатор `x6`), представляющего собой дискретный гармонический сигнал $x(n)$ (идентификатор `x`)

$$x(n) = C \sin(\hat{\omega}_0 n) \quad (7.21)$$

с экспоненциальной огибающей $|a|^n$, на интервале времени (7.12).

Привести аналитическую формулу дискретного сигнала $x_6(n)$ и пояснить операции при его моделировании.

10. Периодическая последовательность дискретных прямоугольных импульсов.

Вывести график пяти периодов периодической последовательности $x_7(n)$ (идентификатор `x7`) дискретных прямоугольных импульсов амплитуды U и длительности n_{imp} с периодом, вдвое большим длительности импульса.

Для формирования пяти периодов последовательности выполнить действия:

- на основе цифрового единичного скачка (7.13) сформировать один период последовательности (идентификатор `xp`);
- сформировать пять периодов последовательности с помощью функции `repmat` (см. разд. 2.1.2).

Пояснить операции при моделировании периодической последовательности.

11. Равномерный белый шум.

Вычислить оценки математического ожидания (идентификатор `mean_uniform`) и дисперсии (идентификатор `var_uniform`) равномерного белого шума (идентификатор `r_uniform`) длины 10 000 с математическим ожиданием и дисперсией, установленными по умолчанию.

Вывести график оценки автоковариационной функции $\hat{r}_x(m)$ шума (идентификатор `r_r_uniform`), центрированной относительно $m = 0$.

Пояснить:

- чему равны истинные значения математического ожидания и дисперсии;
- каков вид истинной автоковариационной функции;
- чему равна длина оценки автоковариационной функции.

12. Нормальный белый шум.

Вычислить оценки математического ожидания (идентификатор `mean_norm`) и дисперсии (идентификатор `var_norm`) нормального белого шума (идентификатор `r_uniform`) длины 10 000 с математическим ожиданием и дисперсией, установленными по умолчанию.

Вывести график оценки АКФ $\hat{R}_x(m)$ шума (идентификатор `R_r_norm`), центрированной относительно $m = 0$.

Пояснить:

- чему равны истинные значения математического ожидания и дисперсии;
- каков вид истинной АКФ;
- чему равна длина оценки АКФ.

13. Аддитивная смесь $x_g(n)$ (идентификатор `x8`) дискретного гармонического сигнала $x(n)$ (7.21) с нормальным белым шумом с выводом графика на интервале времени (7.12).

Пояснить, что понимают под аддитивной смесью сигнала с шумом.

14. Оценка АКФ $\hat{R}_x(m)$ (идентификатор `R`) последовательности $x_g(n)$ (см. п. 13) с выводом графика АКФ, центрированной относительно $m = 0$.

Вывести оценку дисперсии последовательности $x_g(n)$ и значение $R_x(N)$.

Пояснить:

- свойства АКФ;
- соответствие между выведенными значениями.

15. Нормальный белый шум с заданными статистическими характеристиками.

С помощью функции `plot` вывести графики четырех разновидностей нормального белого шума длины 10 000:

- с математическим ожиданием и дисперсией, установленными по умолчанию, — идентификатор шума `r_norm` (см. п. 12);
- с математическим ожиданием `mean` и дисперсией, установленной по умолчанию, — идентификатор шума `r_normMean`;
- с математическим ожиданием, установленным по умолчанию, и дисперсией `var` — идентификатор шума `r_normVar`;
- с математическим ожиданием `mean` и дисперсией `var` — идентификатор шума `r_normMeanVar`.

Для наглядности вывести графики шумов в одинаковом диапазоне по оси ординат $[-MAX \ MAX]$ с помощью функции `ylim`, где `MAX` равно максимальному значению шума среди четырех его разновидностей.

Построить гистограммы четырех разновидностей нормального белого шума с помощью функции `hist` (параметры задать по умолчанию).

Для наглядности вывести гистограммы в одинаковом диапазоне по оси абсцисс $[-MAX \ MAX]$ с помощью функции `xlim`, где значение `MAX` определено ранее.

В заголовке гистограмм вывести значения оценок математического ожидания (Mean value) и дисперсии (Variance).

Пояснить:

- к каким изменениям шума приводит изменение его математического ожидания и дисперсии;
- что отображает гистограмма и как она изменяется при изменении математического ожидания и дисперсии шума.

7.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должна быть представлена табл. 7.1 исходных данных для своего номера бригады $N_{бр}$.

Для *запуска* лабораторной работы необходимо обратиться к script-файлу `lr_07` по его имени:

```
>> lr_07
```

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш <Ctrl>+<Break>.

При выполнении script-файла текущие окна с графиками *не закрывать*.

Листинг script-файла `lr_07` имеет вид:

```
>> type lr_07
script
clc
clear
disp('% ЛР №7. ДИСКРЕТНЫЕ СИГНАЛЫ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
N = input('N = ');             % ДЛИНА ПОСЛЕДОВАТЕЛЬНОСТИ
T = input('T = ');             % ПЕРИОД ДИСКРЕТИЗАЦИИ
a = input('a = ');             % ОСНОВАНИЕ ДИСКРЕТНОЙ ЭКСПОНЕНТЫ
C = input('C = ');             % АМПЛИТУДА ДИСКРЕТНОГО ГАРМОНИЧЕСКОГО СИГНАЛА
w0 = input('w0 = ');           % ЧАСТОТА ДИСКРЕТНОГО ГАРМОНИЧЕСКОГО СИГНАЛА
m = input('m = ');             % ВЕЛИЧИНА ЗАДЕРЖКИ
U = input('U = ');             % АМПЛИТУДА ИМПУЛЬСА
n0 = input('n0 = ');           % МОМЕНТ НАЧАЛА ИМПУЛЬСА
n_imp = input('n_imp = ');     % ДЛИТЕЛЬНОСТЬ ИМПУЛЬСА
B = input('B = ');             % ВЕКТОР АМПЛИТУД
w = input('w = ');             % ВЕКТОР ЧАСТОТ
A = input('A = ');             % ВЕКТОР КОЭФФИЦИЕНТОВ ЛИНЕЙНОЙ КОМБИНАЦИИ
Mean = input('Mean = ');       % ЗАДАННОЕ МАТЕМАТИЧЕСКОЕ ОЖИДАНИЕ ШУМА
Var = input('Var = ');         % ЗАДАННАЯ ДИСПЕРСИЯ ШУМА
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
```



```

disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.1. ЦИФРОВОЙ ЕДИНИЧНЫЙ ИМПУЛЬС')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ цифрового единичного импульса нажмите <ENTER>')
pause
n = 0:(N-1); nT = T.*n;           % ДИСКРЕТНОЕ НОРМИРОВАННОЕ И НЕНОРМИРОВАННОЕ ВРЕМЯ
u0 = [1 zeros(1, (N-1))];        % ЦИФРОВОЙ ЕДИНИЧНЫЙ ИМПУЛЬС
figure('Name','Digital Unit Impulse, Unit Step, and Discrete
Exponent','NumberTitle','off')
subplot(3,2,1),stem(nT,u0,'Linewidth',2), grid
title('Digital Unit Impulse u0(nT)')
subplot(3,2,2),stem(n,u0,'Linewidth',2), grid
title('Digital Unit Impulse u0(n)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ЦИФРОВОЙ ЕДИНИЧНЫЙ СКАЧОК');
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ цифрового единичного скачка нажмите <ENTER>')
pause
u1 = [1 ones(1, (N-1))];         % ЦИФРОВОЙ ЕДИНИЧНЫЙ СКАЧОК
subplot(3,2,3),stem(nT,u1,'Linewidth',2), grid
title('Digital Unit Step u1(nT)'),
subplot(3,2,4),stem(n,u1,'Linewidth',2), grid
title('Digital Unit Step u1(n)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. ДИСКРЕТНАЯ ЭКСПОНЕНТА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ дискретной экспоненты нажмите <ENTER>')
pause

```

```

x1 = a.^n; % ДИСКРЕТНАЯ ЭКСПОНЕНТА
subplot(3,2,5),stem(nT,x1,'Linewidth',2), xlabel('nT'), grid
title('Discrete Exponent x1(nT)')
subplot(3,2,6),stem(n, x1,'Linewidth',2), xlabel('n'), grid
title('Discrete Exponent x1(n)'),
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. ДИСКРЕТНЫЙ КОМПЛЕКСНЫЙ ГАРМОНИЧЕСКИЙ СИГНАЛ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ вещественной и мнимой частей')
disp('% гармонического сигнала нажмите <ENTER>')
pause
x2 = C.*exp(j*w0.*n); % ДИСКРЕТНЫЙ КОМПЛЕКСНЫЙ ГАРМОНИЧЕСКИЙ СИГНАЛ
figure('Name','Discrete Harmonic Signal','NumberTitle','off')
subplot(2,1,1),stem(n,real(x2),'Linewidth',2), grid
title('Discrete Harmonic Signal: REAL [x2(n)]')
subplot(2,1,2),stem(n,imag(x2),'Linewidth',2), xlabel('n'), grid
title('Discrete Harmonic Signal: IMAG [x2(n)]')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ЗАДЕРЖАННЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ задержанных последовательностей нажмите <ENTER>')
pause
u0_m = [zeros(1,m) u0(1:(N-m))]; % ЗАДЕРЖАННЫЙ ЦИФРОВОЙ ЕДИНИЧНЫЙ ИМПУЛЬС
u1_m = [zeros(1,m) u1(1:(N-m))]; % ЗАДЕРЖАННЫЙ ЦИФРОВОЙ ЕДИНИЧНЫЙ СКАЧОК
x1_m = [zeros(1,m) x1(1:(N-m))]; % ЗАДЕРЖАННАЯ ДИСКРЕТНАЯ ЭКСПОНЕНТА
figure('Name','Delayed Discrete Signals','NumberTitle','off')
subplot(3,1,1),stem(n,u0_m,'Linewidth',2), grid
title('Delayed Digital Unit Impulse u0(n-m)')
subplot(3,1,2),stem(n,u1_m,'Linewidth',2), grid
title('Delayed Digital Unit Step u1(n-m)')
subplot(3,1,3),stem(n,x1_m,'Linewidth',2),xlabel('n'), grid
title('Delayed Discrete Exponent x1(n-m)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')

```

```

pause
disp('%')
disp('%')
disp('% п.6. ДИСКРЕТНЫЙ ПРЯМОУГОЛЬНЫЙ ИМПУЛЬС')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ дискретного прямоугольного импульса нажмите <ENTER>')
pause
x3_1 = U*rectpuls(n-n0,2*n_imp); x3_1(1:n0) = 0; % ФОРМИРОВАНИЕ ИМПУЛЬСА
С ПОМОЩЬЮ ФУНКЦИИ rectpuls
x3_2 = [zeros(1,n0) U.*u1((n0+1):(n0+n_imp))...
zeros(1,N-(n0+n_imp))]; % ФОРМИРОВАНИЕ ИМПУЛЬСА С ПОМОЩЬЮ ЦИФРОВОГО
ЕДИНИЧНОГО СКАЧКА
figure('Name','Discrete Rectangular and Triangular Impulses','NumberTitle',
'off')
subplot(3,1,1),stem(n,x3_1, 'Linewidth',2), grid
title('Discrete Rectangular Impulse x3 1(n)')
subplot(3,1,2),stem(n,x3_2, 'Linewidth',2), grid
title('Discrete Rectangular Impulse x3 2 (n)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. ДИСКРЕТНЫЙ ТРЕУГОЛЬНЫЙ ИМПУЛЬС')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА дискретного треугольного импульса нажмите <ENTER>')
pause
x4 = conv(x3_1,x3_1); % ДИСКРЕТНЫЙ ТРЕУГОЛЬНЫЙ ИМПУЛЬС
L = 2*N-1; % ДЛИНА СВЕРТКИ
n = 0:(L-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
subplot(3,1,3),stem(n,x4, 'Linewidth',2), xlabel('n'), grid
title('Discrete Triangular Impulse x4(n)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.8. ЛИНЕЙНАЯ КОМБИНАЦИЯ ДИСКРЕТНЫХ ГАРМОНИЧЕСКИХ СИГНАЛОВ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ гармонических сигналов и их линейной комбинации нажмите <ENTER>')

```

```

pause
n = 0:(5*N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
xi = repmat(B,length(n),1).*sin(n'*w); % МАТРИЦА ДИСКРЕТНЫХ ГАРМОНИК
ai = repmat(A,length(n),1); % МАТРИЦА КОЭФФИЦИЕНТОВ
x5 = sum((ai.* xi)'); % ЛИНЕЙНАЯ КОМБИНАЦИЯ ДИСКРЕТНЫХ ГАРМОНИК
figure('Name','Discrete Harmonic Signals and their Linear
Combination','NumberTitle','off')
subplot(4,1,1),stem(n, xi(:,1),'Linewidth',2), grid
title('First Discrete Harmonic Signal')
subplot(4,1,2),stem(n, xi(:,2),'Linewidth',2), grid
title('Second Discrete Harmonic Signal')
subplot(4,1,3),stem(n, xi(:,3),'Linewidth',2), grid
title('Third Discrete Harmonic Signal')
subplot(4,1,4),stem(n,x5,'Linewidth',2), xlabel('n'), grid
title('Linear Combination x5(n)')
disp('%')
disp('%')
disp('% Для вывода СРЕДНЕГО ЗНАЧЕНИЯ, ЭНЕРГИИ и СРЕДНЕЙ МОЩНОСТИ сигнала x5
нажмите <ENTER>')
pause
mean_x5 = mean(x5); % СРЕДНЕЕ ЗНАЧЕНИЕ СИГНАЛА
E = sum(x5.^2); % ЭНЕРГИЯ СИГНАЛА
P = sum(x5.^2)/length(x5); % СРЕДНЯЯ МОЩНОСТЬ СИГНАЛА
disp('%')
disp('%')
disp([' mean_x5 = ' num2str(mean_x5) ' E = ' num2str(E) ' P = ' num2str(P)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.9. ДИСКРЕТНЫЙ ГАРМОНИЧЕСКИЙ СИГНАЛ С ЭКСПОНЕНЦИАЛЬНОЙ ОГИБАЮЩЕЙ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА гармонического сигнала с экспоненциальной огибающей
нажмите <ENTER>')
pause
n = 0:(N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
x = C.*sin(w0.*n); % ДИСКРЕТНЫЙ ГАРМОНИЧЕСКИЙ СИГНАЛ
x6 = x.*(abs(a).^n); % ДИСКРЕТНЫЙ ГАРМОНИЧЕСКИЙ СИГНАЛ
С ЭКСПОНЕНЦИАЛЬНОЙ ОГИБАЮЩЕЙ
figure('Name','Harmonic Signal with Exponential Envelope. Periodic Sequence of
Rectangular Impulses','NumberTitle','off')
subplot(2,1,1),stem(n,x6,'Linewidth',2), grid
title('Harmonic Signal with Exponential Envelope x6(n)')
disp('%')

```

```

disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.10. ПЕРИОДИЧЕСКАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ДИСКРЕТНЫХ ПРЯМОУГОЛЬНЫХ  
ИМПУЛЬСОВ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА пяти периодов последовательности нажмите <ENTER>')
pause
xp = [U.*ul(1:n_imp) zeros(1,n_imp)];      % ПЕРИОД ПОСЛЕДОВАТЕЛЬНОСТИ
p = 5;                                       % ЧИСЛО ПЕРИОДОВ
x7 = repmat(xp,1,p);                       % ПЕРИОДИЧЕСКАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ
n = 0:(length(x7)-1);                      % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
subplot(2,1,2), stem(n,x7,'Linewidth',2), xlabel('n'), grid
title('Periodic Sequence of Rectangular Impulses x7(n) ')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.11. РАВНОМЕРНЫЙ БЕЛЫЙ ШУМ')
disp('%')
disp('%')
disp('% Для вывода ОЦЕНОК МАТЕМАТИЧЕСКОГО ОЖИДАНИЯ и ДИСПЕРСИИ ШУМА нажмите  
<ENTER>')
pause
r_uniform = rand(1,10000);                 % РАВНОМЕРНЫЙ БЕЛЫЙ ШУМ
mean_uniform = mean(r_uniform);             % ОЦЕНКА МАТ. ОЖИДАНИЯ ШУМА
var_uniform = var(r_uniform);               % ОЦЕНКА ДИСПЕРСИИ ШУМА
disp('%')
disp('%')
disp([' mean_uniform = ' num2str(mean_uniform) ' var_uniform = '
num2str(var_uniform)])
disp('%')
disp('%')
disp('% Для вывода графика АВТОКОВАРИАЦИОННОЙ ФУНКЦИИ нажмите <ENTER>')
pause
r_r_uniform = (1/length(r_uniform)).*xcov(r_uniform); % ОЦЕНКА
АВТОКОВАРИАЦИОННОЙ ФУНКЦИИ РАВНОМЕРНОГО БЕЛОГО ШУМА
m = -(length(r_uniform)-1):(length(r_uniform)-1); % ВЕКТОР ДИСКРЕТНЫХ
СДВИГОВ ДЛЯ АВТОКОВАРИАЦИОННОЙ ФУНКЦИИ
figure('Name','Autocovariance Function of Uniform White Noise','NumberTitle',
'off')
stem(m,r_r_uniform,'Linewidth',2), xlabel('m'), grid
title('Autocovariance Function of Uniform White Noise')

```

```

disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.12. НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ')
disp('%')
disp('%')
disp('% Для вывода ОЦЕНОК МАТЕМАТИЧЕСКОГО ОЖИДАНИЯ и ДИСПЕРСИИ шума нажмите <ENTER>')
pause
r_norm = randn(1,10000);           % НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ
mean_norm = mean(r_norm);         % ОЦЕНКА МАТ. ОЖИДАНИЯ ШУМА
var_norm = var(r_norm);           % ОЦЕНКА ДИСПЕРСИИ ШУМА
disp('%')
disp('%')
disp([' mean_norm = ' num2str(mean_norm) ' var_norm = ' num2str(var_norm)])
disp('%')
disp('%')
disp('% Для вывода графика АКФ нажмите <ENTER>')
pause
R_r_norm = (1/length(r_norm)).*xcorr(r_norm); % ОЦЕНКА АКФ НОРМАЛЬНОГО БЕЛОГО ШУМА
m = -(length(r_norm)-1):(length(r_norm)-1); % ВЕКТОР ДИСКРЕТНЫХ СДВИГОВ ДЛЯ АКФ
figure('Name','ACF of White Gaussian Noise','NumberTitle','off')
stem(m,R_r_norm,'Linewidth',2), xlabel('m'), grid
title('ACF of White Gaussian Noise')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.13. АДДИТИВНАЯ СМЕСЬ ДИСКРЕТНОГО ГАРМОНИЧЕСКОГО СИГНАЛА С НОРМАЛЬНЫМ БЕЛЫМ ШУМОМ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА аддитивной смеси сигнала с шумом нажмите <ENTER>')
pause
n = 0:(N-1);                     % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
x8 = x+randn(1,N);               % АДДИТИВНАЯ СМЕСЬ СИГНАЛА С ШУМОМ
figure('Name','Mixture of Harmonic Signal and White Gaussian Noise and ACF','NumberTitle','off')
subplot(2,1,1),stem(n,x8,'Linewidth',2),xlabel('n'), grid
title('Mixture of Harmonic Signal and White Gaussian Noise x8(n)')

```

```

disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.14. АКФ АДДИТИВНОЙ СМЕСИ ДИСКРЕТНОГО ГАРМОНИЧЕСКОГО СИГНАЛА
С НОРМАЛЬНЫМ БЕЛЫМ ШУМОМ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА АКФ нажмите <ENTER>')
pause
R = (1/N).*xcorr(x8); % ОЦЕНКА АКФ
m = -(N-1):(N-1); % ВЕКТОР ДИСКРЕТНЫХ СДВИГОВ ДЛЯ АКФ
subplot(2,1,2),stem(m),R,'Linewidth',2),xlabel('m'), grid
title('ACF R(m)')
disp('%')
disp('%')
disp('% Для вывода ДИСПЕРСИИ аддитивной смеси сигнала с шумом и АКФ R(N)
нажмите <ENTER>')
pause
disp('%')
disp('%')
disp([' var_x8 = ' num2str(var(x8))])
disp([' R(N) = ' num2str(R(N))])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.15. НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ С ЗАДАНЫМИ СТАТИСТИЧЕСКИМИ
ХАРАКТЕРИСТИКАМИ')
r_normMean = randn(1,10000)+Mean; % НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ С ЗАДАНЫМ
МАТЕМАТИЧЕСКИМ ОЖИДАНИЕМ
r_normVar = sqrt(Var).*randn(1,10000); % НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ С ЗАДАННОЙ
ДИСПЕРСИЕЙ
r_normMeanVar = sqrt(Var).*randn(1,10000)+ Mean; % НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ
С ЗАДАНЫМИ МАТЕМАТИЧЕСКИМ ОЖИДАНИЕМ И ДИСПЕРСИЕЙ
MAX = max([r_norm r_normMean r_normVar r_normMeanVar]);
% МАКСИМАЛЬНОЕ ЗНАЧЕНИЕ ШУМА СРЕДИ ЧЕТЫРЕХ ЕГО РАЗНОВИДНОСТЕЙ
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ нормального белого шума нажмите <ENTER>')
pause
figure('Name','White Gaussian Noises with different statistics','NumberTitle',
'off')
subplot(4,1,1), plot(r_norm), grid, ylim([-MAX MAX])

```

```

title(strcat([' Mean value = ',num2str(mean(r_norm)), '   Variance = ',num2str(var(r_norm))]))
subplot(4,1,2), plot(r_normMean), grid, ylim([-MAX MAX])
title(strcat([' Mean value = ',num2str(mean(r_normMean)), '   Variance = ',num2str(var(r_normMean))]))
subplot(4,1,3), plot(r_normVar), grid, ylim([-MAX MAX])
title(strcat([' Mean value = ',num2str(mean(r_normVar)), '   Variance = ',num2str(var(r_normVar))]))
subplot(4,1,4), plot(r_normMeanVar), xlabel('n'), grid, ylim([-MAX MAX])
title(strcat([' Mean value = ',num2str(mean(r_normMeanVar)), '   Variance = ',num2str(var(r_normMeanVar))]))
disp('%')
disp('%')
disp('% Для вывода ГИСТОГРАММ нормального белого шума нажмите <ENTER>')
pause
figure('Name','Histograms with different statistics','NumberTitle','off')
subplot(4,1,1), hist(r_norm), grid, xlim([-MAX MAX])
title(strcat([' Mean value = ',num2str(mean(r_norm)), '   Variance = ',num2str(var(r_norm))]))
subplot(4,1,2), hist(r_normMean), grid, xlim([-MAX MAX])
title(strcat([' Mean value = ',num2str(mean(r_normMean)), '   Variance = ',num2str(var(r_normMean))]))
subplot(4,1,3), hist(r_normVar), grid, xlim([-MAX MAX])
title(strcat([' Mean value = ',num2str(mean(r_normVar)), '   Variance = ',num2str(var(r_normVar))]))
subplot(4,1,4), hist(r_normMeanVar), grid, xlim([-MAX MAX])
title(strcat([' Mean value = ',num2str(mean(r_normMeanVar)), '   Variance = ',num2str(var(r_normMeanVar))]))
disp('%')
disp('%')
disp('% РАБОТА ЗАВЕРШЕНА')

```

7.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файлов для моделирования последовательностей с использованием исходных данных из табл. 7.1 для своего номера бригады $N_{бр}$.

Моделируемые последовательности выбираются из следующего списка:

1С. Линейная комбинация дискретных гармонических сигналов:

$$x(n) = a_1 x_1(n) + a_2 x_2(n) + a_3 x_3(n),$$

где

$$x_i(n) = \text{Re}\{B_i e^{j\hat{\omega}_i n}\}, i = 1, 2, 3,$$

с выводом графиков последовательностей $x_i(n)$ и $x(n)$ на интервале времени $n \in [0; (3N - 1)]$.

2С. Дискретный прямоугольный импульс с амплитудой U , длительностью $2n_{imp}$ и моментом начала $2n_0$ с выводом графика на интервале времени (7.12).

Определить энергию и мощность импульса.

3С. Периодическая последовательность дискретных прямоугольных импульсов с амплитудой U , длительностью n_{imp} и периодом, вдвое большим длительности импульса, с выводом графика для заданного числа периодов.

4С. Оценка автоковариационной функции $\hat{r}_x(m)$ аддитивной смеси дискретного гармонического сигнала $x(n)$ (7.21) с нормальным белым шумом с параметрами, заданными по умолчанию, с выводом графика оценки автоковариационной функции, центрированной относительно $m = 0$.

5С. Аддитивная смесь дискретного гармонического сигнала $x(n)$ (7.21) с нормальным белым шумом с математическим ожиданием $mean$ и дисперсией var с выводом графика на интервале времени (7.12).

6С. Оценка АКФ нормального белого шума с математическим ожиданием $mean$ и дисперсией var с выводом графика оценки АКФ, центрированной относительно $m = 0$.

7С. Дискретный гармонический сигнал с изменением мгновенной частоты (ЧМ-сигнал):

$$x(t)|_{t=nT} = \cos(2\pi f(t)t). \quad (7.22)$$

Вычислить с помощью функции:

`x = chirp(t,f0,t1,f1,method)`

где t , x — векторы значений дискретного времени nT (с) и последовательности $x(nT)$ (7.22); f_0 — начальная частота f_0 (Гц); t_1 , f_1 — момент дискретного времени t_1 (с) и значение частоты $f_1(t_1)$ (Гц); `method` — закон изменения мгновенной частоты $f(t)$:

- 'linear' — линейный:

$$f(t) = f_0 + (f_1 - f_0)(t/t_1);$$

- 'quadratic' — квадратичный:

$$f(t) = f_0 + (f_1 - f_0)(t/t_1)^2;$$

- 'logarithmic' — логарифмический (в действительности экспоненциальный):

$$f(t) = f_0(f_1/f_0)^{t/t_1}.$$

Вывести графики последовательности $x(nT)$ (7.22) с помощью функции `plot` на интервале дискретного времени $t = nT \in [0; 50(N-1)T]$ с шагом T при $f_0 = 10$, $t_1 = 50(N-1)T$ и $f_1 = 50$ и различных значениях параметра `method`.

8С. Последовательность с однотоновой амплитудной модуляцией (АМ-сигнал):

$$x(n) = C[1 + m\cos(\Omega n + \varphi_\Omega)]\cos(\hat{\omega}_0 n + \varphi_0), \quad (7.23)$$

где C , $\hat{\omega}_0$ и φ_0 — соответственно амплитуда, частота и начальная фаза *несущего* колебания; Ω и φ_Ω — частота и начальная фаза *модулирующего* колебания; m — коэффициент модуляции (глубина модуляции), $m \in [0, 1]$.

Вывести графики последовательности $x(n)$ (7.23) с помощью функции `plot` на интервале $n \in [0; (20N - 1)]$ при следующих значениях параметров АМ-сигнала:

- $\varphi_0 = \pi/3$, $\Omega = \hat{\omega}_0/4$, $\varphi_\Omega = \pi/6$, $m = 0,5$;
- $\varphi_0 = 0$, $\Omega = \hat{\omega}_0/4$, $\varphi_\Omega = 0$ при $m = 0,5$, $m = 0$ и $m = 1$.

9С. Последовательность в виде Гауссова радиоимпульса:

$$x(n) = e^{-an^2} \cos(\hat{\omega}_1 n), \quad (7.24)$$

где a — параметр, управляющий длительностью радиоимпульса, $\hat{\omega}_1$ — несущая частота.

Вывести графики последовательности $x(n)$ (7.24) с помощью функции `plot` при следующих значениях параметров Гауссова радиоимпульса:

- $\hat{\omega}_1 = \hat{\omega}_0/2$;
- $a = 0,0005$; $0,001$; $0,005$,

на интервале $n \in [-3(N - 1); 3(N - 1)]$ и на интервале $n \in [0; 6(N - 1)]$ (со сдвигом в область положительного времени).

10С. Последовательность

$$x(t) \Big|_{t=nT} = \frac{\sin \pi t}{\pi t}. \quad (7.25)$$

Вычислить с помощью функции:

`x=sinc(t)`

где t , x — векторы значений дискретного времени nT (с) и последовательности $x(nT)$ (7.25).

Вывести графики последовательности $x(nT)$ (7.25) на интервале $t = nT \in [-500(N - 1)T; 500(N - 1)T]$ с шагом T и на интервале $t = nT \in [0; 1000(N - 1)T]$ (со сдвигом в область положительного времени).

7.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения каждого пункта задания, включая копируемые из окна **Command Window** результаты вычислений (шрифт Courier New), созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**) и ответы на поставленные вопросы (шрифт Times New Roman).

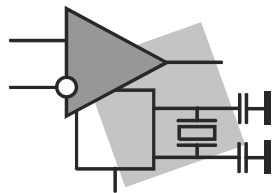
Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. Дайте определение дискретного и цифрового сигналов.
2. Как математически описывается дискретный сигнал?
3. Какой тип данных используется по умолчанию при описании последовательностей в MATLAB?
4. Что такое период и частота дискретизации и как они связаны друг с другом?
5. Дайте определение дискретного нормированного времени.
6. Дайте определение нормированной частоты $\hat{\omega}$.
7. Какие дискретные сигналы называют детерминированными?
8. Назовите основные характеристики детерминированных дискретных сигналов.
9. Поясните, с какой целью и как вычисляются автокорреляционная и автоковариационная функции.
10. Какими свойствами обладает АКФ?
11. Какие дискретные сигналы называют случайными?
12. Что такое ансамбль реализаций случайного дискретного сигнала?
13. Назовите основные статистические характеристики случайных дискретных сигналов.
14. Как определяются основные статистические характеристики случайных дискретных сигналов по ансамблю реализаций?
15. Какие случайные дискретные сигналы называют стационарными в широком смысле?
16. Какие случайные дискретные сигналы называют эргодическими?
17. Дайте определение равномерного белого шума и нормального белого шума.
18. Какой вид имеют АКФ нормального белого шума и автоковариационная функция равномерного белого шума?

7.7. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 8.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010 — Глава 3.
3. Оппенгейм А., Шафер Р. Цифровая обработка сигналов — М.: Техносфера, 2007. — Приложение А.

ГЛАВА 8



Линейные дискретные системы

Цель работы: изучить математическое описание линейных дискретных систем и овладеть программными средствами их моделирования и анализа в MATLAB.

8.1. Краткая теоретическая справка

Системой обработки сигналов (системой) называется объект, выполняющий требуемое преобразование входного сигнала (воздействия) в выходной (реакцию).

В соответствии с определением, системой можно называть как физическое устройство, так и математическое преобразование. По умолчанию под системой будем понимать *математическое преобразование*.

Математической моделью системы называют ее *соотношение вход/выход*, которое устанавливает связь между входным и выходным сигналами.

Систему называют *линейной*, если она отвечает условиям *аддитивности* (реакция на сумму воздействий равна сумме реакций на данные воздействия¹) и *однородности* (воздействию, умноженному на весовой коэффициент, соответствует реакция, умноженная на тот коэффициент).

Систему называют *дискретной*, если она преобразует дискретное воздействие $x(n)$ в дискретную реакцию $y(n)$.

Систему называют *стационарной*, если ее реакция инвариантна по отношению к началу отсчета времени (свойство *инвариантности во времени*). Параметры стационарной системы *неизменны во времени*; задержке воздействия соответствует такая же задержка реакции.

По умолчанию будем рассматривать *стационарные* линейные дискретные системы (ЛДС).

Нулевые начальные условия (ННУ) означают, что все значения воздействия и реакции, которые может помнить ЛДС в моменты времени, предшествующие началу воздействия² $n = 0$, равны нулю:

¹ Принцип суперпозиции.

² Здесь и далее используется дискретное нормированное время $n = nT/T$.

$$\begin{cases} x(n-i) \Big|_{(n-i) < 0, i=1, 2, \dots} = 0; \\ y(n-k) \Big|_{(n-k) < 0, k=1, 2, \dots} = 0, \end{cases}$$

т. е. воздействие и реакция в области отрицательного времени $n < 0$ равны нулю.

В ЛДС с одним входом и одним выходом соотношение вход/выход представляет собой *линейное* математическое преобразование, вид которого однозначно связан с *основной характеристикой* ЛДС во временной области или в z -области.

Под *моделированием* ЛДС понимают вычисление ее реакции в соответствии с соотношением вход/выход, а под *анализом* ЛДС — анализ ее характеристик во временной, z - и частотной областях.

8.1.1. Описание ЛДС во временной области

Основной характеристикой ЛДС во временной области является импульсная характеристика (ИХ).

Импульсной характеристикой $h(n)$ ЛДС называют ее реакцию на цифровой единичный импульс $u_0(n)$ (7.10) при ННУ.

Соотношение вход/выход ЛДС, однозначно связанное с его основной характеристикой во временной области — ИХ, имеет вид линейного математического преобразования в виде *формулы свертки*:

$$y(n) = \sum_{m=0}^{\infty} h(n-m)x(m) = \sum_{m=0}^{\infty} h(m)x(n-m), \quad (8.1)$$

где m — задержка последовательности.

Соотношение вход/выход ЛДС, однозначно связанное с его основной характеристикой в z -области — передаточной функцией (о которой пойдет речь в разд. 8.1.2), имеет вид линейного математического преобразования в виде *разностного уравнения* (РУ):

$$y(n) = \sum_{i=0}^{N-1} b_i x(n-i) - \sum_{k=1}^{M-1} a_k y(n-k), \quad (8.2)$$

где b_i , a_k — вещественные коэффициенты РУ — *параметры ЛДС*; i , k — значения задержек воздействия и реакции; $(N-1)$, $(M-1)$ — константы, определяющие максимальные задержки.

В отличие от линейных *аналоговых* систем, где соответствующие соотношения вход/выход имеют вид интеграла свертки или линейного дифференциального уравнения, вычисление реакции по формуле свертки (8.1) или РУ (8.2) выполняется *методом прямой подстановки* при ННУ, т. е. эти соотношения непосредственно описывают *алгоритмы* вычисления реакции.

По виду РУ различают два типа ЛДС:

- *рекурсивная* ЛДС, реакция которой зависит от текущего и предшествующих отсчетов воздействия и предшествующих отсчетов реакции, т. е.:

$$a_k \neq 0 \text{ хотя бы для одного значения } k ;$$

- *нерекурсивная* ЛДС, реакция которой зависит только от текущего и предшествующих отсчетов воздействия, т. е.:

$$a_k = 0 \text{ для всех } k .$$

Рекурсивные и нерекурсивные ЛДС имеют соответственно *бесконечную* и *конечную* ИХ, отсюда их тождественные названия:

- БИХ ЛДС (IIR — Infinite Impulse Response);
- КИХ ЛДС (FIR — Finite Impulse Response).

Импульсная характеристика КИХ ЛДС совпадает с коэффициентами b_i РУ (8.2):

$$h(n) = b_i, \quad n = i. \quad (8.3)$$

В MATLAB вычисление реакции по формуле свертки (8.1) выполняется с помощью функции:

y = conv(h, x)

где h — вектор отсчетов ИХ длины N_1 ; бесконечная ИХ рекурсивной ЛДС ограничивается до конечной длины; x — вектор отсчетов воздействия длины N_2 ; y — вектор отсчетов реакции длины $L = N_1 + N_2 - 1$ (длина свертки).

Вычисление реакции по РУ (8.2) выполняется с помощью функции:

y = filter(b, a, x)

где b , a — векторы коэффициентов $[b_0, b_1, \dots, b_{N-1}]$ и $[1, a_1, \dots, a_{M-1}]$; x — вектор отсчетов воздействия; y — вектор отсчетов реакции с длиной, равной длине воздействия.

Обратим внимание на то, что коэффициенты a_k записываются без учета знака "минус", стоящего перед суммой в РУ (8.2).

Импульсная характеристика вычисляется с помощью функции:

h = impz(b, a, N)

где b , a — определены ранее для функции `filter`; N — количество отсчетов (длина) ИХ; h — вектор отсчетов ИХ.

Импульсная характеристика может также вычисляться с помощью функции `filter`, если в качестве воздействия используется цифровой единичный импульс (7.10).

8.1.2. Описание ЛДС в z -области

Основной характеристикой ЛДС в z -области является передаточная функция $H(z)$ — z -изображение ИХ $h(n)$:

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n}.$$

Передаточной функцией ЛДС называют отношение z -изображения реакции к z -изображению воздействия при ННУ:

$$H(z) = Y(z)/X(z).$$

Данное отношение легко получить, выполнив Z -преобразование РУ (8.2). Передаточная функция рекурсивной ЛДС имеет вид дробно-рациональной функции:

$$H(z) = \frac{\sum_{i=0}^{N-1} b_i z^{-i}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}}. \quad (8.4)$$

Показатель степени z^{-i} соответствует задержкам воздействия, а z^{-k} — задержкам реакции; коэффициенты a_k передаточной функции и РУ (8.2) имеют противоположные знаки.

Для *нерекурсивных* ЛДС передаточная функция, с учетом (8.3), принимает вид:

$$H(z) = \sum_{i=0}^{N-1} b_i z^{-i} = \sum_{n=0}^{N-1} h(n) z^{-n}. \quad (8.5)$$

Порядок рекурсивной ЛДС равен порядку *знаменателя* передаточной функции (8.4) $(M-1)$ при соблюдении условия $(N-1) \leq (M-1)$ (по умолчанию).

Порядок нерекурсивной ЛДС равен $(N-1)$.

Нулями передаточной функции называют значения z , при которых она равна нулю, а *полюсами* (особыми точками) — значения z , при которых ее знаменатель равен нулю.

Картой нулей и полюсов называют z -плоскость с нанесенной единичной окружностью и символически изображенными нулями и полюсами.

По карте нулей и полюсов можно сделать вывод об устойчивости ЛДС: полюсы устойчивой ЛДС располагаются внутри единичного круга [1].

Комплексно сопряженные нули $z_{\circ k1,2}$ и полюсы $z_{*k1,2}$ представляют в показательной форме, где аргументы — углы (в радианах) на комплексной z -плоскости:

$$\begin{cases} z_{\circ k1,2} = r_{\circ k} e^{\pm j\varphi_{\circ k}}; \\ z_{*k1,2} = r_{*k} e^{\pm j\varphi_{*k}}. \end{cases} \quad (8.6)$$

Помимо общего вида (8.4), передаточная функция *рекурсивных* ЛДС может быть представлена следующими своими разновидностями¹:

□ произведение простейших множителей:

$$H(z) = b_0 \prod_{k=1}^{M-1} \frac{(1 - z_{0k} z^{-1})}{(1 - z_{*k} z^{-1})}, \quad (8.7)$$

где z_{0k} , z_{*k} — соответственно k -е нуль и полюс передаточной функции (8.4).

В общем случае нули и полюсы — *попарно комплексно сопряженные* числа;

□ произведение множителей второго порядка:

$$H(z) = \prod_{k=1}^L \frac{b_{0k} + \tilde{b}_{1k} z^{-1} + \tilde{b}_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}}, \quad (8.8)$$

где b_{0k} , $\tilde{b}_{1k}$, $\tilde{b}_{2k}$, a_{1k} , a_{2k} — *вещественные* коэффициенты рекурсивных звеньев 2-го порядка, называемых также *биквадратными*; L — количество звеньев, равное

$$L = \text{int}\left(\frac{M-1}{2}\right),$$

где int — функция округления до ближайшего целого в сторону увеличения.

В MATLAB используется представление передаточной функции (8.8) в эквивалентном виде, получаемом при вынесении за скобки коэффициентов b_{0k} :

$$H(z) = G \prod_{k=1}^L \frac{1 + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}}, \quad (8.9)$$

где $G = b_{01} \cdot b_{02} \cdot \dots \cdot b_{0L}$ — коэффициент усиления, а соответствующие коэффициенты связаны соотношениями: $b_{1k} = \tilde{b}_{1k} / G$; $b_{2k} = \tilde{b}_{2k} / G$;

□ сумма простых дробей:

$$H(z) = \sum_{k=1}^{M-1} H_k(z) = \sum_{k=1}^{M-1} \frac{A_k}{1 - z_{*k} z^{-1}}, \quad (8.10)$$

где z_{*k} — простой (не кратный) k -й полюс передаточной функции (8.4); A_k — коэффициент разложения при k -м полюсе; A_k и z_{*k} — всегда числа одинакового типа, комплексные или вещественные.

При одинаковых порядках числителя и знаменателя $(M-1) = (N-1)$ в (8.4) будем иметь в (8.10) *целую часть* — вещественную константу C :

¹ Приведены разновидности, представленные в MATLAB стандартными функциями.

$$H(z) = \sum_{k=1}^{M-1} \frac{A_k}{1 - z_{*k} z^{-1}} + C. \quad (8.11)$$

В MATLAB для представления передаточной функции (8.4) в виде произведения простейших множителей (8.7) используется функция:

$$[q, p, K] = \text{tf2zpk}(b, a)$$

где b, a — векторы коэффициентов числителя $[b_0, b_1, \dots, b_{N-1}]$ и знаменателя $[1, a_1, \dots, a_{M-1}]$ передаточной функции (8.4); q, p — векторы-столбцы нулей z_{*k} и полюсов z_{*k} передаточной функции (8.7), представленные в алгебраической форме; K — коэффициент усиления b_0 в (8.7).

Представление передаточной функции (8.4) в виде произведения множителей второго порядка (8.9) выполняется с помощью функции:

$$[s, G] = \text{tf2sos}(b, a)$$

где b, a — определены ранее для функции tf2zpk ; G — коэффициент усиления G в (8.9); s — матрица коэффициентов числителей и знаменателей биквадратных звеньев передаточной функции (8.9) в виде:

$$\begin{bmatrix} 1 & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ 1 & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & b_{1L} & b_{2L} & 1 & a_{1L} & a_{2L} \end{bmatrix}. \quad (8.12)$$

Для представления передаточной функции (8.4) в виде суммы простых дробей (8.11) применяется функция:

$$[r, p, c] = \text{residuez}(b, a)$$

где b, a — определены ранее для функции tf2zpk ; r, p — векторы-столбцы коэффициентов разложения A_k и полюсов z_{*k} в (8.11), представленные в алгебраической форме; c — целая часть C в (8.11); при ее отсутствии выводится пустая матрица $c = []$.

Карта нулей и полюсов передаточной функции выводится с помощью функции:

$$\text{zplane}(b, a)$$

8.1.3. Описание ЛДС в частотной области

Основной характеристикой ЛДС в частотной области является частотная характеристика (ЧХ) $H(e^{j\hat{\omega}})$ — Фурье-изображение ИХ $h(n)$:

$$H(e^{j\hat{\omega}}) = \sum_{n=0}^{\infty} h(n) e^{-j\hat{\omega}n},$$

где $\hat{\omega}$ — нормированная частота:

$$\hat{\omega} = \omega T \text{ (рад)}. \quad (8.13)$$

Частотная характеристика $H(e^{j\hat{\omega}})$ связана с передаточной функцией $H(z)$ соотношением:

$$H(e^{j\hat{\omega}}) = H(z) \Big|_{z=e^{j\hat{\omega}}}, \quad (8.14)$$

что позволяет путем подстановки $z = e^{j\hat{\omega}}$ в (8.4) получить ее в виде:

$$H(e^{j\hat{\omega}}) = \frac{\sum_{i=0}^{N-1} b_i e^{-ji\hat{\omega}}}{1 + \sum_{k=1}^{M-1} a_k e^{-jk\hat{\omega}}}, \quad (8.15)$$

Частотную характеристику $H(e^{j\hat{\omega}})$ (8.15) можно представить в показательной форме:

$$H(e^{j\hat{\omega}}) = \left| H(e^{j\hat{\omega}}) \right| e^{j \arg \{ H(e^{j\hat{\omega}}) \}} = A(\hat{\omega}) e^{j\varphi(\hat{\omega})}. \quad (8.16)$$

Модуль $A(\hat{\omega})$ и аргумент $\varphi(\hat{\omega})$ частотной характеристики соответствуют амплитудно-частотной и фазочастотной характеристикам ЛДС.

Амплитудно-частотная характеристика (АЧХ) отображает частотную зависимость отношения амплитуды реакции к амплитуде гармонического воздействия в установившемся режиме.

Фазочастотная характеристика (ФЧХ) отображает частотную зависимость разности фаз реакции и гармонического воздействия в установившемся режиме.

АЧХ и ФЧХ — *периодические* функции с периодом 2π в шкале частот $\hat{\omega}$ или f_d в шкале частот f (Гц).

АЧХ — *четная*, а ФЧХ — *нечетная* функция частоты.

АЧХ и ФЧХ рассчитываются в *основной полосе частот* $[0; \pi]$ в шкале частот $\hat{\omega}$ или $[0; f_d/2]$ в шкале частот f (Гц).

По карте нулей и полюсов можно определить местоположение максимумов, минимумов и нулей АЧХ в основной полосе частот $[0; \pi]$, а именно:

- частота комплексно сопряженного *полюса* $\hat{\omega}_{*k}$, где $\hat{\omega}_{*k} = \varphi_{*k}$ в (8.6), соответствует частоте *максимума* АЧХ (приблизительно);
- частота комплексно сопряженного *нуля* $\hat{\omega}_{\circ k}$, где $\hat{\omega}_{\circ k} = \varphi_{\circ k}$ в (8.6), соответствует частоте *минимума* АЧХ (приблизительно), если $r_{\circ k} \neq 1$, или *нуля* АЧХ, если $r_{\circ k} = 1$ (комплексно сопряженные нули на единичной окружности); в точке нуля АЧХ наблюдается скачок ФЧХ на π ;

□ вещественным нулям $z_{ok}=1$ и/или $z_{ok}=-1$ (на единичной окружности) соответствует нуль АЧХ на границе основной полосы частот 0 и/или π .

В MATLAB частотная характеристика (8.15) вычисляется с помощью функции `freqz` одного из следующих форматов:

`H = freqz(b, a, f, Fs)`

`H = freqz(b, a, w)`

`H = freqz(b, a, N)`

где: b , a — определены ранее для функции `tf2zpk` (см. разд. 8.1.2); f — вектор частот в герцах; F_s — частота дискретизации f_d (Гц); w — вектор нормированных частот $\hat{\omega}$ (рад); N — количество точек ЧХ; в отсутствии параметра по умолчанию $N = 512$; H — вектор комплексных значений ЧХ.

Модуль частотной характеристики (АЧХ) определяется с помощью функции `abs(H)`, а аргумент (ФЧХ) — с помощью функции `angle(H)` (см. табл. 1.4).

8.1.4. Структуры звеньев 2-го порядка

Структура (структурная схема) ЛДС отображает алгоритм вычисления реакции по РУ и определяется *видом передаточной функции*.

Для рекурсивных звеньев 2-го порядка с передаточной функцией

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (8.17)$$

и разностным уравнением

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$$

поддерживаются следующие структуры:

□ прямая¹ — Direct-Form I (рис. 8.1, а);

□ прямая транспонированная — Direct-Form I Transposed (рис. 8.1, б);

□ прямая каноническая — Direct-Form II (рис. 8.1, в);

□ прямая каноническая транспонированная — Direct-Form II Transposed (рис. 8.1, г).

В MATLAB структуры описываются в виде объекта `dfilt` (от англ. *Discrete-time filter object*):

`Hd = dfilt.structure(b, a)`

где `Hd` — имя объекта; `dfilt` — тип объекта; `structure` — функция, задающая конкретную структуру объекта `Hd` (табл. 8.1); b , a — параметры функции `structure` — векторы коэффициентов передаточной функции (8.4), определенные ранее для функции `tf2zpk` (см. разд. 8.1.2).

¹ Прямой структуре соответствует представление передаточной функции в виде (8.17), а остальным прямым структурам — ее модификации и модификации РУ.

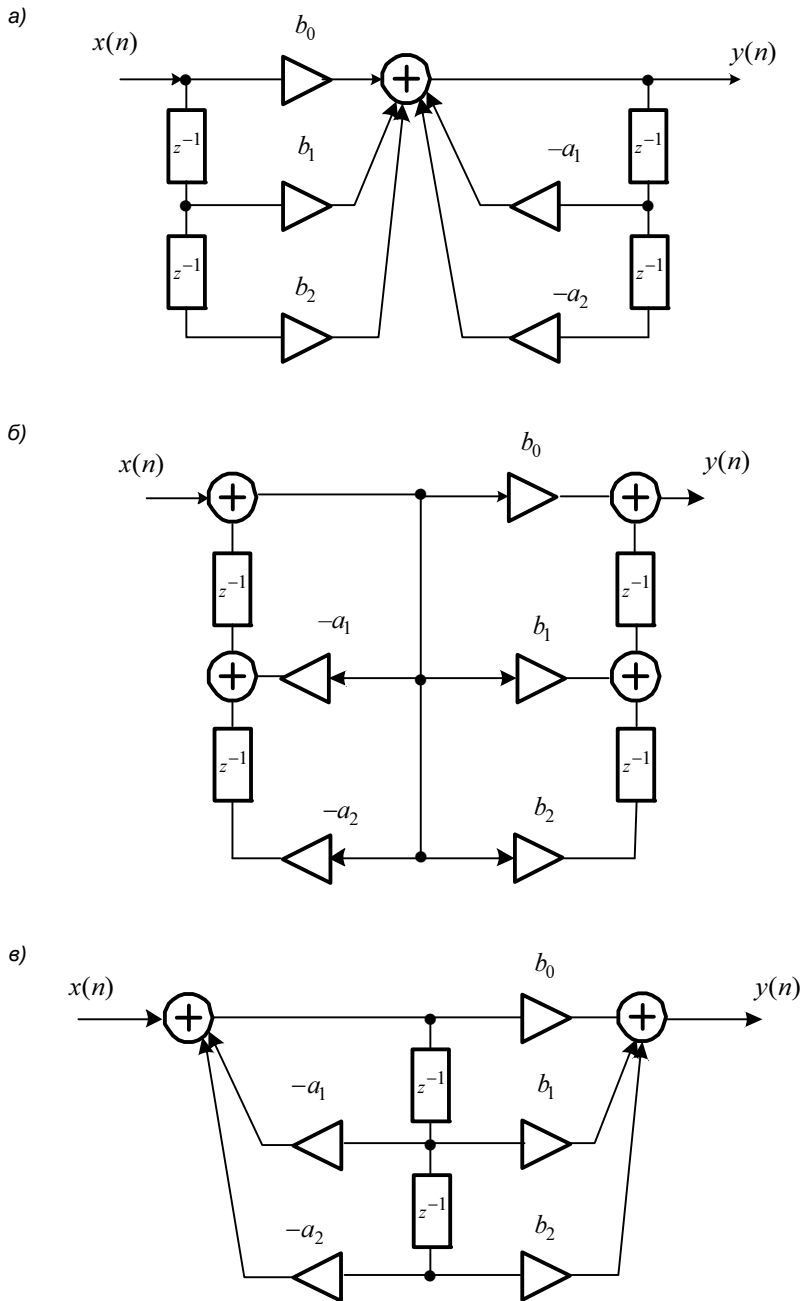


Рис. 8.1. Структурные схемы рекурсивного звена 2-го порядка:
 прямая (Direct-Form I) (а); прямая транспонированная (Direct-Form I Transposed) (б);
 прямая каноническая (Direct-Form II) (в)

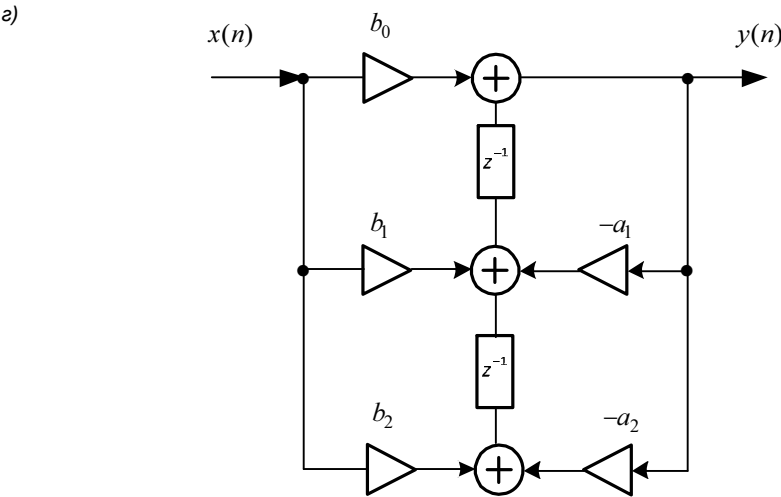


Рис. 8.1. Структурные схемы рекурсивного звена 2-го порядка:
прямая каноническая транспонированная (Direct-Form II Transposed) (з)

Свойства объекта `dfilt` с именем `hd` для рекурсивных звеньев 2-го порядка включают в себя:

- ☐ `FilterStructure` — структура звена;
- ☐ `Arithmetic` — форма представления данных;
- ☐ `Numerator` — коэффициенты числителя передаточной функции;
- ☐ `Denominator` — коэффициенты знаменателя передаточной функции;
- ☐ `PersistentMemory` — начальные условия при вычислении реакции; значение `false` соответствует ННУ (см. разд. 8.1.1).

Структуры звеньев 2-го порядка, описываемые в виде объектов `dfilt`, приведены в табл. 8.1.

Таблица 8.1. Функции `structure` и структуры рекурсивных звеньев 2-го порядка

Функция <i>structure</i>	Структура рекурсивного звена 2-го порядка
<code>df1</code>	Direct-Form I (прямая, см. рис. 8.1, а)
<code>df1t</code>	Direct-Form I Transposed (прямая транспонированная, см. рис. 8.1, б)
<code>df2</code>	Direct-Form II (прямая каноническая, см. рис. 8.1, в)
<code>df2t</code>	Direct-Form II Transposed (прямая каноническая транспонированная, см. рис. 8.1, з)

8.2. Содержание лабораторной работы

Содержание работы связано с моделированием ЛДС, анализом ее характеристик и описанием структур программными средствами MATLAB на примере рекурсивных звеньев 2-го порядка.

8.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файла `lr_08` и function-файла `input_1`, которые хранятся на прилагаемом компакт-диске в папке `LAB_DSP\LAB_08`.

Перед выполнением работы необходимо сохранить путь к папке `LAB_08` по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 8.2 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$. Функция $N_{бр} \bmod M$ в записи исходных данных означает вычисление значения $N_{бр}$ по модулю M .

Коэффициенты передаточной функции (8.17) b_0, b_1, b_2 и a_1, a_2 (см. табл. 8.2) рассчитываются с точностью до четырех значащих цифр¹.

Таблица 8.2. Таблица исходных данных

Переменная	Назначение	Значение	Идентификатор
$N_{бр}$	Номер бригады	$N_{бр}$	Nb =
b_0 b_1 b_2	Коэффициенты числителя передаточной функции	$b_0 = 0,5 + 0,02N_{бр}$ $b_1 = b_0(-1)^{N_{бр}+1}(0,9822 + 0,0178N_{бр})$ $b_2 = b_0[0,8 + 0,2(N_{бр} \bmod 5)]$	Вектор b = [...]
a_0 a_1 a_2	Коэффициенты знаменателя передаточной функции	$a_0 = 1$ $a_1 = (-1)^{N_{бр}}(0,7778 + 0,025N_{бр})$ $a_2 = 0,64 + 0,006N_{бр}$	Вектор a = [1...]
N_1	Длина ИХ	$N_1 = N_{бр} \bmod 10 + 20$	N1 =
N_2	Длина воздействия	$N_2 = N_{бр} \bmod 10 + 30$	N2 =
f_d	Частота дискретизации	$f_d = 1000N_{бр}$	Fs =

¹ Это удобно сделать в MATLAB в режиме прямых вычислений.

На прилагаемом компакт-диске в папке Tables\Tables_08 хранятся табл. 8.2 исходных данных и пример ее заполнения для $N_{\text{ср}} = 1$.

Задание на лабораторную работу связано с моделированием рекурсивного звена 2-го порядка и анализом его характеристик и включает в себя следующие пункты:

1. Вычисление импульсной характеристики (идентификатор h1) длины N_1 с помощью функции `impz` с выводом графика.

Записать аналитическую формулу ИХ рекурсивного звена 2-го порядка с учетом ННУ. Пояснить, чему в действительности равна длина ИХ.

2. Вычисление импульсной характеристики (идентификатор h2) с помощью функции `filter` с выводом графика.

Пояснить, что и почему выбрано в качестве воздействия.

3. Вычисление реакции $y_1(n)$ (идентификатор y1) по формуле свертки.

В качестве воздействия $x(n)$ длины N_2 выбрать дискретный прямоугольный импульс (идентификатор x):

$$x(n) = \begin{cases} 1, & 0 \leq n < \text{int}(N_2/2); \\ 0, & \text{int}(N_2/2) \leq n \leq (N_2 - 1). \end{cases} \quad (8.18)$$

Функция `int` определена в разд. 8.1.2.

Для моделирования воздействия (8.18) использовать function-файл `input_1` (см. разд. 8.4.1).

Вывести график воздействия $x(n)$ и два графика реакции $y_1(n)$ с длиной, равной длине свертки L , и длиной, ограниченной до длины воздействия.

Записать формулу свертки.

Пояснить:

- чему равна длина импульса (8.18);
- чему равна длина свертки аналитически и по графику;
- почему ее ограничивают до длины воздействия.

4. Вычисление реакции $y_2(n)$ (идентификатор y2) по разностному уравнению.

Задать воздействие $x(n)$ (8.18). Вывести графики воздействия и реакции.

Сравнить графики реакций $y_1(n)$ (см. п. 3) и $y_2(n)$.

Записать РУ рекурсивного звена 2-го порядка с заданными коэффициентами.

Пояснить, чему равны длины воздействия и реакции.

5. Вычисление параметров передаточной функции в виде произведения простейших множителей.

Вычислить нули, полюсы и коэффициент усиления (идентификаторы q, p и k) передаточной функции (8.17).

Записать нули и полюсы в алгебраической и показательной формах и пояснить связь между ними.

Выразить значение аргумента полюса и нуля относительно π , например, значению $\varphi = 1,7654$ будет соответствовать:

$$\varphi = 1,7654 \approx 0,562\pi. \quad (8.19)$$

Представить передаточную функцию в виде произведения простейших множителей с нулями и полюсами в показательной форме.

6. Вычисление параметров передаточной функции в виде произведения множителей второго порядка.

Вычислить коэффициент усиления (идентификатор g) и матрицу коэффициентов (идентификатор s) передаточной функции.

Представить передаточную функцию в виде произведения множителей второго порядка.

7. Вычисление параметров передаточной функции в виде суммы простых дробей.

Вычислить полюсы, коэффициенты разложения и целую часть (идентификаторы p , r и c) передаточной функции.

Записать полюсы и коэффициенты разложения в алгебраической и показательной формах.

Выразить значения аргумента полюса и коэффициента разложения в виде (8.19).

Представить передаточную функцию в виде суммы простых дробей с полюсами и коэффициентами разложения в показательной форме.

8. Вывод карты нулей и полюсов.

Изобразить карту нулей и полюсов.

Пояснить:

- является ли рекурсивное звено устойчивым;
- совпадают ли значения нулей и полюсов с вычисленными в п. 5.

9. Вычисление АЧХ и ФЧХ в шкале нормированных частот.

Вычислить АЧХ и ФЧХ (идентификаторы MAG_w и $PHASE\_w$) в шкале нормированных частот $\hat{\omega}$ (идентификатор w) и вывести их графики.

Сравнить значения полученной АЧХ на границах основной полосы со значениями, вычисленными аналитически по формулам:

$$A(0) = |H(z)| \Big|_{z=e^{j0}=1} = \left| \frac{b_0 + b_1 + b_2}{1 + a_1 + a_2} \right|; \quad (8.20)$$

$$A(\pi) = |H(z)| \Big|_{z=e^{j\pi}=-1} = \left| \frac{b_0 - b_1 + b_2}{1 - a_1 + a_2} \right|. \quad (8.21)$$

Пояснить:

- чему равны границы основной полосы частот;
- соответствие между картой нулей и полюсов и видом АЧХ;
- какому значению АЧХ соответствует скачок на π , если он имеется;
- какие частотные составляющие воздействия, низкие или высокие, оказались преимущественно подавленными в реакции.

10. Вычисление АЧХ и ФЧХ в шкале абсолютных частот.

Вычислить АЧХ и ФЧХ (идентификаторы `MAG` и `PHASE`) в шкале частот f (Гц) (идентификатор `f`) при заданной частоте дискретизации f_d и вывести их графики.

Пояснить:

- чему равны границы основной полосы частот;
- соответствие частотами $\hat{\omega}$ и f .

11. Описание структуры рекурсивного звена.

Описать четыре разновидности структур рекурсивного звена 2-го порядка (см. табл. 8.2) в виде объектов `dfilt` с именами `Hd1`—`Hd4`.

Пояснить:

- что отображает структура и чем определяется ее вид;
- свойства каждого из объектов `dfilt`.

12. Анализ влияния нулей и полюсов на вид АЧХ.

В отдельных полях одного графического окна вывести карты нулей и полюсов и соответствующие нормированные АЧХ (идентификатор `MAGN`) в шкале нормированных частот $\hat{\omega}$ для различных вариантов коэффициентов передаточной функции, представленных в табл. 8.3, которые вычисляются автоматически.

Для одновременного вычисления нормированных АЧХ при четырех вариантах коэффициентов, коэффициенты числителей и знаменателей представить в виде матриц размером 4×3 .

Пояснить соответствие между картой нулей и полюсов и видом АЧХ.

Таблица 8.3. Варианты коэффициентов

Вариант	Векторы коэффициентов передаточной функции	
	числителя	знаменателя
1	[1 0 0]	[1 a1 a2]
2	[1 0 0]	[1 -a1 a2]
3	[1 0 0]	[1 a1 1.2*a2]
4	[1 1 0]	[1 a1 a2]

8.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должна быть представлена табл. 8.2 исходных данных для своего номера бригады $N_{бр}$.

Для *запуска* лабораторной работы необходимо обратиться к script-файлу `lr_08` по его имени:

```
>> lr_08
```

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш <Ctrl>+<Break>.

При выполнении script-файла текущие окна с графиками *не закрывать*.

Листинг script-файла `lr_08` имеет вид:

```
>> type lr_08
script
clc
clear
disp('% ЛР №8. ЛИНЕЙНЫЕ ДИСКРЕТНЫЕ СИСТЕМЫ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Nb = input('Nb = '); % НОМЕР БРИГАДЫ
b = input('b = '); % ВЕКТОР КОЭФФИЦИЕНТОВ ЧИСЛИТЕЛЯ ПЕРЕДАТОЧНОЙ ФУНКЦИИ
a = input('a = '); % ВЕКТОР КОЭФФИЦИЕНТОВ ЗНАМЕНАТЕЛЯ ПЕРЕДАТОЧНОЙ ФУНКЦИИ
N1 = input('N1 = '); % ДЛИНА ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ
N2 = input('N2 = '); % ДЛИНА ВОЗДЕЙСТВИЯ
Fs = input('Fs = '); % ЧАСТОТА ДИСКРЕТИЗАЦИИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.1. ВЫЧИСЛЕНИЕ ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ – функция impz')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ нажмите <ENTER>')
```

```

pause
h1 = impz(b,a,N1);           % ИМПУЛЬСНАЯ ХАРАКТЕРИСТИКА
n = 0:(N1-1);               % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ ИХ
figure('Name','Impulse Response','NumberTitle','off')
subplot(2,1,1), stem(n,h1,'fill','MarkerSize',3), grid
xlabel('n'), ylabel('h(n)')
title('Impulse Response h(n) - impz')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ - функция filter')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ нажмите <ENTER>')
pause
u0 = [1 zeros(1,(N1-1))];   % ЦИФРОВОЙ ЕДИНИЧНЫЙ ИМПУЛЬС
h2 = filter(b,a,u0);        % ИМПУЛЬСНАЯ ХАРАКТЕРИСТИКА
subplot(2,1,2), stem(n,h2,'fill','MarkerSize',3), grid
xlabel('n'), ylabel('h(n)'), title('Impulse Response h(n) - filter')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. ВЫЧИСЛЕНИЕ РЕАКЦИИ ПО ФОРМУЛЕ СВЕРТКИ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ВОЗДЕЙСТВИЯ И РЕАКЦИИ, вычисленной по ФОРМУЛЕ
СВЕРТКИ, нажмите <ENTER>')
pause
x = input_1(N2);            % ВОЗДЕЙСТВИЕ (ДИСКРЕТНЫЙ ПРЯМОУГОЛЬНЫЙ ИМПУЛЬС)
y1 = conv(x,h1);            % РЕАКЦИЯ ДЛИНЫ, РАВНОЙ ДЛИНЕ СВЕРТКИ
L = N1+N2-1;                % ДЛИНА СВЕРТКИ
n = 0:(N2-1);               % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ ВОЗДЕЙСТВИЯ
n1 = 0:(L-1);               % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ СВЕРТКИ
figure('Name','Input and Output Signals','NumberTitle','off')
subplot(4,1,1), stem(n,x,'fill','MarkerSize',3), grid, xlabel('n')
ylabel('x(n)'), title('Input Signal - Discrete Rectangular Impulse x(n)')
subplot(4,1,2), stem(n1,y1,'fill','MarkerSize',3), grid
ylabel('y(n)'), title('Output Signal y1(n) - conv (length = L)')
subplot(4,1,3), stem(n,y1(1:N2),'fill','MarkerSize',3), grid
xlabel('n'), ylabel('y1(n)')
title('Output Signal y1(n) - conv (length = N2)')

```

```

disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. ВЫЧИСЛЕНИЕ РЕАКЦИИ ПО РАЗНОСТНОМУ УРАВНЕНИЮ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА РЕАКЦИИ, вычисленной по РАЗНОСТНОМУ УРАВНЕНИЮ, нажмите <ENTER>')
pause
y2 = filter(b,a,x); % РЕАКЦИЯ ЛДС
subplot(4,1,4), stem(n,y2,'fill','MarkerSize',3), grid
xlabel('n'), ylabel('y(n)')
title('Output Signal y2(n) – filter (length = N2)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ПЕРЕДАТОЧНОЙ ФУНКЦИИ В ВИДЕ ПРОИЗВЕДЕНИЯ ПРОСТЕЙШИХ МНОЖИТЕЛЕЙ')
disp('%')
disp('%')
disp('% Для ВЫВОДА нулей (q) и полюсов (p) В АЛГЕБРАИЧЕСКОЙ ФОРМЕ и коэффициента усиления (K) нажмите <ENTER>')
pause
[q,p,K] = tf2zpk(b,a) % НУЛИ (q) И ПОЛЮСЫ (p) В АЛГЕБРАИЧЕСКОЙ ФОРМЕ И  

% КОЭФФИЦИЕНТ УСИЛЕНИЯ (K)
disp('%')
disp('% Для ВЫВОДА нулей (q) в ПОКАЗАТЕЛЬНОЙ ФОРМЕ нажмите <ENTER>')
pause
disp('% rq – РАДИУСЫ, wq – АРГУМЕНТЫ нулей')
rq = abs(q) % РАДИУСЫ КОМПЛЕКСНО СОПРЯЖЕННЫХ НУЛЕЙ
wq = angle(q) % АРГУМЕНТЫ КОМПЛЕКСНО СОПРЯЖЕННЫХ НУЛЕЙ
disp('% Для ВЫВОДА полюсов (p) в ПОКАЗАТЕЛЬНОЙ ФОРМЕ нажмите <ENTER>')
pause
disp('% rp – РАДИУСЫ, wp – АРГУМЕНТЫ полюсов')
rp = abs(p) % РАДИУСЫ КОМПЛЕКСНО СОПРЯЖЕННЫХ ПОЛЮСОВ
wp = angle(p) % АРГУМЕНТЫ КОМПЛЕКСНО СОПРЯЖЕННЫХ ПОЛЮСОВ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')

```

```

disp('%')
disp('% п.6. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ПЕРЕДАТОЧНОЙ ФУНКЦИИ В ВИДЕ ПРОИЗВЕДЕНИЯ  
МНОЖИТЕЛЕЙ ВТОРОГО ПОРЯДКА')
disp('%')
disp('%')
disp('% Для ВЫВОДА матрицы коэффициентов (s) и коэффициента усиления (G)  
нажмите <ENTER>')
pause
[s,G] = tf2sos(b,a) % КОЭФФИЦИЕНТЫ (s) И КОЭФФИЦИЕНТ УСИЛЕНИЯ (G)
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ПЕРЕДАТОЧНОЙ ФУНКЦИИ В ВИДЕ СУММЫ ПРОСТЫХ  
ДРОБЕЙ')
disp('%')
disp('%')
disp('% Для ВЫВОДА коэффициентов разложения (r), полюсов (p) и целой части (c)  
нажмите <ENTER>')
pause
[r,p,c] = residuez(b,a) % КОЭФФИЦИЕНТЫ РАЗЛОЖЕНИЯ (r) и ПОЛЮСЫ (p) В  
АЛГЕБРАИЧЕСКОЙ ФОРМЕ И ЦЕЛАЯ ЧАСТЬ (c)
disp('%')
disp('%Для ВЫВОДА КОЭФФИЦИЕНТОВ РАЗЛОЖЕНИЯ (r) в ПОКАЗАТЕЛЬНОЙ ФОРМЕ нажмите  
<ENTER>')
pause
rr = abs(r) % РАДИУСЫ КОМПЛЕКСНО СОПРЯЖЕННЫХ КОЭФФИЦИЕНТОВ РАЗЛОЖЕНИЯ  
(r)
wr = angle(r) % АРГУМЕНТЫ КОМПЛЕКСНО СОПРЯЖЕННЫХ КОЭФФИЦИЕНТОВ  
РАЗЛОЖЕНИЯ (r)
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.8. ВЫВОД КАРТЫ НУЛЕЙ И ПОЛЮСОВ')
disp('%')
disp('%')
disp('% Для ВЫВОДА КАРТЫ НУЛЕЙ И ПОЛЮСОВ нажмите <ENTER>')
pause
figure('Name',' Z-plane zero-pole plot','NumberTitle', 'off')
zplane(b,a), title('Z-plane zero-pole plot'), grid
xlabel('Re'), ylabel('jIm')
disp('%')
disp('%')

```

```

disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.9.ВЫЧИСЛЕНИЕ АЧХ и ФЧХ В ШКАЛЕ НОРМИРОВАННЫХ ЧАСТОТ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АЧХ и ФЧХ в шкале НОРМИРОВАННЫХ частот нажмите <ENTER>')
pause
w = 0:pi/100:pi;           % ВЕКТОР НОРМИРОВАННЫХ ЧАСТОТ (РАД)
H_w = freqz(b,a,w);        % КОМПЛЕКСНАЯ ЧАСТОТНАЯ ХАРАКТЕРИСТИКА
MAG_w = abs(H_w);          % АЧХ
PHASE_w = angle(H_w);      % ФЧХ
figure('Name','Magnitude and Phase Responses','NumberTitle','off')
subplot(2,2,1), plot(w,MAG_w), grid, xlabel('w (rad)'), title('MAGNITUDE — |H(w)|')
subplot(2,2,3), plot(w,PHASE_w), grid, xlabel('w (rad)'), title('PHASE — arg [H(w)] (rad)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.10. ВЫЧИСЛЕНИЕ АЧХ и ФЧХ В ШКАЛЕ АБСОЛЮТНЫХ ЧАСТОТ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АЧХ и ФЧХ в шкале АБСОЛЮТНЫХ частот нажмите <ENTER>')
pause
f = 0:Fs/100:Fs/2;         % ВЕКТОР АБСОЛЮТНЫХ ЧАСТОТ (Гц)
H = freqz(b,a,f,Fs);       % КОМПЛЕКСНАЯ ЧАСТОТНАЯ ХАРАКТЕРИСТИКА
MAG = abs(H);              % АЧХ
PHASE = angle(H);          % ФЧХ
subplot(2,2,2), plot(f,MAG), grid, xlabel('f (Hz)'), title('MAGNITUDE — |H(f)|')
subplot(2,2,4), plot(f,PHASE), grid, xlabel('f (Hz)'), title('PHASE — arg [H(f)] (rad)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.11. ОПИСАНИЕ СТРУКТУРЫ РЕКУРСИВНОГО ЗВЕНА')
disp('%')

```

```

disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТОВ dfilt нажмите <ENTER>')
pause
Hd1 = dfilt.df1(b,a)      % ПРЯМАЯ СТРУКТУРА (Direct-Form I)
Hd2 = dfilt.df2(b,a)      % ПРЯМАЯ КАНОНИЧЕСКАЯ СТРУКТУРА (Direct-Form II)
Hd3 = dfilt.df1t(b,a)     % ПРЯМАЯ ТРАНСПОНИРОВАННАЯ СТРУКТУРА (Direct-Form I
Transposed)
Hd4 = dfilt.df2t(b,a)     % ПРЯМАЯ КАНОНИЧЕСКАЯ ТРАНСПОНИРОВАННАЯ СТРУКТУРА
(Direct-Form I Transposed)
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.12. АНАЛИЗ ВЛИЯНИЯ НУЛЕЙ И ПОЛЮСОВ НА ВИД АЧХ')
disp('%')
disp('%')
b(1,:) = [1 0 0];          % КОЭФФИЦИЕНТЫ ЧИСЛИТЕЛЯ – 1-я СТРОКА МАТРИЦЫ
b(2,:) = [1 0 0];          % КОЭФФИЦИЕНТЫ ЧИСЛИТЕЛЯ – 2-я СТРОКА МАТРИЦЫ
b(3,:) = [1 0 0];          % КОЭФФИЦИЕНТЫ ЧИСЛИТЕЛЯ – 3-я СТРОКА МАТРИЦЫ
b(4,:) = [1 1 0];          % КОЭФФИЦИЕНТЫ ЧИСЛИТЕЛЯ – 4-я СТРОКА МАТРИЦЫ
a(1,:) = a;               % КОЭФФИЦИЕНТЫ ЗНАМЕНАТЕЛЯ – 1-я СТРОКА МАТРИЦЫ
a(2,:)=[1 -a(1,2) a(1,3)]; % КОЭФФИЦИЕНТЫ ЗНАМЕНАТЕЛЯ – 2-я СТРОКА МАТРИЦЫ
a(3,:)=[1 a(1,2) 1.2*a(1,3)]; % КОЭФФИЦИЕНТЫ ЗНАМЕНАТЕЛЯ – 3-я СТРОКА МАТРИЦЫ
a(4,:)=[1 a(1,2) a(1,3)]; % КОЭФФИЦИЕНТЫ ЗНАМЕНАТЕЛЯ – 4-я СТРОКА МАТРИЦЫ
w = 0:pi/100:pi;         % ВЕКТОР НОРМИРОВАННЫХ ЧАСТОТ (РАД)
for i=1:4
H3(:,i) = freqz(b(i,:),a(i,:),w); % ЧАСТОТНАЯ ХАРАКТЕРИСТИКА – i-й СТОЛБЕЦ
МАТРИЦЫ
MAG3(:,i) = abs(H3(:,i)); MAX(:,i) = max(MAG3(:,i)); % АЧХ – i-й СТОЛБЕЦ
МАТРИЦЫ – И МАКСИМУМ АЧХ
MAGN(:,i) = MAG3(:,i)/MAX(:,i); % НОРМИРОВАННАЯ АЧХ – i-й СТОЛБЕЦ МАТРИЦЫ
end
disp('% Для вывода КАРТЫ НУЛЕЙ И ПОЛЮСОВ И НОРМИРОВАННОЙ АЧХ нажмите <ENTER>')
pause
figure('Name','Z-plane zero-pole plots and Normalized
Magnitudes','NumberTitle','off')
for i = 1:4
subplot(4,2,2*i-1), zplane(b(i,:),a(i,:)), title('Z-plane zero-pole plot'),
grid
xlabel('Re'), ylabel('jIm')
subplot(4,2,2*i), plot(w,MAGN(:,i)), grid
xlabel('w (rad)'), title('Normalized Magnitude A(w)')
end
disp('%')
disp('%')
disp('% РАБОТА ЗАВЕРШЕНА')
```


8.4.1. Используемые внешние функции

В script-файле `lr_08` используется внешняя функция `input_1`, предназначенная для моделирования воздействия (8.18):

```
function x = input_1(N)
%Формирование воздействия x длины N
for n = 0:(N-1)
    if n<round(N/2)
        x(n+1) = 1;
    else
        x(n+1) = 0;
    end
end
end
```

8.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файлов для моделирования и анализа характеристик двух рекурсивных звеньев 2-го порядка.

В качестве исходных данных использовать коэффициенты передаточной функции (8.17) для своего номера бригады $N_{бр}$ и следующего за ним; для номера бригады $N_{бр} = 30$ второй вариант соответствует $N_{бр} = 1$.

При создании function-файлов представить коэффициенты числителей и знаменателей звеньев в виде двух матриц размером 2×3 (см. аналогично в п. 12 script-файла) и организовать ввод строк каждой из матриц с клавиатуры.

Пункты самостоятельного задания включают в себя:

1С. Вычисление и вывод графиков ИХ двух рекурсивных звеньев.

Для вычисления ИХ использовать функцию `filter`. Задать большую длину воздействия, например 1000, и определить длину ИХ N_1 исходя из условия достижения заданной точности $\varepsilon = 10^{-3}$ (итерационный цикл):

$$\|h(n) - h(n-1)\| \leq \varepsilon, \quad n = 0, 1, \dots, 999. \quad (8.22)$$

2С. Вычисление реакции двух рекурсивных звеньев по формуле свертки на воздействии $x(n)$:

$$x(n) = \begin{cases} U, & n_0 \leq n \leq (n_0 + n_{imp} - 1); \\ 0, & 0 \leq n < n_0; (n_0 + n_{imp}) \leq n \leq (N - 1). \end{cases} \quad (8.23)$$

Исходные данные для воздействия (8.23) заданы в табл. 7.1.

Ограничить реакции до длины воздействия и вывести графики воздействия и реакций.

3С. Вычисление реакции двух рекурсивных звеньев по РУ на воздействие $x(n)$ (8.23).

Вывести графики воздействия и реакций.

Описать прямую и обратную каноническую структуры рекурсивных звеньев в виде объектов `dfilt`.

4С. Вычисление АЧХ двух рекурсивных звеньев в шкале нормированных частот $\hat{\omega}$.

В отдельных полях одного графического окна вывести карты нулей и полюсов и АЧХ рекурсивных звеньев.

8.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения каждого пункта задания, включая копируемые из окна **Command Window** результаты вычислений (шрифт Courier New), созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**) и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

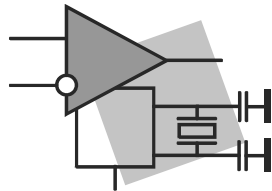
1. Дайте определение ИХ.
2. Запишите формулу свертки.
3. Поясните, как в формуле свертки учитываются ННУ.
4. Запишите РУ общего вида.
5. Поясните, как в РУ учитываются ННУ.
6. Дайте определение рекурсивных и нерекурсивных ЛДС.
7. Поясните принципиальное отличие ИХ рекурсивных и нерекурсивных ЛДС.
8. Приведите тождественные названия рекурсивных и нерекурсивных ЛДС.
9. Дайте определение передаточной функции.
10. Запишите общий вид передаточной функции рекурсивной ЛДС.
11. Приведите основные разновидности передаточной функции рекурсивной ЛДС.
12. Запишите передаточную функцию нерекурсивной ЛДС.
13. Что такое нули и полюсы ЛДС?
14. Что такое карта нулей и полюсов?
15. Дайте определение устойчивости ЛДС.
16. Как определить, является ли ЛДС устойчивой?
17. Дайте определения АЧХ и ФЧХ.
18. Поясните связь частотной характеристики с передаточной функцией.
19. Перечислите основные свойства АЧХ и ФЧХ.

20. Приведите определение и поясните смысл нормированной частоты $\hat{\omega}$.
21. В какой полосе частот и почему рассчитывают АЧХ и ФЧХ?
22. Чем определяется местоположение максимумов АЧХ?
23. Чем определяется местоположение минимумов АЧХ?
24. Чем определяется местоположение нулей АЧХ?
25. В каких точках ФЧХ имеет скачок на π ?
26. Что отображает структура ЛДС и чем определяется ее вид?
27. Назовите четыре разновидности структур рекурсивного звена 2-го порядка.

8.7. Литература

1. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов. — 2-е изд. — СПб.: БХВ-Петербург, 2005. — Главы 4—7.
2. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Главы 9—10.
3. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 4.

ГЛАВА 9



Дискретное преобразование Фурье (часть 1)

Цель работы: изучить дискретное преобразование Фурье (ДПФ) периодических последовательностей и последовательностей конечной длины и овладеть программными средствами его вычисления в MATLAB с использованием алгоритмов быстрого преобразования Фурье (БПФ).

9.1. Краткая теоретическая справка

В гл. 7 мы познакомились с описанием дискретных сигналов во *временной* области. Для описания дискретных сигналов в *частотной* области используется дискретное преобразование Фурье.

9.1.1. Дискретное преобразование Фурье

Дискретным преобразованием Фурье (ДПФ) называется пара взаимно однозначных преобразований:

□ прямое ДПФ (Discrete Fourier Transform — DFT):

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k = 0, 1, \dots, N-1; \quad (9.1)$$

□ обратное ДПФ (ОДПФ) (Inverse Discrete Fourier Transform — IDFT):

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk}, \quad n = 0, 1, \dots, N-1, \quad (9.2)$$

где n — дискретное нормированное время $n = nT/T$; k — дискретная нормированная частота $k = k\Delta\omega/\Delta\omega$; $\Delta\omega = \omega_d/N = 2\pi/NT$ — период дискретизации по частоте (*разрешение по частоте*); $x(n)$ — N -точечная последовательность, т. е. периодическая последовательность во временной области с периодом N ; $X(k)$ — N -точечное ДПФ, т. е. периодическая последовательность в частотной области

с периодом N ; N — период последовательности и ДПФ; $W_N^{nk} = e^{-j\frac{2\pi}{N}nk}$ — поворачивающий множитель; $X(k)W_N^{-nk} = X(k)e^{j\frac{2\pi}{N}nk}$ — k -я дискретная гармоника.

Значения абсолютных частот дискретных гармоник связаны со значениями дискретных нормированных частот соотношением:

$$f = k f_d / N. \quad (9.3)$$

Дискретное преобразование Фурье (9.1) трактуется по-разному в зависимости от вида последовательности $x(n)$ — *периодическая* с периодом N или *конечная* длины N .

Для *периодической* последовательности $x(n)$ с периодом N ДПФ $X(k)$ (9.1) представляет собой ее *спектр* с точностью до множителя $1/N$.

Модуль ДПФ $|X(k)|$ (с точностью до множителя $1/N$) называют *амплитудным спектром*, а аргумент $\arg\{X(k)\}$ — *фазовым спектром* периодической последовательности.

Амплитудный спектр *вещественной* периодической последовательности равен модулю ДПФ $|X(k)|$ с точностью до множителя:

$$\begin{cases} 1/N, & k = 0; \\ 2/N, & k \neq 0. \end{cases} \quad (9.4)$$

При вычислении ДПФ (9.1) периодической последовательности она может задаваться на периоде N или на *целом числе периодов* N , что не меняет результата.

Для *конечной* последовательности $x(n)$ длины N ДПФ $X(k)$ (9.1) представляет собой N дискретных равноотстоящих значений ее спектральной плотности $X(e^{j\omega T})$ на периоде $\omega_d = 2\pi/T$ (см. разд. 9.2.3).

Для *вещественных* последовательностей, периодических и конечных, модуль ДПФ $|X(k)|$ — *четная*, а аргумент $\arg\{X(k)\}$ — *нечетная* функция частоты k .

Согласно определению, при вычислении ДПФ предполагается, что последовательность $x(n)$ является *периодической*, и конечная последовательность представляет собой *один период* периодической последовательности.

При этом *точное* выделение гармоник последовательности $x(n)$ с частотами f_i гарантируется только в том случае, если они кратны периоду дискретизации по частоте $\Delta f = f_d / N$:

$$f_i = q \Delta f, \quad q = 0, 1, \dots, (N-1), \quad (9.5)$$

что, в свою очередь, возможно только в том случае, если на интервале NT последовательности $x(n)$ укладывается целое число периодов T_i , т. е. отношение

$$P_i = \frac{NT}{T_i} = \frac{Nf_i}{f_d} \quad (9.6)$$

является целым числом.

В случае, если условие (9.5) не выполняется, наблюдается эффект *растекания спектра*, который рассматривается в разд. 10.1.1.

В MATLAB ДПФ (9.1)—(9.2) вычисляется с использованием алгоритмов БПФ¹ и ОБПФ с помощью функций:

x = fft(x)

x = ifft(x)

где x и x — N -точечные последовательность $x(n)$ и ее ДПФ $X(k)$ — векторы, нижняя граница индексов которых равна единице, в отличие от ДПФ (9.1)—(9.2), где она равна нулю.

9.1.2. Выделение дискретных гармоник полезного сигнала

При вычислении ДПФ часто ставится задача автоматического определения значений модуля ДПФ $|X(k)|$, превосходящих некоторый заданный порог ε , и соответствующих дискретных нормированных частот k . Фактически, эта задача сводится к выделению полезного сигнала в его аддитивной смеси с шумом.

В учебных целях мы ограничимся рассмотрением двух наиболее простых критериев, согласно которым значение модуля ДПФ $|X(k)|$ аддитивной смеси сигнала с шумом относят к полезному сигналу:

- первый критерий — при заданном пороге ε_1 значение модуля ДПФ $|X(k)|$ относят к полезному сигналу, если выполняется условие:

$$\frac{|X(k)|}{\max |X(k)|} > \varepsilon_1; \quad (9.7)$$

- второй критерий — при заданном пороге ε_2 значение модуля ДПФ $|X(k)|$ относят к полезному сигналу, если выполняется условие:

$$\frac{|X(k)|^2}{P_{cp}} > \varepsilon_2, \quad (9.8)$$

¹ Выбор конкретного алгоритма БПФ скрыт от пользователя и осуществляется автоматически в зависимости от длины исходной последовательности.

где $P_{\text{ср}}$ — средняя мощность аддитивной смеси сигнала с шумом:

$$P_{\text{ср}} = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2. \quad (9.9)$$

Значение порога ε_1 в первом критерии (9.7) задается в пределах:

$$\frac{\max |X(k)_{\text{шума}}|}{\max |X(k)|} < \varepsilon_1 < 1, \quad (9.10)$$

а порога ε_2 во втором критерии (9.8) — в пределах:

$$\frac{\min |X(k)_{\text{сигн}}|^2}{P_{\text{ср}}} < \varepsilon_2 \leq \frac{\max |X(k)|^2}{P_{\text{ср}}} \quad (9.11)$$

при условии, что

$$|X(k)_{\text{сигн}}| > \max |X(k)_{\text{шума}}|. \quad (9.12)$$

Граничные значения порогов в (9.10) и (9.11) можно определить только при априорно известных сигнале и шуме либо их моделях.

При обработке реальных сигналов значение порога ε_1 или ε_2 задается исходя из требований конкретной задачи.

9.1.3. Восстановление спектральной плотности

Спектральная плотность конечной последовательности $x(n)$ длины N :

$$X(e^{j\omega T}) = \sum_{n=0}^{N-1} x(n)e^{-j\omega Tn} \quad (9.13)$$

на периоде $\omega_d = 2\pi/T$ связана с отсчетами ДПФ $X(k)$ (9.1) соотношением:

$$X(k) = X(e^{j\omega T}) \Big|_{\omega=k\frac{2\pi}{NT}}, \quad k = 0, 1, \dots, N-1. \quad (9.14)$$

Значения спектральной плотности (9.13) в L равноотстоящих точках на периоде ω_d при $L > N$ определяются по формуле:

$$X\left(e^{j\frac{2\pi}{L}l}\right) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{L}ln}, \quad l = 0, 1, \dots, L-1, \quad (9.15)$$

где l — дискретная нормированная частота, а $\Delta\omega$ — период дискретизации по частоте:

$$\Delta\omega = \omega_d/L = 2\pi/LT. \quad (9.16)$$

Тот же результат будет получен, если конечную последовательность $x(n)$ длины N дополнить нулями до длины L :

$$\tilde{x}(n) = \begin{cases} x(n), & 0 \leq n \leq (N-1); \\ 0, & N \leq n \leq (L-1), \end{cases} \quad (9.17)$$

и найти ее ДПФ (9.1), заменяя N на L :

$$\tilde{X}(k) = \sum_{n=0}^{L-1} \tilde{x}(n) W_L^{nk}, \quad k = 0, 1, \dots, L-1. \quad (9.18)$$

С учетом (9.17) формула (9.18) принимает вид (сравните с (9.15)):

$$\tilde{X}(k) = \sum_{n=0}^{N-1} x(n) W_L^{nk} = \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi}{L} kn}, \quad k = 0, 1, \dots, L-1.$$

Следует помнить, что *разрешение по частоте*, под которым понимают *минимальное расстояние между дискретными гармониками в ДПФ*, определяется исключительно периодом дискретизации по частоте $\Delta f = f_d/N$ и при фиксированной частоте f_d зависит только от длины (периода) последовательности, поскольку именно она и только она определяет спектральный состав (дискретные гармоники) последовательности.

Поэтому увеличение длины *конечной* последовательности за счет добавления $(L-N)$ нулей и, соответственно, уменьшение периода дискретизации по частоте до $\Delta \tilde{f} = f_d/L$, *не меняет разрешения по частоте*, а лишь *улучшает условия различения* близко расположенных частот дискретных гармоник. Решение этой задачи рассматривается в разд. 10.1.2.

9.1.4. Восстановление аналогового сигнала

Дискретное преобразование Фурье $X(k)$ (9.1) может использоваться для восстановления аналогового периодического сигнала с *финитным спектром*, расположенным в области¹ $(-N/2) \leq k \leq (N/2-1)$, по формуле (усеченный ряд Фурье):

$$x(t) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X_a(k) e^{j \frac{2\pi}{NT} kt}, \quad (9.19)$$

где отсчеты $X_a(k)$ связаны с отсчетами ДПФ $X(k)$ соотношением:

$$X_a(k) = \begin{cases} X(N+k), & -N/2 \leq k \leq -1; \\ X(k), & 0 \leq k \leq (N/2-1). \end{cases} \quad (9.20)$$

¹ При N — четном, и в области $-(N-1)/2 \leq k \leq (N-1)/2$ — при N нечетном.

Тот же результат будет получен при восстановлении аналогового сигнала непосредственно с помощью усеченного ряда Котельникова:

$$x(t) = \sum_{n=0}^{N-1} x(n) \frac{\sin \left[\pi \left(\frac{t}{T} - n \right) \right]}{\pi \left(\frac{t}{T} - n \right)}.$$

(9.21)

В MATLAB для этого удобно воспользоваться функцией:

```
sinc(t/T-n)
```

9.2. Содержание лабораторной работы

Содержание работы связано с вычислением ДПФ периодических и конечных последовательностей и применением ДПФ для выделения полезного сигнала в аддитивной смеси с шумом, восстановления аналогового сигнала и спектральной плотности конечной последовательности с использованием программных средств MATLAB.

9.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файла `lr_09` и function-файлов `fft_e1` и `fft_e2`, которые хранятся на прилагаемом компакт-диске в папке `LAB_DSP\LAB_09`.

Перед выполнением работы необходимо сохранить путь к папке `LAB_09` по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 9.1 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$. Функция $N_{бр} \bmod M$ в записи исходных данных означает вычисление значения $N_{бр}$ по модулю M .

На прилагаемом компакт-диске в папке `Tables\Tables_09` хранятся табл. 9.1 исходных данных и пример ее заполнения для $N_{бр} = 1$.

Таблица 9.1. Таблица исходных данных

Переменная	Назначение	Значение	Идентификатор
$N_{бр}$	Номер бригады	$N_{бр}$	Nb =
N	Период (длина) последовательности	$N = 64$	N = 64
f_d	Частота дискретизации	$f_d = 2000(N_{бр} \bmod 5 + 1)$	Fs =

Таблица 9.1 (окончание)

Переменная	Назначение	Значение	Идентификатор
T	Период дискретизации	$T = 1/f_d$	1 / Fs
A_1	Амплитуды дискретных гармоник	$A_1 = 1 + 0,01N_{бр}$	A1 =
A_2		$A_2 = 2A_1$	A2 =
f_1	Частоты дискретных гармоник	$f_1 = f_d/8$	f1 =

Задание на лабораторную работу связано с вычислением ДПФ и включает в себя следующие пункты:

1. Вычисление амплитудного и фазового спектров периодической последовательности.

Вычислить амплитудный и фазовый спектры периодической последовательности $x(n)$ (идентификатор x) с периодом N :

$$x(nT) = A_1 \cos(2\pi f_1 nT + \pi/4) + A_2 \cos(2\pi f_2 nT + \pi/8), \quad (9.22)$$

используя ее тождественное представление в виде:

$$\begin{aligned} x(n) &= A_1 \cos\left(\frac{2\pi f_1}{f_d} n + \frac{\pi}{4}\right) + A_2 \cos\left(\frac{2\pi f_2}{f_d} n + \frac{\pi}{8}\right) = \\ &= A_1 \cos(\hat{\omega}_1 n + \pi/4) + A_2 \cos(\hat{\omega}_2 n + \pi/8). \end{aligned} \quad (9.23)$$

Вывести графики последовательности $x(n)$ (9.23) на периоде N :

- в шкале дискретного нормированного времени n (идентификатор n);
- в шкале дискретного времени nT (идентификатор nT).

Вычислить ОДПФ от ДПФ последовательности $x(n)$ и вывести график полученной последовательности в шкале дискретного нормированного времени.

Вычислить амплитудный (идентификатор MOD) и фазовый¹ (идентификатор $PHASE$) спектры последовательности $x(n)$ (9.23) с учетом (9.4) и вывести их графики:

- в шкале дискретных нормированных частот k (идентификатор k);
- в шкале абсолютных частот f (Гц) (идентификатор f).

¹ Если модуль ДПФ меньше заданного, близкого к нулю, порога, то значения фазового спектра следует обнулить. В противном случае отношение малых, сравнимых с нулем, мнимой и вещественной частей может существенно отличаться от нуля, что обусловлено спецификой вычислений в MATLAB.

Пояснить:

- связь дискретного нормированного времени с дискретным временем;
- связь частоты f (Гц) с дискретной нормированной частотой;
- вид амплитудного и фазового спектров.

2. Вычисление ДПФ конечной последовательности.

Вычислить ДПФ конечной последовательности $x(n)$ (9.23) длины N .

Вывести графики в шкале дискретных нормированных частот:

- модуля ДПФ (идентификатор `MOD_K`) конечной последовательности;
- амплитудного спектра периодической последовательности (см. п. 1).

Пояснить связь модуля ДПФ конечной последовательности с амплитудным спектром периодической последовательности.

3. Определение амплитуд и частот дискретных гармоник.

Для автоматического определения амплитуд и частот гармоник в амплитудном спектре периодической последовательности $x(n)$ (9.23) использовать function-файл `fft_e1` (см. разд. 9.4.1), задавая малое, сравнимое с нулем, значение порога $\varepsilon_1 = 10^{-7}$ (идентификатор `e1`).

Вывести:

- выходные параметры function-файла `fft_e1`;
- значения амплитуд, дискретных нормированных частот и абсолютных частот (Гц) гармоник.

Пояснить:

- смысл выходных параметров function-файла `fft_e1`;
- соответствие между значениями дискретных нормированных частот и абсолютных частот гармоник.

4. Граничные значения порогов для первого (9.7) и второго (9.8) критериев выделения полезного сигнала.

Сформировать аддитивную смесь $s(n)$ (идентификатор `s`) полезного периодического сигнала $x(n)$ (9.23) с нормальным белым шумом $r(n)$ с нулевым средним значением и единичной дисперсией:

$$s(n) = x(n) + r(n). \quad (9.24)$$

Для аддитивной смеси $s(n)$ (9.24) определить:

- граничные значения порога ε_1 для первого критерия (9.7) (идентификаторы `e1_low` и `e1_up`);

- граничные значения порога ε_2 для второго критерия (9.8) (идентификаторы `e2_low` и `e2_up`).

Пояснить, как рассчитываются граничные значения порогов ε_1 и ε_2 .

5. Выделение полезного сигнала по первому критерию.

Вывести графики:

- аддитивной смеси $s(n)$ (9.24) на периоде N ;
- амплитудного спектра аддитивной смеси $s(n)$ в шкале дискретных нормированных частот;
- амплитудного спектра аддитивной смеси $s(n)$, нормированного к его максимальному значению (см. (9.7)).

Этот график позволяет уточнить значение порога ε_1 в диапазоне его граничных значений, определенных в п. 4.

Ввести значение порога ε_1 .

Для выделения полезного сигнала по первому критерию (9.7) использовать function-файл `fft_e1` (см. разд. 9.4.1).

Вывести выходные параметры function-файла `fft_e1`.

Пояснить:

- какое значение порога ε_1 было выбрано и чем обоснован выбор;
- смысл выходных параметров function-файла `fft_e1`;
- какие амплитуды гармоник соответствуют полезному сигналу согласно первому критерию (9.7);
- в каком случае применение первого критерия будет неэффективным.

6. Выделение полезного сигнала по второму критерию.

Вывести графики:

- амплитудного спектра аддитивной смеси $s(n)$ (9.24) в шкале дискретных нормированных частот;
- квадрата амплитудного спектра аддитивной смеси $s(n)$, нормированного к ее средней мощности (см. (9.8)).

Этот график позволяет уточнить значение порога ε_2 в диапазоне его граничных значений, определенных в п. 4.

Ввести значение порога ε_2 .

Для выделения полезного сигнала по второму критерию (9.8) использовать function-файл `fft_e2` (см. разд. 9.4.1).

Вывести выходные параметры function-файла `fft_e2`.

Пояснить:

- какое значение порога ε_2 было выбрано и чем обоснован выбор;
- смысл выходных параметров function-файла `fft_e2`;
- какие амплитуды гармоник соответствуют полезному сигналу согласно второму критерию (9.8);
- в каком случае применение второго критерия будет неэффективным.

7. Восстановление аналогового сигнала.

Восстановить периодический аналоговый сигнал $x(t)$ (идентификатор `xa`) по отсчетам ДПФ $X(k)$ периодической последовательности $x(n)$ (9.23). Для вычисления значений сигнала $x(t)$ использовать формулу (9.19), задавая значения времени t (идентификатор `t`) на интервале $t \in [0; (N-1)T]$ с шагом $\Delta t = 0,25T$.

В тех же точках вычислить значения исходного аналогового сигнала $x_{\text{исх}}(t)$ (идентификатор `xt`), на основе которого получена последовательность $x(nT)$ (9.22):

$$x_{\text{исх}}(t) = A_1 \cos(2\pi f_1 t + \pi/4) + A_2 \cos(2\pi f_2 t + \pi/8). \quad (9.25)$$

Вывести графики:

- периодической последовательности $x(n)$ (9.23) и модуля ее ДПФ;
- восстановленного аналогового сигнала $x(t)$ и его амплитудного спектра (идентификатор `MODa`);
- исходного аналогового сигнала $x_{\text{исх}}(t)$ (9.25).

Пояснить:

- связь модуля ДПФ последовательности со спектром аналогового сигнала;
- результат визуального сравнения восстановленного и исходного сигналов.

8. Восстановление спектральной плотности конечной последовательности.

Вычислить значения спектральной плотности конечной последовательности $x(n)$ (9.23) длины N в $L = 2N$ точках на периоде $\hat{\omega}_d = 2\pi$ двумя способами:

- по формуле (9.15) — идентификатор `xw`;
- по формуле (9.18) — идентификатор `xz`.

Вывести графики:

- модуля ДПФ конечной последовательности $x(n)$ (см. п. 2) в шкале дискретных нормированных частот с помощью функции `stem`;
- модулей спектральной плотности, вычисленной первым и вторым способами в шкале частот $\hat{\omega}$ (идентификатор `w`) с помощью функции `plot`.

Пояснить:

- связь между ДПФ и спектральной плотностью;
- алгоритмы вычисления значений спектральной плотности по формулам (9.15) и (9.18);
- соответствие между частотами $\hat{\omega}$ (рад) пиков спектральной плотности и их дискретными нормированными частотами.

9. Уменьшение периода дискретизации по частоте при вычислении ДПФ.

Сформировать три конечные последовательности $x(n)$ (9.23) (вектор `xz`) с длинами $L = N, 2N, 4N$ (вектор `L`), дополняя их нулями до длины L при $L > N$.

Вычислить ДПФ $\tilde{X}(k)$ (9.18) данных последовательностей (вектор `xz`).

Вывести графики:

- исходной последовательности и последовательностей, дополненных нулями;
- их модулей ДПФ в шкале дискретных нормированных частот (пунктиром с помощью функции `stem`) и одновременно — восстановленных спектральных плотностей (с помощью функции `plot` красным цветом).

Для сравнения графиков удобно воспользоваться кнопкой **Zoom in** на панели инструментов.

Вывести значения периодов ДПФ (вектор `L`) и соответствующих им периодов дискретизации по частоте (вектор `Delta_f`).

Пояснить:

- причину изменения периода дискретизации по частоте;
- изменяется ли при этом разрешение по частоте;
- чему равно разрешение по частоте;
- с какой целью уменьшают период дискретизации по частоте.

9.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должна быть представлена табл. 9.1 исходных данных для своего номера бригады $N_{бр}$.

Для *запуска* лабораторной работы необходимо обратиться к script-файлу `1r_09` по его имени:

```
>> 1r_09
```

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш `<Ctrl>+<Break>`.

Листинг script-файла lr_09 имеет вид:

```
>> type lr_09
script
clc
clear
disp('% ЛР №9. ДИСКРЕТНОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ (часть 1)')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
N = input('N = ');             % ДЛИНА (ПЕРИОД) ПОСЛЕДОВАТЕЛЬНОСТИ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ
T = input('T = ');             % ПЕРИОД ДИСКРЕТИЗАЦИИ 1/Fs
A1 = input('A1 = ');           % АМПЛИТУДЫ ДИСКРЕТНЫХ ГАРМОНИК
A2 = input('A2 = ');
f1 = input('f1 = ');           % ЧАСТОТЫ (Гц) ДИСКРЕТНЫХ ГАРМОНИК
f2 = input('f2 = ');
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('% Для вывода ИСХОДНЫХ АМПЛИТУД и ЧАСТОТ ДИСКРЕТНЫХ ГАРМОНИК нажмите <ENTER>')
pause
disp('%')
disp(['      A1 = ' num2str(A1) '      A2 = ' num2str(A2)])
disp(['      f1 = ' num2str(f1) '      f2 = ' num2str(f2)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.1. ВЫЧИСЛЕНИЕ АМПЛИТУДНОГО И ФАЗОВОГО СПЕКТРОВ ПЕРИОДИЧЕСКОЙ ПОСЛЕДОВАТЕЛЬНОСТИ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ периодической последовательности нажмите <ENTER>')
pause
n = 0:(N-1);                   % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
k = 0:(N-1);                   % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
w1 = 2*pi*f1/Fs; w2 = 2*pi*f2/Fs; % НОРМИРОВАННЫЕ ЧАСТОТЫ ДИСКРЕТНЫХ ГАРМОНИК
(РАД)
```

```

x = A1*cos(w1*n+pi/4)+A2*cos(w2*n+pi/8);    % ПЕРИОДИЧЕСКАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ
X = fft(x);                                % ДПФ ПЕРИОДИЧЕСКОЙ ПОСЛЕДОВАТЕЛЬНОСТИ
MOD = (2/N)*abs(X);                         % АМПЛИТУДНЫЙ СПЕКТР ПЕРИОДИЧЕСКОЙ
ПОСЛЕДОВАТЕЛЬНОСТИ
MOD(1) = (1/N)*abs(X(1));
PHASE = angle(X);                          % ФАЗОВЫЙ СПЕКТР ПЕРИОДИЧЕСКОЙ
ПОСЛЕДОВАТЕЛЬНОСТИ
for i = 1:N
    if (abs(X(i)) < 1e-4)
        PHASE(i)=0;
    end
end
figure('Name','Periodic Sequence','NumberTitle','off')
subplot(3,1,1), stem(n,x, 'MarkerSize',3,'Linewidth',2)
grid, xlabel('n')
ylabel('x(n)'), title(strcat(['Periodic Sequence x(n)  N = ',num2str(N)]))
subplot(3,1,2), stem(n/Fs,x, 'MarkerSize',3,'Linewidth',2)
grid, xlabel('nT')
ylabel('x(nT)'), title(strcat(['Periodic Sequence x(nT)  N = ',num2str(N)]))
x = ifft(X);                               % ПЕРИОДИЧЕСКАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ,
ВЫЧИСЛЕННАЯ С ПОМОЩЬЮ ОДПФ
subplot(3,1,3), stem(n,x, 'MarkerSize',3,'Linewidth',2)
grid, xlabel('n')
ylabel('x(n)'), title(strcat(['Periodic Sequence x = ifft(X)  N = ',num2str(N)]))
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНОГО СПЕКТРА периодической
последовательности нажмите <ENTER>')
pause
figure('Name','Amplitude Spectrum','NumberTitle','off')
subplot(2,1,1), stem(k,MOD, 'MarkerSize',3,'Linewidth',2), grid
xlabel('k'), ylabel('1/N|X(k)|')
title(strcat(['Amplitude Spectrum of the Periodic Sequence N = ',num2str(N)]))
subplot(2,1,2), stem(k*(Fs/N),MOD, 'MarkerSize',3,'Linewidth',2), grid
xlabel('f (Hz)'), ylabel('1/N|X(f)|')
title(strcat(['Amplitude Spectrum of the Periodic Sequence N = ',num2str(N)]))
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ФАЗОВОГО СПЕКТРА периодической последовательности
нажмите <ENTER>')
pause
figure('Name','Phase Spectrum','NumberTitle','off')
subplot(2,1,1), stem(k, PHASE, 'MarkerSize',3,'Linewidth',2), grid
xlabel('k'), ylabel('arg{X(k)} (rad)')
title(strcat(['Phase Spectrum of the Periodic Sequence N = ',num2str(N)]))
subplot(2,1,2), stem(k*(Fs/N), PHASE, 'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), ylabel('arg{X(f)} (rad)')
```



```

title(strcat(['Phase Spectrum of the Periodic Sequence N = ',num2str(N)]))
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ДПФ КОНЕЧНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ МОДУЛЯ ДПФ конечной последовательности
и АМПЛИТУДНОГО СПЕКТРА')
disp('% периодической последовательности нажмите <ENTER>')
pause
MOD_K = abs(fft(x));           % МОДУЛЬ ДПФ КОНЕЧНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ
figure('Name','DFT Modulus and Amplitude Spectrum', 'NumberTitle','off')
subplot(2,1,1), stem(k,MOD_K, 'MarkerSize',3,'Linewidth',2), grid
xlabel('k'), ylabel('|X(k)|')
title('DFT Modulus of the Finite Sequence')
subplot(2,1,2), stem(k,MOD,'MarkerSize',3,'Linewidth',2), grid
xlabel('k'), ylabel('1/N |X(k)|')
title('Amplitude Spectrum of the Periodic Sequence')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. ОПРЕДЕЛЕНИЕ АМПЛИТУД И ЧАСТОТ ДИСКРЕТНЫХ ГАРМОНИК')
disp('%')
disp('%')
disp('% Для вывода ВЫХОДНЫХ ПАРАМЕТРОВ ФУНКЦИИ fft_e1 нажмите <ENTER>')
pause
e1 = 1e-7;                     % ЗНАЧЕНИЕ ПОРОГА ДЛЯ ПЕРВОГО КРИТЕРИЯ
[MODm,m] = fft_e1(MOD,e1)      % ВНЕШНЯЯ ФУНКЦИЯ ДЛЯ ВЫДЕЛЕНИЯ АМПЛИТУД И ЧАСТОТ
ГАРМОНИК ПОЛЕЗНОГО СИГНАЛА ПО ПЕРВОМУ КРИТЕРИЮ
disp('%')
disp('%')
disp('% Для вывода АМПЛИТУД И ЧАСТОТ ДИСКРЕТНЫХ ГАРМОНИК нажмите <ENTER>')
pause
A1 = MODm(1); A2 = MODm(2);   % АМПЛИТУДЫ ДИСКРЕТНЫХ ГАРМОНИК
k1 = m(1); k2 = m(2);        % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ
f1 = k1*Fs/N; f2 = k2*Fs/N;  % ЧАСТОТЫ (Гц) ДИСКРЕТНЫХ ГАРМОНИК
disp('%')
disp('%')
disp(['          A1 = ' num2str(A1) '          A2 = ' num2str(A2)])
disp(['          k1 = ' num2str(k1) '          k2 = ' num2str(k2)])

```

```

disp(['      f1 = ' num2str(f1) '      f2 = ' num2str(f2)])
disp('%')
disp('%')
disp('% СРАВНИТЕ с ВЫХОДНЫМИ ПАРАМЕТРАМИ функции fft_e1 и исходными данными')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. ГРАНИЧНЫЕ ЗНАЧЕНИЯ ПОРОГОВ ДЛЯ ПЕРВОГО И ВТОРОГО КРИТЕРИЕВ  

ВЫДЕЛЕНИЯ ПОЛЕЗНОГО СИГНАЛА')
disp('%')
disp('%')
disp('% Для вывода граничных значений порога для ПЕРВОГО КРИТЕРИЯ нажмите  

<ENTER>')
pause
noise = randn(1,N);           % НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ
s = x+noise;                 % АДДИТИВНАЯ СМЕСЬ СИГНАЛА С ШУМОМ
S = fft(s);                  % ДПФ СМЕСИ СИГНАЛА С ШУМОМ
MODS = (2/N)*abs(S);         % АМПЛИТУДНЫЙ СПЕКТР СМЕСИ СИГНАЛА С ШУМОМ
MODS(1) = (1/N)*abs(S(1));
NOISE = fft(noise);          % ДПФ ШУМА
MODNOISE = (2/N)*abs(NOISE); % АМПЛИТУДНЫЙ СПЕКТР ШУМА
MODNOISE(1) = (1/N)*abs(NOISE(1));
MAX_NOISE = max(MODNOISE);    % МАКСИМУМ АМПЛИТУДНОГО СПЕКТРА ШУМА
MAXS = max(MODS);            % МАКСИМУМ АМПЛИТУДНОГО СПЕКТРА СМЕСИ СИГНАЛА С  

ШУМОМ
e1_low = MAX_NOISE/MAXS; % НИЖНЯЯ ГРАНИЦА ПОРОГА ДЛЯ ПЕРВОГО КРИТЕРИЯ
e1_up = 1;                  % ВЕРХНЯЯ ГРАНИЦА ПОРОГА ДЛЯ ПЕРВОГО КРИТЕРИЯ
P = (1/N)*sum(MODS.^2);      % СРЕДНЯЯ МОЩНОСТЬ СМЕСИ СИГНАЛА С ШУМОМ
MAXS2 = MAXS.^2;            % КВАДРАТ МАКСИМУМА АМПЛИТУДНОГО СПЕКТРА СМЕСИ СИГНАЛА  

С ШУМОМ
MAX_NOISE2 = MAX_NOISE.^2;   % КВАДРАТ МАКСИМУМА АМПЛИТУДНОГО СПЕКТРА ШУМА
disp('%')
disp('%')
disp(['      e1_low = ' num2str(e1_low) '      e1_up = ' num2str(e1_up)])
disp('%')
disp('%')
disp('% Для вывода граничных значений порога для ВТОРОГО КРИТЕРИЯ нажмите  

<ENTER>')
pause
e2_low = MAX_NOISE2/P;       % НИЖНЯЯ ГРАНИЦА ПОРОГА ДЛЯ ВТОРОГО КРИТЕРИЯ
e2_up = MAXS2/P;            % ВЕРХНЯЯ ГРАНИЦА ПОРОГА ДЛЯ ВТОРОГО КРИТЕРИЯ
disp('%')
disp('%')
disp(['      e2_low = ' num2str(e2_low) '      e2_up = ' num2str(e2_up)])
disp('%')

```

```

disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ВЫДЕЛЕНИЕ ПОЛЕЗНОГО СИГНАЛА ПО ПЕРВОМУ КРИТЕРИЮ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА аддитивной смеси сигнала с шумом нажмите <ENTER>')
pause
figure('Name','Mixture of Signal and Noise','NumberTitle','off')
stem(n,s,'MarkerSize',3,'Linewidth',2), grid
xlabel('n'), ylabel('s(n)')
title(strcat(['Mixture of Signal and Noise N = ',num2str(N)]))
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ амплитудного и НОРМИРОВАННОГО амплитудного спектров')
disp('% аддитивной смеси сигнала с шумом нажмите <ENTER>')
pause
figure('Name','Amplitude Spectrum and Normalized Amplitude Spectrum','NumberTitle','off')
subplot(2,1,1), stem(k,MODS,'MarkerSize',3,'Linewidth',2), grid
xlabel('k'), ylabel('|S(k)|')
title(strcat(['Amplitude Spectrum N = ',num2str(N)]))
subplot(2,1,2), stem(k, MODS/MAXS,'MarkerSize',3,'Linewidth',2)
grid, xlabel('k'), ylabel('|S(k)|/max|S(k)|')
title(strcat(['Normalized Amplitude Spectrum N = ',num2str(N)]))
disp('%')
disp('%')
disp('% Введите выбранное значение порога e1 для ПЕРВОГО КРИТЕРИЯ')
disp('%')
e1 = input('    e1 = '); % ВЫБРАННОЕ ЗНАЧЕНИЕ ПОРОГА ДЛЯ ПЕРВОГО КРИТЕРИЯ
disp('%')
disp('% Для вывода ВЫХОДНЫХ ПАРАМЕТРОВ ФУНКЦИИ fft_e1 нажмите <ENTER>')
pause
[MODm,m] = fft_e1(MODS,e1) % ВНЕШНЯЯ ФУНКЦИЯ ДЛЯ ВЫДЕЛЕНИЯ АМПЛИТУД И ЧАСТОТ ГАРМОНИК ПОЛЕЗНОГО СИГНАЛА ПО ПЕРВОМУ КРИТЕРИЮ
disp('%')
disp('%')
disp('% СРАВНИТЕ значения ВЫДЕЛЕННЫХ ПО ПЕРВОМУ КРИТЕРИЮ АМПЛИТУД И ЧАСТОТ')
disp('% с исходными данными')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')

```

```

disp('% п.6. ВЫДЕЛЕНИЕ ПОЛЕЗНОГО СИГНАЛА ПО ВТОРОМУ КРИТЕРИЮ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ амплитудного спектра и КВАДРАТА амплитудного')
disp('% спектра, НОРМИРОВАННОГО к величине средней мощности')
disp('% аддитивной смеси сигнала с шумом, нажмите <ENTER>')
pause
figure('Name','Amplitude Spectrum and Normalized Amplitude Spectrum Squire','NumberTitle','off')
subplot(2,1,1), stem(k,MODS,'MarkerSize',3,'Linewidth',2), grid
xlabel('k'), ylabel('|S(k)|')
title(strcat(['Amplitude Spectrum N = ',num2str(N)]))
subplot(2,1,2), stem(k,(MODS.^2)/P,'MarkerSize',3,'Linewidth',2)
grid, xlabel('k'), ylabel('|S(k)|^2/P')
title(strcat(['Normalized Amplitude Spectrum Squire N = ',num2str(N)]))
disp('%')
disp('%')
disp('% Введите выбранное значение порога e2 для ВТОРОГО КРИТЕРИЯ')
disp('%')
e2 = input('      e2 = '); % ВЫБРАННОЕ ЗНАЧЕНИЕ ПОРОГА ДЛЯ ВТОРОГО КРИТЕРИЯ
disp('%')
disp('% Для вывода ВЫХОДНЫХ ПАРАМЕТРОВ ФУНКЦИИ fft_e2 нажмите <ENTER> ')
pause
[MODm,m] = fft_e2(MODS,e2)% ВНЕШНЯЯ ФУНКЦИЯ ДЛЯ ВЫДЕЛЕНИЯ АМПЛИТУД И ЧАСТОТ
ГАРМОНИК ПОЛЕЗНОГО СИГНАЛА ПО ВТОРОМУ КРИТЕРИЮ
disp('%')
disp('%')
disp('% СРАВНИТЕ значения ВЫДЕЛЕННЫХ ПО ВТОРОМУ КРИТЕРИЮ АМПЛИТУД И ЧАСТОТ')
disp('% с исходными данными')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. ВОССТАНОВЛЕНИЕ АНАЛОГОВОГО СИГНАЛА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ПОСЛЕДОВАТЕЛЬНОСТИ и МОДУЛЯ ее ДПФ,')
disp('% ВОССТАНОВЛЕННОГО АНАЛОГОВОГО СИГНАЛА и его СПЕКТРА')
disp('% и ИСХОДНОГО АНАЛОГОВОГО СИГНАЛА нажмите <ENTER>')
pause
Xa = [X(N/2+1:N),X(1:N/2)]; % СПЕКТР АНАЛОГОВОГО СИГНАЛА (С ТОЧНОСТЬЮ ДО
ПОСТОЯННОГО МНОЖИТЕЛЯ)
i = 1; % СЧЕТЧИК ЗНАЧЕНИЙ АНАЛОГОВОГО СИГНАЛА
for t = 0:0.25*T:(N-1)*T % ЗНАЧЕНИЯ НЕПРЕРЫВНОГО ВРЕМЕНИ
    s = 0;

```

```

for k = -N/2:N/2-1           % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
    s = s + Xa(k*N/2+1)*exp(j*2*pi*k*t/(N*T)); % ВОССТАНОВЛЕНИЕ АНАЛОГОВОГО
СИГНАЛА
end
xa(i) = (1/N).*s;           % ЗНАЧЕНИЯ ВОССТАНОВЛЕННОГО АНАЛОГОВОГО СИГНАЛА
i = i+1;
end

t = 0:0.25*T:(N-1)*T;
xt = A1*cos(2*pi*f1*t+pi/4)+A2*cos(2*pi*f2*t+pi/8); % ЗНАЧЕНИЯ ИСХОДНОГО
АНАЛОГОВОГО СИГНАЛА
k = 0:N-1;                 % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
MODa = (2/N)*abs(Xa);      % АМПЛИТУДНЫЙ СПЕКТР ВОССТАНОВЛЕННОГО АНАЛОГОВОГО
СИГНАЛА
MODa(1) = (1/N)*abs(Xa(1));
figure('Name','Original Periodic Sequence & FFT, Reconstructed Analog Signal &
Spectrum, Original Analog Signal','NumberTitle','off')
subplot(3,2,1), stem(n,x,'MarkerSize',3), grid
xlabel('n'), ylabel('x(n)')
title(strcat(['Original Periodic Sequence N = ',num2str(N)]))
subplot(3,2,2), stem(k,abs(X),'MarkerSize',3,'Linewidth',2), grid
xlabel('k'), ylabel('|X(k)|')
title(strcat(['DFT of Original Periodic Sequence N = ',num2str(N)]))
subplot(3,2,3), plot(t,real(xa)), grid, xlabel('t')
ylabel('x(t)'),title('Reconstructed Analog Signal')
k = -N/2:N/2-1;
subplot(3,2,4), stem(k,MODa,'MarkerSize',3,'Linewidth',2), grid
xlabel('k'), ylabel('|Xa(k)|')
title('Amplitude Spectrum of Reconstructed Analog Signal')
subplot(3,2,5), plot(t,xt), grid, xlabel('t')
ylabel('x(t)'), title('Original Analog Signal')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('%')
disp('% п.8. ВОССТАНОВЛЕНИЕ СПЕКТРАЛЬНОЙ ПЛОТНОСТИ КОНЕЧНОЙ
ПОСЛЕДОВАТЕЛЬНОСТИ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ДПФ и СПЕКТРАЛЬНОЙ ПЛОТНОСТИ конечной')
disp('% последовательности, вычисленной ДВУМЯ способами, нажмите <ENTER>')
pause
L = 2*N;                   % КОЛИЧЕСТВО ОТСЧЕТОВ СПЕКТРАЛЬНОЙ ПЛОТНОСТИ НА ПЕРИОДЕ
l = 0;
for l = 0:(L-1)             % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
    S = 0;

```

```

for n = 0:(N-1)      % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
    S = S + x(n+1)*exp(-j*2*pi*l*n/L);      % ВОССТАНОВЛЕНИЕ СПЕКТРАЛЬНОЙ
ПЛОТНОСТИ
end
XW(l+1) = S;        % ЗНАЧЕНИЯ ВОССТАНОВЛЕННОЙ СПЕКТРАЛЬНОЙ ПЛОТНОСТИ
l = l+1;
end

xz = [x zeros(1, (L-N))];      % ПОСЛЕДОВАТЕЛЬНОСТЬ, ДОПОЛНЕННАЯ НУЛЯМИ ДО ДЛИНЫ
L
XZ = fft(xz);                  % ДПФ ПОСЛЕДОВАТЕЛЬНОСТИ, ДОПОЛНЕННОЙ НУЛЯМИ
k = 0:(N-1);                   % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
w = 0:2*pi/L:2*pi-2*pi/L;      % НОРМИРОВАННАЯ ЧАСТОТА
l = 0:(L-1);                   % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
figure('Name','DFT and Spectral Density','NumberTitle','off')
subplot(3,1,1), stem(k,abs(X),'MarkerSize',3,'Linewidth',2)
grid, xlabel('k'), ylabel('|X(k)|')
title(strcat(['DFT Modulus N = ',num2str(N)]))
subplot(3,1,2), plot(w,abs(XW),'MarkerSize',3,'Linewidth',2)
grid, xlabel('w'), ylabel('|X(w)|')
title(strcat(['Spectral Density Modulus (option 1) L = ',num2str(L)]))
subplot(3,1,3), plot(w,abs(XZ),'MarkerSize',3,'Linewidth',2)
grid, xlabel('w'), ylabel('|X(w)|')
title(strcat(['Spectral Density Modulus (option 2) L = ',num2str(L)]))
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.9. УМЕНЬШЕНИЕ ПЕРИОДА ДИСКРЕТИЗАЦИИ ПО ЧАСТОТЕ ПРИ ВЫЧИСЛЕНИИ ДПФ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ КОНЕЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ,')
disp('% ДПФ и СПЕКТРАЛЬНЫХ ПЛОТНОСТЕЙ нажмите <ENTER>')
pause
figure('Name','Finite Sequences, DFT and Spectral Densities','NumberTitle',
'off')
L = [N 2*N 4*N];
for i = 1:length(L)
    xz = [x zeros(1, (L(i)-N))];      % ПОСЛЕДОВАТЕЛЬНОСТЬ, ДОПОЛНЕННАЯ НУЛЯМИ ДО
ДЛИНЫ L(i)
    XZ = fft(xz);
    Delta_f(i) = Fs/L(i);
    n = 0:length(xz)-1;                % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
    k = 0:length(XZ)-1;                % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
    subplot(3,2,2*i-1), stem(n,xz,'MarkerSize',3), xlabel('n'), grid
    title(strcat(['Finite Sequence x(n) L = ',num2str(L(i)]))

```

```

subplot(3,2,2*i), plot(k,abs(XZ), 'r','MarkerSize',3, 'Linewidth',2), grid,
hold on, stem(k,abs(XZ),':'), xlabel('k')
title(strcat(['DFT and Spectral Density Modulus L = ',num2str(L(i))]))
end
disp('%')
disp('%')
disp('% Для вывода ПЕРИОДОВ ДПФ и ПЕРИОДОВ ДИСКРЕТИЗАЦИИ ПО ЧАСТОТЕ нажмите
<ENTER>')
pause
disp('%')
disp(['      L = ',num2str(L) ''])
disp('%')
disp(['      Delta_f = ',num2str(Delta_f) ''])
disp('%')
disp('%')
disp('% РАБОТА ЗАВЕРШЕНА')

```

9.4.1. Используемые внешние функции

В script-файле `lr_09` используются две внешние функции.

- Внешняя функция `fft_e1`, предназначенная для определения значений модуля ДПФ (MODm) и дискретных нормированных частот (m) гармоник, которые согласно первому критерию (9.7) при заданном пороге ε_1 (9.10) ($e1$) относят к полезному сигналу:

```

function [MODm,m] = fft_e1(MODX,e1)
% Определение значений модуля ДПФ и частот полезного сигнала
%
% MODX — вектор значений модуля ДПФ смеси сигнала с шумом
% e1 — заданный порог
%
% MODm — вектор значений модуля ДПФ полезного сигнала
% m — вектор значений частот полезного сигнала
%
i = 1;
MAX = max(MODX);
for k = 1:length(MODX)
    if (MODX(k)/MAX) > e1
        MODm(i) = MODX(k);
        m(i) = k-1;
        i = i+1;
    end
end
end

```

- Внешняя функция `fft_e2`, предназначенная для определения значений модуля ДПФ (MODm) и дискретных нормированных частот (m) гармоник, которые согласно второму критерию (9.8) при заданном пороге ε_2 (9.11) ($e2$) относят к полезному сигналу:

```
function [MODm,m] = fft_e2(MODX,e2)
```

```
% Определение значений модуля ДПФ и частот полезного сигнала
```

```
%
```

```
% MODX — вектор значений модуля ДПФ смеси сигнала с шумом
```

```
% e2 — заданный порог
```

```
%
```

```
% MODm — вектор значений модуля ДПФ полезного сигнала
```

```
% m — вектор значений частот полезного сигнала
```

```
%
```

```
i = 1;
```

```
P = sum(MODX.^2)/length(MODX); % P — средняя мощность смеси сигнала с шумом
```

```
for k = 1:length(MODX)
```

```
    if ((MODX(k).^2)/P)>e2
```

```
        MODm(i) = MODX(k);
```

```
        m(i) = k-1;
```

```
        i = i+1;
```

```
    end
```

```
end
```

9.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файлов для вычисления ДПФ последовательностей с использованием исходных данных из табл. 9.1 для своего номера бригады $N_{бр}$.

Последовательности выбираются из представленного далее списка:

1С. Периодическая последовательность с периодом $N/2$:

$$x(n) = A_1 \cos(\hat{\omega}_1 n + \pi/4) + A_2 \cos(\hat{\omega}_2 n + \pi/16). \quad (9.26)$$

Вывести графики амплитудного и фазового спектра периодической последовательности.

Определить амплитуды и частоты дискретных гармоник, используя function-файл `fft_e1`.

2С. Конечная последовательность $x(n)$ (9.26) длины $N/2$.

Вывести графики модуля и аргумента ДПФ конечной последовательности.

3С. Конечная последовательность длины N :

$$x_1(n) = \begin{cases} x_2(n), & n = 0, \dots, N/2 - 1; \\ x_3(n), & n = N/2, \dots, N - 1, \end{cases} \quad (9.27)$$

где

$$x_2(n) = A_1 \cos(\hat{\omega}_1 n), \quad n = 0, \dots, N/2 - 1;$$

$$x_3(n) = A_2 \cos(\hat{\omega}_2 n), \quad n = 0, \dots, N/2 - 1.$$

Вывести графики конечных последовательностей $x(n)$ (9.23) и $x_1(n)$ (9.27) и модулей их ДПФ.

4С. Цифровой единичный импульс (7.10) на интервале $n \in [0; N-1]$.

Вывести графики цифрового единичного импульса и модуля его ДПФ.

5С. Последовательность с однотональной амплитудной модуляцией (7.23):

$$x(n) = C[1 + m\cos(\Omega n + \phi_\Omega)]\cos(\hat{\omega}_0 n + \phi_0).$$

Задать значения $C = 1$, $\hat{\omega}_0 = 2\pi/4$, $\phi_0 = 0$, $\Omega = \hat{\omega}_0/4$, $\phi_\Omega = 0$, $m = 0,5$ и период последовательности $2N$.

Вывести графики последовательности и ее амплитудного спектра.

6С. Последовательность $x(t)$ (7.25):

$$x(t)|_{t=nT} = \frac{\sin \pi t}{\pi t}.$$

Задать частоту дискретизации $f_d = 2000$.

Вывести графики последовательности на интервале $t = nT \in [-500(N-1)T; 500(N-1)T]$ с шагом T и модуля ее ДПФ.

7С. Гауссов радиоимпульс (7.24):

$$x(n) = e^{-an^2} \cos(\hat{\omega}_1 n).$$

Задать $a = 0,0005$ и $\hat{\omega}_1 = \pi/12$.

Вывести графики последовательности на интервале $n \in [-3(N-1); 3(N-1)]$ и модуля ее ДПФ.

9.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения каждого пункта задания, включая копируемые из окна **Command Window** результаты вычислений (шрифт Courier New), созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**) и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

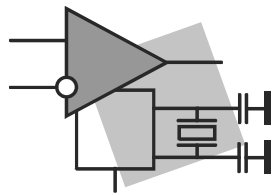
1. Запишите формулы ДПФ.
2. Что такое поворачивающий множитель?
3. Чему равно разрешение по частоте при вычислении ДПФ?
4. С чем связаны трудности прямого вычисления ДПФ по формуле (9.1)?
5. Что такое БПФ?

6. Каков порядок сложности алгоритмов ДПФ и БПФ Кули—Тьюки?
7. Назовите основные свойства ДПФ.
8. Дайте определение дискретной нормированной частоты.
9. Поясните смысл ДПФ для периодической последовательности.
10. Как с помощью ДПФ рассчитывается амплитудный и фазовый спектры периодической последовательности?
11. Поясните смысл ДПФ для конечной последовательности.
12. Как связаны значения абсолютных частот (в герцах [Гц] и радианах в секунду [рад/с]) и дискретных нормированных частот?
13. Поясните смысл приведенных критериев для выделения полезного сигнала из его аддитивной смеси с шумом.
14. Как задаются значения порогов в первом и втором критериях выделения полезного сигнала?
15. Как восстановить аналоговый периодический сигнал с финитным спектром по отсчетам ДПФ и на основе ряда Котельникова?
16. Как вычислить спектральную плотность в L точках на основе ДПФ при $L > N$?
17. Как определить разрешение по частоте в ДПФ при добавлении нулей к исходной последовательности?

9.7. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 11.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 5.

ГЛАВА 10



Дискретное преобразование Фурье (часть 2)

Цель работы: изучить применение ДПФ в условиях растекания спектра, для улучшения различения дискретных гармоник с близко расположенными частотами, быстрого вычисления линейных, круговых и секционированных сверток и овладеть соответствующими программными средствами MATLAB.

10.1. Краткая теоретическая справка

В этой главе мы продолжим знакомство с применением ДПФ для решения типовых задач ЦОС.

10.1.1. Растекание спектра

Растеканием спектра называют появление дополнительных составляющих в спектральном составе последовательности при вычислении ДПФ.

Ранее было определено условие (9.5), при котором гарантируется *точное* выделение гармоник последовательности $x(n)$ с частотами f_i .

Эффект растекания спектра наблюдается в том случае, если хотя бы для одной из дискретных гармоник, входящих в спектральный состав последовательности, с частотой f_i на интервале NT укладывается нецелое число периодов T_i и отношение (9.5):

$$P_i = \frac{NT}{T_i} = \frac{Nf_i}{f_d} \quad (10.1)$$

оказывается *не целым* числом, а частота гармоники f_i — не кратной периоду дискретизации по частоте $\Delta f = f_d/N$ (сравните с (9.5)):

$$f_i = P_i \Delta f.$$

Вследствие этого в периодическом продолжении гармоники с частотой f_i появятся разрывы (скачки) на границах периода последовательности, из-за которых спектр расширяется.

Для уменьшения эффекта растекания спектра (полностью он принципиально неустраним) применяют *весовые функции* (окна) — вещественные неотрицательные последовательности, максимальные в центре и монотонно спадающие к границам, что ослабляет влияние разрывов при периодическом продолжении последовательности. Для стандартных окон MATLAB значения $w(n)$ вычисляются автоматически по известным аналитическим формулам.

Выбор окна — не формализуемая задача, решаемая, как правило, простым перебором окон и/или изменением их параметров.

10.1.2. Улучшение различения дискретных гармоник с близко расположенными частотами

При вычислении ДПФ конечной последовательности длины N разрешение по частоте равно периоду дискретизации по частоте $\Delta f = f_d/N$ (см. разд. 9.1.3).

Для улучшения различения дискретных гармоник с близко расположенными частотами f_1 и f_2 , расстояние между которыми удовлетворяет условию:

$$\Delta f < |f_1 - f_2| < 2\Delta f, \quad (10.2)$$

исходную последовательность надо дополнить нулями до длины L :

$$L \geq \frac{f_d}{|f_1 - f_2| - \Delta f}. \quad (10.3)$$

Затем по L точкам восстановить спектральную плотность с периодом дискретизации по частоте $\tilde{\Delta f} = f_d/L$ и по графику модуля спектральной плотности определить *ближайшие пики* с максимальными амплитудами на частотах, близких к f_1 и f_2 . В общем случае эти частоты могут быть некратными периоду $\tilde{\Delta f}$, а следовательно, они будут определяться с погрешностью.

Для вычисления ДПФ с автоматическим добавлением нулей к последовательности предусмотрен специальный формат функции `fft`:

$\mathbf{x} = \text{fft}(\mathbf{x}, L)$

где $\mathbf{x}$ — N -точечная последовательность; L — длина последовательности, автоматически дополненной нулями; $\mathbf{x}$ — L -точечное ДПФ.

10.1.3. Вычисление линейных и круговых свертки с помощью ДПФ

Ранее (см. разд. 8.1.1) была рассмотрена функция `conv` для вычисления реакции по формуле свертки

$$y(n) = \sum_{m=0}^{L-1} h(n-m)x(m) = \sum_{m=0}^{L-1} h(m)x(n-m), \quad (10.4)$$

где $L = N_1 + N_2 - 1$ — длина *линейной* (апериодической) свертки, а N_1 и N_2 — длины импульсной характеристики $h(n)$ и воздействия $x(n)$.

Для сокращения объема вычислений линейная свертка (10.4) рассчитывается на основе круговой свертки с помощью ДПФ и ОДПФ с использованием алгоритмов БПФ и ОБПФ.

В общем случае *круговая свертка* периодических последовательностей $x_1(n)$ и $x_2(n)$ с одинаковым периодом N определяется как

$$x(n) = \sum_{m=0}^{N-1} x_1(m)x_2[(n-m) \bmod N] = \sum_{m=0}^{N-1} x_1[(n-m) \bmod N]x_2(m), \quad (10.5)$$

где записи $(n-m) \bmod N$ соответствует значение $(n-m)$ по модулю N .

Для вычисления круговой свертки (10.5) рассчитывается N -точечное ДПФ, равное произведению N -точечных ДПФ сворачиваемых последовательностей:

$$X(k) = X_1(k)X_2(k), \quad k = 0, 1, \dots, N-1, \quad (10.6)$$

а затем с помощью ОДПФ — последовательность $x(n)$.

В (10.4) для перехода от линейной свертки к круговой с периодом L последовательности $h(n)$ и $x(n)$ дополняют нулями до длины L :

$$y(n) = \sum_{m=0}^{L-1} \tilde{h}[(n-m) \bmod L] \tilde{x}(m) = \sum_{m=0}^{L-1} \tilde{h}(m) \tilde{x}[(n-m) \bmod L], \quad (10.7)$$

где $\tilde{h}(n)$ и $\tilde{x}(n)$ — дополненные нулями последовательности $h(n)$ и $x(n)$.

Рассчитывается L -точечное ДПФ круговой свертки (10.7):

$$Y(k) = \tilde{H}(k)\tilde{X}(k), \quad k = 0, 1, \dots, L-1, \quad (10.8)$$

а затем с помощью ОДПФ — реакция $y(n)$.

В MATLAB для вычисления линейной свертки на основе круговой свертки с использованием алгоритмов БПФ и ОБПФ предусмотрена функция:

y = fftfilt(h,x)

где h — импульсная характеристика длины N_1 ; x и y — векторы отсчетов воздействия и реакции одинаковой длины N_2 .

Отметим, что длина реакции равна длине *второго* аргумента N_2 (воздействия), в то время как при использовании функции `conv` для вычисления реакции (10.4) ее длина будет равна длине линейной свертки L (см. разд. 8.1.1).

10.1.4. Вычисление секционированных сверток с помощью ДПФ

Вычисление линейных сверток при большой длине воздействия N_2 производится методом перекрытия с накоплением с представлением последовательности $x(n)$ в виде коротких смежных секций длиной L , сравнимой с длиной импульсной характеристики N_1 . Линейная свертка формируется на основе коротких *секционированных сверток*, вычисляемых с помощью ДПФ и ОДПФ.

Для этого в MATLAB предусмотрен следующий формат функции `fftfilt` с использованием алгоритмов БПФ и ОБПФ:

y = fftfilt(h,x,L)

где h — вектор отсчетов импульсной характеристики длины N_1 ; x — вектор отсчетов воздействия длины N_2 ($N_2 \gg N_1$); L — длина L смежных секций; y — вектор отсчетов реакции длины N_2 .

10.2. Содержание лабораторной работы

Содержание работы связано с проверкой равенства Парсеваля, исследованием эффекта растекания спектра, улучшением различения близко расположенных гармоник и вычислением круговых, линейных и секционированных сверток с помощью ДПФ с использованием программных средств MATLAB.

10.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файла `lr_10` и function-файла `input_1`, которые хранятся на прилагаемом компакт-диске в папке `LAB_DSP\LAB_10`.

Перед выполнением работы необходимо сохранить путь к папке `LAB_10` по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 10.1 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$. Функция $N_{бр} \bmod M$ в записи исходных данных означает вычисление значения $N_{бр}$ по модулю M .

На прилагаемом компакт-диске в папке `Tables\Tables_10` хранятся табл. 10.1 исходных данных и пример ее заполнения для $N_{бр} = 1$.

Задание на лабораторную работу связано с вычислением ДПФ и включает в себя следующие пункты:

1. Проверка равенства Парсеваля:

$$\sum_{k=0}^{N-1} x^2(n) = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2. \quad (10.9)$$

Таблица 10.1. Таблица исходных данных

Пере- менная	Назначение	Значение	Идентификатор
$N_{бр}$	Номер бригады	$N_{бр}$	Nb =
N	Длина (период) последовательности	$N = 64$	N = 64
f_d	Частота дискретизации	$f_d = 2000(N_{бр} \bmod 5 + 1)$	Fs =
A_1	Амплитуды дискрет- ных гармоник	$A_1 = 1 + 0,01N_{бр}$	A1 =
A_2		$A_2 = 2A_1$	A2 =
f_1	Частоты дискретных гармоник	$f_1 = f_d / 4$	f1 =
f_2		$f_2 = 1,5f_1$	f2 =
M	Период последова- тельности	$M = 71$	M = 71
f_{11}	Частоты дискретных гармоник	$f_{11} = 1,1f_1$	f1_1 =
f_{21}		$f_{21} = 1,07f_2$	f2_1 =
f_{12}	Частоты дискретных гармоник	$f_{12} = 1,05f_1$	f1_2 =
f_{22}		$f_{22} = f_{12} + 1,1\Delta f_1,$ где $\Delta f = \frac{f_d}{N}$	f2_2 = f1_2+1.1*Fs/N
$x_3(n)$	Периодическая последовательность (период)	$x_3(n) = N_{бр}[0,1; 0,2; 0,3; 0,4; 0,5]$	Вектор x3 = [...]
$x_4(n)$	Периодическая последовательность (период)	$x_4(n) = N_{бр}[0,5; 0,4; 0,3; 0,2; 0,1]$	Вектор x4 = [...]
$x_5(n)$	Конечная последовательность	$x_5(n) = N_{бр}[0,1; 0,2; 0,3]$	Вектор x5 = [...]
$x_6(n)$	Конечная последовательность	$x_6(n) = N_{бр}[0,3; 0,2; 0,1; 0,2; 0,3]$	Вектор x6 = [...]
b_0 b_1 b_2	Коэффициенты числителя передаточной функции	$b_0 = 0,5 + 0,02N_{бр}$ $b_1 = b_0(-1)^{N_{бр}+1}(0,9822 + 0,0178N_{бр})$ $b_2 = b_0[0,8 + 0,2(N_{бр} \bmod 5)]$	Вектор b = [...]

Таблица 10.1 (окончание)

Пере- менная	Назначение	Значение	Идентификатор
a_0 a_1 a_2	Коэффициенты зна- менателя передаточ- ной функции	$a_0 = 1$ $a_1 = (-1)^{N_{\text{бр}}} (0,7778 + 0,025N_{\text{бр}})$ $a_2 = 0,64 + 0,006N_{\text{бр}}$	Вектор $a = [\dots]$
N_1	Длина ИХ	$N_1 = N_{\text{бр}} \bmod 10 + 20$	$N1 =$
N_2	Длина воздействия	$N_2 = N_{\text{бр}} \bmod 10 + 30$	$N2 =$
N_3	Длина воздействия	$N_3 = N_{\text{бр}} \bmod 10 + 200$	$N3 =$

Проверить для периодической последовательности (идентификатор $\times$) с перио-
дом N :

$$x(n) = A_1 \cos(2\pi f_1 nT) + A_2 \cos(2\pi f_2 nT),$$

(10.10)

используя ее тождественное представление в виде:

$$x(n) = A_1 \cos\left(\frac{2\pi f_1}{f_d} n\right) + A_2 \cos\left(\frac{2\pi f_2}{f_d} n\right) = A_1 \cos(\hat{\omega}_1 n) + A_2 \cos(\hat{\omega}_2 n).$$

(10.11)

Левая часть равенства (10.9) — идентификатор E1 , правая — E2 .

Пояснить смысл равенства Парсеваля.

2. Исследование эффекта растекания спектра для одной дискретной гармоники.

Выполнить для последовательности

$$\tilde{x}(n) = A_1 \cos\left(\frac{2\pi f_1 n}{f_d}\right) = A_1 \cos(\hat{\omega}_1 n)$$

(10.12)

с двумя значениями периода:

- N — идентификатор последовательности $\times\_N$;
- M — идентификатор последовательности $\times\_M$.

Вывести:

- соответствующие значения P (10.1) для частоты дискретной гармоники f_1 — идентификаторы P_N и P_M .

Во втором случае в отношение (10.1) следует подставить $N = M$;

- графики амплитудных спектров (идентификаторы MOD_N и MOD_M) в шкале дискретных нормированных частот.

Пояснить:

- с какой целью определяется значение P ;
- в каком случае и почему наблюдается растекание спектра.

3. Исследование возможности уменьшения растекания спектра с помощью окна.

Применить окно Хэмминга (идентификатор `win_m`) для последовательности $\tilde{x}(n)$ (10.12) в условиях растекания спектра.

Вывести графики амплитудных спектров до и после применения окна (идентификаторы `MOD_m` и `MODW_m`) в шкале дискретных нормированных частот.

Пояснить, что изменилось в результате применения окна.

4. Исследование эффекта растекания спектра для суммы двух дискретных гармоник.

Для периодической последовательности $x(n)$ (10.11) с периодом N задать значения частот:

$$\begin{aligned} f_1 &= f_{11}; \\ f_2 &= f_{21}, \end{aligned}$$

и для новой последовательности (идентификатор `x1`) вывести:

- значения P (10.1) для частот дискретных гармоник f_{11} и f_{21} — идентификаторы `P1_1` и `P2_1`;
- применить окно Хэмминга (идентификатор `win_N`) в условиях растекания спектра;
- вывести графики амплитудных спектров до и после применения окна (идентификаторы `MOD1` и `MODW1`) в шкале дискретных нормированных частот.

Пояснить причину растекания спектра и цель применения окна.

5. Улучшение различения дискретных гармоник с близко расположенными частотами.

Для конечной последовательности $x(n)$ (10.11) длины N задать значения частот:

$$\begin{aligned} f_1 &= f_{12}; \\ f_2 &= f_{22}, \end{aligned}$$

и для новой последовательности (идентификатор `x2`) вывести:

- разрешение по частоте $\Delta f = \frac{f_d}{N}$ (идентификатор `Delta_N`);
- расстояние между частотами $|f_{12} - f_{22}|$ (идентификатор `Delta_f`);
- требуемую длину L (10.3) (идентификатор `L`);

- период дискретизации по частоте $\Delta\tilde{f} = \frac{f_d}{L}$ (идентификатор `Delta_L`);
- график модуля спектральной плотности, восстановленной по L отсчетам ДПФ, с помощью функции `plot`, и одновременно — L -точечное ДПФ (идентификатор `MOD2_L`) пунктиром с помощью функции `stem`;
- частоты ближайших пиков в шкалах дискретных нормированных частот (идентификаторы `k_1` и `k_2`) и абсолютных частот (идентификаторы `f_1` и `f_2`) в основной полосе частот $k \in [0; \text{int}(L/2) - 1]$.

Частоты *первого* пика определяются с помощью функции `max`.

Для определения частоты *второго* пика следует найти пики справа и слева от первого пика на периоде дискретизации $\Delta f = f_d/N$ с помощью функции `max` и выбрать наибольший из них.

Справа и слева от первого пика на интервале $\Delta f = f_d/N$ расположено $K = \text{int}(\Delta f / \Delta\tilde{f}) = \text{int}(L/N)$ отсчетов (идентификатор `K`).

По графику спектральной плотности, используя кнопку **Zoom in** на панели инструментов, определить частоты ближайших пиков и сравнить их с выведенными значениями.

Пояснить:

- соответствуют ли близко расположенные частоты условию (10.2);
- соответствует ли выведенная длина L условию (10.3);
- с какой погрешностью определены частоты и причину погрешности.

6. Вычисление круговой свертки.

Вычислить круговую свертку $y_{34}(n)$ (идентификатор `y34`) периодических последовательностей $x_3(n)$ и $x_4(n)$ с помощью функций `fft` и `ifft` и вывести графики трех периодов последовательностей и свертки, используя функцию `repmat`.

Записать формулу круговой свертки и пояснить алгоритм ее вычисления с помощью ДПФ.

7. Вычисление линейной свертки.

Вычислить линейную свертку $y_{56}(n)$ конечных последовательностей $x_5(n)$ и $x_6(n)$ двумя способами:

- с помощью функции `conv` (идентификатор `y56_1`);
- с помощью функции `fftfilt` (идентификатор `y56_2`).

Вывести графики последовательностей $x_5(n)$, $x_6(n)$ и свертки $y_{56}(n)$, вычисленной двумя способами, в одинаковом диапазоне по оси абсцисс с помощью функции `xlim([0 MAX-1])`, где `MAX` — максимальная длина свертки.

Записать формулу линейной свертки и пояснить алгоритм ее вычисления с помощью ДПФ.

8. Вычисление реакции ЛДС по формуле свертки.

В качестве воздействия $x_7(n)$ (идентификатор `x7`) выбрать дискретный прямоугольный импульс длины N_2 :

$$x_7(n) = \begin{cases} 1, & 0 \leq n < \text{int}(N_2/2); \\ 0, & \text{int}(N_2/2) \leq n \leq (N_2 - 1). \end{cases} \quad (10.13)$$

Для моделирования воздействия (10.13) использовать function-файл `input_1` (см. разд. 10.4.1).

Для вычисления импульсной характеристики (ИХ) $h(n)$ длины N_1 с помощью функции `impz` использовать коэффициенты передаточной функции рекурсивного звена 2-го порядка b_i и a_k .

Вычислить реакцию $y_7(n)$ по формуле свертки (10.4) двумя способами:

- с помощью функции `conv` (идентификатор `y7_1`);
- с помощью функции `fftfilt` (идентификатор `y7_2`).

Вывести графики ИХ, воздействия и реакции, вычисленной двумя способами, в одинаковом диапазоне по оси абсцисс с помощью функции `xlim([0 L-1])`, где L — длина свертки, вычисленной с помощью функции `conv`.

Записать формулу свертки.

Пояснить:

- преимущество вычисления реакции по формуле свертки с помощью ДПФ;
- чему равна длина реакции, вычисленной первым и вторым способами;
- в каком случае длину реакции необходимо ограничить до длины воздействия.

9. Вычисление реакции ЛДС методом перекрытия с накоплением.

В качестве воздействия $x_8(n)$ (идентификатор `x8`) выбрать прямоугольный импульс $x_7(n)$ (10.15) длины N_3 .

Вычислить реакцию $y_8(n)$ по формуле свертки двумя способами:

- с помощью функции `fftfilt` без перекрытия (идентификатор `y8_1`);
- с помощью функции `fftfilt` методом перекрытия с накоплением (идентификатор `y8_2`), задавая длину секции равной длине ИХ N_1 .

Вывести графики ИХ, воздействия и реакций в одинаковом диапазоне по оси абсцисс с помощью функции `xlim([0 N3-1])`, где $N3$ — длина воздействия и реакции.

Пояснить, в каком случае целесообразно вычислять реакцию методом перекрытия с накоплением.

10.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должна быть представлена табл. 10.1 исходных данных для своего номера бригады $N_{бр}$.

Для *запуска* лабораторной работы необходимо обратиться к script-файлу `lr_10` по его имени:

```
>> lr_10
```

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш <Ctrl>+<Break>.

При выполнении script-файла текущие окна с графиками *не закрывать*.

Листинг script-файла `lr_10` имеет вид:

```
>> type lr_10
script
clc
clear
disp('% ЛР №10. ДИСКРЕТНОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ (часть 2)')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
N = input('N = ');             % ДЛИНА (ПЕРИОД) ПОСЛЕДОВАТЕЛЬНОСТИ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
A1 = input('A1 = ');           % АМПЛИТУДЫ ДИСКРЕТНЫХ ГАРМОНИК
A2 = input('A2 = ');
f1 = input('f1 = ');           % ЧАСТОТЫ ДИСКРЕТНЫХ ГАРМОНИК (Гц)
f2 = input('f2 = ');
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.1. ПРОВЕРКА РАВЕНСТВА ПАРСЕВАЛЯ')
n = 0:(N-1);                  % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
k = 0:(N-1);                  % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
```

```

w1 = 2*pi*f1/Fs; w2 = 2*pi*f2/Fs; % НОРМИРОВАННЫЕ ЧАСТОТЫ ДИСКРЕТНЫХ ГАРМОНИК
(РАД)
x = A1*cos(w1*n)+A2*cos(w2*n); % ПОСЛЕДОВАТЕЛЬНОСТЬ (ПЕРИОД N)
X = fft(x); % ДПФ ПОСЛЕДОВАТЕЛЬНОСТИ
E1 = sum(x.^2); % ЭНЕРГИЯ ПОСЛЕДОВАТЕЛЬНОСТИ, ВЫЧИСЛЕННАЯ ПО ЕЕ
ОТСЧЕТАМ
E2 = (1/N)*sum(abs(X).^2); % ЭНЕРГИЯ ПОСЛЕДОВАТЕЛЬНОСТИ, ВЫЧИСЛЕННАЯ ПО
ОТСЧЕТАМ ДПФ
disp('%')
disp('%')
disp('% Для вывода левой (E1) и правой (E2) частей РАВЕНСТВА ПАРСЕВАЛЯ нажмите
<ENTER>')
pause
disp('%')
disp('%')
disp([' E1 = ',num2str(E1),' E2 = ' num2str(E2)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ИССЛЕДОВАНИЕ ЭФФЕКТА РАСТЕКАНИЯ СПЕКТРА ДЛЯ ОДНОЙ ДИСКРЕТНОЙ
ГАРМОНИКИ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
M = input('M = '); % ПЕРИОД ПОСЛЕДОВАТЕЛЬНОСТИ M
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
n = 0:(N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ (ПЕРИОД N)
k = 0:(N-1); % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА (ПЕРИОД N)
w1 = 2*pi*f1/Fs; % НОРМИРОВАННАЯ ЧАСТОТА (РАД)
x_N = A1*cos(w1*n); % ПОСЛЕДОВАТЕЛЬНОСТЬ (ПЕРИОД N)
X_N = fft(x_N); % ДПФ ПОСЛЕДОВАТЕЛЬНОСТИ (ПЕРИОД N)
MOD_N = (2/N)*abs(X_N); % АМПЛИТУДНЫЙ СПЕКТР ПОСЛЕДОВАТЕЛЬНОСТИ (ПЕРИОД N)
MOD_N(1) = (1/N)*abs(X_N(1));
n1 = 0:(M-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ (ПЕРИОД M)
k1 = 0:(M-1); % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА (ПЕРИОД M)
x_M = A1*cos(w1*n1); % ПОСЛЕДОВАТЕЛЬНОСТЬ (ПЕРИОД M)
X_M = fft(x_M); % ДПФ ПОСЛЕДОВАТЕЛЬНОСТИ (ПЕРИОД M)
MOD_M = (2/M)*abs(X_M); % АМПЛИТУДНЫЙ СПЕКТР ПОСЛЕДОВАТЕЛЬНОСТИ (ПЕРИОД M)
MOD_M(1) = (1/M)*abs(X_M(1));

```

```

P_N = N*f1/Fs;           % ЧИСЛО ПЕРИОДОВ ДИСКРЕТНОЙ ГАРМОНИКИ С ЧАСТОТОЙ f1
НА ПЕРИОДЕ ПОСЛЕДОВАТЕЛЬНОСТИ N
P_M = M*f1/Fs;           % ЧИСЛО ПЕРИОДОВ ДИСКРЕТНОЙ ГАРМОНИКИ С ЧАСТОТОЙ f1
НА ПЕРИОДЕ ПОСЛЕДОВАТЕЛЬНОСТИ M
disp('%')
disp('%')
disp('% Для вывода ЧИСЛА ПЕРИОДОВ дискретной гармонии С ЧАСТОТОЙ f1 нажмите
<ENTER>')
pause
disp('%')
disp('%')
disp(['N = ',num2str(N),' --> P_N = ' num2str(P_N)])
disp(['M = ',num2str(M),' --> P_M = ' num2str(P_M)])
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ нажмите <ENTER>')
pause
figure('Name','Amplitude Spectrum','NumberTitle','off')
subplot(2,1,1), stem(k,MOD_N,'MarkerSize',3), grid, xlabel('k')
title(strcat(['Amplitude Spectrum of the Periodic Sequence N = ',num2str(N)]))
subplot(2,1,2), stem(k1,MOD_M,'MarkerSize',3), grid, xlabel('k')
title(strcat(['Amplitude Spectrum of the Periodic Sequence M = ',num2str(M)]))
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ УМЕНЬШЕНИЯ РАСТЕКАНИЯ СПЕКТРА С ПОМОЩЬЮ
ОКНА')
win_M = hamming(M)';      % ОКНО ХЭММИНГА — ВЕКТОР-СТОЛБЕЦ ДЛИНЫ M
xw_M = x_M.*win_M;        % ПОСЛЕДОВАТЕЛЬНОСТЬ, ВЗВЕШЕННАЯ ОКНОМ
XW_M = fft(xw_M);         % ДПФ ВЗВЕШЕННОЙ ПОСЛЕДОВАТЕЛЬНОСТИ
MODW_M = (2/M)*abs(XW_M); % АМПЛИТУДНЫЙ СПЕКТР ВЗВЕШЕННОЙ ПОСЛЕДОВАТЕЛЬНОСТИ
MODW_M(1) = (1/M)*abs(XW_M(1));
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ ДО и ПОСЛЕ применения ОКНА
нажмите <ENTER>')
pause
figure('Name','Reducing Spectrum Leakage with the help of Window
Functions','NumberTitle','off')
subplot(2,1,1), stem(k1,MOD_M,'MarkerSize',3), grid, xlabel('k')
title(strcat(['Amplitude spectrum without windowing M = ',num2str(M)]))
subplot(2,1,2), stem(k1,MODW_M,'MarkerSize',3), grid, xlabel('k')
title(strcat(['Amplitude spectrum with Hamming Window M = ',num2str(M)]))
disp('%')
disp('%')

```

```

disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. ИССЛЕДОВАНИЕ ЭФФЕКТА РАСТЕКАНИЯ СПЕКТРА ДЛЯ СУММЫ ДВУХ ДИСКРЕТНЫХ  
ГАРМОНИК')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
f1_1 = input('f1_1 = ');    % ЧАСТОТЫ ДИСКРЕТНЫХ ГАРМОНИК (Гц)
f2_1 = input('f2_1 = ');
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
n = 0:(N-1);                % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
k = 0:(N-1);                % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
w1_1 = 2*pi*f1_1/Fs; w2_1 = 2*pi*f2_1/Fs; % НОРМИРОВАННЫЕ ЧАСТОТЫ ДИСКРЕТНЫХ  
ГАРМОНИК (РАД)
x1 = A1*cos(w1_1*n)+A2*cos(w2_1*n); % ПОСЛЕДОВАТЕЛЬНОСТЬ (ПЕРИОД N)
X1 = fft(x1);               % ДПФ ПОСЛЕДОВАТЕЛЬНОСТИ (ПЕРИОД N)
MOD1 = (2/N)*abs(X1);       % АМПЛИТУДНЫЙ СПЕКТР ПОСЛЕДОВАТЕЛЬНОСТИ
MOD1(1) = (1/N)*abs(X1(1));
P1_1 = N*f1_1/Fs;           % ЧИСЛО ПЕРИОДОВ ДИСКРЕТНОЙ ГАРМОНИКИ С ЧАСТОТОЙ f1_1  
НА ПЕРИОДЕ ПОСЛЕДОВАТЕЛЬНОСТИ N
P2_1 = N*f2_1/Fs;           % ЧИСЛО ПЕРИОДОВ ДИСКРЕТНОЙ ГАРМОНИКИ С ЧАСТОТОЙ f2_1  
НА ПЕРИОДЕ ПОСЛЕДОВАТЕЛЬНОСТИ N
disp('%')
disp('%')
disp('% Для вывода ЧИСЛА ПЕРИОДОВ дискретных гармоник С ЧАСТОТАМИ f1_1 и f2_1  
нажмите <ENTER>')
pause
disp('%')
disp('%')
disp(['      f1_1 = ',num2str(f1_1),' --> P1_1 = ' num2str(P1_1)])
disp(['      f2_1 = ',num2str(f2_1),' --> P2_1 = ' num2str(P2_1)])
win_N = hamming(N);          % ОКНО ХЭММИНГА – ВЕКТОР-СТОЛБЕЦ ДЛИНЫ N
xw1 = x1.*win_N;            % ПОСЛЕДОВАТЕЛЬНОСТЬ, ВЗВЕШЕННАЯ ОКНОМ (ПЕРИОД N)
XW1 = fft(xw1);             % ДПФ ВЗВЕШЕННОЙ ПОСЛЕДОВАТЕЛЬНОСТИ (ПЕРИОД N)
MODW1 = (2/N)*abs(XW1);      % АМПЛИТУДНЫЙ СПЕКТР ПОСЛЕДОВАТЕЛЬНОСТИ
MODW1(1) = (1/M)*abs(XW1(1));
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ ДО и ПОСЛЕ применения ОКНА  
нажмите <ENTER>')

```

```

pause
figure('Name','Reducing Spectrum Leakage with the help of Window
Functions','NumberTitle','off')
subplot(2,1,1), stem(k,MOD1,'MarkerSize',3), grid, xlabel('k')
title(strcat(['Amplitude spectrum without windowing N = ',num2str(N)]))
subplot(2,1,2), stem(k,MODW1,'MarkerSize',3), grid, xlabel('k')
title(strcat(['Amplitude spectrum with Hamming Window N = ',num2str(N)]))
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. УЛУЧШЕНИЕ РАЗЛИЧИЕНИЯ ДИСКРЕТНЫХ ГАРМОНИК С БЛИЗКО РАСПОЛОЖЕННЫМИ
ЧАСТОТАМИ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
f1_2 = input('f1_2 = ');           % ЧАСТОТЫ ДИСКРЕТНЫХ ГАРМОНИК (Гц)
f2_2 = input('f2_2 = ');
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для вывода ПЕРИОДА ПОСЛЕДОВАТЕЛЬНОСТИ и')
disp('% ЧАСТОТ ГАРМОНИК нажмите <ENTER>')
disp('%')
disp('%')
disp(['      N = ',num2str(N)])
disp(['      f1_2 = ',num2str(f1_2),'      f2_2 = ' num2str(f2_2)])
Delta_N = Fs/N;           % РАЗРЕШЕНИЕ ПО ЧАСТОТЕ
Delta_f = abs(f1_2-f2_2); % РАССТОЯНИЕ МЕЖДУ ЧАСТОТАМИ
L = ceil(Fs/(Delta_f-Delta_N)); % ВЫБРАННАЯ ДЛИНА L
Delta_L = Fs/L;           % ПЕРИОД ДИСКРЕТИЗАЦИИ ПО ЧАСТОТЕ ПРИ ДЛИНЕ L
disp('%')
disp('%')
disp('% Для вывода РАЗРЕШЕНИЯ ПО ЧАСТОТЕ Delta_N,')
disp('% РАССТОЯНИЯ между ЧАСТОТАМИ Delta_f,')
disp('% ДЛИНЫ L последовательности')
disp('% и ПЕРИОДА ДИСКРЕТИЗАЦИИ ПО ЧАСТОТЕ Delta_L нажмите <ENTER>')
pause
disp('%')

```



```

disp('%')
disp(['      Delta_N = ',num2str(Delta_N)])
disp(['      Delta_f = ',num2str(Delta_f)])
disp(['      L = ',num2str(L)])
disp(['      Delta_L = ',num2str(Delta_L)])
disp('%')
disp('%')
n = 0:(N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
w1_2 = 2*pi*f1_2/Fs; w2_2 = 2*pi*f2_2/Fs; % НОРМИРОВАННЫЕ ЧАСТОТЫ
x2 = A1*cos(w1_2*n)+A2*cos(w2_2*n); % КОНЕЧНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ
X2 = fft(x2); % ДПФ КОНЕЧНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ДЛИНЫ N
MOD2 = abs(X2); % МОДУЛЬ ДПФ
X2_L = fft(x2,L); % ДПФ КОНЕЧНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ, ДОПОЛНЕННОЙ
    НУЛЯМИ ДО ДЛИНЫ L
MOD2_L = abs(X2_L); % МОДУЛЬ ДПФ
disp('% Для вывода ГРАФИКОВ N-ТОЧЕЧНОГО ДПФ и МОДУЛЯ СПЕКТРАЛЬНОЙ')
disp('% ПЛОТНОСТИ, ВОССТАНОВЛЕННОЙ ПО L ТОЧКАМ, нажмите <ENTER>')
pause
k = 0:(N-1); % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА ПРИ ДЛИНЕ N
k1 = 0:(L-1); % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА ПРИ ДЛИНЕ L
figure('Name','Discrete Harmonic Signal with Close Frequencies','NumberTitle',
    'off')
subplot(2,1,1), stem(k,MOD2), grid, xlabel('k')
title(strcat(['DFT Modulus N = ',num2str(N)]))
subplot(2,1,2), plot(k1,MOD2_L,'r','MarkerSize',3, 'Linewidth',2)
grid, hold on, stem(k1,MOD2_L,':'), xlabel('k')
title(strcat(['Spectral Density Modulus L = ',num2str(L)]))
L_2 = ceil(L/2); % ОСНОВНАЯ ПОЛОСА ЧАСТОТ L/2
[MODm m]= max(MOD2_L(1:(L_2))); % МАКСИМУМ MODm И ИНДЕКС m ВЕКТОРА MOD2_L
    (ПЕРВЫЙ ПИК)
k_1 = (m-1); f_1 = k_1*Delta_L; % ДИСКРЕТНАЯ НОРМИРОВАННАЯ И АБСОЛЮТНАЯ
    (Гц) ЧАСТОТЫ ПЕРВОГО ПИКА
K = ceil(L/N); % КОЛИЧЕСТВО ОТСЧЕТОВ НА ПЕРИОДЕ ДИСКРЕТИЗАЦИИ Fs/N
K1 = m+K; K2 = m+2*K-1; % НИЖНЯЯ K1 и ВЕРХНЯЯ K2 ГРАНИЦЫ ИНТЕРВАЛА
    ПРИ ПОИСКЕ ВТОРОГО ПИКА СПРАВА
[MODm1 m1]= max(MOD2_L(K1:K2)); % МАКСИМУМ MODm1 И ИНДЕКС m1 МОДУЛЯ ДПФ
    MOD2_L НА ИНТЕРВАЛЕ [K1 K2]
K3 = m-(2*K-1); K4 = m-K; % НИЖНЯЯ K3 и ВЕРХНЯЯ K4 ГРАНИЦЫ ИНТЕРВАЛА
    ПРИ ПОИСКЕ ВТОРОГО ПИКА СЛЕВА
[MODm2 m2]= max(MOD2_L(K3:K4)); % МАКСИМУМ MODm2 И ИНДЕКС m2 МОДУЛЯ ДПФ
    MOD2_L НА ИНТЕРВАЛЕ [K3 K4]
if (MODm1>MODm2)
    k_2 = (K1+m1-1)-1; f_2 = k_2*Delta_L; % ДИСКРЕТНАЯ НОРМИРОВАННАЯ
    И АБСОЛЮТНАЯ (Гц) ЧАСТОТЫ ВТОРОГО ПИКА, ЕСЛИ ОН СПРАВА ОТ ПЕРВОГО
else
    k_2 = (K3+m2-1)-1; f_2 = k_2*Delta_L; % ДИСКРЕТНАЯ НОРМИРОВАННАЯ
    И АБСОЛЮТНАЯ (Гц) ЧАСТОТЫ ВТОРОГО ПИКА, ЕСЛИ ОН СЛЕВА ОТ ПЕРВОГО
end

```

```

disp('%')
disp('%')
disp('% Для вывода ЧАСТОТ ГАРМОНИК нажмите <ENTER>')
pause
disp('%')
disp('%')
disp(['      k_1 = ',num2str(k_1),'      f_1 = ' num2str(f_1)])
disp(['      k_2 = ',num2str(k_2),'      f_2 = ' num2str(f_2)])
disp('%')
disp('%')
disp('% Определите ЧАСТОТЫ ГАРМОНИК по ГРАФИКУ')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ВЫЧИСЛЕНИЕ КРУГОВОЙ СВЕРТКИ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
x3 = input('x3 = ');           % ПЕРВАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ
x4 = input('x4 = ');           % ВТОРАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
y34 = ifft(fft(x3).*fft(x4)); % КРУГОВАЯ СВЕРТКА ПОСЛЕДОВАТЕЛЬНОСТЕЙ
L34 = length(y34);           % ПЕРИОД КРУГОВОЙ СВЕРТКИ
disp('%')
disp('%')
disp('% Для вывода графиков ПОСЛЕДОВАТЕЛЬНОСТЕЙ и КРУГОВОЙ свертки (3 периода)
нажмите <ENTER>')
pause
figure('Name','Sequences x3, x4, y34','NumberTitle','off')
subplot(3,1,1), stem((0:3*L34-1),...
repmat(x3,1,3),'fill','Linewidth',2,'MarkerSize',3), grid
xlabel('n'), title('Periodic Sequence x3(n)')
subplot(3,1,2), stem((0:3*L34-1), repmat(x4,1,3),'fill',
'Linewidth',2,'MarkerSize',3), grid
xlabel('n'), title('Periodic Sequence x4(n)')
subplot(3,1,3), stem((0:3*L34-1), repmat(y34,1,3),'fill',
'Linewidth',2,'MarkerSize',3), grid, xlabel('n')
title('Periodic Sequence y34(n) - Convolution with FFT and IFFT')

```

```

disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. ВЫЧИСЛЕНИЕ ЛИНЕЙНОЙ СВЕРТКИ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
x5 = input('x5 = ');           % ПЕРВАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ
x6 = input('x6 = ');           % ВТОРАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
y56_1 = conv(x5,x6);           % ЛИНЕЙНАЯ СВЕРТКА, ВЫЧИСЛЕННАЯ С ПОМОЩЬЮ ФУНКЦИИ
conv
y56_2 = fftfilt(x5,x6);       % ЛИНЕЙНАЯ СВЕРТКА, ВЫЧИСЛЕННАЯ С ПОМОЩЬЮ ФУНКЦИИ
fftfilt
MAX = max([length(y56_1) length(y56_2)]); % МАКСИМАЛЬНАЯ ДЛИНА СВЕРТКИ
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ПОСЛЕДОВАТЕЛЬНОСТЕЙ и ЛИНЕЙНОЙ свертки нажмите
<ENTER>')
pause
figure('Name','Sequences x5, x6, y56_1, y56_2','NumberTitle', 'off')
subplot(4,1,1)
stem((0:length(x5)-1),x5,'fill','Linewidth',2,'MarkerSize',3)
grid, xlabel('n'), title('Sequence x5(n)'), xlim([0 MAX-1])
subplot(4,1,2)
stem((0:length(x6)-1),x6,'fill','Linewidth',2,'MarkerSize',3)
grid, xlabel('n'), title('Sequence x6(n)'), xlim([0 MAX-1])
subplot(4,1,3)
stem((0:length(y56_1)-1),y56_1,'fill','Linewidth',2,'MarkerSize',3)
grid, xlabel('n'), title('Sequence y56(n) - Convolution'), xlim([0 MAX-1])
subplot(4,1,4)
stem((0:length(y56_2)-1),y56_2,'fill','Linewidth',2,'MarkerSize',3)
grid, xlabel('n'), title('Sequence y56(n) - Convolution with FFT and IFFT'),
xlim([0 MAX-1])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause

```

```

disp('%')
disp('%')
disp('% п.8. ВЫЧИСЛЕНИЕ РЕАКЦИИ ЛДС ПО ФОРМУЛЕ СВЕРТКИ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
b = input('b = ');           % КОЭФФИЦИЕНТЫ ЧИСЛИТЕЛЯ ПЕРЕДАТОЧНОЙ ФУНКЦИИ
a = input('a = ');           % КОЭФФИЦИЕНТЫ ЗНАМЕНАТЕЛЯ ПЕРЕДАТОЧНОЙ ФУНКЦИИ
N1 = input('N1 = ');         % ДЛИНА ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ
N2 = input('N2 = ');         % ДЛИНА ВОЗДЕЙСТВИЯ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
h = impz(b,a,N1);           % ИМПУЛЬСНАЯ ХАРАКТЕРИСТИКА
x7 = input_1(N2);           % ВОЗДЕЙСТВИЕ
y7_1 = conv(x7,h);           % РЕАКЦИЯ, ВЫЧИСЛЕННАЯ С ПОМОЩЬЮ ФУНКЦИИ conv
y7_2 = fftfilt(h,x7);        % РЕАКЦИЯ, ВЫЧИСЛЕННАЯ С ПОМОЩЬЮ ФУНКЦИИ fftfilt
L=N1+N2-1;                  % ДЛИНА СВЕРТКИ, ВЫЧИСЛЕННОЙ С ПОМОЩЬЮ ФУНКЦИИ conv
disp('%')
disp('%')
disp('% Для вывода графиков ИХ, ВОЗДЕЙСТВИЯ и РЕАКЦИИ нажмите <ENTER>')
pause
figure('Name','Impulse Response, Input and Output Signals','NumberTitle',
'off')
subplot(4,1,1)
stem(0:length(h)-1,h,'Linewidth',2,'MarkerSize',3), grid,
xlabel('n'), title('Impulse Response h(n)'), xlim([0 L-1])
subplot(4,1,2)
stem(0:length(x7)-1,x7,'Linewidth',2,'MarkerSize',3), grid
xlabel('n'), title('Input Signal x7(n)'), xlim([0 L-1])
subplot(4,1,3)
stem(0:length(y7_1)-1,y7_1,'Linewidth',2,'MarkerSize',3),
grid
xlabel('n'), title('Output Signal y7(n) - Convolution'), xlim([0 L-1])
subplot(4,1,4)
stem(0:length(y7_2)-1,y7_2,'Linewidth',2,'MarkerSize',3), grid
xlabel('n'), title('Output Signal y7(n) - Convolution with FFT and IFFT'),
xlim([0 L-1])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')

```

```

disp('%')
disp('% п.9. ВЫЧИСЛЕНИЕ РЕАКЦИИ ЛДС МЕТОДОМ ПЕРЕКРЫТИЯ С НАКОПЛЕНИЕМ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
N3 = input('N3 = '); % ДЛИНА ВОЗДЕЙСТВИЯ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
x8 = input_1(N3); % ВОЗДЕЙСТВИЕ
y8_1 = fftfilt(h,x8); % РЕАКЦИЯ, ВЫЧИСЛЕННАЯ С ПОМОЩЬЮ ФУНКЦИИ fftfilt
y8_2 = fftfilt(h,x8,N1); % РЕАКЦИЯ, ВЫЧИСЛЕННАЯ С ПОМОЩЬЮ ФУНКЦИИ fftfilt
МЕТОДОМ НАКОПЛЕНИЯ С ПЕРЕКРЫТИЕМ
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ВОЗДЕЙСТВИЯ и РЕАКЦИИ нажмите <ENTER>')
pause
figure('Name','Impulse Response, Input and Output Signals – Overlap-add
method','NumberTitle','off')
subplot(4,1,1)
stem(0:length(h)-1,h,'MarkerSize',3), grid
xlabel('n'), title('Impulse Response h(n)'), xlim([0 N3-1])
subplot(4,1,2), stem(0:length(x8)-1,x8,'MarkerSize',3), grid
xlabel('n'), title('Input Signal x8(n)')
subplot(4,1,3),stem(0:length(y8_1)-1,y8_1,'MarkerSize',3), grid
xlabel('n')
title('Output Signal y8(n) – Convolution with FFT and IFFT')
subplot(4,1,4), stem(0:length(y8_2)-1,y8_2,'MarkerSize',3), grid
xlabel('n')
title('Output Signal y8(n) – Convolution with Overlap-add method')
disp('%')
disp('%')
disp('% РАБОТА ЗАВЕРШЕНА')

```

10.4.1. Используемые внешние функции

В script-файле `lr_10` используется внешняя функция `input_1`, предназначенная для моделирования воздействия (10.15), совпадающего с воздействием (8.18) (см. разд. 8.4.1).

10.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файлов с применением ДПФ в различных приложениях и исходных данных из табл. 10.1 для своего номера бригады $N_{бр}$.

Пункты самостоятельного задания включают в себя:

1С. Проверку равенства Парсевалю для конечной последовательности (10.11) с разными длинами N и M .

2С. Исследование эффекта растекания спектра

Выполнить для периодической последовательности (10.12) с произвольными значениями периода N и частоты f_1 .

Вывести значение P (10.1) и сообщение о наличии/отсутствии эффекта растекания спектра.

При наличии эффекта растекания спектра применить требуемое окно и вывести графики амплитудного спектра последовательности до и после его применения.

Варианты весовых функций (окон) представить в виде векторов-столбцов:

$WIN(:, 1) = \text{hamming}(N);$

$WIN(:, 2) = \text{hanning}(N);$

и т. д. Имена окон найти в GUI Window Design and Analysis Tool (команда `wintool`).

3С. Улучшение различения дискретных гармоник с близко расположенными частотами.

Выполнить для конечной последовательности $x(n)$ (10.11) длины $2N$ при значениях частот $f_{12} = 1,1f_1$ и $f_{22} = 1,15f_1$.

4С. Вычисление круговой свертки с помощью ДПФ и ОДПФ.

Выполнить для последовательностей $x_1(n) = [0; 0,25; 0,5; 0,75; 1]$ и $x_2(n) = [0; 0,5; 1; 0,5; 0]$. Вывести графики последовательностей и их круговой свертки.

5С. Вычисление реакции ЛДС по формуле свертки с помощью ДПФ.

Выполнить для рекурсивного звена 2-го порядка с коэффициентами, заданными в табл. 10.1, и ИХ длины N_1 .

В качестве воздействия выбрать последовательность с однотональной амплитудной модуляцией (7.23):

$$x(n) = C[1 + m\cos(\Omega n + \varphi_\Omega)]\cos(\hat{\omega}_0 n + \varphi_0).$$

Задать значения $C = 1$, $\hat{\omega}_0 = 2\pi/4$, $\varphi_0 = 0$, $\Omega = \hat{\omega}_0/4$, $\varphi_\Omega = 0$, $m = 0,5$ и длину воздействия $2N$.

Вывести графики ИХ, воздействия и реакции.

6С. Вычисление реакции ЛДС методом перекрытия с накоплением.

Выполнить с исходными данными п. 5С при длине воздействия $4N$.

Вывести графики ИХ, воздействия и реакции.

10.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения каждого пункта задания, включая копируемые из окна **Command Window** результаты вычислений (шрифт Courier New), созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**) и ответы на поставленные вопросы (шрифт Times New Roman).

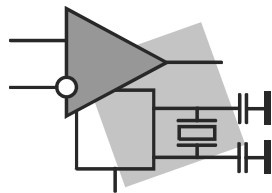
Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. Запишите равенство Парсевалья и поясните его смысл.
2. Что называют растеканием спектра?
3. При каких условиях наблюдается эффект растекания спектра?
4. Какие меры принимают для уменьшения эффекта растекания спектра?
5. Поясните, при каком расстоянии между частотами дискретных гармоник возможно их различение на основе ДПФ?
6. Как улучшить различение дискретных гармоник с близко расположенными частотами?
7. Запишите и поясните формулу круговой свертки.
8. Запишите и поясните формулу линейной свертки.
9. Поясните алгоритм вычисления круговой свертки на основе ДПФ.
10. Поясните алгоритм вычисления линейной свертки на основе ДПФ.
11. С какой целью вычисление реакции ЛДС по формуле свертки выполняется на основе ДПФ?
12. Поясните алгоритм вычисления реакции ЛДС по формуле свертки на основе ДПФ.
13. Поясните алгоритм вычисления реакции ЛДС методом перекрытия с накоплением.

10.7. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 11.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 5.

ГЛАВА 11



Синтез КИХ-фильтров методом окон

Цель работы: изучить процедуру синтеза КИХ-фильтров методом окон и овладеть программными средствами MATLAB для синтеза и анализа КИХ-фильтров; познакомиться с GUI WinTool (Window Design and Analysis Tool — средство создания и анализа окон) и GUI FVTool (Filter Visualization Tool — средство визуализации фильтра).

11.1. Краткая теоретическая справка

Цифровой фильтр (ЦФ) представляет собой линейную дискретную систему (ЛДС), выполняющую преобразование входной последовательности в выходную по алгоритму, описываемому разностным уравнением, который отображается заданной структурой, реализованной аппаратно, программно или аппаратно-программно.

Проектирование ЦФ выполняется в три этапа:

1. Синтез ЦФ, включающий следующие основные шаги:

1.1. Выбор типа ЦФ.

Двум типам ЛДС — нерекурсивная (КИХ) и рекурсивная (БИХ) — соответствуют два типа ЦФ:

- КИХ-фильтр (FIR Filter — Finite Impulse Response Filter);
- БИХ-фильтр (IIR Filter — Infinite Impulse Response Filter).

1.2. Задание требований к характеристикам ЦФ.

Требования к характеристикам ЦФ зависят от его типа (КИХ или БИХ) и назначения ЦФ (частотно-избирательный, преобразователь Гильберта, дифференциатор, амплитудный или фазовый корректор и т. д.).

По умолчанию подразумевают *частотно-избирательные* ЦФ, выполняющие селекцию спектральных составляющих входной последовательности.

Выделяют четыре основных типа избирательности ЦФ:

- ФНЧ — фильтр нижних частот (Lowpass Filter);
- ФВЧ — фильтр верхних частот (Highpass Filter);

- ПФ — полосовой фильтр (Bandpass Filter);
- РФ — режекторный фильтр (Bandstop Filter).

1.3. Выбор метода синтеза.

Метод синтеза зависит от типа ЦФ (КИХ или БИХ), а в рамках одного типа — от специфики дополнительных требований (простоты метода, оптимальности проектируемого фильтра и др.).

1.4. Расчет передаточной функции ЦФ.

1.5. Выбор структуры ЦФ.

2. Моделирование структуры ЦФ с учетом эффектов квантования.

3. Реализация структуры ЦФ.

Структура ЦФ может быть реализована на базе цифрового устройства — цифрового процессора обработки сигналов (ЦПОС), программируемой логической интегральной схеме (ПЛИС) и т. п.

Рассмотрение этого этапа выходит за рамки данной книги.

11.1.1. Свойства КИХ-фильтров

КИХ-фильтр описывается передаточной функцией $H(z)$ (см. (8.3)):

$$H(z) = \sum_{i=0}^{N-1} b_i z^{-i} = \sum_{n=0}^{N-1} h(n) z^{-n}. \quad (11.1)$$

Длиной и порядком КИХ-фильтра называют соответственно число коэффициентов N и порядок R передаточной функции, равный:

$$R = N - 1. \quad (11.2)$$

КИХ-фильтры характеризуются следующими особенностями:

- возможностью обеспечить строго линейную ФЧХ (ЛФЧХ);
- устойчивостью по определению.

Линейная ФЧХ (с точностью до скачков на π)¹ КИХ-фильтра обеспечивается в том случае, если для его импульсной характеристики (ИХ) $h(n)$ выполняется одно из условий:

- симметрии:

$$h(n) = h(N - 1 - n); \quad (11.3)$$

- антисимметрии:

$$h(n) = -h(N - 1 - n), \quad (11.4)$$

где ось симметрии/антисимметрии ИХ $h(n)$ проходит через точку $n = R/2$.

¹ Скачок ФЧХ на π имеет место в точках, где АЧХ равна нулю.

По двум признакам — симметрии/антисимметрии ИХ и четности/нечетности порядка R выделяют четыре типа КИХ-фильтров с ЛФЧХ (табл. 11.1), на базе которых синтезируется ЦФ.

Помимо частотно-избирательных ЦФ, в табл. 11.1 включены два специальных КИХ-фильтра — цифровой преобразователь Гильберта (ЦПГ) и цифровой дифференциатор (ЦД) [1].

Таблица 11.1. Четыре типа КИХ-фильтров с ЛФЧХ

Тип КИХ-фильтра	ЛФЧХ (с точностью до скачков на π)	ЦФ
Тип 1 (Type-1): порядок R — четный; ИХ $h(n)$ — симметричная	$\varphi(\hat{\omega}) = -\frac{\hat{\omega}R}{2}$	ФНЧ, ФВЧ, ПФ, РФ
Тип 2 (Type-2): порядок R — нечетный; $h(n)$ — симметричная	$\varphi(\hat{\omega}) = -\frac{\hat{\omega}R}{2}$	ФНЧ, ПФ
Тип 3 (Type-3): порядок R — четный; $h(n)$ — антисимметричная; $h\left(\frac{R}{2}\right) = 0$	$\varphi(\hat{\omega}) = \frac{\pi}{2} - \frac{\hat{\omega}R}{2}$	ПФ ЦПГ ЦД
Тип 4 (Type-4): порядок R — нечетный; $h(n)$ — антисимметричная	$\varphi(\hat{\omega}) = \frac{\pi}{2} - \frac{\hat{\omega}R}{2}$	ФВЧ, ПФ ЦПГ ЦД

11.1.2. Задание требований к АЧХ

Методы синтеза частотно-избирательных КИХ-фильтров изначально предполагают ЛФЧХ, поэтому требования задаются к нормированной АЧХ $\hat{A}(f)$ в основной полосе частот $[0; f_d/2]$ и включают в себя:

- частоту дискретизации f_d ;
- граничные частоты полос пропускания (ПП) и полос задерживания (ПЗ), для которых введены условные обозначения:
 - f_χ — граничная частота ПП для ФНЧ и ФВЧ;
 - f_k — граничная частота ПЗ для ФНЧ и ФВЧ;

- $f_{-\chi}$, f_{χ} — левая и правая граничные частоты ПП для ПФ и РФ;
- f_{-k} , f_k — левая и правая граничные частоты ПЗ для ПФ и РФ;

□ *максимально допустимые отклонения АЧХ $\hat{A}(f)$, для которых введены условные обозначения:*

- δ_1 — от единицы в ПП (для ФНЧ, ФВЧ и ПФ);
- δ_2 — от нуля в ПЗ (для ФНЧ, ФВЧ и РФ);
- δ_{11} — от единицы в левой полосе пропускания — ПП1 (для РФ);
- δ_{12} — от единицы в правой полосе пропускания — ПП2 (для РФ);
- δ_{21} — от нуля в левой полосе задерживания — ПЗ1 (для ПФ);
- δ_{22} — от нуля в правой полосе задерживания — ПЗ2 (для ПФ).

На рис. 11.1—11.4 приведены примеры идеальной АЧХ и требований к АЧХ для фильтров различного типа избирательности.

Требования могут задаваться к АЧХ в децибелах — к характеристике ослабления:

$$\hat{A}(f) \text{ (дБ)} = 20 \lg(\hat{A}(f)) \quad (11.5)$$

или к характеристике затухания:

$$\hat{A}(f) \text{ (дБ)} = -20 \lg(\hat{A}(f)). \quad (11.6)$$

В MATLAB требования задаются к характеристике затухания — АЧХ (дБ).

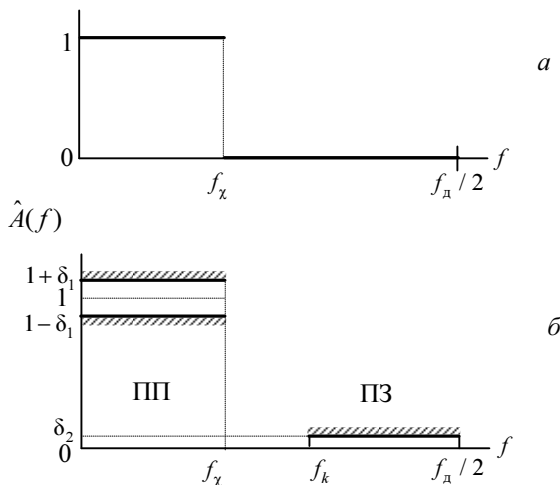


Рис. 11.1. Идеальная АЧХ ФНЧ (а), требования к АЧХ ФНЧ (б)

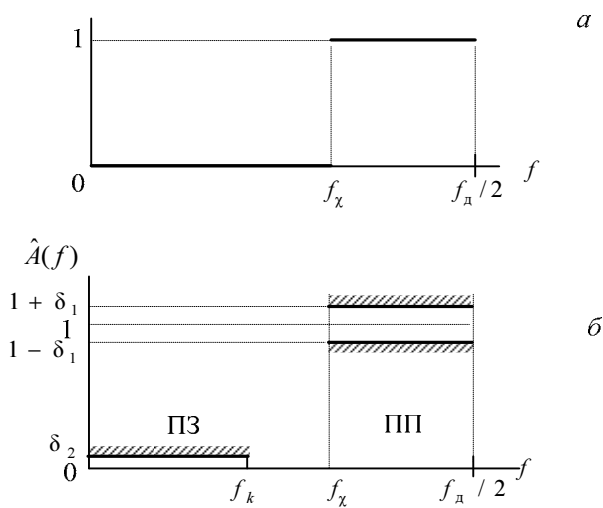


Рис. 11.2. Идеальная АЧХ ФВЧ (а), требования к АЧХ ФВЧ (б)

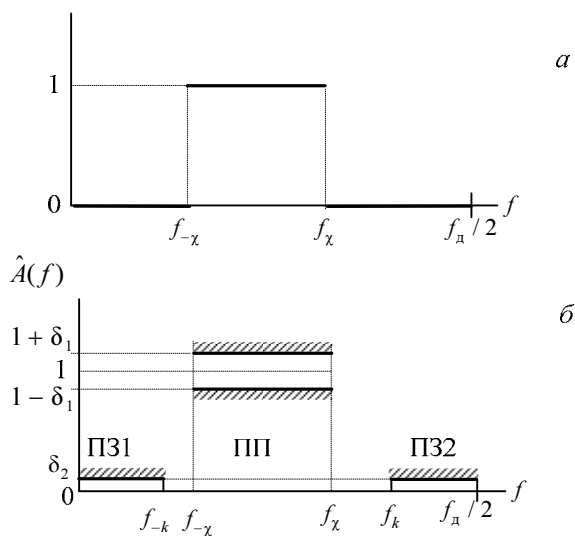


Рис. 11.3. Идеальная АЧХ ПФ (а), требования к АЧХ ПФ (б)

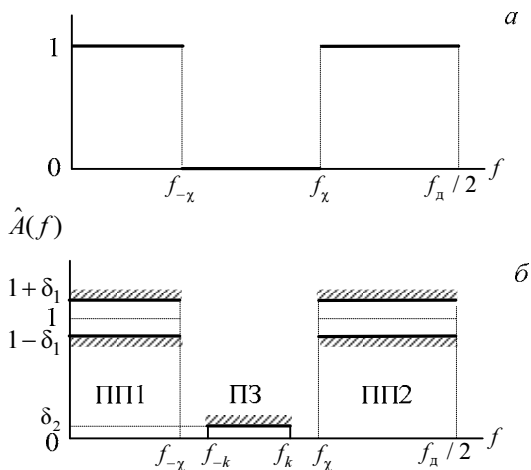


Рис. 11.4. Идеальная АЧХ РФ (а), требования к АЧХ (б) РФ (б)

В требованиях к характеристике затухания (11.6) вместо значений максимально допустимых отклонений δ_1 , δ_2 , δ_{11} , δ_{12} и δ_{22} задаются:

- $a_{\max}$ (дБ) — максимально допустимое затухание в ПП (для ФНЧ, ФВЧ и ПФ);
- $a_{\min}$ (дБ) — минимально допустимое затухание в ПЗ (для ФНЧ, ФВЧ и РФ);
- $a_{1\max}$ (дБ) — максимально допустимое затухание в ПП1 (для РФ);
- $a_{2\max}$ (дБ) — максимально допустимое затухание в ПП2 (для РФ);
- $a_{1\min}$ (дБ) — минимально допустимое затухание в ПЗ1 (для ПФ);
- $a_{2\min}$ (дБ) — минимально допустимое затухание в ПЗ2 (для ПФ).

На рис. 11.5 приведен пример требований к характеристике затухания ФНЧ.

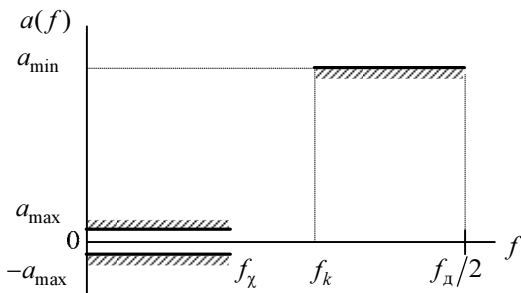


Рис. 11.5. Требования к характеристике затухания ФНЧ

Взаимосвязь между значениями максимально допустимых отклонений и их соответствующими значениями в децибелах, например, между δ_1 и δ_2 и $a_{\max}$ и $a_{\min}$ устанавливается формулами:

$$a_{\max} = -20 \lg(1 - \delta_1) \text{ (дБ);}$$

(11.7)

$$a_{\min} = -20 \lg(\delta_2) \text{ (дБ),}$$

(11.8)

и наоборот:

$$\delta_1 = 1 - 10^{-a_{\max}/20};$$

(11.9)

$$\delta_2 = 10^{-a_{\min}/20}.$$

(11.10)

11.1.3. Структуры КИХ-фильтров

Структура (структурная схема) ЦФ отображает алгоритм вычисления реакции по разностному уравнению и определяется видом передаточной функции.

Структурные схемы КИХ-фильтров с ЛФЧХ приведены на рис. 11.6.

В MATLAB структура КИХ-фильтра с ЛФЧХ описывается в виде объекта `dfilt`:

```
Hd = dfilt.structure(b)
```

где `Hd` — имя объекта `dfilt`; `dfilt` — тип объекта; `structure` — функция, задающая конкретную структуру объекта `Hd` (табл. 11.2); `b` — параметр функции `structure` — вектор коэффициентов передаточной функции (11.1).

Для КИХ-фильтров свойства объекта `dfilt`, выводимые по его имени, включают в себя:

- ☐ `FilterStructure` — структура КИХ-фильтра;
- ☐ `Arithmetic` — форма представления данных;
- ☐ `Numerator` — коэффициенты передаточной функции;
- ☐ `PersistentMemory` — начальные условия при вычислении реакции; значение `false` соответствует ННУ (см. разд. 8.1).

При синтезе КИХ-фильтров с ЛФЧХ методом окон ИХ может быть только симметричной, чему соответствует структура `Direct-Form Symmetric FIR`.

Таблица 11.2. Функции `structure` и структуры КИХ-фильтров с ЛФЧХ

Функция <i>structure</i>	Параметр функции <i>structure</i>	Структура КИХ-фильтра
<code>dffir</code>	<code>b</code> — вектор коэффициентов передаточной функции (11.1)	Direct-Form FIR (прямая)
<code>dfsymfir</code>	<code>b</code> — вектор коэффициентов передаточной функции (11.1), симметричных относительно $N/2$	Direct-Form Symmetric FIR (прямая приведенная с симметричной ИХ, см. рис. 11.6, а)
<code>dfasymfir</code>	<code>b</code> — вектор коэффициентов передаточной функции (11.1), антисимметричных относительно $N/2$; при четном N — $b(N/2)=0$	Direct-Form Antisymmetric FIR (прямая приведенная с антисимметричной ИХ, см. рис. 11.6, б)

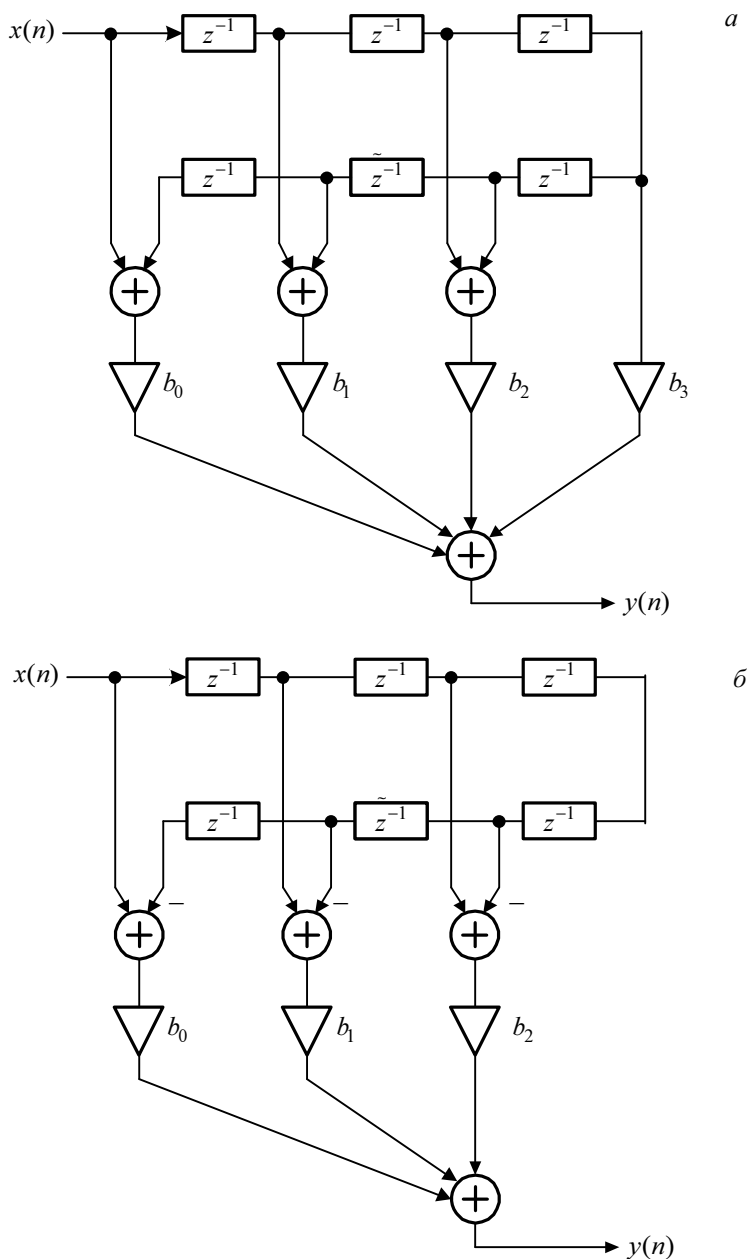


Рис. 11.6. Структурные схемы КИХ-фильтров с ЛФЧХ: прямая приведенная с симметричной ИХ (Direct-Form Symmetric FIR) для КИХ-фильтра 1-го типа длины $N = 7$ (а); прямая приведенная с антисимметричной ИХ (Direct-Form Antisymmetric FIR) для КИХ-фильтра 3-го типа длины $N = 7$ ($b_2 = 0$) (б)

11.1.4. Процедура синтеза КИХ-фильтров методом окон

В общем случае синтез ЦФ заключается в расчете передаточной функции. Согласно (11.1), синтез КИХ-фильтра сводится к расчету его импульсной характеристики.

Процедура синтеза КИХ-фильтра методом окон является *итерационной* и включает в себя следующие шаги¹:

1. Задание требований к АЧХ.
2. Оценка порядка фильтра R и выбор окна.

Оценкой порядка R называют начальное значение порядка в итерационной процедуре синтеза фильтра.

Окном называют *весовую функцию* $w(n)$ — вещественную неотрицательную последовательность длины $N = R + 1$, максимальную в центре и монотонно спадающую к границам. Для стандартных окон MATLAB значения $w(n)$ вычисляются автоматически по известным аналитическим формулам.

3. Расчет импульсной характеристики *идеального* фильтра $h_n(n)$, симметрично усеченной до длины $N = R + 1$ (выделенной окном Дирихле).

Импульсная характеристика $h_n(n)$ может быть только *симметричной* и рассчитывается автоматически по известным для идеальных ФНЧ, ФВЧ, ПФ, РФ аналитическим формулам. Обязательным параметром усеченной ИХ $h_n(n)$ является *частота разрыва* (отсечки), на которой нормированная АЧХ равна 0,5.

Для ФНЧ и ФВЧ указывается одна частота разрыва, равная:

$$f_c = \frac{f_{\chi} + f_k}{2}, \quad (11.11)$$

а для ПФ и РФ — две (левая и правая), равные:

$$f_{c1} = \frac{f_{-k} + f_{-\chi}}{2}; \quad (11.12)$$

$$f_{c2} = \frac{f_k + f_{\chi}}{2}. \quad (11.13)$$

4. Расчет импульсной характеристики *реального* фильтра с симметричной $h(n)$ длины N в виде произведения:

$$h(n) = h_n(n)w(n).$$

¹ С теоретическими основами метода окон можно познакомиться в [2, 3].

5. Проверка выполнения требований к АЧХ.

Проверка заключается в сравнении *фактических* максимальных по модулю отклонений АЧХ от идеальной АЧХ в ПП и ПЗ с *заданными* максимально допустимыми отклонениями.

Возможны две ситуации.

- Требования к АЧХ не выполняются.

В этом случае следует *увеличить* порядок R и вернуться к пп. 3—5.

- Требования к АЧХ выполняются.

В этом случае следует *уменьшить* порядок R и вернуться к пп. 3—5.

В обоих случаях увеличение/уменьшение порядка R продолжается до тех пор, пока не будет найден минимальный порядок $R_{\min}$, при котором выполняются требования к АЧХ.

6. Выбор структуры КИХ-фильтра.

11.1.5. Синтез КИХ-фильтров методом окон в MATLAB

Основной проблемой синтеза КИХ-фильтров методом окон является оценка порядка фильтра R (длины окна $N = R + 1$) по заданным требованиям к АЧХ. В общем случае он может задаваться произвольно, а затем уточняться при проверке выполнения требований к АЧХ. Однако эта задача успешно решена для окна Кайзера, где порядок фильтра и требования к АЧХ связаны аналитической зависимостью [1, 2]. Поэтому далее процедура синтеза КИХ-фильтра методом окон рассматривается с применением окна Кайзера.

Синтез КИХ-фильтров методом окон с произвольным окном выполняется с помощью функции:

b = fir1(R,wc,ftype,win,normalization)

где R — порядок КИХ-фильтра (11.2); wc — вектор нормированных частот разрыва (см. разд. 11.1.4):

□ для ФНЧ и ФВЧ — с одним элементом $wc(1)$, равным

$$\hat{f}_c = \frac{f_c}{f_d/2}; \quad (11.14)$$

□ для ПФ и РФ — с двумя элементами $wc(1)$ и $wc(2)$, соответственно равными:

$$\hat{f}_{c1} = \frac{f_{c1}}{f_d/2}; \quad (11.15)$$

$$\hat{f}_{c2} = \frac{f_{c2}}{f_d/2}; \quad (11.16)$$

`ftype` — параметр, указывающий тип избирательности и принимающий значения:

□ 'high' — для ФВЧ;

□ 'stop' — для РФ;

□ по умолчанию (если значение параметра не задано), синтезируется ФНЧ или ПФ.

`win` — имя стандартной функции для расчета окна $w(n)$ длины $N = R + 1$; для окна Кайзера — `kaiser(R+1,beta)`, где `beta` — параметр β окна Кайзера.

`normalization` — параметр (флаг), управляющий нормированием АЧХ таким образом, чтобы обеспечить ее значение, равное единице, в центре ПП (для РФ — в центре ПП1), и принимающий значения:

□ 'scale' (по умолчанию) — нормирование выполняется;

□ 'noscale' — нормирование не выполняется;

`b` — вектор коэффициентов передаточной функции (11.1) длины $N = R + 1$.

Для окна Кайзера входные параметры функции `fir1` — порядок R и вектор `wc`, а также тип избирательности `ftype` и параметр окна Кайзера `beta` — определяются по заданным требованиям к АЧХ с помощью функции:

`[R,wc,beta,ftype] = kaiserord(f,m,ripple,Fs)`

где `f` — вектор граничных частот ПП и ПЗ в порядке их следования слева направо в шкале частот f (Гц) в основной полосе частот $[0; f_d/2]$; `m` — вектор значений идеальной АЧХ (единица — в ПП и нуль — в ПЗ) в порядке их следования слева направо; соблюдается условие `length(f) = 2*length(m)-2`; `ripple` — вектор максимально допустимых отклонений АЧХ в порядке их следования слева направо; `Fs` — частота дискретизации f_d (Гц); R — оценка порядка фильтра R с точностью до ± 2 ; `wc` — вектор, определенный ранее для функции `fir1`; `beta` — параметр β окна Кайзера; `ftype` — параметр, указывающий тип избирательности и принимающий значения:

□ 'low' — для ФНЧ;

□ 'high' — для ФВЧ;

□ 'DC-0' — для ПФ;

□ 'stop' — для РФ.

11.2. Содержание лабораторной работы

Содержание работы связано с синтезом КИХ-фильтров методом окон с применением окна Кайзера, описанием их структур и анализом характеристик с использованием программных средств MATLAB.

11.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файлов `lr_11_low`, `lr_11_high`, `lr_11_pass` и `lr_11_stop` и function-файлов `check_low`, `check_high`, `check_pass`, `check_stop` и `plot_fir`, которые хранятся на прилагаемом компакт-диске в папке `LAB_DSP\LAB_11`.

Перед выполнением работы необходимо сохранить путь к папке `LAB_11` по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 11.3—11.6 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$, и включают в себя требования к АЧХ КИХ-фильтров ФНЧ, ФВЧ, ПФ и РФ.

На прилагаемом компакт-диске в папке `Tables\Tables_11` хранятся табл. 11.3—11.6 исходных данных, примеры их заполнения для $N_{бр} = 1$ и табл. 11.7 для п. 2 задания.

Таблица 11.3. Требования к АЧХ ФНЧ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_d	Частота дискретизации	$f_d = 5000 + 100N_{бр}$	Fs =
f_χ	Граничная частота ПП	$f_\chi = \frac{f_d}{10} + 20N_{бр}$	ft =
f_k	Граничная частота ПЗ	$f_k = \frac{f_d}{10} + 250 + 25N_{бр}$	fk =
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$	d1 = 0.05
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$	d2 = 0.01

Таблица 11.4. Требования к АЧХ ФВЧ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_d	Частота дискретизации	$f_d = 5000 + 100N_{бр}$	Fs =
f_k	Граничная частота ПЗ	$f_k = \frac{f_d}{10} + 20N_{бр}$	fk =
f_χ	Граничная частота ПП	$f_\chi = \frac{f_d}{10} + 250 + 25N_{бр}$	ft =

Таблица 11.4 (окончание)

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$	d2 = 0.01
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$	d1 = 0.05

Таблица 11.5. Требования к АЧХ ПФ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}}$	Fs =
f_{-k}	Граничная частота ПЗ1	$f_{-k} = \frac{f_d}{20} + 20N_{\text{бр}}$	fk1 =
$f_{-\chi}$	Граничная частота ПП1	$f_{-\chi} = \frac{f_d}{20} + 250 + 25N_{\text{бр}}$	ft1 =
f_{χ}	Граничная частота ПП2	$f_{\chi} = \frac{f_d}{4} + 25N_{\text{бр}}$	ft2 =
f_k	Граничная частота ПЗ2	$f_k = \frac{f_d}{4} + 250 + 30N_{\text{бр}}$	fk2 =
δ_{21}	Максимально допустимое отклонение в ПЗ1	$\delta_{21} = 0,01$	d21 = 0.01
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$	d1 = 0.05
δ_{22}	Максимально допустимое отклонение в ПЗ2	$\delta_{22} = 0,01$	d22 = 0.01

Таблица 11.6. Требования к АЧХ РФ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}}$	Fs =
$f_{-\chi}$	Граничная частота ПП1	$f_{-\chi} = \frac{f_d}{20} + 20N_{\text{бр}}$	ft1 =
f_{-k}	Граничная частота ПЗ1	$f_{-k} = \frac{f_d}{20} + 250 + 25N_{\text{бр}}$	fk1 =

Таблица 11.6 (окончание)

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_k	Граничная частота ПЗ2	$f_k = \frac{f_d}{4} + 25N_{бр}$	fk2 =
f_χ	Граничная частота ПП2	$f_\chi = \frac{f_d}{4} + 250 + 30N_{бр}$	ft2 =
δ_{11}	Максимально допустимое отклонение в ПП1	$\delta_{11} = 0,05$	d11 = 0.05
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$	d2 = 0.01
δ_{12}	Максимально допустимое отклонение в ПП2	$\delta_{12} = 0,05$	d12 = 0.05

Задание на лабораторную работу заключается в синтезе КИХ-фильтров методом окон с применением окна Кайзера и анализе их характеристик и для каждого типа избирательности (ФНЧ, ФВЧ, ПФ и РФ) включает в себя выполнение следующих пунктов:

- 1. Ввод требований к АЧХ.
- 2. Вычисление оценки порядка КИХ-фильтра, нормированных частот разрыва и параметра β окна Кайзера.

Выведенные значения нормированных частот разрыва (w_c) и параметра β (beta) внести в табл. 11.7.

Пояснить:

- какая функция используется для вычисления оценки порядка КИХ-фильтра, нормированных частот разрыва и параметра β ;
- с какой целью рассчитывается оценка порядка КИХ-фильтра;
- как рассчитываются частоты разрыва в шкале частот f (Гц).

Таблица 11.7. Результаты синтеза КИХ-фильтров методом окон

Тип избирательности фильтра	Метод окон с окном Кайзера		
	порядок фильтра R	вектор нормированных частот разрыва w_c	параметр beta
ФНЧ			
ФВЧ			
ПФ			
РФ			

3. Синтез КИХ-фильтра методом окон.

Для синтеза КИХ-фильтра организовать цикл, в теле которого выполнить следующие действия:

- синтезировать КИХ-фильтр;
- проверить выполнение требований к АЧХ.

Для проверки выполнения требований к АЧХ вывести и сравнить *фактические* максимальные (по модулю) отклонения в ПП и ПЗ с *заданными* максимально допустимыми отклонениями.

Для вывода фактических максимальных по модулю отклонений использовать созданные function-файлы (см. разд. 11.4.5);

- по результатам проверки, увеличивая или уменьшая порядок КИХ-фильтра, определить его минимальный порядок, при котором выполняются требования к АЧХ.

При увеличении/уменьшении порядка КИХ-фильтра необходимо учитывать соответствие между типом избирательности ЦФ и типом КИХ-фильтра (см. табл. 11.1).

Полученное в результате итерационной процедуры значение порядка R КИХ-фильтра внести в табл. 11.7.

Пояснить:

- какая функция используется для синтеза КИХ-фильтра;
- какой из параметров данной функции соответствует коэффициентам передаточной функции КИХ-фильтра;
- смысл итерационной процедуры синтеза;
- какие типы КИХ-фильтров можно использовать в методе окон.

4. Анализ характеристик КИХ-фильтра.

Для вывода графиков ИХ, АЧХ и ФЧХ КИХ-фильтра использовать function-файл `plot_fir` (см. разд. 11.4.5).

Пояснить:

- вид ИХ;
- вид АЧХ в ПП и ПЗ (воспользуйтесь кнопкой **Zoom in** на панели инструментов);
- вид ФЧХ.

5. Описание структуры КИХ-фильтра в виде объекта `dfilt` с именами:

- `F_lowpass` — ФНЧ;
- `F_highpass` — ФВЧ;
- `F_bandpass` — ПФ;
- `F_bandstop` — РФ.

Пояснить:

- что отображает структура и чем определяется ее вид;
- свойства объекта `dfilt`.

6. Знакомство с GUI FVTool.

Обратиться к GUI FVTool по команде:

```
fvtool(Hd)
```

где `Hd` — имя объекта `dfilt`, и проанализировать характеристики синтезированных КИХ-фильтров.

7. Знакомство с GUI WinTool.

Для знакомства с окнами и их характеристиками обратиться к GUI WinTool по команде:

```
wintool
```

11.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должны быть представлены табл. 11.3—11.6 исходных данных для своего номера бригады $N_{бр}$.

Для четырех типов избирательности КИХ-фильтра — ФНЧ, ФВЧ, ПФ и РФ — созданы четыре script-файла. Для *запуска* script-файла к нему необходимо обратиться по имени:

```
>> lr_11_low — ФНЧ
```

```
>> lr_11_high — ФВЧ
```

```
>> lr_11_pass — ПФ
```

```
>> lr_11_stop — РФ
```

Листинги данных script-файлов представлены в разд. 11.4.1—11.4.4.

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш <Ctrl>+<Break>.

При выполнении script-файла текущие окна с графиками *не закрывать*.

11.4.1. Синтез и анализ КИХ-фильтра ФНЧ

Листинг script-файла `lr_11_low` имеет вид:

```
>> type lr_11_low
script
clc
clear
```

```

disp('% ЛР №11. СИНТЕЗ КИХ-ФИЛЬТРА ФНЧ МЕТОДОМ ОКОН')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ ФНЧ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ')
DATA=0;
while DATA==0;
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ ( $f_{\text{ц}}$ )
ft = input('ft = ');           % ГРАНИЧНАЯ ЧАСТОТА ПП ( $f_{\text{ц}}$ )
fk = input('fk = ');           % ГРАНИЧНАЯ ЧАСТОТА ПЗ ( $f_{\text{ц}}$ )
d1 = input('d1 = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПП
d2 = input('d2 = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПЗ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ kaiserord')
disp('%')
disp('%')
disp('% Для вывода параметров функции kaiserord нажмите <ENTER>')
pause
m = [1 0];                     % ВЕКТОР ЗНАЧЕНИЙ ИДЕАЛЬНОЙ АЧХ
f = [ft fk];                   % ВЕКТОР ГРАНИЧНЫХ ЧАСТОТ
ripple = [d1 d2];              % ВЕКТОР МАКСИМАЛЬНО ДОПУСТИМЫХ ОТКЛОНЕНИЙ
[R,wc,beta,ftype] = kaiserord(f,m,ripple,Fs); % ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ОКНА
КАЙЗЕРА
disp(['R = ' num2str(R)])      % ОЦЕНКА ПОРЯДКА КИХ-ФИЛЬТРА
disp(['wc = ' num2str(wc)])     % НОРМИРОВАННАЯ ЧАСТОТА РАЗРЫВА
disp(['beta = ' num2str(beta)]) % ПАРАМЕТР ОКНА КАЙЗЕРА
disp(['ftype = ' char(ftype)]) % ТИП КИХ-ФИЛЬТРА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. СИНТЕЗ КИХ-ФИЛЬТРА ФНЧ')

```



```

ORDER = 0; % ПРИЗНАК ОПТИМАЛЬНОСТИ ПОРЯДКА КИХ-ФИЛЬТРА: 0 — НЕОПТИМАЛЬНЫЙ;
1 — ОПТИМАЛЬНЫЙ
while ORDER==0;
disp('%')
disp('%')
disp('% Для синтеза КИХ-фильтра ФНЧ нажмите <ENTER>')
pause
b1 = fir1(R,wc,ftype,kaiser(R+1,beta),'noscale'); % КОЭФФИЦИЕНТЫ КИХ-ФИЛЬТРА
ФНЧ
disp('%')
disp('%')
disp([' Синтезирован КИХ-фильтр ФНЧ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для вывода ФАКТИЧЕСКИХ максимальных отклонений АЧХ')
disp('% в ПП (dp) и ПЗ (ds) и ЗАДАННЫХ отклонений d1 и d2 нажмите <ENTER>')
pause
[dp,ds] = check_low(b1,ft,fk,Fs); % ВЫЧИСЛЕНИЕ ФАКТИЧЕСКИХ МАКСИМАЛЬНЫХ
ПО МОДУЛЮ ОТКЛОНЕНИЙ В ПП И ПЗ
disp('%')
disp(['dp = ' num2str(dp) ' ds = ' num2str(ds)])
disp(['d1 = ' num2str(d1) ' d2 = ' num2str(d2)])
disp('%')
disp('%')
disp('% Сравните ФАКТИЧЕСКИЕ отклонения с ЗАДАННЫМИ')
disp('%')
disp('% Если ПОРЯДОК соответствует МИНИМАЛЬНОМУ, введите 1')
disp('% Если НЕ соответствует, введите 0 и затем ПОРЯДОК R')
ORDER = input('--> ');
if ORDER==0
R = input('R = '); % ПОРЯДОК КИХ-ФИЛЬТРА
end
end
disp('%')
disp([' Синтезирован ФНЧ минимального порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. АНАЛИЗ ХАРАКТЕРИСТИК КИХ-ФИЛЬТРА ФНЧ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИХ, АЧХ и ФЧХ нажмите <ENTER>')
pause
figure('Name','Lowpass FIR Filter — Impulse Response, Magnitude,
Phase','NumberTitle','off')

```

```

plot_fir(R,b1,Fs)           % ПОСТРОЕНИЕ ГРАФИКОВ ИХ, АЧХ и ФЧХ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ОПИСАНИЕ СТРУКТУРЫ КИХ-ФИЛЬТРА ФНЧ В ВИДЕ ОБЪЕКТА dfilt')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_lowpass = dfilt.dfsymfir(b1)      % ОБЪЕКТ dfilt – КИХ-ФИЛЬТР ФНЧ
disp('%')
disp('%')
disp('% СИНТЕЗ КИХ-ФИЛЬТРА ФНЧ ЗАВЕРШЕН')

```

11.4.2. Синтез и анализ КИХ-фильтра ФВЧ

Листинг script-файла lr_11_high имеет вид:

```

>> type lr_11_high
script
clc
clear
disp('% ЛР №11. СИНТЕЗ КИХ-ФИЛЬТРА ФВЧ МЕТОДОМ ОКОН')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ ФВЧ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ')
DATA=0;
while DATA==0;
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
fk = input('fk = ');           % ГРАНИЧНАЯ ЧАСТОТА ПЗ (Гц)
ft = input('ft = ');           % ГРАНИЧНАЯ ЧАСТОТА ПП (Гц)
d2 = input('d2 = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПЗ
d1 = input('d1 = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПП
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')

```

```

disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ kaiserord')
disp('%')
disp('%')
disp('% Для вывода параметров функции kaiserord нажмите <ENTER>')
pause
m = [0 1]; % ВЕКТОР ЗНАЧЕНИЙ ИДЕАЛЬНОЙ АЧХ
f = [fk ft]; % ВЕКТОР ГРАНИЧНЫХ ЧАСТОТ
ripple = [d2 d1]; % ВЕКТОР МАКСИМАЛЬНО ДОПУСТИМЫХ ОТКЛОНЕНИЙ
[R,wc,beta,ftype] = kaiserord(f,m,ripple,Fs); % ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ОКНА
КАЙЗЕРА
disp(['R = ' num2str(R)]) % ОЦЕНКА ПОРЯДКА КИХ-ФИЛЬТРА
disp(['wc = ' num2str(wc)]) % НОРМИРОВАННАЯ ЧАСТОТА РАЗРЫВА
disp(['beta = ' num2str(beta)]) % ПАРАМЕТР ОКНА КАЙЗЕРА
disp(['ftype = ' char(ftype)]) % ТИП КИХ-ФИЛЬТРА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. СИНТЕЗ КИХ-ФИЛЬТРА ФВЧ')
ORDER = 0; % ПРИЗНАК ОПТИМАЛЬНОСТИ ПОРЯДКА КИХ-ФИЛЬТРА: 0 — НЕОПТИМАЛЬНЫЙ;
1 — ОПТИМАЛЬНЫЙ
while ORDER==0;
disp('%')
disp('%')
disp('% Для синтеза КИХ-фильтра ФВЧ нажмите <ENTER>')
pause
b2 = fir1(R,wc,ftype,kaiser(R+1,beta),'noscale'); % КОЭФИЦИЕНТЫ
КИХ-ФИЛЬТРА ФВЧ
disp('%')
disp('%')
disp([' Синтезирован КИХ-фильтр ФВЧ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для вывода ФАКТИЧЕСКИХ максимальных отклонений АЧХ')
disp('% в ПЗ (ds) и ПП (dp) и ЗАДАННЫХ отклонений d2 и d1 нажмите <ENTER>')
pause
[ds,dp] = check_high(b2,fk,ft,Fs); % ВЫЧИСЛЕНИЕ ФАКТИЧЕСКИХ МАКСИМАЛЬНЫХ
ПО МОДУЛЮ ОТКЛОНЕНИЙ В ПЗ И ПП
disp('%')
disp(['ds = ' num2str(ds) ' dp = ' num2str(dp)])
disp(['d2 = ' num2str(d2) ' d1 = ' num2str(d1)])

```

```

disp('%')
disp('%')
disp('% Сравните ФАКТИЧЕСКИЕ отклонения с ЗАДАНЫМИ')
disp('%')
disp('% Если ПОРЯДОК соответствует МИНИМАЛЬНОМУ, введите 1')
disp('% Если НЕ соответствует, введите 0 и затем ПОРЯДОК R')
ORDER = input('--> ');
if ORDER==0
R = input('R = '); % ПОРЯДОК КИХ-ФИЛЬТРА
while rem(R,2)~=0
disp('% Порядок фильтра выбран НЕПРАВИЛЬНО')
R = input('R = '); % ПОРЯДОК КИХ-ФИЛЬТРА
end
end
end
disp('%')
disp([' Синтезирован ФВЧ минимального порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. АНАЛИЗ ХАРАКТЕРИСТИК КИХ-ФИЛЬТРА ФВЧ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИХ, АЧХ и ФЧХ нажмите <ENTER>')
pause
figure('Name','Highpass FIR Filter – Impulse Response, Magnitude,
Phase','NumberTitle', 'off')
plot_fir(R,b2,Fs) % ПОСТРОЕНИЕ ГРАФИКОВ ИХ, АЧХ и ФЧХ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ОПИСАНИЕ СТРУКТУРЫ КИХ-ФИЛЬТРА ФВЧ В ВИДЕ ОБЪЕКТА dfilt')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_highpass = dfilt.dfsymfir(b2) % ОБЪЕКТ dfilt – КИХ-ФИЛЬТР ФВЧ
disp('%')
disp('%')
disp('% СИНТЕЗ КИХ-ФИЛЬТРА ФВЧ ЗАВЕРШЕН')

```

11.4.3. Синтез и анализ КИХ-фильтра ПФ

Листинг script-файла `lr_05_pass` имеет вид:

```
>> type lr_11_pass
script
clc
clear
disp('% ЛР №11. СИНТЕЗ КИХ-ФИЛЬТРА ПФ МЕТОДОМ ОКОН')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ ПФ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ')
DATA=0;
while DATA==0;
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
fk1 = input('fk1 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПЗ1 (Гц)
ft1 = input('ft1 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПП1 (Гц)
ft2 = input('ft2 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПП2 (Гц)
fk2 = input('fk2 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПЗ2 (Гц)
d21 = input('d21 = ');         % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПЗ1
d1 = input('d1 = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПП
d22 = input('d22 = ');         % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПЗ2
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ kaiserord')
disp('%')
disp('%')
disp('% Для вывода параметров функции kaiserord нажмите <ENTER>')
pause
m = [0 1 0];                  % ВЕКТОР ЗНАЧЕНИЙ ИДЕАЛЬНОЙ АЧХ
f = [fk1 ft1 ft2 fk2];        % ВЕКТОР ГРАНИЧНЫХ ЧАСТОТ
ripple = [d21 d1 d22];        % ВЕКТОР МАКСИМАЛЬНО ДОПУСТИМЫХ ОТКЛОНЕНИЙ
[R,wc,beta,ftype] = kaiserord(f,m,ripple,Fs); % ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ОКНА
КАЙЗЕРА
```

```

disp(['R = ' num2str(R)]) % ОЦЕНКА ПОРЯДКА КИХ-ФИЛЬТРА
disp(['wc(1) = ' num2str(wc(1)) ' wc(2) = ' num2str(wc(2))]) % ВЕКТОР
НОРМИРОВАННЫХ ЧАСТОТ РАЗРЫВА
disp(['beta = ' num2str(beta)]) % ПАРАМЕТР ОКНА КАЙЗЕРА
disp(['ftype = ' char(ftype)]) % ТИП КИХ-ФИЛЬТРА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. СИНТЕЗ КИХ-ФИЛЬТРА ПФ')
ORDER = 0; % ПРИЗНАК ОПТИМАЛЬНОСТИ ПОРЯДКА КИХ-ФИЛЬТРА: 0 — НЕОПТИМАЛЬНЫЙ;
1 — ОПТИМАЛЬНЫЙ
while ORDER==0;
disp('%')
disp('%')
disp('% Для синтеза КИХ-фильтра ПФ нажмите <ENTER>')
pause
b3 = fir1(R,wc,ftype,kaiser(R+1,beta),'noscale'); % КОЭФФИЦИЕНТЫ
КИХ-ФИЛЬТРА ПФ
disp('%')
disp('%')
disp([' Синтезирован КИХ-фильтр ПФ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для вывода ФАКТИЧЕСКИХ максимальных отклонений АЧХ')
disp('% в ПЗ1 (ds1), ПП (dp) и ПЗ2 (ds2) и ЗАДАННЫХ отклонений d21, d1 и d22
нажмите <ENTER>')
pause
[ds1,dp,ds2] = check_pass(b3,fk1,ft1,ft2,fk2,Fs); % ВЫЧИСЛЕНИЕ ФАКТИЧЕСКИХ
МАКСИМАЛЬНЫХ ПО МОДУЛЮ ОТКЛОНЕНИЙ В ПЗ1, ПП И ПЗ2
disp('%')
disp(['ds1=' num2str(ds1) ' dp = ' num2str(dp) ' ds2 = '
num2str(ds2)])
disp(['d21 = ' num2str(d21) ' d1 = ' num2str(d1) ' d22 = '
num2str(d22)])
disp('%')
disp('%')
disp('% Сравните ФАКТИЧЕСКИЕ отклонения с ЗАДАННЫМИ')
disp('%')
disp('% Если ПОРЯДОК соответствует МИНИМАЛЬНОМУ, введите 1')
disp('% Если НЕ соответствует, введите 0 и затем ПОРЯДОК R')
ORDER = input('---> ');
if ORDER==0
R = input('R = '); % ПОРЯДОК КИХ-ФИЛЬТРА
end
end
end

```

```

disp('%')
disp([' Синтезирован ПФ минимального порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. АНАЛИЗ ХАРАКТЕРИСТИК КИХ-ФИЛЬТРА ПФ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИХ, АЧХ и ФЧХ нажмите <ENTER>')
pause
figure('Name','Bandpass FIR Filter – Impulse Response, Magnitude,
Phase','NumberTitle', 'off')
plot_fir(R,b3,Fs) % ПОСТРОЕНИЕ ГРАФИКОВ ИХ, АЧХ и ФЧХ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ОПИСАНИЕ СТРУКТУРЫ КИХ-ФИЛЬТРА ПФ В ВИДЕ ОБЪЕКТА dfilt')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_bandpass = dfilt.dfssymfir(b3) % ОБЪЕКТ dfilt – КИХ-ФИЛЬТР ПФ
disp('%')
disp('%')
disp('% СИНТЕЗ КИХ-ФИЛЬТРА ПФ ЗАВЕРШЕН')

```

11.4.4. Синтез и анализ КИХ-фильтра РФ

Листинг script-файла lr_11_stop имеет вид:

```

>> type lr_11_stop
script
clc
clear
disp('% ЛР №11. СИНТЕЗ КИХ-ФИЛЬТРА РФ МЕТОДОМ ОКОН')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ РФ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ')

```

```

DATA=0;
while DATA==0;
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ ( $\Gamma_{\text{ц}}$ )
ft1 = input('ft1 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПП1 ( $\Gamma_{\text{ц}}$ )
fk1 = input('fk1 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПЗ1 ( $\Gamma_{\text{ц}}$ )
fk2 = input('fk2 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПЗ2 ( $\Gamma_{\text{ц}}$ )
ft2 = input('ft2 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПП2 ( $\Gamma_{\text{ц}}$ )
d11 = input('d11 = ');         % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПП1
d2 = input('d2 = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПЗ
d12 = input('d12 = ');         % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПП2

disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ kaiserord')
disp('%')
disp('%')
disp('% Для вывода параметров функции kaiserord нажмите <ENTER>')
pause
m = [1 0 1];                  % ВЕКТОР ЗНАЧЕНИЙ ИДЕАЛЬНОЙ АЧХ
f = [ft1 fk1 fk2 ft2];        % ВЕКТОР ГРАНИЧНЫХ ЧАСТОТ
ripple = [d11 d2 d12];        % ВЕКТОР МАКСИМАЛЬНО ДОПУСТИМЫХ ОТКЛОНЕНИЙ
[R,wc,beta,ftype] = kaiserord(f,m,ripple,Fs); % ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ОКНА
КАЙЗЕРА
disp(['R = ' num2str(R)])      % ОЦЕНКА ПОРЯДКА КИХ-ФИЛЬТРА
disp(['wc(1) = ' num2str(wc(1)) ' wc(2) = ' num2str(wc(2))]) % ВЕКТОР
НОРМИРОВАННЫХ ЧАСТОТ РАЗРЫВА
disp(['beta = ' num2str(beta)]) % ПАРАМЕТР ОКНА КАЙЗЕРА
disp(['ftype = ' char(ftype)]) % ТИП КИХ-ФИЛЬТРА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. СИНТЕЗ КИХ-ФИЛЬТРА РФ')
ORDER = 0; % ПРИЗНАК ОПТИМАЛЬНОСТИ ПОРЯДКА КИХ-ФИЛЬТРА: 0 — НЕОПТИМАЛЬНЫЙ;
1 — ОПТИМАЛЬНЫЙ
while ORDER==0;

```



```

disp('%')
disp('%')
disp('% Для синтеза КИХ-фильтра РФ нажмите <ENTER>')
pause
b4 = fir1(R,wc,ftype, kaiser(R+1,beta), 'noscale');      % КОЭФФИЦИЕНТЫ
КИХ-ФИЛЬТРА РФ
disp('%')
disp('%')
disp([' Синтезирован КИХ-фильтр РФ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для вывода ФАКТИЧЕСКИХ максимальных отклонений АЧХ')
disp('% в ПП1 (dp1), ПЗ (ds) и ПП2 (dp2) и ЗАДАННЫХ отклонений d11, d2 и d12')
disp('% нажмите <ENTER>')
pause
[dp1,ds,dp2] = check_stop(b4,ft1,fk1,fk2,ft2,Fs);      % ВЫЧИСЛЕНИЕ ФАКТИЧЕСКИХ
МАКСИМАЛЬНЫХ ПО МОДУЛЮ ОТКЛОНЕНИЙ В ПП1, ПЗ И ПП2
disp('%')
disp(['dp1=' num2str(dp1) '          ds = ' num2str(ds) '          dp2 = '
num2str(dp2)])
disp(['d11 = ' num2str(d11) '          d2 = ' num2str(d2) '          d12 = '
num2str(d12)])
disp('%')
disp('%')
disp('% Сравните ФАКТИЧЕСКИЕ отклонения с ЗАДАНЫМИ')
disp('%')
disp('% Если ПОРЯДОК соответствует МИНИМАЛЬНОМУ, введите 1')
disp('% Если НЕ соответствует, введите 0 и затем ПОРЯДОК R')
ORDER = input('---> ');
if ORDER==0
R = input('R = ');      % ПОРЯДОК КИХ-ФИЛЬТРА
while rem(R,2)~=0
disp('% Порядок фильтра выбран НЕПРАВИЛЬНО')
R = input('R = ');      % ПОРЯДОК КИХ-ФИЛЬТРА
end
end
end
disp('%')
disp([' Синтезирован РФ минимального порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. АНАЛИЗ ХАРАКТЕРИСТИК КИХ-ФИЛЬТРА РФ')
disp('%')
disp('%')

```

```

disp('% Для вывода ГРАФИКОВ ИХ, АЧХ и ФЧХ нажмите <ENTER>')
pause
figure('Name','Bandstop FIR Filter – Impulse Response, Magnitude,
Phase','NumberTitle','off')
plot_fir(R,b4,Fs)           % ПОСТРОЕНИЕ ГРАФИКОВ ИХ, АЧХ и ФЧХ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ОПИСАНИЕ СТРУКТУРЫ КИХ-ФИЛЬТРА РФ В ВИДЕ ОБЪЕКТА dfilt')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_bandstop = dfilt.dfssymfir(b4)      % ОБЪЕКТ dfilt – КИХ-ФИЛЬТР РФ
disp('%')
disp('%')
disp('% СИНТЕЗ КИХ-ФИЛЬТРА РФ ЗАВЕРШЕН')

```

11.4.5. Используемые внешние функции

В script-файлах lr_11_low, lr_11_high, lr_11_pass и lr_11_stop используются пять внешних функций.

- Внешняя функция check_low, предназначенная для вычисления фактических максимальных по модулю отклонений АЧХ КИХ-фильтра ФНЧ в ПП (dp) и ПЗ (ds):

```

function [dp,ds] = check_low(b,ft,fk,Fs)
% Проверка выполнения требований к АЧХ ФНЧ
%
% b – вектор коэффициентов КИХ-фильтра ФНЧ
% ft, fk – граничные частоты (Гц) ПП и ПЗ
% Fs – частота дискретизации (Гц)
%
% dp, ds – максимальные отклонения АЧХ в ПП и ПЗ
%
% fp, fs – векторы частот (Гц) для ПП и ПЗ (густая сетка)
% H – частотная характеристика
% a = [1] – коэффициент знаменателя передаточной функции
%
a = [1];
fp = 0:ft/1000:ft;
H = freqz(b,a,fp,Fs);
dp = max([max(abs(H))-1 1-min(abs(H))]);

```

```

fs = fk: (Fs/2-fk)/1000:Fs/2;
H = freqz(b,a,fs,Fs);
ds = max(abs(H));

```

- Внешняя функция **check_high**, предназначенная для вычисления фактических максимальных по модулю отклонений АЧХ КИХ-фильтра ФВЧ в ПЗ (**ds**) и ПП (**dp**):

```

function [ds,dp] = check_high(b,fk,ft,Fs)
% Проверка выполнения требований к АЧХ ФВЧ
%
% b – вектор коэффициентов КИХ-фильтра ФВЧ
% fk,ft – граничные частоты (Гц) ПЗ и ПП
% Fs – частота дискретизации (Гц)
%
% ds,dp – максимальные отклонения АЧХ в ПЗ и ПП
% fs,fp – векторы частот (Гц) для ПЗ и ПП (густая сетка)
% H – частотная характеристика
% a = [1] – коэффициент знаменателя передаточной функции
%
a = [1];
fs = 0:fk/1000:fk;
H = freqz(b,a,fs,Fs);
ds = max(abs(H));
fp = ft: (Fs/2-ft)/1000:Fs/2;
H = freqz(b,a,fp,Fs);
dp = max( [max(abs(H))-1 1-min(abs(H))] );

```

- Внешняя функция **check_pass**, предназначенная для вычисления фактических максимальных по модулю отклонений АЧХ КИХ-фильтра ПФ в ПЗ1 (**ds1**), ПП (**dp**) и ПЗ2 (**ds2**):

```

function [ds1,dp,ds2] = check_pass(b,fk1,ft1,ft2,fk2,Fs)
% Проверка выполнения требований к АЧХ ПФ
%
% b – вектор коэффициентов КИХ-фильтра ПФ
% fk1,ft1,ft2,fk2 – граничные частоты (Гц) ПЗ1,ПП и ПЗ2
% Fs – частота дискретизации (Гц)
%
% ds1,dp,ds2 – максимальные отклонения АЧХ в ПЗ1, ПП и ПЗ2
% fs1,fp,fs2 – векторы частот (Гц) для ПЗ1, ПП и ПЗ2 (густая сетка)
% H – частотная характеристика
% a=[1] – коэффициент знаменателя передаточной функции
%
a = [1];
fs1 = 0:fk1/1000:fk1;
H = freqz(b,a,fs1,Fs);
ds1 = max(abs(H));
fp = ft1: (ft2-ft1)/1000:ft2;

```

```

H = freqz(b,a,fp,Fs);
dp = max([max(abs(H))-1 1-min(abs(H))]);
fs2 = fk2:(Fs/2-fk2)/1000:F2/2;
H = freqz(b,a,fs2,Fs);
ds2 = max(abs(H));

```

- Внешняя функция `check_stop`, предназначенная для вычисления фактических максимальных по модулю отклонений АЧХ КИХ-фильтра РФ в ПП1 (`dp1`), ПЗ (`ds`) и ПП2 (`dp2`):

```

function [dp1,ds,dp2] = check_stop(b,ft1,fk1,fk2,ft2,Fs)
% Проверка выполнения требований к АЧХ РФ
%
% b — вектор коэффициентов КИХ-фильтра РФ
% ft1,fk1,fk2,ft2 — граничные частоты ПП1, ПЗ и ПП2
% Fs — частота дискретизации (Гц)
%
% dp1,ds,dp2 — максимальные отклонения АЧХ в ПП и ПЗ
% fp1,fs,fp2 — векторы частот (Гц) для ПП1, ПЗ и ПП2 (густая сетка)
% H — частотная характеристика
% a = [1] — коэффициент знаменателя передаточной функции
%
a = [1];
fp1 = 0:ft1/1000:ft1;
H = freqz(b,a,fp1,Fs);
dp1 = max([max(abs(H))-1 1-min(abs(H))]);
fs = fk1:(fk2-fk1)/1000:fk2;
H = freqz(b,a,fs,Fs);
ds = max(abs(H));
fp2 = ft2:(Fs/2-ft2)/1000:F2/2;
H = freqz(b,a,fp2,Fs);
dp2 = max([max(abs(H))-1 1-min(abs(H))]);

```

- Внешняя функция `plot_fir`, предназначенная для вывода графиков ИХ, АЧХ и ФЧХ КИХ-фильтра:

```

function plot_fir(R,b,Fs)
% Вывод графиков ИХ, АЧХ и ФЧХ КИХ-фильтра
%
% R — порядок КИХ-фильтра
% b — вектор коэффициентов КИХ-фильтра (ИХ КИХ-фильтра)
% Fs — частота дискретизации (Гц)
%
% a = [1] — коэффициент знаменателя передаточной функции
% n — вектор дискретного нормированного времени
% f — сетка частот (Гц) для расчета АЧХ и ФЧХ
% H — частотная характеристика
% MAG и PHASE — АЧХ и ФЧХ
%

```

```

a = [1];
n = 0:R;
subplot(3,1,1), stem(n,b,'fill','MarkerSize',3)
xlabel('n'), title('Impulse Response'), grid
f = 0:((Fs/2)/1000):Fs/2;
H = freqz(b,a,f,Fs);
MAG = abs(H);
PHASE = angle(H);
subplot(3,1,2), plot(f,MAG)
xlabel('f (Hz)'), title('MAGNITUDE'), grid
subplot(3,1,3), plot(f,PHASE)
xlabel('f (Hz)'), title('PHASE'), grid

```

11.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файлов для синтеза КИХ-фильтра ФНЧ методом окон с применением окна Кайзера, анализа его характеристик и моделирования процесса цифровой фильтрации.

Пункты самостоятельного задания включает в себя:

- 1С. Синтез КИХ-фильтра ФНЧ с произвольными требованиями к АЧХ (входные параметры function-файла).

Для проверки выполнения требований к АЧХ использовать function-файл `check_low` (см. разд. 11.4.5), который хранится на диске в папке `LAB_DSP\LAB_11`.

Вывести графики ИХ, АЧХ и ФЧХ с помощью function-файла `plot_fir` (см. разд. 11.4.5), который хранится на диске в папке `LAB_DSP\LAB_11`.

Выходным параметром function-файла является вектор коэффициентов КИХ-фильтра.

- 2С. Вычисление реакции КИХ-фильтра ФНЧ на воздействие в виде периодической последовательности с периодом $N = 64$:

$$x(n) = A_1 \cos\left(\frac{2\pi f_1}{f_d} n\right) + A_2 \cos\left(\frac{2\pi f_2}{f_d} n\right) = A_1 \cos(\hat{\omega}_1 n) + A_2 \cos(\hat{\omega}_2 n).$$

Входными параметрами function-файла являются:

- вектор коэффициентов КИХ-фильтра (см. п. 1С);
- частота дискретизации f_d (она должна совпадать с заданной в требованиях к АЧХ КИХ-фильтра);
- амплитуды гармоник A_1 и A_2 ;
- частоты гармоник f_1 и f_2 .

Вводимые значения частот должны быть согласованы с граничными частотами в требованиях к АЧХ КИХ-фильтра. Частота f_1 должна быть располо-

жена в ПП, а частота f_2 — в ПЗ. При этом отсутствие растекания спектра (см. п. 3С) гарантируется в том случае, если для частот f_1 и f_2 отношение (10.1) будет целым числом.

Реакцию КИХ-фильтра $y(n)$ вычислить с помощью функции `filter`.

Вывести графики воздействия и реакции КИХ-фильтра.

Выходными параметрами function-файла являются векторы отсчетов воздействия и реакции.

3С. Вычисление амплитудных спектров воздействия и реакции КИХ-фильтра ФНЧ.

Входными параметрами function-файла являются векторы отсчетов воздействия и реакции КИХ-фильтра.

Для вычисления амплитудных спектров воздействия и реакции использовать функцию `fft`.

Вывести графики амплитудных спектров.

11.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения пунктов задания, включая заполненную табл. 11.7, созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**), описания структур КИХ-фильтров в виде объектов `dfilt`, копируемые из окна **Command Window** (шрифт Courier New), и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

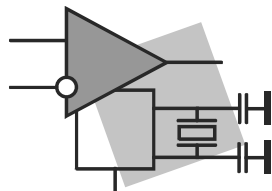
1. Дайте определение цифрового фильтра.
2. Перечислите основные этапы проектирования цифрового фильтра.
3. Запишите передаточную функцию КИХ-фильтра.
4. Дайте определение длины и порядка КИХ-фильтра.
5. Назовите основные особенности КИХ-фильтров.
6. При каком условии КИХ-фильтр будет иметь строго линейную ФЧХ?
7. В каких точках ФЧХ фильтра имеет скачок на π ?
8. Назовите признаки, по которым различают четыре типа КИХ-фильтров с ЛФЧХ.
9. Какие типы КИХ-фильтров с ЛФЧХ могут использоваться для синтеза фильтра методом окон?
10. Что входит в требования к АЧХ КИХ-фильтра?
11. Назовите основные свойства АЧХ и ФЧХ.
12. Что отображает структура ЦФ и чем определяется ее вид?

13. Назовите основные структуры КИХ-фильтров.
14. Перечислите основные этапы итерационной процедуры синтеза КИХ-фильтров методом окон.
15. Дайте определения окна и частоты разрыва.
16. Какой вид имеет АЧХ при синтезе КИХ-фильтров методом окон?
17. Назовите основное преимущество и недостаток метода окон.

11.7. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 13.
2. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов. — 2-е изд. — СПб.: БХВ-Петербург, 2005. — Глава 19.
3. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 6.

ГЛАВА 12



Синтез КИХ-фильтров методом наилучшей равномерной (чебышевской) аппроксимации

Цель работы: изучить процедуру синтеза КИХ-фильтров методом наилучшей равномерной (чебышевской) аппроксимации и овладеть программными средствами MATLAB для синтеза и анализа КИХ-фильтров; познакомиться с GUI FVTool (Filter Visualization Tool — средство визуализации фильтра).

12.1. Краткая теоретическая справка

Краткая теоретическая справка о КИХ-фильтрах с линейной ФЧХ приведена в разд. 11.1, 11.1.1—11.1.3.

12.1.1. Процедура синтеза КИХ-фильтров методом чебышевской аппроксимации

Метод наилучшей равномерной (чебышевской) аппроксимации (коротко — чебышевской аппроксимации) позволяет синтезировать оптимальный КИХ-фильтр¹.

Оптимальным называют КИХ-фильтр минимально возможного порядка R (11.2) при заданных требованиях к АЧХ.

Во избежание путаницы с порядком $R_{\min}$ КИХ-фильтра, синтезированного методом окон (см. разд. 11.1.4) при тех же требованиях к АЧХ, введем обозначение оптимального порядка R_{opt} :

$$R_{\text{opt}} < R_{\min}. \quad (12.1)$$

Коэффициенты КИХ-фильтра² определяются в результате поиска минимума модуля максимальной *взвешенной* ошибки аппроксимации (критерий Чебышева³) на *ин-*

¹ С теоретическими основами метода чебышевской аппроксимации можно познакомиться в [2, 3].

² Линейно связанные с коэффициентами аппроксимирующего тригонометрического полинома.

³ Называемый также наилучшим равномерным или минимаксным критерием.

тервале аппроксимации — совокупности полос пропускания (ПП) и задерживания (ПЗ) КИХ-фильтра.

Веса — числа, всегда большие единицы, — рассчитываются следующим образом:

- ❑ вес, равный единице, присваивается полосе с наибольшим максимально допустимым отклонением;
- ❑ веса в остальных полосах рассчитываются как отношение наибольшего максимально допустимого отклонения к максимально допустимому отклонению в данной полосе.

Согласно теореме Чебышева [2], минимум максимальной (по модулю) взвешенной ошибки аппроксимации $\delta_{\min \max}$ достигается в *точках альтернанса* — частотах, на которых *максимальное* (по модулю) *взвешенное* отклонение амплитудной функции (АЧХ равна ее модулю) от идеальной АЧХ *минимально* $\delta_{\min \max}$, одинаково и чередуется по знаку¹.

Число точек альтернанса взаимосвязано с порядком КИХ-фильтра и не может быть меньшим, чем представленное в табл. 12.1.

Таблица 12.1. Количество точек альтернанса и порядок КИХ-фильтра

Тип КИХ-фильтра	Число точек альтернанса <i>m</i>	Порядок фильтра <i>R</i>
Тип 1 (Type-1): порядок <i>R</i> — четный; ИХ <i>h(n)</i> — симметричная	$m = \frac{R}{2} + 2$	$R = 2m - 4$
Тип 2 (Type-2): порядок <i>R</i> — нечетный, ИХ <i>h(n)</i> — симметричная	$m = \frac{R - 1}{2} + 2$	$R = 2m - 3$
Тип 3 (Type-3): порядок <i>R</i> — четный; ИХ <i>h(n)</i> — антисимметричная и $h\left(\frac{R}{2}\right) = 0$	$m = \frac{R}{2} + 1$	$R = 2m - 2$
Тип 4 (Type-4): порядок <i>R</i> — нечетный; ИХ <i>h(n)</i> — антисимметричная	$m = \frac{R - 1}{2} + 2$	$R = 2m - 3$

Как уже говорилось (см. разд. 11.1.4), синтез КИХ-фильтра сводится к расчету его импульсной характеристики.

¹ Альтернанс (от фр. *alternative*) — чередование противоположных.

Процедура синтеза КИХ-фильтров методом чебышевской аппроксимации является *итерационной* и включает в себя следующие шаги:

1. Задание требований к АЧХ.

2. Оценка порядка фильтра R .

Оценкой порядка R называют начальное значение порядка в итерационной процедуре синтеза фильтра, которое определяется автоматически по эмпирической формуле на основании требований к АЧХ.

3. Расчет импульсной характеристики фильтра $h(n)$.

Расчет ИХ $h(n)$ производится с помощью численного метода, разработанного на основе обменного алгоритма Ремеза и известного в англоязычной литературе как алгоритм Паркса—Мак-Клиллена.

Импульсная характеристика может быть как *симметричной*, так и *антисимметричной*, поэтому необходимо следить за тем, на основе какого из четырех типов КИХ-фильтров может синтезироваться фильтр требуемой избирательности (см. табл. 11.1).

4. Проверка выполнения требований к АЧХ.

При синтезе в MATLAB программными средствами проверка выполнения требований к АЧХ заключается в сравнении максимальной (по модулю) взвешенной ошибки аппроксимации $\delta_{\min \max}$ с допустимым взвешенным отклонением $\delta_{\max}$ АЧХ от идеальной АЧХ, равным (см. разд. 11.1.2).

- для ФНЧ и ФВЧ:

$$\delta_{\max} = \max\{\delta_1, \delta_2\}; \quad (12.2)$$

- для ПФ:

$$\delta_{\max} = \max\{\delta_{21}, \delta_1, \delta_{22}\}; \quad (12.3)$$

- для РФ:

$$\delta_{\max} = \max\{\delta_{11}, \delta_2, \delta_{12}\}. \quad (12.4)$$

Возможны две ситуации.

- Требования к АЧХ не выполняются: $\delta_{\min \max} > \delta_{\max}$.

В этом случае следует *увеличить* порядок R и вернуться к пп. 3—4.

- Требования к АЧХ выполняются: $\delta_{\min \max} \leq \delta_{\max}$.

В этом случае следует *уменьшить* порядок R и вернуться к пп. 3—4.

В обоих случаях увеличение/уменьшение порядка R продолжается до тех пор, пока не будет найден оптимальный (минимальный) порядок R_{opt} , при котором выполняются требования к АЧХ.

5. Выбор структуры КИХ-фильтра (см. табл. 11.2).

12.1.2. Синтез КИХ-фильтров методом чебышевской аппроксимации в MATLAB

Синтез оптимальных КИХ-фильтров методом чебышевской аппроксимации выполняется с помощью функции:

```
[b,error,opt] = firpm(R,f0,m0,weight,ftype,{lgrid})
```

где R — порядок фильтра R (11.2); f_0 — вектор-столбец нормированных частот

$\hat{f} = \frac{f}{f_d/2}$ в основной полосе $[0; 1]$, включающий *левую* границу основной полосы

частот (0), *граничные частоты* ПП и ПЗ в порядке их следования слева направо и *правую* границу основной полосы (1); m_0 — вектор-столбец значений идеальной АЧХ на частотах вектора f_0 ; длины векторов m_0 и f_0 совпадают; $weight$ — вектор-столбец весов в ПП и ПЗ в порядке следования слева направо; $ftype$ — параметр, указывающий тип КИХ-фильтра и принимающий значения (см. табл. 11.1):

- 'hilbert' — для 3-го и 4-го типов и цифровых преобразователей Гильберта;
- 'differentiator' — для 3-го и 4-го типов и цифровых дифференциаторов;
- по умолчанию (если параметр отсутствует) — для 1-го и 2-го типов;
- ' ' (пробел) — тождественно отсутствию параметра $ftype$.

$lgrid$ — коэффициент плотности сетки частот (Density Factor); указывается элементом массива ячеек (см. разд. 3.1.4) в фигурных скобках и равен целому числу, большему 16-ти (по умолчанию — 16); с ростом $lgrid$ возрастает точность вычисления коэффициентов b , и вместе с тем — объем вычислений.

b — вектор коэффициентов передаточной функции (11.1) длины $N = R + 1$.

opt — массив записей (см. разд. 3.1.4) со следующими полями:

- $opt.fgrid$ — сетка нормированных частот (вектор) на интервале аппроксимации (совокупности ПП и ПЗ) в шкале нормированных частот $\hat{f}$; правая граница основной полосы частот, равная единице, не выводится;
- $opt.H$ — вектор значений комплексной частотной характеристики на сетке частот $opt.fgrid$;
- $opt.error$ — вектор отклонений АЧХ от идеальной на сетке частот $opt.fgrid$;
- $opt.des$ — вектор значений идеальной АЧХ на сетке частот $opt.fgrid$;
- $opt.wt$ — вектор весов на сетке частот $opt.fgrid$;
- $opt.iextr$ — вектор номеров элементов вектора $opt.fgrid$, соответствующих частотам альтернанса;
- $opt.fextr$ — вектор нормированных частот альтернанса.

$error$ — максимальная (по модулю) взвешенная ошибка аппроксимации $\delta_{\min \max}$ (см. разд. 12.1.1): $error = \max(abs(opt.error))$.

Оценка порядка R КИХ-фильтра для функции `firpm`, а также вычисление параметров `f0`, `m0`, `weight` производится по требованиям к АЧХ с помощью функции:

```
[R,f0,m0,weight] = firpmord(f,m,ripple,Fs)
```

где f — вектор граничных частот ПП и ПЗ в порядке их следования слева направо в шкале частот f (Гц) в основной полосе $[0; f_d/2]$; m — вектор значений идеальной АЧХ в порядке их следования слева направо; соблюдается условие `length(f) = 2*length(m)-2`; `ripple` — вектор максимально допустимых отклонений АЧХ в порядке их следования слева направо; F_s — частота дискретизации f_d (Гц); R — оценка порядка фильтра R с точностью до ± 2 .

Остальные параметры были определены ранее для функции `firpm`.

12.1.3. Описание требований к характеристике затухания в виде объекта *fdesign*

В MATLAB имеются средства синтеза КИХ- и БИХ-фильтров *непосредственно в виде объекта* `dfilt` (см. разд. 11.1.3). В этом случае требования задаются к характеристике затухания АЧХ (дБ) (11.6) и описываются в виде объекта `fdesign`:

```
Hs = fdesign.type(['sp1,sp2,...'],sp1,sp2,...,Fs)
```

где H_s — имя объекта `fdesign`; `fdesign` — тип объекта; `type` — функция, задающая конкретный тип избирательности ЦФ (табл. 12.2); `'sp1,sp2,...'` — список обязательных параметров функции `type`.

Список обязательных параметров строго регламентирован и соответствует требованиям к АЧХ (дБ). В табл. 12.3—12.6 приводятся списки обязательных параметров для различных функций `type`; в круглых скобках указаны те же параметры, которые используются при *выводе* свойств объекта `fdesign`.

$sp1, sp2, \dots$ — значения *обязательных* параметров в списке `'sp1,sp2,...'`. Принятый по умолчанию список параметров `'sp1,sp2,...'` может отсутствовать, однако его удобно оставлять для идентификации значений параметров.

F_s — частота дискретизации f_d (Гц).

Свойства объекта `fdesign` выводятся по его имени H_s и включают в себя список обязательных параметров функции `type` с их значениями.

Таблица 12.2. Функции *type* для частотно-избирательных ЦФ

Функция <i>type</i>	Тип избирательности ЦФ
<code>lowpass</code>	Lowpass Filter — ФНЧ
<code>highpass</code>	Highpass Filter — ФВЧ
<code>bandpass</code>	Bandpass Filter — ПФ
<code>bandstop</code>	Bandstop Filter — РФ

Таблица 12.3. Список параметров объекта `fdesign.lowpass`

Параметры функции <code>lowpass</code>	Требования к АЧХ (дБ) ФНЧ
<code>Fp (Fpass)</code>	f_{χ} — граничная частота ПП
<code>Fst (Fstop)</code>	f_k — граничная частота ПЗ
<code>Ap (Apass)</code>	$a_{\max}$ (дБ) — максимально допустимое затухание в ПП
<code>Ast (Astop)</code>	$a_{\min}$ (дБ) — минимально допустимое затухание в ПЗ

Таблица 12.4. Список параметров объекта `fdesign.highpass`

Параметры функции <code>highpass</code>	Требования к АЧХ (дБ) ФВЧ
<code>Fst (Fstop)</code>	f_k — граничная частота ПЗ
<code>Fp (Fpass)</code>	f_{χ} — граничная частота ПП
<code>Ast (Astop)</code>	$a_{\min}$ (дБ) — минимально допустимое затухание в ПЗ
<code>Ap (Apass)</code>	$a_{\max}$ (дБ) — максимально допустимое затухание в ПП

Таблица 12.5. Список параметров объекта `fdesign.bandpass`

Параметры функции <code>bandpass</code>	Требования к АЧХ (дБ) ПФ
<code>Fst1 (Fstop1)</code>	f_{-k} — граничная частота ПЗ1
<code>Fp1 (Fpass1)</code>	$f_{-\chi}$ — левая граничная частота ПП
<code>Fp2 (Fpass2)</code>	f_{χ} — правая граничная частота ПП
<code>Fst2 (Fstop2)</code>	f_k — граничная частота ПЗ2
<code>Ast1 (Astop1)</code>	$a_{1\min}$ (дБ) — минимально допустимое затухание в ПЗ1
<code>Ap (Apass)</code>	$a_{\max}$ (дБ) — максимально допустимое затухание в ПП
<code>Ast2 (Astop2)</code>	$a_{2\min}$ (дБ) — минимально допустимое затухание в ПЗ2

Таблица 12.6. Список параметров объекта `fdesign.bandstop`

Параметры функции <code>bandstop</code>	Требования к АЧХ (дБ) РФ
<code>Fp1 (Fpass1)</code>	$f_{-\chi}$ — граничная частота ПП1
<code>Fst1 (Fstop1)</code>	f_{-k} — левая граничная частота ПЗ
<code>Fst2 (Fstop2)</code>	f_k — правая граничная частота ПЗ
<code>Fp2 (Fpass2)</code>	f_{χ} — граничная частота ПП2
<code>Ap1 (Apass1)</code>	$a_{1\max}$ (дБ) — максимально допустимое затухание в ПП1
<code>Ast (Astop)</code>	$a_{\min}$ (дБ) — минимально допустимое затухание в ПЗ
<code>Ap2 (Apass2)</code>	$a_{2\max}$ (дБ) — максимально допустимое затухание в ПП2

12.1.4. Синтез КИХ-фильтров в виде объектов *dfilt* на основе объектов *fdesign*

При задании требований к АЧХ (дБ) в виде объекта `fdesign` для синтеза КИХ-фильтров в виде объекта `dfilt` используются функции, представленные в табл. 12.7. Отметим, что порядки КИХ-фильтров, синтезированных с помощью функций `kaiserwin` и `equiripple`, могут отличаться от соответствующих порядков КИХ-фильтров, синтезированных с помощью функций `firl` и `firpm`, что объясняется различием алгоритмов синтеза.

Таблица 12.7. Функции синтеза КИХ-фильтра в виде объекта `dfilt`

Функция	Метод синтеза
<code>kaiserwin</code>	Окон с использованием окна Кайзера
<code>equiripple</code>	Наилучшей равномерной (чебышевской) аппроксимации

Обобщенный формат функции синтеза КИХ-фильтра в виде объекта `dfilt` на основе объекта `fdesign` представлен двумя разновидностями:

```
Hf = function_fir(Hs)
Hf = design(Hs, 'function_fir')
```

где `function_fir` — имя конкретной функции из табл. 12.7; `Hs` — имя объекта `fdesign`; `Hf` — имя объекта `dfilt`.

По умолчанию выбирается прямая структура КИХ-фильтра (Direct-Form FIR). Для выбора прямой приведенной структуры (см. табл. 11.2) можно воспользоваться расширенным форматом функции синтеза КИХ-фильтра:

```
Hf = design(Hs, 'function_fir', 'FilterStructure', 'structure')
```

где *'structure'* — функция, задающая конкретную структуру объекта **Hf** (см. табл. 11.2).

Вычисление частотной (**H**) и импульсной (**h**) характеристик КИХ-фильтра, синтезированного в виде объекта **dfilt**, выполняется с помощью функций соответственно:

```
H = freqz(Hf, N)
```

```
h = impz(Hf)
```

где **N** — число точек (значений) частотной характеристики; в отсутствии параметра по умолчанию **N** = 512.

При выводе графиков АЧХ и ФЧХ в **N** точках в основной полосе значения частот в герцах задаются в виде вектора (где **Fs** — частота дискретизации):

```
f = 0 : ((Fs/2) / (N-1)) : Fs/2;
```

12.2. Содержание лабораторной работы

Содержание работы связано с синтезом КИХ-фильтров методом наилучшей равномерной (чебышевской) аппроксимации, описанием их структур и анализом характеристик с использованием программных средств MATLAB.

12.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файлов с именами **lr_12_low**, **lr_12_high**, **lr_12_pass** и **lr_12_stop** и function-файлов **plot_fir** и **MAG_fir**, которые хранятся на прилагаемом компакт-диске в папке **LAB_DSP\LAB_12**.

Перед выполнением работы необходимо сохранить путь к папке **LAB_12** по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 12.8—12.11 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$, и для КИХ-фильтров ФНЧ, ФВЧ, ПФ и РФ включают в себя:

- требования к АЧХ;
- требования к характеристике затухания АЧХ (дБ) (11.6) для ее описания в виде объекта **fdesign**. Значения допустимых затуханий рассчитаны по формулам (11.7)—(11.8).

На прилагаемом компакт-диске в папке **Tables\Tables_12** хранятся табл. 12.8—12.11 исходных данных, примеры их заполнения для $N_{бр} = 1$ и табл. 12.12 для п. 2 задания.

Таблица 12.8. Требования к АЧХ и АЧХ (дБ) ФНЧ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}}$	Fs =
f_χ	Граничная частота ПП	$f_\chi = \frac{f_d}{10} + 20N_{\text{бр}}$	ft =
f_k	Граничная частота ПЗ	$f_k = \frac{f_d}{10} + 250 + 25N_{\text{бр}}$	fk =
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$	d1 = 0.05
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$	d2 = 0.01
a_{max}	Максимально допустимое затухание в ПП (дБ)	$a_{\text{max}} = 0,4455$	Ap = 0.4455
a_{min}	Минимально допустимое затухание в ПЗ (дБ)	$a_{\text{min}} = 40$	Ast = 40

Таблица 12.9. Требования к АЧХ и АЧХ (дБ) ФВЧ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}}$	Fs =
f_k	Граничная частота ПЗ	$f_k = \frac{f_d}{10} + 20N_{\text{бр}}$	fk =
f_χ	Граничная частота ПП	$f_\chi = \frac{f_d}{10} + 250 + 25N_{\text{бр}}$	ft =
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$	d2 = 0.01
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$	d1 = 0.05
a_{min}	Минимально допустимое затухание в ПЗ (дБ)	$a_{\text{min}} = 40$	Ast = 40
a_{max}	Максимально допустимое затухание в ПП (дБ)	$a_{\text{max}} = 0,4455$	Ap = 0.4455

Таблица 12.10. Требования к АЧХ и АЧХ (дБ) ПФ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
$f_{\text{д}}$	Частота дискретизации	$f_{\text{д}} = 5000 + 100N_{\text{бр}}$	Fs =
f_{-k}	Граничная частота ПЗ1	$f_{-k} = \frac{f_{\text{д}}}{20} + 20N_{\text{бр}}$	fk1 =
$f_{-\chi}$	Граничная частота ПП1	$f_{-\chi} = \frac{f_{\text{д}}}{20} + 250 + 25N_{\text{бр}}$	ft1 =
f_{χ}	Граничная частота ПП2	$f_{\chi} = \frac{f_{\text{д}}}{4} + 25N_{\text{бр}}$	ft2 =
f_k	Граничная частота ПЗ2	$f_k = \frac{f_{\text{д}}}{4} + 250 + 30N_{\text{бр}}$	fk2 =
δ_{21}	Максимально допустимое отклонение в ПЗ1	$\delta_{21} = 0,01$	d21 = 0.01
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$	d1 = 0.05
δ_{22}	Максимально допустимое отклонение в ПЗ2	$\delta_{22} = 0,01$	d22 = 0.01
$a_{1\text{min}}$	Минимально допустимое затухание в ПЗ1 (дБ)	$a_{1\text{min}} = 40$	Ast1 = 40
a_{max}	Максимально допустимое затухание в ПП (дБ)	$a_{\text{max}} = 0,4455$	Ap = 0.4455
$a_{2\text{min}}$	Минимально допустимое затухание в ПЗ2 (дБ)	$a_{2\text{min}} = 40$	Ast2 = 40

Таблица 12.11. Требования к АЧХ и АЧХ (дБ) РФ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
$f_{\text{д}}$	Частота дискретизации	$f_{\text{д}} = 5000 + 100N_{\text{бр}}$	Fs =
$f_{-\chi}$	Граничная частота ПП1	$f_{-\chi} = \frac{f_{\text{д}}}{20} + 20N_{\text{бр}}$	ft1 =
f_{-k}	Граничная частота ПЗ1	$f_{-k} = \frac{f_{\text{д}}}{20} + 250 + 25N_{\text{бр}}$	fk1 =
f_k	Граничная частота ПЗ2	$f_k = \frac{f_{\text{д}}}{4} + 25N_{\text{бр}}$	fk2 =

Таблица 12.11 (окончание)

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_{χ}	Граничная частота ПП2	$f_{\chi} = \frac{f_d}{4} + 250 + 30N_{бр}$	ft2 =
δ_{11}	Максимально допустимое отклонение в ПП1	$\delta_{11} = 0,05$	d11 = 0.05
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$	d2 = 0.01
δ_{12}	Максимально допустимое отклонение в ПП2	$\delta_{12} = 0,05$	d12 = 0.05
a_{1max}	Максимально допустимое затухание в ПП1 (дБ)	$a_{1max} = 0,4455$	Ap1 = 0.4455
a_{min}	Минимально допустимое затухание в ПЗ (дБ)	$a_{min} = 40$	Ast = 40
a_{2max}	Максимально допустимое затухание в ПП2 (дБ)	$a_{2max} = 0,4455$	Ap2 = 0.4455

Задание на лабораторную работу заключается в синтезе КИХ-фильтров методом наилучшей равномерной (чебышевской) аппроксимации и анализе их характеристик и для каждого типа избирательности (ФНЧ, ФВЧ, ПФ или РФ) включает в себя выполнение следующих пунктов:

1. Ввод требований к АЧХ.
2. Вычисление оценки порядка КИХ-фильтра и значения весов в ПП и ПЗ.
Выведенные значения весов (weight) внести в табл. 12.12.
Пояснить:
 - какая функция используется для вычисления оценки порядка КИХ-фильтра и весов;
 - с какой целью рассчитывается оценка порядка КИХ-фильтра;
 - как рассчитываются веса в ПП и ПЗ.

Таблица 12.12. Результаты синтеза оптимальных КИХ-фильтров

Тип избирательности фильтра	Метод чебышевской аппроксимации		
	порядок фильтра κ	тип КИХ-фильтра	вектор весов weight
ФНЧ			
ФВЧ			
ПФ			
РФ			

3. Синтез оптимального КИХ-фильтра методом чебышевской аппроксимации.

Для синтеза КИХ-фильтра организовать цикл, в теле которого выполнить следующие действия:

- синтезировать КИХ-фильтр;

Тип КИХ-фильтра указывается с помощью параметра `ftype` в функции синтеза.

Для ФНЧ и РФ параметр `ftype` выбирается по умолчанию.

Для ФВЧ и ПФ тип КИХ-фильтра указывается следующими значениями значением параметра `ftype`:

□ ' ' (пробел) — для 1-го или 2-го типа;

□ 'hilbert' — для 3-го или 4-го типа;

- проверить выполнение требований к АЧХ;

Для проверки выполнения требований к АЧХ вывести и сравнить максимальную (по модулю) взвешенную ошибку аппроксимации $\delta_{\min \max}$ (идентификатор `error`) с допустимым взвешенным отклонением $\delta_{\max}$ (см. (12.2)—(12.4));

- по результатам проверки, увеличивая или уменьшая порядок КИХ-фильтра, определить его оптимальный порядок, при котором выполняются требования к АЧХ.

При увеличении/уменьшении порядка КИХ-фильтра необходимо учитывать соответствие между типом избирательности ЦФ и типом КИХ-фильтра (см. табл. 11.1).

Полученное в результате итерационной процедуры значение оптимального порядка R и тип КИХ-фильтра записать в табл. 12.12.

Пояснить:

- какая функция используется для синтеза КИХ-фильтра;
- какой из параметров данной функции соответствует коэффициентам передаточной функции КИХ-фильтра;
- смысл итерационной процедуры синтеза;
- чему равно заданное отклонение $\delta_{\min \max}$.

4. Анализ характеристик КИХ-фильтра.

Для вывода графиков использовать function-файл `plot_fir` (см. разд. 12.4.5).

Пояснить:

- вид ИХ;
- вид АЧХ в ПП и ПЗ (воспользуйтесь кнопкой **Zoom in** на панели инструментов);
- вид ФЧХ.

5. Вывод графика АЧХ оптимального КИХ-фильтра с отмеченными частотами альтернанса.

Для вывода графика использовать function-файл `MAG_fir` (см. разд. 12.4.5).

Пояснить:

- какие частоты называют частотами альтернанса;
- соответствие между количеством частот альтернанса на графике и порядком КИХ-фильтра.

6. Описание требований к АЧХ КИХ-фильтра в виде объекта `fdesign`.

Описать требования к характеристике затухания АЧХ (дБ) (11.6) в виде объектов `fdesign` с именами:

- `MAG_lowpass` — для ФНЧ;
- `MAG_highpass` — для ФВЧ;
- `MAG_bandpass` — для ПФ;
- `MAG_bandstop` — для РФ.

Пояснить, с какой целью создается объект `fdesign` и что в себя включает список его свойств.

7. Синтез КИХ-фильтра в виде объекта `dfilt` на основе объекта `fdesign`.

Синтезировать КИХ-фильтр с помощью функции `equiripple` со следующими именами объектов `dfilt`:

- `F_lowpass` — для ФНЧ;
- `F_highpass` — для ФВЧ;
- `F_bandpass` — для ПФ;
- `F_bandstop` — для РФ.

Сравнить порядок синтезированного КИХ-фильтра с порядком в табл. 12.12.

Пояснить:

- что отображает структура и чем определяется ее вид;
- свойства объекта `dfilt`.

8. Знакомство с GUI FVTool.

Обратиться к GUI FVTool по команде:

```
fvtool(Hd)
```

где `Hd` — имя объекта `dfilt`, и проанализировать характеристики синтезированных КИХ-фильтров.

12.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должны быть представлены табл. 12.8—12.11 исходных данных для своего номера бригады $N_{бр}$.

Для четырех типов избирательности КИХ-фильтра — ФНЧ, ФВЧ, ПФ и РФ — созданы четыре script-файла. Для *запуска* script-файла к нему необходимо обратиться по имени:

```
>> lr_12_low — ФНЧ
>> lr_12_high — ФВЧ
>> lr_12_pass — ПФ
>> lr_12_stop — РФ
```

Листинги данных script-файлов представлены в разд. 12.6.1—12.6.4.

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш <Ctrl>+<Break>.

При выполнении script-файла текущие окна с графиками *не закрывать*.

12.4.1. Синтез и анализ КИХ-фильтра ФНЧ

Листинг script-файла lr_12_low имеет вид:

```
>> type lr_12_low
script
clc
clear
disp('% ЛР №12. СИНТЕЗ КИХ-ФИЛЬТРА ФНЧ МЕТОДОМ ЧЕБЫШЕВСКОЙ АППРОКСИМАЦИИ')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ ФНЧ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ')
DATA=0;
while DATA==0;
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
ft = input('ft = ');           % ГРАНИЧНАЯ ЧАСТОТА ПП (Гц)
fk = input('fk = ');           % ГРАНИЧНАЯ ЧАСТОТА ПЗ (Гц)
d1 = input('d1 = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПП
d2 = input('d2 = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПЗ
Ap = input('Ap = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПП
Ast = input('Ast = ');         % МИНИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПЗ
```

```

disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ firpmord')
disp('%')
disp('%')
disp('% Для вывода ПОРЯДКА КИХ-фильтра (R) и ВЕСОВ в ПП и ПЗ (weight) нажмите <ENTER>')
pause
m = [1 0]; % ВЕКТОР ЗНАЧЕНИЙ ИДЕАЛЬНОЙ АЧХ
f = [ft fk]; % ВЕКТОР ГРАНИЧНЫХ ЧАСТОТ
ripple = [d1 d2]; % ВЕКТОР МАКСИМАЛЬНО ДОПУСТИМЫХ ОТКЛОНЕНИЙ
[R,f0,m0,weight] = firpmord(f,m,ripple,Fs); % ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ firpmord
disp('%')
disp(['R = ' num2str(R)]) % ОЦЕНКА ПОРЯДКА КИХ-ФИЛЬТРА
weight = [weight(1) weight(2)] % ВЕСА В ПП и ПЗ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. СИНТЕЗ КИХ-ФИЛЬТРА ФНЧ')
ORDER = 0; % ПРИЗНАК ОПТИМАЛЬНОСТИ ПОРЯДКА КИХ-ФИЛЬТРА: 0 — НЕОПТИМАЛЬНЫЙ;
1 — ОПТИМАЛЬНЫЙ
while ORDER==0;
disp('%')
disp('%')
disp('% Для синтеза КИХ-фильтра ФНЧ нажмите <ENTER>')
pause
[b1,error,opt] = firpm(R,f0,m0,weight); % КОЭФФИЦИЕНТЫ КИХ-ФИЛЬТРА ФНЧ b1
disp('%')
disp('%')
disp([' Синтезирован КИХ-фильтр ФНЧ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для вывода МАКСИМАЛЬНОЙ ВЗВЕШЕННОЙ ОШИБКИ АППРОКСИМАЦИИ error')
disp('% и ДОПУСТИМОГО ВЗВЕШЕННОГО ОТКЛОНЕНИЯ max{d1,d2} нажмите <ENTER>')

```

```

pause
disp('%')
disp(['error = ' num2str(error)]) % ФАКТИЧЕСКОЕ МАКСИМАЛЬНОЕ
ОТКЛОНЕНИЕ АЧХ ОТ ИДЕАЛЬНОЙ
disp(['max{d1,d2} = ' num2str(max(d1,d2))]) % ЗАДАННОЕ ОТКЛОНЕНИЕ max{d1,d2}
disp('%')
disp('%')
disp('% Сравните ФАКТИЧЕСКОЕ отклонение АЧХ с ЗАДАННЫМ')
disp('%')
disp('% Если ПОРЯДОК соответствует МИНИМАЛЬНОМУ, введите 1')
disp('% Если НЕ соответствует, введите 0 и затем ПОРЯДОК R')
ORDER = input('--> ');
if ORDER==0
R = input('R = '); % ПОРЯДОК КИХ-ФИЛЬТРА
end
end
disp('%')
disp([' Синтезирован оптимальный ФНЧ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. АНАЛИЗ ХАРАКТЕРИСТИК КИХ-ФИЛЬТРА ФНЧ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИХ, АЧХ И ФЧХ нажмите <ENTER>')
pause
figure('Name','Lowpass FIR Filter – Impulse Response, Magnitude,
Phase','NumberTitle','off')
plot_fir(R,b1,Fs) % ПОСТРОЕНИЕ ГРАФИКОВ ИХ, АЧХ и ФЧХ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5 ВЫВОД ГРАФИКА АЧХ ОПТИМАЛЬНОГО КИХ-ФИЛЬТРА С ОТМЕЧЕННЫМИ ЧАСТОТАМИ
АЛЬТЕРНАНСА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА АЧХ С ЧАСТОТАМИ АЛЬТЕРНАНСА нажмите <ENTER>')
pause
fextr = opt.fextr; % ВЕКТОР НОРМИРОВАННЫХ ЧАСТОТ АЛЬТЕРНАНСА
figure('Name','Lowpass Filter – Magnitude','NumberTitle','off')
MAG_fir(b1,opt.fextr,f0,m0,Fs) % ГРАФИК АЧХ С ЧАСТОТАМИ АЛЬТЕРНАНСА
disp('%')

```

```

disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ОПИСАНИЕ ТРЕБОВАНИЙ К АЧХ КИХ-фильтра ФНЧ В ВИДЕ ОБЪЕКТА fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА fdesign нажмите <ENTER>')
pause
MAG_lowpass = fdesign.lowpass('Fp,Fst,Ap,Ast',ft,fk,Ap,Ast,Fs) % ОБЪЕКТ fdesign
ДЛЯ ФНЧ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. СИНТЕЗ КИХ-ФИЛЬТРА В ВИДЕ ОБЪЕКТА dfilt НА ОСНОВЕ ОБЪЕКТА
fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_lowpass = equiripple(MAG_lowpass) % ФНЧ В ВИДЕ ОБЪЕКТА dfilt
disp('%')
disp('%')
disp('% СИНТЕЗ КИХ-ФИЛЬТРА ФНЧ ЗАВЕРШЕН')

```

12.4.2. Синтез и анализ КИХ-фильтра ФВЧ

Листинг script-файла lr_12_high имеет вид:

```

>> type lr_12_high
script
clc
clear
disp('% ЛР №12. СИНТЕЗ КИХ-ФИЛЬТРА ФВЧ МЕТОДОМ ЧЕБЫШЕВСКОЙ АППРОКСИМАЦИИ')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ ФВЧ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ')
DATA=0;
while DATA==0;
Nb = input('Nb = '); % НОМЕР БРИГАДЫ
Fs = input('Fs = '); % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)

```



```

fk = input ('fk = ');           % ГРАНИЧНАЯ ЧАСТОТА ПЗ ( $\Gamma_{\Pi}$ )
ft = input ('ft = ');           % ГРАНИЧНАЯ ЧАСТОТА ПП ( $\Gamma_{\Pi}$ )
d2 = input('d2 = ');            % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПЗ
d1 = input('d1 = ');            % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПП
Ast = input('Ast = ');          % МИНИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПЗ
Ap = input('Ap = ');            % МАКСИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПП

disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end

disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ firpmord')
disp('%')
disp('%')
disp('% Для вывода ПОРЯДКА КИХ-фильтра (R) и ВЕСОВ в ПЗ и ПП (weight) нажмите <ENTER>')
pause
m = [0 1];                      % ВЕКТОР ЗНАЧЕНИЙ ИДЕАЛЬНОЙ АЧХ
f = [fk ft];                     % ВЕКТОР ГРАНИЧНЫХ ЧАСТОТ
ripple = [d2 d1];                % ВЕКТОР МАКСИМАЛЬНО ДОПУСТИМЫХ ОТКЛОНЕНИЙ
[R,f0,m0,weight] = firpmord(f,m,ripple,Fs); % ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ
firpmord
disp('%')
disp(['R = ' num2str(R)])        % ОЦЕНКА ПОРЯДКА ФИЛЬТРА
weight = [weight(1) weight(2)]  % ВЕСА В ПЗ и ПП
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. СИНТЕЗ КИХ-ФИЛЬТРА ФВЧ')
ORDER = 0; % ПРИЗНАК ОПТИМАЛЬНОСТИ ПОРЯДКА КИХ-ФИЛЬТРА: 0 — НЕОПТИМАЛЬНЫЙ;
1 — ОПТИМАЛЬНЫЙ
ftype = ' ';                     % 1-й ТИП КИХ-ФИЛЬТРА ФВЧ ПО УМОЛЧАНИЮ
while ORDER==0;
disp('%')
disp('%')
disp('% Для синтеза КИХ-фильтра ФВЧ нажмите <ENTER>')
pause

```

```

[b2,error,opt] = firpm(R,f0,m0,weight,ftype);           % КОЭФИЦИЕНТЫ КИХ-ФИЛЬТРА
ФВЧ b2
disp('%')
disp('%')
disp([' Синтезирован КИХ-фильтр ФВЧ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для вывода МАКСИМАЛЬНОЙ ВЗВЕШЕННОЙ ОШИБКИ АППРОКСИМАЦИИ error')
disp('% и ДОПУСТИМОГО ВЗВЕШЕННОГО ОТКЛОНЕНИЯ max{d1,d2} нажмите <ENTER>')
pause
disp('%')
disp(['error = ' num2str(error)])                     % ФАКТИЧЕСКОЕ МАКСИМАЛЬНОЕ
ОТКЛОНЕНИЕ АЧХ ОТ ИДЕАЛЬНОЙ
disp(['max{d1,d2} = ' num2str(max(d1,d2))]) % ЗАДАННОЕ ОТКЛОНЕНИЕ max{d1,d2}
disp('%')
disp('%')
disp('% Сравните ФАКТИЧЕСКОЕ отклонение АЧХ с ЗАДАНЫМ')
disp('%')
disp('% Если ПОРЯДОК соответствует МИНИМАЛЬНОМУ, введите 1')
disp('% Если НЕ соответствует, введите 0 и затем ПОРЯДОК R')
disp('% и ТИП КИХ-фильтра ftype – пробел или hilbert В АПОСТРОФАХ')
ORDER = input('--> ');
if ORDER==0
R = input('R = ');                                     % ПОРЯДОК КИХ-ФИЛЬТРА
ftype = input('ftype = ');
while (ftype==' ') & (rem(R,2)~=0) | (ftype=='hilbert') & (rem(R,2)~=1)
% ПРОВЕРКА СООТВЕТСТВИЯ ПОРЯДКА И ТИПА КИХ-ФИЛЬТРА
disp('% ТИП КИХ-фильтра НЕ СООТВЕТСТВУЕТ ПОРЯДКУ')
ftype = input('ftype = ');
end
end
end
disp('%')
disp([' Синтезирован оптимальный ФВЧ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. АНАЛИЗ ХАРАКТЕРИСТИК КИХ-ФИЛЬТРА ФВЧ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИХ, АЧХ И ФЧХ нажмите <ENTER>')
pause
figure('Name','Highpass FIR Filter – Impulse Response, Magnitude,
Phase','NumberTitle','off')

```

```

plot_fir(R,b2,Fs)                                % ПОСТРОЕНИЕ ГРАФИКОВ ИХ, АЧХ и ФЧХ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ВЫВОД ГРАФИКА АЧХ ОПТИМАЛЬНОГО КИХ-ФИЛЬТРА С ОТМЕЧЕННЫМИ ЧАСТОТАМИ
АЛЬТЕРНАНСА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА АЧХ С ЧАСТОТАМИ АЛЬТЕРНАНСА нажмите <ENTER>')
pause
fextr = opt.fextr;                                % ВЕКТОР НОРМИРОВАННЫХ ЧАСТОТ АЛЬТЕРНАНСА
figure('Name','Highpass Filter - Magnitude','NumberTitle','off')
MAG_fir(b2,opt.fextr,f0,m0,Fs) % ГРАФИК АЧХ С ЧАСТОТАМИ АЛЬТЕРНАНСА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ОПИСАНИЕ ТРЕБОВАНИЙ К АЧХ КИХ-ФИЛЬТРА ФВЧ В ВИДЕ ОБЪЕКТА fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА fdesign нажмите <ENTER>')
pause
MAG_highpass = fdesign.highpass('Fst,Fp,Ast,Ap',fk,ft,Ast,Ap,Fs) % ОБЪЕКТ
fdesign ДЛЯ ФВЧ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. СИНТЕЗ КИХ-ФИЛЬТРА В ВИДЕ ОБЪЕКТА dfilt НА ОСНОВЕ ОБЪЕКТА
fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_highpass = equiripple(MAG_highpass) % ФВЧ В ВИДЕ ОБЪЕКТА dfilt
disp('%')
disp('%')
disp('% СИНТЕЗ КИХ-ФИЛЬТРА ФВЧ ЗАВЕРШЕН')

```

12.4.3. Синтез и анализ КИХ-фильтра ПФ

Листинг script-файла `lr_12_pass` имеет вид:

```
>> type lr_12_pass
script
clc
clear
disp('% ЛР №12. СИНТЕЗ КИХ-ФИЛЬТРА ПФ МЕТОДОМ ЧЕБЫШЕВСКОЙ АППРОКСИМАЦИИ')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ ПФ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ')
DATA=0;
while DATA==0;
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
fk1 = input('fk1 = ');          % ГРАНИЧНАЯ ЧАСТОТА ПЗ1 (Гц)
ft1 = input('ft1 = ');          % ГРАНИЧНАЯ ЧАСТОТА ПП (Гц)
ft2 = input('ft2 = ');          % ГРАНИЧНАЯ ЧАСТОТА ПП (Гц)
fk2 = input('fk2 = ');          % ГРАНИЧНАЯ ЧАСТОТА ПЗ2 (Гц)
d21 = input('d21 = ');          % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПЗ1
d1 = input('d1 = ');            % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПП
d22 = input('d22 = ');          % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПЗ2
Ast1 = input('Ast1 = ');        % МИНИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПЗ1
Ap = input('Ap = ');            % МАКСИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПП
Ast2 = input('Ast2 = ');        % МИНИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПЗ2
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ firpmord')
disp('%')
disp('%')
disp('% Для вывода ПОРЯДКА КИХ-фильтра (R) и ВЕСОВ в ПЗ1, ПП и ПЗ2 (weight)')
disp('% нажмите <ENTER>')
pause
m = [0 1 0];                  % ВЕКТОР ЗНАЧЕНИЙ ИДЕАЛЬНОЙ АЧХ
f = [fk1 ft1 ft2 fk2];        % ВЕКТОР ГРАНИЧНЫХ ЧАСТОТ
```

```

ripple = [d21 d1 d22];      % ВЕКТОР МАКСИМАЛЬНО ДОПУСТИМЫХ ОТКЛОНЕНИЙ
[R,f0,m0,weight] = firpmord(f,m,ripple,Fs); % ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ
firpmord
disp('%')
disp(['R = ' num2str(R)])    % ОЦЕНКА ПОРЯДКА ФИЛЬТРА
weight = [weight(1) weight(2) weight(3)] % ВЕСА В ПЗ и ПП
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. СИНТЕЗ КИХ-ФИЛЬТРА ПФ')
ORDER = 0; % ПРИЗНАК ОПТИМАЛЬНОСТИ ПОРЯДКА КИХ-ФИЛЬТРА: 0 — НЕОПТИМАЛЬНЫЙ;
1 — ОПТИМАЛЬНЫЙ
ftype = ' '; % 1-Й ИЛИ 2-Й ТИП ТИП КИХ-ФИЛЬТРА ФВЧ ПО УМОЛЧАНИЮ
while ORDER==0;
disp('%')
disp('%')
disp('% Для синтеза КИХ-фильтра ПФ нажмите <ENTER>')
pause
[b3,error,opt] = firpm(R,f0,m0,weight,ftype); % КОЭФФИЦИЕНТЫ КИХ-ФИЛЬТРА ПФ
b3
disp('%')
disp('%')
disp([' Синтезирован КИХ-фильтр ПФ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для вывода МАКСИМАЛЬНОЙ ВЗВЕШЕННОЙ ОШИБКИ АППРОКСИМАЦИИ error')
disp('% и ДОПУСТИМОГО ВЗВЕШЕННОГО ОТКЛОНЕНИЯ max{d21,d2,d22} нажмите <ENTER>')
pause
disp('%')
disp(['error = ' num2str(error)]) % ФАКТИЧЕСКОЕ МАКСИМАЛЬНОЕ ОТКЛОНЕНИЕ
АЧХ ОТ ИДЕАЛЬНОЙ
disp(['max{d21,d1,d22} = ' num2str(max([d21,d1,d22]))]) % ЗАДАННОЕ ОТКЛОНЕНИЕ
max{d21,d1,d22}
disp('%')
disp('%')
disp('% Сравните ФАКТИЧЕСКОЕ отклонение АЧХ с ЗАДАНЫМ')
disp('%')
disp('%')
disp('% Если ПОРЯДОК соответствует МИНИМАЛЬНОМУ, введите 1')
disp('% Если НЕ соответствует, введите 0 и затем ПОРЯДОК R')
disp('% и ТИП КИХ=фильтра ftype — пробел или hilbert В АПОСТРОФАХ')
disp('% ОПТИМАЛЬНЫЙ ПФ выбирается среди ВСЕХ ЧЕТЫРЕХ ТИПОВ КИХ-фильтров!')
ORDER = input('--> ');
if ORDER==0
R = input('R = '); % ПОРЯДОК КИХ-ФИЛЬТРА

```

```

ftype = input('ftype = ');
end
end
disp('%')
disp([' Синтезирован оптимальный ПФ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. АНАЛИЗ ХАРАКТЕРИСТИК КИХ-ФИЛЬТРА ПФ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИХ, АЧХ И ФЧХ нажмите <ENTER>')
pause
figure('Name','Bandpass FIR Filter – Impulse Response, Magnitude,
Phase','NumberTitle','off')
plot_fir(R,b3,Fs) % ПОСТРОЕНИЕ ГРАФИКОВ ИХ, АЧХ и ФЧХ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ВЫВОД ГРАФИКА АЧХ ОПТИМАЛЬНОГО КИХ-ФИЛЬТРА С ОТМЕЧЕННЫМИ ЧАСТОТАМИ
АЛЬТЕРНАНСА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА АЧХ С ЧАСТОТАМИ АЛЬТЕРНАНСА нажмите <ENTER>')
pause
fextr = opt.fextr; % ВЕКТОР НОРМИРОВАННЫХ ЧАСТОТ АЛЬТЕРНАНСА
figure('Name','Bandpass Filter – Magnitude','NumberTitle','off')
MAG_fir(b3,opt.fextr,f0,m0,Fs) % ГРАФИК АЧХ С ЧАСТОТАМИ АЛЬТЕРНАНСА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ОПИСАНИЕ ТРЕБОВАНИЙ К АЧХ КИХ-фильтра ПФ В ВИДЕ ОБЪЕКТА fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА fdesign нажмите <ENTER>')
pause
MAG_bandpass = fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2',fk1,ft1,ft2,
fk2,Ast1,Ap,Ast2,Fs) % ОБЪЕКТ fdesign ДЛЯ ПФ
disp('%')

```

```

disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. СИНТЕЗ КИХ-ФИЛЬТРА В ВИДЕ ОБЪЕКТА dfilt НА ОСНОВЕ ОБЪЕКТА
fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_bandpass = equiripple(MAG_bandpass)      % ПФ В ВИДЕ ОБЪЕКТА dfilt
disp('%')
disp('%')
disp('% СИНТЕЗ КИХ-ФИЛЬТРА ПФ ЗАВЕРШЕН')

```

12.4.4. Синтез и анализ КИХ-фильтра РФ

Листинг script-файла `lr_12_stop` имеет вид:

```

>> type lr_12_stop
script
clc
clear
disp('% ЛР №12. СИНТЕЗ КИХ-ФИЛЬТРА РФ МЕТОДОМ ЧЕБЫШЕВСКОЙ АППРОКСИМАЦИИ')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ РФ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ')
DATA=0;
while DATA==0;
Nb = input('Nb = ');      % НОМЕР БРИГАДЫ
Fs = input('Fs = ');      % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
ft1 = input('ft1 = ');    % ГРАНИЧНАЯ ЧАСТОТА ПП1 (Гц)
fk1 = input('fk1 = ');    % ГРАНИЧНАЯ ЧАСТОТА ПЗ (Гц)
fk2 = input('fk2 = ');    % ГРАНИЧНАЯ ЧАСТОТА ПЗ (Гц)
ft2 = input('ft2 = ');    % ГРАНИЧНАЯ ЧАСТОТА ПП2 (Гц)
d11 = input('d11 = ');    % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПП1
d2 = input('d2 = ');      % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПЗ
d12 = input('d12 = ');    % МАКСИМАЛЬНО ДОПУСТИМОЕ ОТКЛОНЕНИЕ В ПП2
Ap1 = input('Ap1 = ');    % МАКСИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПП1
Ast = input('Ast = ');    % МИНИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПЗ
Ap2 = input('Ap2 = ');    % МАКСИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПП2
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')

```

```

DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ firpmord')
disp('%')
disp('%')
disp('% Для вывода ПОРЯДКА КИХ-фильтра (R) и ВЕСОВ в ПП1, ПЗ и ПП2 (weight)
нажмите <ENTER>')
pause
m = [1 0 1]; % ВЕКТОР ЗНАЧЕНИЙ ИДЕАЛЬНОЙ АЧХ
f = [ft1 fk1 fk2 ft2]; % ВЕКТОР ГРАНИЧНЫХ ЧАСТОТ
ripple = [d11 d2 d12]; % ВЕКТОР МАКСИМАЛЬНО ДОПУСТИМЫХ ОТКЛОНЕНИЙ
[R,f0,m0,weight] = firpmord(f,m,ripple,Fs); % ВЫЧИСЛЕНИЕ ПАРАМЕТРОВ ФУНКЦИИ
firpmord
disp('%')
disp(['R = ' num2str(R)]) % ОЦЕНКА ПОРЯДКА ФИЛЬТРА
weight = [weight(1) weight(2) weight(3)] % ВЕСА В ПП и ПЗ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. СИНТЕЗ КИХ-ФИЛЬТРА РФ')
ORDER = 0; % ПРИЗНАК ОПТИМАЛЬНОСТИ ПОРЯДКА КИХ-ФИЛЬТРА: 0 — НЕОПТИМАЛЬНЫЙ;
1 — ОПТИМАЛЬНЫЙ
while ORDER==0;
disp('%')
disp('%')
disp('% Для синтеза КИХ-фильтра РФ нажмите <ENTER>')
pause
[b4,error,opt] = firpm(R,f0,m0,weight); % КОЭФФИЦИЕНТЫ КИХ-ФИЛЬТРА РФ b4
disp('%')
disp('%')
disp([' Синтезирован КИХ-фильтр РФ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для вывода МАКСИМАЛЬНОЙ ВЗВЕШЕННОЙ ОШИБКИ АППРОКСИМАЦИИ error')
disp('% и ДОПУСТИМОГО ВЗВЕШЕННОГО ОТКЛОНЕНИЯ max{d11,d2,d12} нажмите <ENTER>')
pause
disp('%')
disp(['error = ' num2str(error)]) % ФАКТИЧЕСКОЕ МАКСИМАЛЬНОЕ ОТКЛОНЕНИЕ
АЧХ ОТ ИДЕАЛЬНОЙ

```



```

disp(['max{d11,d2,d12} = ' num2str(max([d11,d2,d12]))]) % ЗАДАННОЕ ОТКЛОНЕНИЕ
max{d11,d2,d12}
disp('%')
disp('%')
disp('% Сравните ФАКТИЧЕСКОЕ отклонение АЧХ с ЗАДАНЫМ')
disp('%')
disp('% Если ПОРЯДОК соответствует МИНИМАЛЬНОМУ, введите 1')
disp('% Если НЕ соответствует, введите 0 и затем ПОРЯДОК R')
ORDER = input('--> ');
if ORDER==0
R = input('R = '); % ПОРЯДОК КИХ-ФИЛЬТРА
while rem(R,2)~=0
disp('% Порядок фильтра выбран НЕПРАВИЛЬНО')
R = input('R = '); % ПОРЯДОК КИХ-ФИЛЬТРА
end
end
disp('%')
disp([' Синтезирован оптимальный РФ порядка R = ' num2str(R)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. АНАЛИЗ ХАРАКТЕРИСТИК КИХ-ФИЛЬТРА РФ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИХ, АЧХ И ФЧХ нажмите <ENTER>')
pause
figure('Name','Bandstop FIR Filter – Impulse Response, Magnitude,
Phase','NumberTitle','off')
plot_fir(R,b4,Fs) % ПОСТРОЕНИЕ ГРАФИКОВ ИХ, АЧХ и ФЧХ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ВЫВОД ГРАФИКА АЧХ ОПТИМАЛЬНОГО КИХ-ФИЛЬТРА С ОТМЕЧЕННЫМИ ЧАСТОТАМИ
АЛЬТЕРНАНСА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА АЧХ С ЧАСТОТАМИ АЛЬТЕРНАНСА нажмите <ENTER>')
pause
fextr = opt.fextr; % ВЕКТОР НОРМИРОВАННЫХ ЧАСТОТ АЛЬТЕРНАНСА
figure('Name','Bandstop Filter – Magnitude','NumberTitle','off')

```

```

MAG_fir(b4,opt.fextr,f0,m0,Fs)  % ГРАФИК АЧХ С ЧАСТОТАМИ АЛЬТЕРНАНСА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ОПИСАНИЕ ТРЕБОВАНИЙ К АЧХ КИХ-фильтра РФ В ВИДЕ ОБЪЕКТА fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА fdesign нажмите <ENTER>')
pause
MAG_bandstop = fdesign.bandstop('Fp1,Fst1,Fst2,Fp2,Ap1,Ast,Ap2',ft1,fk1,fk2,
ft2,Ap1,Ast,Ap2,Fs)          % ОБЪЕКТ fdesign ДЛЯ РФ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. СИНТЕЗ КИХ-ФИЛЬТРА В ВИДЕ ОБЪЕКТА dfilt НА ОСНОВЕ ОБЪЕКТА
fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_bandstop = equiripple(MAG_bandstop)  % РФ В ВИДЕ ОБЪЕКТА dfilt
disp('%')
disp('%')
disp('% СИНТЕЗ КИХ-ФИЛЬТРА РФ ЗАВЕРШЕН')

```

12.4.5. Используемые внешние функции

В script-файлах `lr_12_low`, `lr_12_high`, `lr_12_pass` и `lr_12_stop` используются две внешние функции:

- внешняя функция `plot_fir`, предназначенная для вывода графиков ИХ, АЧХ и ФЧХ КИХ-фильтра (см. разд. 11.4.5);
- внешняя функция `MAG_fir`, предназначенная для вывода графика АЧХ оптимального КИХ-фильтра с отмеченными частотами альтернанса и идеальной АЧХ:

```

function MAG_fir(b,fextr,f0,m0,Fs)
% Вывод графика АЧХ КИХ-фильтра с точками альтернанса и идеальной АЧХ
%
% b — вектор коэффициентов КИХ-фильтра
% fextr — нормированные частоты альтернанса: fextr=opt.fextr
% f0 — вектор значений нормированных граничных частот ПП и ПЗ

```

```

% m0 — вектор значений идеальной АЧХ на частотах f0
% Fs — частота дискретизации (Гц)
%
% a=[1] — коэффициент знаменателя передаточной функции
% f — сетка частот (Гц) для расчета АЧХ
% H — частотная характеристика КИХ-фильтра
% MAG — АЧХ
% ff — вектор значений граничных частот ПП и ПЗ (Гц)
% fa — вектор частот альтернанса (Гц)
% Ha — частотная характеристика на частотах альтернанса
% ALT — АЧХ на частотах альтернанса
%
a = [1];
f = 0:((Fs/2)/1000):Fs/2;
H = freqz(b,a,f,Fs);
MAG = abs(H);
plot(f,MAG), xlabel('f(Hz)'), title('MAGNITUDE'), grid
hold on
ff = f0.*(Fs/2);
plot(ff,m0,'-','LineWidth',2), legend('MAGNITUDE','IDEAL MAGNITUDE',0)
hold on
fa = (fextr).*(Fs/2);
Ha = freqz(b,a,fa,Fs);
ALT = abs(Ha);
stem(fa,ALT,':','fill','MarkerSize',5)

```

12.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файлов для синтеза КИХ-фильтра ФНЧ методом чебышевской аппроксимации, анализа его характеристик и моделирования процесса цифровой фильтрации.

Пункты самостоятельного задания включают в себя:

- 1С. Синтез оптимального КИХ-фильтра ФНЧ и произвольными требованиями к АЧХ (входные параметры function-файла).

Для проверки выполнения требований к АЧХ вывести и сравнить максимальную (по модулю) взвешенную ошибку аппроксимации $\delta_{\min \max}$ с допустимым взвешенным отклонением $\delta_{\max}$ (см. (12.2)—(12.4)).

Вывести графики ИХ, АЧХ и ФЧХ с помощью function-файла `plot_fir` (см. разд. 11.4.5), который хранится на диске в папке `LAB_DSP\LAB_12`.

Вывести график АЧХ оптимального КИХ-фильтра с отмеченными частотами альтернанса с помощью function-файла `MAG_fir` (см. разд. 12.4.5), который хранится на диске в папке `LAB_DSP\LAB_12`.

Выходным параметром function-файла является вектор коэффициентов КИХ-фильтра.

- 2С. Дублирует п. 2С в *разд. 11.5* для оптимального КИХ-фильтра (см. п. 1С).
- 3С. Дублирует п. 2С в *разд. 11.5* для оптимального КИХ-фильтра (см. п. 1С).
- 4С. Синтез КИХ-фильтра в виде объекта `dfilt` с произвольными требованиями к характеристике затухания (входные параметры `function`-файла), задаваемыми в виде объекта `fdesign`.

Выходным параметром `function`-файла является имя объекта `dfilt`.

12.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения пунктов задания, включая заполненную табл. 12.12, созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**), описания структур КИХ-фильтров в виде объектов `dfilt`, копируемые из окна **Command Window** (шрифт Courier New), и ответы на поставленные вопросы (шрифт Times New Roman).

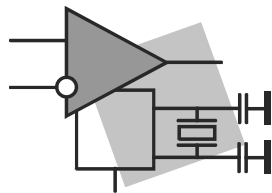
Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. Дайте определение оптимального КИХ-фильтра.
2. Запишите передаточную функцию КИХ-фильтра.
3. Дайте определение длины и порядка КИХ-фильтра.
4. При каком условии КИХ-фильтр будет иметь строго линейную ФЧХ?
5. В каких точках ФЧХ КИХ-фильтра имеет скачок на π ?
6. Назовите признаки, по которым различают четыре типа КИХ-фильтров с ЛФЧХ.
7. Что входит в требования к АЧХ КИХ-фильтра?
8. Дайте определение характеристики затухания.
9. Что входит в требования к характеристике затухания.
10. Назовите основные свойства АЧХ и ФЧХ.
11. Что отображает структура ЦФ и чем определяется ее вид?
12. Назовите основные структуры КИХ-фильтров.
13. Перечислите основные этапы итерационной процедуры синтеза КИХ-фильтров методом чебышевской аппроксимации.
14. С какой целью вводятся веса и как они рассчитываются?
15. Какой вид имеет АЧХ при синтезе КИХ-фильтров методом чебышевской аппроксимации?
16. Дайте определение точек альтернанса.
17. Назовите основное преимущество синтеза КИХ-фильтров методом чебышевской аппроксимации.

12.7. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Главы 13—14.
2. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов. — 2-е изд. — СПб.: БХВ-Петербург, 2005. — Глава 20.
3. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 6.

ГЛАВА 13



Синтез БИХ-фильтров методом билинейного Z-преобразования

Цель работы: изучить процедуру синтеза БИХ-фильтров методом билинейного Z-преобразования и овладеть программными средствами MATLAB для синтеза и анализа БИХ-фильтров; познакомиться с GUI FVTool (Filter Visualization Tool — средство визуализации фильтра).

13.1. Краткая теоретическая справка

БИХ-фильтр описывается передаточной функцией общего вида (8.4):

$$H(z) = \frac{\sum_{i=0}^{N-1} b_i z^{-i}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}} \quad (13.1)$$

и при $(N-1) \leq (M-1)$ (по умолчанию) имеет *порядок* R , равный $R = (M-1)$.

БИХ-фильтры характеризуются следующими особенностями:

- ☐ нелинейной ФЧХ;
- ☐ необходимостью проверки на устойчивость.

Оптимальным называют БИХ-фильтр минимального порядка R при заданных требованиях к АЧХ.

13.1.1. Задание требований к характеристике затухания

Для БИХ-фильтров требования задаются к *характеристике затухания* АЧХ (дБ) (11.6) в *основной полосе частот* $[0; f_d/2]$ и включают в себя:

- ☐ частоту дискретизации f_d (Гц);
- ☐ граничные частоты полос пропускания (ПП) и полос задерживания (ПЗ), такие же, как для КИХ-фильтров (см. разд. 11.1.2):

- f_χ — граничная частота ПП для ФНЧ и ФВЧ;
- f_k — граничная частота ПЗ для ФНЧ и ФВЧ;
- $f_{-\chi}$, f_χ — левая и правая граничные частоты ПП для ПФ и РФ;
- f_{-k} , f_k — левая и правая граничные частоты ПЗ для ПФ и РФ;

□ допустимые отклонения от $\hat{A}(f)$ (дБ) (11.6) (см. разд. 11.1.2):

- $a_{\max}$ (дБ) — максимально допустимое затухание в ПП (для ФНЧ, ФВЧ и ПФ);
- $a_{\min}$ (дБ) — минимально допустимое затухание в ПЗ (для ФНЧ, ФВЧ и РФ);
- $a_{1\max}$ (дБ) — максимально допустимое затухание в ПП1 (для РФ);
- $a_{2\max}$ (дБ) — максимально допустимое затухание в ПП2 (для РФ);
- $a_{1\min}$ (дБ) — минимально допустимое затухание в ПЗ1 (для ПФ);
- $a_{2\min}$ (дБ) — минимально допустимое затухание в ПЗ2 (для ПФ).

13.1.2. Структуры БИХ-фильтров

Структура (структурная схема) ЦФ отображает алгоритм вычисления реакции по разностному уравнению и определяется видом *передаточной функции*.

В MATLAB структура БИХ-фильтра описывается в виде объекта `dfilt`:

`Hd = dfilt.structure(input1,input2)`

где `Hd` — имя объекта `dfilt`; `dfilt` — тип объекта; `structure` — функция, задающая конкретную структуру объекта `Hd` (табл. 13.1); `input1`, `input2` — параметры функции `structure`.

Для БИХ-фильтров свойства объекта `dfilt` с именем `Hd` зависят от выбранной структуры. Для *прямой структуры* БИХ-фильтра они включают в себя:

- `FilterStructure` — структура КИХ-фильтра;
- `Arithmetic` — форма представления данных;
- `Numerator` — коэффициенты числителя передаточной функции (12.1);
- `Denominator` — коэффициенты знаменателя передаточной функции (12.1);
- `PersistentMemory` — начальные условия при вычислении реакции; значение `false` соответствует ННУ (см. разд. 8.1).

Каскадной структуре из биквадратных звеньев соответствует представление передаточной функции в виде произведения (8.9):

$$H(z) = G \prod_{k=1}^L \frac{1 + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}, \quad (13.2)$$

где звенья имеют одну из прямых структур, представленных на рис. 8.1.

В каскадной структуре БИХ-фильтра свойства `Numerator` и `Denominator` заменяются свойствами:

□ `sosMatrix` — матрица коэффициентов в виде (8.12):

$$\begin{bmatrix} 1 & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ 1 & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ \\ 1 & b_{1L} & b_{2L} & 1 & a_{1L} & a_{2L} \end{bmatrix};$$

(13.3)

□ `ScaleValues` — вектор коэффициентов усиления, элементы которого равны:

- первый — коэффициенту усиления на входе первого звена, т. е. на *входе структуры*;
- второй — коэффициенту усиления на входе второго звена и т. д.;
- последний — коэффициенту усиления на выходе последнего звена, т. е. на *выходе структуры*.

Таблица 13.1. Функции `structure` и структуры БИХ-фильтров

Функция <i>structure</i>	Параметры функции <i>structure</i>	Структура БИХ-фильтра
<code>df1</code>	<code>b</code> , <code>a</code> — векторы коэффициентов передаточной функции (13.1)	Direct-Form I (прямая)
<code>df1t</code>		Direct-Form I Transposed (прямая транспонированная)
<code>df2</code>		Direct-Form II (прямая каноническая)
<code>df2t</code>		Direct-Form II Transposed (прямая каноническая транспонированная)
<code>df1sos</code>	<code>s</code> — матрица коэффициентов в виде (13.3); <code>G</code> — коэффициент усиления передаточной функции (13.2)	Direct-Form I, Second-order sections (SOS) (каскадная из звеньев 2-го порядка с прямой структурой)
<code>df1tsos</code>		Direct-Form I Transposed, Second-order sections (SOS) (каскадная из звеньев 2-го порядка с прямой транспонированной структурой)
<code>df2sos</code>		Direct-form II, Second-order sections (SOS) (каскадная из звеньев 2-го порядка с прямой канонической структурой)

13.1.3. Процедура синтеза БИХ-фильтров методом билинейного Z -преобразования

Синтез БИХ-фильтра заключается в расчете его передаточной функции.

Метод билинейного Z -преобразования, позволяющий синтезировать оптимальный БИХ-фильтр, основан на использовании аналогового фильтра-прототипа (АФП)¹.

Процедура синтеза БИХ-фильтра на основе АФП включает в себя следующие шаги:

1. Задание требований к характеристике затухания АЧХ (дБ) (11.6) БИХ-фильтра.
2. Формирование требований к АЧХ (дБ) АФП.

Граничные частоты АФП Ω связаны с граничными частотами БИХ-фильтра ω нелинейной зависимостью:

$$\Omega = \frac{2}{T} \operatorname{tg} \frac{\omega T}{2},$$

которая в шкале частот в герцах соответствует зависимости между частотами АФП F и БИХ-фильтра f :

$$F = \frac{f_d}{\pi} \operatorname{tg} \frac{\pi f}{f_d}. \quad (13.4)$$

3. Выбор типа БИХ-фильтра.

Подобно АФП, четырем видам аппроксимирующих функций соответствуют четыре типа БИХ-фильтров:

- Баттерворта (Butterworth) — с АЧХ, максимально плоской в ПП и монотонной в ПЗ;
- Чебышева I рода (Chebyshev Type I) — с АЧХ, равноволновой в ПП и монотонной в ПЗ;
- Чебышева II рода (Chebyshev Type II) — с АЧХ, максимально плоской в ПП и равноволновой в ПЗ;
- Золотарева—Кауэра (Elliptic — эллиптический) — с АЧХ, равноволновой в ПП и ПЗ.

4. Расчет передаточной функции АФП $H_a(p)$.
5. Преобразование передаточной функции АФП $H_a(p)$ в передаточную функцию БИХ-фильтра $H(z)$ на основе формулы билинейного Z -преобразования:

$$p = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}.$$

6. Выбор структуры БИХ-фильтра.

¹ С теоретическими основами метода билинейного Z -преобразования можно познакомиться в [2, 3].

13.1.4. Синтез аналоговых фильтров в MATLAB

Синтез частотно-избирательных аналоговых фильтров Баттерворта, Чебышева I и II рода и Золотарева—Кауэра выполняется соответственно с помощью функций:

```
[bs,as] = butter(Ra,Wn,ftype,'s')
[bs,as] = cheby1(Ra,rp,Wn,ftype,'s')
[bs,as] = cheby2(Ra,rs,Wn,ftype,'s')
[bs,as] = ellip(Ra,rp,rs,Wn,ftype,'s')
```

где R_a — порядок аналогового фильтра; W_n — вектор *частот среза* в шкале $\omega = 2\pi f$ (рад/с), содержащий один элемент — для ФНЧ и ФВЧ и два — для ПФ и РФ; частотами среза называют частоты, на которых нормированная АЧХ $\hat{A}(f)$ равна $1/\sqrt{2} \approx 0,707$, а затухание $\hat{A}(f)$ (дБ) — 3 дБ; r_p , r_s — максимально и минимально допустимые затухания $a_{\max}$ (дБ) в ПП и $a_{\min}$ (дБ) в ПЗ для характеристики затухания АЧХ (дБ) (11.6).

Для аналогового ПФ, синтезируемого с помощью данных функций, минимально допустимые отклонения в ПЗ1 и ПЗ2 *задаются одинаковыми*. Аналогично, для РФ максимально допустимые отклонения в ПП1 и ПП2 *задаются одинаковыми*.

ftype — параметр, указывающий тип избирательности и принимающий значения:

- 'high' — для ФВЧ;
- 'stop' — для РФ;
- по умолчанию (если параметр отсутствует) — для ФНЧ или ПФ.

's' — признак аналогового фильтра; при его отсутствии по умолчанию подразумевается ЦФ; *bs*, *as* — векторы коэффициентов числителя и знаменателя передаточной функции аналогового фильтра $H_a(p)$ в порядке возрастания степеней p ; $as(1) = 1$.

Порядок аналогового фильтра (R_a) и частоты среза (W_n) определяются по требованиям к АЧХ (дБ) (11.6) с помощью следующих функций, соответственно для АФП Баттерворта, Чебышева I и II рода и Золотарева—Кауэра:

```
[Ra,Wn] = buttord(Wp,Ws,rp,rs,'s')
[Ra,Wn] = cheblord(Wp,Ws,rp,rs,'s')
[Ra,Wn] = cheb2ord(Wp,Ws,rp,rs,'s')
[Ra,Wn] = ellipord(Wp,Ws,rp,rs,'s')
```

где W_p , W_s — векторы граничных частот ПП и ПЗ в порядке следования слева направо в шкале частот $\omega = 2\pi f$ (рад/с).

13.1.5. Синтез БИХ-фильтров методом билинейного Z-преобразования в MATLAB

Для синтеза БИХ-фильтра методом билинейного Z-преобразования используются те же функции, что и для синтеза аналоговых фильтров, но без параметра 's':

```
[b,a] = butter(R,WdN,ftype)
[b,a] = cheby1(R,rp,WdN,ftype)
[b,a] = cheby2(R,rs,WdN,ftype)
[b,a] = ellip(R,rp,rs,WdN,ftype)
```

где R — порядок БИХ-фильтра; WdN — вектор нормированных частот среза (см. разд. 13.1.4) в шкале нормированных частот $\hat{f} = \frac{f}{f_d/2}$:

□ для ФНЧ и ФВЧ с одним элементом $WdN(1)$, равным

$$\hat{f}_0 = \frac{f_0}{f_d/2},$$

где f_0 — абсолютная частота среза (Гц);

□ для ПФ и РФ с двумя элементами $WdN(1)$ и $WdN(2)$, соответственно равными:

$$\hat{f}_{01} = \frac{f_{01}}{f_d/2};$$

$$\hat{f}_{02} = \frac{f_{02}}{f_d/2},$$

где f_{01} , f_{02} — абсолютные частоты среза (Гц).

rp , rs — максимально и минимально допустимые затухания $a_{\max}$ (дБ) в ПП и $a_{\min}$ (дБ) в ПЗ для характеристики затухания АЧХ (дБ) (11.6).

Для ПФ, синтезируемого с помощью данных функций, минимально допустимые отклонения в ПЗ1 и ПЗ2 задаются одинаковыми. Аналогично, для РФ максимально допустимые отклонения в ПП1 и ПП2 задаются одинаковыми. Однако при синтезе БИХ-фильтра в виде объекта `dfilt` на основе объекта `fdesign` (см. разд. 12.1.3) данные ограничения снимаются.

`ftype` — параметр, указывающий тип избирательности и принимающий значения:

□ 'high' — для ФВЧ;

□ 'stop' — для РФ;

□ по умолчанию (если параметр отсутствует) — для ФНЧ и ПФ.

b , a — векторы коэффициентов числителя и знаменателя передаточной функции БИХ-фильтра $H(z)$ (13.1) в порядке возрастания отрицательных степеней z ; $a(1) = 1$.

Порядок (R) и частоты среза (WdN) БИХ-фильтров Баттерворта, Чебышева I и II рода и Золотарева—Кауэра определяются по требованиям к АЧХ (дБ) (11.6) с помощью тех же функций, что и для аналогового фильтра (см. разд. 13.1.4), но без параметра 's':

```
[R,WDn] = buttord(WDp,WDs,rp,rs)
[R,WDn] = cheb1ord(WDp,WDs,rp,rs)
[R,WDn] = cheb2ord(WDp,WDs,rp,rs)
[R,WDn] = ellipord(WDp,WDs,rp,rs)
```

где WDp, WDs — соответственно векторы граничных нормированных частот ПП и ПЗ в порядке их следования слева направо в шкале нормированных частот $\hat{f} = \frac{f}{f_d/2}$.

При синтезе БИХ-фильтров сохраняется свойство оптимальности АФП — ЦФ будет также *оптимальным*.

13.1.6. Синтез БИХ-фильтров в виде объектов *dfilt* на основе объектов *fdesign*

При задании требований к характеристике затухания АЧХ (дБ) (11.6) в виде объекта *fdesign* (см. разд. 12.1.3) для синтеза БИХ-фильтра *методом билинейного Z-преобразования* используются функции, представленные в табл. 13.2. В этом случае для ПФ и РФ соответственно минимально допустимые затухания в ПЗ1 и ПЗ2 и максимально допустимые затухания в ПП1 и ПП2 могут задаваться *неодинаковыми*.

Таблица 13.2. Функции синтеза БИХ-фильтра в виде объекта *dfilt*

Функция	Тип БИХ-фильтра
<code>butter</code>	Фильтр Баттерворта (Butterworth filter)
<code>cheby1</code>	Фильтр Чебышева I рода (Chebyshev Type I filter)
<code>cheby2</code>	Фильтр Чебышева II рода (Chebyshev Type II filter)
<code>ellip</code>	Фильтр Золотарева—Кауэра (Elliptic filter — эллиптический)

Обобщенный формат функции синтеза БИХ-фильтра в виде объекта *dfilt* на основе объекта *fdesign* представлен двумя разновидностями:

```
Hf = _function_iir(Hs, 'MatchExactly', MATCH, 'FilterStructure', 'structure')
Hf = design(Hs, 'function_iir', 'MatchExactly', MATCH, 'FilterStructure', 'structure')
```

где *function_iir* — имя конкретной функции из табл. 13.2; *Hs* — имя объекта *fdesign*; 'MatchExactly' — параметр (флаг), установка которого (присутствие в составе параметров) означает, что требования к АЧХ (дБ) должны выполняться точно; MATCH — параметр, уточняющий, в какой из полос требования должны выполняться точно, и принимающий значения:

- 'stopband' (по умолчанию) — в полосах задерживания;
- 'passband' — в полосах пропускания;
- 'both' — в полосах задерживания и пропускания (только для функции *ellip*).

'FilterStructure' — параметр (флаг), установка которого (присутствие в составе параметров) означает, что для БИХ-фильтра будет указана структура; 'structure' — функция, задающая конкретную структуру объекта `hf` (см. табл. 13.1); `hf` — имя объекта `dfilt`.

Для расчета ЧХ и ИХ БИХ-фильтра в виде объекта `dfilt` используются функции `freqz` и `impz` (см. разд. 12.1.4).

13.1.7. Расстановка звеньев и масштабирование в каскадных структурах БИХ-фильтров

Перед моделированием каскадной структуры БИХ-фильтра с *фиксированной точкой* (ФТ), о чем пойдет речь в гл. 15, необходимо предусмотреть выполнение двух операций [1]:

- формирование и расстановка звеньев.

Звенья в (13.2) формируются посредством объединения полюсов с ближайшими нулями, после чего они расставляются в порядке возрастания радиусов полюсов. Это позволяет минимизировать собственные шумы, обусловленные умножителями. При описании каскадной структуры БИХ-фильтров в виде объекта `dfilt` формирование и расстановка звеньев осуществляются автоматически;

- масштабирование.

Для минимизации переполнений на выходах сумматоров на входах звеньев добавляются масштабирующие множители, которые учитываются в числителях передаточных функций звеньев.

В MATLAB эта операция реализуется с помощью функции:

`scale(hf, norm)`

где `hf` — имя объекта `dfilt` с каскадной структурой из звеньев 2-го порядка; `norm` — вводимое в апострофах имя нормы, на основе которой рассчитываются масштабирующие множители: 'l1' — для нормы $\|x\|_1$ (2.4), 'linf' — для нормы $\|x\|_\infty$ (2.6) и 'l2' — для нормы $\|x\|_2$ (2.8), которое выбирается по умолчанию в отсутствии параметра `norm`.

13.2. Содержание лабораторной работы

Содержание работы связано с синтезом БИХ-фильтров методом билинейного Z-преобразования, описанием их структур и анализом характеристик с использованием программных средств MATLAB.

13.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файлов `lr_13_low`, `lr_13_high`, `lr_13_pass` и `lr_13_stop` и function-файла `plot_iir`, которые хранятся на прилагаемом компакт-диске в папке `LAB_DSP\LAB_13`.

Перед выполнением работы необходимо сохранить путь к папке LAB_13 по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 13.3—13.6 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$, и включают в себя требования к характеристике затухания АЧХ (дБ) (11.6) БИХ-фильтров ФНЧ, ФВЧ, ПФ и РФ. Значения допустимых затуханий рассчитаны по формулам (11.7)—(11.8).

На прилагаемом компакт-диске в папке Tables\Tables_13 хранятся табл. 13.3—13.6 исходных данных, примеры их заполнения для $N_{бр} = 1$ и табл. 13.7 для п. 4 задания.

Таблица 13.3. Требования к АЧХ (дБ) ФНЧ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_d	Частота дискретизации	$f_d = 5000 + 100N_{бр}$	Fs =
f_χ	Граничная частота ПП	$f_\chi = \frac{f_d}{10} + 20N_{бр}$	ft =
f_k	Граничная частота ПЗ	$f_k = \frac{f_d}{10} + 250 + 25N_{бр}$	fk =
$a_{\max}$	Максимально допустимое затухание в ПП (дБ)	$a_{\max} = 0,4455$	rp = 0.4455
$a_{\min}$	Минимально допустимое затухание в ПЗ (дБ)	$a_{\min} = 40$	rs = 40

Таблица 13.4. Требования к АЧХ (дБ) ФВЧ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_d	Частота дискретизации	$f_d = 5000 + 100N_{бр}$	Fs =
f_k	Граничная частота ПЗ	$f_k = \frac{f_d}{10} + 20N_{бр}$	fk =
f_χ	Граничная частота ПП	$f_\chi = \frac{f_d}{10} + 250 + 25N_{бр}$	ft =
$a_{\min}$	Минимально допустимое затухание в ПЗ (дБ)	$a_{\min} = 40$	rs = 40
$a_{\max}$	Максимально допустимое затухание в ПП (дБ)	$a_{\max} = 0,4455$	rp = 0.4455

Таблица 13.5. Требования к АЧХ (дБ) ПФ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}}$	Fs =
f_{-k}	Граничная частота ПЗ1	$f_{-k} = \frac{f_d}{20} + 20N_{\text{бр}}$	fk1 =
$f_{-\chi}$	Граничная частота ПП1	$f_{-\chi} = \frac{f_d}{20} + 250 + 25N_{\text{бр}}$	ft1 =
f_{χ}	Граничная частота ПП2	$f_{\chi} = \frac{f_d}{4} + 25N_{\text{бр}}$	ft2 =
f_k	Граничная частота ПЗ2	$f_k = \frac{f_d}{4} + 250 + 30N_{\text{бр}}$	fk2 =
a_{max}	Максимально допустимое затухание в ПП (дБ)	$a_{\text{max}} = 0,4455$	rp = 0.4455
a_{min}	Минимально допустимое затухание в ПЗ1 и ПЗ2 (дБ)	$a_{\text{min}} = 40$	rs = 40

Таблица 13.6. Требования к АЧХ (дБ) РФ

Условные обозначения	Список требований	Задаваемые значения	Идентификатор
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}}$	Fs =
$f_{-\chi}$	Граничная частота ПП1	$f_{-\chi} = \frac{f_d}{20} + 20N_{\text{бр}}$	ft1 =
f_{-k}	Граничная частота ПЗ1	$f_{-k} = \frac{f_d}{20} + 250 + 25N_{\text{бр}}$	fk1 =
f_k	Граничная частота ПЗ2	$f_k = \frac{f_d}{4} + 25N_{\text{бр}}$	fk2 =
f_{χ}	Граничная частота ПП2	$f_{\chi} = \frac{f_d}{4} + 250 + 30N_{\text{бр}}$	ft2 =
a_{max}	Максимально допустимое затухание в ПП1 и ПП2 (дБ)	$a_{\text{max}} = 0,4455$	rp = 0.4455
a_{min}	Минимально допустимое затухание в ПЗ (дБ)	$a_{\text{min}} = 40$	rs = 40

Задание на лабораторную работу заключается в синтезе БИХ-фильтров методом билинейного Z-преобразования и анализе их характеристик и для каждого типа избирательности (ФНЧ, ФВЧ, ПФ и РФ) включает в себя выполнение следующих пунктов:

1. Ввод требований к характеристике затухания БИХ-фильтра.
2. Синтез БИХ-фильтров Баттерворта, Чебышева I и II рода и Золотарева—Кауэра, выполняемый в два этапа:
 - вычисление порядков и частот среза БИХ-фильтров;
 - синтез БИХ-фильтров.

Выведенные значения порядков БИХ-фильтров записать в табл. 13.7.

Пояснить:

- какая функция используется для вычисления порядка и частот среза;
- какие функции используются для синтеза БИХ-фильтров Баттерворта, Чебышева I и II рода и Золотарева—Кауэра;
- какие из параметров данных функции соответствуют коэффициентам передаточной функции БИХ-фильтра;
- какой из БИХ-фильтров имеет минимальный порядок.

Таблица 13.7. Результаты синтеза БИХ-фильтров и АФП

Тип избирательности фильтра	Тип и порядок БИХ-фильтра			
	Баттерворта	Чебышева I рода	Чебышева II рода	Золотарева—Кауэра
	R1	R2	R3	R4
ФНЧ				
ФВЧ				
ПФ				
РФ				

3. Анализ характеристик БИХ-фильтров Баттерворта, Чебышева I и II рода и Золотарева—Кауэра.

Для вывода графиков АЧХ, ФЧХ, ИХ (50 отсчетов) и карты нулей и полюсов БИХ-фильтров использовать function-файл `plot_iir` (см. разд. 13.4.5).

Пояснить:

- вид ИХ;
- вид АЧХ в ПП и ПЗ (воспользуйтесь кнопкой **Zoom in** на панели инструментов).

4. Синтез АФП Баттерворта, Чебышева I и II рода и Золотарева—Кауэра, выполняемый в три этапа:

- формирование требований к характеристике затухания АФП с выводом граничных частот ПП и ПЗ;
- вычисление порядка и частот среза АФП;
- синтез АФП.

Пояснить:

- соответствие между граничными частотами АФП и БИХ-фильтра;
- соответствие между порядками АФП и БИХ-фильтров.

5. Вывод графиков АЧХ АФП Баттерворта, Чебышева I и II рода и Золотарева—Кауэра.

Вывести графики АЧХ АФП в основной полосе частот $[0; f_d/2]$, вычисленные с помощью функции `freqs`.

Сделать вывод по результатам сравнения АЧХ АФП и БИХ-фильтров.

6. Описание требований к АЧХ БИХ-фильтра в виде объекта `fdesign`.

Описать требования к характеристике затухания в виде объектов `fdesign` с именами:

- `MAG_lowpass` — ФНЧ;
- `MAG_highpass` — ФВЧ;
- `MAG_bandpass` — ПФ;
- `MAG_bandstop` — РФ.

Пояснить, что в себя включает список свойства объекта `fdesign` и с какой целью он создается.

7. Синтез БИХ-фильтра Золотарева—Кауэра в виде объекта `dfilt` на основе объекта `fdesign`.

Синтезировать БИХ-фильтры с прямой канонической структурой звеньев (см. табл. 13.1) в виде объектов `dfilt` со следующими именами:

- `F_lowpass` — ФНЧ;
- `F_highpass` — ФВЧ;
- `F_bandpass` — ПФ;
- `F_bandstop` — РФ.

Пояснить:

- какая функция используется для синтеза БИХ-фильтров Золотарева—Кауэра;
- что в себя включает список свойств объекта `dfilt`;
- совпадает ли порядок синтезированного БИХ-фильтра с порядком в табл. 13.7.

8. Масштабирование в каскадной структуре БИХ-фильтра Золотарева—Кауэра.

Выполнить масштабирование на основе нормы 'Linf' (по умолчанию) для объектов `F_lowpass` и `F_bandpass` и на основе нормы 'L2' для объектов `F_highpass` и `F_bandstop`.

Сохранить новые объекты `dfilt` с именами:

- `F_lowpass_scale` — ФНЧ;
- `F_highpass_scale` — ФВЧ;
- `F_bandpass_scale` — ПФ;
- `F_bandstop_scale` — РФ.

Пояснить, какие свойства объекта `dfilt` изменились после масштабирования.

9. Знакомство с GUI FVTool.

Обратиться к GUI FVTool по команде:

```
fvtool(Hd)
```

где `Hd` — имя объекта `dfilt`, и проанализировать характеристики синтезированных БИХ-фильтров.

13.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должны быть представлены табл. 13.3—13.6 исходных данных для своего номера бригады $N_{бр}$.

Для четырех типов избирательности БИХ-фильтра — ФНЧ, ФВЧ, ПФ и РФ — предусмотрены четыре script-файла. Для запуска script-файла к нему необходимо обратиться по имени:

```
>> lr_13_low — ФНЧ  
>> lr_13_high — ФВЧ  
>> lr_13_pass — ПФ  
>> lr_13_stop — РФ
```

Листинги данных script-файлов представлены в разд. 13.4.1—13.4.4.

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш <Ctrl>+<Break>.

При выполнении script-файла текущие окна с графиками *не закрывать*.

13.4.1. Синтез и анализ БИХ-фильтра ФНЧ

Листинг script-файла `lr_13_low` имеет вид:

```
>> type lr_13_low  
script
```

```

clc
clear
disp('% ЛР №13. СИНТЕЗ БИХ-ФИЛЬТРА ФНЧ МЕТОДОМ БИЛИНЕЙНОГО Z-ПРЕОБРАЗОВАНИЯ')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ (дБ) ФНЧ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ (дБ) ')
DATA=0;
while DATA==0;
Nb = input('Nb = '); % НОМЕР БРИГАДЫ
Fs = input('Fs = '); % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
ft = input('ft = '); % ГРАНИЧНАЯ ЧАСТОТА ПП (Гц)
fk = input('fk = '); % ГРАНИЧНАЯ ЧАСТОТА ПЗ (Гц)
rp = input('rp = '); % МАКСИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПП
rs = input('rs = '); % МИНИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПЗ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. СИНТЕЗ БИХ-ФИЛЬТРОВ БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА  
И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для синтеза БИХ-фильтров ФНЧ нажмите <ENTER>')
pause
WDp = ft/(Fs/2); WDs = fk/(Fs/2); % ГРАНИЧНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ПП и ПЗ
[R1,WDn1] = buttord(WDp,WDs,rp,rs); % ПОРЯДОК И ЧАСТОТА СРЕЗА БИХ-ФИЛЬТРА ФНЧ  
БАТТЕРВОРТА
[R2,WDn2] = cheblord(WDp,WDs,rp,rs); % ПОРЯДОК И ЧАСТОТА СРЕЗА БИХ-ФИЛЬТРА  
ЧЕБЫШЕВА I РОДА
[R3,WDn3] = cheb2ord(WDp,WDs,rp,rs); % ПОРЯДОК И ЧАСТОТА СРЕЗА БИХ-ФИЛЬТРА ФНЧ  
ЧЕБЫШЕВА II РОДА
[R4,WDn4] = ellipord(WDp,WDs,rp,rs); % ПОРЯДОК И ЧАСТОТА СРЕЗА БИХ-ФИЛЬТРА ФНЧ  
ЗОЛОТАРЕВА-КАУЭРА
[b1,a1] = butter(R1,WDn1); % КОЭФИЦИЕНТЫ БИХ-ФИЛЬТРА ФНЧ БАТТЕРВОРТА
[b2,a2] = cheby1(R2,rp,WDn2); % КОЭФИЦИЕНТЫ БИХ-ФИЛЬТРА ФНЧ ЧЕБЫШЕВА  
I РОДА
[b3,a3] = cheby2(R3,rs,WDn3); % КОЭФИЦИЕНТЫ БИХ-ФИЛЬТРА ФНЧ ЧЕБЫШЕВА  
II РОДА

```

```

[b4,a4] = ellip(R4,rp,rs,WDn4);           % КОЭФИЦИЕНТЫ БИХ-ФИЛЬТРА ФНЧ
ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для вывода порядков БИХ-фильтров ФНЧ нажмите <ENTER>')
pause
disp('%')
disp('%')
disp([' R1 = ' num2str(R1),' R2 = ' num2str(R2),' R3 = ' num2str(R3),'
R4 = ' num2str(R4) ])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. АНАЛИЗ ХАРАКТЕРИСТИК БИХ-ФИЛЬТРОВ ФНЧ')
disp('%')
disp('%')
disp('% Для вывода ХАРАКТЕРИСТИК БИХ-ФИЛЬТРОВ ФНЧ (ЧЕТЫРЕ ГРАФИЧЕСКИХ ОКНА)
нажмите <ENTER>')
pause
figure('Name','Lowpass IIR Filter Butterworth','NumberTitle','off')
plot_iir(b1,a1,Fs)           % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ФНЧ БАТТЕРВОРТА
figure('Name','Lowpass IIR Filter Chebyshov I','NumberTitle','off')
plot_iir(b2,a2,Fs)           % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ФНЧ ЧЕБЫШЕВА II РОДА
figure('Name','Lowpass IIR Filter Chebyshov II','NumberTitle','off')
plot_iir(b3,a3,Fs)           % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ФНЧ ЧЕБЫШЕВА II РОДА
figure('Name','Lowpass IIR Filter Elliptic','NumberTitle','off')
plot_iir(b4,a4,Fs)           % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ФНЧ ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. СИНТЕЗ АПФ БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода граничных частот АПФ ФНЧ ПП (Ft) и ПЗ (Fk) нажмите <ENTER>')
pause
disp('%')
disp('%')
Ft = (Fs/pi)*tan(pi*ft/Fs); Fk = (Fs/pi)*tan(pi*fk/Fs); % ГРАНИЧНЫЕ ЧАСТОТЫ ПП
и ПЗ АПФ
disp([' Ft = ' num2str(Ft),' Fk = ' num2str(Fk) ])
disp('%')

```

```

disp('%')
disp('% Для синтеза АФП ФНЧ нажмите <ENTER>')
pause
Wp = 2.*pi.*Ft; Ws = 2.*pi.*Fk; % ГРАНИЧНЫЕ КРУГОВЫЕ ЧАСТОТЫ ПП и ПЗ АФП
[Ra1,Wn1] = buttord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТА СРЕЗА АФП ФНЧ
БАТТЕРВОРТА
[Ra2,Wn2] = cheblord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТА СРЕЗА АФП ФНЧ
ЧЕБЫШЕВА I РОДА
[Ra3,Wn3] = cheb2ord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТА СРЕЗА АФП ФНЧ
ЧЕБЫШЕВА II РОДА
[Ra4,Wn4] = ellipord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТА СРЕЗА АФП ФНЧ
ЗОЛОТАРЕВА-КАУЭРА
[bs1,as1] = butter(Ra1,Wn1,'s'); % КОЭФФИЦИЕНТЫ АФП ФНЧ БАТТЕРВОРТА
[bs2,as2] = cheby1(Ra2,rp,Wn2,'s'); % КОЭФФИЦИЕНТЫ АФП ФНЧ ЧЕБЫШЕВА I РОДА
[bs3,as3] = cheby2(Ra3,rs,Wn3,'s'); % КОЭФФИЦИЕНТЫ АФП ФНЧ ЧЕБЫШЕВА II РОДА
[bs4,as4] = ellip(Ra4,rp,rs,Wn4,'s'); % КОЭФФИЦИЕНТЫ АФП ФНЧ ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для вывода порядков АФП ФНЧ нажмите <ENTER>')
pause
disp('%')
disp('%')
disp([' Ra1 = ' num2str(Ra1), ' Ra2 = ' num2str(Ra2), ' Ra3 = '
num2str(Ra3), ' Ra4 = ' num2str(Ra4)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ВЫВОД ГРАФИКОВ АЧХ АФП БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА
И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АЧХ АФП нажмите <ENTER>')
pause
f = 0:(Fs/2)/1000:Ffs/2; % СЕТКА ЧАСТОТ ДЛЯ ГРАФИКА АЧХ
W = 2.*pi.*f;
Ha1 = freqs(bs1,as1,W); MAG1 = abs(Ha1); % ЧХ и АЧХ АФП ФНЧ БАТТЕРВОРТА
Ha2 = freqs(bs2,as2,W); MAG2 = abs(Ha2); % ЧХ и АЧХ АФП ФНЧ ЧЕБЫШЕВА I РОДА
Ha3 = freqs(bs3,as3,W); MAG3 = abs(Ha3); % ЧХ и АЧХ АФП ФНЧ ЧЕБЫШЕВА II РОДА
Ha4 = freqs(bs4,as4,W); MAG4 = abs(Ha4); % ЧХ и АЧХ АФП ФНЧ ЗОЛОТАРЕВА-КАУЭРА
figure('Name','Lowpass Analog Filter – Magnitude','NumberTitle','off')
subplot(2,2,1),plot(f,abs(Ha1)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Butterworth'),ylim([0 1.2])
subplot(2,2,2),plot(f,abs(Ha2)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Chebyshev I'),ylim([0 1.2])
subplot(2,2,3),plot(f,abs(Ha3)),xlabel('f(Hz)'),grid,...

```

```

ylabel('MAGNITUDE'),title('Analog Filter Chebyshov II'),ylim([0 1.2])
subplot(2,2,4),plot(f, abs(Ha4)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Elliptic'),ylim([0 1.2])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ОПИСАНИЕ ТРЕБОВАНИЙ К АЧХ БИХ-фильтра В ВИДЕ ОБЪЕКТА fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА fdesign нажмите <ENTER>')
pause
MAG_lowpass = fdesign.lowpass('Fp,Fst,Ap,Ast',ft,fk,rp,rs,[Fs]) % ОБЪЕКТ
fdesign ДЛЯ ФНЧ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. СИНТЕЗ БИХ-ФИЛЬТРА ФНЧ ЗОЛОТАРЕВА-КАУЭРА В ВИДЕ ОБЪЕКТА dfilt')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_lowpass = design(MAG_lowpass,'ellip','MatchExactly',
'both','FilterStructure','df2sos') % ФНЧ В ВИДЕ ОБЪЕКТА dfilt
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.8. МАСШТАБИРОВАНИЕ В КАСКАДНОЙ СТРУКТУРЕ БИХ-ФИЛЬТРА ФНЧ
ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt после масштабирования нажмите
<ENTER>')
pause
F_lowpass_scale = scale(F_lowpass) % РЕЗУЛЬТАТ МАСШТАБИРОВАНИЯ
disp('%')
disp('%')
disp('% СИНТЕЗ БИХ-ФИЛЬТРА ФНЧ ЗАВЕРШЕН')
```

13.4.2. Синтез и анализ БИХ-фильтра ФВЧ

Листинг script-файла `lr_13_high` имеет вид:

```
>> type lr_13_high
script
clc
clear
disp('% ЛР №13. СИНТЕЗ БИХ-ФИЛЬТРА ФВЧ МЕТОДОМ БИЛИНЕЙНОГО Z-ПРЕОБРАЗОВАНИЯ')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ (дБ) ФВЧ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ (дБ)')
DATA=0;
while DATA==0;
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
fk = input('fk = ');           % ГРАНИЧНАЯ ЧАСТОТА ПЗ (Гц)
ft = input('ft = ');           % ГРАНИЧНАЯ ЧАСТОТА ПП (Гц)
rs = input('rs = ');           % МИНИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПЗ
rp = input('rp = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПП
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. СИНТЕЗ БИХ-ФИЛЬТРОВ БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА И ЗОЛОТАРЕВА-КАУЗРА')
disp('%')
disp('%')
disp('% Для синтеза БИХ-фильтров ФВЧ нажмите <ENTER>')
pause
WDp = ft/(Fs/2); WDs = fk/(Fs/2); % ГРАНИЧНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ПП и ПЗ
[R1,Wdn1] = buttord(WDp,WDs,rp,rs); % ПОРЯДОК И ЧАСТОТА СРЕЗА БИХ-ФИЛЬТРА ФВЧ
БАТТЕРВОРТА
[R2,Wdn2] = cheblord(WDp,WDs,rp,rs); % ПОРЯДОК И ЧАСТОТА СРЕЗА БИХ-ФИЛЬТРА ФВЧ
ЧЕБЫШЕВА I РОДА
[R3,Wdn3] = cheb2ord(WDp,WDs,rp,rs); % ПОРЯДОК И ЧАСТОТА СРЕЗА БИХ-ФИЛЬТРА ФВЧ
ЧЕБЫШЕВА II РОДА
```

```

[R4,WDn4] = ellipord(WDp,WDs,rp,rs); % ПОРЯДОК И ЧАСТОТА СРЕЗА БИХ-ФИЛЬТРА ФВЧ
ЗОЛОТАРЕВА-КАУЭРА
[b1,a1] = butter(R1,WDn1,'high'); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА ФВЧ БАТТЕРВОРТА
[b2,a2] = cheby1(R2,rp,WDn2,'high'); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА ФВЧ ЧЕБЫШЕВА
I РОДА
[b3,a3] = cheby2(R3,rs,WDn3,'high'); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА ФВЧ ЧЕБЫШЕВА
II РОДА
[b4,a4] = cheby2(R4,rs,WDn4,'high'); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА ФВЧ ЧЕБЫШЕВА
И ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для вывода порядков БИХ-фильтров ФВЧ нажмите <ENTER>')
pause
disp('%')
disp('%')
disp([' R1 = ' num2str(R1),' R2 = ' num2str(R2),' R3 = ' num2str(R3),'
R4 = ' num2str(R4)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. АНАЛИЗ ХАРАКТЕРИСТИК БИХ-ФИЛЬТРОВ ФВЧ')
disp('%')
disp('%')
disp('% Для вывода ХАРАКТЕРИСТИК БИХ-ФИЛЬТРОВ ФВЧ (ЧЕТЫРЕ ГРАФИЧЕСКИХ ОКНА)
нажмите <ENTER>')
pause
figure('Name','Highpass IIR Filter Butterworth','NumberTitle','off')
plot_iir(b1,a1,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ФВЧ БАТТЕРВОРТА
figure('Name','Highpass IIR Filter Chebyshov I','NumberTitle','off')
plot_iir(b2,a2,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ФВЧ ЧЕБЫШЕВА II РОДА
figure('Name','HighpassIIR Filter Chebyshov II','NumberTitle','off')
plot_iir(b3,a3,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ФВЧ ЧЕБЫШЕВА II РОДА
figure('Name','Highpass IIR Filter Elliptic','NumberTitle','off')
plot_iir(b4,a4,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ФВЧ ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. СИНТЕЗ АФП БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА
И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода граничных частот АФП ФВЧ ПЗ (Fk) и ПП (Ft) нажмите <ENTER>')

```



```

pause
disp('%')
disp('%')
Ft = (Fs/pi)*tan(pi*ft/Fs); Fk = (Fs/pi)*tan(pi*fk/Fs); % ГРАНИЧНЫЕ ЧАСТОТЫ ПП
и ПЗ АФП
disp(['   Fk = ' num2str(Fk), '   Ft = ' num2str(Ft)])
disp('%')
disp('%')
disp('% Для синтеза АФП ФВЧ нажмите <ENTER>')
pause
Wp = 2.*pi.*Ft; Ws = 2.*pi.*Fk; % ГРАНИЧНЫЕ КРУГОВЫЕ ЧАСТОТЫ ПП и ПЗ АФП
[Ra1,Wn1] = buttord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТА СРЕЗА АФП ФВЧ
БАТТЕРВОРТА
[Ra2,Wn2] = cheblord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТА СРЕЗА АФП ФВЧ
ЧЕБЫШЕВА I РОДА
[Ra3,Wn3] = cheb2ord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТА СРЕЗА АФП ФВЧ
ЧЕБЫШЕВА II РОДА
[Ra4,Wn4] = ellipord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТА СРЕЗА АФП ФВЧ
ЗОЛОТАРЕВА-КАУЭРА
[bs1,as1] = butter(Ra1,Wn1,'high','s'); % КОЭФФИЦИЕНТЫ АФП ФВЧ БАТТЕРВОРТА
[bs2,as2] = cheby1(Ra2,rp,Wn2,'high','s'); % КОЭФФИЦИЕНТЫ АФП ФВЧ ЧЕБЫШЕВА
I РОДА
[bs3,as3] = cheby2(Ra3,rs,Wn3,'high','s'); % КОЭФФИЦИЕНТЫ АФП ФВЧ ЧЕБЫШЕВА
II РОДА
[bs4,as4] = ellip(Ra4,rp,rs,Wn4,'high','s'); % КОЭФФИЦИЕНТЫ АФП ФВЧ ЧЕБЫШЕВА
И ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для вывода порядков АФП ФВЧ нажмите <ENTER>')
pause
disp('%')
disp('%')
disp(['   Ra1 = ' num2str(Ra1), '   Ra2 = ' num2str(Ra2), '   Ra3 = '
num2str(Ra3), '   Ra4 = ' num2str(Ra4)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ВЫВОД ГРАФИКОВ АЧХ АФП БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА
И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АЧХ АФП нажмите <ENTER>')
pause
f = 0:(Fs/2)/1000:Ff/2; % СЕТКА ЧАСТОТ ДЛЯ ГРАФИКА АЧХ
W = 2.*pi.*f;

```

```

Ha1 = freqs(bs1,as1,W);           % ЧХ АФП БАТТЕРВОРТА
Ha2 = freqs(bs2,as2,W);           % ЧХ АФП ЧЕБЫШЕВА I РОДА
Ha3 = freqs(bs3,as3,W);           % ЧХ АФП ЧЕБЫШЕВА II РОДА
Ha4 = freqs(bs4,as4,W);           % ЧХ АФП ЗОЛОТАРЕВА-КАУЭРА
figure('Name','Highpass Analog Filter – Magnitude','NumberTitle','off')
subplot(2,2,1),plot(f,abs(Ha1)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Butterworth'),ylim([0 1.2])
subplot(2,2,2),plot(f,abs(Ha2)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Chebyshov I'),ylim([0 1.2])
subplot(2,2,3),plot(f,abs(Ha3)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Chebyshov II'),ylim([0 1.2])
subplot(2,2,4),plot(f,abs(Ha4)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Elliptic'),ylim([0 1.2])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ОПИСАНИЕ ТРЕБОВАНИЙ К АЧХ БИХ-фильтра В ВИДЕ ОБЪЕКТА fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА fdesign нажмите <ENTER>')
pause
MAG_highpass = fdesign.highpass('Fst,Fp,Ast,Ap',fk,ft,rs,rp,[Fs]) % ОБЪЕКТ
fdesign ДЛЯ ФВЧ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. СИНТЕЗ БИХ-ФИЛЬТРА ФВЧ ЗОЛОТАРЕВА-КАУЭРА В ВИДЕ ОБЪЕКТА dfilt')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_highpass = design(MAG_highpass,'ellip','MatchExactly',
'both','FilterStructure','df2sos') % ФВЧ В ВИДЕ ОБЪЕКТА dfilt
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.8. МАСШТАБИРОВАНИЕ В КАСКАДНОЙ СТРУКТУРЕ БИХ-ФИЛЬТРА ФВЧ
ЗОЛОТАРЕВА-КАУЭРА')
disp('%')

```

```

disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt после масштабирования нажмите
<ENTER>')
pause
F_highpass_scale = scale(F_highpass,'L2')    % РЕЗУЛЬТАТ МАСШТАБИРОВАНИЯ
disp('%')
disp('%')
disp('% СИНТЕЗ БИХ-ФИЛЬТРА ФВЧ ЗАВЕРШЕН')

```

13.4.3. Синтез и анализ БИХ-фильтра ПФ

Листинг script-файла lr_13_pass имеет вид:

```

>> type lr_13_pass
script
clc
clear
disp('% ЛР №13. СИНТЕЗ БИХ-ФИЛЬТРА ПФ МЕТОДОМ БИЛИНЕЙНОГО Z-ПРЕОБРАЗОВАНИЯ')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ (дБ) ПФ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ (дБ)')
DATA=0;
while DATA==0;
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
fk1 = input('fk1 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПЗ1 (Гц)
ft1 = input('ft1 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПП1 (Гц)
ft2 = input('ft2 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПЗ2 (Гц)
fk2 = input('fk2 = ');         % ГРАНИЧНАЯ ЧАСТОТА ПЗ2 (Гц)
rp = input('rp = ');           % МАКСИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПП
rs = input('rs = ');           % МИНИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПЗ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. СИНТЕЗ БИХ-ФИЛЬТРОВ БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА
И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')

```

```

disp('%')
disp('% Для синтеза БИХ-фильтров ПФ нажмите <ENTER>')
pause
ft = [ft1 ft2]; fk = [fk1 fk2]; % ВЕКТОРЫ ГРАНИЧНЫХ ЧАСТОТ ПП и ПЗ
Wdp = ft/(Fs/2); Wds = fk/(Fs/2); % ВЕКТОРЫ ГРАНИЧНЫХ НОРМИРОВАННЫХ ЧАСТОТ
ПП и ПЗ
[R1,Wdn1] = buttord(Wdp,Wds,rp,rs); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА БИХ-ФИЛЬТРА ПФ
БАТТЕРВОРТА
[R2,Wdn2] = cheblord(Wdp,Wds,rp,rs); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА БИХ-ФИЛЬТРА ПФ
ЧЕБЫШЕВА I РОДА
[R3,Wdn3] = cheb2ord(Wdp,Wds,rp,rs); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА БИХ-ФИЛЬТРА ПФ
ЧЕБЫШЕВА II РОДА
[R4,Wdn4] = ellipord(Wdp,Wds,rp,rs); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА БИХ-ФИЛЬТРА ПФ
ЗОЛОТАРЕВА-КАУЭРА
[b1,a1] = butter(R1,Wdn1); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА ПФ БАТТЕРВОРТА
[b2,a2] = cheby1(R2,rp,Wdn2); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА ПФ ЧЕБЫШЕВА
I РОДА
[b3,a3] = cheby2(R3,rs,Wdn3); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА ПФ ЧЕБЫШЕВА
II РОДА
[b4,a4] = ellip(R4,rp,rs,Wdn4); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА ПФ
ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для вывода порядков БИХ-фильтров ПФ нажмите <ENTER>')
pause
disp('%')
disp('%')
disp([' R1 = ' num2str(R1), ' R2 = ' num2str(R2), ' R3 = ' num2str(R3), '
R4 = ' num2str(R4)'])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. АНАЛИЗ ХАРАКТЕРИСТИК БИХ-ФИЛЬТРОВ ПФ')
disp('%')
disp('%')
disp('% Для вывода ХАРАКТЕРИСТИК БИХ-ФИЛЬТРОВ ПФ (ЧЕТЫРЕ ГРАФИЧЕСКИХ ОКНА)
нажмите <ENTER>')
pause
figure('Name','Bandpass IIR Filter Butterworth','NumberTitle', 'off')
plot_iir(b1,a1,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ПФ БАТТЕРВОРТА
figure('Name','Bandpass IIR Filter Chebyshov I','NumberTitle', 'off')
plot_iir(b2,a2,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ПФ ЧЕБЫШЕВА II РОДА
figure('Name','Bandpass IIR Filter Chebyshov II','NumberTitle', 'off')
plot_iir(b3,a3,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ПФ ЧЕБЫШЕВА II РОДА
figure('Name','Bandpass IIR Filter Elliptic','NumberTitle', 'off')

```

```

plot_iir(b4,a4,Fs)           % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА ПФ ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. СИНТЕЗ АФП БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода граничных частот АФП ПФ ПЗ1 (Fk1), ПП1 (Ft1), ПП2 (Ft2)
и ПЗ2 (Fk2) нажмите <ENTER>')
pause
disp('%')
disp('%')
ft = [ft1 ft2]; fk = [fk1 fk2];           % ВЕКТОРЫ ГРАНИЧНЫХ ЧАСТОТ ПП и ПЗ
БИХ-ФИЛЬТРА
Ft = (Fs/pi)*tan(pi*ft/Fs); Fk = (Fs/pi)*tan(pi*fk/Fs); % ВЕКТОРЫ ГРАНИЧНЫХ
ЧАСТОТ ПП и ПЗ АФП
disp(['   Fk1 = ' num2str(Fk(1)), '   Ft1 = ' num2str(Ft(1)), '   Ft2 = '
num2str(Ft(2)), '   Fk2 = ' num2str(Fk(2))])
disp('%')
disp('%')
disp('% Для синтеза АФП ПФ нажмите <ENTER>')
pause
Wp = 2.*pi.*Ft; Ws = 2.*pi.*Fk;           % ВЕКТОРЫ ГРАНИЧНЫХ КРУГОВЫХ ЧАСТОТ ПП
и ПЗ АФП
[Ra1,Wn1] = buttord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА АФП ПФ
БАТТЕРВОРТА
[Ra2,Wn2] = cheblord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА АФП ПФ ЧЕБЫШЕВА
I РОДА
[Ra3,Wn3] = cheb2ord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА АФП ПФ ЧЕБЫШЕВА
II РОДА
[Ra4,Wn4] = ellipord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА АФП ПФ
ЗОЛОТАРЕВА-КАУЭРА
[bs1,as1] = butter(Ra1,Wn1,'s');           % КОЭФФИЦИЕНТЫ АФП ПФ БАТТЕРВОРТА
[bs2,as2] = cheby1(Ra2,rp,Wn2,'s');         % КОЭФФИЦИЕНТЫ АФП ПФ ЧЕБЫШЕВА I РОДА
[bs3,as3] = cheby2(Ra3,rs,Wn3,'s');         % КОЭФФИЦИЕНТЫ АФП ПФ ЧЕБЫШЕВА II РОДА
[bs4,as4] = ellip(Ra4,rp,rs,Wn4,'s');      % КОЭФФИЦИЕНТЫ АФП ПФ ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для вывода порядков АФП ПФ нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('%')
disp(['   Ra1 = ' num2str(Ra1), '   Ra2 = ' num2str(Ra2), '   Ra3 = '
num2str(Ra3), '   Ra4 = ' num2str(Ra4)])
disp('%')
disp('%')

```

```

disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ВЫВОД ГРАФИКОВ АЧХ АФП БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА  
И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АЧХ АФП нажмите <ENTER>')
pause
f = 0:((Fs/2)/1000):Fs/2;           % СЕТКА ЧАСТОТ ДЛЯ ГРАФИКА АЧХ
W = 2.*pi.*f;
Ha1 = freqs(bs1,as1,W);           % ЧХ АФП БАТТЕРВОРТА
Ha2 = freqs(bs2,as2,W);           % ЧХ АФП ЧЕБЫШЕВА I РОДА
Ha3 = freqs(bs3,as3,W);           % ЧХ АФП ЧЕБЫШЕВА II РОДА
Ha4 = freqs(bs4,as4,W);           % ЧХ АФП ЗОЛОТАРЕВА-КАУЭРА
figure('Name','Bandpass Analog Filter – Magnitude','NumberTitle','off')
subplot(2,2,1),plot(f,abs(Ha1)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Butterworth'),ylim([0 1.2])
subplot(2,2,2),plot(f,abs(Ha2)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Chebyshev I'),ylim([0 1.2])
subplot(2,2,3),plot(f,abs(Ha3)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Chebyshev II'),ylim([0 1.2])
subplot(2,2,4),plot(f,abs(Ha4)),xlabel('f(Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Elliptic'),ylim([0 1.2])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ОПИСАНИЕ ТРЕБОВАНИЙ К АЧХ БИХ-фильтра В ВИДЕ ОБЪЕКТА fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА fdesign нажмите <ENTER>')
pause
MAG_bandpass =
fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2',fk1,ft1,ft2,fk2,rs,rp,rs,[Fs]
) % ОБЪЕКТ fdesign ДЛЯ ПФ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. СИНТЕЗ БИХ-ФИЛЬТРА ПФ ЗОЛОТАРЕВА-КАУЭРА В ВИДЕ ОБЪЕКТА dfilt')
disp('%')

```

```

disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_bandpass = design(MAG_bandpass,'ellip','MatchExactly',
'both','FilterStructure','df2sos')      % ПФ В ВИДЕ ОБЪЕКТА dfilt
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.8. МАСШТАБИРОВАНИЕ В КАСКАДНОЙ СТРУКТУРЕ БИХ-ФИЛЬТРА ПФ ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt после масштабирования нажмите <ENTER>')
pause
F_bandpass_scale = scale(F_bandpass)    % РЕЗУЛЬТАТ МАСШТАБИРОВАНИЯ
disp('%')
disp('%')
disp('% СИНТЕЗ БИХ-ФИЛЬТРА ПФ ЗАВЕРШЕН')

```

13.4.4. Синтез и анализ БИХ-фильтра РФ

Листинг script-файла lr_13_stop имеет вид:

```

>> type lr_13_stop
script
clc
clear
disp('% ЛР №13. СИНТЕЗ БИХ-ФИЛЬТРА РФ МЕТОДОМ БИЛИНЕЙНОГО Z-ПРЕОБРАЗОВАНИЯ')
disp('%')
disp('%')
disp('% п.1. ВВОД ТРЕБОВАНИЙ К АЧХ (дБ) РФ')
disp('%')
disp('%')
disp('% Введите НОМЕР БРИГАДЫ и ТРЕБОВАНИЯ к АЧХ (дБ)')
DATA=0;
while DATA==0;
Nb = input('Nb = ');          % НОМЕР БРИГАДЫ
Fs = input('Fs = ');          % ЧАСТОТА ДИСКРЕТИЗАЦИИ в Гц
ft1 = input('ft1 = ');        % ГРАНИЧНАЯ ЧАСТОТА ПП1 в Гц
fk1 = input('fk1 = ');        % ГРАНИЧНАЯ ЧАСТОТА ПЗ1 в Гц
fk2 = input('fk2 = ');        % ГРАНИЧНАЯ ЧАСТОТА ПЗ2 в Гц
ft2 = input('ft2 = ');        % ГРАНИЧНАЯ ЧАСТОТА ПЗ2 в Гц
rp = input('rp = ');          % МАКСИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПП
rs = input('rs = ');          % МИНИМАЛЬНО ДОПУСТИМОЕ ЗАТУХАНИЕ В ПЗ

```

```

disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. СИНТЕЗ БИХ-ФИЛЬТРОВ БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА
И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для синтеза БИХ-фильтров РФ нажмите <ENTER>')
pause
ft = [ft1 ft2]; fk = [fk1 fk2];           % ВЕКТОРЫ ГРАНИЧНЫХ ЧАСТОТ ПП и ПЗ
WDp = ft/(Fs/2); WDs = fk/(Fs/2);       % ВЕКТОРЫ ГРАНИЧНЫХ НОРМИРОВАННЫХ ЧАСТОТ
ПП и ПЗ
[R1,WDn1] = buttord(WDp,WDs,rp,rs);       % ПОРЯДОК И ЧАСТОТЫ СРЕЗА БИХ-ФИЛЬТРА РФ
БАТТЕРВОРТА
[R2,WDn2] = cheblord(WDp,WDs,rp,rs);     % ПОРЯДОК И ЧАСТОТЫ СРЕЗА БИХ-ФИЛЬТРА РФ
ЧЕБЫШЕВА I РОДА
[R3,WDn3] = cheb2ord(WDp,WDs,rp,rs);     % ПОРЯДОК И ЧАСТОТЫ СРЕЗА БИХ-ФИЛЬТРА РФ
ЧЕБЫШЕВА II РОДА
[R4,WDn4] = ellipord(WDp,WDs,rp,rs);     % ПОРЯДОК И ЧАСТОТЫ СРЕЗА БИХ-ФИЛЬТРА РФ
ЗОЛОТАРЕВА-КАУЭРА
[b1,a1] = butter(R1,WDn1,'stop');       % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА РФ
БАТТЕРВОРТА
[b2,a2] = cheby1(R2,rp,WDn2,'stop');    % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА РФ ЧЕБЫШЕВА
I РОДА
[b3,a3] = cheby2(R3,rs,WDn3,'stop');    % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА РФ ЧЕБЫШЕВА
II РОДА
[b4,a4] = ellip(R4,rp,rs,WDn4,'stop'); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА РФ
ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для вывода порядков БИХ-фильтров РФ нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('%')
disp([' R1 = ' num2str(R1), ' R2 = ' num2str(R2), ' R3 = ' num2str(R3), '
R4 = ' num2str(R4)])
disp('%')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause

```



```

disp('%')
disp('%')
disp('% п.3. АНАЛИЗ ХАРАКТЕРИСТИК БИХ-ФИЛЬТРОВ РФ')
disp('%')
disp('%')
disp('% Для вывода ХАРАКТЕРИСТИК БИХ-ФИЛЬТРОВ РФ (ЧЕТЫРЕ ГРАФИЧЕСКИХ ОКНА)
нажмите <ENTER>')
pause
figure('Name','Bandstop IIR Filter Butterworth','NumberTitle','off')
plot_iir(b1,a1,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА РФ БАТТЕРВОРТА
figure('Name','Bandstop IIR Filter Chebyshov I','NumberTitle','off')
plot_iir(b2,a2,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА РФ ЧЕБЫШЕВА I РОДА
figure('Name','Bandstop IIR Filter Chebyshov II','NumberTitle','off')
plot_iir(b3,a3,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА РФ ЧЕБЫШЕВА II РОДА
figure('Name','Bandstop IIR Filter Elliptic','NumberTitle','off')
plot_iir(b4,a4,Fs) % ХАРАКТЕРИСТИКИ БИХ-ФИЛЬТРА РФ ЗОЛОТАРЕВА-КАУЭРА
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. СИНТЕЗ АФП БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода граничных частот АФП РФ ПП1 (Ft1), ПЗ1 (Fk1), ПЗ2 (Fk2)
и ПП2 (Ft2) нажмите <ENTER>')
pause
disp('%')
disp('%')
ft = [ft1 ft2]; fk = [fk1 fk2]; % ВЕКТОРЫ ГРАНИЧНЫХ ЧАСТОТ ПП и
ПЗ БИХ-ФИЛЬТРА
Ft = (Fs/pi)*tan(pi*ft/Fs); Fk = (Fs/pi)*tan(pi*fk/Fs); % ВЕКТОРЫ ГРАНИЧНЫХ
ЧАСТОТ ПП и ПЗ АФП
disp([' Ft1 = ' num2str(Ft(1)), ' Fk1 = ' num2str(Fk(1)), ' Fk2 = '
num2str(Fk(2)), ' Ft2 = ' num2str(Ft(2))])
disp('%')
disp('%')
disp('% Для синтеза АФП РФ нажмите <ENTER>')
pause
Wp = 2.*pi.*Ft; Ws = 2.*pi.*Fk; % ВЕКТОРЫ ГРАНИЧНЫХ КРУГОВЫХ ЧАСТОТ
ПП и ПЗ АФП
[Ra1,Wn1] = buttord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА АФП РФ
БАТТЕРВОРТА
[Ra2,Wn2] = cheblord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА АФП РФ
ЧЕБЫШЕВА I РОДА
[Ra3,Wn3] = cheb2ord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА АФП РФ
ЧЕБЫШЕВА II РОДА
[Ra4,Wn4] = ellipord(Wp,Ws,rp,rs,'s'); % ПОРЯДОК И ЧАСТОТЫ СРЕЗА АФП РФ
ЗОЛОТАРЕВА-КАУЭРА

```

```

[bs1,as1] = butter(Ra1,Wn1,'stop','s');           % КОЭФФИЦИЕНТЫ АФП РФ БАТТЕРВОРТА
[bs2,as2] = cheby1(Ra2,rp,Wn2,'stop','s');       % КОЭФФИЦИЕНТЫ АФП РФ ЧЕБЫШЕВА
I РОДА
[bs3,as3] = cheby2(Ra3,rs,Wn3,'stop','s');       % КОЭФФИЦИЕНТЫ АФП РФ ЧЕБЫШЕВА
II РОДА
[bs4,as4] = ellip(Ra4,rp,rs,Wn4,'stop','s');     % КОЭФФИЦИЕНТЫ АФП РФ ЗОЛОТАРЕВА-
КАУЭРА
disp('%')
disp('%')
disp('% Для вывода порядков АФП РФ нажмите <ENTER>')
pause
disp('%')
disp('%')
disp(['   Ra1 = ' num2str(Ra1),'   Ra2 = ' num2str(Ra2),'   Ra3 = '
num2str(Ra3),'   Ra4 = ' num2str(Ra4)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ВЫВОД ГРАФИКОВ АЧХ АФП БАТТЕРВОРТА, ЧЕБЫШЕВА I И II РОДА
И ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АЧХ АФП нажмите <ENTER>')
pause
f = 0 : ((Fs/2)/1000):Fs/2;                       % СЕТКА ЧАСТОТ ДЛЯ ГРАФИКА АЧХ
W = 2.*pi.*f;
Ha1 = freqs(bs1,as1,W);                           % ЧХ АФП БАТТЕРВОРТА
Ha2 = freqs(bs2,as2,W);                           % ЧХ АФП ЧЕБЫШЕВА I РОДА
Ha3 = freqs(bs3,as3,W);                           % ЧХ АФП ЧЕБЫШЕВА II РОДА
Ha4 = freqs(bs4,as4,W);                           % ЧХ АФП ЗОЛОТАРЕВА-КАУЭРА
figure('Name','Bandstop Analog Filter — Magnitude','NumberTitle','off')
subplot(2,2,1),plot(f,abs(Ha1)),xlabel('f (Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Butterworth'),ylim([0 1.2])
subplot(2,2,2),plot(f,abs(Ha2)),xlabel('f (Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Chebyshov I'),ylim([0 1.2])
subplot(2,2,3),plot(f,abs(Ha3)),xlabel('f (Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Chebyshov II'),ylim([0 1.2])
subplot(2,2,4),plot(f,abs(Ha4)),xlabel('f (Hz)'),grid,...
ylabel('MAGNITUDE'),title('Analog Filter Elliptic'),ylim([0 1.2])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')

```

```

disp('% п.6. ОПИСАНИЕ ТРЕБОВАНИЙ К АЧХ БИХ-фильтра В ВИДЕ ОБЪЕКТА fdesign')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА fdesign нажмите <ENTER>')
pause
MAG_bandstop =
fdesign.bandstop('Fp1,Fst1,Fst2,Fp2,Ap1,Ast,Ap2',ft1,fk1,fk2,ft2,rp,rs,rp,[Fs])
% ОБЪЕКТ fdesign ДЛЯ РФ
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. СИНТЕЗ БИХ-ФИЛЬТРА РФ ЗОЛОТАРЕВА-КАУЭРА В ВИДЕ ОБЪЕКТА dfilt')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt нажмите <ENTER>')
pause
F_bandstop = design(MAG_bandstop,'ellip','MatchExactly',
'both','FilterStructure','df2sos') % РФ В ВИДЕ ОБЪЕКТА dfilt
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.8. МАСШТАБИРОВАНИЕ В КАСКАДНОЙ СТРУКТУРЕ БИХ-ФИЛЬТРА РФ ЗОЛОТАРЕВА-КАУЭРА')
disp('%')
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА dfilt после масштабирования нажмите <ENTER>')
pause
F_bandstop_scale = scale(F_bandstop,'L2') % РЕЗУЛЬТАТ МАСШТАБИРОВАНИЯ
disp('%')
disp('%')
disp('% СИНТЕЗ БИХ-ФИЛЬТРА РФ ЗАВЕРШЕН')

```

13.4.5. Используемые внешние функции

В script-файлах `lr_13_low`, `lr_13_high`, `lr_13_pass` и `lr_13_stop` используется внешняя функция `plot_iir`, предназначенная для вывода графиков АЧХ, ФЧХ, ИХ и карты нулей и полюсов БИХ-фильтра:

```

function plot_iir(b,a,Fs)
% Вывод графиков АЧХ, ФЧХ, ИХ и карты нулей и полюсов БИХ-фильтра
%
% b — вектор коэффициентов числителя передаточной функции
% a — вектор коэффициентов знаменателя передаточной функции

```

```
% Fs — частота дискретизации (Гц)
%
% M — длина ИХ БИХ-фильтра, ограниченная до 50-ти отсчетов
% n — вектор дискретного нормированного времени
% h — вектор отсчетов ИХ
% f — сетка частот (Гц) для расчета АЧХ и ФЧХ
% H — частотная характеристика
% MAG и PHASE — АЧХ и ФЧХ
%
M = 50;
n = 0:(M-1);
h = impz(b,a,M);
f = 0:((Fs/2)/1000):Fs/2;
H = freqz(b,a,f,Fs);
MAG = abs(H);
PHASE = phase(H);
subplot(2,2,1), plot(f,MAG), xlabel('f (Hz)')
title('MAGNITUDE'), grid, ylim([0 1.2])
subplot(2,2,2), zplane(b,a), title('Z-plane zero-pole plot'), grid
subplot(2,2,3), plot(f,PHASE), xlabel('f (Hz)')
title('PHASE'), grid
subplot(2,2,4), stem(n,h,'fill','MarkerSize',3)
xlabel('n'), title('Impulse Response'), grid
```

13.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файлов для синтеза БИХ-фильтра ФНЧ методом билинейного Z-преобразования, анализа его характеристик и моделирования процесса цифровой фильтрации.

Пункты самостоятельного задания включают в себя:

1С. Синтез БИХ-фильтра ФНЧ Золотарева—Кауэра с произвольными требованиями к характеристике затухания (входные параметры function-файла).

Вывести графики ИХ, АЧХ и ФЧХ с помощью function-файла `plot_iir` (см. разд. 13.4.5), который хранится на диске в папке `LAB_DSP\LAB_13`.

Выходными параметрами function-файла являются векторы коэффициентов БИХ-фильтра.

2С. Дублирует п. 2С в разд. 11.5 для БИХ-фильтра (см. п. 1С).

3С. Дублирует п. 2С в разд. 11.5 для БИХ-фильтра (см. п. 1С).

4С. Синтез БИХ-фильтра ФНЧ Золотарева—Кауэра с выбранной структурой звеньев в виде объекта `dfilt` на основе объекта `fdesign` с произвольными требованиями к характеристике затухания (входные параметры function-файла).

Выполнить масштабирование на основе нормы 'L2'.

Выходным параметром function-файла является имя объекта `dfilt`.

13.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения пунктов задания, включая заполненную табл. 13.7, созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**), описания структур БИХ-фильтров в виде объектов `dfilt`, копируемые из окна **Command Window** (шрифт Courier New), и ответы на поставленные вопросы (шрифт Times New Roman).

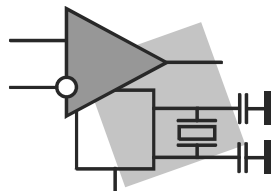
Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. Дайте определение порядка БИХ-фильтра.
2. Как определить устойчивость БИХ-фильтра?
3. Можно ли синтезировать БИХ-фильтр с линейной ФЧХ?
4. Запишите передаточную функцию и разностное уравнение БИХ-фильтра.
5. Дайте определение характеристики затухания АЧХ (дБ).
6. Что входит в требования к характеристике затухания БИХ-фильтра?
7. Перечислите основные этапы процедуры синтеза БИХ-фильтра методом билинейного Z-преобразования.
8. Дайте определение частоты среза.
9. Как связаны граничные частоты АЧХ АФП с граничными частотами АЧХ БИХ-фильтра?
10. Назовите четыре типа БИХ-фильтров и поясните вид их АЧХ.
11. Что отображает структура ЦФ и чем определяется ее вид?
12. Какому виду передаточной функции соответствует каскадная структура БИХ-фильтра?
13. Какую структуру могут иметь биквадратные звенья в каскадной структуре?
14. С какой целью выполняется расстановка и масштабирование звеньев?

13.7. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 14.
2. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов. — 2-е изд. — СПб.: БХВ-Петербург, 2005. — Глава 24.
3. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 6.

ГЛАВА 14



Синтез цифровых фильтров средствами GUI FDATool и FilterBuilder

Цель работы: овладеть средствами GUI FDATool (Filter Design and Analysis Toolbox — средство проектирования и анализа фильтров) и FilterBuilder (Разработчик фильтров) для синтеза и анализа КИХ- и БИХ-фильтров.

14.1. Краткая теоретическая справка

Средства GUI предназначены для моделирования объектов посредством интерактивного общения без прямого доступа к программным средствам MATLAB.

Средства GUI FDATool и FilterBuilder предназначены для проектирования и анализа цифровых фильтров (ЦФ).

14.1.1. Синтез цифровых фильтров в GUI FDATool

Обращение к GUI FDATool происходит по команде:

`fdatool`

после чего открывается окно **Filter Design & Analysis Tool** (Средство проектирования и анализа фильтра) при нажатой кнопке **Design filter** (Синтез фильтра), расположенной на панели инструментов в левом нижнем углу окна (рис. 14.1).

Тип избирательности ЦФ указывается с помощью переключателей в группе **Response Type** (Тип характеристики):

- ☐ **Lowpass** — ФНЧ;
- ☐ **Highpass** — ФВЧ;
- ☐ **Bandpass** — ПФ;
- ☐ **Bandstop** — РФ.

Тип ЦФ выбирается в группе **Design Method** (Метод синтеза):

- ☐ переключатель **FIR** — КИХ-фильтр;
- ☐ переключатель **IIR** — БИХ-фильтр.

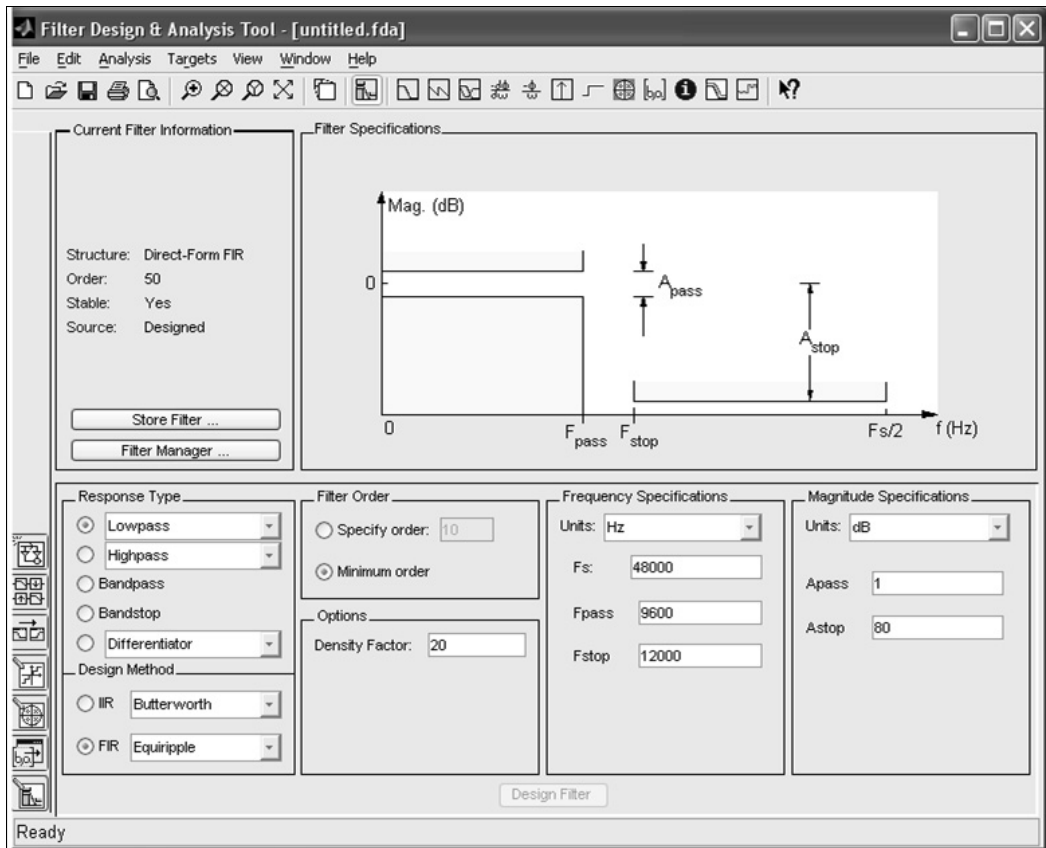


Рис. 14.1. Окно **Filter Design & Analysis Tool** при нажатой кнопке **Design Filter**

Метод синтеза КИХ-фильтра выбирается в раскрывающемся списке **FIR**, например:

- ☐ Window (Метод окон);
- ☐ Equiripple (С равными отклонениями — метод чебышевской аппроксимации).

Метод синтеза БИХ-фильтра выбран по умолчанию (билинейного Z -преобразования), а в раскрывающемся списке **IIR** выбирается *тип* БИХ-фильтра:

- ☐ Butterworth — фильтр Баттерворта;
- ☐ Chebyshev Type I — фильтр Чебышева I рода;
- ☐ Chebyshev Type II — фильтр Чебышева II рода;
- ☐ Elliptic — фильтр Золотарева—Кауэра (эллиптический).

Требования к АЧХ задаются в группах **Frequency Specifications** (Требования к частотам) и **Magnitude Specifications** (Требования к АЧХ):

- ☐ в группе **Frequency Specifications** в списке **Units** (Единицы измерения) выбирается значение Hz (Гц) и задаются частота дискретизации (**Fs**) и граничные частоты ПП (**Fpass**) и ПЗ (**Fstop**);

- в группе **Magnitude Specifications** задаются требования к отклонениям АЧХ в ПП (**Apass**) и ПЗ (**Astop**). Выбор единиц измерения в списке **Units** зависит от *типа* ЦФ, а именно:
 - для КИХ-фильтров требования могут задаваться к нормированной АЧХ — при выборе **Linear** (Безразмерный) или к характеристике затухания АЧХ (дБ) (11.6) — при выборе **dB** (Децибелы);
 - для БИХ-фильтров требования задаются к характеристике затухания АЧХ (дБ) (11.6) — при выборе **dB**.

Синтез ЦФ заключается в расчете его передаточной функции и выполняется после нажатия нижней кнопки **Design Filter** (Синтезировать фильтр).

Сохранение синтезированных ЦФ в буфере Filter Manager (Диспетчер фильтров) выполняется при нажатии кнопки **Store Filter** (Сохранение фильтра) в группе **Current Filter Information** (Информация о текущем фильтре). Имя ЦФ указывается по умолчанию или задается пользователем.

Информация о синтезированном ЦФ выводится в группе **Current Filter Information** (Текущая информация о фильтре):

- **Structure** — структура фильтра;
- **Order** — порядок фильтра;
- **Stable** — устойчивость фильтра: устойчивый (Yes) или неустойчивый (No);
- **Source** — источник получения фильтра: синтезированный (Designed) или импортированный (Imported).

Сеанс работы в GUI FDATool называется *сессией* (Session). По завершении работы в GUI FDATool предусмотрена возможность *сохранения сессии* по команде меню **File | Save Session As** (Файл | Сохранить сессию как).

В открывающемся окне **Save Filter Design Session** (Сохранить сессию проектирования фильтра) указывается *имя сессии* — файла с расширением **fda** и нажимается кнопка **Сохранить**.

При последующих обращениях к GUI FDATool сохраненная сессия открывается по команде меню **File | Open Session** (Файл | Открыть сессию). В окне **Load Filter Design Session** (Загрузить сессию проектирования фильтра) выбирается требуемая папка и в ней — файл с сохраненной сессией.

14.1.2. Экспорт из GUI FDATool в Workspace

Для экспорта ЦФ в виде объекта **dfilt** в рабочее пространство памяти **Workspace** следует:

1. Загрузить экспортируемый ЦФ из буфера **Filter Manager**.
2. Обратиться к команде меню **File | Export** (Файл | Экспорт), после чего откроется окно **Export**.
3. В группе **Export To** (Экспортировать в) выбрать **Workspace**.

4. В группе **Export As** (Экспортировать как) выбрать **Objects** (Объект).
5. В группе **Variable Names** (Имена переменных) в поле ввода **Discrete Filter** (Дискретный фильтр) указать имя объекта `dfilt` и сбросить флажок **Overwrite Variables** (Перезаписать переменные) во избежание конфликта с переменными в **Workspace**.
6. Нажать кнопку **OK**.
7. Проверить содержимое **Workspace**.

14.1.3. Синтез цифровых фильтров в FilterBuilder GUI

Специфика FilterBuilder GUI заключается в том, что по заданным требованиям к АЧХ автоматически формируется объект `fdesign` (см. разд. 12.1.3) и ЦФ синтезируется в виде объекта `dfilt` (см. разд. 12.1.4, 13.1.6).

Обращение к FilterBuilder GUI происходит по команде:

```
filterbuilder
```

В окне **Response Selection** (Выбор характеристики) задается тип избирательности ЦФ, например, **Lowpass** для ФНЧ, после чего открывается окно **Lowpass Design** (рис. 14.2) с тремя вкладками:

- ☐ **Main** (Главное);
- ☐ **Data Types** (Типы данных);
- ☐ **Code Generation** (Генерация кода).

На вкладке **Main** указываются параметры, связанные с синтезом ЦФ:

- ☐ в группе **Filter specifications** (Требования к фильтру) — параметры:
 - **Impulse response** (Импульсная характеристика) — типы ЦФ: **FIR** — КИХ-фильтр; **IIR** — БИХ-фильтр;
 - **Order mode** (Режим для порядка) — режимы определения порядка ЦФ: **Minimum** (Минимальный) и **Specify** (Произвольный);
 - при выборе **Specify** активизируется поле ввода **Order** (Порядок);
 - **Filter type** (Тип фильтра) — назначение КИХ-фильтра:
 - **Single-rate** (Однокоростной) — обычный КИХ-фильтр;
 - **Decimator** (Дециматор) — КИХ-фильтр для системы однократной децимации;
 - **Interpolator** (Интерполятор) — КИХ-фильтр для системы однократной интерполяции;
 - **Sample-rate convertor** (Передискретизатор) — КИХ-фильтр для системы однократной передискретизации.

Моделирование КИХ-фильтров для систем однократной интерполяции, децимации и передискретизации рассматривается в разд. 20.1.2;

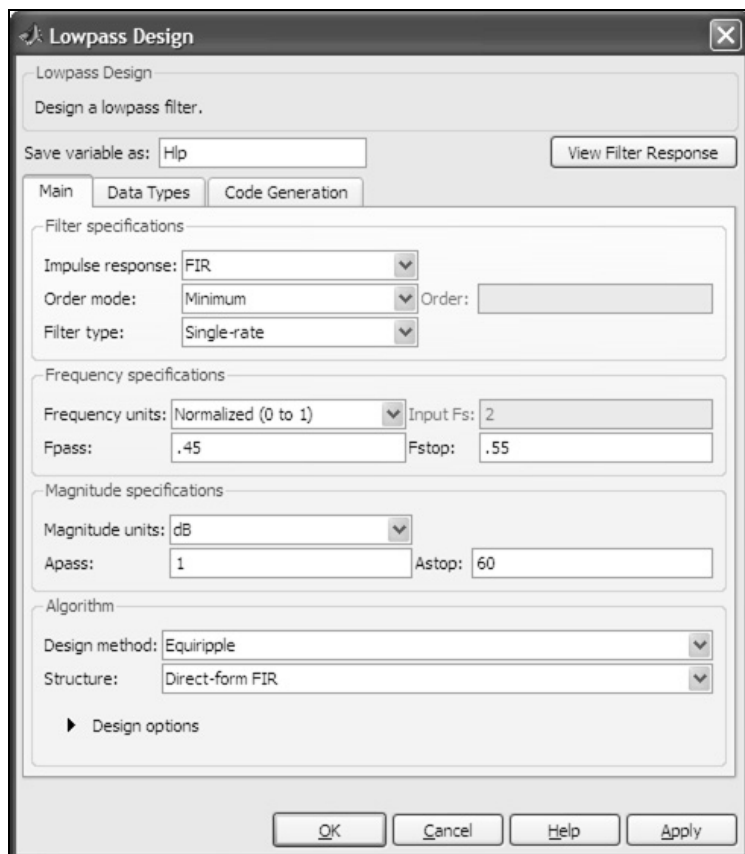


Рис. 14.2. Окно Lowpass Design с открытой вкладкой Main

- ❑ в группе **Frequency specifications** (Требования к частотам) — параметры:
 - **Frequency units** — единицы измерения частот;
 - **Input Fs** (Частота Fs на входе) — частота дискретизации;
 - **Fpass, Fstop** — граничные частоты ПП и ПЗ;
- ❑ в группе **Magnitude specifications** (Требования к АЧХ) — параметры:
 - **Magnitude units** (Единицы измерения АЧХ) — единицы измерения АЧХ:
 - Linear — безразмерная нормированная АЧХ;
 - dB (дБ) — характеристика затухания (11.6) — АЧХ (дБ);
 - **Apass, Astop** — допустимые отклонения в ПП и ПЗ;
- ❑ в группе **Algorithm** (Алгоритм) — параметры:
 - **Design Method** — метод синтеза:
 - Equiripple — метод чебышевской аппроксимации;
 - Kaiser window — метод окон с окном Кайзера.

Для БИХ-фильтров по умолчанию используется метод билинейного Z-преобразования и выбирается тип БИХ-фильтра:

- Butterworth — Баттерворта;
- Chebyshev type I — Чебышева I рода;
- Chebyshev type II — Чебышева II рода;
- Elliptic — Золотарева—Кауэра (эллиптический);
- **Structure** (Структура) — раскрывающийся список для выбора структуры ЦФ; типовые структуры КИХ-фильтров приведены в табл. 11.2, а БИХ-фильтров — в табл. 13.1;
- **Scale SOS filter coefficients to reduce chance to overflow** (Масштабирование коэффициентов звеньев для уменьшения возможности переполнения) — флажок, активный для БИХ-фильтров с каскадной структурой.

При установке флажка (по умолчанию) выполняется *масштабирование коэффициентов передаточной функции ЦФ* для предотвращения или минимизации переполнений на выходе сумматоров при последующей реализации БИХ-фильтра с фиксированной точкой (см. разд. 13.1.7).

Формирование звеньев посредством объединения полюсов с ближайшими нулями и *расстановка звеньев* в порядке возрастания радиусов полюсов для минимизации собственных шумов в БИХ-фильтре с фиксированной точкой выполняются автоматически;

- **Design options** (Параметры проектирования) — список входных параметров, автоматически формируемый в зависимости от типа ЦФ (**Impulse response**), метода синтеза (**Design Method**) и режима определения порядка ЦФ (**Order mode**).

Приведем типовые входные параметры для КИХ- и БИХ-фильтров:

- **Density Factor** (Коэффициент плотности) — коэффициент плотности сетки частот для КИХ-фильтров при выборе в списке **FIR** метода **Equiripple**.
Параметр **Density Factor** тождественен параметру `lgrid` в функции `firpm` (см. разд. 12.1.2);
- **Wpass** и **Wstop** — веса в ПП и ПЗ для КИХ-фильтров при выборе в списке **FIR** метода **Equiripple** и режима для порядка **Specify**.
Веса тождественны параметру `weight` функции `firpm` (см. разд. 12.1.2);
- **Scale Passband** (Масштабирование в ПП) — флажок нормирования АЧХ для КИХ-фильтров при выборе в списке **FIR** метода **Kaiser** и режима для порядка **Minimum**.
Параметр **Scale Passband** тождественен параметру `normalization` в функции `firl` (см. разд. 11.1.5);
- **Match Exactly** (Согласовывать точно) — точное выполнение требований к АЧХ для БИХ-фильтров в ПП и ПЗ (**Both**), в ПП (**Passband**) или в ПЗ (**Stopband**).

Параметр **Match Exactly** тождественен параметру `MATCH` в функциях `butter`, `cheby1`, `cheby2` и `ellip` (см. разд. 13.1.6).

После задания параметров на вкладке **Main** в поле **Save variable as** (Сохранить переменную как) следует указать или выбрать по умолчанию имя объекта `dfilt` для структуры ЦФ и нажать кнопку **Apply** (Применить). Будет выполнен *синтез ЦФ* и одновременно осуществлен *экспорт его структуры* в Workspace. Свойства объекта `dfilt` выводятся по его имени в окне **Command Window**.

Для *анализа характеристик* синтезированного ЦФ предусмотрена кнопка **View Filter Response** (Просмотр характеристик фильтра), при нажатии которой открывается окно **Filter Visualization Tool** (Средство визуализации фильтра) GUI FVTool.

На вкладке **Data Types** в раскрывающемся списке **Arithmetic** (Арифметика) выбирается *тип данных* в структуре синтезированного ЦФ, по умолчанию — `Double precision` (Двойная точность). Тип данных с фиксированной точкой (`Fixed point`) рассматривается в разд. 15.1.4.

На вкладке **Code Generation** (Генерация кода) предусмотрены следующие кнопки для вариантов *экспорта* синтезированного ЦФ:

- ☐ кнопка **Generate HDL** в группе **HDL** — в виде кода на языке VHDL или Verilog;
- ☐ кнопка **Generate M-file** в группе **M-file** — в виде function-файла, описывающего объект `dfilt`;
- ☐ кнопка **Generate Model** в группе **Simulink model** — в виде модели Simulink.

По окончании работы в FilterBuilder GUI нажимается кнопка **OK**.

14.2. Содержание лабораторной работы

Содержание работы связано с синтезом и анализом КИХ- и БИХ-фильтров, средствами GUI FDATool.

Синтез ЦФ в FilterBuilder GUI вынесен в самостоятельное задание.

14.3. Задание на лабораторную работу

Исходные данные для пунктов задания приводятся в табл. 14.1—14.4 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$, и для ФНЧ, ФВЧ, ПФ и РФ включают в себя:

- ☐ требования к АЧХ КИХ-фильтров;
- ☐ требования к характеристике затухания АЧХ (дБ) (11.6) БИХ-фильтров. Значения допустимых затуханий рассчитаны по формулам (11.7)—(11.8).

На прилагаемом компакт-диске в папке `Tables\Tables_14` хранятся табл. 14.1—14.4 исходных данных, примеры их заполнения для $N_{бр} = 1$ и табл. 14.5—14.7 для пунктов задания.

Таблица 14.1. Требования к АЧХ и АЧХ (дБ) ФНЧ

Условные обозначения	Список требований	Задаваемые значения
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}} =$
f_{χ}	Граничная частота ПП	$f_{\chi} = \frac{f_d}{10} + 20N_{\text{бр}} =$
f_k	Граничная частота ПЗ	$f_k = \frac{f_d}{10} + 250 + 25N_{\text{бр}} =$
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$
a_{max}	Максимально допустимое затухание в ПП (дБ)	$a_{\text{max}} = 0,4455$
a_{min}	Минимально допустимое затухание в ПЗ (дБ)	$a_{\text{min}} = 40$

Таблица 14.2. Требования к АЧХ и АЧХ (дБ) ФВЧ

Условные обозначения	Список требований	Задаваемые значения
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}} =$
f_k	Граничная частота ПЗ	$f_{\chi} = \frac{f_d}{10} + 20N_{\text{бр}} =$
f_{χ}	Граничная частота ПП	$f_k = \frac{f_d}{10} + 250 + 25N_{\text{бр}} =$
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$
a_{min}	Минимально допустимое затухание в ПЗ (дБ)	$a_{\text{min}} = 40$
a_{max}	Максимально допустимое затухание в ПП (дБ)	$a_{\text{max}} = 0,4455$

Таблица 14.3. Требования к АЧХ и АЧХ (дБ) ПФ

Условные обозначения	Список требований	Задаваемые значения
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}} =$
f_{-k}	Граничная частота ПЗ1	$f_{-k} = \frac{f_d}{20} + 20N_{\text{бр}} =$
$f_{-\chi}$	Граничная частота ПП1	$f_{-\chi} = \frac{f_d}{20} + 250 + 25N_{\text{бр}} =$
f_{χ}	Граничная частота ПП2	$f_{\chi} = \frac{f_d}{4} + 25N_{\text{бр}} =$
f_k	Граничная частота ПЗ2	$f_k = \frac{f_d}{4} + 250 + 30N_{\text{бр}} =$
δ_{21}	Максимально допустимое отклонение в ПЗ1	$\delta_{21} = 0,01$
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$
δ_{22}	Максимально допустимое отклонение в ПЗ2	$\delta_{22} = 0,01$
$a_{1\min}$	Минимально допустимое затухание в ПЗ1 (дБ)	$a_{1\min} = 40$
$a_{\max}$	Максимально допустимое затухание в ПП (дБ)	$a_{\max} = 0,4455$
$a_{2\min}$	Минимально допустимое затухание в ПЗ2 (дБ)	$a_{2\min} = 40$

Таблица 14.4. Требования к АЧХ и АЧХ (дБ) РФ

Условные обозначения	Список требований	Задаваемые значения
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}} =$
$f_{-\chi}$	Граничная частота ПП1	$f_{-\chi} = \frac{f_d}{20} + 20N_{\text{бр}} =$
f_{-k}	Граничная частота ПЗ1	$f_{-k} = \frac{f_d}{20} + 250 + 25N_{\text{бр}} =$
f_k	Граничная частота ПЗ2	$f_k = \frac{f_d}{4} + 25N_{\text{бр}} =$

Таблица 14.4 (окончание)

Условные обозначения	Список требований	Задаваемые значения
f_{χ}	Граничная частота ПП2	$f_{\chi} = \frac{f_d}{4} + 250 + 30N_{бр} =$
δ_{11}	Максимально допустимое отклонение в ПП1	$\delta_{11} = 0,05$
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$
δ_{12}	Максимально допустимое отклонение в ПП2	$\delta_{12} = 0,05$
a_{1max}	Максимально допустимое затухание в ПП1 (дБ)	$a_{1max} = 0,4455$
a_{min}	Минимально допустимое затухание в ПЗ (дБ)	$a_{min} = 40$
a_{2max}	Максимально допустимое затухание в ПП2 (дБ)	$a_{2max} = 0,4455$

Задание на лабораторную работу заключается в синтезе и анализе КИХ- и БИХ-фильтров в GUI FDATool и для каждого типа избирательности (ФНЧ, ФВЧ, ПФ и РФ) включает в себя выполнение следующих пунктов:

1. Синтез КИХ-фильтра методом окон с использованием окна Кайзера.

Для метода окон (Window) следует в группе **Options** (Параметры) в списке **Window** выбрать Kaiser (Окно Кайзера) и сбросить флажок **Scale Passband** (Масштабирование). Действие этого флажка тождественно параметру `normalization` в функции `firl` (см. разд. 11.1.5).

Итерационная процедура синтеза КИХ-фильтра методом окон включает в себя следующие шаги.

- Задание требований к АЧХ.
- Оценка порядка КИХ-фильтра.

В группе **Filter Order** (Порядок фильтра) выбрать переключатель **Minimum order** (Минимальный порядок) и выполнить синтез КИХ-фильтра.

Порядок синтезируемого КИХ-фильтра по умолчанию определяется с точностью до ± 2 , т. е. производится оценка порядка.

- Проверка выполнения требований к АЧХ КИХ-фильтра.

По умолчанию в графическом окне выводится характеристика ослабления (11.5). Для вывода нормированной АЧХ следует обратиться к команде меню **Analysis | Analysis Parameters** (Анализ | Параметры анализа), в раскрывающемся списке **Magnitude Display** (Изображение АЧХ) выбрать **Magnitude** и нажать кнопку **OK**.

Для выделения АЧХ в ПП и ПЗ использовать кнопку **Zoom in** на панели инструментов.

В каждой из этих полос, отжав кнопку **Zoom in**, следует подвести курсор к максимальному (по модулю) отклонению АЧХ и выполнить щелчок левой кнопкой мыши. Рядом с выделенной точкой будут выданы ее координаты — значения частоты и АЧХ. Максимальные (по модулю) отклонения АЧХ следует сравнить с заданными максимально допустимыми отклонениями.

- Уточнение порядка КИХ-фильтра.

Для определения *минимального* порядка КИХ-фильтра в группе **Filter Order** следует установить переключатель **Specify order** (Произвольный порядок) и в поле ввода указать порядок — больший, если требования к АЧХ не выполняются, и меньший, если выполняются.

При уменьшении/увеличении порядка КИХ-фильтра необходимо помнить о соответствии между типом избирательности фильтра и типом КИХ-фильтра (см. табл. 11.1).

- Указание входных параметров метода.

При установке переключателя **Specify order** для метода окон определяются следующие входные параметры:

- в группе **Options** в поле ввода **Beta** — значение параметра β окна Кайзера;
- в группе **Frequency Specifications** — частота дискретизации **Fs** и частоты разрыва **Fc**, одна — для ФНЧ и ФВЧ или две — для ПФ и РФ.

Входные параметры метода окон соответствуют входным параметрам функции `firl` (см. разд. 11.1.5): **Beta** — `beta`; **wc** — нормализованным частотам разрыва `wc` (11.14)—(11.16) при выборе в списке **Units** значения **Normalized (0 to 1)**. Значения `beta` и `wc` можно выбрать из табл. 11.7, если она была заполнена, или рассчитать с помощью функции `kaiserord` (см. разд. 11.1.5) в окне **Command Window**. Частоты разрыва **Fc** (Гц) определяются по формулам (11.11)—(11.13).

- Синтез КИХ-фильтра.

После указания уточненного порядка КИХ-фильтра и входных параметров метода выполняется его синтез.

По завершении итерационной процедуры будет синтезирован КИХ-фильтр минимального порядка $R_{\min}$ при заданных требованиях к АЧХ.

По результатам синтеза заполнить табл. 14.5.

- Выбор структуры КИХ-фильтра.

По умолчанию для КИХ-фильтра выбирается прямая структура Direct-Form FIR (см. табл. 11.2).

Выполнить ее преобразование в прямую приведенную структуру Direct-Form Symmetric FIR по команде меню **Edit | Convert Structure** (Редактирование | Преобразование структуры).

Синтезированные КИХ-фильтры сохранять в буфере **Filter Manager** с именами:

FIR_<тип избирательности>_Window.

Например, КИХ-фильтр ФНЧ:

FIR_lowpass_Window.

- Экспорт КИХ-фильтра Workspace в виде объекта `dfilt` с тем же именем (см. разд. 14.1.3).

Пояснить:

- в расчете чего заключается синтез ЦФ и к чему он сводится для КИХ-фильтра;
- какие типы КИХ-фильтров можно использовать в методе окон;
- свойства объекта `dfilt`, выведенные в окне **Command Window**.

Таблица 14.5. Результаты синтеза КИХ-фильтров методом окон

Тип избирательности фильтра	Метод окон с окном Кайзера	
	порядок фильтра $R_{\min}$	тип КИХ-фильтра
ФНЧ		
ФВЧ		
ПФ		
РФ		

2. Анализ характеристик КИХ-фильтра, синтезированного методом окон.

Проанализировать вид АЧХ, ФЧХ и ИХ КИХ-фильтра, обращаясь к командам меню **Analysis**.

Пояснить:

- вид АЧХ в ПП и ПЗ;
- вид ИХ;
- вид ФЧХ (линейная, нелинейная, в каких точках имеет скачки на π).

3. Синтез оптимального КИХ-фильтра методом чебышевской аппроксимации.

Для метода чебышевской аппроксимации (Equiripple) в группе **Options** в поле ввода **Density Factor** (Коэффициент плотности сетки частот) следует указать значение 20 (по умолчанию). Этот параметр тождественен параметру `lgrid` в функции `firpm` (см. разд. 12.1.2).

Итерационная процедура синтеза КИХ-фильтра методом чебышевской аппроксимации включает в себя следующие шаги.

- Задание требований к АЧХ.
- Оценка порядка КИХ-фильтра (см. п. 1).
- Проверка выполнения требований к АЧХ КИХ-фильтра (см. п. 1).
- Уточнение порядка КИХ-фильтра.

Для определения *оптимального* порядка КИХ-фильтра в группе **Filter Order** следует установить переключатель **Specify order** и в поле ввода указать порядок — больший, если требования к АЧХ не выполняются, и меньший, если выполняются.

При уменьшении/увеличении порядка КИХ-фильтра необходимо помнить о соответствии между типом избирательности фильтра и типом КИХ-фильтра (см. табл. 11.1).

- Указание входных параметров метода.

При установке переключателя **Specify order** для метода чебышевской аппроксимации входные параметры зависят от *типа КИХ-фильтра* (см. табл. 11.1).

Для КИХ-фильтров 1-го или 2-го типа определяются следующие входные параметры:

- в группе **Frequency Specifications** — частота дискретизации и граничные частоты ПП и ПЗ;
- в группе **Magnitude Specifications** — значения весов в ПП (**Wpass**) и ПЗ (**Wstop**).

Для КИХ-фильтров 3-го типа (для ПФ) или 4-го типа (для ФВЧ и ПФ) в группе **Response Type** следует установить переключатель **Hilbert Transformer** (Преобразователь Гильберта) и в группе **Frequency and Magnitude Specifications** определить следующие входные параметры:

- в поле ввода **Fs** — частота дискретизации ($\Gamma_{ц}$);
- в поле ввода **Freq. vector** — вектор частот ($\Gamma_{ц}$): для ФВЧ — $[0 \ f_k \ f_{\chi} \ f_d/2]$; для ПФ — $[0 \ f_{-k} \ f_{-\chi} \ f_{\chi} \ f_k \ f_d/2]$;
- в поле ввода **Mag. vector** — вектор значений идеальной АЧХ: для ФВЧ — $[0 \ 0 \ 1 \ 1]$; для ПФ — $[0 \ 0 \ 1 \ 1 \ 0 \ 0]$;
- в поле ввода **Weight vector** — вектор весов в ПП и ПЗ в порядке следования слева направо: для ФВЧ — двухэлементный вектор, для ПФ — трехэлементный вектор.

Весы соответствуют входному параметру `weight` функции `firpm` (см. разд. 12.1.2), их значения можно выбрать из табл. 12.12, если она была заполнена, или рассчитать самостоятельно (см. разд. 12.1.1).

- Синтез КИХ-фильтра.
После указания уточненного порядка КИХ-фильтра и входных параметров метода выполняется его синтез.
По завершении итерационной процедуры будет синтезирован оптимальный КИХ-фильтр порядка R_{opt} при заданных требованиях к АЧХ.
По результатам синтеза заполнить табл. 14.6.
- Выбор структуры КИХ-фильтра.
По умолчанию для КИХ-фильтра выбирается прямая структура Direct-Form FIR (см. табл. 11.2).
Выполнить ее преобразование в прямую приведенную структуру Direct-Form Symmetric FIR (для КИХ-фильтров 1-го или 2-го типа) или Direct-Form Antisymmetric FIR (для КИХ-фильтров 3-го или 4-го типа) по команде меню **Edit | Convert Structure**.
Синтезированные КИХ-фильтры сохранять в буфере **Filter Manager** с именами:
FIR_<тип избирательности>_Equiripple.
Например, КИХ-фильтр ФНЧ:
FIR_lowpass_Equiripple.
- Экспорт КИХ-фильтра в Workspace в виде объекта dfilt с тем же именем.
Пояснить:
 - какие типы КИХ-фильтров можно использовать в методе чебышевской аппроксимации;
 - соответствие между порядками R_{min} в табл. 14.5 и R_{opt} в табл. 14.6;
 - свойства объекта dfilt, выведенные в окне **Command Window**.

Таблица 14.6. Результаты синтеза оптимальных КИХ-фильтров

Тип избирательности фильтра	Метод чебышевской аппроксимации	
	порядок фильтра R_{opt}	тип КИХ-фильтра
ФНЧ		
ФВЧ		
ПФ		
РФ		

4. Анализ характеристик КИХ-фильтра, синтезированного методом чебышевской аппроксимации.

Проанализировать АЧХ, ФЧХ и ИХ КИХ-фильтра, обращаясь к командам меню **Analysis**.

Пояснить:

- вид АЧХ в ПП и ПЗ;
- вид ИХ;
- вид ФЧХ (линейная, нелинейная, в каких точках имеет скачки на π).

5. Синтез БИХ-фильтров методом билинейного Z-преобразования.

Выполнить для БИХ-фильтров Баттерворта, Чебышева I и II рода и Золотарева—Кауэра.

В группе **Filter Order** установить переключатель **Minimum order**.

Процедура синтеза БИХ-фильтра методом билинейного Z-преобразования (по умолчанию) включает в себя следующие шаги.

- Задание требований к характеристике затухания.
- Выбор типа БИХ-фильтра.
- Указание входных параметров метода.

В группе **Options** в списке **Match Exactly** (Согласовывать точно) рекомендуется выбрать значение **passband** для фильтров Баттерворта, Чебышева I и II рода и значение **both** для фильтра Золотарева—Кауэра.

Значения в списке **Match Exactly** тождественны значениям параметра **MATCH** в функциях синтеза БИХ-фильтров (см. разд. 13.1.6).

- Синтез БИХ-фильтра.

Синтезированные БИХ-фильтры сохранить в буфере **Filter Manager** с именами:

`PR_<тип избирательности>_<тип фильтра>`.

Например, БИХ-фильтр ФНЧ Золотарева—Кауэра:

`PR_lowpass_Elliptic`.

По результатам синтеза заполнить табл. 14.7.

- Выбор структуры БИХ-фильтра.

Выбрать каскадную структуру **Direct-form II, Second-order sections** (см. табл. 13.1) (по умолчанию).

При необходимости преобразование структуры выполняется по команде меню **Edit | Convert Structure**.

Пояснить:

- в расчете чего заключается синтез БИХ-фильтра;
- какой из типов БИХ-фильтра имеет наименьший порядок;
- что отображает структура и чем определяется ее вид.

Таблица 14.7. Результаты синтеза БИХ-фильтров

Тип избирательности фильтра	Тип и порядок БИХ-фильтра			
	Баттерворта	Чебышева I рода	Чебышева II рода	Золотарева—Кауэра
	R1	R2	R3	R4
ФНЧ				
ФВЧ				
ПФ				
РФ				

6. Анализ характеристик БИХ-фильтров.

Проанализировать АЧХ, ФЧХ и ИХ БИХ-фильтров ФНЧ Баттерворта, Чебышева I и II рода и Золотарева—Кауэра, обращаясь к командам меню **Analysis**.

Пояснить:

- вид АЧХ в ПП и ПЗ;
- вид ИХ;
- вид ФЧХ (линейная, нелинейная, в каких точках имеет скачки на π).

7. Расстановка звеньев и масштабирование в каскадной структуре БИХ-фильтра.

Выполнить для БИХ-фильтров Баттерворта, Чебышева I и II рода и Золотарева—Кауэра.

По команде меню **Edit | Reorder and Scale Second-Order Sections** (Редактирование | Расстановка и масштабирование звеньев 2-го порядка) открыть окно **Reordering and Scaling of Second-Order Sections**.

В группе **Reordering** (Расстановка) установить переключатель **Auto** для автоматического формирования звеньев посредством объединения полюсов с ближайшими нулями и расстановки звеньев в порядке возрастания радиусов полюсов.

В группе **Scaling** (Масштабирование) выполнить следующие действия:

- установить флажок **Scale** (Масштаб);
- установить бегунок в положение **Lin f** — для ФНЧ и ПФ и **L2** — для ФВЧ и РФ для масштабирования на основе норм $\|\mathbf{x}\|_\infty$ и $\|\mathbf{x}\|_2$, что эквивалентно масштабированию с помощью функции `scale` (см. разд. 13.1.7);
- в поле ввода **Maximum Numerator** (Максимальное значение числителя) установить 1;
- в списке **Numerator Constraint** (Ограничение числителя) выбрать **None**, что соответствует отсутствию ограничений на масштабирование коэффициентов числителя;

- в списке **Overflow Mode** (Режим переполнения) выбрать **Wrap**.
Параметр **Overflow Mode** управляет переполнением при сохранении данных для БИХ-фильтров с фиксированной точкой, о чем пойдет речь далее (см. разд. 15.1.3.3);
- в списке **Scale Value Constraint** (Ограничение коэффициента усиления) выбрать **Unit** — максимальный коэффициент усиления равен единице.

Нажать кнопку **Apply** (Применить) и затем кнопку **ОК**.

Проверить коэффициенты БИХ-фильтра до и после масштабирования по команде меню **Analysis | Filter Coefficients** (Анализ | Коэффициенты фильтра).

БИХ-фильтры *после масштабирования* сохранить в буфере **Filter Manager** с именами:

`PIR_<тип избирательности>_<тип фильтра>_Scale`.

Например, БИХ-фильтр ФНЧ Золотарева—Кауэра после масштабирования:

`PIR_lowpass_Elliptic_Scale`.

Пояснить, с какой целью выполняется масштабирование в каскадной структуре.

8. Экспорт БИХ-фильтров в Workspace.

Экспортировать БИХ-фильтры в виде объектов `dfilt` с теми же именами.

Пояснить свойства объекта `dfilt`, выведенные в окне **Command Window**.

9. Анализ взаимосвязи между порядком ЦФ и требованиями к АЧХ.

Выполнить для КИХ-фильтра ФНЧ, синтезированного методом чебышевской аппроксимации (см. п. 3).

Синтезировать КИХ-фильтры *без сохранения имен в буфере* **Filter Manager** и записать их порядки:

- при *увеличении вдвое частоты дискретизации* и неизменности остальных требований к АЧХ;
- при исходной частоте дискретизации, *увеличенной вдвое граничной частоте* ПП и неизменности остальных требований к АЧХ;
- при исходных частоте дискретизации, граничных частотах ПП и ПЗ и *уменьшенном вдвое максимально допустимом отклонении* в ПП.

Пояснить, как и почему изменился порядок фильтра в каждом из этих случаев.

10. Сохранение сессии FDATool.

Сохранить сессию с именем `Filters.fda`.

14.4. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в синтезе КИХ- и БИХ-фильтров, рассмотренных ранее, в FilterBuilder GUI и их экспорте в Workspace в виде объектов `dfilt`.

14.5. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения пунктов задания, включая заполненные табл. 14.5—14.8, свойства объектов `dfilt`, копируемые из окна **Command Window** (шрифт Courier New), необходимые графики (копируются при нажатии комбинации клавиш `<Alt>+<Print Screen>`) и ответы на поставленные вопросы (шрифт Times New Roman).

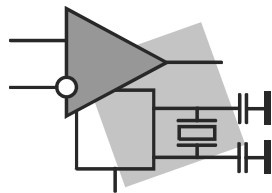
Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. Для чего предназначены средства GUI FDATool и FilterBuilder?
2. Как называется сеанс работы в GUI FDATool?
3. Как сохранить результаты сеанса работы в GUI FDATool?
4. Как указывается тип избирательности ЦФ в GUI FDATool?
5. Как указывается тип ЦФ в GUI FDATool?
6. Чем определяется целесообразность выбора типа ЦФ (КИХ или БИХ)?
7. Как выбирается метод синтеза ЦФ в GUI FDATool?
8. Что в себя включают требования к АЧХ и характеристике затухания?
9. Как осуществляется экспорт синтезированного ЦФ из FDATool в Workspace?
10. Как вывести свойства объекта `dfilt` в окне **Command Window**?

14.6. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 20.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 6.

ГЛАВА 15



Цифровые фильтры с фиксированной точкой

Цель работы: изучить влияние эффектов квантования на характеристики цифровых фильтров (ЦФ) и овладеть средствами GUI FDATool (Filter Design and Analysis Toolbox — средство проектирования и анализа фильтров) и FilterBuilder (Разработчик фильтров) для моделирования структур ЦФ с фиксированной точкой (ЦФ с ФТ).

15.1. Краткая теоретическая справка

До сих пор рассматривались структуры ЦФ в предположении, что все данные (входные, выходные, результаты промежуточных арифметических операций) представлены числами бесконечной разрядности. В MATLAB таким данным условно сопоставлены числа максимально возможной разрядности типа `double`.

Реализация ЦФ предполагает моделирование его структуры с учетом эффектов квантования данных.

Квантованием данных называют их представление с помощью конечного числа двоичных разрядов (битов).

В этой главе рассматриваются эффекты квантования в ЦФ, в структуре которого все данные представлены в форме с *фиксированной точкой* (ФТ).

Форме с ФТ соответствует следующее распределение битов в ячейке памяти (рис. 15.1): старший бит — знаковый, следующие биты — значащие. Между знаковым и значащими битами логически фиксируется точка (запятая), отделяющая *целую часть*, по умолчанию *равную нулю*, от дробной, представленной значащими битами.

Формат представления данных с ФТ отображает *измеряемую в битах длину* представления данных в ячейках памяти и регистрах. Различают следующие основные разновидности форматов данных:

- слово — для представления данных в ячейках памяти;
- двойное слово MSB : LSB — для представления результатов умножения (произведения).

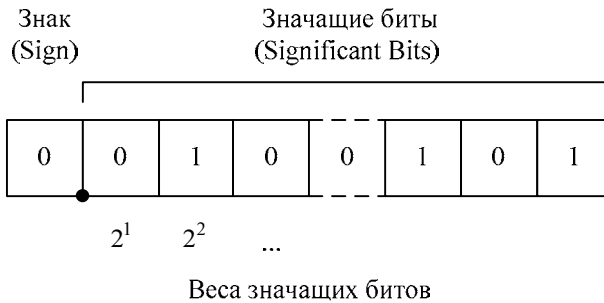


Рис. 15.1. Распределение битов в ячейке памяти с ФТ

Обозначение MSB соответствует старшему слову (Most Significant Bits), а LSB — младшему слову (Least Significant Bits). *Знаковым* является старший бит MSB : LSB;

- расширенное слово EXT : MSB : LSB — для представления результатов сложения (суммы) в аккумуляторе (Accumulator).

Обозначение EXT соответствует битам расширения (Extension). Их назначение зависит от *целой части* суммы: если она *равна нулю*, то биты EXT хранят расширение (дублирование) знака суммы — старшего бита MSB : LSB, а если *не равна нулю*, то старший бит EXT хранит *знак* суммы, остальные биты EXT *плюс старший бит* MSB : LSB — ее целую часть, а остальные биты MSB : LSB — дробную часть.

15.1.1. Эффекты квантования в структуре ЦФ с ФТ

Нелинейная процедура квантования сопровождается внесением в структуру ЦФ ошибок, источниками которых являются:

- аналого-цифровой преобразователь (АЦП) — в каждый момент дискретного времени вносит ошибку квантования, равную разности между дискретным и цифровым сигналами.

Ошибки квантования, обусловленные АЦП, называют *шумом АЦП*;

- умножители — в каждый момент дискретного времени вносят ошибку квантования, обусловленную округлением результата умножения (произведения), представленного в формате двойного слова MSB : LSB, при его сохранении в ячейке памяти в формате слова MSB.

Ошибки квантования, обусловленные умножителями, называют *собственным шумом*;

- квантование коэффициентов передаточной функции — вносит не зависящую от времени *ошибку квантования*;
- сумматоры — могут вносить *ошибку переполнения* при сохранении результата сложения (суммы), представленного в формате расширенного слова EXT : MSB : LSB, в ячейке памяти в формате слова MSB. Ошибка переполнения

возникает в том случае, если сумма по модулю превосходит единицу. Знаковый бит MSB : LSB оказывается значащим и результат операции недостоверным.

Ошибки переполнения можно предотвратить либо минимизировать, в то время как *ошибки квантования* принципиально неустранимы, поэтому необходим анализ их влияния на характеристики фильтра.

15.1.2. Моделирование структуры исходного ЦФ в GUI FDATool

Моделирование ЦФ с ФТ в GUI FDATool начинается с моделирования структуры *исходного ЦФ*, которое включает в себя:

- ☐ синтез ЦФ;
- ☐ выбор структуры ЦФ;
- ☐ для БИХ-фильтров с каскадной структурой — расстановку звеньев и масштабирование.

Исходные ЦФ сохраняются в буфере **Filter Manager** (Диспетчер фильтров) с заданными пользователем или выбранными по умолчанию именами.

15.1.3. Моделирование структуры ЦФ с ФТ в GUI FDATool

Для моделирования структуры ЦФ с ФТ в GUI FDATool на основе исходного ЦФ необходимо выполнить следующие действия (рис. 15.2):

1. Исходный ЦФ загрузить из буфера **Filter Manager** (для БИХ-фильтров с каскадной структурой — после масштабирования).
2. Нажать кнопку **Set quantization parameters** (Установка параметров квантования) на панели инструментов в левом нижнем углу.

Нижняя половина окна будет содержать раскрывающийся список **Filter arithmetic** (Тип арифметики).

3. Выбрать значение Fixed-point (Фиксированная точка) в раскрывающемся списке **Filter arithmetic**.

Нижняя половина окна изменится — в правой части появятся три вкладки:

- **Coefficients** (Коэффициенты) — для установки свойств коэффициентов передаточной функции;
- **Input/Output** (Вход/Выход) — для установки свойств входного и выходного сигналов;
- **Filter Internals** (Внутреннее состояние фильтра) — для установки свойств арифметических операций.

Элементы управления на данных вкладках контекстно связаны со свойствами ЦФ с ФТ в виде объекта `dfilt` (см. разд. 11.1.3 и 13.1.2) при выборе Arithmetic: 'fixed' [1].

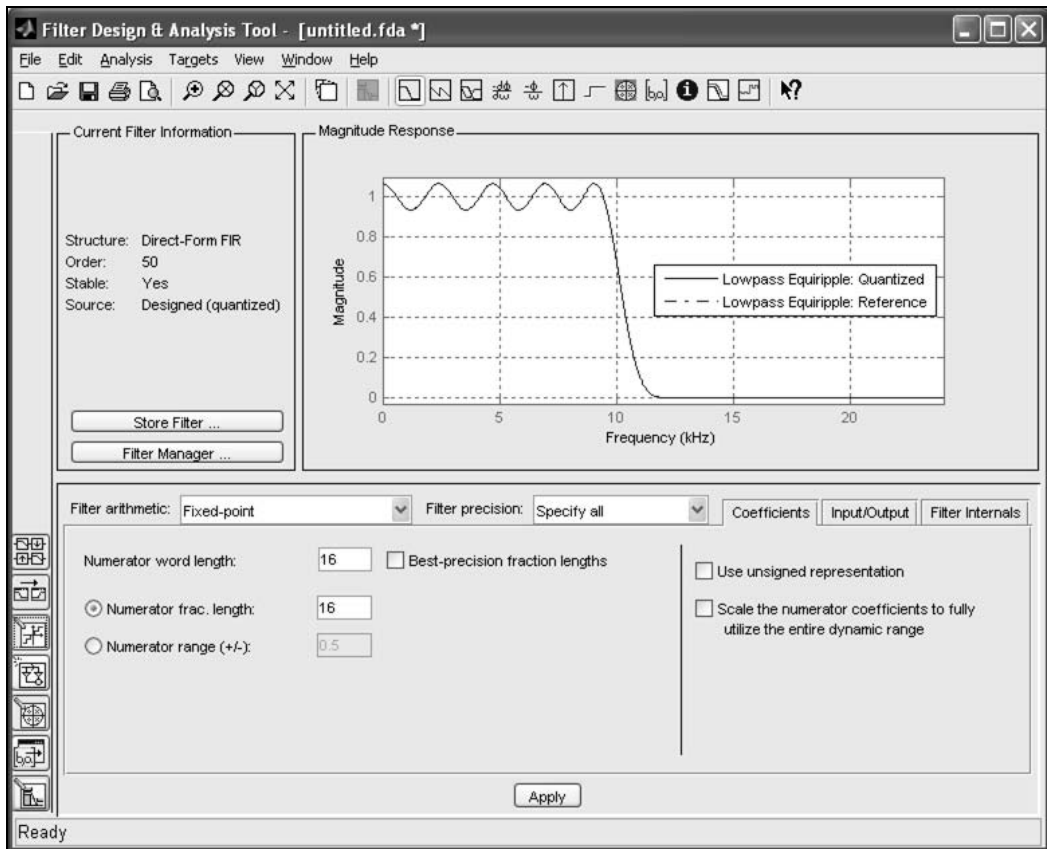


Рис. 15.2. Окно **Filter Design & Analysis Tool** при нажатой кнопке **Set quantization parameters** и открытой вкладке **Coefficients** для КИХ-фильтра с ФТ

4. Установить требуемые свойства ЦФ с ФТ на вкладках **Coefficients**, **Input/Output** и **Filter Internals** (см. разд. 15.1.3.1—15.1.3.3).
5. Нажать кнопку **Apply** (Применить), после чего будет выполнено моделирование структуры ЦФ с ФТ.
6. Проанализировать характеристики созданного ЦФ с ФТ с помощью команд меню **Analysis**.

В поле графика одновременно выводятся графики характеристик ЦФ с ФТ (непрерывной линией) и исходного ЦФ (пунктирной линией).

Размещение легенды выполняется по команде меню **View | Legend** (Вид | Легенда).

В результате квантования коэффициентов передаточной функции АЧХ ЦФ изменяется, поэтому необходимо *контролировать выполнение требований к АЧХ*.

15.1.3.1. Установка свойств ЦФ с ФТ на вкладке **Coefficients**

На вкладке **Coefficients** (см. рис. 15.2) устанавливаются свойства коэффициентов передаточной функции в структуре ЦФ с ФТ, соответствующие свойствам объекта `dfilt` при выборе `Arithmetic: 'fixed'` [1].

Для *КИХ-фильтров с ФТ* перед установкой свойств следует в раскрывающемся списке **Filter precision** (Точность фильтра) выбрать режим для точности представления данных:

- ☐ Full — с максимально возможной точностью;
- ☐ Specify all — с точностью, определяемой пользователем.

Список свойств *КИХ-фильтров с ФТ* на вкладке **Coefficients** при выборе режима `Specify all` (см. рис. 15.2) представлен в табл. 15.1.

Таблица 15.1. Свойства КИХ-фильтров с ФТ на вкладке **Coefficients**

Свойство КИХ-фильтра с ФТ	Свойство объекта <code>dfilt</code>
Numerator word length (Длина слова числителя) — длина слова для коэффициентов	<code>CoeffWordLength</code>
Best-precision fraction lengths (Длина дробной части при наилучшей точности) — флажок управления длиной дробной части слова для коэффициентов. При установке флажка длина дробной части рассчитывается автоматически для представления коэффициентов с максимальной точностью без переполнений	<code>CoeffAutoScale</code>
Numerator frac. length (Длина дробной части числителя) — длина дробной части слова для коэффициентов	<code>NumFracLength</code>
Numerator range (+/-) (Диапазон значений числителя) — положительное число для диапазона, в котором автоматически определяется длина дробной части слова для коэффициентов. Например, при Numerator range , равном 1, длина дробной части будет выбираться из диапазона $[-1; 1]$ ¹	—
Use unsigned representation (Использовать беззнаковое представление) — флажок управления знаковым/беззнаковым представлением коэффициентов. При установке флажка коэффициенты представляются без знака (положительные) и старший бит считается значащим, а при сбросе — со знаком, где старший бит — знаковый	<code>Signed</code>

¹ В MATLAB квантованные данные представляются в диапазоне $[-1; 1]$, а не $[-1; 0,999\dots]$, как в ЦПИОС с ФТ в дополнительном коде.

Таблица 15.1 (окончание)

Свойство КИХ-фильтра с ФТ	Свойство объекта dfilt
Scale the numerator coefficients to fully utilize the entire dynamic range (Масштабировать коэффициенты в соответствии с заданным динамическим диапазоном) — флажок, при установке которого коэффициенты масштабируются так, чтобы обеспечить максимальный динамический диапазон — максимальное по модулю значение коэффициента в заданном формате, соответствующее 1 в формате с ПТ. Значения АЧХ умножаются на коэффициент, равный отношению максимальных коэффициентов после и до масштабирования	—

Список свойств *БИХ-фильтров с ФТ* на вкладке **Coefficients** (рис. 15.3) представлен в табл. 15.2.

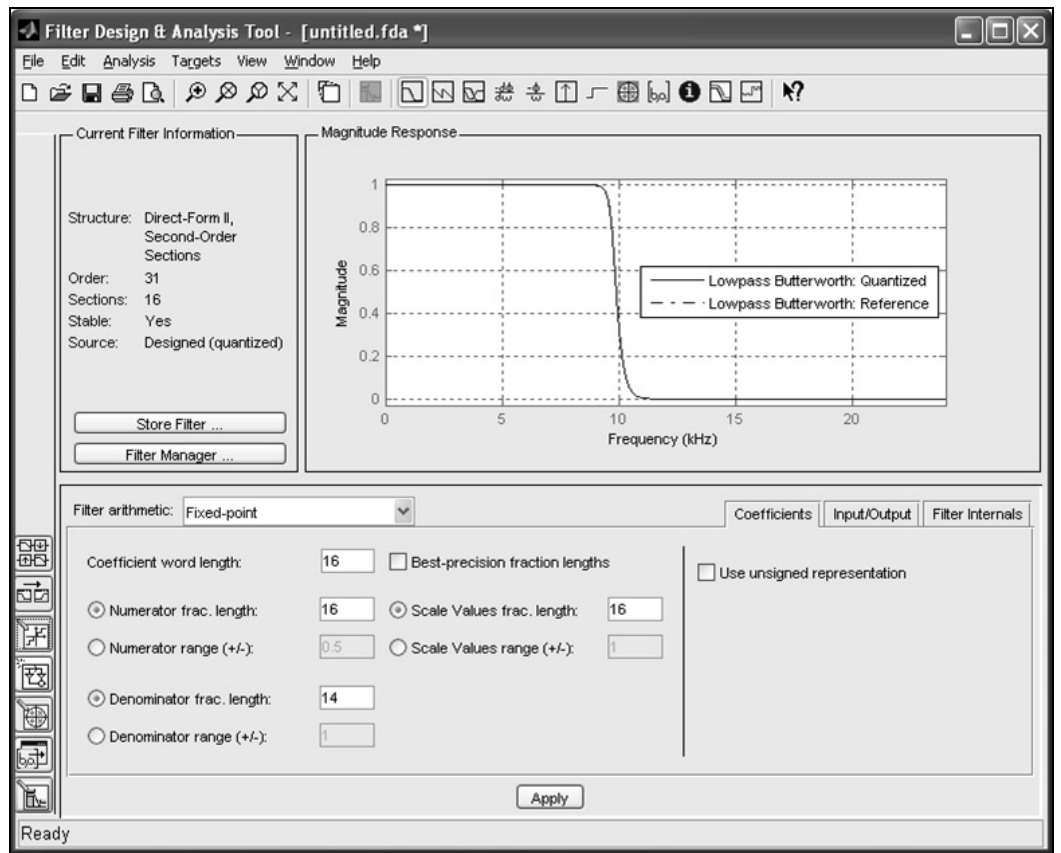


Рис. 15.3. Окно **Filter Design & Analysis Tool** при нажатой кнопке **Set quantization parameters** и открытой вкладке **Coefficients** для БИХ-фильтра с ФТ

Таблица 15.2. Свойства БИХ-фильтров с ФТ на вкладке **Coefficients**

Свойство БИХ-фильтра с ФТ	Свойство объекта <code>dfilt</code>
Coefficient word length (Длина слова коэффициентов) — длина слова для коэффициентов числителя и знаменателя	<code>CoeffWordLength</code>
Best-precision fraction lengths — см. табл. 15.1	<code>CoeffAutoScale</code>
Numerator frac. length (Длина дробной части числителя) — длина дробной части слова для коэффициентов числителя	<code>NumFracLength</code>
Numerator range (+/-) — см. табл. 15.1 для коэффициентов числителя	—
Denominator frac. length (Длина дробной части знаменателя) — длина дробной части слова для коэффициентов знаменателя	<code>DenFracLength</code>
Denominator range (+/-) (Диапазон значений знаменателя) — аналогично свойству Numerator frac. length (см. табл. 15.1) для коэффициентов знаменателя	—
Scale Values frac. length (Длина дробной части слова коэффициентов усиления) — длина дробной части слова для коэффициентов усиления в каскадной структуре	<code>ScaleValueFracLength</code>
Scale Values range (+/-) (Диапазон значений коэффициентов усиления) — аналогично свойству Numerator frac. length (см. табл. 15.1) для коэффициентов усиления	—
Use unsigned representation — флажок управления знаковым/беззнаковым представлением коэффициентов числителя и знаменателя (см. табл. 15.1)	<code>Signed</code>

15.1.3.2. Установка свойств ЦФ с ФТ на вкладке *Input/Output*

На вкладке **Input/Output** (рис. 15.4) устанавливаются свойства входного и выходного сигналов ЦФ с ФТ, соответствующие свойствам объекта `dfilt` при выборе `Arithmetic: 'fixed' [1]`.

Для *КИХ-фильтров с ФТ* перед установкой свойств следует в раскрывающемся списке **Filter precision** выбрать режим для точности представления данных (см. разд. 15.1.3.1).

Список свойств *КИХ-фильтров с ФТ* на вкладке **Input/Output** (см. рис. 15.4) при выборе режима `Specify all` представлен в табл. 15.3.

Таблица 15.3. Свойства КИХ-фильтров с ФТ на вкладке **Input/Output**

Свойство КИХ-фильтра с ФТ	Свойство объекта <code>dfilt</code>
Input word length (Длина слова на входе) — длина слова для значений входного сигнала	<code>InputWordLength</code>

Таблица 15.3 (окончание)

Свойство КИХ-фильтра с ФТ	Свойство объекта <code>dfilt</code>
Input fraction length (Длина дробной части на входе) — длина дробной части слова для значений входного сигнала	<code>InputFracLength</code>
Input range (+/-) (Диапазон значений на входе) — аналогично свойству Numerator frac. length (см. табл. 15.1) для значений входного сигнала	—
Output word length (Длина слова на выходе) — длина слова для значений выходного сигнала	<code>OutputWordLength</code>
Output fraction length (Длина дробной части на выходе) — длина дробной части слова для значений выходного сигнала	<code>OutputFracLength</code>
Output range (+/-) (Диапазон значений на выходе) — аналогично свойству Numerator frac. length (см. табл. 15.1) для выходного сигнала	—

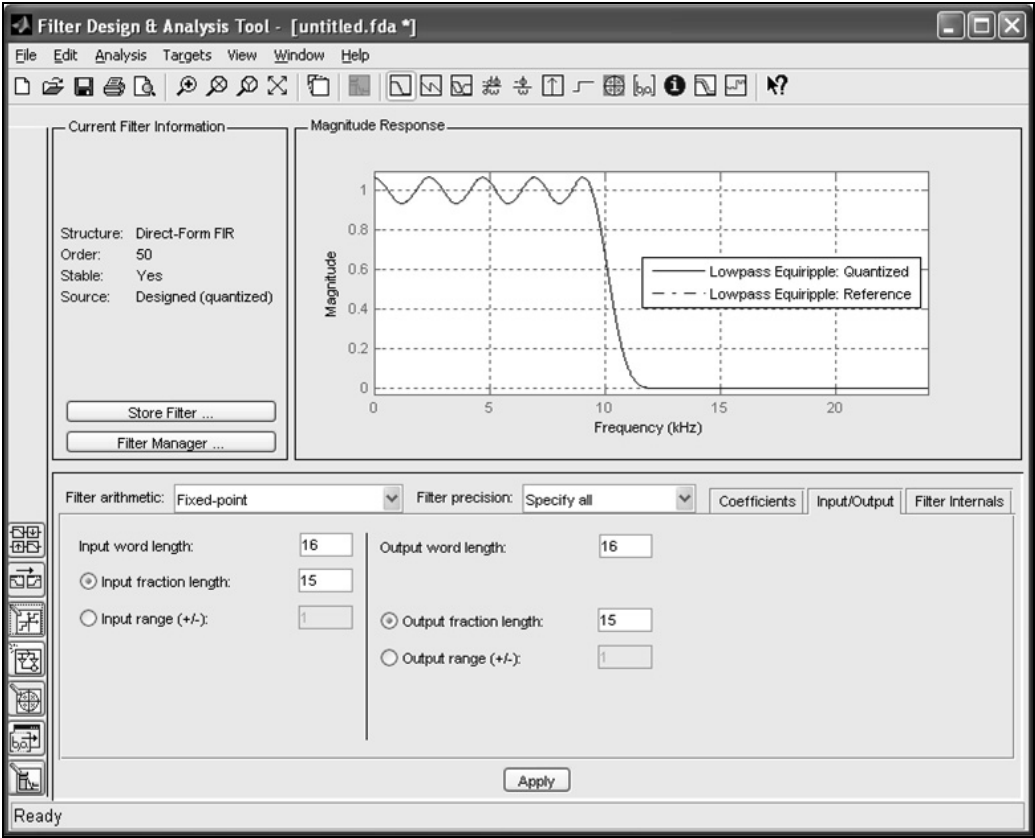


Рис. 15.4. Окно **Filter Design & Analysis Tool** при нажатой кнопке **Set quantization parameters** и открытой вкладке **Input/Output** для КИХ-фильтра с ФТ

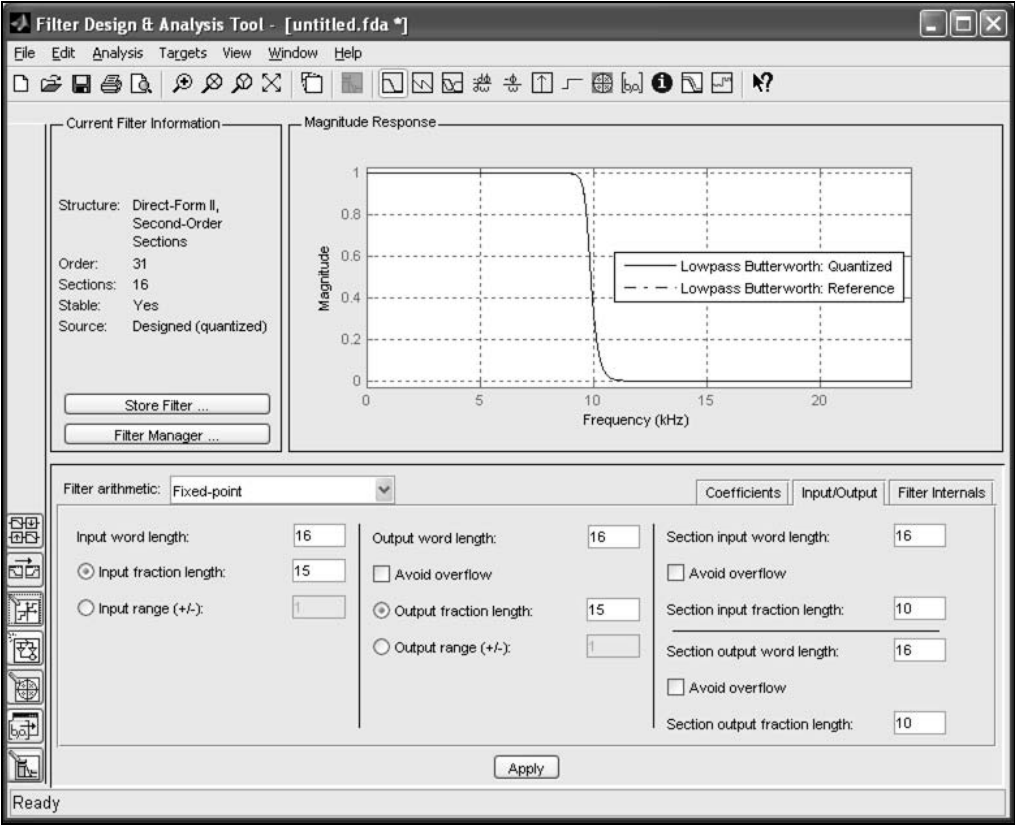


Рис. 15.5. Окно Filter Design & Analysis Tool при нажатой кнопке Set quantization parameters и открытой вкладке Input/Output для БИХ-фильтра с ФТ

Список свойств БИХ-фильтров с ФТ на вкладке Input/Output (рис. 15.5) представлен в табл. 15.4.

Таблица 15.4. Свойства БИХ-фильтров с ФТ на вкладке Input/Output

Свойство БИХ-фильтра с ФТ	Свойство объекта ddfilt
Input word length — см. табл. 15.3	InputWordLength
Input fraction length — см. табл. 15.3	InputFracLength
Input range (+/-) — аналогично свойству Numerator frac. length (см. табл. 15.1) для значений входного сигнала	—
Output word length — см. табл. 15.3	OutputWordLength
Avoid overflow (Предотвращение переполнения) — флажок управления переполнением для выходного сигнала. При сбросе флажка предотвращение переполнений достигается путем установки длины дробной части для значений выходного сигнала с помощью свойств Output fraction length и Output range (+/-).	—

Таблица 15.4 (окончание)

Свойство БИХ-фильтра с ФТ	Свойство объекта <code>dfilt</code>
При установке флажка длина дробной части устанавливается автоматически для минимизации ошибок при переполнении	
Output fraction length — см. табл. 15.3	<code>OutputFracLength</code>
Output range (+/-) — аналогично свойству Numerator frac. length (см. табл. 15.1) для выходного сигнала	—
Section input word length (Длина слова на входе звена) — длина слова для значений сигнала на входе звеньев в каскадной структуре	<code>SectionInputWordLength</code>
Avoid overflow — флажок управления переполнением для значений сигнала на входе звеньев в каскадной структуре. При сбросе флажка предотвращение переполнений достигается путем установки длины дробной части для значений сигнала на входе звеньев с помощью свойства Section input fraction length (Длина дробной части слова на входе звена). При установке флажка длина дробной части слова устанавливается автоматически для минимизации ошибок при переполнении	<code>SectionInputAutoScale</code>
Section input fraction length (Длина дробной части слова на входе звена) — длина дробной части слова для значений сигнала на входе звеньев в каскадной структуре	<code>SectionInputFracLength</code>
Section output word length (Длина слова на выходе звена) — длина слова для значений сигнала на выходе звеньев в каскадной структуре	<code>SectionOutputWordLength</code>
Avoid overflow — флажок управления переполнением для значений сигнала на выходе звеньев в каскадной структуре. При сбросе флажка предотвращение переполнений достигается путем установки длины дробной части для значений сигнала на выходе звеньев с помощью свойства Section output fraction length . При установке флажка длина дробной части устанавливается автоматически для минимизации ошибок при переполнении	<code>SectionOutputAutoScale</code>
Section output fraction length (Длина дробной части слова на выходе звена) — длина дробной части слова для значений сигнала на выходе звеньев в каскадной структуре	<code>SectionOutputFracLength</code>

15.1.3.3. Установка свойств ЦФ с ФТ на вкладке *Filter Internals*

На вкладке **Filter Internals** (рис. 15.6) устанавливаются свойства арифметических операций в ЦФ с ФТ, соответствующие свойствам объекта `dfilt` при выборе `Arithmetic: 'fixed'` [1].

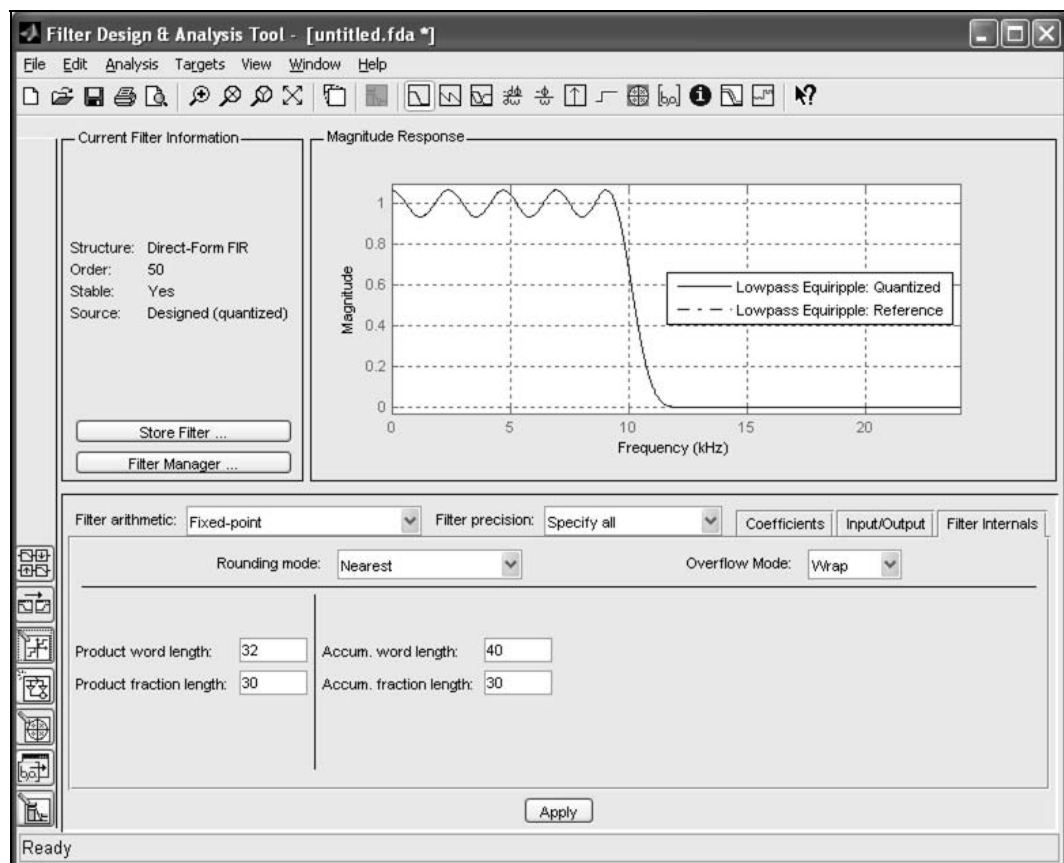


Рис. 15.6. Окно **Filter Design & Analysis Tool** при нажатой кнопке **Set quantization parameters** и открытой вкладке **Filter Internals** для КИХ-фильтра с ФТ

Для *КИХ-фильтров с ФТ* перед установкой свойств следует в раскрывающемся списке **Filter precision** выбрать режим для точности представления данных (см. разд. 15.1.3.1).

Список свойств *КИХ-фильтров с ФТ* на вкладке **Filter Internals** (см. рис. 15.6) при выборе режима `Specify all` представлен в табл. 15.5.

Список свойств *БИХ-фильтров с ФТ* на вкладке **Filter Internals** (рис. 15.7) представлен в табл. 15.6.

Таблица 15.5. Свойства КИХ-фильтров с ФТ на вкладке **Filter Internals**

Свойство КИХ-фильтра с ФТ	Свойство объекта <code>dfilt</code>
Rounding mode (Режим округления) — методы округления при представлении данных в заданном формате: • Ceiling — округление в направлении ∞: отрицательное число усекается, а для положительного берется значение ближайшего верхнего уровня квантования; • Nearest — округление до ближайшего целого: выбирается значение ближайшего уровня квантования, и если оно точно попадает на границу между соседними уровнями, то берется значение ближайшего верхнего уровня; • Nearest (convergent) — округление до ближайшего четного: выбирается значение ближайшего уровня квантования, и если оно попадает точно на границу между соседними уровнями квантования, то выбирается значение ближайшего уровня, соответствующего четному числу; • Round — округление до ближайшего целого: выбирается значение ближайшего уровня квантования, и если оно попадает точно на границу между соседними уровнями, то берется значение ближайшего верхнего уровня для положительного и нижнего — для отрицательного числа; • Zero — округление в направлении нуля: выбирается значение ближайшего нижнего уровня квантования (тождественно усечению); • Floor — округление в направлении $-\infty$: положительное число усекается, а для отрицательного берется значение ближайшего нижнего уровня квантования	RoundMode RoundMode: 'ceil' RoundMode: 'nearest' RoundMode: 'convergent' RoundMode: 'round' RoundMode: 'fix' RoundMode: 'floor'
Overflow Mode (Режим переполнения) — режимы переполнения при представлении данных в заданном формате: • Wrap — модульная арифметика: при переполнении результат автоматически заменяется значением по модулю 2 (отрицательное число, меньшее либо равное -1, суммируется с ближайшим по модулю 2^N, а из положительного, большего 1, вычитается ближайшее 2^N). Например, десятичные числа [-2 -1.5 -1 -0.5 0 0.5 1 1.5 2] будут заменены числами [0 0.5 1 -0.5 0 0.5 1 -0.5 0] • Saturate — арифметика насыщения: при переполнении результат автоматически заменяется максимально возможным (по модулю) для формата слова.	OverflowMode OverflowMode: 'wrap' OverflowMode: 'saturate'

Таблица 15.5 (окончание)

Свойство КИХ-фильтра с ФТ	Свойство объекта dfilter
Например, десятичные числа [-2 -1.5 -1 -0.5 0 0.5 1 1.5 2] будут заменены числами [-1 -1 -1 -0.5 0 0.5 1 1 1]	
Product word length (Длина слова произведения) — длина двойного слова MSB : LSB для произведения	ProductWordLength
Product fraction length (Длина дробной части произведения) — длина дробной части двойного слова MSB : LSB для произведения	ProductFracLength
Accum. word length (Длина слова для суммы) — длина расширенного слова EXT : MSB : LSB для суммы	AccumWordLength
Accum. fraction length (Длина дробной части слова для суммы) — длина дробной части расширенного слова EXT : MSB : LSB для суммы	AccumFracLength

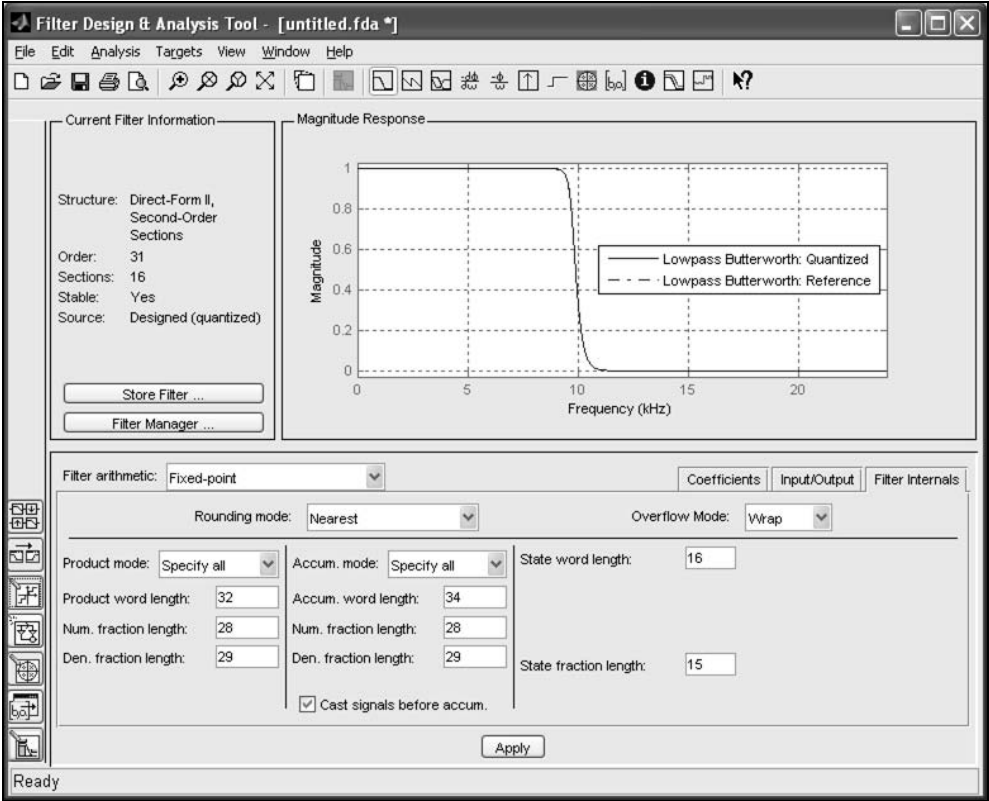


Рис. 15.7. Окно Filter Design & Analysis Tool при нажатой кнопке Set quantization parameters и открытой вкладке Filter Internals для БИХ-фильтра с ФТ

Таблица 15.6. Свойства БИХ-фильтров с ФТ на вкладке **Filter Internals**

Свойство БИХ-фильтра с ФТ	Свойство объекта <code>dfilt</code>
Rounding mode — см. табл. 15.5	RoundMode
Overflow Mode — см. табл. 15.5	
Product mode (Режим для произведения) — формат при пересылке (сохранении) результата умножения (произведения), представленного в формате двойного слова MSB : LSB: - Specify all — сохранение в произвольном формате, которое отображается свойствами Product word length и Product fraction length; - Full precision — сохранение двойного слова MSB : LSB; - Keep LSB — сохранение младшего слова LSB; - Keep MSB — сохранение старшего слова MSB	ProductMode ProductMode: 'SpecifyPrecision' ProductMode: 'FullPrecision' ProductMode: 'KeepLSB' ProductMode: 'KeepMSB'
Product word length — см. табл. 15.5	ProductWordLength
Num. fraction length — длина дробной части для локальных произведений в нерекурсивной части разностного уравнения	NumProdFracLength
Den. fraction length — длина дробной части для локальных произведений в рекурсивной части разностного уравнения	DenProdFracLength
Accum. mode — формат при пересылке (сохранении) результата сложения (суммы), представленного в формате расширенного слова EXT : MSB : LSB. Принимает такие же значения, как свойство Product mode	AccumMode
Accum. word length — см. табл. 15.5	AccumWordLength
Num. fraction length — длина дробной части для суммы в нерекурсивной части разностного уравнения	NumAccumFracLength
Den. fraction length — длина дробной части для суммы в рекурсивной части разностного уравнения	DenAccumFracLength
Cast signals before accum. (Отбрасываемые сигналы перед суммированием) — флажок управления форматом слагаемых перед суммированием. При установке флажка слагаемые представляются в формате расширенного слова, а при сбросе сохраняют формат предыдущей операции	CastBeforeSum

Таблица 15.6 (окончание)

Свойство БИХ-фильтра с ФТ	Свойство объекта <code>dfilt</code>
State word length (Длина слова состояния) — длина слова для элементов задержки (ячеек памяти) в каскадной структуре Direct-Form II, SOS (см. табл. 13.1)	<code>StateWordLength</code>
State fraction length (Длина дробной части состояния) — длина дробной части слова для элементов задержки в каскадной структуре Direct-Form II, SOS	<code>StateFracLength</code>
Num. state word length (Длина слова состояния числителя) — длина слова для элементов задержки воздействия в каскадной структуре Direct-Form I, SOS (см. табл. 13.1)	<code>NumStateWordLength</code>
Den. state word length (Длина слова состояния знаменателя) — длина слова для элементов задержки реакции в каскадной структуре Direct-Form I, SOS	<code>DenStateWordLength</code>
Num. fraction length (Длина дробной части числителя) — длина дробной части слова для элементов задержки воздействия в каскадной структуре Direct-Form I, SOS	<code>NumStateFracLength</code>
Den. fraction length (Длина дробной части знаменателя) — длина дробной части слова для элементов задержки воздействия в каскадной структуре Direct-Form I, SOS	<code>DenStateFracLength</code>

15.1.4. Моделирование структуры ЦФ с ФТ в FilterBuilder GUI

Для моделирования структуры ЦФ с ФТ в FilterBuilder GUI необходимо выполнить следующие действия:

1. На вкладке **Main** определить параметры для моделирования структуры исходного ЦФ и нажать кнопку **Apply** (Применить) (см. разд. 14.1.3).
- Структура исходного ЦФ автоматически экспортируется в Workspace в виде объекта `dfilt` со свойством `Arithmetic: 'double'` и именем, указываемым в поле **Save variable as** (Сохранить переменную как). В окне **Command Window** свойства объекта `dfilt` можно вывести по его имени¹.

¹ Экспортировать исходный ЦФ в Workspace удобно для сравнения с ЦФ с ФТ.

- На вкладке **Data Types** (Типы данных) в списке **Arithmetic** (Арифметика) выбрать **Fixed point** (Фиксированная точка) и определить параметры для структуры ЦФ с ФТ (рис. 15.8), которые описываются после п. 4.

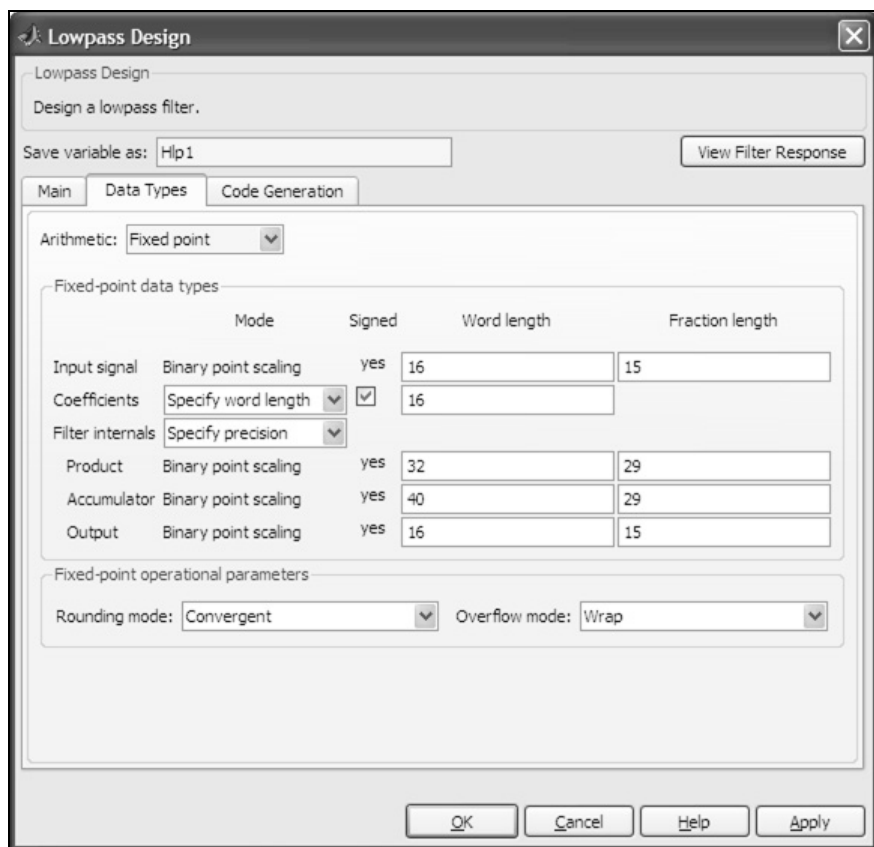


Рис. 15.8. Окно **Lowpass Design** с открытой вкладкой **Data Types** для КИХ-фильтра ФНЧ с ФТ

- Нажать кнопку **Apply**.

Структура ЦФ с ФТ автоматически экспортируется в *Workspace* в виде объекта `dfilt` со свойством `Arithmetic: 'fixed' [1]`.

- По окончании работы в *FilterBuilder GUI* нажать кнопку **OK**.

На вкладке **Data Types** группа **Fixed-point data types** (Типы данных с фиксированной точкой) содержит список параметров структуры ЦФ с ФТ, значения которых определяются одновременно в следующих четырех полях:

- ☐ **Mode** (Режим) — режим представления данных с ФТ;
- ☐ **Signed** (Знаковый) — флажок управления знаковым (по умолчанию) или беззнаковым представлением данных;

- **Word length** (Длина слова) — поле ввода для формата данных с ФТ;
- **Fraction length** (Длина дробной части) — поле ввода для дробной части формата данных с ФТ.

Список параметров для структур КИХ- и БИХ-фильтров с ФТ включает в себя:

- **Input signal** (Входной сигнал) — формат для значений входного сигнала — слово.
Значение **Mode** представлено одним режимом **Binary point scaling** (Масштабирование двоичной точки), при котором длина слова и его дробной части определяются в полях ввода **Word length** и **Fraction length**;

- **Coefficients** (Коэффициенты) — формат для коэффициентов передаточной функции — слово.

Значения **Mode** представлены двумя режимами:

- **Binary point scaling** — см. параметр **Input signal**;
- **Specify precision** (Произвольная точность) — длина слова определяется в поле ввода **Word length**, а длина его дробной части устанавливается автоматически для представления с максимальной точностью без переполнений;

- **Filter internals** (Внутреннее состояние фильтра) — точность представления результатов арифметических операций для структур КИХ-фильтров с ФТ.

Значения **Mode** представлены двумя режимами:

- **Specify precision** — формат и длина дробной части определяются в полях ввода формат слова **Word length** и **Fraction length**;
- **Full precision** (Максимальная точность) — формат и длина дробной части устанавливаются автоматически для представления с максимальной точностью без переполнений;

- **Product** (Произведение) — формат для произведения — двойное слово MSB : LSB.

Значение **Mode** для КИХ-фильтров представлено режимом **Binary point scaling** (см. параметр **Input signal**).

Значения **Mode** для БИХ-фильтров представлено режимами:

- **Full precision** — см. параметр **Filter internals**;
- **Keep LSB** — сохраняется младшее слово LSB;
- **Keep MSB** — сохраняется старшее слово MSB;
- **Specify precision** — длина двойного слова MSB : LSB определяется в поле ввода **Word length**, а его дробной части (**Fraction length**) в полях ввода **Num** и **Den** соответственно для нерекурсивной и рекурсивной частей разностного уравнения;

- **Accumulator** (Аккумулятор) — формат для суммы — расширенное слово EXT : MSB : LSB.

Значения **Mode** такие же, как для параметра **Product**.

□ **Output** (Выходной сигнал) — формат для значений выходного сигнала — слово. Значение **Mode** для КИХ-фильтров представлено режимом Binary point scaling (см. параметр **Input signal**).

Значения **Mode** для БИХ-фильтров представлено режимами:

- Avoid overflow (Предотвращение переполнения) — длина слова определяется в поле ввода **Word length**, а длина его дробной устанавливается автоматически для предотвращения переполнений;
- Best precision — длина слова определяется в поле ввода **Word length**, а длина его дробной части устанавливается автоматически для представления с максимальной точностью;
- Specify precision — см. параметр **Coefficients**;

□ **Section Input** (Сигнал на входе каскада в каскадной структуре) — формат для значений сигнала на входе каскада в каскадной структуре БИХ-фильтра — слово.

Значения **Mode** представлены режимами Binary point scaling и Specify precision (см. параметр **Coefficients**);

□ **Section Output** (Сигнал на выходе каскада в каскадной структуре) — формат для значений сигнала на выходе каскада в каскадной структуре БИХ-фильтра — слово.

Значения **Mode** представлены режимами Binary point scaling и Specify precision (см. параметр **Coefficients**);

□ **State** (Состояние) — формат для элементов задержки (ячеек памяти) в структуре БИХ-фильтра — слово.

Значение **Mode** представлено режимом Binary point scaling (см. параметр **Input signal**).

Группа **Fixed-point operational parameters** (Рабочие параметры для данных с фиксированной точкой) содержит три параметра:

□ **Rounding mode** — см. одноименное свойство в табл. 15.5;

□ **Overflow mode** — см. одноименное свойство в табл. 15.5;

□ **Cast before sum** (Отбрасывание перед суммированием) — тождественен свойству **Cast signals before accum.** в табл. 15.6.

15.2. Содержание лабораторной работы

Содержание работы связано с моделированием структур КИХ- и БИХ-фильтров с ФТ и анализом их характеристик средствами GUI FDATool.

Моделирование ЦФ с ФТ в FilterBuilder GUI вынесено в самостоятельное задание.

15.3. Задание на лабораторную работу

Исходные данные для пунктов задания приводятся в табл. 15.7—15.10 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$, и для ФНЧ, ФВЧ, ПФ и РФ включают в себя:

- требования к АЧХ КИХ-фильтров;
- требования к характеристике затухания АЧХ (дБ) (11.6) БИХ-фильтров. Значения допустимых затуханий рассчитаны по формулам (11.7)—(11.8).

На прилагаемом компакт-диске в папке `Tables\Tables_15` хранятся табл. 15.7—15.10 исходных данных и примеры их заполнения для $N_{бр} = 1$.

В работе исследуются КИХ- и БИХ-фильтры ФНЧ. Другие типы избирательности вынесены в самостоятельное задание.

Таблица 15.7. Требования к АЧХ и АЧХ (дБ) ФНЧ

Условные обозначения	Список требований	Задаваемые значения
f_d	Частота дискретизации	$f_d = 5000 + 100N_{бр} =$
f_χ	Граничная частота ПП	$f_\chi = \frac{f_d}{10} + 20N_{бр} =$
f_k	Граничная частота ПЗ	$f_k = \frac{f_d}{10} + 250 + 25N_{бр} =$
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$
$a_{\max}$	Максимально допустимое затухание в ПП (дБ)	$a_{\max} = 0,4455$
$a_{\min}$	Минимально допустимое затухание в ПЗ (дБ)	$a_{\min} = 40$

Таблица 15.8. Требования к АЧХ и АЧХ (дБ) ФВЧ

Условные обозначения	Список требований	Задаваемые значения
f_d	Частота дискретизации	$f_d = 5000 + 100N_{бр} =$
f_k	Граничная частота ПЗ	$f_\chi = \frac{f_d}{10} + 20N_{бр} =$

Таблица 15.8 (окончание)

Условные обозначения	Список требований	Задаваемые значения
f_{χ}	Граничная частота ПП	$f_k = \frac{f_d}{10} + 250 + 25N_{\text{бр}} =$
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$
a_{min}	Минимально допустимое затухание в ПЗ (дБ)	$a_{\text{min}} = 40$
a_{max}	Максимально допустимое затухание в ПП (дБ)	$a_{\text{max}} = 0,4455$

Таблица 15.9. Требования к АЧХ и АЧХ (дБ) ПФ

Условные обозначения	Список требований	Задаваемые значения
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}} =$
f_{-k}	Граничная частота ПЗ1	$f_{-k} = \frac{f_d}{20} + 20N_{\text{бр}} =$
$f_{-\chi}$	Граничная частота ПП1	$f_{-\chi} = \frac{f_d}{20} + 250 + 25N_{\text{бр}} =$
f_{χ}	Граничная частота ПП2	$f_{\chi} = \frac{f_d}{4} + 25N_{\text{бр}} =$
f_k	Граничная частота ПЗ2	$f_k = \frac{f_d}{4} + 250 + 30N_{\text{бр}} =$
δ_{21}	Максимально допустимое отклонение в ПЗ1	$\delta_{21} = 0,01$
δ_1	Максимально допустимое отклонение в ПП	$\delta_1 = 0,05$
δ_{22}	Максимально допустимое отклонение в ПЗ2	$\delta_{22} = 0,01$
$a_{1\text{min}}$	Минимально допустимое затухание в ПЗ1 (дБ)	$a_{1\text{min}} = 40$
a_{max}	Максимально допустимое затухание в ПП (дБ)	$a_{\text{max}} = 0,4455$
$a_{2\text{min}}$	Минимально допустимое затухание в ПЗ2 (дБ)	$a_{2\text{min}} = 40$

Таблица 15.10. Требования к АЧХ и АЧХ (дБ) РФ

Условные обозначения	Список требований	Задаваемые значения
f_d	Частота дискретизации	$f_d = 5000 + 100N_{\text{бр}} =$
$f_{-\chi}$	Граничная частота ПП1	$f_{-\chi} = \frac{f_d}{20} + 20N_{\text{бр}} =$
f_{-k}	Граничная частота ПЗ1	$f_{-k} = \frac{f_d}{20} + 250 + 25N_{\text{бр}} =$
f_k	Граничная частота ПЗ2	$f_k = \frac{f_d}{4} + 25N_{\text{бр}} =$
f_{χ}	Граничная частота ПП2	$f_{\chi} = \frac{f_d}{4} + 250 + 30N_{\text{бр}} =$
δ_{11}	Максимально допустимое отклонение в ПП1	$\delta_{11} = 0,05$
δ_2	Максимально допустимое отклонение в ПЗ	$\delta_2 = 0,01$
δ_{12}	Максимально допустимое отклонение в ПП2	$\delta_{12} = 0,05$
$a_{1\text{max}}$	Максимально допустимое затухание в ПП1 (дБ)	$a_{1\text{max}} = 0,4455$
a_{min}	Минимально допустимое затухание в ПЗ (дБ)	$a_{\text{min}} = 40$
$a_{2\text{max}}$	Максимально допустимое затухание в ПП2 (дБ)	$a_{2\text{max}} = 0,4455$

Задание на лабораторную работу заключается в синтезе и анализе КИХ- и БИХ-фильтров ФНЧ с ФТ в GUI FDATool и включает в себя выполнение следующих пунктов:

1. Моделирование структуры исходного КИХ-фильтра ФНЧ¹.
- Синтезировать исходный КИХ-фильтр ФНЧ методом чебышевской аппроксимации и выбрать структуру Direct-Form Symmetric FIR (см. п. 3 в разд. 14.3).
- Сохранить исходный КИХ-фильтр в буфере **Filter Manager** с именем FIR_lowpass_Equiripple.

¹ В работе по синтезу ЦФ была сохранена сессия FDATool с именем Filters.fda (см. разд. 14.5), где в буфере **Filter Manager** содержатся КИХ- и БИХ-фильтры, используемые в качестве исходных в данной работе с теми же именами. Поэтому достаточно открыть сессию (см. п. 11 в разд. 14.3), сохранить ее под новым именем Filters_Fixed_Point.fda и продолжить работу, начиная с пункта 4.

2. Моделирование структуры исходного БИХ-фильтра ФНЧ Золотарева—Кауэра.

Синтезировать исходный БИХ-фильтр ФНЧ Золотарева—Кауэра и выбрать структуру Direct-Form II, Second-order sections (см. п. 5 в разд. 14.3).

Выполнить расстановку звеньев и масштабирование в каскадной структуре БИХ-фильтра (см. п. 7 в разд. 14.3).

Сохранить исходный БИХ-фильтр в буфере **Filter Manager** с именем IIR_lowpass_Elliptic_Scale.

3. Экспорт исходных ЦФ в Workspace.

Экспортировать в Workspace исходные КИХ- и БИХ-фильтры FIR_lowpass_Equiripple и IIR_lowpass_Elliptic_Scale в виде объектов dfilt с теми же именами (см. пп. 5—7 в разд. 14.3).

Обратите внимание! Экспорт исходных КИХ- и БИХ-фильтров должен быть выполнен до моделирования их структуры с ФТ!

4. Моделирование структуры КИХ-фильтра с ФТ.

Загрузить исходный КИХ-фильтр FIR_lowpass_Equiripple из буфера **Filter Manager** и выполнить моделирование его структуры с ФТ (см. разд. 15.1.3).

Перед установкой свойств на вкладках **Coefficients**, **Input/Output** и **Filter Internals** проверить значения коэффициентов передаточной функции КИХ-фильтра.

Особое внимание обратить на значения, превышающие единицу, т. к. от этого будет зависеть установка свойств!

Для коэффициентов передаточной функции, входного и выходного сигналов выбрать длину слова 8 битов и дробной части 7 битов (при отсутствии значений, больших единицы по модулю).

В зависимости от этого установить длины двойного слова для произведения, расширенного слова для суммы и т. д.

Если хотя бы один из коэффициентов *превосходит единицу*, отвести один бит на целую часть (оставить 6 битов для дробной части).

Сохранить в буфере **Filter Manager** КИХ-фильтр с ФТ с именем FIR_lowpass_Equiripple_FP.

5. Сравнительный анализ характеристик КИХ-фильтра с ФТ и исходного КИХ-фильтра.

Обращаясь к командам меню **Analysis**, вывести АЧХ, ФЧХ и ИХ и коэффициенты передаточной функции КИХ-фильтров FIR_lowpass_Equiripple и FIR_lowpass_Equiripple_FP.

На графиках АЧХ, ФЧХ и ИХ вывести легенду по команде меню **View | Legend**.

Пояснить:

- выполняются ли требования к АЧХ КИХ-фильтра с ФТ;
- как изменились коэффициенты передаточной функции;

- сохраняются ли свойства симметрии ИХ и линейности ФЧХ КИХ-фильтра с ФТ.
6. Экспорт КИХ-фильтра с ФТ в Workspace.
- Экспортировать КИХ-фильтр `FIR_lowpass_Equiripple_FP` в виде объекта `dfilt` с тем же именем.
7. Анализ свойств экспортированных объектов `dfilt`.
- Вывести свойства объектов `FIR_lowpass_Equiripple` и `FIR_lowpass_Equiripple_FP` в окне **Command Window**, обращаясь к каждому из них по имени.
- Привести результаты сравнения свойств данных объектов.
- Обращаясь к команде меню **Analysis | Filter Information** (Анализ | Информация о фильтре), сравнить выведенную информацию о КИХ-фильтрах со свойствами объектов `dfilt`.
8. Моделирование структуры БИХ-фильтра с ФТ.
- Загрузить исходный БИХ-фильтр `IIR_lowpass_Elliptic_Scale` из буфера **Filter Manager** и выполнить моделирование его структуры с ФТ (см. разд. 15.1.3).
- Перед установкой свойств на вкладках **Coefficients**, **Input/Output** и **Filter Internals** проверить значения коэффициентов передаточных функций биквадратных звеньев и коэффициентов усиления.*
- Особое внимание обратить на значения, превышающие единицу, т. к. от этого будет зависеть установка свойств!*
- Для коэффициентов передаточной функции, входного и выходного сигналов выбрать длину слова 8 битов и дробной части 7 битов (при отсутствии значений, больших единицы по модулю).
- В зависимости от этого установить длины двойного слова для произведения, расширенного слова для суммы и т. д.
- Если хотя бы один из коэффициентов a_1 в знаменателе передаточной функции звена *превосходит единицу*, для коэффициентов знаменателей отвести один бит на целую часть (оставить 6 битов для дробной части).
- Сохранить в буфере **Filter Manager** БИХ-фильтр с ФТ с именем `IIR_lowpass_Elliptic_Scale_FP`.
9. Сравнительный анализ характеристик БИХ-фильтра с ФТ и исходного БИХ-фильтра.
- Обращаясь к командам меню **Analysis**, вывести АЧХ и карту нулей и полюсов БИХ-фильтров `IIR_lowpass_Elliptic_Scale` и `IIR_lowpass_Elliptic_Scale_FP`.
- На графиках вывести легенду по команде меню **View | Legend**.
- Пояснить:
- выполняются ли требования к АЧХ БИХ-фильтра с ФТ;
 - является ли БИХ-фильтр с ФТ устойчивым.

10. Экспорт БИХ-фильтра с ФТ в Workspace.

Экспортировать БИХ-фильтр `IIR_lowpass_Elliptic_Scale_FP` в виде объекта `dfilt` с тем же именем.

11. Анализ свойств экспортированных объектов `dfilt`.

Вывести свойства объектов `IIR_lowpass_Elliptic_Scale` и `IIR_lowpass_Elliptic_Scale_FP` в окне **Command Window**, обращаясь к каждому из них по имени.

Пояснить результаты сравнения свойств данных объектов.

Обращаясь к команде меню **Analysis | Filter Information**, сравнить выведенную информацию о БИХ-фильтрах со свойствами объектов `dfilt`.

12. Сохранение сессии `FDATool`.

Сохранить сессию с именем `Filters_Fixed_Point.fda`.

15.4. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в повторении пунктов задания для КИХ- и БИХ-фильтров ФВЧ, ПФ и РФ с исходными данными для своего номера бригады $N_{бр}$.

Пункты задания включает в себя:

1С. Моделирование структур КИХ-фильтров, сохраненных в сессии `FDATool` с именем `Filters.fda` (см. п. 11 в *разд. 14.3*).

При установке свойств выбрать длину слова 4, 8 и 16 битов и сравнить результаты.

2С. Моделирование структур БИХ-фильтров, сохраненных в сессии `FDATool` с именем `Filters.fda` (см. п. 11 в *разд. 14.3*).

При установке свойств выбрать длину слова 4, 8 и 16 битов и сравнить результаты.

3С. Моделирование структур КИХ- и БИХ-фильтров с ФТ в `FilterBuilder GUI` (см. *разд. 15.1.4*).

На вкладке **Data Types** в списке **Arithmetic** выбрать тип данных `Fixed point` и задать параметры самостоятельно.

Исследовать свойства экспортированного в `Workspace` объекта `dfilt`.

15.5. Отчет и контрольные вопросы

Отчет составляется в редакторе `MS Word` и содержит исходные данные и результаты выполнения пунктов задания, включая свойства объектов `dfilt`, копируемые из окна **Command Window** (шрифт `Courier New`), необходимые графики (копируются при нажатии комбинации клавиш `<Alt>+<Print Screen>`) и ответы на поставленные вопросы (шрифт `Times New Roman`).

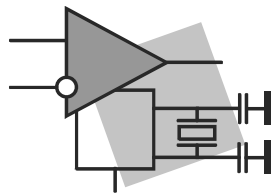
Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. Поясните форматы данных с ФТ: слово, двойное слово, расширенное слово.
2. Назовите источники ошибок квантования в цифровой системе с ФТ.
3. Поясните основные этапы моделирования структуры ЦФ с ФТ в GUI FDATool.
4. Поясните, с какой целью анализируется АЧХ ЦФ с ФТ.
5. Поясните, какие свойства устанавливаются на вкладке **Coefficients**.
6. Поясните, какие свойства устанавливаются на вкладке **Input/Output**.
7. Поясните, какие свойства устанавливаются на вкладке **Filter Internals**.

15.6. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Главы 16, 20.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 7.

ГЛАВА 16



Спектральный анализ: непараметрические методы

Цель работы: изучить методы непараметрического спектрального анализа (спектрального оценивания) случайных последовательностей и овладеть программными средствами их реализации в MATLAB.

16.1. Краткая теоретическая справка

Спектральный анализ предназначен для оценки частотного состава случайного дискретного сигнала.

Методы спектрального анализа разделяют на две группы:

- ☐ непараметрические;
- ☐ параметрические.

Непараметрические методы основаны на вычислении оценок СПМ непосредственно по отсчетам исходной последовательности, что определяет их основные достоинства — применимость к широкому классу стационарных сигналов и шумов и высокую вычислительную эффективность за счет применения алгоритмов БПФ.

По умолчанию будем подразумевать *эргодические* случайные дискретные сигналы — *случайные последовательности* (см. разд. 7.1.2).

В зависимости от способа обработки исходной последовательности различают следующие непараметрические методы оценивания СПМ:

- ☐ метод периодограмм (periodogram);
- ☐ методы модифицированных периодограмм (modified periodogram), в том числе:
 - метод периодограмм Даниелла (Daniell periodogram);
 - метод периодограмм Бартлетта (Bartlett periodogram);
 - метод периодограмм Уэлча (Welch periodogram);
- ☐ метод Блэкмана—Тьюки¹ (Blackman-Tukey).

¹ Называемый также коррелограммным методом [4].

Ранее (см. разд. 7.1.2) были рассмотрены статистические характеристики случайного дискретного сигнала во *временной области*. Для спектрального анализа случайного дискретного сигнала используется его статистическая характеристика в *частотной области*, называемая *спектральной плотностью мощности*¹ СПМ (PSD²).

Спектральную плотность мощности $S(\omega)$ можно определить как усредненный квадрат спектральной плотности случайной последовательности теоретически бесконечной длины ($N \rightarrow \infty$):

$$S(\omega) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \frac{|X(e^{j\omega T})|^2}{f_d}, \quad (16.1)$$

где $X(e^{j\omega T})$ — спектральная плотность последовательности $x(n)$:

$$X(e^{j\omega T}) = \sum_{n=-N}^N x(n)e^{-j\omega Tn};$$

f_d — частота дискретизации.

Множитель $1/f_d = T$ в (16.1) учитывает связь между СПМ дискретного и аналогового сигналов, подобно связи между спектральными плотностями данных сигналов.

Эквивалентным определением СПМ как Фурье-изображения АКФ случайной последовательности воспользуемся при описании метода Блэкмана—Тьюки в разд. 16.1.6.

При *конечной* длине N последовательности говорят о вычислении *оценки* СПМ $\hat{S}(\omega)$.

Подобно оценке СПМ случайного аналогового сигнала, оценка СПМ случайной последовательности дает *картину распределения средней мощности по частоте*, измеряется в ваттах на герц (Вт/Гц) или децибелах на герц (дБ/Гц) и не содержит информации о фазах спектральных составляющих, которые собственно и определяют *случайный характер спектральной плотности случайного сигнала*, в то время как ее усредненный квадрат стремится к некоторому пределу.

16.1.1. Метод периодограмм

Метод периодограмм заключается в вычислении *оценки* СПМ $\hat{S}(\omega)$ конечной случайной последовательности длины N , которая называется *периодограммой*³:

¹ Ее также называют энергетическим спектром.

² Power Spectral Density — спектральная плотность мощности.

³ Немодифицированной.

$$\hat{S}(\omega) = \frac{|X(e^{j\omega T})|^2}{Nf_d}, \quad (16.2)$$

где $X(e^{j\omega T})$ — спектральная плотность конечной последовательности $x(n)$:

$$X(e^{j\omega T}) = \sum_{n=0}^{N-1} x(n)e^{-j\omega Tn}.$$

Периодограмма $\hat{S}(\omega)$ является неотрицательной четной периодической функцией частоты с периодом $\omega_d = 2\pi f_d = 2\pi/T$, для вещественных последовательностей $x(n)$ — симметричной относительно частоты $\omega_d/2$ в основной полосе частот $[0; \omega_d]$.

Для уменьшения эффекта растекания спектра при вычислении периодограммы с помощью ДПФ (см. разд. 10.1.1) и, как следствие, сглаживания периодограммы применяют *весовые функции* (окна), и видоизмененная периодограмма (16.2), называемая *модифицированной периодограммой* $\hat{S}_w(\omega)$, принимает вид:

$$\hat{S}_w(\omega) = \frac{\frac{1}{f_d} |X_w(e^{j\omega T})|^2}{\sum_{n=0}^{N-1} |w(n)|^2}, \quad (16.3)$$

где $w(n)$ — весовая функция (окно) длины N ; $X_w(e^{j\omega T})$ — спектральная плотность произведения $x(n)w(n)$:

$$X_w(e^{j\omega T}) = \sum_{n=0}^{N-1} x(n)w(n)e^{-j\omega Tn}.$$

В MATLAB в шкале частот f (Гц) периодограммы (16.2) и (16.3) рассчитываются на основе ДПФ с привлечением алгоритмов БПФ с помощью функции:

[S,f] = periodogram(x,window(N,[winparameter]),Nfft,Fs['range'])

где x — исходная последовательность $x(n)$ длины N (`length(x)`); `window(N,winparameter)` — имя окна длины $N = \text{length}(x)$ с параметром `winparameter` (если он имеется); например, `hamming(length(x))` — окно Хэмминга, `kaiser(length(x),0.5)` — окно Кайзера с параметром $\beta = 0,5$; пустая матрица `[]` указывается для периодограммы (16.2) с прямоугольным окном (по умолчанию).

С окнами и их параметрами можно познакомиться, обращаясь к GUI Window Design and Analysis Tool (Средство создания и анализа окон) по команде:

`wintool`

N_{fft} — размерность ДПФ, определяющая длины векторов f и s на периоде и период дискретизации по частоте.

При указании пустой матрицы $[]$ по умолчанию $N_{fft} = \max([256, 2^{\text{nextpow2}(N)}])$, где $2^{\text{nextpow2}(N)}$ — ближайшая степень двойки в сторону увеличения.

При несовпадении размерности ДПФ и длины исходной последовательности последняя автоматически дополняется нулями или усекается до длины N_{fft} .

F_s — частота дискретизации f_d (Гц).

Разрешение по частоте (см. разд. 9.1.3) будет равно $F_s/\text{length}(x)$ при $N_{fft} \geq \text{length}(x)$ и F_s/N_{fft} при $N_{fft} < \text{length}(x)$.

'range' — символьная переменная, равная 'twosided' для периодограммы на периоде $[0; F_s]$ или 'onesided' — в основной полосе частот $[0; F_s/2]$; по умолчанию в отсутствии параметра 'range' для вещественных последовательностей выбирается 'onesided', а для комплексных — 'twosided'.

f , s — векторы значений частот (Гц) и периодограммы (16.2) или (16.3) (Вт/Гц).

График периодограммы $\hat{S}(f)$ (Вт/Гц) выводится с помощью функции:

`plot(f, s)`

а периодограммы $\hat{S}(f)$ (дБ/Гц):

$$\hat{S}(f) \text{ (дБ/Гц)} = 10 \lg \hat{S}(f) \quad (16.4)$$

с помощью функции:

`periodogram(x, window, Nfft, Fs, 'range')`

Периодограммы (16.2) и (16.3) могут описываться в виде объектов `spectrum`:

`Hs = spectrum.periodogram`

`Hs = spectrum.periodogram({'window', winparameter})`

где H_s — имя объекта.

Смысл параметра в фигурных скобках такой же, как и в функции `periodogram`.

Список свойств объекта H_s включает в себя:

□ `FFTLengh` — размерность ДПФ;

□ `WindowName` — используемое окно.

После создания объекта `spectrum` выводится график периодограммы $\hat{S}(f)$ (дБ/Гц) с помощью функции:

`psd(Hs, x, 'Fs', Fs)`

16.1.2. Основные показатели качества оценок СПМ

Для сравнения различных оценок СПМ воспользуемся следующими основными показателями качества.

□ Несмещенность/смещенность оценки СПМ.

Согласно определению [4, 5], *смещением* β некоторого параметра α называют разность между истинным значением данного параметра и математическим ожиданием его оценки $\hat{\alpha}$:

$$\beta = \alpha - M\{\hat{\alpha}\},$$

где $M\{\cdot\}$ — оператор математического ожидания.

С учетом равенства $\alpha = M\{\alpha\}$ имеем:

$$\beta = M\{\alpha\} - M\{\hat{\alpha}\} = M\{\alpha - \hat{\alpha}\}. \quad (16.5)$$

Оценку параметра $\hat{\alpha}$ называют *несмещенной*, если при усреднении по ансамблю с возрастанием числа реализаций смещение β стремится к нулю.

Для эргодического случайного дискретного сигнала — *случайной последовательности* длины N — *смещение* β оценки СПМ $\hat{S}(\omega)$ равно математическому ожиданию разности между истинной СПМ $S(\omega)$ и ее оценкой $\hat{S}(\omega)$:

$$\beta = M\{S(\omega) - \hat{S}(\omega)\}. \quad (16.6)$$

Оценку СПМ называют *несмещенной*, если смещение $\beta = 0$, и *асимптотически несмещенной*, если при длине $N \rightarrow \infty$ смещение β (скаляр) стремится к нулю:

$$\beta = M\{S(\omega) - \hat{S}(\omega)\} \rightarrow 0.$$

□ Состоятельность/несостоятельность оценки СПМ.

Согласно определению [4, 5], оценку параметра $\hat{\alpha}$ называют *состоятельной*, если при усреднении по ансамблю с возрастанием числа реализаций математическое ожидание квадрата отклонения истинного значения параметра α от его оценки $\hat{\alpha}$ стремится к нулю:

$$M\{(\alpha - \hat{\alpha})^2\} = D\{\hat{\alpha}\} + \beta^2 \rightarrow 0.$$

Для *случайной последовательности* длины N оценку СПМ называют *состоятельной*, если с увеличением длины последовательности математическое ожидание квадрата отклонения истинной СПМ $S(\omega)$ от ее оценки $\hat{S}(\omega)$ (скаляр) стремится к нулю:

$$M\left\{\left[S(\omega) - \hat{S}(\omega)\right]^2\right\} = D\{\hat{S}(\omega)\} + \beta^2 \rightarrow 0, \quad (16.7)$$

где $D\{\cdot\}$ — оператор дисперсии.

Убедимся в равенстве правой и левой частей (16.7), используя обозначения: $\alpha = S(\omega)$ и $\hat{\alpha} = \hat{S}(\omega)$.

Согласно свойству линейности оператора $M\{\cdot\}$, слева в (16.7) имеем:

$$M\{(\alpha - \hat{\alpha})^2\} = M\{\alpha^2\} - 2M\{\alpha\}M\{\hat{\alpha}\} + M\{\hat{\alpha}^2\}.$$

Согласно свойству оператора $D\{\cdot\}$ и (16.5), справа в (16.7) имеем то же самое:

$$D\{\hat{\alpha}\} + M\{(\alpha - \hat{\alpha})^2\} = M\{\hat{\alpha}^2\} - \underline{M^2\{\hat{\alpha}\}} + M\{\alpha^2\} - 2M\{\alpha\}M\{\hat{\alpha}\} + \underline{M^2\{\hat{\alpha}\}}.$$

Из (16.7) следует, что *смещенная* оценка СПМ свидетельствует о ее *несостоятельности*, а при *несмещенной* оценке СПМ ее *состоятельность* определяется стремлением *дисперсии* к нулю с увеличением длины последовательности.

Следствием *несостоятельности* оценки СПМ является эффект *изрезанности* периодограммы, усиливающийся с *увеличением* длины последовательности.

Признаком улучшения качества оценки СПМ является ослабление данного эффекта с увеличением длины последовательности.

Анализ несмещенности/смещенности и состоятельности/несостоятельности различных оценок СПМ выходит за рамки данной книги, поэтому далее будут приведены лишь известные результаты.

Оценки СПМ (16.2) и (16.3) являются *асимптотически несмещенными*, но при этом *несостоятельными*, вследствие чего периодограммы оказываются сильно осциллирующими (изрезанными) и с увеличением длины последовательности этот эффект усиливается. Для *сглаживания* периодограммы применяют то или иное ее *усреднение*, реализованное в периодограммах Даньелла, Бартлетта, Уэлча и в методе Блэкмана—Тьюки.

- Добротность оценки СПМ — отношение квадрата ее математического ожидания (среднего значения) $\hat{S}_{\text{cp}}$ к дисперсии $\sigma_{\hat{S}}^2$:

$$Q = \hat{S}_{\text{cp}}^2 / \sigma_{\hat{S}}^2. \quad (16.8)$$

Несмещенная оценка дисперсии $\sigma_{\hat{S}}^2$ вычисляется по формуле:

$$\sigma_{\hat{S}}^2 = \frac{1}{N-1} \sum_{k=0}^{N-1} \left(\hat{S}(\omega_k) - \hat{S}_{\text{cp}} \right)^2, \quad (16.9)$$

а смещенная — по формуле:

$$\sigma_{\hat{S}}^2 = \frac{1}{N} \sum_{k=0}^{N-1} \left(\hat{S}(\omega_k) - \hat{S}_{\text{cp}} \right)^2,$$

где ω_k — значения частот в N равноотстоящих точках на периоде СПМ; $\hat{S}_{\text{cp}}$ — среднее значение оценки СПМ:

$$\hat{S}_{\text{cp}} = \frac{1}{N} \sum_{k=0}^{N-1} \hat{S}(\omega_k).$$

Добротность характеризует отношение квадрата постоянной составляющей оценки СПМ к среднему квадрату амплитуд осцилляций, поэтому *чем больше добротность, тем выше качество оценки СПМ*.

В MATLAB несмещенная и смещенная оценки дисперсии вычисляются соответственно с помощью функций (см. табл. 2.2):

`var (x)`

`var (x, 1)`

где x — исходная последовательность длины N .

- Среднеквадратическое (стандартное) отклонение СКО (STD¹) оценки СПМ.

Несмещенная оценка СКО $\sigma_{\hat{S}}$ СПМ $\hat{S}(\omega)$ вычисляется по формуле:

$$\sigma_{\hat{S}} = \sqrt{\frac{1}{N-1} \sum_{k=0}^{N-1} \left(\hat{S}(\omega_k) - \hat{S}_{\text{cp}} \right)^2}, \quad (16.10)$$

а смещенная — по формуле:

$$\sigma_{\hat{S}} = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} \left(\hat{S}(\omega_k) - \hat{S}_{\text{cp}} \right)^2}.$$

СКО оценки СПМ совпадает с размерностью оценки СПМ и характеризует интенсивность (усредненную амплитуду) ее осцилляций, поэтому *чем меньше СКО, тем выше качество оценки СПМ*.

В MATLAB несмещенная и смещенная оценки СКО вычисляются соответственно с помощью функций (см. табл. 2.2):

`std (x)`

`std (x, 1)`

где x — исходная последовательность длины N .

Отметим, что для аддитивной смеси сигнала с шумом информативность периодограммы (16.2) — возможность определения частот гармоник — зависит от СКО шума. Однако в общем случае при заранее неизвестном случайном сигнале существенная изрезанность периодограммы снижает ее информативность, т. к. вследствие несостоятельности она дает искаженную картину распределения средней мощности по частоте.

16.1.3. Метод периодограмм Даньелла

В методе периодограмм Даньелла сглаживание периодограммы (16.2) достигается за счет ее усреднения по соседним $(2K+1)$ частотам, симметрично расположенным относительно текущей частоты ω_k : $[\omega_{k-K} \dots \omega_{k-1} \omega_k \omega_{k+1} \dots \omega_{k+K}]$.

¹ Standard Deviation — стандартное отклонение.

Периодограмма Даньелла вычисляется на основе исходной периодограммы $\hat{S}(\omega)$ (16.2) по формуле:

$$\hat{S}_{DANIELL}(\omega_i) = \frac{1}{2K+1} \sum_{k=i-K}^{i+K} \hat{S}(\omega_k), \quad i = K, K+1, \dots, N-1-K, \quad (16.11)$$

где N — длина (количество отсчетов) исходной периодограммы $\hat{S}(\omega)$ (16.2), совпадающая с длиной исходной последовательности; ω_k — значения частот для усреднения периодограммы (16.2); их количество для каждого значения i равно $(2K+1)$.

Согласно (16.11), периодограмма Даньелла имеет длину $(N-2K)$. Для вычисления периодограммы Даньелла длины N следует исходную периодограмму $\hat{S}(\omega)$ (16.2) дополнить K отсчетами в начале и в конце с учетом ее периодичности.

Оценка СПМ (16.11) является *асимптотически несмещенной*. За счет операции усреднения по соседним частотам периодограмма Даньелла будет менее осциллирующей (изрезанной), чем периодограмма (16.2).

В MATLAB для вычисления периодограммы Даньелла можно воспользоваться функцией `smooth` (вычисление скользящего среднего).

Сначала рассчитывается вектор s отсчетов исходной периодограммы $\hat{S}(\omega)$ (16.2) длины N с помощью функции `periodogram` и формируется новый вектор s длины $(N+2K)$, дополненный K отсчетами в начале и в конце с учетом периодичности периодограммы:

```
s = [S(N-K+1:N); S; S(1:K)];
```

Затем с помощью функции `smooth` рассчитывается периодограмма Даньелла:

```
SD1 = smooth(S,M)
```

```
SD = SD1(K+1:K+N)
```

где M — количество усредняемых соседних частот $M = 2K+1$; $SD1$ — результат вычисления скользящего среднего — вектор длины $(N+2K)$; SD — вектор отсчетов периодограммы Даньелла длины N (после усечения K первых и последних элементов вектора $SD1$).

Другой способ вычисления периодограммы Даньелла заключается в определении реакции КИХ-фильтра длины $M = 2K+1$ с одинаковыми коэффициентами $b_i = 1/M$, $i = 0, \dots, M-1$, с помощью функции `filter`.

Сначала формируется вектор b коэффициентов КИХ-фильтра с помощью функции `ones` (см. табл. 2.1):

```
b = (1/M)*ones(1,M);
```

Затем с помощью функции `filter` рассчитывается периодограмма Даньелла:

```
SD1 = filter(b,1,S)
```

```
SD = SD1(2*K+1:2*K+N)
```


где 1 — значение коэффициента $a_1 = 1$ для КИХ-фильтра; s — вектор отсчетов исходной периодограммы $\hat{S}(\omega)$ (16.2) длины $(N + 2K)$, дополненной K отсчетами в начале и в конце с учетом периодичности; формирование вектора s приведено выше для функции `smooth`; `SD1` — реакция КИХ-фильтра длины $(N + 2K)$; `SD` — вектор отсчетов периодограммы Даньелла длины N (усечение первых $2K$ отсчетов реакции, рассчитываемых с учетом ННУ).

16.1.4. Метод периодограмм Бартлетта

В методе периодограмм Бартлетта сглаживание периодограммы $\hat{S}(\omega)$ (16.2) достигается за счет разбиения исходной последовательности длины N на *неперекрывающиеся* фрагменты $x^{(p)}(n)$ длины L :

$$x^{(p)}(n) = x(n + pL), \quad n = 0, \dots, L-1; \quad p = 0, \dots, P-1, \quad (16.12)$$

где p — номер фрагмента, $P = N/L$ — их количество¹, и вычисления периодограмм $\hat{S}^{(p)}(\omega)$ (16.2) *фрагментов*:

$$\hat{S}_{BARTLETT}^{(p)}(\omega) = \hat{S}^{(p)}(\omega), \quad p = 0, \dots, P-1. \quad (16.13)$$

Периодограмма Бартлетта исходной последовательности определяется как усредненная сумма периодограмм фрагментов (16.13):

$$\hat{S}_{BARTLETT}(\omega) = \frac{1}{P} \sum_{p=0}^{P-1} \hat{S}_{BARTLETT}^{(p)}(\omega). \quad (16.14)$$

Оценка СПМ (16.14) является *асимптотически несмещенной* и при этом *состоятельной* за счет деления последовательности на *неперекрывающиеся* фрагменты и усреднения периодограмм фрагментов.

Если АКФ $R(m)$ фрагментов затухает до пренебрежимо малых величин, т. е.

$$R(m) \approx 0 \quad \text{при } |m| > L,$$

то периодограммы фрагментов можно считать независимыми. Доказано [4], что в этом случае дисперсия периодограммы Бартлетта будет обратно пропорциональна количеству фрагментов P , и, как следствие, периодограмма Бартлетта будет более сглаженной (менее осциллирующей), чем периодограмма Даньелла (16.11). По этой причине количество фрагментов P стремятся увеличивать. С другой стороны, это приводит к уменьшению длины фрагмента L , а следовательно, уменьшению разрешения по частоте — увеличению $\Delta f = f_d/L$. Поэтому количество фрагментов P выбирается из соображений компромисса между уменьшением дисперсии оценки СПМ и приемлемым разрешением по частоте.

¹ При необходимости последний фрагмент дополняется нулями до длины L , с тем чтобы количество фрагментов P было целым числом.

Периодограмму Бартлетта можно рассматривать как частный случай периодограммы Уэлча и для ее вычисления в MATLAB использовать функцию `pwelch` (см. разд. 16.1.5).

Прямой способ вычисления периодограммы Бартлетта длины N заключается в вычислении периодограмм длины N неперекрывающихся фрагментов длины L и усреднении периодограмм фрагментов.

16.1.5. Метод периодограмм Уэлча

В методе периодограмм Уэлча сглаживание периодограммы (16.2) достигается за счет разбиения исходной последовательности длины N на *перекрывающиеся* фрагменты $x^{(p)}(n)w(n)$ длины L со сглаживающим окном $w(n)$ и величиной перекрытия Q , где $Q < L$:

$$x^{(p)}(n)w(n) = x(n + pQ)w(n), \quad n = 0, \dots, L-1; \quad p = 0, \dots, P-1, \quad (16.15)$$

где p — номер фрагмента, P — их количество на длине N исходного сигнала, равное:

$$P = \frac{N-L}{Q} + 1, \quad (16.16)$$

и вычисления периодограмм $\hat{S}_w^{(p)}(\omega)$ (16.3) перекрывающихся фрагментов:

$$\hat{S}_{WELCH}^{(p)}(\omega) = \hat{S}_w^{(p)}(\omega), \quad p = 0, \dots, P-1. \quad (16.17)$$

Периодограмма Уэлча исходной последовательности определяется как усредненная сумма периодограмм фрагментов (16.17):

$$\hat{S}_{WELCH}(\omega) = \frac{1}{P} \sum_{p=0}^{P-1} \hat{S}_{WELCH}^{(p)}(\omega). \quad (16.18)$$

Оценка СПМ (16.18) является асимптотически несмещенной и при этом *состоятельной* за счет деления последовательности на *перекрывающиеся* фрагменты и усреднения периодограмм фрагментов. Качество оценки, по сравнению с периодограммой Бартлетта (16.14), повышается за счет увеличения общего количества фрагментов при их перекрытии. Как следствие, периодограмма Уэлча будет менее осциллирующей (изрезанной), чем периодограмма Бартлетта.

В MATLAB в шкале частот f (Гц) периодограмма Уэлча (16.18) рассчитывается с помощью функции:

`[SW,f] = pwelch(x,window(L,[winparameter]),noverlap,Lfft,Fs,['range'])`

где `window(L,winparameter)` подобен одноименному параметру функции `periodogram` (см. разд. 16.1.1) для фрагмента длины L ; по умолчанию используется окно Хэмминга, для которого достаточно указать длину L ; `noverlap` — величина перекрытия Q ; при указании пустой матрицы `[]` по умолчанию равно $Q = L/2$;

Lfft подобен параметру Nfft функции `periodogram` для фрагмента длины L и определяет длины векторов f и SW .

Параметры F_s и `'range'` определены ранее для функции `periodogram`.

Разрешение по частоте будет равно F_s/L при $L_{fft} \geq L$ и F_s/L_{fft} при $L_{fft} < L$.

f , SW — векторы значений частот (Гц) и периодограммы Уэлча (Вт/Гц).

Периодограмма Бартлетта (16.14) может рассматриваться как частный случай периодограммы Уэлча в отсутствии перекрытия фрагментов и с прямоугольным окном:

```
[SB,f] = pwelch(x,rectwin(L),0,Lfft,Fs,'range')
```

График периодограммы Уэлча $\hat{S}_{WELCH}(f)$ (Вт/Гц) выводится с помощью функции:

```
plot(f,SW)
```

а $\hat{S}_{WELCH}(\omega)$ (дБ/Гц) — с помощью функции:

```
pwelch(x>window(L,winparameter),noverlap,Lfft,Fs,'range')
```

Для расчета периодограмм Бартлетта и Уэлча длины N с целью их сравнения с другими оценками СПМ следует задать значение L_{fft} равным длине N исходной последовательности.

Периодограмма Уэлча (16.18) может описываться в виде объекта `spectrum`:

```
Hs = spectrum.welch
```

где H_s — имя объекта.

Список свойств объекта H_s включает в себя:

- ☐ `FFTLlength` — размерность ДПФ для фрагмента;
- ☐ `SegmentLength` — длина фрагмента L ;
- ☐ `OverlapPercent` — длина перекрытия Q ;
- ☐ `WindowName` — используемое окно.

После создания объекта `spectrum` выводится график периодограммы Уэлча $\hat{S}_{WELCH}(f)$ (дБ/Гц) с помощью функции:

```
psd(Hs,x,'Fs',Fs)
```

16.1.6. Метод Блэкмана—Тьюки

Метод Блэкмана—Тьюки основан на использовании теоремы Винера—Хинчина, согласно которой СПМ $S(\omega)$ есть Фурье-изображение АКФ $R_x(m)$ последовательности $x(n)$, теоретически бесконечной длины:

$$S(\omega) = \frac{1}{f_d} \sum_{m=-\infty}^{\infty} R_x(m) e^{-j\omega m T}.$$

(О множителе $1/f_d = T$ см. в разд. 16.1.1.)

При конечной длине N последовательности имеем формулу для оценки СПМ:

$$\hat{S}(\omega) = \frac{1}{f_d} \sum_{m=-(N-1)}^{N-1} \hat{R}_x(m) e^{-j\omega mT}, \quad (16.19)$$

где $\hat{R}_x(m)$ — оценка АКФ — четная функция длины $L = 2N - 1$, центрированная относительно $m = 0$.

Блэкманом и Тьюки было предложено:

- при быстро затухающей оценке АКФ ограничить максимальный сдвиг по времени m значением $N_1 = \text{int}(N/10)$ (округление до ближайшего целого в сторону увеличения);
- использовать весовую функцию (окно) для уменьшения эффекта растекания спектра при вычислении оценки СПМ с помощью ДПФ (см. разд. 10.1.1), за счет чего достигается сглаживание оценки СПМ.

Оценка СПМ по методу Блэкмана—Тьюки $\hat{S}_{BT}(\omega)$ случайной последовательности $x(n)$ длины N определяется по формуле:

$$\hat{S}_{BT}(\omega) = \frac{1}{f_d} \sum_{m=-(N_1-1)}^{N_1-1} \hat{R}_x(m) w(m) e^{-j\omega mT}, \quad (16.20)$$

где $\hat{R}_x(m)$ — оценка АКФ¹ — четная функция длины $L_1 = 2N_1 - 1$, центрированная относительно $m = 0$; $w(m)$ — весовая функция (окно) длины $L_1 = 2N_1 - 1$, центрированная относительно $m = 0$.

Смещенная оценка АКФ вычисляется по формуле:

$$\hat{R}_x(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x(n) x(n+m), \quad -(N_1-1) \leq m \leq (N_1-1), \quad (16.21)$$

а несмещенная — по формуле:

$$\hat{R}_x(m) = \frac{1}{N-|m|} \sum_{n=0}^{N-|m|-1} x(n) x(n+m), \quad -(N_1-1) \leq m \leq (N_1-1). \quad (16.22)$$

Доказано [4], что смещенная оценка АКФ (16.21) гарантирует неотрицательность оценки СПМ (16.20), в то время как использование несмещенной оценки АКФ (16.22) может привести к отрицательной оценке СПМ, что противоречит физическому смыслу СПМ, которая является неотрицательной. Кроме того, в [2] показано, что при $N_1 \ll N$ в смещенной оценке АКФ (16.21) смещение незначительно.

Другой причиной получения отрицательной оценки СПМ (16.20) может стать неудачный выбор окна $w(m)$, в том случае если его Фурье-изображение имеет отри-

¹ Называемая также коррелограммой, откуда другое название метода — коррелограммный [4].

цательные значения. В частности, это возможно (но не обязательно) для окон Дирихле (прямоугольного), Хэмминга, Хэннинга, Кайзера, в отличие, например, от окна Бартлетта [2, 4, 5].

При быстро затухающей оценке АКФ оценка СПМ Блэкмана—Тьюки является *асимптотически несмещенной и состоятельной*.

В MATLAB для вычисления смещенной оценки АКФ (16.21), соответственно без учета или с учетом множителя $1/N$, используются функции:

```
R1 = xcorr(x,N1)
R1 = xcorr(x,N1,'biased')
```

где x — исходная последовательность длины N ; N_1 — максимальный сдвиг по времени, равный N_1 ; R_1 — вектор значений оценки АКФ длины $(2N_1 + 1)$, центрированной относительно $m = N_1 + 1$.

После этого формируется вектор R для оценки АКФ с длиной $L_1 = 2N_1 - 1$ в (16.20):

```
R = R1(2:length(R1)-1)
```

В MATLAB данная АКФ будет центрирована относительно $m = N_1$.

Для вычисления несмещенной оценки АКФ (16.22) с учетом множителя $1/N$ используется функция:

```
R1 = xcorr(x,N1,'unbiased')
```

и аналогичным образом формируется вектор R .

Для вычисления оценки СПМ Блэкмана—Тьюки (16.20) удобно воспользоваться формулой ДПФ для *симметричной* последовательности $\hat{R}_x(m)$ нечетной длины L_1 [3]:

$$\hat{S}_{BT}(k) = \frac{2}{f_d} \left[\sum_{m=0}^{N_1-1} \hat{R}_x(m) w(m) \cos\left(\frac{2\pi}{N_1} mk\right) - \hat{R}_x(0) \right], \quad k = 0, 1, \dots, L_1 - 1. \quad (16.23)$$

В MATLAB с учетом того, что оценка АКФ центрирована относительно N_1 , вещественное ДПФ длины L_1 рассчитывается с помощью функции `fft` (без учета множителя $1/f_d$):

```
S = 2*real(fft(R(N1:L1),L1))-R(N1)
```

где R и S — векторы значений оценок АКФ и СПМ длины L_1 .

Для расчета оценки СПМ длины N с целью ее сравнения с различными периодаграммами можно воспользоваться форматом функции `fft` с автоматическим согласованием длин АКФ и ДПФ:

```
S = 2*real(fft(R(N1:L1),N))-R(N1)
```

16.1.7. Моделирование случайной последовательности с требуемой АКФ

При моделировании различных методов обработки сигналов удобно иметь модель исходной случайной последовательности с заданными статистическими характеристиками. В этом разделе рассматривается моделирование случайной последовательности с заданной (требуемой) АКФ на основе нормального белого шума $x_{WN}(n)$ с математическим ожиданием $\mu_{WN} = 0$ и дисперсией $\sigma_{WN}^2 = 1$.

СПМ $S_{WN}(\omega)$ такого шума равномерна в бесконечной полосе частот и равна¹ (без учета множителя $1/f_d$):

$$S_{WN}(\omega) = N_0/2, \quad -\infty < \omega < \infty,$$

а АКФ $R_{WN}(m)$ равна²:

$$R_{WN}(m) = \begin{cases} (N_0/2)u_0(m), & m = 0; \\ 0, & m \neq 0, \end{cases} \quad (16.24)$$

где $u_0(m)$ — цифровой единичный импульс, $N_0/2$ — средняя мощность шума:

$$N_0/2 = R_{WN}(0) = \sigma_{WN}^2 + \mu_{WN}^2. \quad (16.25)$$

При длине шума $N \rightarrow \infty$ его средняя мощность равна $N_0/2 = \sigma_{WN}^2 = 1$.

Моделирование случайной последовательности с требуемой АКФ базируется на паре известных соотношений вход/выход ЛДС для СПМ и АКФ [1—3]:

$$S_y(\omega) = S_x(\omega) |H(e^{j\omega T})|^2; \quad (16.26)$$

$$R_y(m) = R_x(m) * R_h(m), \quad (16.27)$$

где символ "*" соответствует операции свертки. Для нормального белого шума $x_{WN}(n)$ эти соотношения принимают вид:

$$S_y(\omega) = (N_0/2) |H(e^{j\omega T})|^2; \quad (16.28)$$

$$R_y(m) = (N_0/2) R_h(m). \quad (16.29)$$

¹ В соответствии с теоремой Винера—Хинчина СПМ и АКФ случайной последовательности связаны парой формул преобразования Фурье. Для вещественной последовательности четность СПМ и АКФ позволяет в обеих формулах перейти к нулевому нижнему пределу, добавляя коэффициент 2. С этим связаны понятия двусторонней СПМ — в области $-\infty < \omega < \infty$ и односторонней — в области $\omega \geq 0$. Для СПМ белого шума приняты обозначения: $N_0/2$ — для двусторонней СПМ и N_0 — для односторонней.

² В отличие от аналогового белого шума с АКФ $(N_0/2)\delta(\tau)$, дискретный белый шум с АКФ $(N_0/2)u_0(m)$ является физически реализуемым процессом.

При выборе в качестве ЛДС КИХ-фильтра моделирование последовательности с требуемой АКФ включает в себя следующие шаги:

1. Моделирование воздействия КИХ-фильтра — нормального белого шума $x_{WN}(n)$ длины N и вычисление константы $N_0/2$ (16.25) с помощью функции `var`.
2. Моделирование требуемой АКФ $R_y(m)$ длины $L = 2N - 1$.
3. Вычисление СПМ $S_y(k)$ (16.28) с использованием (16.19) и ДПФ (16.23):

$$S_y(k) = \sum_{m=0}^{N-1} R_y(m) \cos\left(\frac{2\pi}{N}mk\right) - R_y(0), \quad k = 0, 1, \dots, L-1, \quad (16.30)$$

с помощью функции `fft`:

$$S = 2 * \text{real}(\text{fft}(R(N:L), L)) - R(N)$$

4. Вычисление АЧХ КИХ-фильтра $|H(k)|$, $k = 0, 1, \dots, L-1$, на основе (16.28).
5. Моделирование ФЧХ $\varphi(k)$ КИХ-фильтра 1-го типа¹:

$$\varphi(k) = -k\pi R/L, \quad k = 0, 1, \dots, L-1, \quad (16.31)$$

где R — четный порядок КИХ-фильтра, $(R+1)$ — его длина (длина ИХ), $[2(R+1)+1]$ — длина АКФ ИХ.

Согласно (16.29), требуемая АКФ совпадает с АКФ ИХ (с точностью до множителя), однако *быстро затухающую* АКФ можно представить как АКФ ИХ длины $[2(R+1)+1]$, дополненную справа и слева нулями до длины $L = 2N - 1$, где $R \ll N$.

В этом случае порядок R будет равен длине требуемой АКФ от ее центрального отсчета до ближайшего отсчета, значение которого близко к нулю.

Необходимо иметь в виду, что порядок R определяет начальные нули реакции КИХ-фильтра при ННУ, поэтому его не следует завышать.

6. Вычисление частотной характеристики КИХ-фильтра:

$$H(k) = |H(k)| e^{j\varphi(k)}, \quad k = 0, 1, \dots, L-1. \quad (16.32)$$

7. Вычисление ИХ КИХ-фильтра на периоде L круговой свертки с помощью ОДПФ $H(k)$ с использованием функции `ifft` (см. разд. 9.1.1).
8. Вычисление реакции КИХ-фильтра длины N (случайной последовательности с требуемой АКФ) по формуле свертки на основе ДПФ с использованием функции `fftfilt` (см. разд. 10.1.3).

¹ Выбор КИХ-фильтра 1-го типа не требует проверки на соответствие типу избирательности фильтра с полученной АЧХ.

16.1.8. Основные параметры окон

Окна, используемые в MATLAB, анализируются в GUI Window Design and Analysis Tool (см. разд. 16.1.1). Для каждого окна выводятся график во временной области и модуль спектральной плотности в децибелах (дБ) в шкале нормированных частот $f/(f_d/2)$. Под графиками указываются три основных параметра окна, влияющие на оценку СПМ.

- Leakage Factor — коэффициент утечки, равный отношению энергии боковых лепестков окна к его общей энергии (в %).

Чем больше коэффициент утечки, тем сильнее эффект сглаживания оценки СПМ.

Однако в отдельных случаях высокий уровень энергии, приходящейся на боковые лепестки, может приводить к значительному искажению полезных спектральных составляющих с малой энергией.

- Relative sidelobe attenuation — максимальный уровень модуля спектральной плотности боковых лепестков окна относительно главного (в дБ).

Чем выше максимальный уровень модуля спектральной плотности боковых лепестков относительно главного лепестка, тем сильнее эффект сглаживания оценки СПМ.

- Mainlobe width — ширина полосы модуля спектральной плотности окна на уровне -3 дБ от максимума главного лепестка¹, сосредоточенного вокруг нулевой частоты.

Чем больше ширина полосы модуля спектральной плотности, тем сильнее эффект сглаживания оценки СПМ. Однако, чем шире главный лепесток, тем меньше уровень боковых лепестков и наоборот. Таким образом, выбор окна определяется из соображений компромисса между шириной главного лепестка и уровнем боковых лепестков.

Анализ влияния параметров окна на оценку СПМ предусмотрен в одном из пунктов самостоятельного задания (см. разд. 16.5). Подробно он рассматривается в [1, 4—6].

16.1.9. Спектрограмма

Спектрограммой (spectrogram) сигнала называют его *мгновенный спектр*, зависящий от времени.

Спектрограмму принято представлять в виде графика в системе координат "время—частота" с отображением зависящей от времени СПМ с помощью цвета. Для этого исходная последовательность делится на фрагменты (допустимо их перекрывание), взвешенные окном.

¹ Данный уровень соответствует двукратному уменьшению мощности, поскольку $-10 \lg 2 \approx -3$ дБ.

В MATLAB для построения графика спектрограммы используется функция:

`spectrogram(x, window(L, winparameter), noverlap, Lfft, Fs, 'flegloc')`

где x — исходная последовательность длины N ; `window(L, winparameter)` подобен одноименному параметру функции `periodogram` (см. разд. 16.1.1) для фрагмента длины L ; по умолчанию используется окно Хэмминга, для которого достаточно указать длину L ; при указании пустой матрицы `[]` исходная последовательность разбивается на 8 фрагментов, а "лишние" последние отсчеты отбрасываются; `nooverlap` — величина перекрытия (количество отсчетов); при указании пустой матрицы `[]` по умолчанию равно `length(L)/2`; `Lfft` — размерность ДПФ при вычислении СПМ; при указании пустой матрицы `[]` по умолчанию равно `max([256, 2^nextpow2(L)])`; F_s — частота дискретизации f_d (Гц); при указании пустой матрицы `[]` по умолчанию равно 2 Гц; `flegloc` — параметр, определяющий направление оси частот и принимающий значения: `xaxis` — ось частот направлена по горизонтали (по умолчанию) и `yaxis` — по вертикали.

Спектрограмма вещественной последовательности автоматически выводится в основной полосе частот $[0; f_d/2]$, а комплексной — на периоде $[0; f_d]$.

По умолчанию для отображения спектрограммы выбирается цветовая палитра `jet` (см. табл. 4.5), которую можно изменить с помощью функции:

`colormap(<символическое имя палитры>)`

стоящей до или после функции `spectrogram`.

При желании на графике спектрограммы можно вывести шкалу цветов, верхний уровень которой соответствует максимальному значению СПМ, по команде:

`colorbar`

стоящей обязательно после функции `spectrogram`.

16.2. Содержание лабораторной работы

Содержание работы связано с моделированием методов непараметрического спектрального анализа случайных последовательностей, определением показателей качества оценок СПМ и расчетом спектрограмм сигналов программными средствами MATLAB.

16.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файла `lr_16`, который хранится на прилагаемом компакт-диске в папке `LAB_DSP\LAB_16`.

Перед выполнением работы необходимо сохранить путь к папке `LAB_16` по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 16.1 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$. Функция $N_{бр} \bmod M$ в записи исходных данных означает вычисление значения $N_{бр}$ по модулю M .

На прилагаемом компакт-диске в папке Tables\Tables_16 хранятся табл. 16.1 исходных данных и пример ее заполнения для $N_{бр} = 1$.

Таблица 16.1. Таблица исходных данных

Переменная	Назначение	Значение	Идентификатор
$N_{бр}$	Номер бригады	$N_{бр}$	Nb =
N	Длина последовательности	$N = 128$	N = 128
f_d	Частота дискретизации	$f_d = 1000(N_{бр} \bmod 5 + 1)$	Fs =
A_1	Амплитуды дискретных гармоник	$A_1 = 0,8 + 0,01N_{бр}$	A1 =
A_2		$A_2 = 1,5A_1$	A2 =
f_1	Частоты дискретных гармоник	$f_1 = f_d/8$	f1 =
f_2		$f_2 = 2f_1$	f2 =
σ_1 σ_2 σ_3 σ_4	Значения СКО	$\sigma_1 = 0$ $\sigma_2 = A_1$ $\sigma_3 = 2A_1$ $\sigma_4 = 4A_1$	Вектор sigma = [...]

Задание на лабораторную работу заключается в моделировании методов непараметрического спектрального анализа и включает в себя следующие пункты:

1. Проверка информативности периодограммы в зависимости от уровня шума.
- Проверить информативность периодограммы (16.2) последовательности $x_e(n)$ (идентификатор x_e) длины N — аддитивной смеси суммы двух дискретных гармоник с нормальным белым шумом $e(n)$:

$$x_e(n) = x(n) + e(n), \tag{16.33}$$

где

$$x(n) = A_1 \cos\left(\frac{2\pi f_1 n}{f_d}\right) + A_2 \cos\left(\frac{2\pi f_2 n}{f_d}\right) = A_1 \cos(\hat{\omega}_1 n) + A_2 \cos(\hat{\omega}_2 n), \tag{16.34}$$

при различных уровнях шума, задаваемых значениями СКО¹ σ_1 , σ_2 , σ_3 и σ_4 .

¹ СКО (STD) равно корню квадратному из дисперсии, поэтому для моделирования нормального белого шума с заданным СКО достаточно умножить шум на значение СКО (см. разд. 7.2.2).

Вывести графики:

- последовательности $x_e(n)$ при различных значениях СКО с помощью функции `plot`;
- их периодограмм (16.2) (идентификатор `se`) на периоде в шкале частот f (Гц).

Пояснить:

- влияние уровня шума на информативность периодограммы;
 - при каком уровне шума невозможно визуальное различение гармоник.
2. Проверка информативности периодограммы в зависимости от периода дискретизации по частоте.

Проверить информативность периодограммы (16.2) последовательности $x_e(n)$ (16.33) нормального белого шума с нулевым средним и единичной дисперсией.

Вывести графики периодограммы (16.2) (идентификатор `se`) в основной полосе частот, задавая в функции `periodogram` различную размерность ДПФ $M = N/16, N/8, N, 8N$ (вектор `m`).

Пояснить:

- чему равно разрешение по частоте при заданных размерностях ДПФ;
 - при какой размерности ДПФ и почему периодограмма оказалась неинформативной;
 - при какой размерности ДПФ периодограмма наиболее информативна (наиболее узкая и гладкая и содержит обе гармоники).
3. Проверка оценки СПМ на асимптотическую несмещенность и состоятельность.

Для нормального белого шума $e(n)$ (идентификатор `e`) с нулевым средним и единичной дисперсией вычислить:

- истинную СПМ $S_{WN}(f)$ (16.25) (идентификатор `SWN`) шума с учетом множителя $1/f_d$, задавая длину шума $N = 100000$ (теоретически $N \rightarrow \infty$);
- в цикле для каждой длины шума N от $N_{\text{нач}} = 1000$ до $N_{\text{кон}} = 100000$ с шагом $\Delta N = 1000$ (вектор `N_WN`):
 - периодограмму $\hat{S}_{WN}(f)$ (16.2) (идентификатор `SWN_estimate`);
 - смещение β (16.6) (идентификатор `beta`);
 - средний квадрат (16.7) отклонения истинной СПМ от ее оценки (идентификатор `mean_square`).

Вывести графики зависимости от длины шума N :

- смещения β ;
- среднего квадрата отклонения истинной СПМ от ее оценки.

Сделать вывод об оценке СПМ (16.2) нормального белого шума (смещенная, асимптотически несмещенная, состоятельная, несостоятельная).

4. Моделирование случайной последовательности с требуемой АКФ.

Сформировать случайную последовательность (идентификатор y) длины $N=1000$ (идентификатор N) с требуемой АКФ $R_y(m)$ (идентификатор $R_{y\_required}$):

$$R_y(m) = 0,25 \cdot 0,95^{|m|}, \quad |m| = 0, \dots, N-1. \quad (16.35)$$

Вывести графики:

- требуемой АКФ $R_y(m)$, центрированной относительно $m = N$.

По графику АКФ, используя кнопку **Zoom in** на панели инструментов, определить *четный* порядок R КИХ-фильтра (идентификатор R), равный длине АКФ от ее центрального отсчета до *ближайшего* отсчета, значение которого *близко* к нулю, и ввести его с клавиатуры.

Порядок R определяет начальные нули реакции КИХ-фильтра при ННУ, поэтому его не следует завышать;

- требуемой АКФ $R_y(m)$ и оценки АКФ $\hat{R}_y(m)$ (идентификатор $R_{y\_estimate}$), центрированных относительно $m = N$;
- ИХ $h(n)$ КИХ-фильтра (идентификатор h) с помощью функции `stem`, воздействия $xw(n)$ (идентификатор xw) и реакции КИХ-фильтра (идентификатор y_ACF) с помощью функции `plot`.

Пояснить:

- какой тип КИХ-фильтра выбран и почему;
- что используется в качестве воздействия КИХ-фильтра;
- что собой представляет реакция КИХ-фильтра.

5. Фильтрация случайной последовательности с требуемой АКФ.

Удалить тренд (подавить низкочастотные составляющие) в последовательности с требуемой АКФ (идентификатор y_ACF). В данном случае тренд приведет к искаженной картине распределения средней мощности по частоте за счет низкочастотных составляющих с большой энергией.

Использовать БИХ-фильтр ФВЧ Баттерворта. Требования к АЧХ не критичны, поэтому выбрать фильтр 3-го порядка с нормированной частотой среза $W_{dn}=0.3$ (см. разд. 13.1.5).

Вывести графики исходной и преобразованной $y(n)$ (идентификатор y) последовательностей с помощью функции `plot`.

Пояснить:

- что называют трендом во временной области;
- чему равны значения АЧХ и АЧХ (дБ) на частоте среза.

6. Расчет периодограммы.

Рассчитать периодограмму $\hat{S}(f)$ (16.2) (идентификатор `s`) случайной последовательности $y(n)$ (см. п. 5).

Вывести графики периодограммы $\hat{S}(f)$ (Вт/Гц) и $\hat{S}(f)$ (дБ/Гц) и пояснить связь между ними.

7. Расчет периодограммы Даньелла.

Рассчитать периодограмму Даньелла $\hat{S}_{DANIELL}(f)$ (16.11) (идентификатор `SD`) случайной последовательности $y(n)$ (см. п. 5) с помощью функции `smooth`.

Исследовать эффективность сглаживания осцилляций в зависимости от количества соседних усредняемых частот $2K+1$ при $K=5, 10, 20$ (вектор `k`).

Вывести графики периодограммы $\hat{S}(f)$ (см. п. 6) и периодограмм Даньелла.

Пояснить:

- при каком значении K периодограмма Даньелла наименее осциллирующая;
- как изменилась интенсивность осцилляций периодограммы Даньелла по сравнению с периодограммой.

8. Расчет периодограммы Бартлетта.

Рассчитать периодограмму Бартлетта $\hat{S}_{BARTLETT}(f)$ (16.14) (идентификатор `SB`) случайной последовательности $y(n)$ (см. п. 5).

Исследовать эффективность сглаживания осцилляций в зависимости от длины фрагмента $L=10, 20, 40$ (вектор `L`).

Вывести графики периодограммы $\hat{S}(f)$ (см. п. 6) и периодограмм Бартлетта.

Пояснить:

- при какой длине фрагмента L периодограмма Бартлетта наименее осциллирующая;
- как изменилась интенсивность осцилляций периодограмм Бартлетта по сравнению с периодограммами Даньелла.

9. Расчет периодограммы Уэлча.

Рассчитать периодограмму Уэлча $\hat{S}_{WELCH}(f)$ (16.18) (идентификатор `SW`) случайной последовательности $y(n)$ (см. п. 5).

Исследовать эффективность сглаживания осцилляций в зависимости от длины фрагмента $L=0,1N; 0,05N; 0,025N$ (вектор `L`) и величины перекрытия $Q=0,0125N$ (вектор `Q`) для каждой длины фрагмента L .

Длины фрагментов и величину перекрытия округлить до ближайшего целого в сторону увеличения.

Использовать окно Хэмминга (по умолчанию).

Вывести графики периодограммы $\hat{S}(f)$ (см. п. 6) и периодограмм Уэлча.

Пояснить:

- при какой длине фрагмента L периодограмма Уэлча наименее осциллирующая;
- как изменилась интенсивность осцилляций периодограмм Уэлча по сравнению с периодограммой и периодограммами Даньелла и Бартлетта.

10. Расчет оценки СПМ по методу Блэкмана—Тьюки.

Рассчитать оценку СПМ по методу Блэкмана—Тьюки $\hat{S}_{BT}(f)$ (16.21) (идентификатор `SBT`) случайной последовательности $y(n)$ (см. п. 5).

Исследовать для прямоугольного окна и окон Хэмминга и Чебышева с параметрами, заданными по умолчанию.

Вывести графики периодограммы $\hat{S}(f)$ (см. п. 6) и оценок СПМ по методу Блэкмана—Тьюки.

Пояснить:

- при каком окне оценка СПМ по методу Блэкмана—Тьюки наименее осциллирующая;
- как изменилась интенсивность осцилляций оценки СПМ по методу Блэкмана—Тьюки по сравнению с периодограммой и периодограммами Даньелла, Бартлетта и Уэлча.

11. Определение показателей качества оценок СПМ.

Вывести значения СКО (16.10) и добротностей (16.8) для следующих оценок СПМ:

- периодограммы (идентификаторы `STD_S` и `Q_S`);
- периодограмм Даньелла (идентификаторы `STD_SD` и `Q_SD`);
- периодограмм Бартлетта (идентификаторы `STD_SB` и `Q_SB`);
- периодограмм Уэлча (идентификаторы `STD_SW` и `Q_SW`);
- оценок СПМ по методу Блэкмана—Тьюки (идентификаторы `STD_SBT` и `Q_SBT`).

Пояснить:

- какая из оценок СПМ является наилучшей и наихудшей по критериям СКО и добротности;
- соответствие между показателями качества и графиками оценок СПМ.

12. Построение спектрограммы.

Вывести график спектрограммы дискретного гармонического сигнала $x(n)$ (16.34) (идентификатор `x`) длины $N = 4000$.

Для функции `spectrogram` задать параметры: окно Хэмминга, длина фрагмента 128, величина перекрытия 120, размерность ДПФ 128, ось частот направлена по вертикали.

Пояснить, с какой целью строится спектрограмма.

16.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должна быть представлена табл. 16.1 исходных данных для своего номера бригады $N_{бр}$.

Для *запуска* лабораторной работы необходимо обратиться к script-файлу `lr_16` его имени:

```
>> lr_16
```

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш <Ctrl>+<Break>.

При выполнении script-файла текущие окна с графиками *не закрывать*.

Листинг script-файла `lr_16` имеет вид:

```
>> type lr_16
script
clc
clear
disp('% ЛР №16. НЕПАРАМЕТРИЧЕСКИЙ СПЕКТРАЛЬНЫЙ АНАЛИЗ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
N = input('N = ');             % ДЛИНА ПОСЛЕДОВАТЕЛЬНОСТИ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
A1 = input('A1 = ');           % АМПЛИТУДЫ ДИСКРЕТНЫХ ГАРМОНИК
A2 = input('A2 = ');
f1 = input('f1 = ');           % ЧАСТОТЫ ДИСКРЕТНЫХ ГАРМОНИК (Гц)
f2 = input('f2 = ');
sigma = input('sigma = '); % ВЕКТОР ЗНАЧЕНИЙ СКО ШУМА
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
```

```

disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.1. ПРОВЕРКА ИНФОРМАТИВНОСТИ ПЕРИОДОГРАММЫ В ЗАВИСИМОСТИ ОТ УРОВНЯ  
ШУМА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ПОСЛЕДОВАТЕЛЬНОСТЕЙ и ПЕРИОДОГРАММ')
disp('% с различными СКО шума нажмите <ENTER>')
pause
w1 = 2*pi*f1/Fs; w2 = 2*pi*f2/Fs; % НОРМИРОВАННЫЕ ЧАСТОТЫ ДИСКРЕТНЫХ ГАРМОНИК  
(РАД)
n = 0:(N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
x = A1*cos(w1*n')+A2*cos(w2*n'); % ПОСЛЕДОВАТЕЛЬНОСТЬ – СУММА ДВУХ ГАРМОНИК  
(ВЕКТОР-СТОЛБЕЦ)
figure('Name',' Harmonic Signals Embedded in White Gaussian Noise and  
Periodograms','NumberTitle','off')
for i = 1:length(sigma) % ИНДЕКС ЭЛЕМЕНТОВ ВЕКТОРА sigma
xe = x'+sigma(i).*randn(1,N); % ПОСЛЕДОВАТЕЛЬНОСТИ С ЗАДАНЫМИ СКО ШУМА (x'-  
ВЕКТОР-СТРОКА)
subplot(4,2,2*i-1), plot(n,xe,'Linewidth',2), grid
xlabel('n'), ylabel(strcat('xe',num2str(i),'(n)'))
title(strcat(['Sequence ',num2str(i), ' STD = ',num2str(sigma(i))]))
[Se,f] = periodogram(xe,[],N,Fs,'twosided'); % ПЕРИОДОГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ  
С ЗАДАНЫМ СКО ШУМА
subplot(4,2,2*i), plot(f,Se,'Linewidth',2), grid
xlabel('f (Hz)'), ylabel(strcat('S',num2str(i),'(f)'))
title(strcat(['Periodogram ',num2str(i),' (Vt/Hz) STD = ',num2str(sigma(i))]))
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ПРОВЕРКА ИНФОРМАТИВНОСТИ ПЕРИОДОГРАММЫ В ЗАВИСИМОСТИ  
ОТ ПЕРИОДА ДИСКРЕТИЗАЦИИ ПО ЧАСТОТЕ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ПЕРИОДОГРАММ при различной размерности ДПФ нажмите  
<ENTER>')
pause
figure('Name','Periodograms of Harmonic Signal with defferent DFT  
size','NumberTitle','off')
xe = x'+randn(1,N); % ПОСЛЕДОВАТЕЛЬНОСТЬ ДЛИНЫ N
M = [N/16 N/8 N 8*N]; % ВЕКТОР РАЗМЕРНОСТЕЙ ДПФ

```



```

for i = 1:length(M)
[Se,f] = periodogram(xe,[],M(i),Fs,'onesided');    % ПЕРИОДОГРАММА В ОСНОВНОЙ
ПОЛОСЕ ЧАСТОТ ПРИ ЗАДАННОЙ РАЗМЕРНОСТИ ДПФ
subplot(4,1,i), plot(f, Se,'Linewidth',2),grid
xlabel('f (Hz)'), ylabel(strcat('S',num2str(i),'(f)'))
title(strcat(['Periodogram ',num2str(i),' (Vt/Hz)   length DFT=',
num2str(M(i))]))
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. ПРОВЕРКА ОЦЕНКИ СПМ НА АСИМПТОТИЧЕСКУЮ НЕСМЕЩЕННОСТЬ
И СОСТОЯТЕЛЬНОСТЬ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ зависимости СМЕЩЕНИЯ и СРЕДНЕГО КВАДРАТА')
disp('% ОТКЛОНЕНИЯ ИСТИННОЙ СПМ ОТ ЕЕ ОЦЕНКИ нажмите <ENTER>')
pause
SWN = var(randn(1,100000))/Fs;    % ИСТИННАЯ СПМ НОРМАЛЬНОГО БЕЛОГО ШУМА
N_WN = 1000:1000:100000;    % ВЕКТОР ДЛИН ШУМА
for i = 1:length(N_WN)    % ИНДЕКС ЭЛЕМЕНТОВ ВЕКТОРА N_WN
e = randn(1,N_WN(i));    % ШУМ ЗАДАННОЙ ДЛИНЫ
SWN_estimate = periodogram(e,[],N_WN(i),Fs,'twosided');    % ОЦЕНКА СПМ ШУМА
beta(i) = mean(SWN-SWN_estimate);    % СМЕЩЕНИЕ ОЦЕНКИ СПМ
mean_square (i) = var(SWN_estimate)+beta(i)^2;    % СРЕДНИЙ КВАДРАТ
ОТКЛОНЕНИЯ ИСТИННОЙ СПМ ОТ ЕЕ ОЦЕНКИ
end
figure('Name','Bias and Mean square Deviation of true PSD from its
Estimate','NumberTitle','off')
subplot(2,1,1), plot(N_WN, beta,'LineWidth',2), grid, xlabel('N')
ylabel('beta')
title('Bias')
subplot(2,1,2), plot(N_WN, mean_square,'LineWidth',2), grid, xlabel('N')
ylabel('meanerr')
title('Mean square Deviation of true PSD from its Estimate')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. ФОРМИРОВАНИЕ СЛУЧАЙНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ С ТРЕБУЕМОЙ АКФ')
N = 1000;    % ДЛИНА ШУМА
xw = randn(1,N);    % НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ ДЛИНЫ N

```

```

N02 = var(xw)+(mean(xw)).^2;           % КОНСТАНТА N0/2
m = -(N-1):(N-1);                     % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ АКФ,
ЦЕНТРИРОВАННОЙ ОТНОСИТЕЛЬНО m = 0
Ry_required = 0.25.*0.95.^abs(m);      % ТРЕБУЕМАЯ АКФ
L = 2*N-1;                            % ДЛИНА АКФ
m = 0:L-1;                            % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ АКФ,
ЦЕНТРИРОВАННОЙ ОТНОСИТЕЛЬНО m = N
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА ТРЕБУЕМОЙ АКФ нажмите <ENTER>')
pause
figure('Name','Required ACF','NumberTitle','off')
stem(m,Ry_required), grid, xlabel('m'), title('Required ACF Ry')
disp('%')
disp('%')
disp('% С помощью кнопки Zoom in определите и введите ЧЕТНЫЙ ПОРЯДОК
КИХ-ФИЛЬТРА')
DATA=0;
while DATA==0
disp('%')
R = input('      R = ');               % ЧЕТНЫЙ ПОРЯДОК КИХ-ФИЛЬТРА
disp('%')
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
Sy = 2*real(fft(Ry_required(N:L),L))-Ry_required(N); % СПМ В L ТОЧКАХ,
ВЫЧИСЛЕННАЯ ПО ТРЕБУЕМОЙ АКФ
A = sqrt(real(Sy)./N02);               % АЧХ КИХ-ФИЛЬТРА В L ТОЧКАХ
k = 0:L-1;                            % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
F = -k*pi*R/L;                        % ЛФЧХ КИХ-ФИЛЬТРА В L ТОЧКАХ
j = sqrt(-1);
H = A.*exp(j*F);                      % ЧХ КИХ-ФИЛЬТРА В L ТОЧКАХ
h1 = real(ifft(H));                   % ИХ КИХ-ФИЛЬТРА НА ПЕРИОДЕ L КРУГОВОЙ
СВЕРТКИ
y_ACF = fftfilt(h1,xw);                % СЛУЧАЙНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ДЛИНЫ N
С ТРЕБУЕМОЙ АКФ НА ВЫХОДЕ КИХ-ФИЛЬТРА
h = h1(1:R+1);                        % ИХ ДЛИНЫ R+1
Ry_estimate = xcorr(y_ACF)./N;         % ОЦЕНКА АКФ РЕАКЦИИ КИХ-ФИЛЬТРА,
ЦЕНТРИРОВАННАЯ ОТНОСИТЕЛЬНО N
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА ТРЕБУЕМОЙ АКФ И ОЦЕНКИ АКФ РЕАКЦИИ нажмите <ENTER>')
pause
figure('Name','Required ACF and ACF Estimate of Output Signal','NumberTitle',
'off')
subplot(2,1,1), stem(m,Ry_required), grid, title('Required ACF Ry')

```

```

subplot(2,1,2), stem(m,Ry_estimate), grid, xlabel('m')
title(' ACF Estimate of Output Signal – Ry estimate')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИХ, НОРМАЛЬНОГО БЕЛОГО ШУМА И РЕАКЦИИ КИХ-фильтра
нажмите <ENTER>')
pause
n = 0:(N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ ВОЗДЕЙСТВИЯ И РЕАКЦИИ
КИХ-ФИЛЬТРА
figure('Name','Impulse Response, Input and Output Signals','NumberTitle',
'off')
subplot(3,1,1), stem(0:R,h), grid, title('Impulse Response h(n)')
subplot(3,1,2), plot(n,xw), grid
title('Input Signal – White Gaussian Noise')
subplot(3,1,3), plot(n,y_ACF), grid, xlabel('n')
title('Output Signal with Required ACF – y ACF')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ФИЛЬТРАЦИЯ СЛУЧАЙНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ С ТРЕБУЕМОЙ АКФ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИСХОДНОЙ И ПРЕОБРАЗОВАННОЙ последовательностей
нажмите <ENTER>')
pause
[b,a] = butter(3,0.3,'high'); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА ФВЧ БАТТЕРВОРТА
y = filter(b,a,y_ACF); % ПОСЛЕДОВАТЕЛЬНОСТЬ НА ВЫХОДЕ БИХ-ФИЛЬТРА
ПОСЛЕ УДАЛЕНИЯ ТРЕНДА
figure('Name','Input and Output Signals of Butterworth filter','NumberTitle',
'off')
subplot(2,1,1), plot(n,y_ACF), grid, title('Input Signal of Butterworth
filter – y ACF')
subplot(2,1,2), plot(n,y), grid, xlabel('n'), ylabel('y(n)')
title('Output Signal of Butterworth filter – y')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. РАСЧЕТ ПЕРИОДОГРАММЫ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА ПЕРИОДОГРАММЫ нажмите <ENTER>')
pause

```

```

[S,f] = periodogram(y,[],N,Fs,'twosided');           % ПЕРИОДОГРАММА СЛУЧАЙНОЙ
ПОСЛЕДОВАТЕЛЬНОСТИ
figure('Name','Periodogram of the Non-white Gaussian Noise','NumberTitle',
'off')
subplot(2,1,1), plot(f,S,'Linewidth',2), grid
xlabel('f (Hz)'), ylabel('S(f) (Vt/Hz)')
title('Periodogram of the Non-white Gaussian noise')
subplot(2,1,2), periodogram(y,[],N,Fs,'twosided')
xlabel('f (Hz)'), ylabel('S(f) (dB/Hz)')
title('Periodogram of the Non-white Gaussian noise')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. РАСЧЕТ ПЕРИОДОГРАММЫ ДАНЬЕЛЛА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ПЕРИОДОГРАММ ДАНЬЕЛЛА нажмите <ENTER>')
pause
K = [5 10 20];                                     % ВЕКТОР КОЛИЧЕСТВА УСРЕДНЯЕМЫХ ЧАСТОТ
figure('Name','Daniell Periodograms for the Different Number of Frequency
Intervals','NumberTitle','off')
for i = 1:3                                         % ИНДЕКС ЭЛЕМЕНТОВ ВЕКТОРА K
S1 = [S(N-K(i)+1:N); S; S(1:K(i))]; % ПЕРИОДОГРАММА, ПЕРИОДИЧЕСКИ ПРОДОЛЖЕННАЯ
СЛЕВА И СПРАВА НА K ОТСЧЕТОВ
S2 = smoooth(S1,K(i)); % РЕЗУЛЬТАТ ВЫЧИСЛЕНИЯ СКОЛЬЗЯЩЕГО СРЕДНЕГО
(ВЕКТОР ДЛИНЫ N+2K(i))
SD(:,i) = S2(K(i)+1:N+K(i)); % ПЕРИОДОГРАММА ДАНЬЕЛЛА (ВЕКТОР ДЛИНЫ N) ДЛЯ
КОЛИЧЕСТВА УСРЕДНЯЕМЫХ ЧАСТОТ K(i)
subplot(4,1,i+1), plot(f, SD(:,i),'Linewidth',2), grid
xlabel('f (Hz)'), ylabel(strcat('SD',num2str(i),'(f)'))
title(strcat(['Daniell Periodogram ',num2str(i),' Frequency Interval
2K+1, K=',num2str(K(i))]))
end
subplot(4,1,1)
plot(f,S,'Linewidth',2), grid, xlabel('f (Hz)'), ylabel('S(f)')
title('Original non-modified periodogram (Vt/Hz)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.8. РАСЧЕТ ПЕРИОДОГРАММЫ БАРТЛЕТТА')
disp('%')
disp('%')

```

```

disp('% Для вывода ГРАФИКОВ ПЕРИОДОГРАММЫ БАРТЛЕТТА нажмите <ENTER>')
pause
L = [10 20 40]; % ВЕКТОР ДЛИН ФРАГМЕНТОВ
figure('Name', ' Bartlett Periodograms for Different Fragment
Lengths', 'NumberTitle', 'off')
for i = 1:3 % ИНДЕКС ЭЛЕМЕНТОВ ВЕКТОРА L
SB(:,i) = pwelch(y,rectwin(L(i)),0,N,Fs,'twosided'); % ПЕРИОДОГРАММА БАРТЛЕТТА
ДЛЯ ФРАГМЕНТА ДЛИНЫ L(i)
subplot(4,1,i+1), plot(f, SB(:,i),'Linewidth',2), grid
xlabel('f (Hz)'), ylabel(strcat('SB',num2str(i),'(f)'))
title(strcat([' Bartlett Periodogram ',num2str(i),' L =',num2str(L(i))]))
end
subplot(4,1,1)
plot(f,S,'Linewidth',2), grid, xlabel('f (Hz)'), ylabel('S(f)')
title('Original non-modified periodogram (Vt/Hz)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.9. РАСЧЕТ ПЕРИОДОГРАММЫ УЭЛЧА')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ПЕРИОДОГРАММЫ УЭЛЧА нажмите <ENTER>')
pause
num = 1; % ПОРЯДКОВЫЙ НОМЕР ПЕРИОДОГРАММЫ УЭЛЧА
L = ceil([0.1 0.05 0.025].*length(y)); % ВЕКТОР ДЛИН ФРАГМЕНТОВ
figure('Name','Welch Periodograms for Different Fragment Lengths and
Overlapping', 'NumberTitle', 'off')
for i = 1:3 % ИНДЕКС ЭЛЕМЕНТОВ ВЕКТОРА L
Q = ceil(0.0125*length(y)); % ВЕЛИЧИНА ПЕРЕКРЫТИЯ
SW(:,num) = pwelch(y,L(i),Q,N,Fs,'twosided'); % ПЕРИОДОГРАММА УЭЛЧА ПРИ ДЛИНЕ
ФРАГМЕНТА L(i) И ВЕЛИЧИНЕ ПЕРЕКРЫТИЯ Q
subplot(4,1,i+1), plot(f,SW(:,num),'Linewidth',2), grid
xlabel('f (Hz)'), ylabel('S(f)')
title(strcat(['Welch Periodogram Fragment length L = ',num2str(L(i)), '
Overlapping Q = ',num2str(Q)]))
num = num+1;
subplot(4,1,1), plot(f,S,'Linewidth',2), grid, xlabel('f (Hz)')
ylabel('S(f)'), title('Original non-modified periodogram (Vt/Hz)')
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')

```

```

disp('% п.10. РАСЧЕТ ОЦЕНКИ СПМ ПО МЕТОДУ БЛЭКМАНА-ТУЬКИ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ОЦЕНКИ СПМ, вычисленных по методу БЛЭКМАНА-ТУЬКИ,
нажмите <ENTER>')
pause
N1 = ceil(N/10); % МАКСИМАЛЬНЫЙ СДВИГ ПО ВРЕМЕНИ ДЛЯ ОЦЕНКИ АКФ
ОТФИЛЬТРОВАННОГО ШУМА
R1 = (1/N).*xcorr(y,N1); % ОЦЕНКА АКФ ОТФИЛЬТРОВАННОГО ШУМА ДЛИНЫ 2N1+1,
ЦЕНТРИРОВАННАЯ ОТНОСИТЕЛЬНО N1+1
R = R1(2:length(R1)-1); % ОЦЕНКА АКФ ОТФИЛЬТРОВАННОГО ШУМА ДЛИНЫ 2N1-1,
ЦЕНТРИРОВАННАЯ ОТНОСИТЕЛЬНО N1
L1 = length(R); % ДЛИНА ОЦЕНКИ АКФ
Rw(:,1) = R'.*rectwin(L1); % АКФ, ВЗВЕШЕННАЯ ПРЯМОУГОЛЬНЫМ ОКНОМ
Rw(:,2) = R'.*hamming(L1); % АКФ, ВЗВЕШЕННАЯ ОКНОМ ХЭММИНГА
Rw(:,3) = R'.*chebwin(L1); % АКФ, ВЗВЕШЕННАЯ ОКНОМ ЧЕБЫШЕВА
name(1).win = 'Rectangular Window'; % ИМЕНА ОКОН (МАССИВ ЗАПИСЕЙ name С ОДНИМ
ПОЛЕМ win)
name(2).win = 'Hamming Window';
name(3).win = 'Chebyshev Window';
f = 0:Fs/N:Fs-Fs/N; % ВЕКТОР ЧАСТОТ (Гц)
figure('Name','PSD estimates by the Blackman-Tukey method','NumberTitle',
'off')
for i = 1:3 % НОМЕРА ОКОН
SBT(:,i) = (1/Fs)*(2*real(fft(Rw(N1:L1,i),N))-Rw(N1,i)); % ОЦЕНКА СПМ ДЛИНЫ
N, ВЫЧИСЛЕННАЯ ПО АКФ, ВЗВЕШЕННОЙ ОКНОМ
subplot(4,1,i+1), plot(f,SBT(:,i),'Linewidth',2), grid
xlabel('f (Hz)'), ylabel(strcat('SBT',num2str(i),'(f)'))
title(['PSD Estimate - ',strcat(name(i).win)])
end
subplot(4,1,1)
plot(f,S,'Linewidth',2), grid, xlabel('f (Hz)'), ylabel('S(f)')
title('Original non-modified periodogram (Vt/Hz)')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.11. ОПРЕДЕЛЕНИЕ ПОКАЗАТЕЛЕЙ КАЧЕСТВА ОЦЕНОК СПМ')
disp('%')
disp('%')
disp('% Для вывода ЗНАЧЕНИЙ СКО нажмите <ENTER>')
pause
disp('%')
disp(' СКО периодограммы')
format long % ДЛИННЫЙ ФОРМАТ ДЛЯ ВЫВОДА СКО
STD_S = std(S) % СКО ПЕРИОДОГРАММЫ

```

```

disp('    СКО периодограмм ДАНЬЕЛЛА при различном количестве усредняемых частот K')
STD_SD = [K' std(SD)'] % КОЛИЧЕСТВО УСРЕДНЯЕМЫХ ЧАСТОТ И СКО
ПЕРИОДОГРАММ ДАНЬЕЛЛА
disp('    СКО периодограмм БАРТЛЕТТА при различной длине фрагмента L')
STD_SB = [L' std(SB)'] % ДЛИНЫ ФРАГМЕНТОВ И СКО ПЕРИОДОГРАММ БАРТЛЕТТА
disp('    СКО периодограммы УЭЛЧА при различной длине фрагмента L и величине
перекрытия Q')
LL = [L(1) L(2) L(3)]; % ДЛИНЫ ФРАГМЕНТОВ
Q = [Q Q Q]; % ВЕЛИЧИНА ПЕРЕКРЫТИЯ
STD_SW = [LL' Q' std(SW)'] % ДЛИНЫ ФРАГМЕНТОВ, ВЕЛИЧИНА ПЕРЕКРЫТИЯ И СКО
ПЕРИОДОГРАММ УЭЛЧА
disp('    СКО оценок СПМ по методу БЛЭКМАНА-ТЬЮКИ при различных окнах')
WINDOW = {name.win}; % ИМЕНА ОКОН (МАССИВ ЯЧЕЕК – cell array)
STD_SBT = std(SBT); % СКО ОЦЕНОК СПМ ПО МЕТОДУ БЛЭКМАНА-ТЬЮКИ
STD_SBT = [WINDOW(1)' STD_SBT(1)'; WINDOW(2)' STD_SBT(2)'; WINDOW(3)'
STD_SBT(3)'] % ИМЕНА ОКОН И СКО ОЦЕНОК СПМ ПО МЕТОДУ БЛЭКМАНА-ТЬЮКИ
disp('%')
disp('%')
disp('% Для вывода ЗНАЧЕНИЙ ДОБРОТНОСТИ нажмите <ENTER>')
pause
disp('%')
format % ВОЗВРАТ К ИСХОДНОМУ ФОРМАТУ ДЛЯ ВЫВОДА ДОБРОТНОСТЕЙ
disp('    ДОБРОТНОСТЬ периодограммы')
Q_S = mean(S).^2./var(S) % ДОБРОТНОСТЬ ПЕРИОДОГРАММЫ
disp('    ДОБРОТНОСТЬ периодограмм ДАНЬЕЛЛА при различном количестве усредняемых
частот K')
Q_SD = mean(SD).^2./var(SD); % ДОБРОТНОСТЬ ПЕРИОДОГРАММ ДАНЬЕЛЛА
Q_SD = [K' Q_SD']
disp('    ДОБРОТНОСТЬ периодограмм БАРТЛЕТТА при различной длине фрагмента L')
Q_SB = mean(SB).^2./var(SB); % ДОБРОТНОСТЬ ПЕРИОДОГРАММ БАРТЛЕТТА
Q_SB = [L' Q_SB']
disp('    ДОБРОТНОСТЬ периодограмм УЭЛЧА при различной длине фрагмента L
и величине перекрытия Q')
Q_SW = mean(SW).^2./var(SW); % ДОБРОТНОСТЬ ПЕРИОДОГРАММ УЭЛЧА
Q_SW = [LL' Q' Q_SW']
disp('    ДОБРОТНОСТЬ оценок СПМ по методу БЛЭКМАНА-ТЬЮКИ при различных окнах')
Q_SBT = real(mean(SBT).^2./var(SBT)); % ДОБРОТНОСТЬ ОЦЕНОК СПМ ПО МЕТОДУ
БЛЭКМАНА-ТЬЮКИ
Q_SBT = [WINDOW(1)' Q_SBT(1)'; WINDOW(2)' Q_SBT(2)'; WINDOW(3)' Q_SBT(3)']
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.12. ПОСТРОЕНИЕ СПЕКТРОГРАММЫ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА СПЕКТРОГРАММЫ дискретного гармонического сигнала
нажмите <ENTER>')

```

```

pause
N = 4000; % ДЛИНА ПОСЛЕДОВАТЕЛЬНОСТИ
n = 0:N-1; % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
x = A1*cos(w1*n')+A2*cos(w2*n'); % ПОСЛЕДОВАТЕЛЬНОСТЬ — СУММА ДВУХ ГАРМОНИК
figure('Name','Harmonic Signal Spectrogram','NumberTitle','off')
spectrogram(x,128,120,128,Fs,'yaxis')
colorbar
xlabel('Time (s)'), ylabel('Frequency (Hz)')
title('Harmonic Signal Spectrogram')
disp('%')
disp('%')
disp('% РАБОТА ЗАВЕРШЕНА')

```

16.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файла для моделирования непараметрических методов спектрального анализа.

Пункты самостоятельного задания включают в себя:

1С. Проверка оценки СПМ на асимптотическую несмещенность и состоятельность для равномерного белого шума.

Выполнить аналогично п. 3 задания в *разд. 16.3*.

2С. Формирование случайной последовательности длины $N \geq 1000$ (входной параметр function-файла) с требуемой АКФ $R_y(m)$:

$$R_y(m) = 1 / (|m| + 1), \quad |m| = 0, \dots, N - 1$$

с последующей фильтрацией последовательности с целью удаления тренда с помощью БИХ-фильтра ФВЧ Золотарева—Кауэра 5-го порядка с нормированной частотой среза $W_{Dn} = 0.4$, максимально допустимым затуханием в ПП $r_p = 0.5$ и минимально допустимым затуханием в ПЗ $r_s = 40$ (см. *разд. 13.1.5*).

Вывести графики:

- требуемой АКФ $R_y(m)$, центрированной относительно $m = N$;
- случайной последовательности с требуемой АКФ, оценки ее АКФ, центрированной относительно $m = N$, и случайной последовательности $y(n)$ на выходе БИХ-фильтра (выходной параметр function-файла).

3С. Расчет немодифицированной периодограммы случайной последовательности $y(n)$ длины N (см. п. 2С) с выводом графика и значений добротности и СКО.

4С. Расчет периодограммы Даньелла случайной последовательности $y(n)$ длины N (см. п. 2С) для произвольного значения K с выводом графика и значений добротности и СКО.

- 5С. Расчет периодограммы Бартлетта случайной последовательности $y(n)$ длины N (см. п. 2С) для произвольной длины фрагмента L с выводом графика и значений добротности и СКО.
- 6С. Расчет периодограммы Уэлча случайной последовательности $y(n)$ длины N (см. п. 2С) для произвольной длины фрагмента L и величины перекрытия Q с окном Хэмминга с выводом графика и значений добротности и СКО.
- 7С. Расчет оценки СПМ по методу Блэкмана—Тьюки случайной последовательности $y(n)$ длины N (см. п. 2С) с заданным окном с выводом графика и значений добротности и СКО.
Выполнить с различными окнами (прямоугольное, Хэмминга и Бартлетта). В GUI Window Design and Analysis Tool определить параметры окна и проанализировать их влияние на указанные показатели качества оценки СПМ.
- 8С. Описание немодифицированной периодограммы последовательности $y(n)$ (см. п. 2С) в виде объекта `spectrum` с выводом графика.
- 9С. Построение спектрограммы гармонического сигнала с изменением мгновенной частоты (см. п. 7С в разд. 7.5).

16.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения каждого пункта задания, включая копируемые из окна **Command Window** результаты вычислений (шрифт Courier New), созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**) и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

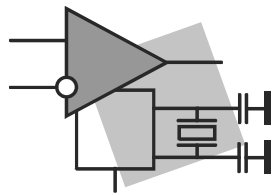
1. Какая исходная информация используется в непараметрических методах?
2. Дайте определение СПМ.
3. Почему на практике доступны для анализа только оценки СПМ?
4. Что такое периодограмма? Как вычисляется оценка СПМ по методу периодограмм?
5. Дайте определение смещенной и несмещенной оценки СПМ для эргодического случайного дискретного сигнала.
6. Дайте определение состоятельной и несостоятельной оценки СПМ для эргодического случайного дискретного сигнала.
7. Дайте определение добротности и СКО оценки СПМ.
8. Как вычисляется оценка СПМ по методу периодограмм Даньелла?
9. За счет чего достигается сглаживание в периодограмме Даньелла по сравнению с немодифицированной периодограммой?

10. Как вычисляется оценка СПМ по методу периодограмм Бартлетта?
11. За счет чего достигается сглаживание в периодограмме Бартлетта по сравнению с периодограммой Даньелла?
12. Как вычисляется оценка СПМ по методу периодограмм Уэлча?
13. За счет чего достигается сглаживание в периодограмме Уэлча по сравнению с периодограммой Бартлетта?
14. Как вычисляется оценка СПМ по методу Блэкмана—Тьюки?
15. За счет чего достигается сглаживание в оценке СПМ по методу Блэкмана—Тьюки по сравнению с периодограммой?
16. С какой целью в методах оценки СПМ используют окна?
17. Перечислите основные этапы моделирования случайной последовательности с требуемой АКФ.
18. С какой целью строится спектрограмма?

16.7. Литература

1. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 5.
2. Оппенгейм А., Шафер Р. Цифровая обработка сигналов. — М.: Техносфера, 2007. — Глава 10.
3. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 12.
4. Марпл С. Л. (мл.). Цифровой спектральный анализ и его приложения / Пер. с англ. — М.: Мир, 1990. — Глава 5.
5. Айфичер Э., Джервис Б. Цифровая обработка сигналов. — М., СПб., Киев: Вильямс, 2004. — Глава 11.
6. Шахтарин Б. И., Ковригин В. А. Методы спектрального оценивания случайных процессов. — 2-е изд. — М.: Горячая Линия — Телеком, 2011. — Глава 2.

ГЛАВА 17



Спектральный анализ: параметрические методы

Цель работы: изучить методы параметрического спектрального анализа (спектрального оценивания) случайных последовательностей и овладеть программными средствами их реализации в MATLAB.

17.1. Краткая теоретическая справка

Параметрические методы оценки СПМ основаны на построении математической модели анализируемого случайного сигнала и определении (*оценке*) параметров модели, при которых обеспечивается наилучшее приближение моделируемого сигнала к анализируемому сигналу по заданному критерию.

По умолчанию будем подразумевать *эргодические* случайные дискретные сигналы — *случайные последовательности* (см. разд. 7.2.2).

Основными преимуществами параметрических методов, по сравнению с непараметрическими, являются:

- ❑ возможность получения более точных оценок СПМ — с отсутствием осцилляций (изрезанности) и искажений, связанных с применением оконных функций (эффекта растекания спектра);
- ❑ информативность оценок СПМ при коротких последовательностях — улучшение различения близко расположенных спектральных составляющих.

Основная сложность параметрических методов заключается в выборе математической модели, адекватной анализируемой последовательности, и оценке ее параметров с целью получения достоверных оценок СПМ — соответствующих структуре истинной СПМ.

В зависимости от вида математической модели различают следующие группы параметрических методов оценивания СПМ:

- ❑ авторегрессионные (АР) — на основе АР-модели (Autoregressive, AR);
- ❑ скользящего среднего (СС) — на основе СС-модели (Moving Average, MA);
- ❑ авторегрессии скользящего среднего (АРСС) — на основе АРСС-модели (Autoregressive Moving Average, ARMA).

17.1.1. АРСС-, АР- и СС-модели

АРСС-модель описывается разностным уравнением¹ (РУ) *БИХ-фильтра*, представленным в виде:

$$y(n) = - \sum_{k=1}^{M-1} a_k y(n-k) + \sum_{i=0}^{N-1} b_i e(n-i), \quad (17.1)$$

где $e(n)$ — входной сигнал (воздействие), в качестве которого используется белый шум, обычно нормальный, с нулевым средним и дисперсией σ^2 ; $y(n)$ — выходной сигнал (реакция) — *моделируемая случайная последовательность*; a_k и b_i — *параметры АРСС-модели*, где $b_0 = 1$; $(M-1)$ — *порядок АРСС-модели* при $(N-1) \leq (M-1)$ (по умолчанию).

АРСС-модели (17.1) соответствует *БИХ-фильтр* с дробно-рациональной передаточной функцией:

$$H(z) = \frac{1 + \sum_{i=1}^{N-1} b_i z^{-i}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}} = \frac{B(z)}{A(z)}. \quad (17.2)$$

Структурная схема АРСС модели представлена на рис. 17.1, а.

АР-модель описывается РУ (17.1) при $b_i = 0$, $i = 1, 2, \dots, (N-1)$:

$$y(n) = - \sum_{k=1}^{M-1} a_k y(n-k) + e(n), \quad (17.3)$$

где a_k — *параметры АР-модели*, $(M-1)$ — *порядок АР-модели*

АР-модели (17.3) соответствует *БИХ-фильтр полюсного вида* ("чисто рекурсивный") с передаточной функцией

$$H(z) = \frac{1}{1 + \sum_{k=1}^{M-1} a_k z^{-k}} = \frac{1}{A(z)}. \quad (17.4)$$

Структурная схема АР-модели представлена на рис. 17.1, б.

СС-модель описывается РУ (17.1) при $a_k = 0$, $k = 1, 2, \dots, (M-1)$:

$$y(n) = e(n) + \sum_{i=1}^{N-1} b_i e(n-i), \quad (17.5)$$

где b_i — *параметры СС-модели*, $(N-1)$ — *порядок СС-модели*.

¹ Математической моделью.

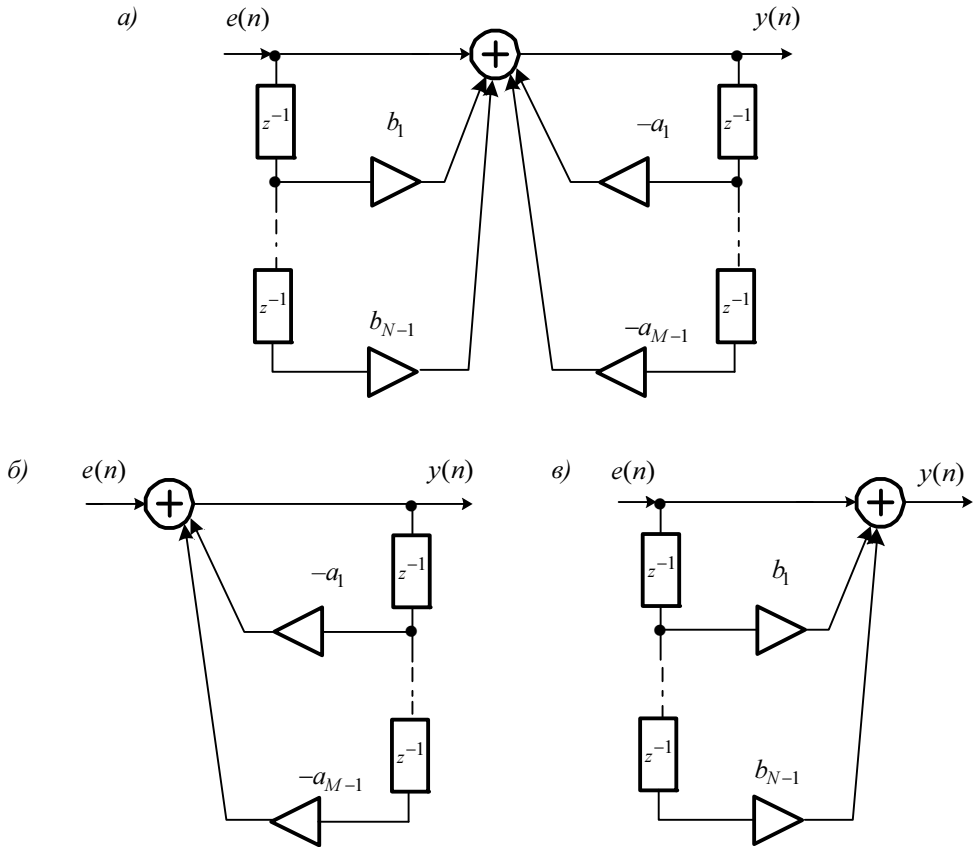


Рис. 17.1. Структурные схемы моделей: а — APCC; б — AP; в — CC

СС-модели (17.5) соответствует *КИХ-фильтр* с передаточной функцией

$$H(z) = 1 + \sum_{i=1}^{N-1} b_i z^{-i} = B(z). \quad (17.6)$$

Структурная схема СС-модели представлена на рис. 17.1, в.

При известных параметрах модели можно путем подстановки $z = e^{j\omega T}$ в передаточную функцию $H(z)$ определить частотную характеристику фильтра $H(e^{j\omega T})$ (см. разд. 8.1.3) и вычислить СПМ $S_y(\omega)$ моделируемой последовательности $y(n)$ по формуле (16.28):

$$S_y(\omega) = \frac{\sigma^2}{f_d} \left| H(e^{j\omega T}) \right|^2, \quad (17.7)$$

в которой учтен множитель $1/f_d$ (см. разд. 16.1.1), а значение $(N_0/2)$, согласно (16.25), равно дисперсии нормального белого шума σ^2 .

Формулу (17.7) можно будет использовать для вычисления оценки СПМ $\hat{S}_x(\omega)$ анализируемой последовательности $x(n)$, если будут определены оценки параметров модели, которые обеспечат ее наилучшее приближение к моделируемой последовательности $y(n)$ по заданному критерию, о чем пойдет речь далее в разд. 17.1.2.

Таким образом, расчет оценки СПМ $\hat{S}_x(\omega)$ включает в себя следующие этапы:

1. Выбор модели (АР, СС или АРСС).
2. Оценка порядка модели.
3. Оценка параметров модели.
4. Расчет оценки СПМ по формуле (17.7).

Выбор модели определяется требованиями конкретной задачи и обычно предполагает сведения о возможном характере СПМ.

АР-модель считается наиболее подходящей для оценки СПМ с острыми пиками, но без глубоких впадин, а СС-модель, наоборот, — с глубокими впадинами, но без острых пиков.

АРСС-модель, как наиболее общая, может использоваться для оценки СПМ с пиками и с впадинами, однако при наличии острых пиков она обеспечивает меньшую точность, чем АР-модель, а при наличии глубоких впадин — меньшую точность, чем СС-модель.

По вычислительным затратам наиболее простой будет АР-модель, а наиболее сложной — АРСС-модель.

В общем случае выбору модели должна предшествовать ее проверка на адекватность анализируемому сигналу. Для этого разработаны специальные критерии [6].

На практике наибольшее распространение получила АР-модель. Это обусловлено ее адекватностью широкому классу реальных сигналов и наименьшими вычислительными затратами. Для оценки параметров АР-модели и расчета оценки СПМ можно воспользоваться стандартными функциями MATLAB.

Далее будем рассматривать расчет оценки СПМ на основе АР-модели (см. рис. 17.1, б). Влияние порядка АР-модели на качество оценки СПМ рассматривается в разд. 17.1.5.

17.1.2. Метод Юла—Уолкера (автокорреляционный) оценки параметров АР-модели

Известны различные методы оценки параметров АР-модели. Рассмотрим один из них, наиболее простой и распространенный — автокорреляционный. В этом методе оценивание параметров АР-модели производится по отсчетам анализируемой последовательности длины L , а моделируемая последовательность $y(n)$ рассмат-

вается как *линейное предсказание вперед*¹ $\hat{x}(n)$ анализируемой последовательности $x(n)$ — линейная комбинация ее предшествующих отсчетов²:

$$y(n) = \hat{x}(n) = - \sum_{k=1}^{M-1} a_k x(n-k). \quad (17.8)$$

Ошибкой линейного предсказания $\varepsilon(n)$ называют разность между истинным значением отсчета $x(n)$ и его оценкой $\hat{x}(n)$:

$$\varepsilon(n) = x(n) - \hat{x}(n), \quad (17.9)$$

которая, с учетом (17.8), равна:

$$\varepsilon(n) = x(n) + \sum_{k=1}^{M-1} a_k x(n-k). \quad (17.10)$$

Отсюда имеем представление анализируемой последовательности $x(n)$ в виде:

$$x(n) = - \sum_{k=1}^{M-1} a_k x(n-k) + \varepsilon(n). \quad (17.11)$$

Сравнивая анализируемую последовательность $x(n)$ (17.11) с моделируемой $y(n)$ (17.3), видим, что в случае, если ошибка линейного предсказания $\varepsilon(n)$ представляет собой нормальный белый шум $e(n)$, параметры линейного предсказания a_k будут совпадать с параметрами АР-модели.

В действительности, однако, ошибка $\varepsilon(n)$ не обязательно является нормальным белым шумом, поэтому, вычислив параметры линейного предсказания a_k , можно получить *оценки* параметров АР-модели $\hat{a}_k$, которые обеспечат наилучшее приближение моделируемой последовательности к анализируемой по заданному критерию.

С учетом (17.8) анализируемую последовательность (17.11) можно представить в виде:

$$x(n) = y(n) + \varepsilon(n)$$

и в качестве критерия наилучшего приближения моделируемой последовательности $y(n)$ к анализируемой $x(n)$ выбрать *минимум среднего квадрата ошибки линейного предсказания* $\varepsilon(n)$ (согласно (17.9), это также критерий наилучшего приближения оценки анализируемой последовательности к ее истинному значению):

$$M \left\{ \varepsilon^2(n) \right\} = M \left\{ [x(n) - y(n)]^2 \right\} \rightarrow \min_a,$$

¹ Forward linear prediction.

² Знак линейной комбинации не имеет значения, т. к. значения a_k могут быть и положительными, и отрицательными.

или, с учетом (17.8):

$$M\{\varepsilon^2(n)\} = M\left\{\left[x(n) + \sum_{k=1}^{M-1} a_k x(n-k)\right]^2\right\} \rightarrow \min_{\mathbf{a}}, \quad (17.12)$$

где $M\{\cdot\}$ — оператор математического ожидания; $\mathbf{a}$ — вектор параметров линейного предсказания a_k , $k = 0, 1, \dots, M-1$.

Для случайной последовательности $x(n)$ длины L средний квадрат ошибки $\varepsilon(n)$ определяется посредством усреднения по длине последовательности (теоретически $L \rightarrow \infty$):

$$M\{\varepsilon^2(n)\} = \frac{1}{L} \sum_{n=0}^{L-1} \varepsilon^2(n) \quad (17.13)$$

или, с учетом (17.10):

$$M\{\varepsilon^2(n)\} = \frac{1}{L} \sum_{n=0}^{L-1} \left[x(n) + \sum_{k=1}^{M-1} a_k x(n-k) \right]^2. \quad (17.14)$$

Вектор $\mathbf{a}$ находится в результате решения оптимизационной задачи — поиска минимума оптимизируемой (целевой) функции (17.14), который достигается при равенстве нулю ее частных производных по всем a_k ; их совокупность можно записать в виде системы линейных алгебраических уравнений (СЛАУ):

$$\frac{2}{L} \sum_{n=0}^{L-1} \left[x(n) + \sum_{k=1}^{M-1} a_k x(n-k) \right] x(n-k) = 0,$$

откуда получаем СЛАУ относительно a_k :

$$\frac{1}{L} \sum_{n=0}^{L-1} \left[\sum_{k=1}^{M-1} a_k x(n-k) \right] x(n-k) = -\frac{1}{L} \sum_{n=0}^{L-1} x(n)x(n-k). \quad (17.15)$$

В правой части (17.15) имеем значения оценки (при конечной длине последовательности) автокорреляционной функции (АКФ):

$$-R_x(k) = -\frac{1}{L} \sum_{n=0}^{L-1} x(n)x(n-k), \quad k = 1, 2, \dots, (M-1), \quad (17.16)$$

а в левой, меняя порядок суммирования, — сумму взвешенных значений оценки АКФ (с весовыми коэффициентами a_k):

$$\begin{aligned} & \sum_{k=1}^{M-1} a_k \left\{ \frac{1}{L} \sum_{n=0}^{L-1} x(n-k)x(n-1) + \frac{1}{L} \sum_{n=0}^{L-1} x(n-k)x(n-2) + \dots \right. \\ & \left. + \frac{1}{L} \sum_{n=0}^{L-1} x(n-k)x[n-(M-1)] \right\} = \sum_{k=1}^{M-1} a_k R_x(k-m), \quad m = 1, 2, \dots, (M-1). \end{aligned}$$

Таким образом, СЛАУ (17.15) можно представить в виде:

$$\sum_{k=1}^{M-1} a_k R_x(k-m) = -R_x(m), \quad m=1, 2, \dots, (M-1), \quad (17.17)$$

или в его матричной записи, известной как *система уравнений Юла—Уолкера* (Yule—Walker), с корреляционной матрицей Теплица (см. табл. 2.1):

$$\begin{bmatrix} R_x(0) & R_x(1) & \dots & R_x(M-1) \\ R_x(1) & R_x(0) & \dots & R_x(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(M-1) & R_x(M-2) & \dots & R_x(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{bmatrix} = - \begin{bmatrix} R_x(1) \\ R_x(2) \\ \vdots \\ R_x(M) \end{bmatrix}. \quad (17.18)$$

Решением СЛАУ (17.18) является вектор параметров линейного предсказания **a** — вектор *оценок параметров АР-модели* $\hat{\mathbf{a}}$.

Обратим внимание на то, что в СЛАУ (17.18) справа и слева имеем значения АКФ, на основе чего для поиска решения был разработан *рекуррентный* алгоритм Левинсона—Дарбина (Levinson—Darbin) [1, 3].

Рассмотренный *автокорреляционный метод* оценки параметров АР-модели, известный также как *метод Юла—Уолкера* (Yule-Walker method), основан на традиционном вычислении оценки АКФ последовательности $x(n)$ по формуле (7.1), при этом неизвестные $(M-1)$ отсчетов последовательности (в ее начале и конце) полагаются равными нулю¹, и оценка АКФ нормируется к длине последовательности L . После решения СЛАУ (17.18) можно на основе (17.14) вычислить средний квадрат ошибки линейного предсказания σ_ε^2 :

$$\sigma_\varepsilon^2 = M \left\{ \varepsilon^2(n) \right\}, \quad (17.19)$$

представляющий собой *оценку дисперсии нормального белого шума* в АР-модели:

$$\sigma_\varepsilon^2 = \hat{\sigma}^2. \quad (17.20)$$

При известных *оценках* параметров АР-модели и *оценке дисперсии нормального белого шума* можно воспользоваться формулой (17.7) для вычисления *оценки СПМ* анализируемой последовательности.

17.1.3. Методы оценки параметров АР-модели

В MATLAB реализованы следующие методы вычисления оценок параметров АР-модели $\hat{\mathbf{a}}$:

- Юла—Уолкера (автокорреляционный);
- Берга;

¹ Для того чтобы убедиться в этом, рекомендуем самостоятельно вычислить АКФ по формуле (7.1) при короткой длине последовательности.

- ковариационный;
- модифицированный ковариационный.

Ограниченный объем теоретической справки не позволяет подробно остановиться на приведенных методах, поэтому ограничимся их краткой характеристикой.

Метод Юла—Уолкера (автокорреляционный) рассмотрен в разд. 17.1.2.

Метод Берга (Burg method) основан на вычислении оценок параметров АР-модели непосредственно по отсчетам последовательности (без вычисления оценки АКФ) с линейным предсказанием вперед (forward) и назад (backward)¹ с минимизацией усредненной суммы средних квадратов ошибок линейного предсказания [4].

*Ковариационный метод*² (Covariance method) основан на вычислении оценки АКФ по известным отсчетам последовательности $x(n)$ (без добавления нулей в начале и в конце), для чего последняя усекается симметрично справа и слева на $(M-1)$ отсчетов, и оценка АКФ нормируется к длине усеченной последовательности $[L-2(M-1)]$. В соответствующей СЛАУ корреляционная матрица не является матрицей Теплица.

Модифицированный ковариационный метод (Modified covariance method) основан на вычислении оценки АКФ для расширенной последовательности, составленной из последовательностей с линейным предсказанием вперед и назад. Оценка АКФ вычисляется так же, как в ковариационном методе, и нормируется к удвоенной длине последовательности $2[L-2(M-1)]$. В соответствующей СЛАУ корреляционная матрица не является матрицей Теплица.

Методы Юла—Уолкера и Берга гарантируют устойчивость БИХ-фильтра, соответствующего АР-модели, а ковариационный и модифицированный ковариационный методы требуют проверки фильтра на устойчивость.

В MATLAB для вычисления оценок параметров АР-модели $\hat{\mathbf{a}}$ используется функция:

$[a, D] = \langle \text{имя функции} \rangle (x, p)$

где $\langle \text{имя функции} \rangle$ — функция, определяющая метод оценки параметров АР-модели:

- `aryule` — Юла—Уолкера;
- `arburg` — Берга;
- `arcov` — ковариационный;
- `armcov` — модифицированный ковариационный.

x — вектор отсчетов анализируемой последовательности длины L ; p — порядок АР-модели $(M-1)$; a — вектор оценок параметров АР-модели $\hat{\mathbf{a}}$; D — средний

¹ В линейном предсказании назад значение текущего отсчета определяется как взвешенная сумма $(M-1)$ последующих отсчетов.

² Название метода не связано с автоковариационной функцией.

квадрат ошибки линейного предсказания σ_e^2 (17.19) — оценка дисперсии нормального белого шума в АР-модели $\hat{\sigma}^2$ (17.20).

17.1.4. Методы оценки СПМ

В MATLAB оценка СПМ $\hat{S}(f)$ (17.7) рассчитывается *после* вычисления оценок параметров АР-модели $\hat{\mathbf{a}}$ с помощью функции¹:

`[S,f] = <имя функции>(x,p,Nfft,Fs,'range')`

где `<имя функции>` — функция, определяющая метод расчета оценки СПМ, соответствующая функции вычисления оценок параметров АР-модели (см. разд. 17.1.3):

□ `pyulear` — Юла—Уолкера;

□ `pburg` — Берга;

□ `pcov` — ковариационный;

□ `pmcov` — модифицированный ковариационный.

$\mathbf{x}$ — вектор отсчетов анализируемой последовательности длины L ; p — порядок АР-модели $(M-1)$; $\mathbf{f}$, $\mathbf{s}$ — векторы значений частот f (Гц) и оценки СПМ $\hat{S}(f)$ (Вт/Гц) (17.7).

Параметры `Nfft`, `Fs` и `'range'` определены ранее для функции `periodogram` (см. разд. 16.1.1).

График оценки СПМ $\hat{S}(f)$ (Вт/Гц) выводится с помощью функции:

`plot(f,s)`

а оценки СПМ $\hat{S}(f)$ (дБ/Гц):

$$\hat{S}(f) \text{ (дБ/Гц)} = 10 \lg \hat{S}(f) \quad (17.21)$$

с помощью функции:

`<имя функции>(x,p,Nfft,Fs,'range')`

где `<имя функции>` — функция, соответствующая методу расчета оценки СПМ.

Параметрические оценки СПМ могут описываться в виде объектов `spectrum`:

`hs = spectrum.estmethod`

где `estmethod` — название метода:

□ `yulear` — Юла—Уолкера;

□ `burg` — Берга;

□ `cov` — ковариационный;

□ `mcov` — модифицированный ковариационный.

¹ Таким образом, для вычисления оценки СПМ используются две функции — функция вычисления оценок параметров АР-модели и функция вычисления собственно оценки СПМ.

H_s — имя объекта.

Список свойств объекта H_s включает в себя:

☐ `EstimationMethod` — метод оценки СПМ;

☐ `Order` — порядок АР-модели ($M - 1$).

После создания объекта `spectrum` выводится график оценки СПМ $\hat{S}(f)$ (дБ/Гц) с помощью функции:

`psd(Hs, x, 'Fs', Fs)`

Приведем некоторые особенности методов оценки СПМ на основе АР-модели.

Метод Юла—Уолкера обычно применяют при анализе длинных последовательностей. Для коротких последовательностей завышенный порядок АР-модели может сопровождаться смещением пиков оценки СПМ и их расщеплением (появлением ложных пиков).

Метод Берга дает удовлетворительные результаты и при анализе коротких последовательностей, однако завышенный порядок АР-модели может также приводить к смещению и расщеплению пиков оценки СПМ.

Ковариационный и модифицированный ковариационный методы обеспечивают более высокую точность при анализе коротких последовательностей, по сравнению с методом Юла—Уолкера с тем же порядком АР-модели. Модифицированный ковариационный метод, как правило, не приводит к расщеплению пиков оценки СПМ и, по сравнению с ковариационным методом, обеспечивает их меньшее смещение.

17.1.5. Оценка порядка АР-модели

Порядок АР-модели ($M - 1$) обычно заранее неизвестен, поэтому приходится испытывать его различные значения. Заниженный порядок АР-модели может привести к сглаживанию оценки СПМ (неразличимости малых пиков), а завышенный — к появлению ложных пиков, наряду с возрастанием трудоемкости оценки параметров при большом порядке СЛАУ и риском неустойчивости ее решения при плохо обусловленной матрице коэффициентов¹. Поэтому при выборе порядка АР-модели приходится искать компромисс между уменьшением среднего квадрата ошибки линейного предсказания и увеличением порядка модели.

Для оптимальной оценки порядка АР-модели анализируемой последовательности можно воспользоваться *информационными критериями* [4]. Одним из простейших является информационный критерий Байеса (BIC)²:

$$\text{BIC}(L, p, \sigma_e^2) = L \ln \sigma_e^2 + p \ln L, \quad (17.22)$$

¹ Оценка обусловленности матрицы рассматривается в разд. 3.8.7 [2]. В плохо обусловленной матрице малые погрешности ее элементов могут привести к существенному изменению и недостоверности решения СЛАУ.

² При использовании других информационных критериев результаты могут отличаться, но незначительно.

где L — длина последовательности; $p = (M - 1)$ — порядок АР-модели; σ_ε^2 — средний квадрат ошибки линейного предсказания (17.19).

Задача определения оптимального порядка p_{opt} сводится к поиску порядка p , при котором достигается минимум функции (17.22):

$$p_{\text{opt}} \rightarrow \min_p \text{BIC}(L, p, \sigma_\varepsilon^2), \quad p \in [p_{\min}; p_{\max}]. \quad (17.23)$$

Минимальный $p_{\min}$ и максимальный $p_{\max}$ порядки обычно выбираются из эмпирических соображений, и для каждого порядка p рассчитывается σ_ε^2 .

17.1.6. Сравнение оценок СПМ с истинной СПМ

Для сравнения параметрических оценок СПМ, рассчитанных различными методами, основным показателем качества оценки СПМ является среднеквадратическая ошибка RMSE¹ между истинной СПМ $S(\omega)$ и ее оценкой $\hat{S}(\omega)$ при длине последовательности L :

$$\text{RMSE} = \sqrt{\frac{1}{L} \sum_{k=0}^{L-1} |S(\omega_k) - \hat{S}(\omega_k)|^2}, \quad (17.24)$$

где ω_k — значения частот в L равноотстоящих точках СПМ на периоде.

На практике истинная СПМ $S(\omega)$ неизвестна, поэтому показатель RMSE используется на этапе *моделирования* для сравнительного анализа различных оценок СПМ. В этом случае *истинная* СПМ $S(\omega)$ определяется на основе (17.7) для моделируемой последовательности при заданных параметрах АР-модели a_k . Далее, полагая, что анализируемая последовательность совпадает с моделируемой, определяют оценки параметров АР-модели $\hat{a}_k$ с использованием соответствующей функции MATLAB, а затем — оценку СПМ $\hat{S}(\omega)$.

Чем меньше значение RMSE, тем лучше используемая оценка СПМ.

17.2. Содержание лабораторной работы

Содержание работы связано с моделированием методов параметрического спектрального анализа случайных последовательностей на основе АР-модели, проверкой устойчивости БИХ-фильтров, соответствующих АР-моделям, и определением показателя качества оценок СПМ программными средствами MATLAB.

¹ Root Mean Squared Error. По существу это сравнение на основе нормы $\|\mathbf{x}\|_2$ (см. разд. 2.1.5).

17.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файла с именем `lr_17`, который хранится на прилагаемом компакт-диске в папке `LAB_DSP\LAB_17`.

Перед выполнением работы необходимо сохранить путь к папке `LAB_17` по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 17.1 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$.

На прилагаемом компакт-диске в папке `Tables\Tables_17` хранятся табл. 17.1 исходных данных и пример ее заполнения для $N_{бр} = 1$.

Таблица 17.1. Таблица исходных данных

Переменная	Назначение	Значение			Идентификатор
$N_{бр}$	Номер бригады	$N_{бр}$			Nb =
L	Длина воздействия	$L = 2^{10+N_{бр} \bmod 3}$			L =
f_d	Частота дискретизации	$f_d = 1000(N_{бр} \bmod 5 + 1)$			Fs =
a	Заданные (истинные) параметры АР-модели	Номера бригад $N_{бр}$			Вектор $a = [1 \ \dots]$
		1—10	11—20	21—30	
a_0		1	1	1	
a_1		−0,86	−0,10	0,50	
a_2		0,54	0,20	0,17	
a_3		−0,30	0,20	0,30	
a_4		−0,17	0,10	0,10	
a_5		0,22	0,30	0,10	
a_6		−0,10	—	−0,10	
a_7		—	—	−0,50	

Задание на лабораторную работу заключается в моделировании методов параметрического спектрального анализа и включает в себя следующие пункты:

1. Моделирование случайной последовательности на основе АР-модели.
- Моделируемую последовательность $y(n)$ (идентификатор y) длины L сформировать на основе АР-модели (17.3) с параметрами a_k , заданными вектором **a**.

В MATLAB параметры АР-модели — элементы вектора **a** — вводятся с добавлением элемента $a_0 = 1$ (см. табл. 17.1).

В качестве входного сигнала $e(n)$ (идентификатор **e**) в РУ (17.3) в АР-модели выбрать нормальный белый шум с нулевым средним значением и единичной дисперсией.

Вывести график моделируемой последовательности $y(n)$ с помощью функции `plot`.

Пояснить:

- с какой целью используют АР-модель;
- чему равен порядок АР-модели;
- какие элементы вектора **a** относят к параметрам АР-модели.

2. Вычисление истинной СПМ (PSD¹) моделируемой последовательности.

Вычислить истинную СПМ $S(f)$ (идентификатор **s**) последовательности $y(n)$ (см. п. 1) на основе (17.7) и вывести ее график на периоде.

Пояснить:

- чему равен период СПМ;
- в каком случае достаточно вычислить СПМ в основной полосе частот;
- вид СПМ (наличие пиков и впадин) в основной полосе частот.

3. Оценка оптимального порядка АР-модели анализируемой последовательности $x(n)$ на основе критерия Байеса.

Полагая, что анализируемая последовательность $x(n)$ (идентификатор **x**) совпадает с моделируемой $y(n)$, сформированной в п. 1:

$$x(n) = y(n), \quad (17.25)$$

задать значения порядка АР-модели p (идентификатор **p**) в диапазоне $p \in [1; 3(M-1)]$, где $(M-1)$ — порядок АР-модели с заданными параметрами.

При заданных значениях p вычислить соответствующие значения среднего квадрата ошибки линейного предсказания σ_{ε}^2 (17.19) (идентификатор **d**), используя метод Юла—Уолкера, и значения критерия Байеса BIC (17.22) (идентификатор **bic**).

Вывести графики следующих зависимостей от порядка АР-модели p :

- среднего квадрата ошибок линейного предсказания σ_{ε}^2 ;
- значений критерия Байеса BIC.

¹ Power Spectral Density — спектральная плотность мощности.

Определить оптимальный порядок АР-модели p_{opt} (идентификатор `p_opt`) по графику значений критерия Байеса ВИС (при необходимости используя кнопку **Zoom in** на панели инструментов) и ввести значение p_{opt} с клавиатуры.

Пояснить:

- какая последовательность выбрана в качестве анализируемой;
- как определяется оптимальный порядок АР-модели по графику;
- зависимость среднего квадрата ошибки линейного предсказания от порядка модели.

4. Вычисление оценок параметров АР-модели.

Для последовательности $x(n)$ (17.25) вычислить и вывести оценки параметров $\hat{a}_k$ АР-модели следующими методами:

- Юла—Уолкера (идентификатор `aYW`);
- Берга (идентификатор `aBURG`);
- ковариационным (идентификатор `aCOV`);
- модифицированным ковариационным (идентификатор `aMCOV`).

В *MATLAB* оценки параметров АР-модели — элементы вектора $\hat{\mathbf{a}}$ — выводятся с добавлением элемента $\hat{a}_0 = 1$.

Вывести значения среднего квадрата ошибок линейного предсказания (идентификаторы `D_aYW`, `D_aBURG`, `D_aCOV` и `D_aMCOV`).

Вывести значение оценки дисперсии нормального белого шума (идентификатор `D`) — входного сигнала АР-модели — с помощью функции `var`.

Пояснить:

- что такое оценки параметров АР-модели;
- с чем сравниваются полученные оценки параметров;
- какие выводы следуют по результатам сравнения;
- какова связь между средним квадратом ошибки линейного предсказания и дисперсией нормального белого шума.

5. Проверка устойчивости БИХ-фильтра.

На основе оценок параметров АР-моделей, полученных разными методами, построить карты нулей и полюсов БИХ-фильтров, соответствующих АР-моделям.

Пояснить:

- являются ли БИХ-фильтры устойчивыми;
- по какому критерию проверяется устойчивость;
- какие методы гарантируют устойчивость БИХ-фильтра.

6. Вычисление оценок СПМ.

На основе полученных оценок параметров АР-модели вычислить оценки СПМ на периоде следующими методами:

- Юла—Уолкера — $\hat{S}_{YW}(f)$ (идентификатор `SYW`);
- Берга — $\hat{S}_{BURG}(f)$ (идентификатор `SBURG`);
- ковариационным — $\hat{S}_{COV}(f)$ (идентификатор `SCOV`);
- модифицированным ковариационным — $\hat{S}_{MCOV}(f)$ (идентификатор `SMCOV`).

Вывести графики оценок СПМ, вычисленных различными методами.

Сделать вывод о наличии пиков и впадин по графикам оценок СПМ в основной полосе частот.

7. Сравнение оценок СПМ с истинной СПМ.

Вывести на одних координатных осях линиями разных цветов (с размещением легенды) графики оценок СПМ, вычисленных различными методами, и истинной СПМ.

Используя кнопку **Zoom in** на панели инструментов, сравнить графики оценок СПМ друг с другом и с истинной СПМ.

Пояснить результаты сравнения оценок СПМ с истинной СПМ (появление ложных пиков и впадин, смещение пиков и впадин, изменение значений СПМ, соответствующих частотам пиков и впадин).

8. Вычисление значений RMSE (17.24) для оценок СПМ, вычисленных следующими методами:

- Юла—Уолкера (идентификатор `RMSE_YW`);
- Берга (идентификатор `RMSE_BURG`);
- ковариационным (идентификатор `RMSE_COV`);
- модифицированным ковариационным (идентификатор `RMSE_MCOV`).

Пояснить результаты сравнения оценок СПМ (близость к истинной СПМ).

9. Исследование влияния порядка модели на оценку СПМ.

Вывести на одних координатных осях линиями разных цветов (с размещением легенды) графики истинной СПМ и оценок СПМ по методу Юла—Уолкера при различных порядках АР-модели¹: заниженном $p_{\text{low}} = \text{int}(p_{\text{opt}}/2)$ (идентификатор `p_low`) и завышенном $p_{\text{high}} = 3p_{\text{opt}}$ (идентификатор `p_high`), где функция `int` соответствует округлению до ближайшего целого в большую сторону.

¹ При изменении порядка АР-модели автоматическая определяется оценка ее новых параметров, а затем — оценка СПМ.

Пояснить, к каким изменениям в оценке СПМ по методу Юла—Уолкера приводит занижение и завышение порядка АР-модели относительно его оптимального значения (сглаживание оценок СПМ, расщепление пиков, смещение пиков и впадин, изменение значений СПМ, соответствующих частотам пиков и впадин).

10. Исследование влияния длины последовательности на оценку СПМ.

Вывести на одних координатных осях линиями разных цветов (с размещением легенды) графики истинной СПМ и оценок СПМ по методу Юла—Уолкера при различных длинах анализируемой последовательности: L и $100L$.

Пояснить, к каким изменениям в оценке СПМ по методу Юла—Уолкера приводит существенное возрастание длины последовательности (смещение пиков, изменение значений СПМ, соответствующих частотам пиков и впадин).

17.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должна быть представлена табл. 17.1 исходных данных для своего номера бригады $N_{бр}$.

Для *запуска* лабораторной работы необходимо обратиться к script-файлу `lr_17` по его имени:

```
>> lr_17
```

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш <Ctrl>+<Break>.

При выполнении script-файла текущие окна с графиками *не закрывать*.

Листинг script-файла `lr_17` имеет вид:

```
>> type lr_17
script
clc
clear
disp('% ЛР №17. ПАРАМЕТРИЧЕСКИЙ СПЕКТРАЛЬНЫЙ АНАЛИЗ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
L = input('L = ');             % ДЛИНА ПОСЛЕДОВАТЕЛЬНОСТИ
Fs = input('Fs = ');           % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
a = input('a = ');             % ВЕКТОР ПАРАМЕТРОВ АР-МОДЕЛИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
```

```

disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.1. МОДЕЛИРОВАНИЕ СЛУЧАЙНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ НА ОСНОВЕ АР-МОДЕЛИ')
n = 0:(L-1);           % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
e = randn(1,L);        % ВОЗДЕЙСТВИЕ — НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ ДЛИНЫ L
y = filter(1,a,e);      % МОДЕЛИРУЕМАЯ СЛУЧАЙНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ДЛИНЫ L
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА МОДЕЛИРУЕМОЙ последовательности нажмите <ENTER>')
pause
figure('Name','AR-sequence and True PSD','NumberTitle','off')
subplot(2,1,1), plot(n,y), grid, xlabel('n'), ylabel('y(n)')
title('AR-sequence')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ ИСТИННОЙ СПМ МОДЕЛИРУЕМОЙ ПОСЛЕДОВАТЕЛЬНОСТИ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКА ИСТИННОЙ СПМ моделируемой последовательности нажмите <ENTER>')
pause
f = 0:Fs/(L-1):Fs;     % ВЕКТОР ЧАСТОТ (Гц)
H = freqz(1,a,f,Fs);   % КОМПЛЕКСНАЯ ЧАСТОТНАЯ ХАРАКТЕРИСТИКА
S = (1/Fs)*abs(H).^2;   % ИСТИННАЯ СПМ МОДЕЛИРУЕМОЙ ПОСЛЕДОВАТЕЛЬНОСТИ
subplot(2,1,2), plot(f,S), grid, xlabel('f (Hz)'), ylabel('S(f)')
title('True PSD')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. ОЦЕНКА ОПТИМАЛЬНОГО ПОРЯДКА АР-МОДЕЛИ АНАЛИЗИРУЕМОЙ ПОСЛЕДОВАТЕЛЬНОСТИ')
disp('%')
disp('%')

```

```

x = y; % АНАЛИЗИРУЕМАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ПОЛАГАЕТСЯ РАВНОЙ МОДЕЛИРУЕМОЙ
pmin = 1; pmax = 3*(length(a)-1); % МИНИМАЛЬНОЕ И МАКСИМАЛЬНОЕ ЗНАЧЕНИЯ
ПОРЯДКА
p = [pmin:pmax]; % ВЕКТОР ЗНАЧЕНИЙ ПОРЯДКА АР-МОДЕЛИ
for i = pmin:pmax
    [aYW D] = aryule(x,p(i)); % ПАРАМЕТРЫ АР-МОДЕЛИ И СРЕДНИЙ КВАДРАТ
ОШИБКИ ЛИНЕЙНОГО ПРЕДСКАЗАНИЯ
    BIC(i) = L*log(D)+p(i)*log(L); % ЗНАЧЕНИЕ КРИТЕРИЯ БАЙЕСА
    variance(i) = D; % СРЕДНИЙ КВАДРАТ ОШИБКИ ЛИНЕЙНОГО
ПРЕДСКАЗАНИЯ – ОЦЕНКА ДИСПЕРСИИ НОРМАЛЬНОГО БЕЛОГО ШУМА В АР-МОДЕЛИ
end
[BIC_min p_opt] = min(BIC); % МИНИМАЛЬНОЕ ЗНАЧЕНИЕ КРИТЕРИЯ БАЙЕСА И
ОПТИМАЛЬНЫЙ ПОРЯДОК МОДЕЛИ
disp('% Для вывода ГРАФИКОВ зависимостей СРЕДНЕГО КВАДРАТА ОШИБКИ ЛИНЕЙНОГО
ПРЕДСКАЗАНИЯ')
disp('% и ЗНАЧЕНИЙ КРИТЕРИЯ БАЙЕСА от ПОРЯДКА МОДЕЛИ нажмите <ENTER>')
pause
figure('Name','Mean Square of the Linear Prediction Error and Bayesian
Information Criterion','NumberTitle','off')
subplot(2,1,1), plot(p,variance,'Linewidth',2), grid
xlabel('p'), ylabel('D'), title('Mean Square of the Linear Prediction Error')
subplot(2,1,2), plot(p,BIC,'r','Linewidth',2), grid
xlabel('p'), ylabel('BIC')
title('Bayesian Information Criterion')
disp('%')
disp('%')
disp('% Введите значение ОПТИМАЛЬНОГО ПОРЯДКА АР-модели')
DATA=0;
while DATA==0
    disp('%')
    p_opt = input(' p_opt = '); % ОПТИМАЛЬНЫЙ ПОРЯДОК АР-МОДЕЛИ
    disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
    disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
    disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
    DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('%')
disp('% п.4. ВЫЧИСЛЕНИЕ ОЦЕНОК ПАРАМЕТРОВ АР-МОДЕЛИ')
disp('%')
disp('%')
[aYW DYW] = aryule(x,p_opt); % ОЦЕНКА ПАРАМЕТРОВ АР-МОДЕЛИ ПО МЕТОДУ ЮЛА-
УОЛКЕРА
[aBURG DBURG] = arburg(x,p_opt); % ОЦЕНКА ПАРАМЕТРОВ АР-МОДЕЛИ ПО МЕТОДУ БЕРГА

```

```

[aCOV DCOV] = arcov(x,p_opt);      % ОЦЕНКА ПАРАМЕТРОВ АР-МОДЕЛИ КОВАРИАЦИОННЫМ
МЕТОДОМ
[aMCOV DMCOV] = armcov(x,p_opt); % ОЦЕНКА ПАРАМЕТРОВ АР-МОДЕЛИ МОДИФИЦИРОВАННЫМ
КОВАРИАЦИОННЫМ МЕТОДОМ
D = var(e);      % ОЦЕНКА ДИСПЕРСИИ НОРМАЛЬНОГО БЕЛОГО ШУМА ДЛИНЫ L
disp('% Для вывода ИСТИННЫХ ПАРАМЕТРОВ АР-МОДЕЛИ и их ОЦЕНОК нажмите <ENTER>')
pause
disp('%')
disp(' ИСТИННЫЕ ПАРАМЕТРЫ');
disp([' a =      [' num2str(a),'] ']);
disp('%')
disp(' МЕТОД КЮЛА-УОЛКЕРА');
disp('%')
disp([' aYW =    [' num2str(aYW),'] ']);
disp('%')
disp(' МЕТОД БЕРГА');
disp('%')
disp([' aBURG =  [' num2str(aBURG),'] ']);
disp('%')
disp(' КОВАРИАЦИОННЫЙ МЕТОД');
disp('%')
disp([' aCOV =   [' num2str(aCOV),'] ']);
disp('%')
disp(' МОДИФИЦИРОВАННЫЙ КОВАРИАЦИОННЫЙ МЕТОД');
disp('%')
disp([' aMCOV =  [' num2str(aMCOV) ,'] ']);
disp('%')
disp('%')
disp('% Для вывода СРЕДНЕГО КВАДРАТА ОШИБОК ЛИНЕЙНОГО ПРЕДСКАЗАНИЯ нажмите
<ENTER>')
pause
disp('%')
disp([' МЕТОД КЮЛА-УОЛКЕРА:                DYW = ' num2str(DYW)]);
disp('%')
disp([' МЕТОД БЕРГА:                DBURG = ' num2str(DBURG)]);
disp('%')
disp([' КОВАРИАЦИОННЫЙ МЕТОД:        DCOV = ' num2str(DCOV)]);
disp('%')
disp([' МОДИФИЦИРОВАННЫЙ КОВАРИАЦИОННЫЙ МЕТОД:  DMCOV = ' num2str(DMCOV)]);
disp('%')
disp('%')
disp('% Для вывода ОЦЕНКИ ДИСПЕРСИИ НОРМАЛЬНОГО БЕЛОГО ШУМА нажмите <ENTER>')
pause
disp('%')
disp([' ОЦЕНКА ДИСПЕРСИИ НОРМАЛЬНОГО БЕЛОГО ШУМА: D = ' num2str(D)]);
disp('%')
disp('%')

```

```

disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ПРОВЕРКА УСТОЙЧИВОСТИ БИХ-ФИЛЬТРА')
disp('%')
disp('%')
disp('% Для вывода КАРТ НУЛЕЙ И ПОЛЮСОВ нажмите <ENTER>')
pause
figure('Name','Z-plane zero-pole plots','NumberTitle','off')
subplot(2,2,1), zplane(1,aYW), title ('Yule-Walker method'), grid
xlabel('Re'), ylabel('jIm')
subplot(2,2,2), zplane(1,aBURG), title ('Burg method'), grid
xlabel('Re'), ylabel('jIm')
subplot(2,2,3), zplane(1,aCOV), title ('Covariance method'),grid
xlabel('Re'), ylabel('jIm')
subplot(2,2,4), zplane(1,aMCOV),title ('Modified Covariance method')
xlabel('Re'), ylabel('jIm'), grid
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ВЫЧИСЛЕНИЕ ОЦЕНОК СПМ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ОЦЕНОК СПМ нажмите <ENTER>')
pause
[SYW,f] = pyulear(x,p_opt,L,Fs,'twosided'); % ОЦЕНКА СПМ ПО МЕТОДУ ЮЛА-ВОЛКЕРА
[SBURG,f] = pburg(x,p_opt,L,Fs,'twosided'); % ОЦЕНКА СПМ ПО МЕТОДУ БЕРГА
[SCOV,f] = pcov(x,p_opt,L,Fs,'twosided'); % ОЦЕНКА СПМ КОВАРИАЦИОННЫМ
МЕТОДОМ
[SMCOV,f] = pmcov(x,p_opt,L,Fs,'twosided'); % ОЦЕНКА СПМ МОДИФИЦИРОВАННЫМ
КОВАРИАЦИОННЫМ МЕТОДОМ
figure('Name','PSD Estimates','NumberTitle','off')
subplot(4,1,1), plot(f,SYW,'Linewidth',2), grid
xlabel('f (Hz)'), ylabel('SYW(f)'), title(' PSD estimate using Yule-Walker
method')
subplot(4,1,2), plot(f,SBURG,'Linewidth',2), grid
xlabel('f (Hz)'), ylabel('SBURG(f)'), title('PSD estimate using Burg method')
subplot(4,1,3), plot(f,SCOV,'Linewidth',2), grid
xlabel('f (Hz)'), ylabel('SCOV(f)'), title('PSD estimate using Covariance
method')
subplot(4,1,4), plot(f,SMCOV,'Linewidth',2), grid
xlabel('f (Hz)'), ylabel('SMCOV(f)')
title('PSD estimate using Modified Covariance method')
disp('%')

```

```

disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. СРАВНЕНИЕ ОЦЕНОК СПМ С ИСТИННОЙ СПМ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ОЦЕНОК СПМ и ИСТИННОЙ СПМ нажмите <ENTER>')
pause
figure('Name','True PSD and Different PSD Estimates', 'NumberTitle','off')
plot(f,S,'Linewidth',2), xlabel('f (Hz)'), ylabel('S(f)'), grid
hold on
plot(f,SYW,'m','Linewidth',2), grid
plot(f,SBURG,'r','Linewidth',2), grid
plot(f,SCOV,'k','Linewidth',2), grid
plot(f,SMCOV,'g','Linewidth',2), grid
legend('True PSD','PSD estimate Yule-Walker method', 'PSD estimate Burg
method', 'PSD estimate Covariance method', 'PSD estimate Modified Covariance
method', 0);
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.8. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ RMSE')
RMSE_YW = sqrt((1/L).*sum((S'-SYW).^2)); % RMSE ДЛЯ ОЦЕНКИ СПМ ПО МЕТОДУ
ЮЛА-УОЛКЕРА
RMSE_BURG = sqrt((1/L).*sum((S'-SBURG).^2)); % RMSE ДЛЯ ОЦЕНКИ СПМ ПО МЕТОДУ
БЕРГА
RMSE_COV = sqrt((1/L).*sum((S'-SCOV).^2)); % RMSE ДЛЯ ОЦЕНКИ СПМ
КОВАРИАЦИОННЫМ МЕТОДОМ
RMSE_MCOV = sqrt((1/L).*sum((S'-SMCOV).^2)); % RMSE ДЛЯ ОЦЕНКИ СПМ
МОДИФИЦИРОВАННЫМ КОВАРИАЦИОННЫМ МЕТОДОМ
disp('%')
disp('%')
disp('% Для вывода ЗНАЧЕНИЙ RMSE ОЦЕНОК СПМ нажмите <ENTER>')
pause
disp('%')
disp([' МЕТОД ЮЛА-УОЛКЕРА: RMSE = ' num2str(RMSE_YW )]);
disp('%')
disp([' МЕТОД БЕРГА: RMSE = ' num2str(RMSE_BURG )]);
disp('%')
disp([' КОВАРИАЦИОННЫЙ МЕТОД: RMSE = ' num2str(RMSE_COV )]);
disp('%')
disp([' МОДИФИЦИРОВАННЫЙ КОВАРИАЦИОННЫЙ МЕТОД: RMSE = ' num2str(RMSE_MCOV )]);
disp('%')

```

```

disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.9. ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПОРЯДКА АР-МОДЕЛИ НА ОЦЕНКУ СПМ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИСТИННОЙ СПМ и ОЦЕНОК СПМ по методу ЮЛА-УОЛКЕРА
нажмите <ENTER>')
pause
p_low = round(p_opt/2); % ЗАНИЖЕННЫЙ ПОРЯДОК АР-МОДЕЛИ
p_high = 3*p_opt; % ЗАВЫШЕННЫЙ ПОРЯДОК АР-МОДЕЛИ
[SYW_low,f] = pyulear(x,p_low,L,Fs,'twosided'); % ОЦЕНКА СПМ ПО МЕТОДУ
ЮЛА-УОЛКЕРА С ЗАНИЖЕННЫМ ПОРЯДКОМ МОДЕЛИ
[SYW_high,f] = pyulear(x,p_high,L,Fs,'twosided'); % ОЦЕНКА СПМ ПО МЕТОДУ
ЮЛА-УОЛКЕРА С ЗАВЫШЕННЫМ ПОРЯДКОМ МОДЕЛИ
figure('Name','Different AR Model Orders for Yule-Walker Method',
'NumberTitle','off')
plot(f,S,'Linewidth',2), xlabel('f (Hz)'), grid, hold on
plot(f,SYW_low,'r','Linewidth',2), grid
plot(f,SYW_high,'k','Linewidth',2), grid
legend(['True PSD: p opt = ' num2str(p_opt)], ['PSD estimate Yule-Walker
method: p = ' num2str(p_low)], ['PSD estimate Yule-Walker method: p = '
num2str(p_high)], 0);
title('PSD estimates using Yule-Walker method for different model orders')
disp('%')
disp('%')
disp('% п.10. ИССЛЕДОВАНИЕ ВЛИЯНИЯ ДЛИНЫ ПОСЛЕДОВАТЕЛЬНОСТИ НА ОЦЕНКУ СПМ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИСТИННОЙ СПМ и ОЦЕНОК СПМ по методу ЮЛА-УОЛКЕРА
нажмите <ENTER>')
pause
e1 = randn(1,100*L); % ВОЗДЕЙСТВИЕ — НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ ДЛИНЫ 100*L
y1 = filter(1,a,e1); % МОДЕЛИРУЕМАЯ СЛУЧАЙНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ
ДЛИНЫ 100*L
x1 = y1; % АНАЛИЗИРУЕМАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ПОЛАГАЕТСЯ
РАВНОЙ МОДЕЛИРУЕМОЙ
[SYW_1,f] = pyulear(x,p_opt,L,Fs,'twosided'); % ОЦЕНКА СПМ ПО МЕТОДУ ЮЛА-
УОЛКЕРА ДЛЯ ПОСЛЕДОВАТЕЛЬНОСТИ ДЛИНЫ L
[SYW_2,f] = pyulear(x1,p_opt,L,Fs,'twosided'); % ОЦЕНКА СПМ ПО МЕТОДУ ЮЛА-
УОЛКЕРА ДЛЯ ПОСЛЕДОВАТЕЛЬНОСТИ ДЛИНЫ 100*L
figure('Name','Different Sequence Lengths for Yule-Walker Method',
'NumberTitle','off')
plot(f,S,'Linewidth',2), xlabel('f (Hz)'), grid, hold on
plot(f,SYW_1,'r','Linewidth',2), grid
plot(f,SYW_2,'k','Linewidth',2), grid

```



```

legend(['True PSD: L = ' num2str(L)], [' PSD estimate Yule-Walker method: L = '
num2str(L)], [' PSD estimate Yule-Walker method: L = ' num2str(100*L)], 0);
title('PSD estimates using Yule-Walker method for different lengths')
disp('%')
disp('%')
disp('% РАБОТА ЗАВЕРШЕНА')

```

17.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файлов для моделирования методов параметрического спектрального анализа.

Проверить выполнение созданных function-файлов в режиме прямых вычислений с исходными данными для своего номера бригады $N_{бр}$, а также для длины случайной последовательности $L = 10\,000$ и вектора **a** параметров АР-модели (с добавлением $a_0 = 1$): $\mathbf{a} = [1 \ -0.20 \ 0.20 \ -0.40 \ 0.10 \ -0.30 \ 0.20 \ -0.20]$.

Пункты самостоятельного задания включают в себя:

1С. Моделирование случайной последовательности длины L на основе АР-модели с параметрами, заданными вектором **a**, и вычисление ее истинной СПМ (Вт/Гц).

В качестве входных параметров function-файла использовать длину последовательности L и вектор **a**, а выходных — моделируемую случайную последовательность и истинную СПМ.

2С. Оценка оптимального порядка АР-модели на основе критерия Байеса.

Выбрать в качестве анализируемой последовательность, сформированную в п. 1С, и определить графически оптимальный порядок модели по критерию Байеса (выходной параметр function-файла).

3С. Вычисление оценки СПМ различными методами.

Выбрать в качестве анализируемой последовательность, сформированную в п. 1С, и значение оптимального порядка АР-модели, найденное в п. 2С (входные параметры function-файла), и выполнить следующие действия:

- вычислить оценки параметров АР-модели методами Юла—Уолкера, Берга, ковариационным и модифицированным ковариационным;
- построить карты нулей и полюсов для проверки устойчивости БИХ-фильтров, соответствующих АР-модели;
- вычислить оценку СПМ методами Юла—Уолкера, Берга, ковариационным и модифицированным ковариационным;
- вывести графики оценок СПМ и истинной СПМ на одних координатных осях линиями разных цветов с легендой.

В качестве выходных параметров function-файла использовать оценки СПМ.

4С. Сравнение оценок СПМ с истинной СПМ.

Для сравнения оценок СПМ с истинной СПМ вывести значения RMSE (выходной параметр function-файла).

5С. Сравнение оценок СПМ с использованием непараметрических и параметрических методов.

Выбирая в качестве анализируемой последовательность, сформированную в п. 1С, и значение оптимального порядка АР-модели, найденное в п. 2С, выполнить следующие действия:

- вычислить оценку СПМ методом Берга;
- вычислить периодограмму (16.2);
- вычислить периодограмму Уэлча (16.18) при длине фрагмента $L = 50$ и величине перекрытия $Q = \text{int}(L/2)$ с применением окна Хэмминга (по умолчанию).

Вывести графики оценок СПМ и истинной СПМ на одних координатных осях линиями разных цветов с легендой.

6С. Описание оценок СПМ в виде объектов `spectrum`.

Описать оценки СПМ в виде объектов `spectrum` и вывести их графики.

17.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения каждого пункта задания, включая копируемые из окна **Command Window** результаты вычислений (шрифт Courier New), созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**) и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

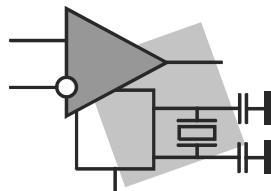
1. На чем основаны параметрические методы оценки СПМ и чем они принципиально отличаются от непараметрических методов?
2. Назовите основные преимущества параметрических методов по сравнению с непараметрическими.
3. В чем заключается основная сложность применения параметрических методов?
4. Назовите основные группы параметрических методов оценивания СПМ.
5. Запишите соотношение вход/выход для АРСС-, АР- и СС-моделей и поясните термин "моделируемая последовательность".
6. Запишите передаточные функции фильтров, соответствующие АРСС-, АР- и СС-моделям, и поясните их вид.
7. Перечислите основные этапы расчета параметрической оценки СПМ.
8. Какая из моделей, АР, СС или АРСС, получила наибольшее распространение на практике и почему?

9. Поясните термин "анализируемая последовательность" применительно к параметрическому спектральному оцениванию.
10. Поясните связь между моделируемой и анализируемой последовательностями в методе Юла—Уолкера.
11. Что такое "ошибка линейного предсказания"?
12. По какому критерию определяют наилучшее приближение моделируемой последовательности к анализируемой в методе Юла—Уолкера?
13. Какая оптимизационная задача решается при вычислении параметров линейного предсказания?
14. Почему вычисленные параметры являются оценками параметров АР-модели, а не их истинными значениями?
15. Как и с какой целью определяется оценка дисперсии нормального белого шума в АР-модели?
16. Какой вид имеет корреляционная матрица в СЛАУ Юла—Уолкера?
17. Перечислите рассмотренные параметрические методы оценивания СПМ на основе АР-модели и кратко охарактеризуйте их особенности.
18. Какие из методов оценки параметров АР-модели гарантируют устойчивость БИХ-фильтра?
19. Запишите и поясните критерий Байеса. К каким последствиям могут привести занижение и завышение порядка АР-модели?
20. Запишите и поясните основной показатель качества при сравнении параметрических оценок СПМ с истинной СПМ.

17.7. Литература

1. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов. — 2-е изд. — СПб.: БХВ-Петербург, 2005. — Глава 25.
2. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 12.
3. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 5.
4. Марпл С. Л. (мл.). Цифровой спектральный анализ и его приложения / Пер. с англ. — М.: Мир, 1990. — Глава 5.
5. Айфичер Э., Джервис Б. Цифровая обработка сигналов. — М., СПб., Киев: Вильямс, 2004. — Глава 11.
6. Бокс Дж., Дженкинс Г. Анализ временных рядов, прогноз и управление / Пер. с англ. — М.: Мир, 1974. — Глава 3.
7. Шахтарин Б. И., Ковригин В. А. Методы спектрального оценивания случайных процессов. — 2-е изд. — М.: Горячая Линия — Телеком, 2011. — Глава 3.

ГЛАВА 18



Спектральный анализ средствами GUI SPTool

Цель работы: овладеть средствами GUI SPTool (Signal Processing Toolbox — средство обработки сигнала) для моделирования систем цифровой фильтрации и спектрального анализа сигналов.

18.1. Краткая теоретическая справка

Средства GUI SPTool предназначены для моделирования систем цифровой фильтрации, анализа сигналов во временной области и спектрального анализа сигналов с использованием непараметрических и параметрических методов.

18.1.1. Обращение к GUI SPTool

Обращение к GUI SPTool происходит по команде:

```
sptool
```

после чего открывается окно **SPTool: startup.spt** с тремя группами (рис. 18.1):

- ☐ **Signals** (Сигналы);
- ☐ **Filters** (Фильтры);
- ☐ **Spectra** (Спектры).

Список каждой группы содержит тестовые объекты в помощь начинающему пользователю при освоении средств GUI SPTool.

Назначение групп рассматривается в *разд. 18.1.2—18.1.4*.

Сеанс работы в GUI SPTool называется *сессией* (Session). В GUI SPTool предусмотрена возможность *сохранения сессии* по команде меню **File | Save Session As** (Файл | Сохранить сессию как).

В открываемом окне **Save Session** (Сохранить сессию) указывается имя сессии — файла с расширением `spt` и нажимается кнопка **Сохранить**.

При последующих обращениях к GUI SPTool сохраненная сессия открывается по команде меню **File | Open Session** (Файл | Открыть сессию). В окне **Open Session** выбирается требуемая папка и в ней — файл с сохраненной сессией.

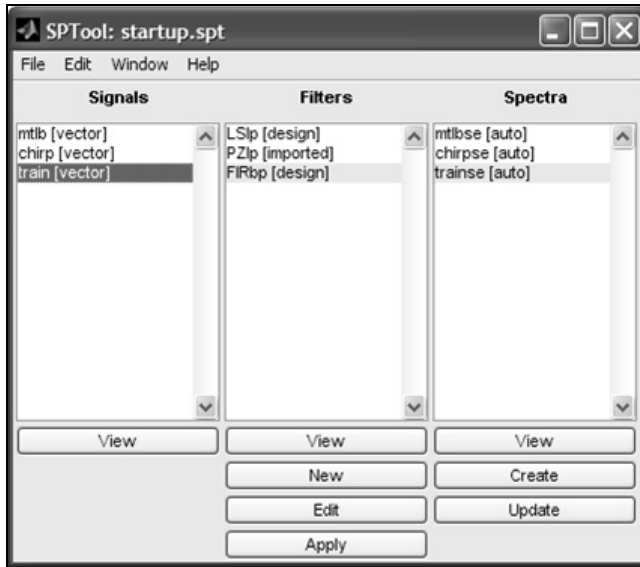


Рис. 18.1. Окно SPTool: startup.spt

18.1.2. Сигналы: группа *Signals*

Группа **Signals** включает список имен входных и выходных сигналов системы цифровой фильтрации и кнопку **View** (Вид) для визуализации сигналов (см. рис. 18.1).

Источником *входного* сигнала для GUI SPTool может являться:

- ☐ рабочее пространство памяти Workspace;
- ☐ диск, если сигнал сохранен как mat-файл данных в *текущей* папке.

Импорт входного сигнала в GUI SPTool выполняется по команде меню **File | Import** (Файл | Импорт), после чего открывается окно **Import to SPTool** (Импорт в SPTool) с двумя переключателями в группе **Source** (Источник):

- ☐ **From Workspace** (Из Workspace) — импорт из Workspace;
- ☐ **From Disk** (С диска) — импорт с диска.

При установке переключателя **From Workspace** необходимо выполнить следующие действия (рис. 18.2):

1. В группе **Workspace Contents** (Содержимое Workspace) выделить имя импортируемого сигнала, в примере — *s*.
2. Нажать кнопку **—>**, после чего имя переменной *s* будет отображено в поле **Data** (Данные).
3. В раскрывающемся списке **Import As** (Импортировать как) выбрать значение **Signal** (Сигнал).
4. В поле ввода **Sampling Frequency** (Частота дискретизации) указать частоту дискретизации в герцах, в примере — 1000.

5. В поле ввода **Name** (Имя) указать имя сигнала — по умолчанию sig с номером N, в примере — sig1.
6. Нажать кнопку **OK** и убедиться, что имя сигнала отобразилось в группе **Signals** окна **SPTool: startup.spt**.

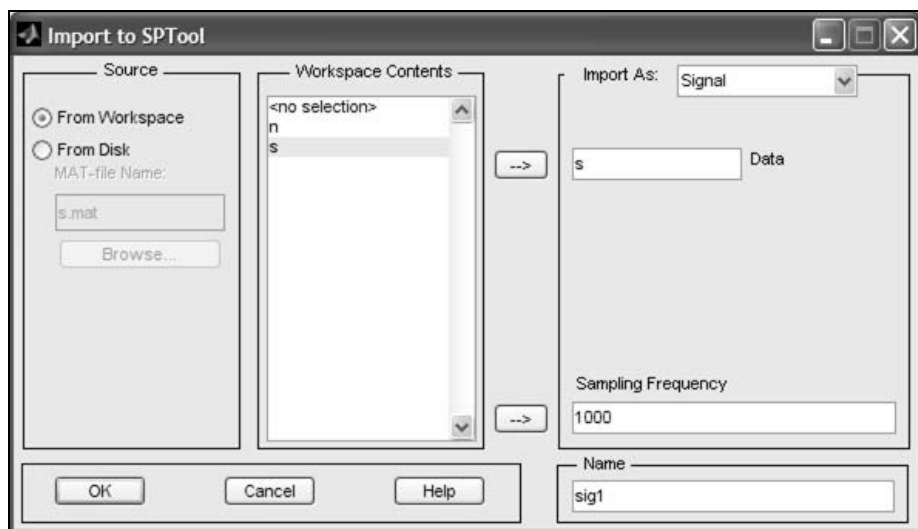


Рис. 18.2. Окно **Import to SPTool** при импорте **From Workspace**

При установке переключателя **From Disk** необходимо выполнить следующие действия (рис. 18.3):

1. В группе **Source** нажать кнопку **Browse** (Просмотреть). Откроется окно **Select File to Open** (Выбрать файл для открытия), в котором следует выделить имя mat-файла, в примере — signal.mat, и нажать кнопку **Открыть**. После этого произойдет автоматический возврат в окно **Import to SPTool**.
2. В группе **File Contents** (Содержимое файла) выделить имя переменной, соответствующей импортируемому сигналу, в примере — s (в файле signal.mat были сохранены две переменные — n и s).
3. Нажать кнопку **—>**, после чего имя переменной s будет отображено в поле **Data**.
4. В раскрывающемся списке **Import As** выбрать **Signal**.
5. В поле ввода **Sampling Frequency** указать частоту дискретизации в герцах, в примере — 1000.
6. В поле ввода **Name** указать имя входного сигнала, в примере — по умолчанию sig с номером N, в примере — sig2.
7. Нажать кнопку **OK** и убедиться, что имя сигнала отобразилось в группе **Signals** окна **SPTool: startup.spt**.

Формирование выходного сигнала рассматривается в разд. 18.1.4.

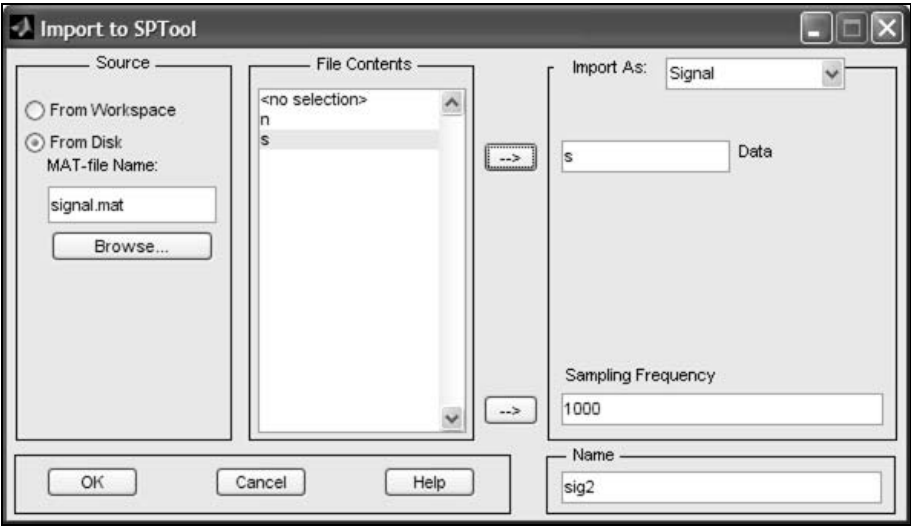


Рис. 18.3. Окно **Import to SPTool** при импорте **From Disk**

Визуализация и анализ сигнала выполняются после выделения имени сигнала (одного или нескольких с помощью клавиши <Ctrl>) в группе **Signals** и нажатия кнопки **View**, после чего открывается окно **Signal Browser** (Просмотр сигнала). На рис. 18.4 оно представлено для тестового сигнала **train**.

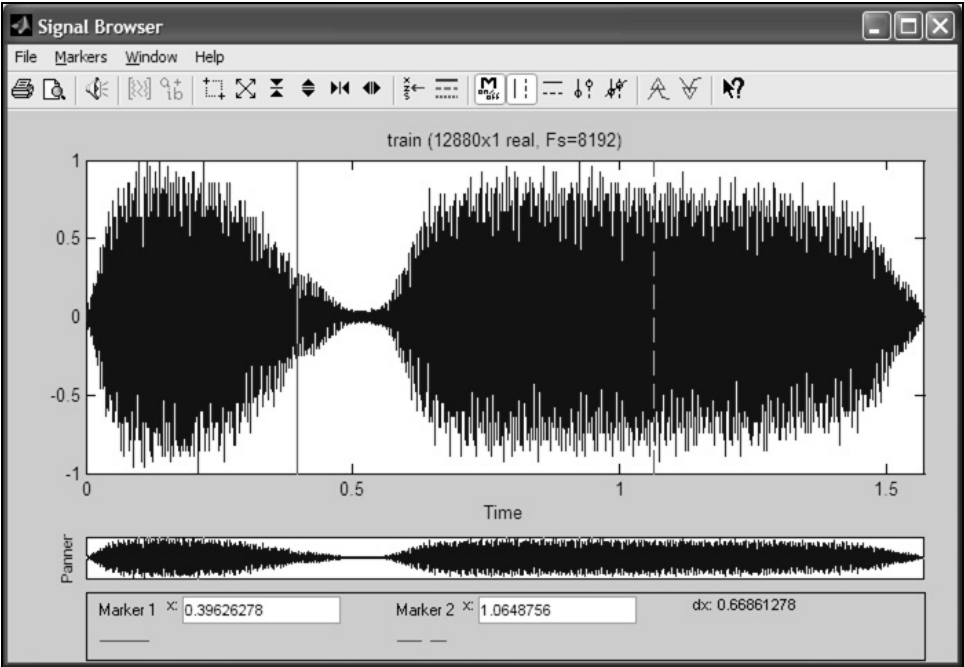


Рис. 18.4. Окно **Signal Browser** для тестового сигнала **train**

Окно **Signal Browser** включает в себя два графических поля (см. рис. 18.4):

- верхнее — для анализируемого сигнала в виде непрерывной функции с указанием количества отсчетов (длины сигнала) и частоты дискретизации F_s в герцах.

При выделении в группе **Signals** нескольких имен сигналов над графиком отображается их список;

- нижнее **Panner** (Панорама) — для панорамного (общего) вида сигнала с отображением в рамке его анализируемого участка при изменении масштаба.

С помощью кнопок на панели инструментов можно изменять масштаб, интервал наблюдения и активизировать маркеры, фиксирующие значения сигнала, а также его локальные максимумы и минимумы.

Задавая требуемые значения в полях ввода вертикальных или горизонтальных маркеров, можно зафиксировать их положение на графике.

Если анализируются одновременно несколько сигналов, то для их выделения разным цветом следует воспользоваться кнопкой **Line Properties** (Свойства линии) на панели инструментов или одноименной командой контекстного меню, щелкнув правой кнопкой мыши на поле графика. Рекомендуется внимательно относиться к выбору цветов, т. к. более темный цвет может скрыть более светлый.

Удаление или *переименование* сигнала, имя которого выделено в группе **Signals**, а также *изменение частоты дискретизации* выполняются с помощью соответствующих команд меню **Edit** окна **SPTool: startup.spt**.

Изменить имя сигнала или частоту дискретизации можно непосредственно в окне **Signal Browser** с помощью контекстного меню, которое открывается после щелчка правой кнопки мыши на поле графика.

18.1.3. Моделирование системы цифровой фильтрации: группа **Filters**

Группа **Filters** содержит список имен синтезированных цифровых фильтров (ЦФ) и четыре кнопки (см. рис. 18.1):

- **View** (Вид) — анализ ЦФ, имя которого выделено в группе **Filters**, средствами GUI FVTool (Filter Visualization Tool — средство визуализации фильтра);
- **New** (Новый фильтр) — синтез нового ЦФ с сохранением имени в группе **Filters**;
- **Edit** (Редактирование) — синтез ЦФ без изменения имени в группе **Filters**;
- **Apply** (Применить) — моделирование системы цифровой фильтрации.

Синтез нового ЦФ выполняется при нажатии кнопки **New**, после чего в группе **Filters** автоматически появляется имя ЦФ — по умолчанию **filt** с номером **N** — и открывается окно **Filter Design & Analysis Tool** GUI FDATool (см. разд. 14.1.1).

Для импорта ЦФ из Workspace необходимо выполнить следующие действия:

1. В группе **Filters** нажать кнопку **New**.
2. В открывшемся окне **Filter Design & Analysis Tool** выбрать команду меню **File | Import Filter from Workspace** (Файл | Импортировать фильтр из Workspace).
3. В полях ввода **Numerator** (Числитель) и **Denominator** (Знаменатель) ввести имена векторов коэффициентов ЦФ, сохраненные в Workspace.
4. Если импортируемый ЦФ был сохранен в виде объекта `dfilt`, то в раскрывающемся списке **Filter Structure** (Структура фильтра) следует выбрать значение **Filter object** и в поле ввода **Discrete Filter** ввести имя объекта `dfilt`.
5. В группе **Sampling Frequency** в поле **Units** выбрать Hz и указать частоту дискретизации в поле **Fs**.
6. Нажать кнопку **Import Filter** (Импортировать фильтр).

Удаление или переименование ЦФ, имя которого выделено в группе **Filters**, а также изменение частоты дискретизации выполняются с помощью соответствующих команд меню **Edit** окна **SPTool: startup.spt**.

Для моделирования системы цифровой фильтрации¹ необходимо выполнить следующие действия:

1. В группе **Signals** выделить имя входного сигнала.
2. В группе **Filters** выделить имя ЦФ.
3. Нажать кнопку **Apply**.

Открывается окно **Apply Filter** (Применить к фильтру) с именами входного сигнала, фильтра и выходного сигнала, по умолчанию `sig` с номером N (рис. 18.5).

4. В раскрывающемся списке **Algorithm** (Алгоритм) выбрать алгоритм вычисления выходного сигнала (реакции) ЦФ; по умолчанию реакция вычисляется по разностному уравнению при заданной структуре ЦФ.
5. Нажать кнопку **OK** — имя выходного сигнала отобразится в группе **Signals**.

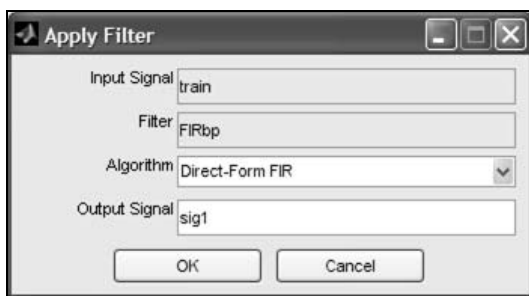


Рис. 18.5. Окно **Apply Filter**

¹ Система цифровой фильтрации включает в себя входной сигнал (воздействие), цифровой фильтр заданной структуры и выходной сигнал (реакцию).

18.1.4. Спектральный анализ: группа **Spectra**

Группа **Spectra** содержит список имен спектральной плотности мощности СПМ (PSD¹) сигналов и три кнопки (см. рис. 18.1):

- ❑ **View** — визуализация и анализ СПМ с именем, выделенным в группе **Spectra**;
- ❑ **Create** (Создать) — расчет СПМ сигнала с именем, выделенным в группе **Signals**, с сохранением имени СПМ в группе **Spectra**;
- ❑ **Update** (Обновление) — расчет СПМ без изменения имени в группе **Spectra**.

Визуализация и анализ СПМ выполняются после выделения имени СПМ (одного или нескольких с помощью клавиши <Ctrl>) в группе **Spectra** и нажатия кнопки **View**, после чего открывается окно **Spectrum Viewer** (Просмотр спектра). На рис. 18.6 оно представлено для СПМ trainse тестового сигнала train в списке **Signals**.

Если анализируются одновременно несколько СПМ, то для их выделения разным цветом следует воспользоваться кнопкой **Line Properties** на панели инструментов или одноименной командой контекстного меню, щелкнув правой кнопкой мыши на поле графика.

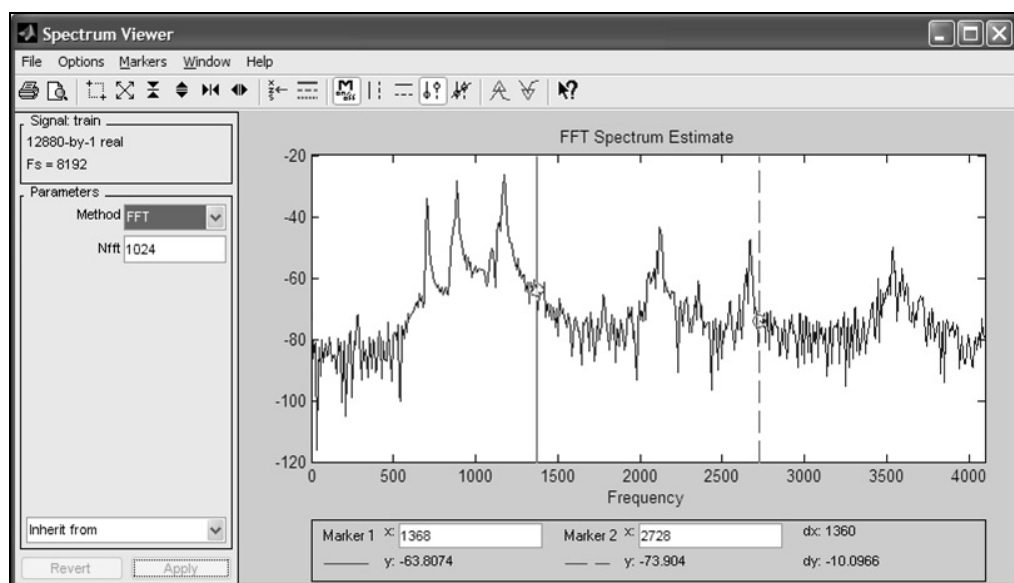


Рис. 18.6. Окно **Spectrum Viewer** для СПМ trainse тестового сигнала train

Вычисление и анализ СПМ сигнала, имя которого выделено в группе **Signals**, выполняются при нажатии кнопки **Create**, после чего в группе **Spectra** автоматически отображается имя СПМ — по умолчанию spect с номером N — и открывается окно **Spectrum Viewer** (см. рис. 18.6).

¹ Power Spectral Density — спектральная плотность мощности.

Окно **Spectrum Viewer** включает в себя две группы и одно графическое поле.

- ☐ Группу **Signal** — для отображения имени, длины и частоты дискретизации F_s анализируемого сигнала.

При выделении в группе **Spectra** нескольких имен СПМ, соответствующих различным сигналам, в группе **Signal** будет отображаться имя сигнала, соответствующее первому (верхнему) имени СПМ в группе **Spectra**.

- ☐ Группу **Parameters** — для выбора метода спектрального анализа и параметров настройки при расчете СПМ.

В списке **Method** представлены два метода *непараметрического* спектрального анализа:

- FFT — метод периодограмм.

Параметр настройки **Nfft** тождественен входному параметру `Nfft` функции `periodogram` (см. разд. 16.1.1);

- Welch — метод Уэлча.

Параметры настройки **Nfft**, **Nwind**, **Window** и **Overlap** тождественны входным параметрам функции `pwelch` (см. разд. 16.1.5), а именно: **Nfft** — `Lfft`; **Nwind** и **Window** — `window(L, winparameter)`; **Overlap** — `noverlap`.

Среди методов *параметрического* спектрального анализа выделим четыре, рассмотренные ранее в разд. 17.1.4:

- Yule AR — метод Юла—Уолкера;
- Burg — метод Берга;
- Covariance — ковариационный метод;
- Mod. Covar. — модифицированный ковариационный метод.

Параметры настройки данных методов одинаковы и соответствуют входным параметрам функций `pyulear`, `pburg`, `pcov` и `pmcov` (см. разд. 17.1.4), а именно: **Order** — `p`; **Nfft** — `Nfft`.

Нижний раскрывающийся список с заголовком **Inherit from** (Наследовать от) дублирует список СПМ в группе **Spectra**.

Кнопка **Revert** (Вернуться) активна *до нажатия кнопки Apply* и позволяет оперативно вернуться к предыдущему методу в списке **Method**.

- ☐ Графическое поле — для оценки СПМ (Power Spectral Density Estimate).

С помощью следующих команд меню **Options** можно управлять свойствами графика СПМ:

- **Magnitude Scale** (Шкала по оси ординат) — единицами измерения СПМ:
 - **decibels** — дБ/Гц (по умолчанию);
 - **Linear** — Вт/Гц;

- **Frequency Range** (Диапазон частот) — интервалом частот при выводе графика СПМ:
 - **[0, Fs/2]** — в основной полосе частот (по умолчанию), где **Fs** — частота дискретизации;
 - **[0, Fs]** — на периоде;
 - **[-Fs/2, Fs/2]** — на периоде, центрированном относительно нулевой частоты;
- **Frequency Scale** (Масштаб по оси частот) — масштабом по оси частот:
 - **Linear** — линейный;
 - **Log** — логарифмический.

С помощью кнопок на панели инструментов можно изменять масштаб, интервал наблюдения, активизировать маркеры, фиксирующие значения СПМ, а также ее пики и впадины.

После выбора метода спектрального анализа и установки необходимых параметров нажимается кнопка **Apply**.

Для расчета СПМ того же сигнала другим методом с *сохранением нового имени* в группе **Spectra** следует *закрыть* окно **Spectrum Viewer** и повторить процедуру.

При нажатой кнопке **View** в группе **Spectra** кнопка **Apply** в окне **Spectrum Viewer** позволяет оперативно анализировать оценки СПМ, вычисляемые разными методами, *без сохранения имени* в группе **Spectra**.

Удаление или *переименование* СПМ, имя которого выделено в группе **Spectra**, а также *изменение частоты дискретизации* выполняются с помощью соответствующих команд меню **Edit** окна **SPTool: startup.spt**.

Изменить имя СПМ или частоту дискретизации можно непосредственно в окне **Spectrum Viewer** с помощью контекстного меню, которое открывается после щелчка правой кнопки мыши на поле графика.

Вывод графика СПМ *в шкале дискретных нормированных частот* выполняется по команде контекстного меню **Sampling Frequency** и выборе частоты дискретизации, равной параметру настройки **Nfft**. Взаимосвязь между значениями абсолютных и дискретных нормированных частот определена в (9.3).

18.1.5. Экспорт данных из GUI SPTool

Экспорт данных (сигнала, фильтра или СПМ) из GUI SPTool выполняется в окне **SPTool: startup.spt** по команде меню **File | Export** (Файл | Экспорт), после чего открывается окно **Export from SPTool** (Экспорт из SPTool), в котором необходимо выполнить следующие действия:

1. В группе **Export List** (Список экспортируемых данных) выделить имя экспортируемых данных (если их несколько, то с помощью клавиши <Ctrl>).
2. С помощью соответствующих кнопок указать, куда экспортируются данные (**Export to disk** или **Export to workspace**).

При экспорте данных *в Workspace* они автоматически сохраняются с теми же именами, что и в GUI SPTool.

При экспорте данных *на диск* в автоматически открывающемся окне **Export to Disk** (Экспорт на диск) следует продублировать имя mat-файла (без расширения), после чего нажать кнопку **Сохранить**, и данные сохраняются в *текущей* папке. При экспорте на диск одновременно нескольких данных они объединяются в один mat-файл, имя которого указывается пользователем. Загрузка mat-файла с диска в *Workspace* осуществляется по команде `load`.

При экспорте в *Workspace* или на диск данные представляются в виде *массивов записей* (см. разд. 3.1.3). Обращение к массиву записей (структуре) производится по его имени, совпадающему с именем в окне **SPTool: startup.spt**. Список полей массива зависит от экспортируемых данных.

18.2. Содержание лабораторной работы

Содержание работы связано с моделированием систем цифровой фильтрации и методов непараметрического и параметрического спектрального анализа сигналов средствами GUI SPTool.

18.3. Задание на лабораторную работу

Исходные данные для пунктов задания приводятся в табл. 18.1 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$. Функция $N_{бр} \bmod M$ в записи исходных данных означает вычисление значения $N_{бр}$ по модулю M .

На прилагаемом компакт-диске в папке `Tables\Tables_18` хранятся табл. 18.1 с исходными данными и пример ее заполнения для $N_{бр} = 1$.

Таблица 18.1. Таблица исходных данных

Переменная	Назначение	Значение	Идентификатор
$N_{бр}$	Номер бригады	$N_{бр}$	Nb =
N	Длина последовательности	$N = 512$	N = 512
f_d	Частота дискретизации	$f_d = 1000(N_{бр} \bmod 5 + 1)$	Fs =
A_1	Амплитуды дискретных гармоник	$A_1 = 0,8 + 0,01N_{бр}$	A1 =
A_2		$A_2 = 1,5A_1$	A2 =
f_1	Частоты дискретных гармоник	$f_1 = f_d/8$	f1 =
f_2		$f_2 = 2f_1$	f2 =

Таблица 18.1 (окончание)

Переменная	Назначение	Значение			Идентификатор
N_1	Длина нормального белого шума	$N_1 = 1000(N_{\text{бр}} \bmod 3 + 1)$			N1 =
f_k	Граничная частота ПЗ	$f_k = f_{\text{д}}/8$			—
f_{χ}	Граничная частота ПП	$f_{\chi} = f_{\text{д}}/8 + f_{\text{д}}/10$			—
a_{min}	Минимально допустимое затухание в ПЗ	$a_{\text{min}} = 40$			—
a_{max}	Максимально допустимое затухание в ПП	$a_{\text{max}} = 0,4455$			—
a	Заданные (истинные) параметры АР-модели	Номера бригад $N_{\text{бр}}$			Вектор $a = [1 \ \dots]$
		1—10	11—20	21—30	
a_0		1	1	1	
a_1		−0,86	−0,10	0,50	
a_2		0,54	0,20	0,17	
a_3		−0,30	0,20	0,30	
a_4		−0,17	0,10	0,10	
a_5		0,22	0,30	0,10	
a_6		−0,10	—	−0,10	
a_7		—	—	−0,50	

Задание на лабораторную работу заключается в моделировании системы цифровой фильтрации и методов непараметрического и параметрического спектрального анализа в GUI SPTool и включает в себя следующие пункты:

1. Импорт и анализ сигнала в GUI SPTool.

В окне **Command Window** выполнить моделирование дискретного сигнала — конечной последовательности $x(n)$ (идентификатор $\times$) длины N :

$$x(n) = A_1 \cos\left(\frac{2\pi f_1 n}{f_{\text{д}}}\right) + A_2 \cos\left(\frac{2\pi f_2 n}{f_{\text{д}}}\right) = A_1 \cos(\hat{\omega}_1 n) + A_2 \cos(\hat{\omega}_2 n). \quad (18.1)$$

В окне **Command Window** ввести следующие строки, подставляя вместо многоточия значения переменных:

```
Nb = ...;
N = 512;
Fs = ...;
A1 = ...;    A2 = ...;
f1 = ...;    f2 = ...;
w1 = 2*pi*f1/Fs;  w2 = 2*pi*f2/Fs;    % НОРМИРОВАННЫЕ ЧАСТОТЫ
n = 0:(N-1);      % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
x = A1*cos(w1*n)+A2*cos(w2*n);        % ДИСКРЕТНЫЙ СИГНАЛ
```

Импортировать сигнал x из **Workspace** в **SPTool** и сохранить в группе **Signal** с именем **sig1** по умолчанию (см. разд. 18.1.2).

Проанализировать вид сигнала **sig1** в окне **Signal Browser**.

Пояснить:

- чему равна длительность сигнала **sig1** — сравнить вычисленное значение со значением на графике;
- чему равны наибольшее и наименьшее значения сигнала.

2. Расчет СПМ гармонического сигнала методом периодограмм.

Рассчитать СПМ сигнала **sig1** методом периодограмм. Задать параметр настройки **Nfft** равным длине сигнала **sig1** и вывести его СПМ (Вт/Гц) на периоде (см. разд. 18.1.4).

Сохранить СПМ сигнала **sig1** в группе **Spectra** с именем **spect1** (по умолчанию).

Проанализировать СПМ **spect1** в окне **Spectrum Viewer** с помощью кнопок на панели инструментов.

Пояснить:

- чему равны период и основная полоса частот СПМ в шкале частот в герцах;
- чему равны частоты гармоник в герцах и в шкале дискретных нормированных частот; проверить соответствие по формуле (9.3);
- чему равны значения СПМ (Вт/Гц) и СПМ (дБ/Гц) на частотах гармоник; проверить соответствие по формуле (16.4).

3. Импорт входного сигнала (воздействия) системы цифровой фильтрации в GUI SPTool.

В окне **Command Window** выполнить моделирование входного сигнала — нормального белого шума $e(n)$ (идентификатор e) длины N_1 с нулевым средним и единичной дисперсией:

```
N1 = ...;    % ДЛИНА ВОЗДЕЙСТВИЯ
e = randn(1,N1);    % НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ
```

Импортировать сигнал e из **Workspace** в **SPTool**, задавая частоту дискретизации f_d , и сохранить в группе **Signal** с именем **sig2** (по умолчанию).

Проанализировать сигнал sig2 в окне **Signal Browser**.

Пояснить:

- чему равна оценка среднего значения шума заданной длины (рассчитать с помощью функции `mean`);
- как с помощью горизонтального маркера **Marker 1** зафиксировать оценку среднего значения шума на графике сигнала sig2.

4. Синтез цифрового фильтра.

Синтезировать БИХ-фильтр ФНЧ Баттерворта по требованиям к характеристике затухания АЧХ (дБ) (11.6): граничные частоты ПЗ f_k и ПП f_χ , минимально допустимое затухание в ПЗ $a_{\min}$ и максимально допустимое затухание в ПП $a_{\max}$.

Сохранить синтезированный фильтр в группе **Filters** с именем `filt2` (для соответствия с именем входного сигнала sig2).

Проанализировать характеристики синтезированного фильтра.

Пояснить:

- чему равен порядок БИХ-фильтра;
- какая структура БИХ-фильтра выбрана по умолчанию.

5. Моделирование системы цифровой фильтрации.

Вычислить выходной сигнал (реакцию) БИХ-фильтра с именем `filt2` на воздействие в виде нормального белого шума — сигнала с именем sig2.

Сохранить реакцию в группе **Signals** с именем sig3.

Вывести и проанализировать графики воздействия sig2 и реакции sig3, выделяя их в окне **Signal Browser** разными цветами.

Пояснить:

- какой из сигналов, sig2 или sig3, имеет бóльшую дисперсию визуально; подтвердить вычислением в окне **Command Window**, предварительно экспортируя сигнал sig3 в Workspace:

```
var(e) =  
var(sig3.data) =
```

где `data` — поле массива записей sig3, значением которого является вектор значений отсчетов сигнала sig3;

- что в себя включает система цифровой фильтрации.

6. Расчет СПМ случайного сигнала методом периодограмм.

Рассчитать СПМ воздействия sig2 и реакции sig3 методом периодограмм на периоде, задавая параметр настройки **Nfft** равным N_1 и сохраняя СПМ в группе **Spectra** с именами соответственно `spect2` и `spect3` (по умолчанию).

Вывести и сравнить графики СПМ (Вт/Гц), выделяя их в окне **Spectrum Viewer** разными цветами.

Пояснить, как изменилась СПМ реакции по сравнению с СПМ воздействия.

7. Расчет СПМ случайного сигнала методом периодограмм Уэлча.

Рассчитать СПМ реакции sig3 методом периодограмм Уэлча при следующих параметрах настройки:

- **Nfft** — равен длине реакции;
- **Window** — окно Хэмминга;
- **Nwind** — 10%, 5% и 2,5% от длины реакции;
- **Overlap** — 1,25% от длины реакции.

Длину фрагментов и величину перекрытия округлить до ближайшего целого в сторону увеличения.

Сохранить в группе **Spectra** три полученные СПМ с именами соответственно Welch1, Welch2 и Welch3.

Вывести и сравнить графики СПМ Welch1, Welch2 и Welch3, выделяя их в окне **Spectrum Viewer** разными цветами.

Пояснить:

- как параметр **Nwind** влияет на вид периодограммы Уэлча (она становится более или менее осциллирующей);
- при каких параметрах настройки будет рассчитана периодограмма Бартлетта (см. разд. 16.1.4).

8. Импорт БИХ-фильтра, соответствующего AP-модели, в GUI SPTool.

В окне **Command Window** сформировать векторы коэффициентов БИХ-фильтра: $b_0 = 1$ (**b**) и **a** (вектор **a**):

```
b = 1; % КОЭФФИЦИЕНТЫ ЧИСЛИТЕЛЯ
a = [1 ...]; % КОЭФФИЦИЕНТЫ ЗНАМЕНАТЕЛЯ
```

Импортировать БИХ-фильтр, соответствующий AP-модели (см. разд. 18.1.3).

Пояснить:

- вид передаточной функции БИХ-фильтра;
- как проверить устойчивость БИХ-фильтра.

9. Моделирование случайного дискретного сигнала на основе AP-модели.

Вычислить выходной сигнал (реакцию) БИХ-фильтра с именем filt3 — на воздействие в виде нормального белого шума — сигнала с именем sig2.

Сохранить реакцию в группе **Signals** с именем sig4 (по умолчанию).

Проанализировать вид сигнала sig4 в окне **Signal Browser**.

Пояснить:

- что используется в качестве воздействия в АР-модели;
- вид разностного уравнения, описывающего АР-модель.

10. Расчет оценок СПМ с использованием параметрических методов.

Полагая, что анализируемый сигнал совпадает с реакцией sig4 АР-модели, рассчитать его оценки СПМ с использованием параметрических методов и сохранить в группе **Spectra** со следующими именами:

- метод Юла—Уолкера — Yule;
- метод Берга — Burg;
- ковариационный метод — Cov;
- модифицированный ковариационный метод — ModCov.

Задать параметры настройки данных методов: **Order**, равным порядку АР-модели, а **Nfft**, равным N_1 .

Сравнить полученные оценки СПМ, выделенные разными цветами, в окне **Spectrum Viewer**.

Пояснить:

- наблюдаются ли существенные расхождения в оценках СПМ, вычисленных различными методами;
- какое количество пиков и впадин содержится в каждой оценке СПМ;
- на каких частотах расположены пики и впадины в оценках СПМ;
- чему равны значения СПМ на частотах, соответствующих пикам и впадинам.

11. Сохранение сессии GUI SPTool.

По завершении работы в GUI SPTool сохранить сессию с именем `Spectrum_Analysis.spt` (см. разд. 18.1.1).

18.4. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в моделировании системы цифровой фильтрации и методов непараметрического и параметрического спектрального анализа в GUI SPTool с использованием новых исходных данных и данных из табл. 18.1 для своего номера бригады $N_{бр}$.

Пункты самостоятельного задания включают в себя:

1С. Моделирование случайной последовательности.

В окне **Command Window** смоделировать случайную последовательность длины $N=10000$ с именем `sequence` на основе АР-модели с коэффициентами, заданными вектором **a**: $\mathbf{a} = [1 \ -0.20 \ 0.20 \ -0.40 \ 0.10 \ -0.30 \ 0.20 \ -0.20]$.

2С. Сохранение случайной последовательности на диске.

Сохранить последовательность `sequence` в файле данных с тем же именем `sequence` в *текущей* папке.

3С. Импорт и анализ случайной последовательности в GUI SPTool.

Импортировать случайную последовательность `sequence` с диска в GUI SPTool, задавая частоту дискретизации f_d , и сохранить в группе **Signal** с именем Seq1.

Проанализировать сигнал Seq1 в окне **Signal Browser**.

4С. Синтез цифрового фильтра и моделирование системы цифровой фильтрации.

Синтезировать КИХ-фильтр ФВЧ методом чебышевской аппроксимации по требованиям к характеристике затухания (дБ) (11.6): граничные частоты в ПЗ f_k и ПП f_χ , минимально допустимое затухание в ПЗ $a_{\min}$ и максимально допустимое затухание в ПП $a_{\max}$.

Сохранить синтезированный ФВЧ в группе **Filters** с именем Filt1 и проанализировать его характеристики.

Вычислить реакцию КИХ-фильтра с именем Filt1 на воздействие в виде случайной последовательности с именем Seq1.

Сохранить реакцию в группе **Signals** с именем Seq2.

Проанализировать сигналы Seq1 и Seq2 в окне **Signal Browser**.

5С. Расчет и анализ оценок СПМ на входе системы цифровой фильтрации с использованием непараметрических и параметрических методов.

В качестве исходной последовательности (анализируемого сигнала) выбрать последовательность Seq1 на входе КИХ-фильтра (воздействие).

Сохранить вычисленные СПМ в группе **Spectra** со следующими именами:

- метод периодограмм — Periodogram1.

Параметр настройки **Nfft** задать равным длине воздействия Seq1;

- метод периодограмм Уэлча — Welch1.

Задать следующие параметры настройки:

- **Nfft** — равен длине воздействия;
- **Window** — окно Хэмминга;
- **Nwind** — 10% от длины воздействия;
- **Overlap** — 50% от длины окна **Nwind**;

- метод Юла—Уолкера — Yule;

- метод Берга — Burg.

Для методов Юла—Уолкера и Берга задать параметры настройки: **Order** — равным порядку АР-модели и **Nfft** — равным длине последовательности Seq1.

Проанализировать оценки СПМ в окне **Spectrum Viewer**.

6С. Расчет и анализ оценок СПМ на выходе системы цифровой фильтрации с использованием непараметрических методов.

В качестве исходной последовательности выбрать последовательность Seq2 на выходе КИХ-фильтра (реакцию).

Сохранить вычисленные СПМ в группе **Spectra** со следующими именами:

- метод периодограмм — Periodogram2.

Параметр настройки **Nfft** задать равным длине реакции Seq2;

- метод периодограмм Уэлча — Welch2.

Параметры настройки задать такие же, как в п. 5С.

Проанализировать оценки СПМ реакции в окне **Spectrum Viewer**.

Сравнить оценки СПМ воздействия Periodogram1 и реакции и Periodogram2 в окне **Spectrum Viewer**.

7С. Сохранение сессии GUI SPTool.

По завершении работы в GUI SPTool сохранить сессию с указанным пользователем именем.

18.5. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения каждого пункта задания, включая полученные графики (копируются при нажатии комбинации клавиш <Alt>+<Print Screen>) и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

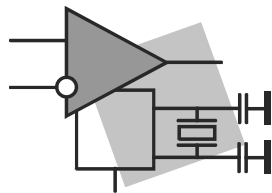
1. Для чего предназначены средства GUI SPTool?
2. Какие группы содержит окно **SPTool: startup.spt**?
3. Как называется сеанс работы в GUI SPTool?
4. Как сохранить результаты сеанса работы в GUI SPTool?
5. Как осуществляется импорт сигнала из рабочего пространства Workspace и с диска в GUI SPTool?
6. Какие графические поля и средства управления графиками представлены в окне **Signal Browser**?
7. Как осуществляется импорт ЦФ из Workspace в GUI SPTool?
8. Какие действия необходимо выполнить для моделирования системы цифровой фильтрации в GUI SPTool?
9. Какие методы расчета СПМ используются в GUI SPTool?
10. Какие средства управления графиками представлены в окне **Spectrum Viewer**?

11. Как осуществляется экспорт данных из GUI SPTool на диск и в Workspace?
12. В каком виде представляются экспортируемые данные?

18.6. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 21.
2. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Приложение 4.

ГЛАВА 19



Многоскоростные системы ЦОС

Цель работы: изучить математическое описание многоскоростных систем и овладеть программными средствами их моделирования в MATLAB.

19.1. Краткая теоретическая справка

Многоскоростными называют системы ЦОС, в которых различные этапы цифровой обработки выполняются на разных частотах дискретизации — разных скоростях поступления отсчетов. В таких системах необходима "стыковка" соответствующих этапов цифровой обработки, которая сводится к *преобразованию частоты дискретизации* в одном из следующих вариантов или их комбинации:

- от меньшей к большей — повышение частоты дискретизации в *целое* число раз L , называемое *интерполяцией* (interpolation) и выполняемое *системой интерполяции* с коэффициентом интерполяции L , равным:

$$L = \frac{f'_d}{f_d} \Rightarrow f'_d = Lf_d, \quad (19.1)$$

где f_d и f'_d — частоты дискретизации сигналов на *входе* и *выходе* системы интерполяции соответственно;

- от большей к меньшей — понижение частоты дискретизации в *целое* число раз M , называемое *децимацией* (decimation) и выполняемое *системой децимации* с коэффициентом децимации M , равным:

$$M = \frac{f_d}{f'_d} \Rightarrow f'_d = \frac{f_d}{M}, \quad (19.2)$$

где f_d и f'_d — частоты дискретизации сигналов на *входе* и *выходе* системы децимации соответственно.

Повышение или понижение частоты дискретизации на рациональный коэффициент L/M , называемое *передискретизацией* (resampling), реализуется каскадным соеди-

нением систем интерполяции с коэффициентом L и децимации с коэффициентом M .

Системы преобразования частоты называют *однократными*, если увеличение (уменьшение) частоты дискретизации выполняется за один прием — однократно.

Многократными называют системы, образованные каскадным соединением однократных систем, что оправдано при больших значениях L и M , т. к. требования к однократным системам существенно менее жесткие.

В ЦОС преобразование частоты дискретизации выполняется средствами *цифровой фильтрации*, в результате которой формируется выходной сигнал с новой частотой дискретизации.

19.1.1. Система однократной интерполяции

Система однократной интерполяции включает в себя два блока (рис. 19.1):

- экспандер;
- цифровой ФНЧ.

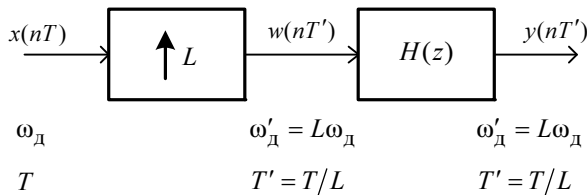


Рис. 19.1. Система однократной интерполяции

Экспандер (блок со стрелкой вверх) формирует промежуточный сигнал $w(nT')$ с периодом дискретизации $T' = T/L$ (частотой дискретизации $\omega'_d = L\omega_d$) путем добавления $(L-1)$ равноотстоящих нулей на каждом периоде дискретизации T входного сигнала $x(nT)$:

$$w(nT') = \begin{cases} x\left(\frac{nT}{L}\right), & n = 0, L, 2L, \dots; \\ 0 & \text{при других } n. \end{cases} \quad (19.3)$$

Соотношение между периодами дискретизации входного $x(nT)$ и промежуточного $w(nT')$ сигналов позволяет интерпретировать "новую" шкалу nT' как *растяжение* "старой" шкалы nT в L раз (рис. 19.2, а, б).

Соотношение между частотами дискретизации $\omega'_d = L\omega_d$ спектральных плотностей данных сигналов $X(e^{j\omega T})$ и $W(e^{j\omega' T'})$ позволяет интерпретировать "новую" шкалу ω' , напротив, как *сжатие* "старой" шкалы ω в L раз (рис. 19.3, а, б).

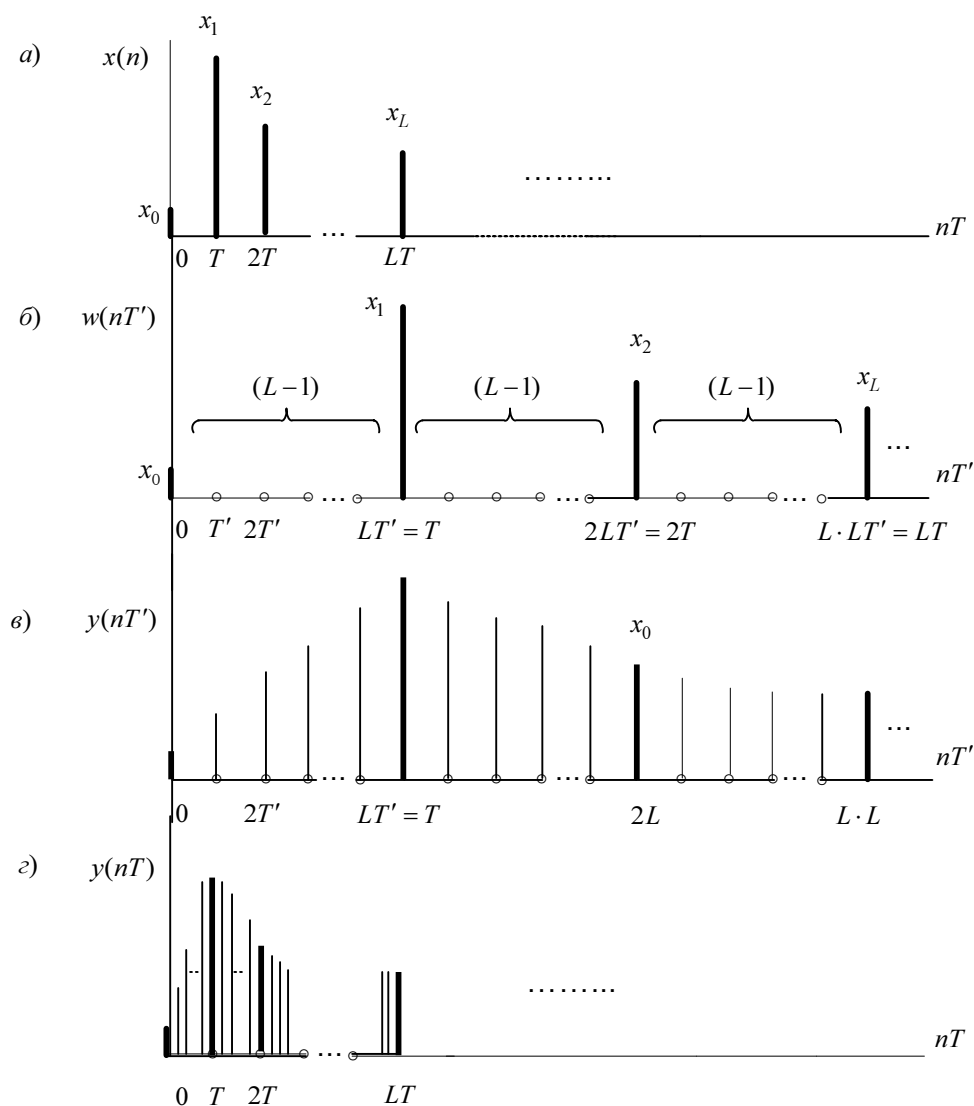


Рис. 19.2. Интерпретация процедуры интерполяции во временной области: входной (а), промежуточный (б) и выходной сигналы в "новой" (в) и "старой" (г) шкале времени

Цифровой ФНЧ предназначен для подавления $(L-1)$ "лишних" составляющих спектральной плотности промежуточного сигнала на периоде ω'_d , следовательно, его идеальная АЧХ в основной полосе частот шкалы ω' должна удовлетворять требованиям (рис. 19.3, в):

$$A(\omega') = \begin{cases} L & \text{в полосе пропускания } 0 \leq \omega' \leq \omega'_d/2L; \\ 0 & \text{при других значениях } \omega', \end{cases} \quad (19.4)$$

или в шкале частот ω :

$$A(\omega)=\begin{cases} L & \text{в полосе пропускания } 0 \leq \omega \leq \omega_{\text{д}}/2; \\ 0 & \text{при других значениях } \omega. \end{cases}$$

(19.5)

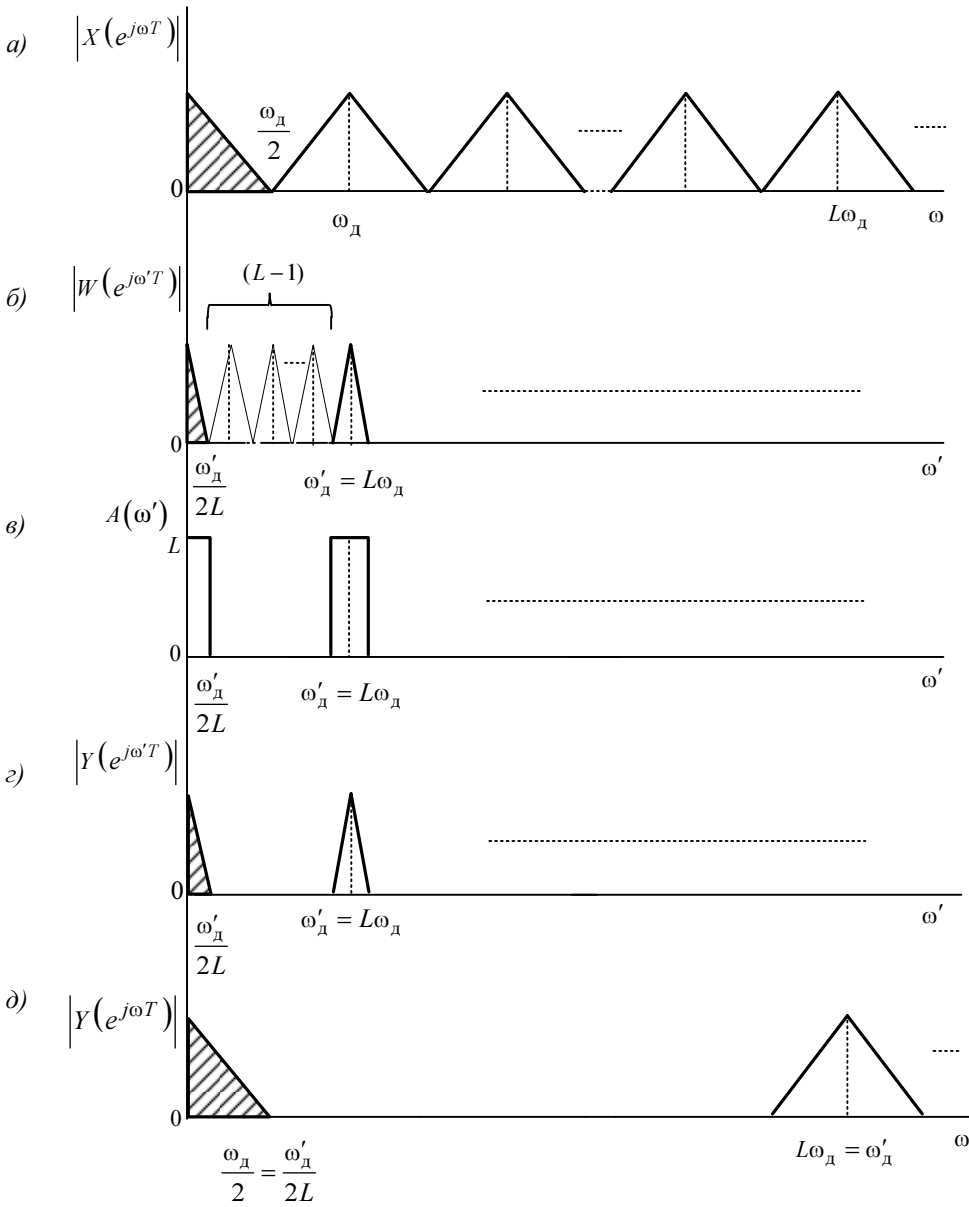


Рис. 19.3. Интерпретация процедуры интерполяции в частотной области: модули спектральных плотностей входного сигнала (а); промежуточного сигнала в "новой" шкале частот (б); идеальная АЧХ (в); выходной сигнал в "новой" (г) и "старой" (д) шкалах частот

На рис. 19.2, в и рис. 19.3, з изображены выходной сигнал $y(nT')$ и модуль его спектральной плотности $Y(e^{j\omega'T})$, а на рис. 19.2, з и рис. 19.3, д — они же в "стационарных" шкалах времени nT и частот ω . Период дискретизации выходного сигнала уменьшился, а частота дискретизации увеличилась в L раз.

Модули спектральной плотности дискретного сигнала на выходе ФНЧ (см. рис. 19.3, з) и спектральной плотности дискретного сигнала, которая была бы получена путем непосредственной дискретизации исходного аналогового сигнала с частотой $\omega'_d = L\omega_d$, совпадают с точностью до множителя L (который учтен в (19.5)), а аргументы различаются¹.

В действительности АЧХ не является идеальной, поэтому не только аргументы, но и модули спектральных плотностей могут отличаться. Для сохранения формы исходного сигнала (исключения влияния фазовых искажений) в качестве ФНЧ выбирают КИХ-фильтр с линейной ФЧХ.

С учетом (19.3), соотношение вход/выход системы интерполяции в z -области имеет вид:

$$Y(z) = W(z)H(z) = X(z^L)H(z), \quad (19.6)$$

где:

$$W(z) = \sum_{n=0, L, 2L, \dots}^{\infty} x\left(\frac{nT}{L}\right) z^{-n} = \sum_{m=0}^{\infty} x(mT) z^{-Lm} = X(z^L)$$

при подстановке $n = Lm$.

В MATLAB моделирование системы однократной интерполяции реализуется с помощью функции:

`[y,h] = interp(x,L,I,fx)`

где x — входной сигнал; L — коэффициент интерполяции; I — необязательный параметр, управляющий длиной КИХ-фильтра, равной $2 \cdot L \cdot I + 1$; по умолчанию $I = 4$; y — выходной сигнал, длина которого в L раз больше длины входного сигнала x ; h — импульсная характеристика КИХ-фильтра ФНЧ; по умолчанию КИХ-фильтр синтезируется методом окон (выбор окна скрыт от пользователя); fx — необязательный параметр, задающий нормированную частоту разрыва $\hat{f}_c = \frac{f_c}{f_d/2}$

(см. разд. 11.1.5) в диапазоне $0 < fx \leq 1$; по умолчанию $fx = 0.5$.

По умолчанию граничная частота полосы пропускания (ПП) АЧХ не превосходит $f_d/4$, где f_d — частота дискретизации входного сигнала, при этом в ПП максимально допустимое отклонение АЧХ от L минимально (см. (19.5)).

Для увеличения граничной частоты ПП следует увеличить значение I или/и fx , однако это может сопровождаться возрастанием неравномерности АЧХ в ПП.

¹ Подробный вывод этих соотношений приводится в [2].

19.1.2. Система однократной децимации

Система однократной децимации включает в себя два блока (рис. 19.4):

- цифровой ФНЧ;
- компрессор.

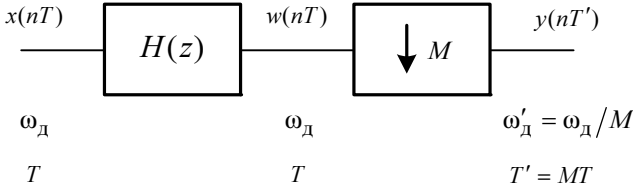


Рис. 19.4. Система однократной децимации

Цифровой ФНЧ с передаточной функцией $H(z)$ предназначен для ограничения верхней частоты $\omega_d/2$ спектральной плотности $X(e^{j\omega T})$ входного сигнала $x(nT)$ до $\omega'_d/2 = \omega_d/2M$, что требует пояснения.

Соотношение между частотами дискретизации $\omega'_d = \omega_d/M$ спектральных плотностей $X(e^{j\omega T})$ и $X(e^{j\omega' T'})$ позволяет интерпретировать "новую" шкалу ω' как *растяжение* "старой" шкалы ω в M раз (рис. 19.5, а, б).

В общем случае спектральная плотность $X(e^{j\omega T})$ может занимать всю основную полосу $[0; \omega_d/2]$ (рис. 19.5, а). При этом для "новой" частоты дискретизации $\omega'_d = \omega_d/M$ не будет выполняться условие теоремы Котельникова $\omega'_d \geq 2\omega_b$, где $\omega_b = \omega_d/2$, что приведет к *элайсингу*¹ (см. пунктирные линии на рис. 19.5, б). Чтобы этого избежать, идеальная АЧХ ФНЧ в основной полосе частот шкалы ω' должна удовлетворять требованиям (рис. 19.5, в):

$$A(\omega') = \begin{cases} 1 & \text{в полосе пропускания } 0 \leq \omega' \leq \omega'_d/2; \\ 0 & \text{при других значениях } \omega', \end{cases}$$

или в шкале частот ω :

$$A(\omega) = \begin{cases} 1 & \text{в полосе пропускания } 0 \leq \omega \leq \omega_d/2M; \\ 0 & \text{при других значениях } \omega. \end{cases} \quad (19.7)$$

Модуль спектральной плотности $W(e^{j\omega' T'})$ промежуточного сигнала $w(nT)$ на выходе ФНЧ показан на рис. 19.5, г.

¹ Элайсингом (aliasing) называют эффект наложения спектральных плотностей в области $\omega'_d/2$.

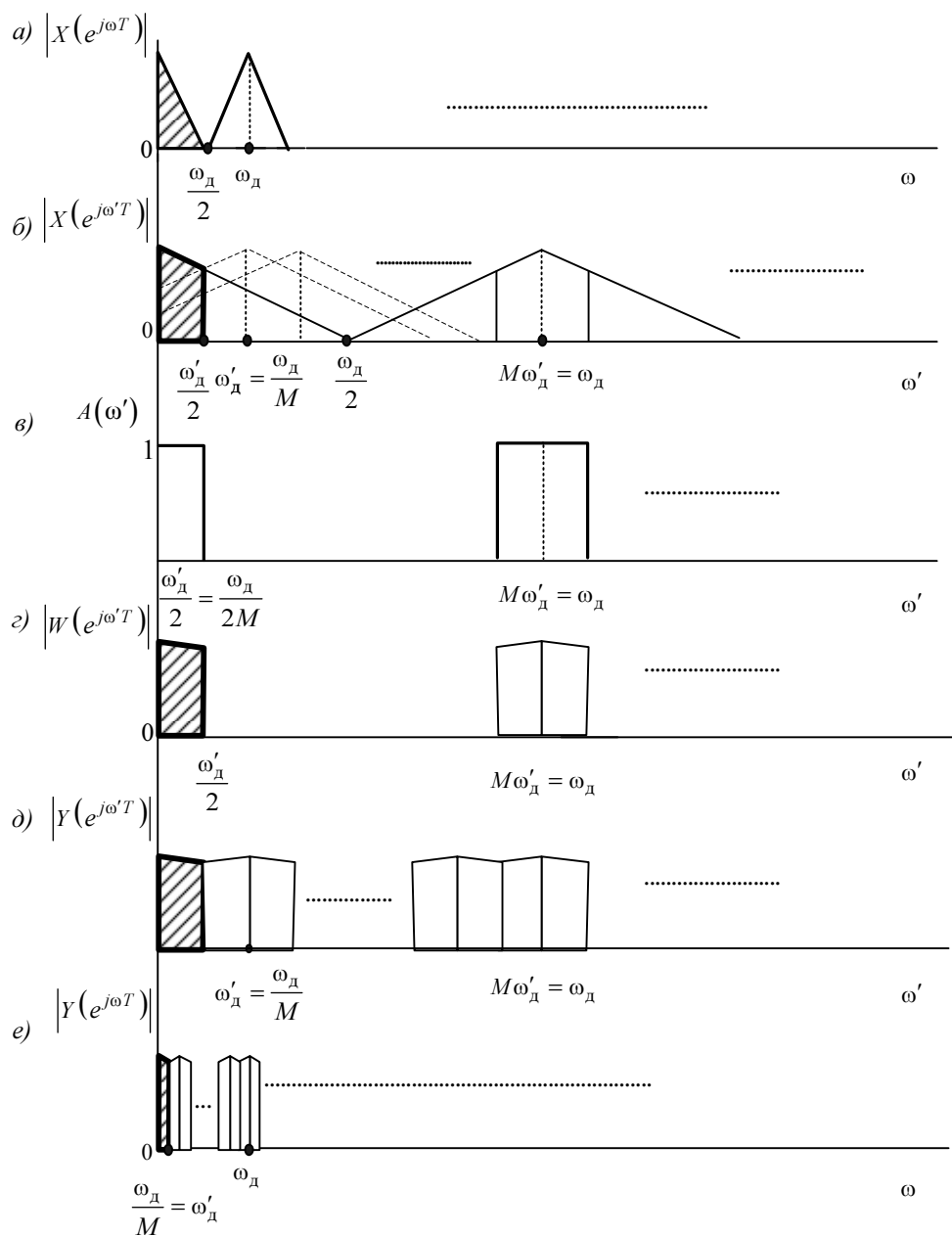


Рис. 19.5. Интерпретация процедуры децимации в частотной области: модуль спектральной плотности входного сигнала в "старой" (а) и "новой" (б) шкале частот; идеальная АЧХ (в); модули спектральных плотностей промежуточного сигнала (г) и выходного сигнала в "новой" (д) и "старой" (е) шкалах частот

Компрессор (блок со стрелкой вниз на рис. 19.4) формирует выходной сигнал $y(nT')$ путем *прореживания* промежуточного сигнала $w(nT)$, из которого берется каждый M -й отсчет, в результате чего "новый" период дискретизации оказывается равным $T' = MT$ (рис. 19.6, а, б):

$$y(nT') = w(MnT), \quad n = 0, 1, 2, \dots$$

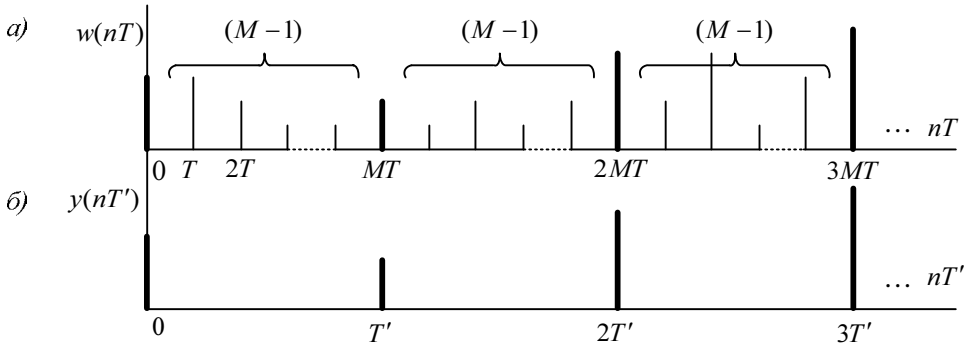


Рис. 19.6. Интерпретация процедуры децимации во временной области: промежуточный сигнал в "старой" шкале времени (а) и выходной сигнал в "новой" шкале времени (б)

Модуль спектральной плотности выходного сигнала в "новой" ω' и "старой" ω шкалах частот показан на рис. 19.5, д, е.

Модули спектральной плотности дискретного сигнала на выходе компрессора и спектральной плотности дискретного сигнала, которая была бы получена при непосредственной дискретизации исходного аналогового сигнала с частотой $\omega'_d = \omega_d / M$, совпадают, а *аргументы* различаются¹.

В действительности АЧХ не является идеальной, поэтому не только аргументы, но и модули спектральных плотностей могут отличаться. Для сохранения формы исходного сигнала (исключения влияния фазовых искажений) в качестве ФНЧ выбирают *КИХ-фильтр с линейной ФЧХ*.

В MATLAB моделирование системы однократной децимации с КИХ-фильтром ФНЧ реализуется с помощью функции:

```
y = decimate(x,M[,R,'fir'])
```

где x — входной сигнал; M — коэффициент децимации; R — необязательный параметр — порядок синтезируемого фильтра; 'fir' — необязательный параметр, указывающий на выбор КИХ-фильтра порядка R ; по умолчанию $R = 30$.

В отсутствии этого параметра по умолчанию синтезируется БИХ-фильтр Чебышева I рода порядка $R = 8$ с компенсацией фазовых искажений (в этом случае $R \leq 13$).

¹ Подробный вывод этих соотношений приводится в [2].

y — выходной сигнал, длина которого в M раз меньше длины входного сигнала x ; если отношение длин сигналов вход/выход оказывается *не целым* числом, длина выходного сигнала автоматически округляется до ближайшего целого, при этом в ДПФ выходного сигнала наблюдается эффект растекания спектра.

19.1.3. Система однократной передискретизации

Система однократной передискретизации представлена на рис. 19.7.

Повышение или понижение частоты дискретизации на коэффициент передискретизации в виде рациональной дроби L/M реализуется каскадным соединением систем интерполяции с коэффициентом L и децимации с коэффициентом M (рис. 19.7, а). В результате объединения двух каскадно включенных ФНЧ — блоков $H_i(z)$ и $H_d(z)$, — работающих на одинаковой частоте дискретизации $L\omega_d$, переходят к системе однократной передискретизации с единственным ФНЧ (рис. 19.7, б). Его идеальная АЧХ, согласно (19.5) и (19.7), должна удовлетворять требованиям:

$$A(\omega) = \begin{cases} L & \text{в полосе пропускания } 0 \leq \omega \leq \min\{\omega_d/2; L\omega_d/2M\}; \\ 0 & \text{при других значениях } \omega. \end{cases} \quad (19.8)$$

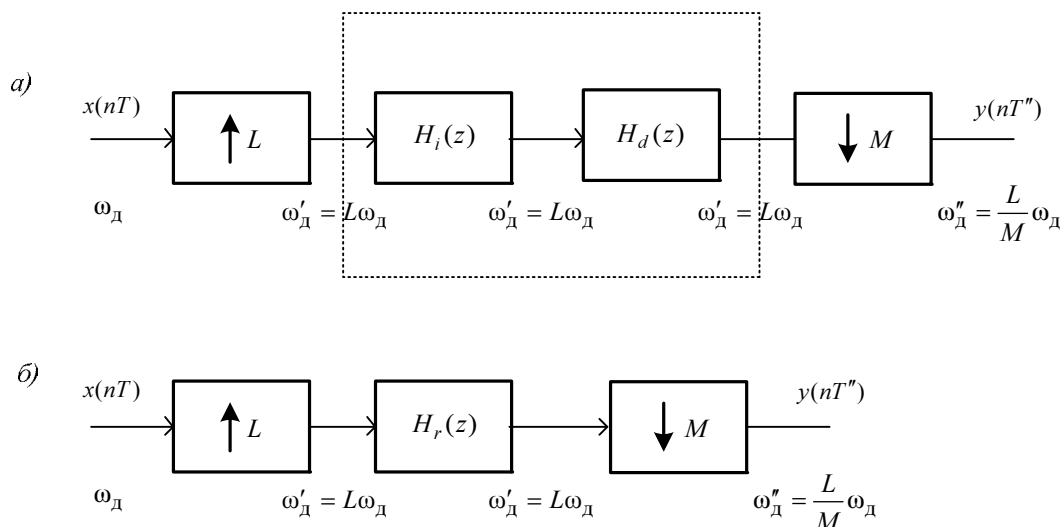


Рис. 19.7. Система однократной передискретизации с двумя ФНЧ (а) и одним ФНЧ (б)

В MATLAB моделирование системы однократной передискретизации реализуется с помощью функции:

```
[y,h] = resample(x,L,M[,I,beta])
```

где x — входной сигнал; L, M — коэффициенты интерполяции и децимации, а L/M — коэффициент передискретизации; I — необязательный параметр, управляющий длиной КИХ-фильтра ФНЧ, пропорциональной I ; по умолчанию $I = 10$.

По умолчанию КИХ-фильтр синтезируется методом окон с окном Кайзера. Гра-
ничная частота ПП АЧХ не превосходит $f_d/4$, где f_d — частота дискретизации
входного сигнала, при этом в ПП максимально допустимое отклонение АЧХ от 1
минимально (см. (19.8)).

beta — необязательный параметр окна Кайзера β ; по умолчанию $\beta=5$.

y — выходной сигнал, длина которого равна $\text{ceil}(\text{length}(x) * L/M)$; если отношение
длин сигналов вход/выход оказывается не целым числом, в ДПФ выходного сигнала
наблюдается эффект растекания спектра.

h — импульсная характеристика КИХ-фильтра ФНЧ.

19.1.4. Полифазные структуры
многоскоростных систем

Полифазные структуры систем однократной интерполяции и децимации предна-
значены для повышения быстродействия многоскоростных систем и основаны на
замещении одного сложного КИХ-фильтра ФНЧ, работающего на "высокой" частоте
дискретизации, эквивалентной системой более простых КИХ-фильтров ФНЧ,
работающих на "низкой" частоте.

Идея построения полифазных структур основана на использовании двух замеча-
тельных тождеств [4] (рис. 19.8).

Рассмотрим систему однократной интерполяции (см. рис. 19.1).

Замечательному тождеству на рис. 19.8, а соответствует тождество для соотноше-
ния вход/выход (см. (19.6)):

$$Y(z) = X(z^L)H(z) \equiv Y(z) = H(z^L)X(z).$$
 (19.9)

Оно означает, что сначала можно выполнить фильтрацию входного сигнала с "низ-
кой" частотой дискретизации КИХ-фильтром с передаточной функцией $H(z^L)$
с "низкой" частотой дискретизации, в L раз меньшей, чем у КИХ-фильтра с пере-
даточной функцией $H(z)$. Тем самым исключается обработка $(L-1)$ промежуточ-
ных нулей, которые затем, при переходе к "высокой" частоте дискретизации, до-
бавляются с помощью экспандера.

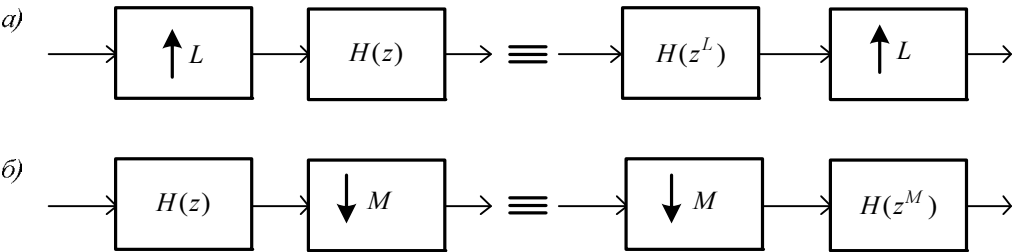


Рис. 19.8. Тождества для систем однократной интерполяции (а) и децимации (б)

Эта идея реализована в *полифазной* структуре однократной системы интерполяции. Для ее формирования запишем передаточную функцию КИХ-фильтра длины N :

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} = h_0z^{-1} + h_1z^{-2} + \dots + h_{L-1}z^{-L-1} + \dots + h_{N-1}z^{-(N-1)},$$

разобьем сумму из N слагаемых на $G = N/L$ (целое число) сумм из L слагаемых и используем матричное представление передаточной функции $\mathbf{H}(z)$ в виде произведения матрицы $\mathbf{A}$ размером $G \times L$ на вектор-столбец¹ $\mathbf{B}'$ длины L :

$$\mathbf{H}(z) = \begin{bmatrix} h_0 & h_1 & \dots & h_{L-1} \\ h_L z^{-L} & h_{L+1} z^{-L} & \dots & h_{L+(L-1)} z^{-L} \\ \vdots & \vdots & \vdots & \vdots \\ h_{(G-1)L} z^{-(G-1)L} & h_{(G-1)L+1} z^{-(G-1)L} & \dots & h_{(G-1)L+(L-1)} z^{-(G-1)L} \end{bmatrix} \times \begin{bmatrix} z^0 \\ z^{-1} \\ \vdots \\ z^{-(L-1)} \end{bmatrix} = \mathbf{AB}'.$$

Здесь $h_{(G-1)L+(L-1)} = h_{N-1}$ при $G \times L = N$.

Представим произведение $\mathbf{AB}'$ — вектор-столбец — в виде вектора-строки $\mathbf{BA}'$ с теми же элементами (см. разд. 2.1.4.1):

$$\mathbf{H}(z) = \begin{bmatrix} z^0 & z^{-1} & \dots & z^{-(L-1)} \end{bmatrix} \times \begin{bmatrix} h_0 & h_L z^{-L} & \dots & h_{(G-1)L} z^{-(G-1)L} \\ h_1 & h_{L+1} z^{-L} & \dots & h_{(G-1)L+1} z^{-(G-1)L} \\ \vdots & \vdots & \vdots & \vdots \\ h_{L-1} & h_{L+(L-1)} z^{-L} & \dots & h_{(G-1)L+(L-1)} z^{-(G-1)L} \end{bmatrix} = \mathbf{BA}'.$$

Здесь строки матрицы длины G соответствуют передаточным функциям $H_k(z^L)$, $k = 0, 1, \dots, L-1$, КИХ-фильтров длины G , что позволяет перейти к записи $\mathbf{H}(z)$ в виде произведения векторов:

$$\mathbf{H}(z) = \begin{bmatrix} z^0 & z^{-1} & \dots & z^{-(L-1)} \end{bmatrix} \times \begin{bmatrix} H_0(z^L) \\ H_1(z^L) \\ \vdots \\ H_{L-1}(z^L) \end{bmatrix} = \sum_{k=0}^{L-1} H_k(z^L) z^{-k}, \quad (19.10)$$

где $H_k(z^L)$ — k -я строка матрицы $\mathbf{A}'$:

$$H_k(z^L) = \sum_{n=0}^{G-1} h_{Ln+k} z^{-Ln}, \quad k = 0, 1, L-1.$$

¹ Символ "апостроф" соответствует операции транспонирования вектора или матрицы.

Представление передаточной функции $\mathbf{H}(z)$ в виде суммы (19.10) соответствует параллельной структуре КИХ-фильтров длины G с передаточными функциями $H_k(z^L)$, $k = 0, 1, \dots, L-1$, и базовым элементом задержки z^{-1} между ними (рис. 16.9). Такое представление $\mathbf{H}(z)$ называют *полифазным*, соответствующую структуру — *полифазной*, ее ветви — *фазами*, а КИХ-фильтры с передаточными функциями $H_k(z^L)$, $k = 0, 1, \dots, L-1$, — *полифазными фильтрами*.

Переход к "высокой" частоте дискретизации реализуется добавлением экспандеров *после* фильтрации (см. рис. 19.9).

Z -изображение *реакции* равно (см. (19.6) и (19.10))

$$Y(z) = X(z^L)H(z) = \sum_{k=0}^{L-1} X(z^L)H_k(z^L)z^{-k} = \sum_{k=0}^{L-1} Y_k(z^L)z^{-k}$$

и имеет полифазное представление, подобное (19.10).

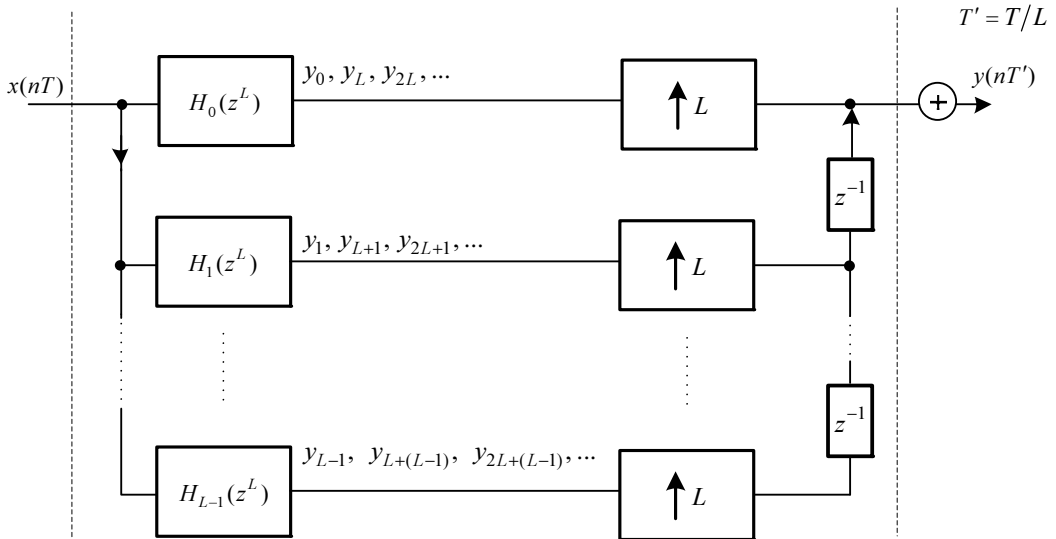


Рис. 19.9. Полифазная структура системы однократной интерполяции

Полифазную структуру *системы однократной децимации* можно получить из следующих соображений. Сравнивая системы однократной интерполяции и децимации в *правых* частях тождеств на рис. 19.8, *а, б*, видим, что при $L = M$ они *дуальны*, а следовательно, дуальны их полифазные структуры. В соответствии с принципом дуальности, меняя местами вход с выходом и направления всех стрелок в структуре на рис. 19.9, получаем полифазную структуру системы однократной децимации (рис. 19.10). В ней полифазное представление имеет z -изображение *воздействия*:

$$X(z) = \sum_{k=0}^{M-1} X_k(z^M)z^{-k}.$$

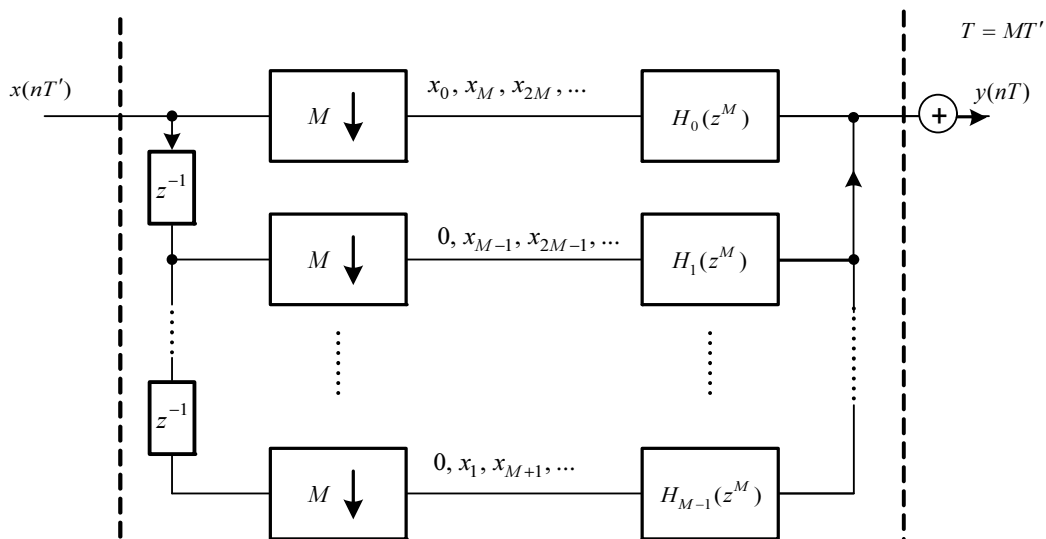


Рис. 19.10. Полифазная структура системы однократной децимации

Переход к "низкой" частоте дискретизации выходного сигнала реализуется добавлением компрессоров *перед* фильтрацией.

Полифазная структура *системы однократной передискретизации* реализуется каскадным соединением полифазных структур однократной системы интерполяции и децимации и их объединением.

В MATLAB полифазная структура, соответственно системы интерполяции, децимации и передискретизации описывается в виде объекта `mfilt` (от англ. *Multirate filter object*):

```

Hi = mfilt.firinterp(L[,Num])
Hd = mfilt.firdecim(M[,Num])
Hr = mfilt.firsrc(L,M[,Num])

```

где L , M — коэффициенты интерполяции и децимации; Num — коэффициенты передаточной функции КИХ-фильтра ФНЧ, работающего на "высокой" частоте дискретизации.

В отсутствии параметра Num автоматически синтезируется *фильтр Найквиста* с линейной ФЧХ, идеальная АЧХ которого отвечает следующим условиям:

□ для *системы однократной интерполяции*: в ПП равна L и достигает половинного значения (-6 дБ) в шкале нормированных частот $\hat{f} = \frac{f}{f_d/2}$ на частоте $1/L$ или

на частоте $\frac{f_d}{2L}$ в шкале абсолютных частот, где f_d — частота дискретизации входного сигнала;

□ для *системы однократной децимации*: в ПП равна единице и достигает половинного значения в шкале нормированных частот на частоте $1/M$ или на частоте

$\frac{f_d}{2M}$ в шкале абсолютных частот, где f_d — частота дискретизации входного сигнала;

- для системы *однократной передискретизации*: в ПП равна L и достигает половинного значения в шкале нормированных частот на частоте $1/\max\{L, M\}$.

Список свойств объекта `mfilt` включает в себя свойство `Numerator`, которое хранит коэффициенты КИХ-фильтра ФНЧ, работающего на "высокой" частоте дискретизации.

Представленным объектам `mfilt` соответствуют следующие структуры с КИХ-фильтрами¹ прямой структуры (см. табл. 11.2):

- Direct-Form FIR Polyphase Interpolator — полифазная структура системы интерполяции с КИХ-фильтрами, длиной `length(Hi.Numerator/L)` (целое число), работающими на "низкой" частоте дискретизации *входного* сигнала;
- Direct-Form FIR Polyphase Decimator — полифазная структура системы децимации с КИХ-фильтрами, длиной `length(Hd.Numerator/M)` (целое число), работающими на "низкой" частоте дискретизации *выходного* сигнала;
- Direct-Form FIR Polyphase Sample-Rate Converter — полифазная структура системы передискретизации с КИХ-фильтрами, длина и частота дискретизации которых зависит от соотношения L и M , а именно:
 - при $L > M$ длина КИХ-фильтров равна `length(Hr.Numerator/L)` и они работают на "низкой" частоте дискретизации *входного* сигнала;
 - при $L < M$ длина КИХ-фильтров равна `length(Hr.Numerator/M)` и они работают на "низкой" частоте дискретизации *выходного* сигнала.

Свойства объекта `mfilt` выводятся по его имени.

Моделирование многоскоростной системы с полифазной структурой выполняется на основе созданного объекта `mfilt` с помощью функции:

`y = filter(H,x)`

где H — имя объекта `mfilt`; x — входной сигнал (воздействие); y — выходной сигнал (реакция), длина которого зависит от типа многоскоростной системы и будет следующей:

- в L раз больше длины входного сигнала для системы однократной интерполяции;
- в M раз меньше длины входного сигнала для системы однократной децимации;
- равна `ceil(length(x)*L/M)` для системы однократной передискретизации.

В том случае, если в системе однократной *децимации* или *передискретизации* отношение длин сигналов вход/выход оказывается не *целым* числом, длина выходного сигнала автоматически округляется до ближайшего целого, при этом в ДПФ выходного сигнала наблюдается эффект растекания спектра.

¹ Полифазными фильтрами.

Необходимо иметь в виду, что при моделировании систем с полифазными структурами в реакции системы будет наблюдаться переходный процесс на интервале дискретного нормированного времени $n \in [0; n_{\text{нач}}]$, а в ДПФ реакции — эффект растекания спектра. Поэтому сравнивать реакцию с воздействием имеет смысл в установившемся режиме.

Автоматическое определение момента $n_{\text{нач}}$ — отдельная задача. На этапе моделирования при длине реакции, равной степени двойки и в отсутствии растекания спектра, момент времени $n_{\text{нач}}$ удобно определить с помощью функции `ceil(length(y)/2+1)`. В этом случае длина реакции в установившемся режиме останется равной степени двойки. Например, при длине реакции 256 получим $n_{\text{нач}} = 129$ и реакцию в установившемся режиме длины $256 - 129 + 1 = 128$, в ДПФ которой не будет растекания спектра.

При произвольной длине реакции, ее длина в установившемся режиме будет целым числом и в ДПФ ее реакции может наблюдаться эффект растекания спектра.

19.2. Содержание лабораторной работы

Содержание работы связано с моделированием систем однократной интерполяции, децимации и передискретизации, в том числе их полифазных структур, программными средствами MATLAB.

19.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файла `lr_19`, который хранится на прилагаемом компакт-диске в папке `LAB_DSP\LAB_19`.

Перед выполнением работы необходимо сохранить путь к папке `LAB_19` по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 19.1 для номера бригады $N_{\text{бр}}$, где $N_{\text{бр}} = 1, 2, \dots, 30$. Функция $N_{\text{бр}} \bmod M$ в записи исходных данных означает вычисление значения $N_{\text{бр}}$ по модулю M .

На прилагаемом компакт-диске в папке `Tables\Tables_19` хранятся табл. 19.1 исходных данных и пример ее заполнения для $N_{\text{бр}} = 1$.

Таблица 19.1. Таблица исходных данных

Переменная	Назначение	Значение	Идентификатор
$N_{\text{бр}}$	Номер бригады	$N_{\text{бр}}$	Nb =
A_1	Амплитуды дискретных гармоник входного сигнала	$A_1 = N_{\text{бр}} \bmod 3 + 1$	A1 =
A_2		$A_2 = 2A_1$	A2 =

Таблица 19.1 (продолжение)

Переменная	Назначение	Значение	Идентификатор
f_1	Частоты дискретных гармоник входного сигнала	$f_1 = 25(N_{\text{бп}} \bmod 5 + 1)$	f1 =
f_2		$f_2 = 2f_1$	f2 =
Система однократной интерполяции			
N_i	Период входного сигнала	$N_i = 32$	Ni = 32
$f_{\text{д}}^i$	Частота дискретизации входного сигнала	$f_{\text{д}}^i = 200(N_{\text{бп}} \bmod 5 + 1)$	Fs_i =
L	Коэффициент интерполяции	$L = 2, 4, 8$	Вектор L =[2 4 8]
Система однократной децимации			
N_d	Период входного сигнала	$N_d = 256$	Nd = 256
$f_{\text{д}}^d$	Частота дискретизации входного сигнала	$f_{\text{д}}^d = 1600(N_{\text{бп}} \bmod 5 + 1)$	Fs_d =
M	Коэффициент децимации	$M = 2, 4, 8$	Вектор M =[2 4 8]
Система однократной передискретизации			
N_r	Период входного сигнала	$N_r = 64$	Nr = 64
$f_{\text{д}}^r$	Частота дискретизации входного сигнала	$f_{\text{д}}^r = 400(N_{\text{бп}} \bmod 5 + 1)$	Fs_r =
L/M	Коэффициент передискретизации	$L/M = 5/2$	L = 5 M = 2
Система однократной интерполяции с полифазной структурой			
N_{ip}	Период входного сигнала	$N_{ip} = 32$	Nip = 32
$f_{\text{д}}^{ip}$	Частота дискретизации входного сигнала	$f_{\text{д}}^{ip} = 200(N_{\text{бп}} \bmod 5 + 1)$	Fs_ip =
L	Коэффициент интерполяции	$L = 4$	L = 4
Система однократной децимации с полифазной структурой			
N_{dp}	Период входного сигнала	$N_{dp} = 256$	Ndp = 256

Таблица 19.1 (окончание)

Переменная	Назначение	Значение	Идентификатор
f_d^{dp}	Частота дискретизации входного сигнала	$f_d^{dp} = 1600(N_{\text{бр}} \bmod 5 + 1)$	Fs_dp =
M	Коэффициент децимации	$M = 4$	M = 4

Задание на лабораторную работу заключается в моделировании многоскоростных систем и включает в себя следующие пункты:

1. Моделирование системы однократной интерполяции.

В качестве входного сигнала системы однократной интерполяции выбрать периодическую последовательность $x(n)$ (идентификатор x) с периодом N (идентификатор N) и частотой дискретизации f_d (идентификатор Fs):

$$x(n) = A_1 \sin\left(\frac{2\pi f_1}{f_d} n\right) + A_2 \sin\left(\frac{2\pi f_2}{f_d} n\right).$$

(19.11)

Ввести значения $N = N_i$, $f_d = f_d^i$ и коэффициенты интерполяции $L = 2, 4, 8$ (вектор L).

Вывести графики:

- входного сигнала $x(n)$;
- выходных сигналов $y_i(n)$, $i = 1, 2, 3$, (массив ячеек y) при $L = 2, 4, 8$.

Здесь и далее для организации цикла с вычислением векторов различных длин используется массив ячеек (cell array), элементами которого являются данные векторы (см. разд. 3.1.4);

- импульсных характеристик КИХ-фильтров ФНЧ $h_i(n)$, $i = 1, 2, 3$, (массив ячеек h) при $L = 2, 4, 8$.

Для функции interp значения входных параметров I и fx установить по умолчанию.

Пояснить связь между:

- периодами дискретизации, длинами и длительностями входного и выходного сигналов;
- длиной КИХ-фильтра и коэффициентом интерполяции;
- видом импульсной характеристики и видом ФЧХ КИХ-фильтра.

2. Вычисление амплитудных спектров сигналов и АЧХ КИХ-фильтров системы однократной интерполяции.

Вывести графики:

- амплитудного спектра¹ (идентификатор `MODX`) входного сигнала $x(n)$ на периоде в шкале частот f (Гц);
- амплитудных спектров (массив ячеек `MODY`) выходных сигналов $y_i(n)$, $i = 1, 2, 3$, на периоде амплитудного спектра входного сигнала (т. е. в шкале "новых" частот f' (см. рис. 19.3, а, б)) и одновременно АЧХ КИХ-фильтров² (идентификатор `MAG`) с помощью функции `plot` красным цветом.

Вывести те же графики, каждый на своем периоде в шкале "старых" частот f (см. рис. 19.3, а, д) в одинаковом диапазоне $[0 \text{ max}(L) * Fs]$ с помощью функции `xlim`.

Пояснить:

- вид амплитудных спектров входного и выходного сигналов в шкале "новых" и "старых" частот;
- каким требованиям должна удовлетворять идеальная АЧХ КИХ-фильтра.

3. Моделирование системы однократной децимации.

В качестве входного сигнала системы однократной децимации выбрать $x(n)$ (19.11).

Ввести значения $N = N_d$, $f_d = f_d^d$ и коэффициенты децимации $M = 2, 4, 8$ (вектор `M`).

Вывести графики:

- входного сигнала $x(n)$ (идентификатор `x`);
- выходных сигналов $y_i(n)$, $i = 1, 2, 3$, (массив ячеек `y`) при $M = 2, 4, 8$.

Для функции `decimate` выбрать КИХ-фильтр с порядком, установленным по умолчанию.

Пояснить связь между периодами дискретизации, длинами и длительностями входного и выходного сигналов.

4. Вычисление амплитудных спектров сигналов системы однократной децимации.

Вывести графики в шкале частот f (Гц) в одинаковом диапазоне $[0 \text{ Fs}]$:

- амплитудного спектра входного сигнала $x(n)$ (идентификатор `MODX`) на периоде;
- амплитудных спектров (массив ячеек `MODY`) выходных сигналов $y_i(n)$, $i = 1, 2, 3$, каждый на своем периоде.

¹ Вычислить с помощью функции `fft` (см. разд. 9.1.1).

² Вычислить с помощью функции `freqz` (см. разд. 8.1.3).

Пояснить:

- вид амплитудных спектров входного и выходного сигналов;
- выполняется ли условие теоремы Котельникова для новой частоты дискретизации $f'_d = f_d/M$;
- при каком значении M и по какой причине в спектре выходного сигнала будет наблюдаться эффект растекания спектра;
- каким требованиям должна удовлетворять идеальная АЧХ КИХ-фильтра.

5. Моделирование системы однократной передискретизации.

В качестве входного сигнала системы однократной передискретизации выбрать $x(n)$ (19.11). Ввести значения $N = N_r$, $f_d = f_d^r$ и $L = 5$, $M = 2$ для коэффициента передискретизации L/M .

Вывести графики:

- входного сигнала $x(n)$ (идентификатор x);
- выходного сигнала $y(n)$ (идентификатор y).

Для функции `resample` значения входных параметров `I` и `beta` установить по умолчанию.

Пояснить связь между периодами дискретизации, длинами и длительностями входного и выходного сигналов.

6. Вычисление амплитудных спектров сигналов и АЧХ КИХ-фильтра системы однократной передискретизации.

Вывести графики в шкале частот f (Гц) в одинаковом диапазоне $[0 \text{ } (L/M) * F_s]$:

- амплитудного спектра (идентификатор `MODX`) входного сигнала $x(n)$ на периоде;
- амплитудного спектра (идентификатор `MODY`) выходного сигнала $y(n)$ на периоде и одновременно АЧХ КИХ-фильтра (идентификатор `MAG`) с помощью функции `plot` красным цветом.

Пояснить:

- вид амплитудных спектров входного и выходного сигналов;
- каким требованиям должна удовлетворять идеальная АЧХ КИХ-фильтра.

7. Моделирование системы однократной интерполяции с полифазной структурой.

В качестве входного сигнала системы однократной интерполяции с полифазной структурой выбрать $x(n)$ (19.11). Ввести значения $N = N_{ip}$, $f_d = f_d^{ip}$ и коэффициент интерполяции $L = 4$.

Описать полифазную структуру в виде объекта `mfilt` с именем `hi` и вывести его свойства.

Вывести графики:

- входного сигнала $x(n)$ (идентификатор `x`);
- выходного сигнала $y(n)$ (идентификатор `y`);
- выходного сигнала $y(n)$ в установившемся режиме (идентификатор `y1`) с автоматическим определением момента его начала $n_{\text{нач}}$ (идентификатор `ni_start`).

Вывести графики в шкале частот f (Гц) в одинаковом диапазоне $[0 \text{ } L \cdot F_s]$:

- амплитудного спектра (идентификатор `MODX`) входного сигнала $x(n)$ на периоде;
- амплитудного спектра (идентификатор `MODY`) выходного сигнала $y(n)$ в установившемся режиме на периоде и одновременно АЧХ КИХ-фильтра (идентификатор `MAG`), работающего на "высокой" частоте дискретизации с помощью функции `plot` красным цветом.

Пояснить:

- связь между периодами и частотами дискретизации входного и выходного сигналов;
- почему входной и выходной сигналы и их амплитудные спектры имеет смысл сравнивать в установившемся режиме;
- чему равна длина КИХ-фильтра, работающего на "высокой" частоте дискретизации выходного сигнала;
- какое количество КИХ-фильтров используется в полифазной структуре, чему равна их длина и на какой частоте они работают.

8. Моделирование системы однократной децимации с полифазной структурой.

В качестве входного сигнала системы однократной децимации с полифазной структурой выбрать $x(n)$ (19.11). Ввести значения $N = N_{dp}$, $f_d = f_d^{dp}$ и коэффициент децимации $M = 4$.

Описать полифазную структуру в виде объекта `mfilt` с именем `hd` и вывести его свойства.

Вывести графики:

- входного сигнала $x(n)$ (идентификатор `x`);
- выходного сигнала $y(n)$ при $M = 4$ (идентификатор `y`);
- выходного сигнала $y(n)$ в установившемся режиме (идентификатор `yd`) с автоматическим определением момента его начала $n_{\text{нач}}$ (идентификатор `nd_start`).

Вывести графики в шкале частот f (Гц) в одинаковом диапазоне $[0 \text{ } F_s]$:

- амплитудного спектра (идентификатор `MODX`) входного сигнала $x(n)$ на периоде и одновременно АЧХ КИХ-фильтра (идентификатор `MAG`), работающего на "высокой" частоте дискретизации с помощью функции `plot` красным цветом;
- амплитудного спектра выходного сигнала $y(n)$ в установившемся режиме на периоде (идентификатор `MODY`).

Пояснить:

- связь между периодами и частотами дискретизации входного и выходного сигналов;
- чему равна длина КИХ-фильтра, работающего на "высокой" частоте дискретизации входного сигнала;
- какое количество КИХ-фильтров используется в полифазной структуре, чему равна их длина и на какой частоте они работают.

19.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должна быть представлена табл. 19.1 исходных данных для своего номера бригады $N_{бр}$.

Для *запуска* лабораторной работы необходимо обратиться к script-файлу `lr_19` по его имени:

```
>> lr_19
```

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш `<Ctrl>+<Break>`.

При выполнении script-файла текущие окна с графиками *не закрывать*.

Листинг script-файла `lr_19` имеет вид:

```
>> type lr_19
script
clc
clear
disp('% ЛР №19. МНОГОСКОРОСТНЫЕ СИСТЕМЫ ЦОС')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Nb = input('Nb = '); % НОМЕР БРИГАДЫ
A1 = input('A1 = '); % АМПЛИТУДЫ ДИСКРЕТНЫХ ГАРМОНИК ВХОДНОГО СИГНАЛА
A2 = input('A2 = ');
```

```

f1 = input('f1 = ');      % ЧАСТОТЫ (Гц) ДИСКРЕТНЫХ ГАРМОНИК ВХОДНОГО СИГНАЛА
f2 = input('f2 = ');
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA=input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.1. МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ИНТЕРПОЛЯЦИИ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Ni = input('Ni = ');      % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs_i = input('Fs_i = ');  % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
L = input('L = ');        % ВЕКТОР КОЭФФИЦИЕНТОВ ИНТЕРПОЛЯЦИИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('% Для вывода значений Ni, Fs_i и L нажмите <ENTER>')
pause
disp('%')
disp(['      N = Ni = ',num2str(Ni),'      Fs = Fs_i = ',num2str(Fs_i)])
disp('%')
disp(['      L = [',num2str(L) ' ]'])
disp('%')
disp('% Для вывода ГРАФИКОВ ВХОДНОГО и ВЫХОДНЫХ сигналов нажмите <ENTER>')
pause
N = Ni;      % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs = Fs_i;   % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
n = 0:(N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ВХОДНОГО СИГНАЛА
x = A1*sin((2*pi*f1/Fs).*n)+A2*sin((2*pi*f2/Fs).*n);% ВХОДНОЙ СИГНАЛ
figure('Name','Input and Output Signals in Interpolation System','NumberTitle',
'off')
subplot(4,1,1), stem(n,x), grid, xlabel('n')
title(strcat(['Input Signal x(n)      N = ',num2str(N)]))
for i = 1:length(L)      % ИНДЕКСЫ ВЕКТОРА L
[y{i},h{i}] = interp(x,L(i)); % ВЫХОДНОЙ СИГНАЛ И ИХ КИХ-ФИЛЬТРА (cell
array)

```

```

subplot(4,1,i+1), stem(0:length(y{i})-1,y{i}), grid, xlabel('n')
title(strcat(['L = ',num2str(L(i)), '    Output Signal y(n)    length(y) = ',num2str(length(y{i}))]))
end
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИМПУЛЬСНЫХ ХАРАКТЕРИСТИК КИХ-фильтров нажмите <ENTER>')
pause
figure('Name','Impulse Responses FIR in Interpolation System','NumberTitle','off')
for i = 1:length(L) % ИНДЕКСЫ ВЕКТОРА L
    subplot(3,1,i), stem(0:length(h{i})-1, h{i}, 'MarkerSize',3,'Linewidth',2),
    grid, xlabel('n')
    title(strcat(['L = ',num2str(L(i)), '    Impulse Response h(n)    length(h) = ',num2str(length(h{i}))]))
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. ВЫЧИСЛЕНИЕ АМПЛИТУДНЫХ СПЕКТРОВ СИГНАЛОВ И АЧХ КИХ-ФИЛЬТРОВ')
disp('% СИСТЕМЫ ОДНОКРАТНОЙ ИНТЕРПОЛЯЦИИ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ и АЧХ КИХ-фильтров')
disp('% в "НОВОЙ" шкале частот нажмите <ENTER>')
pause
X = fft(x); % ДПФ ВХОДНОГО СИГНАЛА
MODX = (2/N)*abs(X); % АМПЛИТУДНЫЙ СПЕКТР ВХОДНОГО СИГНАЛА
MODX(1) = (1/N)*abs(X(1));
k = 0:N-1; % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВХОДНОГО СИГНАЛА
figure('Name','Amplitude Spectrums and Magnitude Responses ("new frequencies)','NumberTitle','off')
subplot(4,1,1), stem(k*(Fs/N),MODX,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), title(strcat(['Amplitude Spectrum x(n)    N = ',num2str(N)]))
f = 0:Fs/1000:Fs; % СЕТКА ЧАСТОТ ДЛЯ АЧХ КИХ-ФИЛЬТРА В "НОВОЙ" ШКАЛЕ
for i = 1:length(L) % ИНДЕКСЫ ВЕКТОРА L
    Y{i} = fft(y{i}); % ДПФ ВЫХОДНОГО СИГНАЛА(cell array)
    Y_D = Y{i}; % ДПФ ВЫХОДНОГО СИГНАЛА(double)
    MODY_D = (2/length(Y_D))*abs(Y_D); % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА(double)
    MODY_D(1) = (1/length(Y_D))*abs(Y_D(1));
    MODY{i} = MODY_D; % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА(cell array)
    MAG = abs(freqz(h{i},1,f,Fs)); % АЧХ КИХ-ФИЛЬТРА

```

```

k_out = 0:length(Y{i})-1;           % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВЫХОДНОГО
СИГНАЛА
subplot(4,1,i+1),
stem(k_out*(Fs/length(Y{i})),MODY{i},'MarkerSize',3,'Linewidth',2)
grid, xlabel('f ' (Hz) ("new" frequencies'))
title(strcat(['L = ',num2str(L(i)),' Amplitude Spectrum y(n) and FIR
Magnitude Response']))
hold on, plot(f,MAG,'r','Linewidth',2)
end
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ и АЧХ КИХ-фильтров')
disp('% в "СТАРОЙ" шкале частот нажмите <ENTER>')
pause
figure('Name','Amplitude Spectrums and Magnitude Responses ("old"
frequencies)','NumberTitle','off')
subplot(4,1,1), stem(k*(Fs/N),MODX,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 max(L)*Fs])
title(strcat(['Amplitude Spectrum x(n) N=',num2str(N)]))
for i = 1:length(L)                 % ИНДЕКСЫ ВЕКТОРА L
f = 0:(Fs*L(i))/1000:(Fs*L(i));    % СЕТКА ЧАСТОТ ДЛЯ АЧХ КИХ-ФИЛЬТРА
В "СТАРОЙ" ШКАЛЕ
MAG = abs(freqz(h{i},1,f,(Fs*L(i)))); % АЧХ КИХ-ФИЛЬТРА
k_out = 0:length(Y{i})-1;          % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ
ВЫХОДНОГО СИГНАЛА
subplot(4,1,i+1),stem(k_out*(Fs*L(i)/length(Y{i})),MODY{i},'MarkerSize',3,'Line
width',2)
grid, xlabel('f (Hz) ("old" frequencies)'), xlim([0 max(L)*Fs])
title(strcat(['L = ',num2str(L(i)),' Amplitude Spectrum y(n) and FIR
Magnitude Response']))
hold on, plot(f,MAG,'r','Linewidth',2)
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.3. МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ДЕЦИМАЦИИ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Nd = input('Nd = ');              % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs_d = input('Fs_d = ');          % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
M = input('M = ');                % ВЕКТОР КОЭФФИЦИЕНТОВ ДЕЦИМАЦИИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')

```

```

disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('% Для вывода значений Nd, Fs_d и M нажмите <ENTER>')
pause
disp('%')
disp(['          N = Nd = ',num2str(Nd),'          Fs = Fs_d = ',num2str(Fs_d)])
disp('%')
disp(['          M = ',num2str(M) ''])
disp('%')
disp('% Для вывода ГРАФИКОВ ВХОДНОГО и ВЫХОДНЫХ сигналов нажмите <ENTER>')
pause
N = Nd;          % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs = Fs_d;       % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
n = 0:(N-1);     % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ВХОДНОГО СИГНАЛА
x = A1*sin((2*pi*f1/Fs).*n)+A2*sin((2*pi*f2/Fs).*n); % ВХОДНОЙ СИГНАЛ
figure('Name','Input and Output Signals in Decimation System','NumberTitle',
'off')
subplot(4,1,1), stem(n,x), grid, xlabel('n')
title(strcat(['Input Signal x(n)          N = ',num2str(N)]))
for i = 1:length(M)          % ИНДЕКСЫ ВЕКТОРА M
y{i} = decimate(x,M(i),'fir'); % ВЫХОДНОЙ СИГНАЛ(cell array)
subplot(4,1,i+1), stem(0:length(y{i})-1,y{i}), grid, xlabel('n')
title(strcat(['M = ',num2str(M(i))','      Output Signal y(n)      length(y) =
',num2str(length(y{i}))]))
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. ВЫЧИСЛЕНИЕ АМПЛИТУДНЫХ СПЕКТРОВ СИГНАЛОВ')
disp('% СИСТЕМЫ ОДНОКРАТНОЙ ДЕЦИМАЦИИ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ нажмите <ENTER>')
pause
X = fft(x);          % ДПФ ВХОДНОГО СИГНАЛА
MODX = (2/N)*abs(X); % АМПЛИТУДНЫЙ СПЕКТР ВХОДНОГО СИГНАЛА
MODX(1) = (1/N)*abs(X(1));
k = 0:N-1;          % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВХОДНОГО СИГНАЛА
figure('Name','Amplitude Spectrums in Decimation System','NumberTitle', 'off')
subplot(4,1,1), stem(k*(Fs/N),MODX,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'),title(strcat(['Amplitude Spectrum x(n)
N = ',num2str(N)])), xlim([0 Fs])

```

```

for i = 1:length(M)                                % ИНДЕКСЫ ВЕКТОРА M
Y{i} = fft(y{i});                                  % ДПФ ВЫХОДНОГО СИГНАЛА
Y_D = Y{i};                                         % ДПФ ВЫХОДНОГО СИГНАЛА(double)
MODY_D = (2/length(Y_D))*abs(Y_D);                 % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО
СИГНАЛА(double)
MODY_D(1) = (1/length(Y_D))*abs(Y_D(1));
MODY{i} = MODY_D; % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА(cell array)
k_out = 0:length(Y{i})-1;                          % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ
ВЫХОДНОГО СИГНАЛА
subplot(4,1,i+1),stem(k_out*(Fs/M(i)/length(Y{i})),MODY{i},'MarkerSize',3,
'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 Fs])
title(strcat(['M = ',num2str(M(i)),'   Amplitude Spectrum y(n)']))
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ПЕРЕДИСКРЕТИЗАЦИИ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Nr = input('Nr = ');                               % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs_r = input('Fs_r = ');                            % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
L = input('L = ');                                  % КОЭФФИЦИЕНТ ИНТЕРПОЛЯЦИИ
M = input('M = ');                                  % КОЭФФИЦИЕНТ ДЕЦИМАЦИИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('% Для вывода значений Nr, Fs_r и L/M нажмите <ENTER>')
pause
disp('%')
disp(['      N = Nr = ',num2str(Nr),'      Fs = Fs_r = ',num2str(Fs_r)])
disp('%')
disp(['      L/M = ',num2str(L),'/',num2str(M)])
disp('%')
disp('% Для вывода ГРАФИКОВ ВХОДНОГО и ВЫХОДНЫХ сигналов нажмите <ENTER>')
pause
N = Nr;                                              % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs = Fs_r;                                           % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА

```

```

n = 0:(N-1);           % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ВХОДНОГО СИГНАЛА
x = A1*sin((2*pi*f1/Fs).*n)+A2*sin((2*pi*f2/Fs).*n); % ВХОДНОЙ СИГНАЛ
figure('Name','Input and Output Signals in Resampling System','NumberTitle',
'off')
subplot(2,1,1), stem(n,x), grid, xlabel('n')
title(strcat(['Input Signal x(n)      N = ',num2str(N)]))
[y h] = resample(x,L,M);           % ВЫХОДНОЙ СИГНАЛ И ИХ КИХ-ФИЛЬТРА
subplot(2,1,2), stem(0:length(y)-1,y), grid, xlabel('n')
title(strcat(['L/M = ',num2str(L),'/',num2str(M),' Output Signal y(n)
length(y) = ',num2str(length(y))]))
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ВЫЧИСЛЕНИЕ АМПЛИТУДНЫХ СПЕКТРОВ СИГНАЛОВ И АЧХ КИХ-ФИЛЬТРА')
disp('% СИСТЕМЫ ОДНОКРАТНОЙ ПЕРЕДИСКРЕТИЗАЦИИ')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ и АЧХ КИХ-фильтра нажмите
<ENTER>')
pause
figure('Name','Amplitude Spectrums and Magnitude Responses in Resampling
System','NumberTitle','off')
X = fft(x);           % ДПФ ВХОДНОГО СИГНАЛА
MODX = (2/N)*abs(X);   % АМПЛИТУДНЫЙ СПЕКТР ВХОДНОГО СИГНАЛА
MODX(1) = (1/N)*abs(X(1));
k = 0:N-1;           % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВХОДНОГО СИГНАЛА
subplot(2,1,1),stem(k*(Fs/N),MODX,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), title(strcat(['Amplitude Spectrum x(n)      N =
',num2str(N)])), xlim([0 (L/M)*Fs])
Y = fft(y);           % ДПФ ВЫХОДНОГО СИГНАЛА
MODY = (2/length(Y))*abs(Y); % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА
MODY(1) = (1/length(Y))*abs(Y(1));
k_out = 0:length(Y)-1; % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ
ВЫХОДНОГО СИГНАЛА
f = 0:Fs*(L/M)/1000:(Fs*(L/M)); % СЕТКА ЧАСТОТ ДЛЯ АЧХ КИХ-ФИЛЬТРА
MAG = abs(freqz(h,1,f,(Fs*(L/M)))); % АЧХ КИХ-ФИЛЬТРА
subplot(2,1,2), stem(k_out*(Fs*(L/M)/length(Y)),MODY,'MarkerSize',3,
'Linewidth',2)
grid, xlabel('f (Hz)')
title(strcat(['L/M = ',num2str(L),'/',num2str(M),' Amplitude Spectrum y(n) and
FIR Magnitude Response']))
hold on, plot(f,MAG,'r','Linewidth',2), xlim([0 (L/M)*Fs])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')

```



```

pause
disp('%')
disp('%')
disp('% п.7. МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ИНТЕРПОЛЯЦИИ С ПОЛИФАЗНОЙ
СТРУКТУРОЙ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Nip = input('Nip = ');           % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs_ip = input('Fs_ip = ');       % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
L = input('L = ');               % КОЭФФИЦИЕНТ ИНТЕРПОЛЯЦИИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('% Для вывода значений Ni, Fs_ip и L нажмите <ENTER>')
pause
disp('%')
disp(['          N = Nip = ',num2str(Nip),'          Fs = Fs_ip = ',num2str(Fs_ip)])
disp('%')
disp(['          L = ',num2str(L)])
N = Nip;                         % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs = Fs_ip;                       % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
n = 0:(N-1);                     % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ВХОДНОГО СИГНАЛА
x = A1*sin((2*pi*f1/Fs).*n)+A2*sin((2*pi*f2/Fs).*n); % ВХОДНОЙ СИГНАЛ
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА Ni нажмите <ENTER>')
pause
Ni = mfilt.firinterp(L)           % ОПИСАНИЕ ПОЛИФАЗНОЙ СТРУКТУРЫ СИСТЕМЫ
ОДНОКРАТНОЙ ИНТЕРПОЛЯЦИИ
y = filter(Ni,x);                 % ВЫХОДНОЙ СИГНАЛ
ni_start = ceil(length(y)/2+1);  % НАЧАЛО УСТАНОВИВШЕГОСЯ РЕЖИМА
yi = y(ni_start:end);            % ВЫХОДНОЙ СИГНАЛ В УСТАНОВИВШЕМСЯ РЕЖИМЕ
ni = ni_start:(ni_start+length(yi)-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
ДЛЯ УСТАНОВИВШЕГОСЯ РЕЖИМА
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ВХОДНОГО и ВЫХОДНОГО сигналов нажмите <ENTER>')
pause
figure('Name','Input and Output Signals in Polyphase Structure Interpolation
System','NumberTitle','off')
subplot(3,1,1), stem(n,x), grid, xlabel('n')
title(strcat(['Input Signal x(n)          N = ',num2str(N)]))

```

```

subplot(3,1,2), stem(0:length(y)-1,y), grid
title(strcat(['L = ',num2str(L),'   Output Signal y(n)   length(y) = ',num2str(length(y))]))
subplot(3,1,3), stem(ni,yi), grid, xlabel('n')
title(strcat(['Output Signal y(n)   Starting point n = ',num2str(ni_start),' length(y) = ',num2str(length(yi))]))
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ и АЧХ КИХ-фильтра нажмите <ENTER>')
pause
X = fft(x); % ДПФ ВХОДНОГО СИГНАЛА
MODX = (2/N)*abs(X); % АМПЛИТУДНЫЙ СПЕКТР ВХОДНОГО СИГНАЛА
MODX(1) = (1/N)*abs(X(1));
Yi = fft(yi); % ДПФ ВЫХОДНОГО СИГНАЛА
MODYi = (2/length(Yi))*abs(Yi); % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА
MODYi(1) = (1/length(Yi))*abs(Yi(1));
k = 0:(N-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВХОДНОГО СИГНАЛА
ki = 0:(length(Yi)-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВЫХОДНОГО СИГНАЛА
f = 0:(Fs*L)/1000:F*s*L; % СЕТКА ЧАСТОТ ДЛЯ АЧХ КИХ-ФИЛЬТРА
h = Hi.Numerator; % ИХ КИХ-ФИЛЬТРА
MAG = abs(freqz(h,1,f,Fs*L)); % АЧХ КИХ-ФИЛЬТРА
figure('Name','Amplitude Spectrums and Magnitude Response in Polyphase Structure Interpolation System','NumberTitle','off')
subplot(2,1,1), stem(k*(Fs/N),MODX,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 L*Fs])
title(strcat(['Amplitude Spectrum x(n)   N = ',num2str(N)]))
subplot(2,1,2), stem(ki*(Fs*L/length(Yi)),MODYi,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 L*Fs])
title(strcat(['L = ',num2str(L),'   Amplitude Spectrum y(n) and FIR Magnitude Response']))
hold on, plot(f,MAG,'r','Linewidth',2)
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('%')
disp('% п.8. МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ДЕЦИМАЦИИ С ПОЛИФАЗНОЙ СТРУКТУРОЙ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Ndp = input('Ndp = '); % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs_dp = input('Fs_dp = '); % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
M = input('M = '); % КОЭФФИЦИЕНТ ДЕЦИМАЦИИ

```

```

disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('% Для вывода значений Ndp, Fs_dp и M нажмите <ENTER>')
pause
disp('%')
disp(['      N = Ndp = ', num2str(Ndp), '      Fs = Fs_dp = ', num2str(Fs_dp)])
disp('%')
disp(['      M = ', num2str(M)])
N = Ndp;           % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs = Fs_dp;        % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
n = 0:(N-1);       % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ВХОДНОГО СИГНАЛА
x = A1*sin((2*pi*f1/Fs).*n)+A2*sin((2*pi*f2/Fs).*n);   % ВХОДНОЙ СИГНАЛ
disp('%')
disp('% Для вывода СВОЙСТВ ОБЪЕКТА Hd нажмите <ENTER>')
pause
Hd = mfilterdecim(M) % ОПИСАНИЕ ПОЛИФАЗНОЙ СТРУКТУРЫ СИСТЕМЫ ОДНОКРАТНОЙ
ДЕЦИМАЦИИ
y = filter(Hd,x);           % ВЫХОДНОЙ СИГНАЛ
nd_start = ceil(length(y)/2+1); % НАЧАЛО УСТАНОВИВШЕГОСЯ РЕЖИМА
yd = y(nd_start:end);       % ВЫХОДНОЙ СИГНАЛ В УСТАНОВИВШЕМСЯ РЕЖИМЕ
nd = nd_start:(nd_start+length(yd)-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
ДЛЯ УСТАНОВИВШЕГОСЯ РЕЖИМА
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ВХОДНОГО и ВЫХОДНОГО сигналов нажмите <ENTER>')
pause
figure('Name','Input and Output Signals in Polyphase Structure Decimation
System','NumberTitle','off')
subplot(3,1,1), stem(n,x), grid, xlabel('n')
title(strcat(['Input Signal x(n)   N = ', num2str(N)]))
subplot(3,1,2), stem(0:length(y)-1,y), grid
title(strcat(['M = ', num2str(M), '   Output Signal y(n)   length(y) = ',
num2str(length(y))]))
subplot(3,1,3), stem(nd,yd), grid, xlabel('n')
title(strcat(['Output Signal y(n)   Starting point n = ', num2str(nd_start), '
length(y) = ', num2str(length(yd))]))
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ и АЧХ КИХ-фильтра нажмите
<ENTER>')
pause
X = fft(x);           % ДПФ ВХОДНОГО СИГНАЛА
MODX = (2/N)*abs(X);   % АМПЛИТУДНЫЙ СПЕКТР ВХОДНОГО СИГНАЛА
MODX(1) = (1/N)*abs(X(1));

```

```

Yd = fft(Yd); % ДПФ ВЫХОДНОГО СИГНАЛА В УСТАНОВИВШЕМСЯ РЕЖИМЕ
MODYd = (2/length(Yd))*abs(Yd); % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА
MODYd(1) = (1/length(Yd))*abs(Yd(1));
k = 0:(N-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВХОДНОГО СИГНАЛА
kd = 0:(length(Yd)-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВЫХОДНОГО СИГНАЛА
f = 0:Fs/1000:Fs; % СЕТКА ЧАСТОТ ДЛЯ АЧХ КИХ-ФИЛЬТРА
hd = Hd.Numerator; % ИХ КИХ-ФИЛЬТРА
MAG = abs(freqz(hd,1,f,Fs)); % АЧХ КИХ-ФИЛЬТРА
figure('Name','Amplitude Spectrums and Magnitude Response in Polyphase
Structure Decimation System','NumberTitle','off')
subplot(2,1,1), stem(k*(Fs/N),MODX,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 Fs])
title(strcat('Amplitude Spectrum x(n) and FIR Magnitude Response'))
hold on, plot(f,MAG,'r','Linewidth',2)
subplot(2,1,2), stem(kd*((Fs/M)/length(Yd)),MODYd,'MarkerSize',3,
'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 Fs])
title(strcat(['M = ',num2str(M),' Amplitude Spectrum y(n)']))
disp('%')
disp('%')
disp('% РАБОТА ЗАВЕРШЕНА')

```

19.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файлов для моделирования многоскоростных систем с использованием исходных данных из табл. 19.1 для своего номера бригады $N_{бр}$.

Пункты самостоятельного задания включает в себя:

1С. Моделирование системы однократной интерполяции.

Исследовать влияние параметров I и f_x функции `interp` на АЧХ КИХ-фильтра.

Выполнить для входного сигнала $x(n)$ (19.11) при $N = N_i$, $f_d = f_d^i$ и $f_2 = 3f_1$ для коэффициента интерполяции $L = 4$.

Включить параметры N , L , F_s , I и f_x в список входных параметров function-файла и проанализировать:

- при каком значении I ($f_x = 0.5$ по умолчанию) гарантируется попадание частот f_1 и f_2 в ПП АЧХ;
- чему в этом случае равно максимально допустимое значение I ;
- при каком значении f_x ($I = 4$ по умолчанию) гарантируется попадание частот f_1 и f_2 в ПП АЧХ;
- к каким изменениям АЧХ приводит увеличение и уменьшение значения f_x .

Вывести графики в шкале частот f (Гц):

- амплитудного спектра входного сигнала на периоде;
- амплитудного спектра выходного сигнала на периоде и одновременно АЧХ КИХ-фильтра с помощью функции `plot` красным цветом.

2С. Моделирование системы однократной децимации.

Исследовать влияние частоты дискретизации входного сигнала f_d на амплитудный спектр выходного сигнала.

Выполнить для входного сигнала $x(n)$ (19.11) при $N = N_d$ и $f_d = f_d^d$ для коэффициента децимации $M = 4$.

Включить параметры N , M и F_s в список входных параметров `function`-файла и проанализировать:

- к каким изменениям в амплитудном спектре выходного сигнала приводит уменьшение частоты дискретизации входного сигнала: $f_d = f_d^d / 2^v$, где $v = 1, 2, 3, \dots$;
- при каком значении f_d не будет выполняться условие теоремы Котельникова ($f_d \geq 2f_v$) для новой частоты дискретизации $f_d' = f_d / M$.

Вывести графики в шкале частот f (Гц):

- амплитудного спектра входного сигнала на периоде;
- амплитудного спектра выходного сигнала на периоде.

3С. Моделирование систем однократной и многократной интерполяции.

Выполнить моделирование системы однократной интерполяции для входного сигнала $x(n)$ (19.11) при $N = N_i$ и $f_d = f_d^i$ для коэффициента интерполяции $L = 8$.

Систему многократной интерполяции представить в виде каскадного соединения трех систем однократной интерполяции с коэффициентами интерполяции $L/4$.

Для функции `interp` значения входных параметров `I` и `fx` установить по умолчанию.

Вывести графики:

- входного сигнала;
- выходного сигнала системы однократной интерполяции;
- выходного сигнала системы многократной интерполяции.

4С. Моделирование системы однократной передискретизации с полифазной структурой.

Выполнить для входного сигнала $x(n)$ (19.11) при $N = 128$ и $f_d = 800$ для коэффициентов передискретизации $L/M = 3/4$ и $L/M = 2/5$.

Вывести графики:

- входного сигнала;
- выходного сигнала;
- выходного сигнала в установившемся режиме;
- амплитудного спектра входного сигнала на периоде;
- амплитудного спектра выходного сигнала в установившемся режиме на периоде.

Определить, в каком из этих случаев и по какой причине наблюдается эффект растекания спектра.

19.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения пунктов задания, включая созданные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**), описания полифазных структур в виде объектов `mfilt`, копируемые из окна **Command Window** (шрифт Courier New), и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

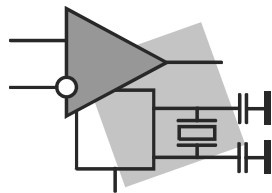
1. Какие системы называют многоскоростными?
2. Какое преобразование выполняется системой однократной интерполяции?
3. Какое преобразование выполняется системой однократной децимации?
4. Какое преобразование выполняется системой однократной передискретизации?
5. Какие многоскоростные системы называют однократными?
6. Какие многоскоростные системы называют многократными?
7. Что собой представляет многократная система преобразования частоты?
8. Какие блоки включает в себя система однократной интерполяции?
9. Какая обработка выполняется каждым из блоков?
10. Поясните интерпретацию процедуры интерполяции во временной области.
11. Поясните интерпретацию процедуры интерполяции в частотной области.
12. Какие блоки включает в себя система однократной децимации?
13. Какая обработка выполняется каждым из блоков?
14. Поясните интерпретацию процедуры децимации в частотной области.
15. Поясните интерпретацию процедуры децимации во временной области.

16. Какие блоки включает в себя система однократной передискретизации?
17. Какие фильтры ФНЧ рекомендуется использовать в системах интерполяции, децимации и передискретизации для исключения влияния фазовых искажений?
18. В чем преимущество полифазных структур интерполяции и децимации?
19. Как это преимущество реализовано в полифазной структуре системы однократной интерполяции?
20. Как это преимущество реализовано в полифазной структуре системы однократной децимации?
21. Какой вид имеет идеальная АЧХ фильтра Найквиста для системы однократной интерполяции?
22. Какой вид имеет идеальная АЧХ фильтра Найквиста для системы однократной децимации?

19.7. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 17.
2. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов. — 2-е изд. — СПб.: БХВ-Петербург, 2005. — Глава 29.
3. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 10.
4. Оппенгейм А., Шафер Р. Цифровая обработка сигналов — М.: Техносфера, 2007. — Глава 4.

ГЛАВА 20



Моделирование полифазных структур многоскоростных систем средствами GUI FDATool и FilterBuilder

Цель работы: овладеть средствами GUI FDATool (Filter Design and Analysis Toolbox — средство проектирования и анализа фильтров) и FilterBuilder (Разработчик фильтров) для моделирования полифазных структур многоскоростных систем ЦОС.

20.1. Краткая теоретическая справка

Ранее (см. разд. 19.2.4) были рассмотрены программные средства MATLAB для моделирования многоскоростных систем в виде объектов `mfilt` с полифазными структурами, использующими *фильтры Найквиста* (по умолчанию), идеальная АЧХ которых имеет ограничения на полосу пропускания (ПП).

При использовании средств GUI FDATool и FilterBuilder для моделирования полифазных структур с КИХ-фильтрами эти ограничения отсутствуют, и требования к АЧХ задаются пользователем.

20.1.1. Моделирование полифазных структур в GUI FDATool

Обращение к GUI FDATool происходит по команде:

```
fdatool
```

после чего открывается окно **Filter Design & Analysis Tool** (Средство проектирования и анализа фильтра), представленное на рис. 20.1.

Моделирование полифазной структуры многоскоростной системы в GUI FDATool осуществляется в три этапа:

1. Синтез КИХ-фильтра ФНЧ, работающего на "высокой" частоте дискретизации.
2. Моделирование полифазной структуры многоскоростной системы.

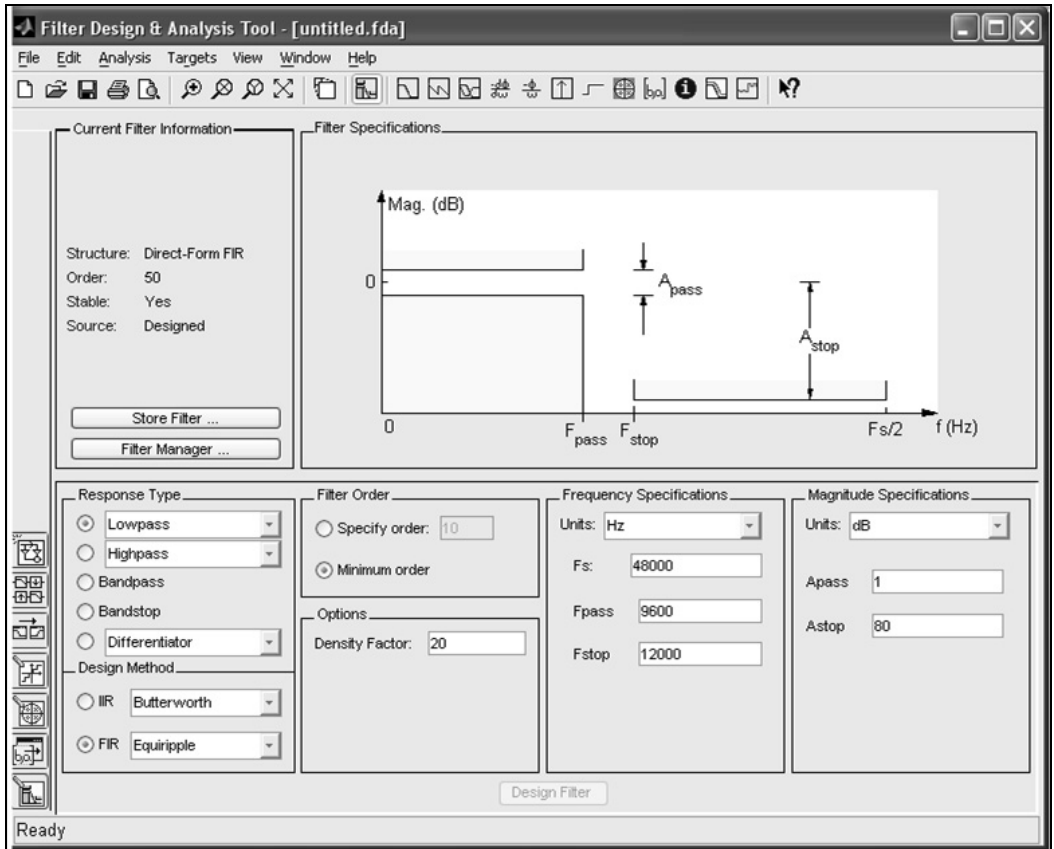


Рис. 20.1. Окно **Filter Design & Analysis Tool** при нажатой кнопке **Design Filter**

3. Экспорт полифазной структуры многоскоростной системы в рабочее пространство памяти Workspace в виде объекта `mfilt`.

Первый этап — синтез КИХ-фильтра ФНЧ, работающего на "высокой" частоте дискретизации — выполняется при нажатой кнопке **Design Filter** (Синтезировать фильтр), расположенной на панели инструментов в левом нижнем углу (см. рис. 20.1).

Синтез КИХ-фильтров рассматривался в *разд. 14.1.1*. Сформулируем его особенно при моделировании КИХ-фильтров для полифазных структур.

Требования к частотам КИХ-фильтра ФНЧ (**Lowpass**) задаются в группе **Frequency Specifications** (Требования к частотам) при выборе в списке **Units** значения Hz и зависят от верхней частоты f_v и частоты дискретизации f_d входного сигнала и типа многоскоростной системы, а именно:

- для системы однократной интерполяции с частотой дискретизации выходного сигнала Lf_d , где L — коэффициент интерполяции, задаются (см. (19.5)):

- частота дискретизации (**Fs**) выходного сигнала Lf_d ;

- граничная частота ПП f_χ (**Fpass**), выбираемая из интервала $f_b \leq f_\chi < \frac{f_d}{2}$;
 - граничная частота ПЗ f_k (**Fstop**), выбираемая из интервала $f_\chi < f_k < \frac{f_d}{2}$;
- для системы однократной децимации с частотой дискретизации выходного сигнала $\frac{f_d}{M}$, где M — коэффициент децимации, задаются (см. (19.7)):
- частота дискретизации (**Fs**) входного сигнала f_d ;
 - граничная частота ПП f_χ (**Fpass**), выбираемая из интервала $f_b \leq f_\chi < \frac{f_d}{2M}$;
 - граничная частота ПЗ f_k (**Fstop**), выбираемая из интервала $f_\chi < f_k < \frac{f_d}{2M}$;
- для системы однократной передискретизации с частотой дискретизации выходного сигнала $\frac{L}{M} f_d$, где $\frac{L}{M}$ — коэффициент передискретизации в виде рациональной дроби, задаются (см. (19.8)):
- частота дискретизации (**Fs**), равная $\max \left\{ f_d; \frac{L f_d}{M} \right\}$;
 - граничная частота ПП f_χ (**Fpass**), выбираемая из интервала $f_b \leq f_\chi < \min \left\{ \frac{f_d}{2}; \frac{L f_d}{2M} \right\}$;
 - граничная частота ПЗ f_k (**Fstop**), выбираемая из интервала $f_\chi < f_k < \min \left\{ \frac{f_d}{2}; \frac{L f_d}{2M} \right\}$.

При выборе значений граничных частот ПП f_χ и ПЗ f_k необходимо иметь в виду, что они определяют ширину переходной полосы ФНЧ, а следовательно, отклонение АЧХ от идеальной и порядок КИХ-фильтра.

Требования к максимально допустимым отклонениям от идеальной АЧХ задаются в группе **Magnitude Specifications** (Требования к АЧХ) при выборе в списке **Units** значения Linear (для отклонений в децибелах — dB) и не зависят от типа многоскоростной системы.

При выборе максимально допустимого отклонения от единицы δ_1 в ПП (**Apass**) и от нуля δ_2 в ПЗ (**Astop**) необходимо помнить, что, с одной стороны, они определяют отклонение АЧХ от идеальной, а следовательно, искажение выходного сигнала, но с другой — порядок КИХ-фильтра, т. е. его сложность.

Процедура синтеза КИХ-фильтра ФНЧ (**Lowpass**) включает в себя следующие шаги:

- ☐ выбор метода синтеза в списке переключателя **FIR** (КИХ) — метод чебышевской аппроксимации (**Equiripple**);
- ☐ оценка начального порядка КИХ-фильтра — установка переключателя **Minimum order** (Минимальный порядок) в группе **Filter Order** (Порядок фильтра);
- ☐ синтез КИХ-фильтра минимального порядка после нажатия нижней кнопки **Design Filter** (Синтезировать фильтр);
- ☐ коррекция порядка КИХ-фильтра — установка переключателя **Specify order** (Произвольный порядок) в группе **Filter Order**.

Коррекция порядка КИХ-фильтра выполняется для того, чтобы его *длина*¹ была кратной коэффициенту интерполяции (децимации) и, соответственно, длина полифазных фильтров — *целым числом*.

В поле переключателя **Specify order** указывается порядок, ближайший к минимальному, такой чтобы длина КИХ-фильтра была кратной коэффициенту интерполяции (децимации). В группе **Magnitude Specifications** указываются веса (см. разд. 12.2.1) в ПП (**Wpass**) и ПЗ (**Wstop**); при выборе $\delta_1 = \delta_2$ они будут равны единице.

Порядок, ближайший к минимальному, выбирается из соображений компромисса между сложностью КИХ-фильтра и близостью АЧХ к идеальной;

- ☐ синтез КИХ-фильтра указанного порядка;
- ☐ сохранение КИХ-фильтра в буфере **Filter Manager** (Диспетчер фильтров) после нажатия кнопки **Store Filter** (Сохранить фильтр) в группе **Current Filter Information** (Информация о текущем фильтре).

Второй этап — моделирование полифазной структуры многоскоростной системы — выполняется при нажатой кнопке **Create a multirate filter** (Создать многоскоростной фильтр), расположенной на панели инструментов в левом нижнем углу, и включает в себя следующие шаги (рис. 20.2):

- ☐ загрузка из буфера **Filter Manager** КИХ-фильтра, работающего на "высокой" частоте дискретизации; на его основе будет моделироваться полифазная структура;
- ☐ выбор типа многоскоростной системы в списке **Type** (Тип):
 - **Interpolator** (Интерполятор) — система однократной интерполяции;
 - **Decimator** (Дециматор) — система однократной децимации;
 - **Fractional-rate converter** (Передискретизатор) — система однократной передискретизации;

¹ Напомним, что длина КИХ-фильтра на единицу больше его порядка.

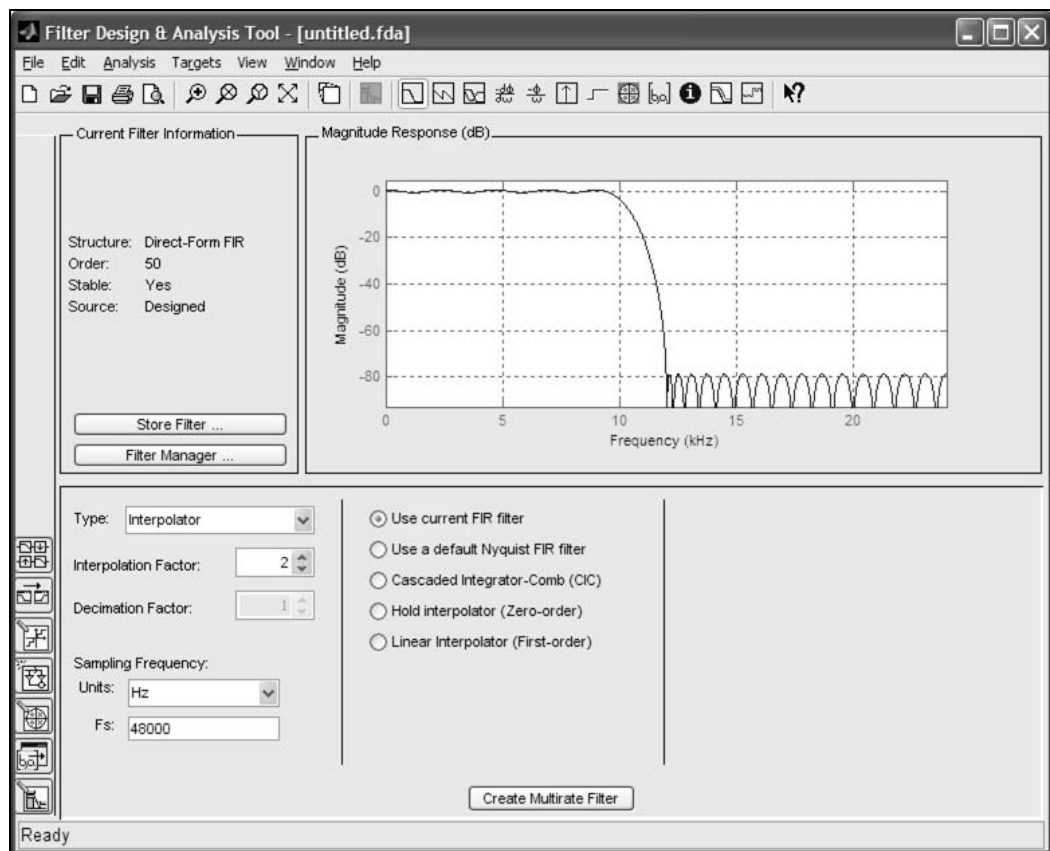


Рис. 20.2. Окно Filter Design & Analysis Tool при нажатой кнопке Create Multirate Filter

- ☐ задание коэффициента преобразования частоты:
 - в поле **Interpolation Factor** — коэффициента интерполяции L ;
 - в поле **Decimation Factor** — коэффициента децимации M ;
- ☐ выбор единиц измерения частоты в списке **Units** — Hz;
- ☐ задание частоты дискретизации *выходного сигнала* **F_s**.

Частота **F_s** может задаваться непосредственно или арифметическим выражением на языке MATLAB. Например, для системы однократной передискретизации частоту **F_s** можно указать как $500 \cdot (3/5)$, где 500 — частота дискретизации входного сигнала, а 3 и 5 — значения коэффициентов интерполяции L и децимации M ;

- ☐ выбор фильтра, на основе которого будет моделироваться полифазная структура, — установка переключателя **Use current FIR filter** (Использовать текущий КИХ-фильтр);
- ☐ моделирование полифазной структуры после нажатия нижней кнопки **Create Multirate Filter** (Синтезировать многоскоростной фильтр);

- сохранение полифазной структуры в буфере **Filter Manager** после нажатия кнопки **Store Filter** (Сохранение фильтра) в группе **Current Filter Information** (Информация о текущем фильтре).

Результаты моделирования отображаются в группе **Current Filter Information** и включают в себя:

- **Structure** — полифазная структура (см. разд. 19.1.4);
- **Order** — порядок КИХ-фильтра, работающего на "высокой" частоте дискретизации;
- **Rate Change** — отношение *периодов дискретизации* входного и выходного сигналов;
- **Stable** — устойчивость КИХ-фильтра, работающего на "высокой" частоте дискретизации: Yes (устойчив) по определению;
- **Source** — источник получения полифазной структуры: моделируемая (Multirate Design) или импортированная (Imported).

Третий этап — экспорт полифазной структуры многоскоростной системы в рабочее пространство памяти Workspace в виде объекта `mfilt` — выполняется точно так же, как экспорт цифрового фильтра в виде объекта `dfilt` (см. разд. 14.1.2).

Предварительно в буфере **Filter Manager** выделяется имя экспортируемой полифазной структуры.

20.1.2. Моделирование полифазных структур в FilterBuilder GUI

Обращение к FilterBuilder происходит по команде:

```
filterbuilder
```

В окне **Response Selection** (Выбор характеристики) выбирается тип избирательности Lowpass (ФНЧ), после чего открывается окно, представленное на рис. 14.2.

Моделирование полифазной структуры многоскоростной системы в FilterBuilder GUI осуществляется после указания параметров на вкладке **Main**, которые включают в себя:

- требования к АЧХ КИХ-фильтра ФНЧ, работающего на "высокой" частоте дискретизации, и метод его синтеза (см. разд. 14.1.3 и 20.2.1);
- тип многоскоростной системы, реализуемой на базе КИХ-фильтра, в списке **Filter type** (Тип фильтра):
 - Decimator (Дециматор) — система однократной децимации с указанием коэффициента децимации в поле ввода **Decimation Factor** (Коэффициент децимации);
 - Interpolator (Интерполятор) — система однократной интерполяции с указанием коэффициента интерполяции в поле ввода **Interpolation Factor** (Коэффициент интерполяции);

- Sample-rate convertor (Передискретизатор) — система однократной передискретизации с указанием коэффициентов интерполяции в поле ввода **Interpolation Factor** и децимации в поле ввода **Decimation Factor**;

□ тип полифазной структуры в списке **Structure**.

После задания параметров на вкладке **Main** следует в поле **Save variable as** (Сохранить переменную как) указать или выбрать по умолчанию имя объекта `mfilt` для полифазной структуры и нажать кнопку **Apply** (Применить). Будет создана полифазная структура и одновременно осуществлен ее *экспорт* в *Workspace*. Свойства объекта `mfilt` выводятся по его имени в окне **Command Window**.

20.1.3. Моделирование многоскоростных систем с полифазными структурами

Моделирование многоскоростной системы с полифазной структурой выполняется на основе объекта `mfilt`, экспортированного в *Workspace* из GUI **FDATool** или **FilterBuilder GUI** с помощью функции:

y = filter(H,x)

Сравнивать выходной сигнал (реакцию) y с входным сигналом (воздействием) x имеет смысл в *установившемся режиме* (см. разд. 19.1.4).

Необходимо иметь в виду, что при синтезе КИХ-фильтра требования задаются к *нормированной АЧХ*, поэтому *выходной сигнал системы интерполяции и передискретизации следует умножить на коэффициент L* (см. (19.5) и (19.8)).

20.2. Содержание лабораторной работы

Содержание работы связано с моделированием полифазных структур систем однократной интерполяции, децимации и передискретизации средствами GUI **FDATool** и моделированием многоскоростных систем с полифазными структурами программными средствами **MATLAB**.

Моделирование полифазных структур в **FilterBuilder GUI** вынесено в самостоятельное задание.

20.3. Задание на лабораторную работу

Лабораторная работа выполняется с использованием GUI **FDATool** и *script*-файлов `lr_20_interp`, `lr_20_decim`, `lr_20_res_up` и `lr_20_res_down`, которые хранятся на прилагаемом компакт-диске в папке `LAB_DSP\LAB_20`.

Перед выполнением работы необходимо сохранить путь к папке `LAB_20` по команде контекстного меню **Add to Path | Selected Folders**.

Исходные данные для пунктов задания приводятся в табл. 20.1 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$. Функция $N_{бр} \bmod M$ в записи исходных данных означает вычисление значения $N_{бр}$ по модулю M .

На прилагаемом компакт-диске в папке Tables\Tables_20 хранятся табл. 20.1 исходных данных и пример ее заполнения для $N_{\text{бр}} = 1$.

При задании требований к АЧХ КИХ-фильтров, работающих на "высокой" частоте дискретизации, рекомендуется указать следующие параметры:

- ☐ верхняя частота входного сигнала $f_{\text{в}} = f_2$ (см. табл. 20.1);
- ☐ граничная частота ПП $f_{\chi} = f_{\text{в}}$;
- ☐ максимально допустимые отклонения в ПП δ_1 и ПЗ δ_2 — одинаковые ($\delta_1 = \delta_2$), выбираемые из диапазона $0,01 \leq \delta_1 \leq 0,02$.

Остальные параметры указываются в пунктах задания или выбираются в соответствии с рекомендациями в разд. 20.1.1.

Таблица 20.1. Таблица исходных данных

Переменная	Назначение	Значение	Идентификатор
$N_{\text{бр}}$	Номер бригады	$N_{\text{бр}}$	Nb =
A_1	Амплитуды дискретных гармоник входного сигнала	$A_1 = N_{\text{бр}} \bmod 3 + 1$	A1 =
A_2		$A_2 = 2A_1$	A2 =
f_1	Частоты дискретных гармоник входного сигнала	$f_1 = 50(N_{\text{бр}} \bmod 5 + 1)$	f1 =
f_2		$f_2 = 1,5 f_1$	f2 =
Система однократной интерполяции с полифазной структурой			
N_i	Период входного сигнала	$N_i = 64$	Ni = 64
$f_{\text{д}}^i$	Частота дискретизации входного сигнала	$f_{\text{д}}^i = 200(N_{\text{бр}} \bmod 5 + 1)$	Fs_i =
L	Коэффициент интерполяции	$L = 5$	L = 5
Система однократной децимации с полифазной структурой			
N_d	Период входного сигнала	$N_d = 256$	Nd = 256
$f_{\text{д}}^d$	Частота дискретизации входного сигнала	$f_{\text{д}}^d = 800(N_{\text{бр}} \bmod 5 + 1)$	Fs_d =
M	Коэффициент децимации	$M = 4$	M = 4

Таблица 20.1 (окончание)

Переменная	Назначение	Значение	Идентификатор
Система однократной передискретизации с полифазной структурой при повышении частоты дискретизации			
N_{ri}	Период входного сигнала	$N_{ri} = 64$	Nri = 64
f_d^{ri}	Частота дискретизации входного сигнала	$f_d^{ri} = 400(N_{bp} \bmod 5 + 1)$	Fs_ri =
L/M	Коэффициент передискретизации	$L/M = 5/2$	L = 5 M = 2
Система однократной передискретизации с полифазной структурой при понижении частоты дискретизации			
N_{rd}	Период входного сигнала	$N_{rd} = 128$	Nrd = 128
f_d^{rd}	Частота дискретизации входного сигнала	$f_d^{rd} = 800(N_{bp} \bmod 5 + 1)$	Fs_rd =
L/M	Коэффициент передискретизации	$L/M = 3/4$	L = 3 M = 4

Задание на лабораторную работу заключается в моделировании полифазных структур и многоскоростных систем с полифазными структурами и включает в себя следующие пункты:

1. Моделирование полифазной структуры системы однократной интерполяции.
Выполнить моделирование полифазной структуры системы однократной интерполяции в GUI FDATool, задавая частоту дискретизации входного сигнала $f_d = f_d^i$ и коэффициент интерполяции $L = 5$.
Экспортировать полифазную структуру в Workspace в виде объекта mfilt с именем hi и вывести его свойства в окне **Command Window**.
Пояснить:
- чему равна длина КИХ-фильтра, работающего на "высокой" частоте дискретизации входного сигнала;
 - какое количество КИХ-фильтров используется в полифазной структуре, чему равна их длина и на какой частоте они работают.
2. Моделирование полифазной структуры системы однократной децимации.
Выполнить моделирование полифазной структуры системы однократной децимации в GUI FDATool, задавая частоту дискретизации входного сигнала $f_d = f_d^d$ и коэффициент децимации $M = 4$.

Экспортировать полифазную структуру в Workspace в виде объекта `mfilt` с именем `hd` и вывести его свойства в окне **Command Window**.

Пояснить:

- чему равна длина КИХ-фильтра, работающего на "высокой" частоте дискретизации входного сигнала;
- какое количество КИХ-фильтров используется в полифазной структуре, чему равна их длина и на какой частоте они работают.

3. Моделирование полифазной структуры системы однократной передискретизации при повышении частоты дискретизации.

Выполнить моделирование полифазной структуры системы однократной передискретизации с повышением частоты дискретизации в GUI `FDATool`, задавая частоту дискретизации входного сигнала $f_d = f_d^{ri}$ и коэффициент передискретизации $L/M = 5/2$.

Экспортировать полифазную структуру в Workspace в виде объекта `mfilt` с именем `hri` и вывести его свойства в окне **Command Window**.

Пояснить:

- чему равна длина КИХ-фильтра, работающего на "высокой" частоте дискретизации входного сигнала;
- какое количество КИХ-фильтров используется в полифазной структуре, чему равна их длина и на какой частоте они работают.

4. Моделирование полифазной структуры системы однократной передискретизации при понижении частоты дискретизации.

Выполнить моделирование полифазной структуры системы однократной передискретизации с понижением частоты дискретизации в GUI `FDATool`, задавая частоту дискретизации входного сигнала $f_d = f_d^{rd}$ и коэффициент передискретизации $L/M = 3/4$.

Экспортировать полифазную структуру в Workspace в виде объекта `mfilt` с именем `hrd` и вывести его свойства в окне **Command Window**.

Пояснить:

- чему равна длина КИХ-фильтра, работающего на "высокой" частоте дискретизации входного сигнала;
- какое количество КИХ-фильтров используется в полифазной структуре, чему равна их длина и на какой частоте они работают.

5. Сохранение сессии `FDATool`.

Сохранить сессию с именем `Multirate_Filters.fda` (см. разд. 14.1.1).

6. Моделирование системы однократной интерполяции с полифазной структурой.

В качестве системы однократной интерполяции с полифазной структурой использовать объект `mfilt` с именем `hi` (см. п. 1).

В качестве входного сигнала выбрать периодическую последовательность $x(n)$ (идентификатор x) с периодом N (идентификатор N) и частотой дискретизации f_d (идентификатор Fs):

$$x(n) = A_1 \sin\left(\frac{2\pi f_1}{f_d} n\right) + A_2 \sin\left(\frac{2\pi f_2}{f_d} n\right). \quad (20.1)$$

Ввести значения $N = N_i$, $f_d = f_d^i$ и коэффициент интерполяции $L = 5$.

Вывести графики:

- входного сигнала $x(n)$;
- выходного сигнала $y(n)$ (идентификатор y);
- выходного сигнала $y(n)$ в установившемся режиме (идентификатор y_i) с автоматическим определением момента его начала $n_{\text{нач}}$ (идентификатор nd_start), о чем шла речь в разд. 19.1.4.

Вывести графики в шкале частот f (Гц) в одинаковом диапазоне $[0 \ L \cdot Fs]$ с помощью функции `xlim`:

- амплитудного спектра (идентификатор `MODX`) входного сигнала $x(n)$ на периоде;
- амплитудного спектра (идентификатор `MODY`) выходного сигнала $y(n)$ в установившемся режиме на периоде и одновременно АЧХ КИХ-фильтра (идентификатор `MAG`), работающего на "высокой" частоте дискретизации с помощью функции `plot` красным цветом.

Пояснить:

- связь между периодами и частотами дискретизации, длинами и длительностями входного и выходного сигналов;
- вид амплитудных спектров входного и выходного сигналов;
- почему входной и выходной сигналы и их амплитудные спектры имеет смысл сравнивать в установившемся режиме;
- каким требованиям должна удовлетворять идеальная АЧХ КИХ-фильтра, работающего на "высокой" частоте дискретизации.

7. Моделирование системы однократной децимации с полифазной структурой.

В качестве системы однократной децимации с полифазной структурой использовать объект `mfilt` с именем `Hd` (см. п. 2).

В качестве входного сигнала системы выбрать $x(n)$ (20.1). Ввести значения $N = N_d$, $f_d = f_d^d$ и коэффициент децимации $M = 4$.

Вывести графики:

- входного сигнала $x(n)$ (идентификатор `x`);
- выходного сигнала $y(n)$ (идентификатор `y`);
- выходного сигнала $y(n)$ в установившемся режиме (идентификатор `yd`) с автоматическим определением момента его начала $n_{\text{нач}}$ (идентификатор `nd_start`).

Вывести графики в шкале частот f (Гц) в одинаковом диапазоне $[0 \text{ } Fs]$:

- амплитудного спектра (идентификатор `MODX`) входного сигнала $x(n)$ на периоде и одновременно АЧХ КИХ-фильтра (идентификатор `MAG`), работающего на "высокой" частоте дискретизации с помощью функции `plot` красным цветом;
- амплитудного спектра (идентификатор `MODY`) выходного сигнала $y(n)$ в установившемся режиме на периоде.

Пояснить:

- связь между периодами и частотами дискретизации, длинами и длительностями входного и выходного сигналов;
- вид амплитудных спектров входного сигнала и выходного сигналов;
- каким требованиям должна удовлетворять идеальная АЧХ КИХ-фильтра, работающего на "высокой" частоте дискретизации.

8. Моделирование системы однократной передискретизации с полифазной структурой при повышении частоты дискретизации.

В качестве системы однократной передискретизации с полифазной структурой использовать объект `mfilt` с именем `Hri` (см. п. 3).

В качестве входного сигнала системы выбрать $x(n)$ (20.1). Ввести значения

$$N = N_{ri}, \quad f_d = f_d^{ri} \quad \text{и} \quad L = 5, \quad M = 2 \quad \text{для коэффициента передискретизации} \quad L/M.$$

Вывести графики:

- входного сигнала $x(n)$ (идентификатор `x`);
- выходного сигнала $y(n)$ (идентификатор `y`);
- выходного сигнала $y(n)$ в установившемся режиме (идентификатор `yri`) с автоматическим определением момента его начала $n_{\text{нач}}$ (идентификатор `nri_start`).

Вывести графики в шкале частот f (Гц) в одинаковом диапазоне $[0 \text{ } (L/M) * Fs]$:

- амплитудного спектра (идентификатор `MODX`) входного сигнала $x(n)$ на периоде;

- амплитудного спектра (идентификатор `MODY`) выходного сигнала $y(n)$ в установившемся режиме на периоде и одновременно АЧХ КИХ-фильтра (идентификатор `MAG`), работающего на "высокой" частоте дискретизации с помощью функции `plot` красным цветом.

Пояснить:

- связь между периодами и частотами дискретизации, длинами и длительностями входного и выходного сигналов;
- вид амплитудных спектров входного сигнала и выходного сигналов;
- каким требованиям должна удовлетворять идеальная АЧХ КИХ-фильтра, работающего на "высокой" частоте дискретизации.

9. Моделирование системы однократной передискретизации с полифазной структурой при понижении частоты дискретизации.

В качестве системы однократной передискретизации с полифазной структурой использовать объект `mfilt` с именем `Hrd` (см. п. 4).

В качестве входного сигнала системы выбрать $x(n)$ (20.1). Ввести значения $N = N_{rd}$, $f_d = f_d^{rd}$ и $L = 3$, $M = 4$ для коэффициента передискретизации L/M .

Вывести графики:

- входного сигнала $x(n)$ (идентификатор `x`);
- выходного сигнала $y(n)$ (идентификатор `y`);
- выходного сигнала $y(n)$ в установившемся режиме (идентификатор `yrd`) с автоматическим определением момента его начала $n_{\text{нач}}$ (идентификатор `nrd_start`).

Вывести графики в шкале частот f (Гц) в одинаковом диапазоне $[0 \text{ } F_s]$:

- амплитудного спектра (идентификатор `MODX`) входного сигнала $x(n)$ на периоде и одновременно АЧХ КИХ-фильтра (идентификатор `MAG`), работающего на "высокой" частоте дискретизации с помощью функции `plot` красным цветом;
- амплитудного спектра (идентификатор `MODY`) выходного сигнала $y(n)$ в установившемся режиме на периоде.

Пояснить:

- связь между периодами и частотами дискретизации, длинами и длительностями входного и выходного сигналов;
- вид амплитудных спектров входного сигнала и выходного сигналов;
- каким требованиям должна удовлетворять идеальная АЧХ КИХ-фильтра, работающего на "высокой" частоте дискретизации.

10. Проверка созданных полифазных структур.

Оценить корректность созданных в GUI FDATool полифазных структур, анализируя графики результатов моделирования многоскоростных систем (см. пп. 6—9).

Недостовверные результаты свидетельствуют о некорректности созданной полифазной структуры и необходимости ее моделирования заново. Для этого необходимо выполнить следующие действия:

- запустить GUI FDATool и открыть сохраненную сессию с именем `Multirate_Filters.fda` (см. разд. 14.1.1);
- в буфере **Filter Manager** удалить имя некорректной полифазной структуры и соответствующего КИХ-фильтра;
- вновь выполнить моделирование полифазной структуры и сохранить ее в буфере **Filter Manager**;
- удалить из Workspace имя некорректной полифазной структуры по команде:
`clear <имя полифазной структуры>`
- экспортировать в Workspace новую полифазную структуру с тем же именем;
- запустить script-файл для моделирования многоскоростной системы с новой полифазной структурой и проверить результат.

20.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должна быть представлена табл. 20.1 исходных данных для своего номера бригады $N_{бр}$.

Для четырех типов многоскоростных систем созданы четыре script-файла. Для запуска script-файла к нему необходимо обратиться по имени:

- >> `lr_20_interp` — система однократной интерполяции с полифазной структурой;
- >> `lr_20_decim` — система однократной децимации с полифазной структурой;
- >> `lr_20_res_up` — система однократной передискретизации с полифазной структурой при повышении частоты дискретизации;
- >> `lr_20_res_down` — система однократной передискретизации с полифазной структурой при понижении частоты дискретизации.

Листинги данных script-файлов представлены в разд. 20.41—20.4.4.

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш <Ctrl>+<Break>.

При выполнении script-файла текущие окна с графиками *не закрывать*.

20.4.1. Система однократной интерполяции с полифазной структурой

Листинг script-файла lr_20_interp имеет вид:

```
>> type lr_20_interp
script
clc
disp('% ЛР №20. МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ИНТЕРПОЛЯЦИИ С ПОЛИФАЗНОЙ СТРУКТУРОЙ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Nb = input('Nb = '); % НОМЕР БРИГАДЫ
A1 = input('A1 = '); % АМПЛИТУДЫ ДИСКРЕТНЫХ ГАРМОНИК ВХОДНОГО СИГНАЛА
A2 = input('A2 = ');
f1 = input('f1 = '); % ЧАСТОТЫ (Гц) ДИСКРЕТНЫХ ГАРМОНИК ВХОДНОГО СИГНАЛА
f2 = input('f2 = ');
Ni = input('Ni = '); % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs_i = input('Fs_i = '); % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
L = input('L = '); % КОЭФФИЦИЕНТ ИНТЕРПОЛЯЦИИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('% Для вывода значений Ni, Fs_i и L нажмите <ENTER>')
pause
disp('%')
disp(['N = Ni = ',num2str(Ni), ' Fs = Fs_i = ',num2str(Fs_i)])
disp('%')
disp([' L = ',num2str(L)])
N = Ni; % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs = Fs_i; % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
n = 0:(N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ВХОДНОГО СИГНАЛА
x = A1*sin((2*pi*f1/Fs).*n)+A2*sin((2*pi*f2/Fs).*n); % ВХОДНОЙ СИГНАЛ
y = L*filter(Ni,x); % ВЫХОДНОЙ СИГНАЛ, УМНОЖЕННЫЙ НА КОЭФФИЦИЕНТ ИНТЕРПОЛЯЦИИ
ni_start = ceil(length(y)/2+1); % НАЧАЛО УСТАНОВИВШЕГОСЯ РЕЖИМА
yi = y(ni_start:end); % ВЫХОДНОЙ СИГНАЛ В УСТАНОВИВШЕМСЯ РЕЖИМЕ
ni = ni_start:(ni_start+length(yi)-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ УСТАНОВИВШЕГОСЯ РЕЖИМА
disp('%')
disp('% Для вывода ГРАФИКОВ ВХОДНОГО И ВЫХОДНОГО сигналов нажмите <ENTER>')
pause
```

```

figure('Name','Input and Output Signals in Polyphase Structure Interpolation
System','NumberTitle','off')
subplot(3,1,1), stem(n,x), grid, xlabel('n')
title(strcat(['Input Signal x(n)      N = ',num2str(N)]))
subplot(3,1,2), stem(0:length(y)-1,y), grid
title(strcat(['L = ',num2str(L),'      Output Signal y(n)      length(y) =
',num2str(length(y))]))
subplot(3,1,3), stem(ni,yi), grid, xlabel('n')
title(strcat(['Output Signal y(n)      Starting point n = ',num2str(ni_start),
length(y) = ',num2str(length(yi))]))
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ и АЧХ КИХ-фильтра нажмите
<ENTER>')
pause
X = fft(x); % ДПФ ВХОДНОГО СИГНАЛА
MODX = (2/N)*abs(X); % АМПЛИТУДНЫЙ СПЕКТР ВХОДНОГО СИГНАЛА
MODX(1) = (1/N)*abs(X(1));
Yi = fft(yi); % ДПФ ВЫХОДНОГО СИГНАЛА
MODYi = (2/length(Yi))*abs(Yi); % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА
MODYi(1) = (1/length(Yi))*abs(Yi(1));
k = 0:(N-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВХОДНОГО СИГНАЛА
ki = 0:(length(Yi)-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВЫХОДНОГО СИГНАЛА
f = 0:(Fs*L)/1000:FsWithL; % СЕТКА ЧАСТОТ ДЛЯ АЧХ КИХ-ФИЛЬТРА
h = Hi.Numerator; % ИХ КИХ-ФИЛЬТРА
MAG = abs(freqz(h,1,f,Fs*L)); % АЧХ КИХ-ФИЛЬТРА
figure('Name','Amplitude Spectrums and Magnitude Response in Polyphase
Structure Interpolation System','NumberTitle','off')
subplot(2,1,1), stem(k*(Fs/N),MODX,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 L*Fs])
title(strcat(['Amplitude Spectrum x(n)      N = ',num2str(N)]))
subplot(2,1,2), stem(ki*(Fs*L/length(Yi)),MODYi,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 L*Fs])
title(strcat(['L = ',num2str(L),'      Amplitude Spectrum y(n) and FIR Magnitude
Response']))
hold on, plot(f,MAG,'r','Linewidth',2)
disp('%')
disp('%')
disp('% МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ИНТЕРПОЛЯЦИИ С ПОЛИФАЗНОЙ СТРУКТУРОЙ
ЗАВЕРШЕНО')

```

20.4.2. Система однократной децимации с полифазной структурой

Листинг script-файла lr_20_decim имеет вид:

```

>> type lr_20_decim
script
clc

```

```

disp('% ЛР №20. МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ДЕЦИМАЦИИ С ПОЛИФАЗНОЙ
СТРУКТУРОЙ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0
Nb = input('Nb = '); % НОМЕР БРИГАДЫ
A1 = input('A1 = '); % АМПЛИТУДЫ ДИСКРЕТНЫХ ГАРМОНИК ВХОДНОГО СИГНАЛА
A2 = input('A2 = ');
f1 = input('f1 = '); % ЧАСТОТЫ (Гц) ДИСКРЕТНЫХ ГАРМОНИК ВХОДНОГО СИГНАЛА
f2 = input('f2 = ');
Nd = input('Nd = '); % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs_d = input('Fs_d = '); % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
M = input('M = '); % КОЭФФИЦИЕНТ ДЕЦИМАЦИИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('% Для вывода значений Nd, Fs_d и M нажмите <ENTER>')
pause
disp('%')
disp(['N = Nd = ',num2str(Nd),' Fs = Fs_d = ',num2str(Fs_d)])
disp('%')
disp(['M = ',num2str(M)])
N = Nd; % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs = Fs_d; % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
n = 0:(N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ВХОДНОГО СИГНАЛА
x = A1*sin((2*pi*f1/Fs).*n)+A2*sin((2*pi*f2/Fs).*n); % ВХОДНОЙ СИГНАЛ
y = filter(Hd,x); % ВЫХОДНОЙ СИГНАЛ
nd_start = ceil(length(y)/2+1); % НАЧАЛО УСТАНОВИВШЕГОСЯ РЕЖИМА
yd = y(nd_start:end); % ВЫХОДНОЙ СИГНАЛ В УСТАНОВИВШЕМСЯ РЕЖИМЕ
nd = nd_start:(nd_start+length(yd)-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ
УСТАНОВИВШЕГОСЯ РЕЖИМА
disp('%')
disp('% Для вывода ГРАФИКОВ ВХОДНОГО И ВЫХОДНОГО сигналов нажмите <ENTER>')
pause
figure('Name','Input and Output Signals in Polyphase Structure Decimation
System','NumberTitle','off')
subplot(3,1,1), stem(n,x), grid, xlabel('n')
title(strcat(['Input Signal x(n) N = ',num2str(N)]))
subplot(3,1,2), stem(0:length(y)-1,y), grid
title(strcat(['M = ',num2str(M),' Output Signal y(n) length(y) = ',num2str(length(y))]))
subplot(3,1,3), stem(nd,yd), grid, xlabel('n')

```



```

title(strcat(['Output Signal y(n)   Starting point n = ',num2str(nd_start),'
length(y) = ',num2str(length(yd))]))
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ и АЧХ КИХ-фильтра нажмите
<ENTER>')
pause
X = fft(x); % ДПФ ВХОДНОГО СИГНАЛА
MODX = (2/N)*abs(X); % АМПЛИТУДНЫЙ СПЕКТР ВХОДНОГО СИГНАЛА
MODX(1) = (1/N)*abs(X(1));
Yd = fft(yd); % ДПФ ВЫХОДНОГО СИГНАЛА В УСТАНОВИВШЕМСЯ РЕЖИМЕ
MODYd = (2/length(Yd))*abs(Yd); % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА
MODYd(1) = (1/length(Yd))*abs(Yd(1));
k = 0:(N-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВХОДНОГО СИГНАЛА
kd = 0:(length(Yd)-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВЫХОДНОГО СИГНАЛА
f = 0:Fs/1000:Fs; % СЕТКА ЧАСТОТ ДЛЯ АЧХ КИХ-ФИЛЬТРА
hd = Hd.Numerator; % ИХ КИХ-ФИЛЬТРА
MAG = abs(freqz(hd,1,f,Fs)); % АЧХ КИХ-ФИЛЬТРА
figure('Name','Amplitude Spectrums and Magnitude Response in Polyphase
Structure Decimation System','NumberTitle','off')
subplot(2,1,1), stem(k*(Fs/N),MODX,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 Fs])
title(strcat('Amplitude Spectrum x(n) and FIR Magnitude Response'))
hold on, plot(f,MAG,'r','Linewidth',2)
subplot(2,1,2),stem(kd*((Fs/M)/length(Yd)),MODYd,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 Fs])
title(strcat(['M = ',num2str(M),'   Amplitude Spectrum y(n)']))
disp('%')
disp('%')
disp('% МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ДЕЦИМАЦИИ С ПОЛИФАЗНОЙ СТРУКТУРОЙ
ЗАВЕРШЕНО')

```

20.4.3. Система однократной передискретизации с полифазной структурой при повышении частоты дискретизации

Листинг script-файла lr_20_res_up имеет вид:

```

>> type lr_20_res_up
script
clc
disp('% ЛР №20. МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ПЕРЕДИСКРЕТИЗАЦИИ
С ПОЛИФАЗНОЙ СТРУКТУРОЙ')
disp('% ПРИ ПОВЫШЕНИИ ЧАСТОТЫ ДИСКРЕТИЗАЦИИ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;

```

```

while DATA==0
Nb = input('Nb = '); % НОМЕР БРИГАДЫ
A1 = input('A1 = '); % АМПЛИТУДЫ ДИСКРЕТНЫХ ГАРМОНИК ВХОДНОГО СИГНАЛА
A2 = input('A2 = ');
f1 = input('f1 = '); % ЧАСТОТЫ (Гц) ДИСКРЕТНЫХ ГАРМОНИК ВХОДНОГО СИГНАЛА
f2 = input('f2 = ');
Nri = input('Nri = '); % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs_ri = input('Fs_ri = '); % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
L = input('L = '); % КОЭФФИЦИЕНТ ИНТЕРПОЛЯЦИИ
M = input('M = '); % КОЭФФИЦИЕНТ ДЕЦИМАЦИИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для вывода значений Nri, Fs_ri и L/M нажмите <ENTER>')
pause
disp('%')
disp([' N = Nri = ',num2str(Nri), ' Fs = Fs_ri = ',num2str(Fs_ri)])
disp('%')
disp([' L/M = ',num2str(L), '/', num2str(M)])
N = Nri; % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs = Fs_ri; % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
n = 0:(N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ВХОДНОГО СИГНАЛА
x = A1*sin((2*pi*f1/Fs).*n)+A2*sin((2*pi*f2/Fs).*n); % ВХОДНОЙ СИГНАЛ
y = L*filter(Hri,x); % ВЫХОДНОЙ СИГНАЛ, УМНОЖЕННЫЙ НА КОЭФФИЦИЕНТ
ИНТЕРПОЛЯЦИИ
nri_start = ceil(length(y)/2+1); % НАЧАЛО УСТАНОВИВШЕГОСЯ РЕЖИМА
yri = y(nri_start:end); % ВЫХОДНОЙ СИГНАЛ В УСТАНОВИВШЕМСЯ РЕЖИМЕ
nri = nri_start:(nri_start+length(yri)-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
ДЛЯ УСТАНОВИВШЕГОСЯ РЕЖИМА
disp('%')
disp('% Для вывода ГРАФИКОВ ВХОДНОГО И ВЫХОДНОГО сигналов нажмите <ENTER>')
pause
figure('Name','Input and Output Signals in Polyphase Structure Resampling
System','NumberTitle','off')
subplot(3,1,1), stem(n,x), grid, xlabel('n')
title(strcat(['Input Signal x(n) N = ',num2str(N)]))
subplot(3,1,2), stem(0:length(y)-1,y), grid
title(strcat(['L = ',num2str(L), ' M = ',num2str(M), ' Output Signal y(n)
length(y) = ',num2str(length(y))]))
subplot(3,1,3), stem(nri,yri), grid, xlabel('n')
title(strcat(['Output Signal y(n) Starting point n = ',num2str(nri_start), '
length(y) = ',num2str(length(yri))]))
disp('%')
disp('%')

```

```

disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ и АЧХ КИХ-фильтра нажмите
<ENTER>')
pause
X = fft(x); % ДПФ ВХОДНОГО СИГНАЛА
MODX = (2/N)*abs(X); % АМПЛИТУДНЫЙ СПЕКТР ВХОДНОГО СИГНАЛА
MODX(1) = (1/N)*abs(X(1));
Yri = fft(yri); % ДПФ ВЫХОДНОГО СИГНАЛА В УСТАНОВИВШЕМСЯ РЕЖИМЕ
MODYri = (2/length(Yri))*abs(Yri); % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА
MODYri(1) = (1/length(Yri))*abs(Yri(1));
k = 0:(N-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВХОДНОГО СИГНАЛА
kri = 0:(length(Yri)-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВЫХОДНОГО СИГНАЛА
f = 0:(Fs*(L/M))/1000:Fs*(L/M); % СЕТКА ЧАСТОТ ДЛЯ АЧХ КИХ-ФИЛЬТРА
h = Hri.Numerator; % ИХ КИХ-ФИЛЬТРА
MAG = abs(freqz(h,1,f,Fs*(L/M))); % АЧХ КИХ-ФИЛЬТРА
figure('Name','Amplitude Spectrums and Magnitude Response in Polyphase
Structure Resampling System','NumberTitle','off')
subplot(2,1,1), stem(k*(Fs/N),MODX,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 (L/M)*Fs])
title(strcat(['Amplitude Spectrum x(n) N = ',num2str(N)]))
subplot(2,1,2)
stem(kri*(Fs*(L/M)/length(Yri)),MODYri,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 (L/M)*Fs])
title(strcat(['L/M = ',num2str(L),'/',num2str(M),' Amplitude Spectrum y(n) and
FIR Magnitude Response']))
hold on, plot(f,MAG,'r','Linewidth',2)
disp('%')
disp('%')
disp('% МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ПЕРЕДИСКРЕТИЗАЦИИ С ПОЛИФАЗНОЙ
СТРУКТУРОЙ ЗАВЕРШЕНО')

```

20.4.4. Система однократной передискретизации с полифазной структурой при понижении частоты дискретизации

Листинг script-файла `lr_20_res_down` имеет вид:

```

>> type lr_20_res_down
script
clc
disp('% ЛР №20. МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ПЕРЕДИСКРЕТИЗАЦИИ
С ПОЛИФАЗНОЙ СТРУКТУРОЙ')
disp('% ПРИ ПОНИЖЕНИИ ЧАСТОТЫ ДИСКРЕТИЗАЦИИ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ');
DATA=0;
while DATA==0

```

```

Nb = input('Nb = '); % НОМЕР БРИГАДЫ
A1 = input('A1 = '); % АМПЛИТУДЫ ДИСКРЕТНЫХ ГАРМОНИК ВХОДНОГО СИГНАЛА
A2 = input('A2 = ');
f1 = input('f1 = '); % ЧАСТОТЫ (Гц) ДИСКРЕТНЫХ ГАРМОНИК ВХОДНОГО СИГНАЛА
f2 = input('f2 = ');
Nrd = input('Nrd = '); % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs_rd = input('Fs_rd = '); % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
L = input('L = '); % КОЭФФИЦИЕНТ ИНТЕРПОЛЯЦИИ
M = input('M = '); % КОЭФФИЦИЕНТ ДЕЦИМАЦИИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для вывода значений Nrd, Fs_rd и L/M нажмите <ENTER>')
pause
disp('%')
disp(['      N = Nrd = ',num2str(Nrd), '      Fs = Fs_rd = ',num2str(Fs_rd)])
disp('%')
disp(['      L/M = ',num2str(L), '/' ,num2str(M)])
pause
N = Nrd; % ПЕРИОД ВХОДНОГО СИГНАЛА
Fs = Fs_rd; % ЧАСТОТА ДИСКРЕТИЗАЦИИ ВХОДНОГО СИГНАЛА
n = 0:(N-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ВХОДНОГО СИГНАЛА
x = A1*sin((2*pi*f1/Fs).*n)+A2*sin((2*pi*f2/Fs).*n); % ВХОДНОЙ СИГНАЛ
y = L*filter(Nrd,x); % ВЫХОДНОЙ СИГНАЛ, УМНОЖЕННЫЙ НА КОЭФФИЦИЕНТ
ИНТЕРПОЛЯЦИИ
nrd_start = ceil(length(y)/2+1); % НАЧАЛО УСТАНОВИВШЕГОСЯ РЕЖИМА
yrd = y(nrd_start:end); % ВЫХОДНОЙ СИГНАЛ В УСТАНОВИВШЕМСЯ РЕЖИМЕ
nrd = nrd_start:(nrd_start+length(yrd)-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
ДЛЯ УСТАНОВИВШЕГОСЯ РЕЖИМА
disp('%')
disp('% Для вывода ГРАФИКОВ ВХОДНОГО И ВЫХОДНОГО сигналов нажмите <ENTER>')
figure('Name','Input and Output Signals in Polyphase Structure Resampling
System','NumberTitle','off')
subplot(3,1,1), stem(n,x), grid, xlabel('n')
title(strcat(['Input Signal x(n)      N = ',num2str(N)]))
subplot(3,1,2), stem(0:length(y)-1,y), grid
title(strcat(['L = ',num2str(L),' M = ',num2str(M),' Output Signal y(n)
length(y) = ',num2str(length(y))]))
subplot(3,1,3), stem(nrd,yrd), grid, xlabel('n')
title(strcat(['Output Signal y(n) Starting point n = ',num2str(nrd_start),'
length(y) = ',num2str(length(yrd))]))
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ и АЧХ КИХ-фильтра нажмите
<ENTER>')

```

```

pause
X = fft(x); % ДПФ ВХОДНОГО СИГНАЛА
MODX = (2/N)*abs(X); % АМПЛИТУДНЫЙ СПЕКТР ВХОДНОГО СИГНАЛА
MODX(1) = (1/N)*abs(X(1));
Yrd = fft(yrd); % ДПФ ВЫХОДНОГО СИГНАЛА В УСТАНОВИВШЕМСЯ РЕЖИМЕ
MODYrd = (2/length(Yrd))*abs(Yrd); % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА
MODYrd(1) = (1/length(Yrd))*abs(Yrd(1));
k = 0:(N-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВХОДНОГО СИГНАЛА
krd = 0:(length(Yrd)-1); % ДИСКРЕТНЫЕ НОРМИРОВАННЫЕ ЧАСТОТЫ ВЫХОДНОГО СИГНАЛА
f = 0:Fs/1000:Fs; % СЕТКА ЧАСТОТ ДЛЯ АЧХ КИХ-ФИЛЬТРА
h = Hrd.Numerator; % ИХ КИХ-ФИЛЬТРА
MAG = abs(freqz(h,1,f,Fs)); % АЧХ КИХ-ФИЛЬТРА
figure('Name','Amplitude Spectrums and Magnitude Response in Polyphase
Structure Resampling System','NumberTitle','off')
subplot(2,1,1), stem(k*(Fs/N),MODX,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 Fs])
title(strcat('Amplitude Spectrum x(n) and FIR Magnitude Response'))
hold on, plot(f,MAG,'r','Linewidth',2)
subplot(2,1,2)
stem(krd*(Fs*(L/M)/length(Yrd)),MODYrd,'MarkerSize',3,'Linewidth',2)
grid, xlabel('f (Hz)'), xlim([0 Fs])
title(strcat(['L/M = ',num2str(L),'/',num2str(M),' Amplitude Spectrum y(n)']))
disp('%')
disp('%')
disp('% МОДЕЛИРОВАНИЕ СИСТЕМЫ ОДНОКРАТНОЙ ПЕРЕДИСКРЕТИЗАЦИИ С ПОЛИФАЗНОЙ
СТРУКТУРОЙ ЗАВЕРШЕНО')

```

20.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в моделировании полифазных структур систем однократной интерполяции, децимации и передискретизации средствами GUI FDATool и FilterBuilder GUI с использованием исходных данных табл. 20.1, но при других, выбираемых пользователем, коэффициентах интерполяции и децимации.

После экспорта полифазной структуры (объекта `mfilt`) в Workspace необходимо выполнить моделирование многоскоростной системы, используя соответствующий script-файл, для проверки корректности созданной в GUI полифазной структуры.

При моделировании систем однократной децимации и передискретизации обратить внимание на возможное растекание спектра и объяснить его причину.

20.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения пунктов задания, свойства объектов `mfilt`, копируемые из окна **Command Window** (шрифт Courier New), созданные графики (копируются при на-

жати комбинации клавиш <Alt>+<Print Screen>) и ответы на поставленные вопросы (шрифт Times New Roman).

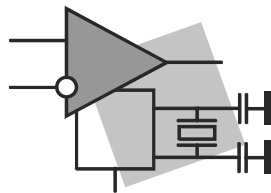
Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. Перечислите основные этапы моделирования полифазной структуры в GUI FDATool.
2. Сформулируйте требования к частотам КИХ-фильтра, работающего на "высокой" частоте дискретизации, для полифазной структуры системы однократной интерполяции.
3. Сформулируйте требования к частотам КИХ-фильтра, работающего на "высокой" частоте дискретизации, для полифазной структуры системы однократной децимации.
4. Сформулируйте требования к частотам КИХ-фильтра, работающего на "высокой" частоте дискретизации, для полифазной структуры системы однократной передискретизации.
5. На что влияют значения максимально допустимых отклонений АЧХ КИХ-фильтра, работающего на "высокой" частоте дискретизации?
6. Поясните процедуру синтеза КИХ-фильтра, работающего на "высокой" частоте дискретизации.
7. Поясните процедуру моделирования полифазной структуры многоскоростной системы.
8. Как осуществляется экспорт полифазной структуры из FDATool в Workspace?
9. Поясните свойства объекта `mfilt`.

20.7. Литература

1. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 17.
2. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов. — 2-е изд. — СПб.: БХВ-Петербург, 2005. — Глава 29.
3. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 10.
4. Оппенгейм А., Шафер Р. Цифровая обработка сигналов. — М.: Техносфера, 2007. — Глава 4.

ГЛАВА 21



Адаптивные фильтры

Цель работы: изучить методы и алгоритмы цифровой адаптивной фильтрации и овладеть средствами их программной реализации в MATLAB.

21.1. Краткая теоретическая справка

Использование традиционных ЦФ — КИХ- и БИХ-фильтров с постоянными коэффициентами — основано на предположении, что полосы частот полезного сигнала и шума известны и разнесены, поэтому коэффициенты ЦФ могут быть определены до обработки входного сигнала.

Однако во многих случаях входной сигнал не адекватен подобной модели, поэтому коэффициенты ЦФ невозможно определить заранее. В подобных случаях применяют адаптивные фильтры.

Адаптивным фильтром (АФ) называют систему, параметры которой адаптируются¹ (подстраиваются) к сигналу с заранее неопределенной статистической моделью в процессе его обработки.

Среди адаптивных фильтров наибольшее распространение получили *линейные адаптивные фильтры с обратной связью*, реализованные на основе *КИХ-фильтров*. Структурная схема (структура) такого АФ представлена на рис. 21.1.

Адаптивный фильтр включает в себя (см. рис. 21.1):

- КИХ-фильтр;
- алгоритм адаптации.

Параметрами АФ называют коэффициенты КИХ-фильтра, подстраиваемые к входному сигналу.

По определению, параметры АФ *изменяются* в процессе адаптивной фильтрации.

На вход АФ одновременно поступают два сигнала (см. рис. 21.1):

- входной сигнал $x(n)$ — заранее неизвестный;
- образцовый сигнал (desired signal) $d(n)$ — заранее известный.

¹ Отсюда название "адаптивный фильтр".

КИХ-фильтр

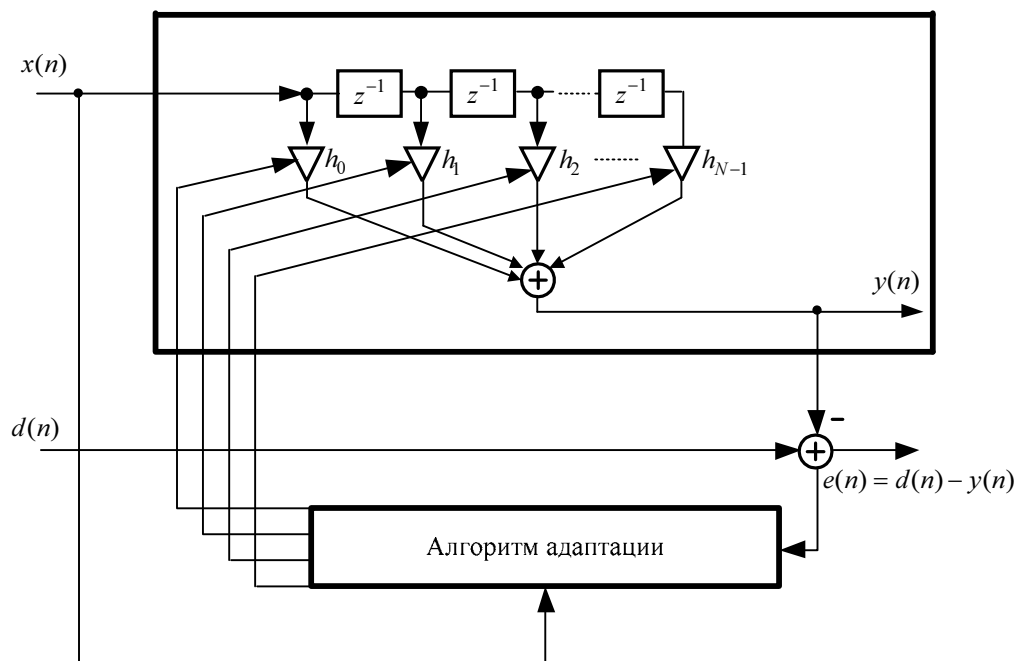


Рис. 21.1. Структурная схема линейного адаптивного фильтра с обратной связью

Пока мы не ставим вопрос о получении образцового сигнала, а лишь констатируем его доступность в ряде задач, решаемых средствами адаптивной фильтрации.

Сигнал $x(n)$ подается одновременно на вход КИХ-фильтра и алгоритма адаптации.

На выходе АФ формируются два сигнала:

- выходной сигнал $y(n)$.

Согласно разностному уравнению (РУ), выходной сигнал КИХ-фильтра равен линейной комбинации отсчетов входного сигнала:

$$y(n) = \sum_{i=0}^{N-1} h_i x(n-i), \quad (21.1)$$

где h_i , $i = 0, 1, \dots, N-1$, — коэффициенты КИХ-фильтра.

Напомним (см. разд. 11.1.1), что коэффициенты РУ КИХ-фильтра совпадают с отсчетами его импульсной характеристики (ИХ);

- сигнал ошибки $e(n)$ — разность между образцовым и выходным сигналами:

$$e(n) = d(n) - y(n). \quad (21.2)$$

Согласно (21.1), сигнал ошибки равен

$$e(n) = d(n) - \sum_{i=0}^{N-1} h_i x(n-i). \quad (21.3)$$

Под *линейностью* АФ понимают линейность входящего в его состав КИХ-фильтра. Для любого *фиксированного* набора коэффициентов КИХ-фильтр представляет собой линейную дискретную систему (ЛДС), которая удовлетворяет условиям аддитивности и однородности (см. разд. 8.1). Соотношение вход/выход ЛДС описывается *линейным* РУ (21.1) с *постоянными* коэффициентами h_i .

Однако в процессе адаптивной фильтрации параметры АФ (набор фиксированных коэффициентов КИХ-фильтра) *изменяются во времени*, подстраиваясь к входному сигналу с помощью *алгоритма адаптации*. Соотношение вход/выход АФ принимает вид *нелинейного* РУ с *переменными* коэффициентами $h_i(n)$:

$$y(n) = \sum_{i=0}^{N-1} h_i(n) x_k(n-i),$$

и АФ не удовлетворяет условиям аддитивности и однородности, а следовательно, представляет собой *нелинейную систему* обработки сигналов.

Таким образом, в *каждый момент времени* n КИХ-фильтр в составе АФ имеет фиксированные коэффициенты и представляет собой линейную систему, однако в целом в процессе адаптации АФ функционирует как нелинейная система.

Под *обратной связью* в традиционном ЦФ понимают связь текущего значения выходного сигнала с его предшествующими значениями, что имеет место только в БИХ-фильтрах. В АФ под *обратной связью* понимают связь *алгоритма адаптации* с сигналом ошибки $e(n)$, на основе которой подстраиваются коэффициенты КИХ-фильтра (изменяются параметры АФ).

Помимо линейности и обратной связи, для АФ вводится понятие *устойчивости*, которое также связано с *алгоритмом адаптации*. Под устойчивостью алгоритма адаптации понимают сходимость вычисляемых параметров АФ к оптимальным (по заданному критерию) параметрам. При этом для входящего в состав АФ КИХ-фильтра устойчивость гарантируется для любого набора фиксированных коэффициентов.

Целью адаптивной обработки является обеспечение наилучшего приближения выходного сигнала $y(n)$ к образцовому сигналу $d(n)$ по заданному критерию.

Основная сложность в проектировании АФ связана с разработкой алгоритмов адаптации.

21.1.1. Фильтр Винера

Фильтром Винера (Wiener filter) называют АФ, представленный на рис. 21.1, в котором алгоритм адаптации реализует вычисление *оптимальных* параметров АФ при выборе в качестве *критерия* наилучшего приближения выходного сигнала

$y(n)$ (21.1) к образцовому сигналу $d(n)$ минимума среднего квадрата сигнала ошибки:

$$M\{e^2(n)\} = M\left\{\left[d(n) - \sum_{i=0}^{N-1} h_i x(n-i)\right]^2\right\} \rightarrow \min_{\mathbf{h}}, \quad (21.4)$$

где $M\{\cdot\}$ — оператор математического ожидания; $\mathbf{h}$ — вектор параметров АФ (искомых коэффициентов h_i , $i = 0, 1, \dots, N-1$).

В предположении, что входной $x(n)$ и образцовый $d(n)$ сигналы являются эргодическими и коррелированными — случайными последовательностями длины L — средний квадрат сигнала ошибки определяется посредством усреднения по длине L (теоретически $L \rightarrow \infty$):

$$M\{e^2(n)\} = \frac{1}{L} \sum_{n=0}^{L-1} e^2(n), \quad (21.5)$$

или, с учетом (21.3):

$$M\{e^2(n)\} = \frac{1}{L} \sum_{n=0}^{L-1} \left[d(n) - \sum_{i=0}^{N-1} h_i x(n-i) \right]^2. \quad (21.6)$$

Вектор $\mathbf{h}$ находится в результате решения оптимизационной задачи — поиска минимума оптимизируемой (целевой) функции $F_1(\mathbf{h})$, которая, согласно (21.5), не зависит от времени:

$$F_1(\mathbf{h}) = M\{e^2(n)\} \rightarrow \min_{\mathbf{h}}. \quad (21.7)$$

Минимум функции $F_1(\mathbf{h})$ (21.7) достигается при равенстве нулю ее частных производных по всем h_i в (21.6); их совокупность можно записать в виде системы линейных алгебраических уравнений (СЛАУ):

$$\frac{2}{L} \sum_{n=0}^{L-1} \left[d(n) - \sum_{i=0}^{N-1} h_i x(n-i) \right] x(n-i) = 0, \quad (21.8)$$

откуда получаем СЛАУ относительно h_i :

$$\frac{1}{L} \sum_{n=0}^{L-1} \left[\sum_{i=0}^{N-1} h_i x(n-i) \right] x(n-i) = \frac{1}{L} \sum_{n=0}^{L-1} d(n) x(n-i). \quad (21.9)$$

В правой части (21.9) имеем значения оценки (при конечных длинах сигналов) взаимной корреляционной функции (ВКФ) $R_{dx}(i)$ между образцовым $d(n)$ и входным $x(n)$ сигналами:

$$R_{dx}(i) = \frac{1}{L} \sum_{n=0}^{L-1} d(n) x(n-i), \quad i = 0, 1, \dots, (N-1), \quad (21.10)$$

а в левой, меняя порядок суммирования, — сумму взвешенных значений оценки автокорреляционной функции (АКФ) входного сигнала $x(n)$ (с весовыми коэффициентами h_i):

$$\sum_{i=0}^{N-1} h_i \left\{ \frac{1}{L} \sum_{n=0}^{L-1} x(n-i)x(n) + \frac{1}{L} \sum_{n=0}^{L-1} x(n-i)x(n-1) + \frac{1}{L} \sum_{n=0}^{L-1} x(n-i)x(n-2) + \dots + \frac{1}{L} \sum_{n=0}^{L-1} x(n-i)x[n-(N-1)] \right\} = \sum_{i=0}^{N-1} h_i R_x(i-m), \quad m=0, 1, \dots, (N-1).$$

Таким образом, СЛАН (21.9) можно представить в виде:

$$\sum_{i=0}^{N-1} h_i R_x(i-m) = R_{dx}(m), \quad m=0, 1, \dots, (N-1), \quad (21.11)$$

или в матричной записи, известной как *система уравнений Винера—Хопфа* (Wiener—Hopf), с корреляционной матрицей Теплица (см. табл. 2.1):

$$\begin{bmatrix} R_x(0) & R_x(1) & \dots & R_x(N-1) \\ R_x(1) & R_x(0) & \dots & R_x(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_x(N-1) & R_x(N-2) & \dots & R_x(0) \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{N-1} \end{bmatrix} = \begin{bmatrix} R_{dx}(0) \\ R_{dx}(1) \\ \vdots \\ R_{dx}(N-1) \end{bmatrix}. \quad (21.12)$$

Краткая запись (21.12) имеет вид:

$$\mathbf{R}_x \mathbf{h} = \mathbf{R}_{dx}, \quad (21.13)$$

где:

□ $\mathbf{R}_x$ — корреляционная матрица входного сигнала;

□ $\mathbf{h}$ — вектор-столбец параметров АФ;

□ $\mathbf{R}_{dx}$ — вектор-столбец значений ВКФ между образцовым и входным сигналами.

Решением СЛАН (21.13) является вектор *оптимальных* (по критерию минимума среднего квадрата сигнала ошибки) параметров АФ $\mathbf{h}_{opt}$:

$$\mathbf{h}_{opt} = \mathbf{R}_x^{-1} \mathbf{R}_{dx} \quad (21.14)$$

где $\mathbf{R}_x^{-1}$ — обратная корреляционная матрица.

Фильтр Винера может использоваться для обработки эргодических сигналов со статистическими характеристиками, инвариантными во времени.

Однако в общем случае, при обработке сигнала, статистическая модель которого заранее не определена, практическое применение фильтра Винера ограничено вследствие вычислительной сложности алгоритма адаптации. В этом случае принято говорить о *текущих* оценках статистических характеристик на интервале $[0; n]$ (см. разд. 7.1.2), которые приходится определять заново с поступлением каждого нового отсчета сигнала. Следовательно, требуется заново определять корреля-

ционную матрицу и вектор взаимных корреляций и повторять трудоемкую операцию обращения матрицы при вычислении оптимальных параметров АФ¹ $\mathbf{h}(n)$ (21.14).

Для сокращения вычислительных затрат разработаны алгоритмы *рекуррентного* вычисления параметров АФ. В следующих разделах будут рассмотрены два таких алгоритма:

□ алгоритм LMS (Least Mean Squares) и его модификация NLMS (Normalized LMS).

В отечественной литературе иногда используют аббревиатуры МНК (метод наименьших квадратов) и НМНК (нормализованный метод наименьших квадратов);

□ алгоритм RLS (Recursive Least Squares).

В отечественной литературе — РНК (рекуррентный метод наименьших квадратов).

21.1.2. Алгоритм LMS

В алгоритме LMS реализовано *рекуррентное* вычисление оценок параметров АФ $\hat{\mathbf{h}}(n)$.

В качестве *критерия* наилучшего приближения выходного сигнала $y(n)$ к образцовому сигналу $d(n)$ используется *минимум квадрата сигнала ошибки*:

$$e^2(n) = \left[d(n) - \sum_{i=0}^{N-1} h_i x(n-i) \right]^2 \rightarrow \min_{\mathbf{h}}. \quad (21.15)$$

В отличие от оптимизируемой функции $F_1(\mathbf{h})$ (21.7) в фильтре Винера, в данном случае оптимизируемая функция является функцией времени n :

$$F_2(n, \mathbf{h}) = e^2(n) = \left[d(n) - \sum_{i=0}^{N-1} h_i x(n-i) \right]^2. \quad (21.16)$$

Для поиска минимума функции $F_2(n, \mathbf{h})$ используют градиентный *метод наискорейшего спуска* — итерационную процедуру, определяющую траекторию наискорейшего пошагового приближения (спуска) к минимуму, где шагам итерации соответствуют *моменты дискретного нормированного времени* n .

В соответствии с данным методом, на каждом шаге итерации оценивается вектор $\hat{\mathbf{h}}(n+1)$, смещаемый относительно вектора $\hat{\mathbf{h}}(n)$ на величину, пропорциональную *градиенту*² функции $F_2(n, \mathbf{h})$ в точке n :

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \frac{\mu}{2} \nabla_n, \quad (21.17)$$

¹ Индекс "opt" снят, т. к., строго говоря, оптимальные коэффициенты определяются в установившемся режиме.

² От лат. *gradiens* — шагающий.

где $\hat{\mathbf{h}}(n+1)$ — вектор оценок параметров АФ в момент времени $(n+1)$; μ — положительная константа, называемая *шагом адаптации*; ∇_n — градиент функции $F_2(n, \mathbf{h})$ (21.16), определяемый как вектор, элементами которого являются частные производные данной функции по всем h_i в момент времени n :

$$\nabla_n = 2 \left[d(n) - \sum_{i=0}^{N-1} h_i x(n-i) \right] x(n-i), \quad (21.18)$$

или, с учетом (21.3):

$$\nabla_n = 2e(n)x(n-i), \quad i = 0, 1, \dots, N-1. \quad (21.19)$$

Подставляя (21.19) в (21.17), получаем *рекуррентную формулу* для оценок параметров АФ:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu e(n)\mathbf{x}(n), \quad (21.20)$$

где $\mathbf{x}(n)$ — вектор отсчетов входного сигнала $x(n-i)$, $i = 0, 1, \dots, N-1$.

Значение шага адаптации μ влияет на скорость сходимости оценок параметров АФ к оптимальным параметрам $\mathbf{h}_{opt}$ в фильтре Винера.

При выборе значения μ из диапазона

$$0 < \mu < 2/\lambda_{\max}, \quad (21.21)$$

где $\lambda_{\max}$ — максимальное собственное значение матрицы¹ АКФ $\mathbf{R}_x$ в (21.13), гарантируется *сходимость в среднем* — для средних значений оценок параметров АФ при $n \rightarrow \infty$.

Практический интерес представляет диапазон значений шага адаптации μ , гарантирующих *сходимость в среднем квадрате*² — для средних квадратов оценок параметров АФ, при условии, что последние стремятся к фиксированным значениям. Это обеспечивается для значений μ в диапазоне [3]:

$$0 < \mu \leq \frac{2}{NP_x}, \quad (21.22)$$

где P_x — средний квадрат входного сигнала $x(n)$ длины L :

$$P_x = \frac{1}{L} \sum_{n=0}^{L-1} x^2(n). \quad (21.23)$$

Значение шага адаптации μ выбирается из компромиссных соображений: с одной стороны, оно влияет на *скорость сходимости* алгоритма LMS (чем больше μ , тем

¹ С этим понятием можно познакомиться в разд. 3.8.6 [2].

² Сходимость в среднем квадрате гарантирует сходимость в среднем, но не наоборот.

она выше), а с другой — на сигнал ошибки $e(n)$ (чем больше μ , тем больше он отличается от сигнала ошибки в фильтре Винера).

В нормализованном алгоритме NLMS рекуррентная формула (21.20) заменяется другой, в которой шаг адаптации μ зависит от времени:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu(n)e(n)\mathbf{x}(n), \quad (21.24)$$

т. к. он нормируется к энергии сигнала $\mathbf{x}(n-i)$, $i = 0, 1, \dots, N-1$:

$$\mu(n) = \frac{\mu_0}{\mathbf{x}'(n)\mathbf{x}(n) + \varepsilon}, \quad (21.25)$$

где $\mathbf{x}'(n)\mathbf{x}(n)$ — энергия сигнала, равная произведению вектора-строки¹ $\mathbf{x}'(n)$ на вектор-столбец $\mathbf{x}(n)$; μ_0 — фиксированное значение шага, влияющее на сходимость алгоритма адаптации, выбираемое из диапазона $0 < \mu_0 < 2$; ε — малая положительная константа, определяющая максимальное значение $\mu(n)$, равное μ_0/ε при нулевом входном сигнале.

Итерационная процедура вычисления оценок параметров АФ в алгоритмах LMS и NLMS включает в себя следующие шаги:

1. Присваивание $n = 0$ и задание начальных (обычно нулевых) значений оценок параметров АФ $\hat{\mathbf{h}}(0)$.
2. Вычисление выходного сигнала $y(n)$ (21.1).
3. Вычисление сигнала ошибки $e(n)$ (21.3).
4. Обновление оценок параметров АФ $\hat{\mathbf{h}}(n+1)$, согласно (21.20) или (21.24).
5. Присваивание $n = n+1$.
6. Повторение пп. 2—5.

Основным достоинством алгоритма LMS является его простота (на каждом шаге требуется всего N операций умножения-сложения), а недостатком — относительно медленная сходимость итерационной процедуры вычисления параметров АФ.

В MATLAB структура АФ (см. рис. 21.1) с алгоритмом LMS описывается в виде объекта `adaptfilt` (от англ. *Adaptive Filter*):

```
hlms = adaptfilt.lms(N, step, leakage, coeffs, states)
```

где `hlms` — имя объекта; `lms` — алгоритм адаптации (LMS); N — длина КИХ-фильтра N , по умолчанию, в отсутствии данного параметра, $N = 10$; `step` — шаг адаптации μ , по умолчанию, в отсутствии данного параметра, `step = 0.1`; `leakage` — коэффициент утечки (Leakage Factor) — множитель при $\hat{\mathbf{h}}(n)$ в (21.20),

¹ Здесь и далее символ "апостроф" соответствует операции транспонирования вектора или матрицы.

влияющий на сходимость алгоритма LMS; выбирается из диапазона $0 < \text{leakage} \leq 1$, по умолчанию, в отсутствии данного параметра, $\text{leakage} = 1$; **coeffs** — вектор длины N начальных значений оценок параметров АФ $\hat{\mathbf{h}}(0)$, по умолчанию, в отсутствии данного параметра, нулевых; **states** — вектор длины $(N-1)$ начальных условий КИХ-фильтра, по умолчанию, в отсутствии данного параметра, ННУ (см. разд. 8.1).

Свойства объекта `adaptfilt.lms`, выводимые по его имени, включают в себя:

- **Algorithm** — структура КИХ-фильтра и алгоритм адаптации;
- **FilterLength** — значение параметра N ;
- **StepSize** — значение параметра step ;
- **Leakage** — значение параметра leakage ;
- **PersistentMemory** — начальные условия **states**; значение `false` соответствует ННУ.

Если заданное при описании объекта `Hlms` значение μ (параметра step) выйдет за границы диапазона в (21.22), то будет выдано предупреждение (Warning).

Структура АФ (см. рис. 21.1) с алгоритмом NLMS описывается в виде объекта:

`Hnlms = adaptfilt.nlms(N, step, leakage, offset, coeffs, states)`

где `Hnlms` — имя объекта; `nlms` — алгоритм адаптации (NLMS); step — значение μ_0 в (21.25), выбираемое из диапазона $0 < \text{step} < 2$, по умолчанию, в отсутствии данного параметра, $\text{step} = 1$; **offset** — константа ε в (21.25), по умолчанию, в отсутствии данного параметра, $\text{offset} = 0$.

Остальные параметры определены ранее для объекта `adaptfilt.lms`.

Свойства объекта `adaptfilt.nlms` дублируют свойства объекта `adaptfilt.lms` и дополнительно включают в себя:

Offset — значение параметра **offset**.

При известной структуре АФ вычисление выходного сигнала $y(n)$ и сигнала ошибки $e(n)$ выполняется с помощью функции:

`[y,e] = filter(H,x,d)`

где **H** — имя объекта `adaptfilt` (`Hlms` или `Hnlms`); **x**, **d** — векторы отсчетов входного и образцового сигналов; **y**, **e** — векторы отсчетов выходного сигнала и сигнала ошибки.

По завершении процесса адаптивной фильтрации можно вывести параметры АФ:

`H.coefficients`

21.1.3. Алгоритм RLS

В алгоритме RLS реализовано *рекуррентное* вычисление *оптимальных* параметров АФ $\mathbf{h}(n)$ (в установившемся режиме¹) с существенным сокращением вычислительных затрат, по сравнению с их *нерекуррентным* вычислением (см. разд. 21.1.1).

В качестве *критерия* наилучшего приближения выходного сигнала $y(n)$ к образцовому сигналу $d(n)$ используется *минимум суммы квадратичных значений сигнала ошибки*²:

$$\sum_{n=0}^{L-1} e^2(n) \rightarrow \min_{\mathbf{h}}, \quad (21.26)$$

или, с учетом (21.3):

$$\sum_{n=0}^{L-1} \left[d(n) - \sum_{i=0}^{N-1} h_i x(n-i) \right]^2 \rightarrow \min_{\mathbf{h}}, \quad (21.27)$$

где L — длина входного сигнала $x(n)$, а $\mathbf{h}$ — вектор параметров АФ.

Подобно оптимизируемой функции $F_1(\mathbf{h})$ (21.7) в фильтре Винера, в данном случае оптимизируемая функция не является функцией времени n :

$$F_3(\mathbf{h}) = \sum_{n=0}^{L-1} \left[d(n) - \sum_{i=0}^{N-1} h_i x(n-i) \right]^2. \quad (21.28)$$

Минимум функции $F_3(\mathbf{h})$ (21.29) достигается при равенстве нулю частных производных по всем h_i в (21.28); их совокупность можно записать в виде СЛАУ:

$$2 \sum_{n=0}^{L-1} \left[d(n) - \sum_{i=0}^{N-1} h_i x(n-i) \right] x(n-i) = 0, \quad (21.29)$$

откуда получаем СЛАУ относительно h_i :

$$\sum_{n=0}^{L-1} \left[\sum_{i=0}^{N-1} h_i x(n-i) \right] x(n-i) = \sum_{n=0}^{L-1} d(n)x(n-i), \quad i = 0, 1, \dots, N-1, \quad (21.30)$$

совпадающую (с точностью до множителя $1/L$) со СЛАУ (21.9) для фильтра Винера.

¹ В начале процесса адаптации в течение некоторого времени наблюдается переходный процесс, во время которого дисперсия сигнала ошибки оказывается много большей, чем по его окончании — в установившемся режиме.

² Минимум суммарной квадратичной ошибки.

Запишем в матричном виде левую часть (21.30) с учетом ННУ ($x(n)|_{(n-i)<0} = 0$):

$$\begin{bmatrix} x(0) & 0 & \dots & 0 \\ x(1) & x(0) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ x(L-1) & x[(L-1)-1] & \dots & x[(L-1)-(N-1)] \end{bmatrix} \times \begin{bmatrix} x(0) & x(1) & \dots & x(L-1) \\ 0 & x(0) & \dots & x[(L-1)-1] \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & x[(L-1)-(N-1)] \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{N-1} \end{bmatrix} \quad (21.31)$$

и правую часть (21.30):

$$\begin{bmatrix} x(0) & x(1) & \dots & x(L-1) \\ 0 & x(0) & \dots & x[(L-1)-1] \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & x[(L-1)-(N-1)] \end{bmatrix} \begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(L-1) \end{bmatrix}. \quad (21.32)$$

Краткая матричная запись СЛАУ (21.30) имеет вид:

$$\mathbf{XX}'\mathbf{h} = \mathbf{Xd}, \quad (21.33)$$

где:

- $\mathbf{X}$ — матрица размером $N \times L$ отсчетов входного сигнала $x(n)$ длины L ;
- $\mathbf{X}'$ — транспонированная матрица $\mathbf{X}$ размером $L \times N$;
- $\mathbf{XX}'$ — квадратная матрица порядка N ;
- $\mathbf{h}$ — вектор-столбец параметров АФ длины N ;
- $\mathbf{XX}'\mathbf{h}$ — вектор-столбец длины N ;
- $\mathbf{d}$ — вектор-столбец отсчетов образцового сигнала $d(n)$ длины L ;
- $\mathbf{Xd}$ — вектор-столбец длины N .

Матрица $\mathbf{XX}'$ и вектор $\mathbf{Xd}$ совпадают (с точностью до множителя $1/L$) соответственно с корреляционной матрицей $\mathbf{R}_x$ и вектором ВКФ $\mathbf{R}_{dx}$ в (21.12), поэтому решением СЛАУ (21.33) является вектор *оптимальных* параметров АФ $\mathbf{h}_{opt}$, как в фильтре Винера (21.14):

$$\mathbf{h}_{opt} = (\mathbf{XX}')^{-1} \mathbf{Xd}. \quad (21.34)$$

Ранее говорилось (см. разд. 21.1.1), что в общем случае, когда статистическая модель входного сигнала заранее не определена, переходят к вычислению *оптимальных* параметров АФ $\mathbf{h}(n)$.

С учетом этого запишем (21.34) в виде:

$$\mathbf{h}(n) = [\mathbf{X}(n)\mathbf{X}'(n)]^{-1}\mathbf{X}(n)\mathbf{d}(n), \quad (21.35)$$

где $\mathbf{X}(n)$ и $\mathbf{d}(n)$ вычисляются на *текущем интервале* $[0; n]$ (сравните с (21.32)):

$$\mathbf{X}(n) = \begin{bmatrix} x(0) & x(1) & \dots & x(n) \\ 0 & x(0) & \dots & x(n-1] \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & x[n-(N-1)] \end{bmatrix}; \quad \mathbf{d}(n) = \begin{bmatrix} d(0) \\ d(1) \\ \vdots \\ d(n) \end{bmatrix}. \quad (21.36)$$

Теперь запишем (21.35) для следующего $(n+1)$ -го момента времени:

$$\mathbf{h}(n+1) = [\mathbf{X}(n+1)\mathbf{X}'(n+1)]^{-1}\mathbf{X}(n+1)\mathbf{d}(n+1), \quad (21.37)$$

и увидим, что это приведет к добавлению в матрице $\mathbf{X}(n)$ нового столбца $\mathbf{x}(n+1)$:

$$\mathbf{x}(n+1) = \begin{bmatrix} x(n+1) \\ x[(n+1)-1] \\ \vdots \\ x[(n+1)-(N-1)] \end{bmatrix} \quad (21.38)$$

и в векторе $\mathbf{d}(n)$ — нового элемента $d(n+1)$.

В результате для произведения $\mathbf{X}(n+1)\mathbf{X}'(n+1)$ имеем рекуррентную связь с произведением $\mathbf{X}(n)\mathbf{X}'(n)$:

$$\mathbf{X}(n+1)\mathbf{X}'(n+1) = \mathbf{X}(n)\mathbf{X}'(n) + \mathbf{x}(n+1)\mathbf{x}'(n+1), \quad (21.39)$$

а для произведения $\mathbf{X}(n+1)\mathbf{d}(n+1)$ — рекуррентную связь с произведением $\mathbf{X}(n)\mathbf{d}(n)$:

$$\mathbf{X}(n+1)\mathbf{d}(n+1) = \mathbf{X}(n)\mathbf{d}(n) + \mathbf{x}(n+1)d(n+1). \quad (21.40)$$

Для получения рекуррентной связи обратной матрицы $[\mathbf{X}(n+1)\mathbf{X}'(n+1)]^{-1}$ в (21.37) с матрицей $[\mathbf{X}(n)\mathbf{X}'(n)]^{-1}$ введем краткие обозначения:

$$\begin{aligned} \mathbf{P}(n) &= [\mathbf{X}(n)\mathbf{X}'(n)]^{-1}; \\ \mathbf{P}(n+1) &= [\mathbf{X}(n+1)\mathbf{X}'(n+1)]^{-1} \end{aligned} \quad (21.41)$$

и воспользуемся тождеством из линейной алгебры [3]:

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}. \quad (21.42)$$

Сопоставляя левую часть (21.42) с правой частью (21.39), видим следующее соответствие:

□ $\mathbf{A} = \mathbf{X}(n)\mathbf{X}'(n)$ — квадратная матрица размером $N \times N$ (см. (21.36));

□ $\mathbf{B} = \mathbf{x}(n+1)$ — вектор-столбец (21.38) длины N ;

□ $\mathbf{C} = 1$ — скаляр;

□ $\mathbf{D} = \mathbf{x}'(n+1)$ — вектор-строка длины N .

Используя тождество (21.42), на основе (21.39) запишем обратную матрицу $[\mathbf{X}(n+1)\mathbf{X}'(n+1)]^{-1}$ с учетом обозначений (21.41):

$$\mathbf{P}(n+1) = \mathbf{P}(n) - \mathbf{P}(n)\mathbf{x}(n+1)[1 + \mathbf{x}'(n+1)\mathbf{P}(n)\mathbf{x}(n+1)]^{-1}\mathbf{x}'(n+1)\mathbf{P}(n). \quad (21.43)$$

Результатом выполнения операций в квадратных скобках будет *скаляр* (проверьте самостоятельно), для которого операции обращения соответствует возведение в минус первую степень:

$$\mathbf{P}(n+1) = \mathbf{P}(n) - \frac{\mathbf{P}(n)\mathbf{x}(n+1)\mathbf{x}'(n+1)\mathbf{P}(n)}{1 + \mathbf{x}'(n+1)\mathbf{P}(n)\mathbf{x}(n+1)}. \quad (21.44)$$

Таким образом, получена рекуррентная связь матриц $\mathbf{P}(n+1)$ и $\mathbf{P}(n)$.

Запишем (21.37) с учетом обозначения $\mathbf{P}(n+1)$ (21.41) и связи (21.40):

$$\mathbf{h}(n+1) = \mathbf{P}(n+1)[\mathbf{X}(n)\mathbf{d}(n) + \mathbf{x}(n+1)d(n+1)]. \quad (21.45)$$

Подставив в (21.45) $\mathbf{P}(n+1)$ (21.44), раскрыв скобки и приведя подобные слагаемые [3], получим *рекуррентную формулу* для *оптимальных* параметров АФ (в установленном режиме):

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mathbf{K}(n+1)e(n+1), \quad (21.46)$$

где $\mathbf{K}(n+1)$ — вектор коэффициентов усиления длины N :

$$\mathbf{K}(n+1) = \frac{\mathbf{P}(n)\mathbf{x}(n+1)}{1 + \mathbf{x}'(n+1)\mathbf{P}(n)\mathbf{x}(n+1)}, \quad (21.47)$$

$e(n+1)$ — сигнал ошибки (21.3), определяемый по параметрам АФ на n -м шаге:

$$e(n+1) = d(n+1) - \mathbf{x}'(n+1)\mathbf{h}(n). \quad (21.48)$$

Итерационная процедура вычисления оптимальных параметров АФ в алгоритме RLS включает в себя следующие шаги:

1. Задание начальных (обычно нулевых) значений параметрам АФ $\mathbf{h}(-1)$.
2. Задание начальных значений элементам матрицы $\mathbf{P}(-1)$. Рекомендуется *диагональным* элементам присваивать большие положительные значения, а остальным — нулевые.
3. Присваивание $n = 0$.

4. Вычисление выходного сигнала $y(n)$ и сигнала ошибки $e(n)$ (см. (21.48)):

$$\begin{aligned} y(n) &= \mathbf{x}'(n)\mathbf{h}(n-1); \\ e(n) &= d(n) - y(n). \end{aligned}$$

5. Вычисление вектора коэффициентов усиления $\mathbf{K}(n)$ (см. (21.47)):

$$\mathbf{K}(n) = \frac{\mathbf{P}(n-1)\mathbf{x}(n)}{1 + \mathbf{x}'(n)\mathbf{P}(n-1)\mathbf{x}(n)}. \quad (21.49)$$

6. Обновление матрицы $\mathbf{P}(n)$ (см. (21.44) с учетом (21.49)):

$$\mathbf{P}(n) = \mathbf{P}(n-1) - \mathbf{K}(n)\mathbf{x}'(n)\mathbf{P}(n-1). \quad (21.50)$$

7. Обновление вектора $\mathbf{h}(n)$ (см. (21.46)):

$$\mathbf{h}(n) = \mathbf{h}(n-1) + \mathbf{K}(n)e(n). \quad (21.51)$$

8. Присваивание $n = n + 1$.

9. Повторение пп. 4—8.

Как видим, вычислительная сложность алгоритма RLS существенно возрастает, по сравнению с алгоритмом LMS. Однако этой ценой удастся определить *оптимальные* параметры АФ в установившемся режиме, в то время как в алгоритме LMS определяются их оценки.

Критерий (21.26) можно модифицировать, добавив в него *коэффициент забывания* λ , выбираемый из диапазона $0 < \lambda \leq 1$, что позволяет учитывать предшествующие отсчеты входного сигнала с экспоненциально уменьшаемым весом:

$$\sum_{n=0}^{L-1} \lambda^{(L-1)-n} e^2(n) \rightarrow \min_{\mathbf{h}}. \quad (21.52)$$

В этом случае формулы (21.49) и (21.50) принимают вид:

$$\mathbf{K}(n) = \frac{\mathbf{P}(n-1)\mathbf{x}(n)}{\lambda + \mathbf{x}'(n)\mathbf{P}(n-1)\mathbf{x}(n)}; \quad (21.53)$$

$$\mathbf{P}(n) = \frac{1}{\lambda} [\mathbf{P}(n-1) - \mathbf{K}(n)\mathbf{x}(n)\mathbf{P}(n-1)], \quad (21.54)$$

а в (21.51) определяются *оценки* параметров АФ.

В MATLAB структура АФ (см. рис. 21.1) с алгоритмом RLS описывается в виде объекта:

`Hrls = adaptfilt_rls(N, lambda, invcov, coeffs, states)`

где `Hrls` — имя объекта; `rls` — алгоритм адаптации (RLS); `lambda` — коэффициент забывания λ , выбираемый из диапазона $0 < \lambda \leq 1$, по умолчанию, в отсутствии данного параметра `lambda = 1`; `invcov` — начальные значения элементов матрицы $\mathbf{P}(-1)$, по умолчанию, в отсутствии данного параметра, формируется матрица

$1000 \times \text{eye}(N)$ (см. табл. 2.1); `coeffs` — вектор длины N начальных значений параметров АФ $\mathbf{h}(-1)$, по умолчанию, в отсутствии данного параметра, нулевых.

Параметры `N` и `states` определены ранее для объекта `adaptfilt.lms` (см. разд. 21.1.2).

Свойства объекта `adaptfilt.rls`, выводимые по его имени, включают в себя:

- ☐ `Algorithm` — структура КИХ-фильтра и алгоритм адаптации;
- ☐ `FilterLength` — значение параметра `N`;
- ☐ `ForgettingFactor` — значение параметра `lambda`;
- ☐ `Leakage` — значение параметра `leakage`;
- ☐ `PersistentMemory` — начальные условия `states`; значение `false` соответствует ННУ.

При известной структуре АФ вычисление выходного сигнала $y(n)$ и сигнала ошибки $e(n)$ выполняется с помощью функции `filter` (см. разд. 21.1.2).

По завершении процесса адаптивной фильтрации можно вывести параметры АФ:

`hrls.coefficients`

21.1.4. Применение адаптивных фильтров

Рассмотрение многочисленных практических применений АФ для решения задач ЦОС выходит за рамки данной книги. В учебных целях мы ограничимся моделированием типовых методов адаптивной фильтрации, лежащих в основе многих практических приложений, к числу которых относятся:

- ☐ оценка импульсной характеристики неизвестной системы;
- ☐ очистка сигнала от шума;
- ☐ выравнивание частотной характеристики неизвестной системы;
- ☐ оценка параметров линейного предсказания сигнала.

21.1.4.1. Идентификация систем

Применение адаптивных фильтров во многих случаях сводится к задаче идентификации неизвестной системы.

Под идентификацией неизвестной системы понимают процесс, в результате которого обеспечивается совпадение или сходство (по заданному критерию) ее входного и выходного сигналов с входным и выходным сигналами известной системы. По завершении данного процесса структура и параметры известной системы могут совпадать или не совпадать со структурой и параметрами известной системы.

Различают две разновидности идентификации систем:

- ☐ прямая идентификация — это процесс, в результате которого обеспечивается совпадение входных сигналов неизвестной и известной систем и сходство (по заданному критерию) их выходных сигналов;

□ обратная идентификация — это процесс, в результате которого обеспечивается совпадение *выходного* сигнала неизвестной системы с *входным* сигналом известной системы и сходство (по заданному критерию) *входного* сигнала неизвестной системы с *выходным* сигналом известной системы.

Рассмотрим идентификацию *неизвестной* системы с использованием в качестве *известной* системы адаптивного фильтра (см. рис. 21.1).

Структурная схема *прямой* идентификации приведена на рис. 21.2, а.

При *прямой* идентификации *входные* сигналы неизвестной системы и АФ совпадают. В качестве образцового сигнала АФ $d(n)$ используется *выходной* сигнал неизвестной системы. В процессе адаптации АФ стремится преобразовать входной сигнал $x(n)$ так, чтобы обеспечить наилучшее приближение (по заданному критерию) *выходного* сигнала АФ $y(n)$ к *выходному* сигналу неизвестной системы $d(n)$.

Решение задачи *прямой* идентификации заключается в определении *параметров* АФ, обеспечивающих наилучшее приближение (по заданному критерию) выходных сигналов АФ и неизвестной системы. В этом смысле говорят, что АФ представляет собой *модель* неизвестной системы [4].

Примеры решения задач на основе прямой идентификации рассматриваются в разд. 21.1.4.2—21.1.4.3.

Структурная схема *обратной* идентификации приведена на рис. 21.2, б.

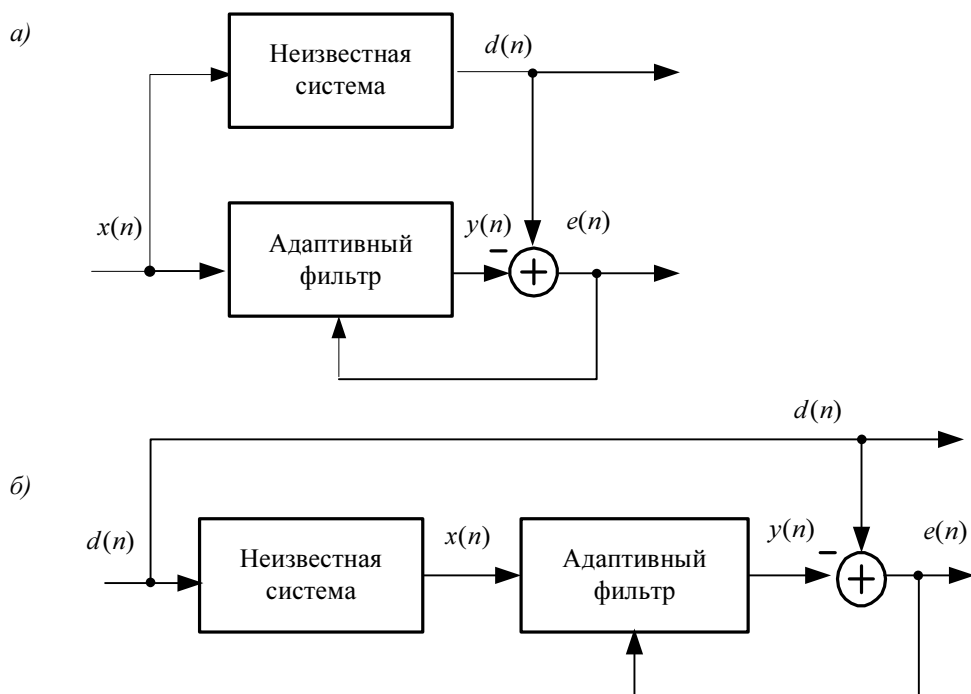


Рис. 21.2. Структурная схема идентификации систем: прямой (а), обратной (б)

При *обратной* идентификации *выходной* сигнал неизвестной системы $x(n)$ совпадает с *входным* сигналом АФ. В качестве образцового сигнала АФ $d(n)$ используется *входной* сигнал неизвестной системы. В процессе адаптации АФ стремится преобразовать входной сигнал $x(n)$ так, чтобы обеспечить наилучшее приближение (по заданному критерию) *выходного* сигнала АФ $y(n)$ к *входному* сигналу неизвестной системы $d(n)$.

Решение задачи *обратной* идентификации заключается в определении *параметров* АФ, обеспечивающих наилучшее приближение (по заданному критерию) выходного сигнала АФ к входному сигналу неизвестной системы. В этом смысле говорят, что АФ представляет собой *обратную модель* неизвестной системы [4].

Пример решения задачи на основе обратной идентификации рассматривается в разд. 21.1.4.4.

21.1.4.2. Оценка импульсной характеристики неизвестной системы

Задача оценки импульсной характеристики (ИХ) неизвестной системы, в предположении, что она представляет собой ЛДС (КИХ- или БИХ-фильтр), в *условиях случайного воздействия* сводится к решению задачи *прямой* идентификации (см. рис. 21.2).

При *прямой* идентификации АФ представляет собой *модель* неизвестной системы (см. разд. 21.1.4.1), на основании чего полагают, что по завершении процесса адаптации ИХ КИХ-фильтра в составе АФ будет *оценкой* ИХ неизвестной системы на интервале дискретного нормированного времени, равном длине КИХ-фильтра.

Моделирование процесса вычисления оценки ИХ неизвестной системы в MATLAB включает в себя следующие шаги:

1. Моделирование входного сигнала неизвестной системы — входного сигнала АФ $x(n)$, в качестве которого можно выбрать нормальный или равномерный белый шум.
2. Моделирование неизвестной системы — КИХ- или БИХ-фильтра.
3. Вычисление выходного сигнала неизвестной системы — образцового сигнала АФ $d(n)$ с помощью функции `filter`.
4. Вычисление истинной ИХ неизвестной системы $h(n)$.
5. Моделирование структуры АФ — объекта `adaptfilt`.
6. Моделирование процесса адаптивной фильтрации — вычисление выходного сигнала АФ $y(n)$ и сигнала ошибки $e(n)$ с помощью функции `filter`.
7. Определение параметров АФ — оценки ИХ $\hat{h}(n)$ неизвестной системы:

`H.coefficients`

где `H` — имя объекта `adaptfilt`.

8. Вывод графика сигнала ошибки АФ $e(n)$.
9. Вывод графиков истинной ИХ $h(n)$ и ее оценки $\hat{h}(n)$ на интервале дискретного нормированного времени $[0; (N-1)]$, где N — длина КИХ-фильтра в составе АФ.
10. Сравнение оценки $\hat{h}(n)$ с истинной ИХ $h(n)$ по критерию среднего абсолютно-го отклонения их отсчетов на основе нормы $\|\mathbf{x}\|_1$ (см. разд. 2.1.5).

21.1.4.3. Очистка сигнала от шума

Задача оценки полезного сигнала в аддитивной смеси с шумом — *очистки сигнала от шума* — сводится к решению задачи *прямой* идентификации, структурная схема которой представлена на рис. 21.3.

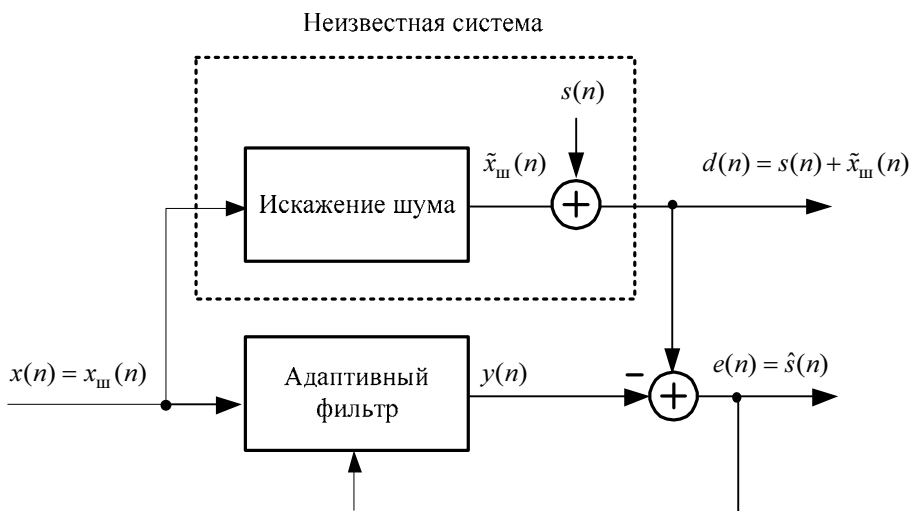


Рис. 21.3. Структурная схема прямой идентификации при очистке сигнала от шума

При *прямой* идентификации *входные* сигналы неизвестной системы и АФ совпадают и представляют собой шум, доступный для наблюдения $x(n) = x_{\text{ш}}(n)$. Под *искаженным шумом* $\tilde{x}_{\text{ш}}(n)$ понимают шум $x_{\text{ш}}(n)$, измененный под влиянием каких-либо факторов, например внешней среды, при разнесении источников шума и полезного сигнала. В обобщенной постановке задачи, которая рассматривается в данном разделе, искажение шума $x_{\text{ш}}(n)$ моделируется посредством его обработки (преобразования) линейной системой в виде КИХ- или БИХ-фильтра. При этом предполагается, что искаженный шум $\tilde{x}_{\text{ш}}(n)$ будет коррелирован с исходным шумом $x_{\text{ш}}(n)$, но шум $x_{\text{ш}}(n)$ — не коррелирован с полезным сигналом $s(n)$.

В качестве образцового сигнала АФ $d(n)$ используется *выходной* сигнал неизвестной системы — аддитивная смесь полезного сигнала $s(n)$ с искаженным шумом $\tilde{x}_{\text{ш}}(n)$:

$$d(n) = s(n) + \tilde{x}_{\text{ш}}(n). \quad (21.55)$$

В процессе адаптации АФ стремится преобразовать входной сигнал $x(n)$ так, чтобы обеспечить наилучшее приближение (по заданному критерию) *выходного* сигнала АФ $y(n)$ к *выходному* сигналу неизвестной системы $d(n)$, формируя сигнал ошибки $e(n)$ (21.2):

$$e(n) = d(n) - y(n) = s(n) + \tilde{x}_{\text{ш}}(n) - y(n), \quad (21.56)$$

или, с учетом (21.1):

$$e(n) = d(n) - y(n) = s(n) + \tilde{x}_{\text{ш}}(n) - \sum_{i=0}^{N-1} h_i x_{\text{ш}}(n-i). \quad (21.57)$$

Выберем в качестве критерия наилучшего приближения выходного сигнала АФ $y(n)$ к образцовому сигналу $d(n)$ (21.55) минимум среднего квадрата сигнала ошибки¹ (21.4) и запишем равенство (21.9) с учетом (21.56):

$$\frac{1}{L} \sum_{n=0}^{L-1} \left[\sum_{i=0}^{N-1} h_i x_{\text{ш}}(n-i) \right] x_{\text{ш}}(n-i) = \frac{1}{L} \sum_{n=0}^{L-1} s(n) x_{\text{ш}}(n-i) + \frac{1}{L} \sum_{n=0}^{L-1} \tilde{x}_{\text{ш}}(n) x_{\text{ш}}(n-i).$$

Поскольку шум $x_{\text{ш}}(n)$ не коррелирован с полезным сигналом $s(n)$, их взаимно корреляционная функция (первое слагаемое в правой части) будет равна нулю. Следовательно, в процессе адаптации обеспечивается наилучшее приближение (по заданному критерию) выходного сигнала АФ $y(n)$ к искаженному шуму $\tilde{x}_{\text{ш}}(n)$, т. е. на выходе АФ получаем *оценку* шума $\hat{\tilde{x}}_{\text{ш}}(n)$:

$$y(n) = \hat{\tilde{x}}_{\text{ш}}(n), \quad (21.58)$$

а сигнал ошибки, согласно (21.55) и (21.57), представляет собой *оценку* полезного сигнала $\hat{s}(n)$:

$$e(n) = d(n) - y(n) = s(n) + \tilde{x}_{\text{ш}}(n) - \hat{\tilde{x}}_{\text{ш}}(n) = \hat{s}(n). \quad (21.59)$$

Моделирование процесса очистки сигнала от шума в MATLAB включает в себя следующие шаги:

1. Моделирование входного сигнала неизвестной системы — входного сигнала АФ в виде шума $x(n) = x_{\text{ш}}(n)$, например, нормального или равномерного белого шума.

¹ Данный критерий соответствует фильтру Винера, но для критериев (21.15) и (21.26) будет получен тот же результат.

2. Моделирование неизвестной системы — КИХ- или БИХ-фильтра, искажающего шум $x_{\text{ш}}(n)$, и вычисление его реакции $\tilde{x}_{\text{ш}}(n)$ с помощью функции `filter`.
3. Моделирование полезного гармонического сигнала $s(n)$.
4. Моделирование выходного сигнала неизвестной системы — образцового сигнала АФ $d(n)$ в виде аддитивной смеси полезного сигнала $s(n)$ с искаженным шумом $\tilde{x}_{\text{ш}}(n)$.
5. Моделирование структуры АФ — объекта `adaptfilt`.
6. Моделирование процесса адаптивной фильтрации — вычисление выходного сигнала АФ $y(n)$ (21.58) и сигнала ошибки $e(n)$ (21.59) с помощью функции `filter`.
7. Вывод графиков полезного сигнала $s(n)$ и его оценки $e(n) = \hat{s}(n)$ (21.59).
8. Вывод графиков амплитудного спектра полезного сигнала $s(n)$, его смеси с шумом $d(n)$ и оценки полезного сигнала $e(n) = \hat{s}(n)$ с помощью функции `fft`.

В начале процесса адаптации в течение некоторого времени наблюдается переходный процесс, во время которого дисперсия сигнала ошибки АФ оказывается много большей, чем по его окончании — в установившемся режиме. Поэтому амплитудный спектр оценки полезного сигнала имеет смысл вычислять в установившемся режиме. На этапе моделирования начальный момент установившегося режима $n_{\text{нач}}$ можно задать приближенно, например, равным 0,05 от длины сигнала ошибки АФ.

9. Сравнение оценки полезного сигнала длины L с истинным полезным сигналом по критерию среднеквадратической ошибки — RMSE¹:

$$\text{RMSE} = \sqrt{\frac{1}{L} \sum_{k=0}^{L-1} |s(n) - \hat{s}(n)|^2}.$$

На практике истинный полезный сигнал $s(n)$ неизвестен, поэтому показатель RMSE используется на этапе *моделирования* для сравнительного анализа оценок полезного сигнала при различных алгоритмах адаптации.

Чем меньше значение RMSE, тем выше качество оценки полезного сигнала.

Подобно амплитудному спектру, вычислять значение RMSE имеет смысл в установившемся режиме.

21.1.4.4. Выравнивание частотной характеристики неизвестной системы

Одним из практических приложений АФ является выравнивание частотной характеристики канала связи¹. В учебных целях мы ограничимся решением более узкой

¹ Root Mean Squared Error. По существу это сравнение на основе нормы $\|x\|_2$ (см. разд. 2.1.5).

задачи — выравниванием частотной характеристики неизвестной системы с ограниченной полосой пропускания при обработке случайного сигнала.

Реальный канал связи во многих случаях описывается моделью в виде КИХ-фильтра с несимметричной ИХ. Для решения поставленной задачи в качестве модели неизвестной системы с ограниченной полосой пропускания предлагается выбрать КИХ-фильтр ФНЧ или ПФ с симметричной ИХ, что не нарушает общности выводов.

Искажения, вносимые неизвестной системой, можно скомпенсировать, подключив к выходу неизвестной системы АФ с тем, чтобы АЧХ *каскадного соединения* неизвестной системы и АФ оказалась более широкополосной, чем АЧХ неизвестной системы. Такую операцию называют *выравниванием частотной характеристики* неизвестной системы².

Задача выравнивания частотной характеристики неизвестной системы сводится к решению задачи *обратной* идентификации, структурная схема которой представлена на рис. 21.4.

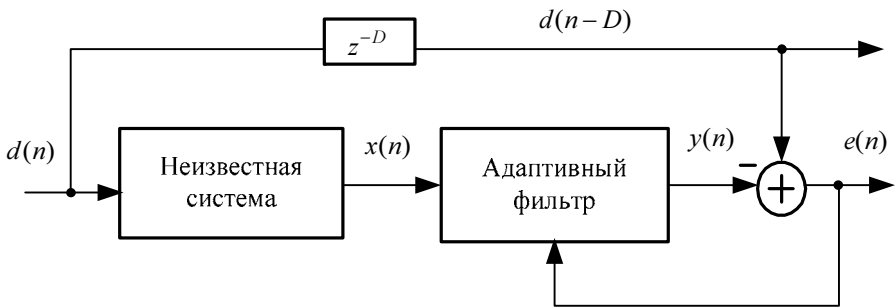


Рис. 21.4. Структурная схема обратной идентификации для выравнивания частотной характеристики неизвестной системы с ограниченной полосой пропускания

При *обратной* идентификации *выходной* сигнал неизвестной системы $x(n)$ совпадает с *входным* сигналом АФ. В качестве образцового сигнала АФ используется *входной* сигнал неизвестной системы $d(n)$, задержанный на D отсчетов — $d(n - D)$, что объясняется следующим. Проходя неизвестную систему и АФ, выходной сигнал $y(n)$ задерживается относительно входного сигнала $d(n)$, поэтому при вычислении сигнала ошибки $e(n)$ (21.2) сигналы, поступающие на сумматор по двум параллельным ветвям, должны быть согласованы по времени. Выбор значения задержки D требует отдельного исследования. Показано [4], что при длине N_1 КИХ-фильтра, существенно меньшей длины N КИХ-фильтра в составе АФ, удовлетворительный результат будет получен при задержке $D = \text{int}(N/2)$ (округление до ближайшего целого в сторону увеличения).

¹ Под каналом связи подразумевается среда распространения сигнала.

² При этом АЧХ неизвестной системы остается неизменной.

В процессе адаптации АФ стремится преобразовать входной сигнал $x(n)$ так, чтобы обеспечить наилучшее приближение (по заданному критерию) *выходного* сигнала АФ $y(n)$ к *входному* сигналу неизвестной системы $d(n-D)$:

$$y(n) = \hat{d}(n-D), \quad (21.60)$$

где $\hat{d}(n-D)$ — оценка задержанного входного сигнала $d(n-D)$.

Передаточная функция $V(z)$ каскадного соединения равна произведению передаточных функций неизвестной системы $G(z)$ и АФ¹ $H(z)$:

$$V(z) = G(z)H(z), \quad (21.61)$$

где $G(z)$ и $H(z)$, по определению (см. разд. 8.1.2), равны отношению z -изображения реакции к z -изображению воздействия при ННУ:

$$G(z) = \frac{X(z)}{D(z)}; \quad (21.62)$$

$$H(z) = \frac{Y(z)}{X(z)}. \quad (21.63)$$

Согласно (21.60), z -изображение выходного сигнала АФ равно

$$Y(z) = \hat{D}(z)z^{-D}. \quad (21.64)$$

Подставляя (21.64) в (21.63) с учетом (21.62), получим передаточную функцию АФ:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\hat{D}(z)z^{-D}}{X(z)} = \frac{z^{-D}}{\hat{G}(z)} \quad (21.65)$$

и передаточную функцию каскадного соединения неизвестной системы и АФ $V(z)$ (21.61) в виде:

$$V(z) = G(z) \frac{z^{-D}}{\hat{G}(z)} = \hat{U}_0(z)z^{-D}, \quad (21.66)$$

где $\hat{U}_0(z)$ — оценка z -изображения цифрового единичного импульса $u_0(n)$ (как известно [1], $U_0(z) = 1$).

Подставляя $z = e^{j\hat{\omega}}$ в (21.66), получим комплексную частотную характеристику каскадного соединения неизвестной системы и АФ:

$$V(e^{j\hat{\omega}}) = \left| \hat{U}_0(e^{j\hat{\omega}}) \right| e^{-j(D+\Delta)\hat{\omega}}, \quad (21.67)$$

где $\left| \hat{U}_0(e^{j\hat{\omega}}) \right| \approx 1$ — выровненная АЧХ (оценка АЧХ каскадного соединения); $(D+\Delta)\hat{\omega}$ — линейная ФЧХ (оценка ФЧХ каскадного соединения);

¹ КИХ-фильтра в составе АФ.

$\Delta = \arg\{\hat{U}_0(e^{j\hat{\omega}})\}$ — слагаемое, учитывающее искажение ФЧХ за счет того, что вместо Фурье-изображения цифрового единичного импульса $U_0(e^{j\hat{\omega}}) = 1$ имеем его оценку $\hat{U}_0(e^{j\hat{\omega}})$.

Во временной области произведению передаточных функций (21.61) соответствует линейная свертка импульсных характеристик неизвестной системы $g(n)$ и АФ $h(n)$:

$$v(n) = g(n) * h(n), \quad (21.68)$$

которая, согласно (21.66), представляет собой *оценку* задержанного цифрового единичного импульса $\hat{u}_0(n - D)$:

$$v(n) = \hat{u}_0(n - D). \quad (21.69)$$

На практике образцовый сигнал $d(n)$ недоступен для наблюдения (при известном сигнале задача теряет смысл). Существуют различные способы формирования образцового сигнала. Например [3], при выравнивании частотной характеристики канала связи, по которому передается ограниченный и известный приемной стороне набор сигналов (символов), первоначально в качестве образцового сигнала генерируется специальный настроечный сигнал. После выравнивания частотной характеристики реальный сигнал поступает на вход АФ, а в качестве образцового выбирается сигнал (символ), ближайший к реальному по заданному критерию. Применение АФ позволяет компенсировать искажение реального сигнала при его передаче по каналу связи.

Моделирование выравнивания частотной характеристики неизвестной системы в MATLAB в предположении, что образцовый сигнал известен, включает в себя следующие шаги:

1. Моделирование входного сигнала неизвестной системы — образцового сигнала АФ $d(n)$.
2. Моделирование задержанного образцового сигнала $d(n - D)$ посредством обнуления $D = \text{int}(N/2)$ начальных значений сигнала $d(n)$.
3. Моделирование неизвестной системы — КИХ-фильтра ФНЧ (или ПФ) порядка $R = (N_1 - 1)$, где N_1 — длина КИХ-фильтра.
4. Вычисление выходного сигнала неизвестной системы — входного сигнала АФ $x(n)$ с помощью функции `filter`.
5. Моделирование структуры АФ с КИХ-фильтром длины N ($N \gg N_1$) в виде объекта `adaptfilt`.
6. Моделирование процесса адаптивной фильтрации — вычисление выходного сигнала АФ $y(n) = \hat{d}(n - D)$ и сигнала ошибки $e(n)$ с помощью функции `filter`.

7. Определение параметров АФ — ИХ КИХ-фильтра:

`H.coefficients`

где `H` — имя объекта `adaptfilt`.

8. Вычисление ИХ $v(n)$ (21.68) каскадного соединения неизвестной системы и АФ.

9. Вывод графиков ИХ, АЧХ и ФЧХ неизвестной системы, АФ и их каскадного соединения.

Отметим, что при моделировании выравнивания АЧХ КИХ-фильтра ФНЧ или ПФ в АЧХ каскадного соединения будут наблюдаться скачки на частотах, соответствующих нулям АЧХ КИХ-фильтра в полосе задерживания. При этом ИХ каскадного соединения остается близкой к цифровому единичному импульсу.

21.2.4.5. Оценка параметров линейного предсказания сигнала

Линейным предсказанием сигнала¹ называют *оценку* его текущего отсчета $\hat{x}(n)$ на основе линейной комбинации предшествующих отсчетов.

Ранее (см. разд. 17.1.2) линейное предсказание (17.8) рассматривалось при определении оценок АР-модели методом Юла—Уолкера.

Задача линейного предсказания заключается в определении *параметров линейного предсказания* a_k , $k = 1, 2, \dots, (M-1)$, обеспечивающих наилучшее приближение оценки текущего отсчета $\hat{x}(n)$ к его истинному значению $x(n)$ по заданному критерию, в качестве которого выбирается *минимум среднего квадрата ошибки линейного предсказания* $\varepsilon(n)$ (17.14).

Параметры линейного предсказания a_k определяются в результате решения СЛАУ (17.15):

$$\frac{1}{L} \sum_{n=0}^{L-1} \left[\sum_{k=1}^{M-1} a_k x(n-k) \right] x(n-k) = -\frac{1}{L} \sum_{n=0}^{L-1} x(n)x(n-k) \quad (21.70)$$

и представляют собой *оценки параметров АР-модели* $\hat{a}_k$ (см. разд. 17.1.2).

Сравним СЛАУ (21.69) со СЛАУ (21.9) в фильтре Винера:

$$\frac{1}{L} \sum_{n=0}^{L-1} \left[\sum_{i=0}^{N-1} h_i x(n-i) \right] x(n-i) = \frac{1}{L} \sum_{n=0}^{L-1} d(n)x(n-i). \quad (21.71)$$

Выберем в качестве *образцового* сигнала АФ $d(n)$ сигнал $x(n)$, для которого требуется определить параметры линейного предсказания:

$$d(n) = x(n), \quad (21.72)$$

¹ По умолчанию речь идет о линейном предсказании вперед (forward linear prediction).

а в качестве входного сигнала АФ — задержанный сигнал $x(n-1)$, и с учетом этого запишем СЛАУ (21.71):

$$\frac{1}{L} \sum_{n=0}^{L-1} \left[\sum_{i=0}^{N-1} h_i x[n-(i+1)] \right] x[n-(i+1)] = \frac{1}{L} \sum_{n=0}^{L-1} x(n)x[n-(i+1)]. \quad (21.73)$$

Выполнив замену переменных $k = (i+1)$, получим:

$$\frac{1}{L} \sum_{n=0}^{L-1} \left[\sum_{k=1}^N h_{k-1} x(n-k) \right] x(n-k) = \frac{1}{L} \sum_{n=0}^{L-1} x(n)x(n-k), \quad (21.74)$$

где выражение в квадратных скобках, согласно (21.1), соответствует выходному сигналу АФ $y(n-1)$.

Сравним СЛАУ (21.74) с (21.70). При совпадении (с точностью до знака) правых частей имеем совпадение левых частей, следовательно, для параметров АФ h_i и параметров линейного предсказания a_k при $(N-1) \geq (M-1)$ справедливо соотношение:

$$\begin{cases} a_k = -h_i, & k=1, 2, \dots, (M-1); \quad i = (k-1) = 0, 1, \dots, (M-2); \\ h_i = 0, & (M-1) \leq i \leq (N-1). \end{cases} \quad (21.75)$$

Для случайного сигнала $x(n)$, статистическая модель которого заранее неизвестна, используют один из алгоритмов рекуррентного вычисления оценок параметров АФ — оценок параметров линейного предсказания.

Структурная схема оценки параметров линейного предсказания сигнала представлена на рис. 21.5.

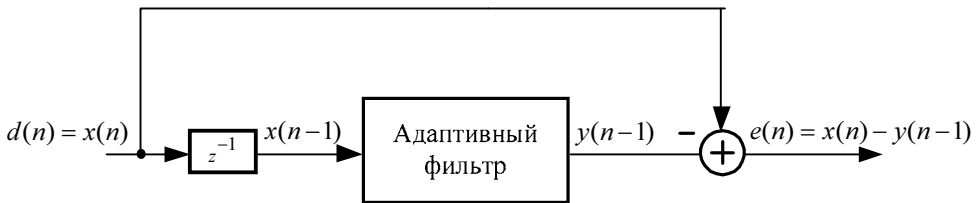


Рис. 21.5. Структурная схема оценки параметров линейного предсказания сигнала

Вычисление оценок параметров АР-модели и оценок параметров линейного предсказания (с целью их сравнения) в MATLAB включает в себя следующие шаги:

1. Моделирование входного сигнала АР-модели — нормального белого шума $e_{AR}(n)$.
2. Задание известных параметров АР-модели порядка $p = (M-1)$ (известных параметров линейного предсказания) — вектора $\mathbf{a}$ с учетом $a_0 = 1$.
3. Вычисление выходного (моделируемого) сигнала АР-модели $y_{AR}(n)$.

4. Вычисление оценок параметров АР-модели $\hat{\mathbf{a}}$ методом Юла—Уолкера в предположении, что анализируемый случайный сигнал $x(n)$ совпадает с выходным сигналом АР-модели $y_{AR}(n)$: $x(n) = y_{AR}(n)$.
5. Моделирование структуры АФ в виде объекта `adaptfilt`.
6. Порядок КИХ-фильтра $(N - 1)$ выбирается из условия $(N - 1) \geq (M - 1)$.
7. Моделирование образцового сигнала АФ $d(n)$, совпадающего с анализируемым сигналом $x(n)$: $d(n) = x(n)$.
8. Моделирование входного сигнала АФ — задержанного сигнала $x(n - 1)$ посредством обнуления первого отсчета сигнала $x(n)$.
9. Моделирование процесса адаптивной фильтрации — вычисление выходного сигнала АФ $y(n - 1)$ и сигнала ошибки $e(n)$ с помощью функции `filter`.
10. Определение параметров АФ с противоположным знаком — оценок параметров линейного предсказания:
`-H.coefficients`
 где `H` — имя объекта `adaptfilt`.
11. Вывод графиков векторов: заданных параметров АР-модели, их оценок и оценок параметров линейного предсказания на интервале, равном длине КИХ-фильтра в составе АФ.
12. Сравнение полученных оценок по критерию средней абсолютной ошибки — MAE^1 — на одинаковой длине N КИХ-фильтра в составе АФ:

$$\text{MAE} = \frac{1}{N} \sum_{k=0}^{N-1} |\alpha_k - \hat{\alpha}_k|,$$

где $\hat{\alpha}_k$ — оценка параметра; α_k — истинное значение параметра; для АР-модели без учета $\alpha_0 = 1$ с дополнением $N - (M - 1)$ нулями до длины N .

21.2. Содержание лабораторной работы

Содержание работы связано с моделированием систем адаптивной фильтрации программными средствами MATLAB.

21.3. Задание на лабораторную работу

Лабораторная работа выполняется на основе script-файла с именем `lr_21`, который хранится на прилагаемом компакт-диске в папке `LAB_DSP\LAB_21`.

Перед выполнением работы необходимо сохранить путь к папке `LAB_21` по команде контекстного меню **Add to Path | Selected Folders**.

¹ Mean Absolute Value. По существу это сравнение на основе нормы $\|\mathbf{x}\|_1$ (см. разд. 2.1.5).

Исходные данные для пунктов задания приводятся в табл. 21.1 для номера бригады $N_{бр}$, где $N_{бр} = 1, 2, \dots, 30$. Функция $N_{бр} \bmod M$ в записи исходных данных означает вычисление значения $N_{бр}$ по модулю M .

На прилагаемом компакт-диске в папке Tables\Tables_21 хранятся табл. 21.1 исходных данных и пример ее заполнения для $N_{бр} = 1$.

Таблица 21.1. Таблица исходных данных

Переменная	Назначение	Значение			Идентификатор
$N_{бр}$	Номер бригады	$N_{бр}$			Nb =
N	Длина КИХ-фильтра в составе АФ	$N = N_{бр} \bmod 10 + 41$			N =
L	Длина входного сигнала АФ	$L = 1000(N_{бр} \bmod 2 + 3)$			L =
f_d	Частота дискретизации	$f_d = 2000(N_{бр} \bmod 5 + 1)$			Fs =
A_1	Амплитуды дискретных гармоник	$A_1 = 0,03 + 0,001N_{бр}$			A1 =
A_2		$A_2 = 0,5A_1$			A2 =
f_1	Частоты дискретных гармоник	$f_1 = f_d/8$			f1 =
f_2		$f_2 = f_d/4$			f2 =
a	Заданные (истинные) параметры АР-модели	Номера бригад $N_{бр}$			Вектор a = [1 ...]
		1—10	11—20	21—30	
a_0		1	1	1	
a_1		−0,86	−0,10	0,50	
a_2		0,54	0,20	0,17	
a_3		−0,30	0,20	0,30	
a_4		−0,17	0,10	0,10	
a_5		0,22	0,30	0,10	
a_6		−0,10	—	−0,10	
a_7		—	—	−0,50	

Задание на лабораторную работу заключается в моделировании систем адаптивной фильтрации и включает в себя следующие пункты:

1. Моделирование нормального белого шума.

Создать модель нормального белого шума $r(n)$ (идентификатор `r_gauss`) длины L с нулевым средним значением и единичной дисперсией, которая будет использована в дальнейших исследованиях.

2. Моделирование структуры АФ с алгоритмом LMS.

Создать модель структуры АФ с алгоритмом LMS в виде объекта `adaptfilt.lms` с именем `hlms` и вывести его свойства.

Задать входные параметры объекта `adaptfilt.lms`:

- длину КИХ-фильтра N ;
- шаг адаптации μ (идентификатор `mu`), равный половине максимального шага адаптации (идентификатор `mu_max`) в (21.22).

При вычислении шага адаптации μ в качестве входного сигнала АФ $x(n)$ (идентификатор `x`) использовать нормальный белый шум $r(n)$ (см. п. 1):

$$x(n) = r(n);$$

- остальные параметры — по умолчанию.

Пояснить:

- какой из входных параметров объекта `adaptfilt.lms` соответствует шагу адаптации и как он определяется;
- свойства объекта `adaptfilt.lms`.

3. Моделирование структуры АФ с алгоритмом NLMS.

Создать модель структуры АФ с алгоритмом NLMS в виде объекта `adaptfilt.nlms` с именем `hnlms` и вывести его свойства.

Задать входные параметры объекта `adaptfilt.nlms`:

- длину КИХ-фильтра N ;
- константу $\varepsilon = 10^{-6}$ (идентификатор `epsilon`) в (21.25);
- остальные параметры — по умолчанию.

Пояснить:

- какой из входных параметров объекта `adaptfilt.nlms` соответствует константе ε и что она определяет;
- свойства объекта `adaptfilt.nlms`.

4. Моделирование структуры АФ с алгоритмом RLS.

Создать модель структуры АФ с алгоритмом RLS в виде объекта `adaptfilt.rls` с именем `hrls` и вывести его свойства.

Задать входные параметры объекта `adaptfilt.rls`:

- длину КИХ-фильтра N ;
- остальные параметры — по умолчанию.

Пояснить:

- какой из входных параметров объекта `adaptfilt.rls` соответствует коэффициенту забывания λ в (21.53)—(21.54) и с какой целью он вводится;
- свойства объекта `adaptfilt.rls`.

5. Оценка импульсной характеристики неизвестной ЛДС.

Вычисление оценки ИХ неизвестной ЛДС включает в себя следующие шаги:

- моделирование входного сигнала неизвестной ЛДС — входного сигнала АФ $x(n)$ (идентификатор `x`) в виде нормального белого шума $r(n)$ (см. п. 1):

$$x(n) = r(n);$$

- моделирование неизвестной ЛДС — КИХ-фильтра ФНЧ (см. разд. 11.2.5) порядка $R_1 = \text{int}(N/2)$ (идентификатор `R1`) с нормированной частотой разрыва $\hat{f}_c = 0,5$ (идентификатор `wc`);
- вычисление выходного сигнала неизвестной ЛДС — образцового сигнала АФ $d(n)$ (идентификатор `d`) с помощью функции `filter`;
- вычисление истинной ИХ неизвестной ЛДС $h(n)$ (идентификатор `h`);
- моделирование структур АФ — объектов `adaptfilt` с именами `Hlms` (см. п. 2) и `Hrls` (см. п. 3);
- вычисление выходного сигнала $y(n)$ и сигнала ошибки $e(n)$ АФ с именем `Hlms` (идентификаторы `y_lms` и `e_lms`) и АФ с именем `Hrls` (идентификаторы `y_rls` и `e_rls`) с помощью функции `filter`;
- определение параметров АФ с именами `Hlms` и `Hrls` (идентификаторы `h_lms` и `h_rls`) — оценок ИХ неизвестной ЛДС $\hat{h}(n)$.

Вывести графики:

- сигналов ошибки АФ — объектов с именами `Hlms` и `Hrls`;
- истинной ИХ и ее оценок на одинаковом интервале времени $n \in [0; (N-1)]$.

Сравнить оценки ИХ с истинной ИХ по критерию среднего абсолютного отклонения их отсчетов на основе нормы $\|\mathbf{x}\|_1$ (идентификаторы `norm1_lms` и `norm1_rls`).

Повторить процедуру для другой неизвестной ЛДС — БИХ-фильтра ФВЧ Баттерворта (см. разд. 13.1.5) порядка $R_1 = \text{int}(N/2)$ (идентификатор `R1`) с нормированной частотой среза $\hat{f}_0 = 0,3$ (идентификатор `WDn`).

Пояснить:

- на основе какой системы решается задача оценки ИХ неизвестной ЛДС;
- чему в АФ соответствует оценка ИХ неизвестной ЛДС;
- на каком интервале дискретного нормированного времени определяются оценки ИХ неизвестной ЛДС;
- какой из алгоритмов (LMS или RLS) обеспечивает более точную оценку ИХ по заданному критерию;
- какой из неизвестных ЛДС (КИХ или БИХ) соответствует сигнал ошибки АФ с большей дисперсией.

6. Очистка сигнала от шума.

Моделирование процесса очистки сигнала от шума включает в себя следующие шаги:

- моделирование входного сигнала неизвестной ЛДС — входного сигнала АФ $x(n) = x_{\text{ш}}(n)$ (идентификатор `x`);

В качестве шума $x_{\text{ш}}(n)$ (идентификатор `x_noise`) выбрать нормальный белый шум $r(n)$ (см. п. 1):

$$x_{\text{ш}}(n) = r(n);$$

- моделирование неизвестной ЛДС — КИХ-фильтра ФНЧ (см. п. 5), искажающего шум $x_{\text{ш}}(n)$, и вычисление его реакции $\tilde{x}_{\text{ш}}(n)$ (идентификатор `x_noiseNEW`) с помощью функции `filter`;
- моделирование полезного сигнала $s(n)$ (идентификатор `s`) в виде периодической последовательности (суммы двух гармоник) с периодом L :

$$s(n) = A_1 \cos(2\pi f_1 nT) + A_2 \cos(2\pi f_2 nT), \quad (21.76)$$

используя ее тождественное представление в виде:

$$s(n) = A_1 \cos\left(\frac{2\pi f_1}{f_d} n\right) + A_2 \cos\left(\frac{2\pi f_2}{f_d} n\right) = A_1 \cos(\hat{\omega}_1 n) + A_2 \cos(\hat{\omega}_2 n); \quad (21.77)$$

- моделирование выходного сигнала неизвестной ЛДС — образцового сигнала АФ $d(n)$ (идентификатор `d`) в виде аддитивной смеси сигнала $s(n)$ с искаженным шумом $\tilde{x}_{\text{ш}}(n)$:

$$d(n) = s(n) + \tilde{x}_{\text{ш}}(n); \quad (21.78)$$

- моделирование структур АФ — объектов `adaptfilt` с именами `Hlms` (см. п. 2) и `Hrls` (см. п. 3);
- вычисление выходного сигнала $y(n)$ и сигнала ошибки $e(n)$ АФ с именем `Hlms` (идентификаторы `y_lms` и `e_lms`) и АФ с именем `Hrls` (идентификаторы `y_rls` и `e_rls`) с помощью функции `filter`.

Вывести графики:

- полезного сигнала $s(n)$ и его аддитивной смеси с шумом $d(n)$;
- сигналов ошибки АФ (оценок полезного сигнала) — объектов с именами `Hlms` и `Hrls`;
- следующих амплитудных спектров на периоде $[0; f_d]$:
 - полезного сигнала (идентификатор `MOD_s`);
 - его аддитивной смеси с шумом (идентификатор `MOD_d`);
 - оценок полезного сигнала в установившемся режиме с использованием алгоритмов LMS и RLS (идентификаторы `MOD_lms` и `MOD_rls`).

Начальный момент установившегося режима $n_{\text{нач}}$ (идентификатор `n_start`) задать равным $0,05L$.

Вычислить значения RMSE (см. разд. 21.1.4.3) для оценок полезного сигнала с использованием алгоритмов LMS и RLS (идентификаторы `RMSE_lms` и `RMSE_rls`) в установившемся режиме на интервале дискретного нормированного времени $n \in [n_{\text{нач}}; (L-1)]$.

Пояснить:

- на основе какой системы решается задача очистки сигнала от шума;
- какому сигналу АФ соответствует оценка полезного сигнала;
- как по графику сигнала ошибки АФ оценить длительность переходного процесса в АФ;
- какой из алгоритмов (LMS или RLS) обеспечивает лучшую очистку сигнала от шума (по результатам визуального сравнения амплитудных спектров и значениям RMSE).

7. Выравнивание частотной характеристики неизвестной ЛДС.

Моделирование выравнивания частотной характеристики неизвестной ЛДС включает в себя следующие шаги:

- моделирование входного сигнала неизвестной ЛДС — образцового сигнала АФ $d(n)$ (идентификатор `d`);

В качестве входного сигнала неизвестной ЛДС $d(n)$ выбрать аддитивную смесь сигнала $s(n)$ (21.77) (идентификатор `s`) с нормальным белым шумом $r(n)$ (см. п. 1) с СКО, равным $3 \max\{A_1, A_2\}$:

$$d(n) = s(n) + 3 \max\{A_1, A_2\} r(n). \quad (21.79)$$

Для сигнала $s(n)$ оставить неизменной частоту f_1 , а частоту f_2 задать равной $f_2 = 3f_1$.

- моделирование задержанного образцового сигнала $d(n-D)$ (идентификатор `d_delay`) посредством обнуления $D = \text{int}(N/2)$ (идентификатор `D`) начальных значений сигнала $d(n)$;
- моделирование неизвестной ЛДС — КИХ-фильтра ФНЧ (см. разд. 11.1.5) порядка $R_2 = \text{int}(N/7)$ (идентификатор `R2`) с нормированной частотой разрыва $\hat{f}_c = 0,5$ (идентификатор `wc`);
- моделирование структуры АФ — объекта `adaptfilt` с именем `Hrls` (см. п. 3);
- вычисление выходного сигнала неизвестной ЛДС — входного сигнала АФ $x(n)$ (идентификатор `x`) с помощью функции `filter`;
- вычисление выходного сигнала АФ $y(n) = \hat{d}(n-D)$ и сигнала ошибки $e(n)$ (идентификаторы `y_rls` и `e_rls`) с помощью функции `filter`;
- определение параметров АФ — ИХ $h(n)$ его КИХ-фильтра (идентификатор `h_rls`);
- вычисление ИХ $v(n) = g(n) * h(n)$ (идентификатор `h_conv`) каскадного соединения неизвестной ЛДС и АФ $h(n)$.

Вывести графики:

- входного сигнала неизвестной ЛДС и его амплитудного спектра (идентификатор `MODS`) на периоде $[0; f_d]$;
- выходного сигнала неизвестной ЛДС и его амплитудного спектра (идентификатор `MODX`) на периоде $[0; f_d]$;
- импульсных характеристик неизвестной ЛДС, КИХ-фильтра в составе АФ и каскадного соединения неизвестной ЛДС и АФ;
- АЧХ неизвестной ЛДС (идентификатор `MAG_US`), КИХ-фильтра в составе АФ (идентификатор `MAG_AF`) и их каскадного соединения (идентификатор `MAG`) в основной полосе частот $[0; f_d/2]$;
- ФЧХ неизвестной ЛДС (идентификатор `PH_US`), КИХ-фильтра в составе АФ (идентификатор `PH_AF`) и их каскадного соединения (идентификатор `PH`) в основной полосе частот $[0; f_d/2]$;
- выходного сигнала АФ и его амплитудного спектра на периоде $[0; f_d]$.

Пояснить:

- чему равны частоты дискретных гармоник входного сигнала неизвестной ЛДС;
- на основе какой системы решается задача выравнивания частотной характеристики неизвестной ЛДС;

- вид амплитудного спектра входного сигнала неизвестной ЛДС;
 - какому сигналу АФ соответствует входной сигнал неизвестной ЛДС;
 - чему равна задержка образцового сигнала АФ $d(n - D)$;
 - вид амплитудного спектра выходного сигнала неизвестной ЛДС и причину его изменения относительно амплитудного спектра входного сигнала;
 - вид импульсных характеристик неизвестной ЛДС, КИХ-фильтра в составе АФ и их каскадного соединения;
 - какому виду ИХ соответствует выровненная АЧХ;
 - вид ФЧХ неизвестной ЛДС, КИХ-фильтра в составе АФ и их каскадного соединения;
 - результат сравнения амплитудных спектров выходного сигнала неизвестной ЛДС и выходного сигнала АФ.
8. Вычисление оценок параметров АР-модели и оценок параметров линейного предсказания.

Вычисление оценок параметров АР-модели и оценок параметров линейного предсказания включает в себя следующие шаги:

- моделирование входного сигнала АР-модели — нормального белого шума $e_{AR}(n)$ (идентификатор e_AR).

В качестве входного сигнала АР-модели выбрать нормальный белый шум $r(n)$ (см. п. 1):

$$e_{AR}(n) = r(n); \quad (21.80)$$

- задание вектора $\mathbf{a}$ известных параметров АР-модели (известных параметров линейного предсказания).

В MATLAB параметры АР-модели — элементы вектора $\mathbf{a}$ — вводятся с добавлением элемента $a_0 = 1$ (см. табл. 21.1);

- вычисление выходного (моделируемого) сигнала АР-модели $y_{AR}(n)$ (идентификатор y_AR);
- вычисление оценок параметров АР-модели $\hat{\mathbf{a}}$ методом Юла—Уолкера (идентификатор a_{AR}) в предположении, что анализируемый сигнал $x(n)$ (идентификатор x) совпадает с моделируемым сигналом $y_{AR}(n)$:

$$x(n) = y_{AR}(n); \quad (21.81)$$

- моделирование структуры АФ — объекта `adaptfilt` с именем `hrls` (см. п. 3);
- моделирование образцового сигнала АФ $d(n)$ (идентификатор `d`), совпадающего с анализируемым сигналом $x(n)$:

$$d(n) = x(n); \quad (21.82)$$

- моделирование входного сигнала АФ — задержанного сигнала $x(n-1)$ (идентификатор `x_delay`) посредством обнуления первого отсчета сигнала $x(n)$;
- вычисление выходного сигнала АФ $y(n-1)$ и сигнала ошибки $e(n)$ (идентификаторы `y_rls` и `e_rls`) с помощью функции `filter`;
- определение параметров АФ с противоположным знаком — вектора оценок параметров линейного предсказания (идентификатор `h_rls`).

Вывести графики на интервале дискретного нормированного времени $n \in [0; (N-1)]$:

- вектора заданных параметров АР-модели (без учета $a_0 = 1$);
- вектора оценок параметров АР-модели;
- вектора оценок параметров линейного предсказания.

Для наглядности вывести графики векторов в одинаковом диапазоне по оси ординат $[-\text{MAX } \text{MAX}]$ с помощью функции `ylim`, где `MAX` равно максимальному по модулю элементу среди всех векторов.

Вычислить значения MAE оценок параметров АР-модели и линейного предсказания (идентификаторы `MAE_AR` и `MAE_LP`) на одинаковой длине N КИХ-фильтра в составе АФ (см. разд. 21.1.4.5).

Пояснить:

- связь параметров АФ с оценками параметров линейного предсказания;
- связь параметров АР-модели с параметрами линейного предсказания;
- связь между порядками АР-модели и КИХ-фильтра в составе АФ;
- какая из оценок параметров (АР-модели или линейного предсказания) определена с большей точностью по критерию MAE.

21.4. Типовой script-файл для выполнения лабораторной работы

Перед выполнением работы должна быть представлена табл. 21.1 исходных данных для своего номера бригады $N_{\text{бр}}$.

Для *запуска* лабораторной работы необходимо обратиться к script-файлу `lr_21` по его имени:

```
>> lr_21
```

Для *принудительного снятия* script-файла с выполнения следует нажать комбинацию клавиш `<Ctrl>+<Break>`.

При выполнении script-файла текущие окна с графиками *не закрывать*.

Листинг script-файла `lr_21` имеет вид:

```
>> type lr_21
clc
clear
disp('% ЛР №21. АДАПТИВНЫЕ ФИЛЬТРЫ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
Nb = input('Nb = ');           % НОМЕР БРИГАДЫ
N = input('N = ');            % ДЛИНА КИХ-ФИЛЬТРА В СОСТАВЕ АФ
L = input('L = ');            % ДЛИНА ВХОДНОГО СИГНАЛА АФ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.1. МОДЕЛИРОВАНИЕ НОРМАЛЬНОГО БЕЛОГО ШУМА')
n = 0:(L-1);                  %ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
r_gauss = randn(1,L);         % НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ
disp('%')
disp('%')
disp('% Созданная модель НОРМАЛЬНОГО БЕЛОГО ШУМА будет использована
в дальнейших исследованиях')
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.2. МОДЕЛИРОВАНИЕ СТРУКТУРЫ АФ С АЛГОРИТМОМ LMS')
x = r_gauss;                  % ВХОДНОЙ СИГНАЛ АФ
Px = var(x);                  % СРЕДНИЙ КВАДРАТ ВХОДНОГО СИГНАЛА АФ
mu_max = 2/(N*Px);            % МАКСИМАЛЬНЫЙ ШАГ АДАПТАЦИИ
mu = 0.5*mu_max;              % ЗАДАННЫЙ ШАГ АДАПТАЦИИ
Hlms = adaptfilt.lms(N,mu)    % СТРУКТУРА АФ С АЛГОРИТМОМ LMS
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
```

```

disp('%')
disp('%')
disp('% п.3. МОДЕЛИРОВАНИЕ СТРУКТУРЫ АФ С АЛГОРИТМОМ NLMS')
epsilon = 1e-6; % КОНСТАНТА, ОПРЕДЕЛЯЮЩАЯ МАКСИМАЛЬНЫЙ ШАГ АДАПТАЦИИ
Hnlms = adaptfilt.nlms(N,1,1,epsilon) % СТРУКТУРА АФ С АЛГОРИТМОМ NLMS
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.4. МОДЕЛИРОВАНИЕ СТРУКТУРЫ АФ С АЛГОРИТМОМ RLS')
Hrls = adaptfilt.rls(N) % СТРУКТУРА АФ С АЛГОРИТМОМ RLS
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.5. ОЦЕНКА ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ НЕИЗВЕСТНОЙ ЛДС')
disp('%')
disp('%')
disp('% 5.1. ОЦЕНКА ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ НЕИЗВЕСТНОЙ ЛДС – КИХ-ФИЛЬТРА ФНЧ  
(FIR)')
disp('%')
disp('%')
x = r_gauss; % ВХОДНОЙ СИГНАЛ НЕИЗВЕСТНОЙ ЛДС
R1 = round(N/2); % ПОРЯДОК КИХ-ФИЛЬТРА ФНЧ
wc = 0.5; % НОРМИРОВАННАЯ ЧАСТОТА РАЗРЫВА КИХ-ФИЛЬТРА ФНЧ
b = fir1(R1,wc); % КОЭФФИЦИЕНТЫ КИХ-ФИЛЬТРА ФНЧ
d = filter(b,1,x); % ВЫХОДНОЙ СИГНАЛ НЕИЗВЕСТНОЙ ЛДС (FIR)
h = b; % ИСТИННАЯ ИМПУЛЬСНАЯ ХАРАКТЕРИСТИКА НЕИЗВЕСТНОЙ ЛДС (FIR)
дЛИНЫ (R1+1)
[y_lms,e_lms] = filter(Hlms,x,d); % ВЫХОДНОЙ СИГНАЛ И СИГНАЛ ОШИБКИ АФ С
АЛГОРИТМОМ LMS
[y_rls,e_rls] = filter(Hrls,x,d); % ВЫХОДНОЙ СИГНАЛ И СИГНАЛ ОШИБКИ АФ С
АЛГОРИТМОМ RLS
h_lms = Hlms.coefficients; % ОЦЕНКА ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ
НЕИЗВЕСТНОЙ ЛДС (FIR) – ПАРАМЕТРЫ АФ С АЛГОРИТМОМ LMS
h_rls = Hrls.coefficients; % ОЦЕНКА ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ
НЕИЗВЕСТНОЙ ЛДС (FIR) – ПАРАМЕТРЫ АФ С АЛГОРИТМОМ RLS
disp('% Для вывода ГРАФИКОВ СИГНАЛА ОШИБКИ АФ нажмите <ENTER>')
pause
n = 0:length(x)-1; % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ СИГНАЛА
ОШИБКИ АФ
figure('Name','Error signal of AF for LMS and RLS','NumberTitle', 'off')
subplot(2,1,1), plot(n,e_lms), grid, xlabel('n'), title('Error signal for LMS')
subplot(2,1,2), plot(n,e_rls), grid, xlabel('n'), title('Error signal for RLS')
n1 = 0:N-1; % ИНТЕРВАЛ ДИСКРЕТНОГО НОРМИРОВАННОГО ВРЕМЕНИ ДЛЯ ОЦЕНОК
ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ

```

```

if length(h)<N
    h = [h zeros(1,(N-length(h)))]; % ИСТИННАЯ ИМПУЛЬСНАЯ ХАРАКТЕРИСТИКА
    НЕИЗВЕСТНОЙ ЛДС (FIR), ДОПОЛНЕННАЯ НУЛЯМИ ДО ДЛИНЫ N
end
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИСТИННОЙ ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ (FIR) и ее
ОЦЕНОК нажмите <ENTER>')
pause
figure('Name','True Impulse response FIR and its Estimates','NumberTitle',
'off')
subplot(3,1,1), stem(n1,h), grid, xlabel('n1')
title('True Impulse response FIR – h(n) with length N')
subplot(3,1,2), stem(n1,h_lms), grid, xlabel('n1')
title('LMS Impulse response FIR – h_lms')
subplot(3,1,3), stem(n1,h_rls), grid, xlabel('n1')
title('RLS Impulse response FIR – h_rls')
disp('%')
disp('%')
disp('% Для вывода СРЕДНЕГО АБСОЛЮТНОГО ОТКЛОНЕНИЯ')
disp('% ОТСЧЕТОВ ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ от ее ОЦЕНОК (FIR) нажмите <ENTER>')
pause
disp('%')
disp(['          norml_lms = ',num2str((1/N)*norm(h-h_lms,1))])
disp(['          norml_rls = ',num2str((1/N)*norm(h-h_rls,1))])
disp('%')
disp('%')
disp('% 5.2. ОЦЕНКА ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ НЕИЗВЕСТНОЙ ЛДС – БИХ-ФИЛЬТРА ФВЧ
(IIR)')
disp('%')
disp('%')
R1 = round(N/2); % ПОРЯДОК БИХ-ФИЛЬТРА ФВЧ
WDn = 0.3; % НОРМИРОВАННАЯ ЧАСТОТА СРЕЗА БИХ-ФИЛЬТРА ФВЧ
[b,a] = butter(R1,WDn,'high'); % КОЭФФИЦИЕНТЫ БИХ-ФИЛЬТРА ФВЧ
d = filter(b,a,x); % ВЫХОДНОЙ СИГНАЛ НЕИЗВЕСТНОЙ ЛДС (IIR)
h = impz(b,a,N); % ИСТИННАЯ ИМПУЛЬСНАЯ ХАРАКТЕРИСТИКА НЕИЗВЕСТНОЙ ЛДС
(IIR) ДЛИНЫ N (ВЕКТОР-СТОЛБЕЦ)
[y_lms,e_lms] = filter(Hlms,x,d); % ВЫХОДНОЙ СИГНАЛ И СИГНАЛ ОШИБКИ АФ
С АЛГОРИТМОМ LMS
[y_rls,e_rls] = filter(Hrls,x,d); % ВЫХОДНОЙ СИГНАЛ И СИГНАЛ ОШИБКИ АФ
С АЛГОРИТМОМ RLS
h_lms = Hlms.coefficients; % ПАРАМЕТРЫ АФ С АЛГОРИТМОМ LMS
h_rls = Hrls.coefficients; % ПАРАМЕТРЫ АФ С АЛГОРИТМОМ RLS
disp('% Для вывода ГРАФИКОВ СИГНАЛА ОШИБКИ АФ нажмите <ENTER>')
pause
figure('Name','Error signal of AF for LMS and RLS','NumberTitle', 'off')
subplot(2,1,1), plot(n,e_lms), grid, xlabel('n'), title('Error signal for LMS')
subplot(2,1,2), plot(n,e_rls), grid, xlabel('n'), title('Error signal for RLS')

```

```

disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИСТИННОЙ ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ (IIR) и ее
ОЦЕНОК нажмите <ENTER>')
pause
figure('Name','True Impulse response IIR and its Estimates','NumberTitle',
'off')
subplot(3,1,1), stem(n1,h), grid, xlabel('n1')
title('True Impulse response IIR – h(n) with length N')
subplot(3,1,2), stem(n1,h_lms), grid, xlabel('n1')
title('LMS Impulse response IIR – h_lms(n)')
subplot(3,1,3), stem(n1,h_rls), grid, xlabel('n1')
title('RLS Impulse response IIR – h_rls(n)')
disp('%')
disp('%')
disp('% Для вывода СРЕДНЕГО АБСОЛЮТНОГО ОТКЛОНЕНИЯ')
disp('% ОТСЧЕТОВ ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ от ее ОЦЕНОК (IIR) нажмите <ENTER>')
disp('%')
disp(['          norm1_lms = ',num2str((1/N)*norm(h'-h_lms,1))])
disp(['          norm1_rls = ',num2str((1/N)*norm(h'-h_rls,1))])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.6. ОЧИСТКА СИГНАЛА ОТ ШУМА')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
Fs = input('Fs = ');          % ЧАСТОТА ДИСКРЕТИЗАЦИИ (Гц)
A1 = input('A1 = ');          % АМПЛИТУДЫ ДИСКРЕТНЫХ ГАРМОНИК
A2 = input('A2 = ');
f1 = input('f1 = ');          % ЧАСТОТЫ (Гц) ДИСКРЕТНЫХ ГАРМОНИК
f2 = input('f2 = ');
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
x_noise = r_gauss;          % ВХОДНОЙ СИГНАЛ АФ – НОРМАЛЬНЫЙ БЕЛЫЙ ШУМ
b = fir1(R1,wc);          % КОЭФФИЦИЕНТЫ КИХ-ФИЛЬТРА ФНЧ, ИСКАЖАЮЩЕГО ШУМ
x_noiseNEW = filter(b,1,x_noise); % ИСКАЖЕННЫЙ ШУМ НА ВЫХОДЕ КИХ-ФИЛЬТРА ФНЧ
w1 = 2*pi*f1/Fs; w2 = 2*pi*f2/Fs; % НОРМИРОВАННЫЕ ЧАСТОТЫ ДИСКРЕТНЫХ ГАРМОНИК
(РАД)

```

```

n = 0:(L-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
s = A1*cos(w1*n)+A2*cos(w2*n); % ПОЛЕЗНЫЙ СИГНАЛ
d = s+x_noiseNEW; % ОБРАЗЦОВЫЙ СИГНАЛ АФ
[y_lms,e_lms] = filter(Hlms,x_noise,d); % ВЫХОДНОЙ СИГНАЛ И СИГНАЛ ОШИБКИ АФ
C АЛГОРИТМОМ LMS
[y_rls,e_rls] = filter(Hrls,x_noise,d); % ВЫХОДНОЙ СИГНАЛ И СИГНАЛ ОШИБКИ АФ
C АЛГОРИТМОМ RLS
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ПОЛЕЗНОГО СИГНАЛА, ЕГО АДДИТИВНОЙ СМЕСИ С ШУМОМ')
disp('% и ОЦЕНОК ПОЛЕЗНОГО СИГНАЛА (LMS) И (RLS) нажмите <ENTER>')
pause
figure('Name','Harmonic Signal, Mixture of Signal and Noise, Estimates of
Harmonic Signal (LMS and RLS)','NumberTitle','off')
subplot(4,1,1), plot(n,s), grid, title('Harmonic Signal – s(n)')
subplot(4,1,2), plot(n,d), grid, title('Mixture of Harmonic Signal and Noise –
d(n)')
subplot(4,1,3), plot(n,e_lms), grid, title('Estimate Harmonic Signal (LMS)')
subplot(4,1,4), plot(n,e_rls), grid, xlabel('n'), title('Estimate Harmonic
Signal (RLS)')
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АМПЛИТУДНЫХ СПЕКТРОВ ПОЛЕЗНОГО СИГНАЛА, ЕГО
АДДИТИВНОЙ СМЕСИ С ШУМОМ')
disp('% и ОЦЕНОК ПОЛЕЗНОГО СИГНАЛА (LMS) И (RLS)')
disp('% в УСТАНОВИВШЕМСЯ РЕЖИМЕ нажмите <ENTER>')
pause
k = 0:(L-1); % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
f = (k/L)*Fs; % ЧАСТОТА (Гц)
MOD_s = (2/L)*abs(fft(s)); % АМПЛИТУДНЫЙ СПЕКТР ПОЛЕЗНОГО СИГНАЛА
MOD_s(1) = (1/L)*abs(fft(s(1)));
MOD_d = (2/L)*abs(fft(d)); % АМПЛИТУДНЫЙ СПЕКТР СМЕСИ ПОЛЕЗНОГО СИГНАЛА
C ШУМОМ
MOD_d(1) = (1/L)*abs(fft(d(1)));
n_start = round(0.05*L); % НАЧАЛО УСТАНОВИВШЕГОСЯ РЕЖИМА
Ls = (L- n_start)+1; % ДЛИНА ОЦЕНОК СИГНАЛА И ШУМА В УСТАНОВИВШЕМСЯ
РЕЖИМЕ
e1_lms = e_lms(n_start:end); % ОЦЕНКА ПОЛЕЗНОГО СИГНАЛА В УСТАНОВИВШЕМСЯ
РЕЖИМЕ (LMS)
e1_rls = e_rls(n_start:end); % ОЦЕНКА ПОЛЕЗНОГО СИГНАЛА В УСТАНОВИВШЕМСЯ
РЕЖИМЕ (RLS)
ks = 0:(Ls-1); % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА ДЛЯ
УСТАНОВИВШЕГОСЯ РЕЖИМА
fs = (ks/Ls)*Fs; % ЧАСТОТА (Гц)
MOD_lms = (2/Ls)*abs(fft(e1_lms)); % АМПЛИТУДНЫЙ СПЕКТР ОЦЕНКИ ПОЛЕЗНОГО
СИГНАЛА В УСТАНОВИВШЕМСЯ РЕЖИМЕ (LMS)
MOD_lms(1) = (1/Ls)*abs(fft(e1_lms(1)));
MOD_rls = (2/Ls)*abs(fft(e1_rls)); % АМПЛИТУДНЫЙ СПЕКТР ОЦЕНКИ ПОЛЕЗНОГО
СИГНАЛА В УСТАНОВИВШЕМСЯ РЕЖИМЕ (RLS)

```

```

MOD_rls(1) = (1/Ls)*abs(fft(e1_rls(1)));
figure('Name','Amplitude Spectrums of Harmonic Signal, Mixture of Signal and
Noise, and Estimates of Harmonic Signal (LMS and RLS)','NumberTitle','off')
subplot(4,1,1), stem(f,MOD_s), grid, title('Amplitude Spectrum of Harmonic
Signal')
subplot(4,1,2), stem(f,MOD_d), grid, title('Amplitude Spectrum of Harmonic
Signal and Noise - d(n)')
subplot(4,1,3), stem(fs,MOD_lms), grid, title('Amplitude Spectrum of Estimate
Harmonic Signal (LMS)')
subplot(4,1,4), stem(fs,MOD_rls), grid, xlabel('f'), title('Amplitude Spectrum
of Estimate Harmonic (RLS)')

ns = 0:(Ls-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ
УСТАНОВИВШЕГОСЯ РЕЖИМА
s1 = s(n_start:end); % ПОЛЕЗНЫЙ СИГНАЛ НА ИНТЕРВАЛЕ ВРЕМЕНИ ДЛЯ ЕГО
ОЦЕНОК
RMSE_lms = sqrt((1/Ls).*sum((s1-e1_lms).^2)); % RMSE ДЛЯ ОЦЕНКИ ПОЛЕЗНОГО
СИГНАЛА (LMS)
RMSE_rls = sqrt((1/Ls).*sum((s1-e1_rls).^2)); % RMSE ДЛЯ ОЦЕНКИ ПОЛЕЗНОГО
СИГНАЛА (RLS)
disp('%')
disp('%')
disp('% Для вывода ЗНАЧЕНИЙ RMSE нажмите <ENTER>')
pause
disp('%')
disp(['          RMSE_lms =      ',num2str(RMSE_lms)])
disp(['          RMSE_rls =      ',num2str(RMSE_rls)])
disp('%')
disp('%')
disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.7. ВЫРАВНИВАНИЕ ЧАСТОТНОЙ ХАРАКТЕРИСТИКИ НЕИЗВЕСТНОЙ ЛДС')
f2 = 3*f1; % ИЗМЕНЕННАЯ ЧАСТОТА (Гц) ГАРМОНИКИ
w2 = 2*pi*(3*f1)/Fs; % НОРМИРОВАННАЯ ДИСКРЕТНАЯ ЧАСТОТА (РАД)
n = 0:(L-1); % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ
s = A1*cos(w1*n)+A2*cos(w2*n)+3*max(A1,A2)*r_gauss; % ВХОДНОЙ СИГНАЛ
НЕИЗВЕСТНОЙ ЛДС - ОБРАЗЦОВЫЙ СИГНАЛ АФ
k = 0:(L-1); % ДИСКРЕТНАЯ НОРМИРОВАННАЯ ЧАСТОТА
f = k*(Fs/L); % ЧАСТОТА (Гц)
S = fft(s); % ДПФ ВХОДНОГО СИГНАЛА НЕИЗВЕСТНОЙ ЛДС
MODS = (2/L)*abs(S); % АМПЛИТУДНЫЙ СПЕКТР ВХОДНОГО СИГНАЛА НЕИЗВЕСТНОЙ
ЛДС
MODS(1) = (1/L)*abs(S(1));
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ВХОДНОГО СИГНАЛА НЕИЗВЕСТНОЙ ЛДС и его АМПЛИТУДНОГО
СПЕКТРА нажмите <ENTER>')
pause

```

```

figure('Name','Input Signal of Unknown LDS and its Amplitude Spectrum', 'NumberTitle', 'off')
subplot(2,1,1), plot(n,s), grid, xlabel('n'), title('Input Signal of Unknown LDS')
subplot(2,1,2), stem(f,MODS), grid, xlabel('f'), title('Amplitude Spectrum')
d = s; % ОБРАЗЦОВЫЙ СИГНАЛ АФ – ВХОДНОЙ СИГНАЛ НЕИЗВЕСТНОЙ ЛДС
D = floor(N/2); % ВЕЛИЧИНА ЗАДЕРЖКИ ОБРАЗЦОВОГО СИГНАЛА
D_zeros = zeros(1,round(N/2)); % ФОРМИРОВАНИЕ НАЧАЛЬНЫХ НУЛЕВЫХ ЗНАЧЕНИЙ ЗАДЕРЖАННОГО ОБРАЗЦОВОГО СИГНАЛА
d_delay = [D_zeros d(1:(length(d)-length(D_zeros)))]; % ЗАДЕРЖАННЫЙ ОБРАЗЦОВЫЙ СИГНАЛ
R2 = round(N/7); % ПОРЯДОК КИХ-ФИЛЬТРА ФНЧ
wc = 0.5; % НОРМИРОВАННАЯ ЧАСТОТА РАЗРЫВА КИХ-ФИЛЬТРА ФНЧ
b = fir1(R2,wc); % ИМПУЛЬСНАЯ ХАРАКТЕРИСТИКА НЕИЗВЕСТНОЙ ЛДС
x = filter(b,1,d); % ВЫХОДНОЙ СИГНАЛ НЕИЗВЕСТНОЙ ЛДС – ВХОДНОЙ СИГНАЛ АФ
X = fft(x); % ДПФ ВЫХОДНОГО СИГНАЛА НЕИЗВЕСТНОЙ ЛДС
MODX = (2/L)*abs(X); % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА НЕИЗВЕСТНОЙ ЛДС
MODX(1) = (1/L)*abs(X(1));
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ВЫХОДНОГО СИГНАЛА НЕИЗВЕСТНОЙ ЛДС и его АМПЛИТУДНОГО СПЕКТРА нажмите <ENTER>')
pause
figure('Name','Output Signal of Unknown LDS and its Amplitude Spectrum', 'NumberTitle', 'off')
subplot(2,1,1), plot(n,x), grid, xlabel('n'), title('Output Signal of Unknown LDS')
subplot(2,1,2), stem(f,MODX), grid, xlabel('f'), title('Amplitude Spectrum')
Hrls = adaptfilt.rls(N); % СТРУКТУРА АФ С АЛГОРИТМОМ RLS
[y_rls,e_rls] = filter(Hrls,x,d_delay); % ВЫХОДНОЙ СИГНАЛ И СИГНАЛ ОШИБКИ АФ С АЛГОРИТМОМ RLS
h_rls = Hrls.coefficients; % ПАРАМЕТРЫ АФ (ИМПУЛЬСНАЯ ХАРАКТЕРИСТИКА ЕГО КИХ-ФИЛЬТРА)
h_conv = conv(h_rls,b); % ИМПУЛЬСНАЯ ХАРАКТЕРИСТИКА КАСКАДНОГО СОЕДИНЕНИЯ НЕИЗВЕСТНОЙ ЛДС И АФ
n2 = 0:length(b)-1; % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ НЕИЗВЕСТНОЙ ЛДС (КИХ-ФИЛЬТРА ФНЧ)
n3 = 0:length(h_conv)-1; % ДИСКРЕТНОЕ НОРМИРОВАННОЕ ВРЕМЯ ДЛЯ СВЕРТКИ ИМПУЛЬСНОЙ ХАРАКТЕРИСТИКИ НЕИЗВЕСТНОЙ ЛДС И АФ
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ИМПУЛЬСНЫХ ХАРАКТЕРИСТИК НЕИЗВЕСТНОЙ ЛДС, КИХ-ФИЛЬТРА В СОСТАВЕ АФ')
disp('% и ИХ КАСКАДНОГО СОЕДИНЕНИЯ нажмите <ENTER>')
pause
figure('Name','Impulse Response of AF, Unknown LDS, and Cascade Connection', 'NumberTitle', 'off')
subplot(3,1,1), stem(n2,b), grid, title('Impulse Response of Unknown LDS')
subplot(3,1,2), stem(n1,h_rls), grid, title('Impulse Response of AF')

```

```

subplot(3,1,3), stem(n3,h_conv), grid, xlabel('n'), title('Impulse Response of
Cascade Connection')
fa = 0:Fs/500:(Fs/2); % ВЕКТОР ЧАСТОТ ДЛЯ АЧХ
MAG_US = abs(freqz(b,1,fa,Fs)); % АЧХ НЕИЗВЕСТНОЙ ЛДС
MAG_AF = abs(freqz(h_rls,1,fa,Fs)); % АЧХ КИХ-ФИЛЬТРА В СОСТАВЕ АФ
MAG = MAG_US.*MAG_AF; % АЧХ КАСКАДНОГО СОЕДИНЕНИЯ НЕИЗВЕСТНОЙ
ЛДС И АФ
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ АЧХ НЕИЗВЕСТНОЙ ЛДС, КИХ-ФИЛЬТРА В СОСТАВЕ АФ,')
disp('% и ИХ КАСКАДНОГО СОЕДИНЕНИЯ нажмите <ENTER>')
pause
figure('Name','Magnitude Response of Unknown LDS, AF, and Cascade
Connection','NumberTitle','off')
subplot(3,1,1), plot(fa,MAG_US), grid, title('Magnitude Response of Unknown
LDS')
subplot(3,1,2), plot(fa,MAG_AF), grid, title('Magnitude Response of AF')
subplot(3,1,3), plot(fa,MAG), grid, xlabel('f'), title('Magnitude Response of
Cascade Connection')
PH_US = angle(freqz(b,1,fa,Fs)); % ФЧХ НЕИЗВЕСТНОЙ ЛДС
PH_AF = angle(freqz(h_rls,1,fa,Fs)); % ФЧХ КИХ-ФИЛЬТРА В СОСТАВЕ АФ
PH = PH_US+PH_AF; % ФЧХ КАСКАДНОГО СОЕДИНЕНИЯ НЕИЗВЕСТНОЙ
ЛДС И АФ
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ФЧХ НЕИЗВЕСТНОЙ ЛДС, КИХ-ФИЛЬТРА В СОСТАВЕ АФ')
disp('% и ИХ КАСКАДНОГО СОЕДИНЕНИЯ нажмите <ENTER>')
pause
figure('Name','Phase Response of Unknown LDS, AF, and Cascade
Connection','NumberTitle','off')
subplot(3,1,1), plot(fa,PH_US), grid, title('Phase Response of Unknown LDS')
subplot(3,1,2), plot(fa,PH_AF), grid, title('Phase Response of AF')
subplot(3,1,3), plot(fa,PH), grid, xlabel('f'), title('Phase Response of
Cascade Connection')
Y = fft(y_rls); % ДПФ ВЫХОДНОГО СИГНАЛА АФ
MODY = (2/L)*abs(Y); % АМПЛИТУДНЫЙ СПЕКТР ВЫХОДНОГО СИГНАЛА АФ
MODY(1) = (1/L)*abs(Y(1));
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ВЫХОДНОГО СИГНАЛА АФ и его АМПЛИТУДНОГО СПЕКТРА
нажмите <ENTER>')
pause
figure('Name','Output Signal and Amplitude Spectrum of AF','NumberTitle',
'off')
subplot(2,1,1), plot(n,y_rls), grid, xlabel('n'), title('Output Signal of AF')
subplot(2,1,2), stem(f,MODY), grid, xlabel('f'), title('Amplitude Spectrum')
disp('%')
disp('%')

```



```

disp('% Для продолжения нажмите <ENTER>')
pause
disp('%')
disp('%')
disp('% п.8. ВЫЧИСЛЕНИЕ ОЦЕНОК ПАРАМЕТРОВ АР-МОДЕЛИ И ОЦЕНОК ПАРАМЕТРОВ  
ЛИНЕЙНОГО ПРЕДСКАЗАНИЯ')
disp('%')
disp('%')
disp('% Введите ИСХОДНЫЕ ДАННЫЕ')
DATA=0;
while DATA==0
a = input('a = '); % ВЕКТОР ПАРАМЕТРОВ АР-МОДЕЛИ
disp('% Проверьте ПРАВИЛЬНОСТЬ ввода ИСХОДНЫХ ДАННЫХ')
disp('% При ПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 1')
disp('% При НЕПРАВИЛЬНЫХ ИСХОДНЫХ ДАННЫХ введите 0 и ПОВТОРИТЕ ввод')
DATA = input('--> ');
end
e_AR = r_gauss; % ВХОДНОЙ СИГНАЛ АР-МОДЕЛИ
y_AR = filter(1,a,e_AR); % ВЫХОДНОЙ СИГНАЛ АР-МОДЕЛИ
x = y_AR; % АНАЛИЗИРУЕМЫЙ СИГНАЛ ПОЛАГАЕТСЯ РАВНЫМ  
МОДЕЛИРУЕМОМУ СИГНАЛУ
p = length(a)-1; % ПОРЯДОК АР-МОДЕЛИ
[ayW DYW] = aryule(x,p); % ОЦЕНКА ПАРАМЕТРОВ АР-МОДЕЛИ ПО МЕТОДУ  
ЮЛА-УОЛКЕРА
d = x; % ОБРАЗЦОВЫЙ СИГНАЛ АФ
x_delay = [0 x(1:(L-1))]; % ВХОДНОЙ СИГНАЛ АФ – ЗАДЕРЖАННЫЙ ОБРАЗЦОВЫЙ  
СИГНАЛ
[y_rls,e_rls] = filter(Hrls,x_delay,d); % ВЫХОДНОЙ СИГНАЛ И СИГНАЛ ОШИБКИ АФ  
С АЛГОРИТМОМ RLS
h_rls = -Hrls.coefficients; % ВЕКТОР ОЦЕНОК ПАРАМЕТРОВ ЛИНЕЙНОГО  
ПРЕДСКАЗАНИЯ
disp('%')
disp('%')
disp('% Для вывода ГРАФИКОВ ВЕКТОРА ЗАДАННЫХ ПАРАМЕТРОВ АР-МОДЕЛИ, их ОЦЕНОК')
disp('% и ВЕКТОРА ОЦЕНОК ПАРАМЕТРОВ ЛИНЕЙНОГО ПРЕДСКАЗАНИЯ нажмите <ENTER>')
pause
figure('Name','True AR parameters, Estimated AR parameters, and Estimated  
Linear Prediction parameters','NumberTitle','off')
YMAX = max([max(abs(a)) max(abs(ayW)) max(abs(h_rls))]); % МАКСИМАЛЬНЫЙ ПО  
МОДУЛЮ ЭЛЕМЕНТ ВЕКТОРОВ
subplot(3,1,1), stem(a(2:end)), grid, xlim([0 N-1]), ylim([-YMAX YMAX])
title('True parameters - a')
subplot(3,1,2), stem(ayW(2:end)), grid, xlim([0 N-1]), ylim([-YMAX YMAX])
title('Estimated parameters AR - ayW')
subplot(3,1,3), stem(h_rls), grid, xlim([0 N-1]), ylim([-YMAX YMAX])
title('Estimated Linear Prediction parameters - h_rls')
disp('%')
disp('%')
disp('% Для вывода ЗНАЧЕНИЙ MAE нажмите <ENTER>')

```

```

pause
a_N = [a(2:end) zeros(1, (N-length(a(2:end))))]; % ВЕКТОР ИСТИННЫХ
ПАРАМЕТРОВ АР-МОДЕЛИ, ДОПОЛНЕННЫЙ НУЛЯМИ ДО ДЛИНЫ N
aYW_N = [aYW(2:end) zeros(1, (N-length(aYW(2:end))))]; % ВЕКТОР ОЦЕНОК
ПАРАМЕТРОВ АР-МОДЕЛИ, ДОПОЛНЕННЫЙ НУЛЯМИ ДО ДЛИНЫ N
MAE_AR = (1/N).*sum(abs(a_N-aYW_N)); % MAE ОЦЕНОК ПАРАМЕТРОВ АР-МОДЕЛИ
MAE_LP = (1/N).*sum(abs(h_rls-a_N)); % MAE ОЦЕНОК ПАРАМЕТРОВ ЛИНЕЙНОГО
ПРЕДСКАЗАНИЯ
disp('%')
disp(['          MAE_AR =          ', num2str(MAE_AR)])
disp(['          MAE_LP =          ', num2str(MAE_LP)])
disp('%')
disp('%')
disp('% РАБОТА ЗАВЕРШЕНА')

```

21.5. Задание на самостоятельную работу

Задание на самостоятельную работу заключается в создании function-файлов для моделирования систем адаптивной фильтрации.

Проверить выполнение созданных function-файлов в режиме прямых вычислений с исходными данными для своего номера бригады $N_{бр}$, а также для произвольных длин входного сигнала АФ L из диапазона $[4000; 7000]$ и КИХ-фильтра в составе АФ N из диапазона $[40; 60]$ (входные параметры *всех* function-файлов).

Самостоятельное задание включает в себя следующие пункты:

1С. Оценка импульсной характеристики неизвестной ЛДС.

Выполнить аналогично п. 5 задания в *разд. 21.3* для БИХ-фильтра ФНЧ при другом случайном входном сигнале и структуре АФ, а именно:

- входной сигнал неизвестной ЛДС — равномерный белый шум;
- структура АФ — с алгоритмом NLMS.

Вывести графики сигнала ошибки АФ, истинной ИХ неизвестной ЛДС и ее оценки.

Вычислить среднее абсолютное отклонение отсчетов оценки ИХ от их истинных значений (выходной параметр function-файла).

2С. Очистка сигнала от шума.

Выполнить аналогично п. 6 задания в *разд. 21.3* для другого входного шума $x_{ш}(n)$ и полезного сигнала $s(n)$, а именно:

- входной шум $x_{ш}(n)$ — нормальный белый шум с СКО σ (дополнительный входной параметр function-файла);
- полезный сигнал $s(n)$ — периодическая последовательность (сумма трех гармоник) с периодом L , амплитудами $A_1 = 0,03\sigma$, $A_2 = 0,02\sigma$, $A_3 = 0,015\sigma$ и частотами $f_1 = 0,08f_d$, $f_2 = 0,17f_d$, $f_3 = 0,24f_d$.

Выбрать структуру АФ с алгоритмом RLS.

Вывести графики:

- полезного сигнала, его аддитивной смеси с искаженным шумом и оценки полезного сигнала в установившемся режиме;
- амплитудных спектров данных сигналов.

Вычислить значение RMSE (выходной параметр function-файла).

3С. Выравнивание частотной характеристики неизвестной ЛДС.

Выполнить аналогично п. 7 задания в *разд. 21.3* для другого сигнала $s(n)$ и шума в аддитивной смеси (21.78), а именно:

- сигнал $s(n)$ — периодическая последовательность (сумма трех гармоник) с периодом L , амплитудами $A_1 = 0,03$, $A_2 = 0,02$, $A_3 = 0,015$ и нормированными частотами $\hat{\omega}_1 = \pi/4$, $\hat{\omega}_2 = 3\pi/4$, $\hat{\omega}_3 = 5\pi/4$;
- шум — нормальный белый шум с СКО, равным $4 \max\{A_1, A_2, A_3\}$.

Вывести графики:

- входного и выходного сигналов неизвестной ЛДС и их амплитудных спектров в основной полосе частот $[0; \pi]$;
- импульсных характеристик неизвестной ЛДС, КИХ-фильтра в составе АФ и каскадного соединения неизвестной ЛДС и АФ;
- АЧХ неизвестной ЛДС, КИХ-фильтра в составе АФ и их каскадного соединения в основной полосе частот $[0; \pi]$;
- выходного сигнала АФ и его амплитудного спектра в основной полосе частот $[0; \pi]$.

4С. Вычисление оценок параметров АР-модели и оценок параметров линейного предсказания.

Выполнить аналогично п. 8 задания в *разд. 21.5* для заданных параметров АР-модели — вектора $\mathbf{a}$ (с учетом $a_0 = 1$):

$$\mathbf{a} = [1 \ -0.20 \ 0.20 \ -0.40 \ 0.10 \ -0.30 \ 0.20 \ -0.20].$$

Для расчета оценок параметров АР-модели выбрать метод Берга.

Вывести графики вектора заданных параметров АР-модели, вектора оценок параметров АР-модели и вектора оценок параметров линейного предсказания.

Вычислить значения МАЕ оценок параметров АР-модели и линейного предсказания (выходные параметры function-файла).

21.6. Отчет и контрольные вопросы

Отчет составляется в редакторе MS Word и содержит исходные данные и результаты выполнения каждого пункта задания, включая полученные графики (копируются по команде **Edit | Copy Figure** в окне **Figure**) и ответы на поставленные вопросы (шрифт Times New Roman).

Защита лабораторной работы проводится на основании представленного отчета и контрольных вопросов из следующего списка:

1. В каких случаях применяют традиционные ЦФ?
2. В каких случаях применяют АФ?
3. Дайте определение АФ.
4. Какие части включает в себя АФ?
5. Что такое параметры АФ?
6. Назовите входные и выходные сигналы АФ.
7. Что понимают под линейностью АФ?
8. К какому типу систем относится АФ (линейная, нелинейная)?
9. Что понимают под обратной связью в АФ?
10. Что понимают под устойчивостью АФ?
11. Какой АФ называют фильтром Винера?
12. Какая оптимизационная задача решается при вычислении параметров АФ в фильтре Винера?
13. Чем ограничено практическое применение фильтра Винера?
14. С какой целью разработаны алгоритмы LMS и RLS?
15. Какая оптимизационная задача решается при вычислении оценок параметров АФ в алгоритме LMS?
16. Приведите рекуррентную формулу для вычисления оценок параметров АФ в алгоритме LMS.
17. Приведите рекуррентную формулу для вычисления оценок параметров АФ в алгоритме NLMS.
18. При каком условии обеспечивается сходимость в среднем квадрате в алгоритме LMS?
19. Назовите преимущества и недостатки алгоритма LMS.
20. Какая оптимизационная задача решается при вычислении параметров АФ в алгоритме RLS?
21. Приведите рекуррентную формулу для вычисления оценок параметров АФ в алгоритме RLS.
22. Назовите преимущества и недостатки алгоритма RLS.

23. Поясните структурную схему оценки импульсной характеристики с помощью АФ.
24. Поясните структурную схему очистки сигнала от шума с помощью АФ.
25. Что понимают под выравниванием частотной характеристики?
26. Поясните структурную схему выравнивания частотной характеристики неизвестной ЛДС с помощью АФ.
27. Поясните структурную схему оценки параметров линейного предсказания с помощью АФ.

21.7. Литература

1. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов. — 2-е изд. — СПб.: БХВ-Петербург, 2005. — Глава 25.
2. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в MATLAB. — СПб.: БХВ-Петербург, 2008. — Глава 15.
3. Сергиенко А. Б. Цифровая обработка сигналов. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Глава 9.
4. Уидроу В., Стирнз С. Адаптивная обработка сигналов. — М.: Радио и связь, 1989.
5. Айфичер Э., Джервис Б. Цифровая обработка сигналов. — М., СПб., Киев: Вильямс, 2004.
6. Адаптивные фильтры / Под ред. К. Ф. Коузэна и П. М. Гранта. — М.: Мир, 1988.

Список сокращений на английском языке

ACF — Autocorrelation Function (автокорреляционная функция).

AF — Adaptive Filter (адаптивный фильтр).

DFT — Discrete Fourier Transform (дискретное преобразование Фурье).

EXT — Extension (расширение).

FFT — Fast Fourier Transform (быстрое преобразование Фурье).

FIR — Finite Impulse Response (конечная импульсная характеристика).

GUI — Graphical User Interface (графический интерфейс пользователя).

HTML — Hyper Text Markup Language (язык гипертекстовой маркировки).

IDFT — Inverse Discrete Fourier Transform (обратное дискретное преобразование Фурье).

IFFT — Inverse Fast Fourier Transform (обратное быстрое преобразование Фурье).

IIR — Infinite Impulse Response (бесконечная импульсная характеристика).

LMS — Least Mean Square (метод наименьших квадратов).

LSB — Least Significant Bits (младшее слово).

MSB — Most Significant Bits (старшее слово).

NLMS — Normalize Least Mean-Square (нормализованный метод наименьших квадратов).

PDF — Portable Document Format (формат переносного документа).

PSD — Power Spectral Density (спектральная плотность мощности).

RLS — Recursive Least Square (рекурсивный метод наименьших квадратов).

RMSE — Root Mean Squared Error (среднеквадратическая ошибка).

STD — Standard Deviation (стандартное отклонение).

Список сокращений на русском языке

АКФ — автокорреляционная функция.

АФ — адаптивный фильтр.

АФП — аналоговый фильтр-прототип.

АЦП — аналого-цифровой преобразователь.

АЧХ — амплитудно-частотная характеристика.

БИХ — бесконечная импульсная характеристика (тип фильтра).

БПФ — быстрое преобразование Фурье.

ВКФ — взаимная корреляционная функция.

ДПФ — дискретное преобразование Фурье.

ИХ — импульсная характеристика.

КИХ — конечная импульсная характеристика (тип фильтра).

ЛДС — линейная дискретная система.

ЛФЧХ — линейная ФЧХ.

МНК — метод наименьших квадратов.

НМНК — нормализованный метод наименьших квадратов.

ННУ — нулевые начальные условия.

ОБПФ — обратное быстрое преобразование Фурье.

ОДПФ — обратное дискретное преобразование Фурье.

ОДУ — обыкновенные дифференциальные уравнения.

ПЗ — полоса задерживания.

ПП — полоса пропускания.

ПТ — плавающая точка.

ПФ — полосовой фильтр.

РНК — рекуррентный метод наименьших квадратов.

РУ — разностное уравнение.

РФ — режекторный фильтр.

СКО — среднеквадратическое (стандартное) отклонение.

СЛАУ — система линейных алгебраических уравнений.

СПМ — спектральная плотность мощности.

ФВЧ — фильтр верхних частот.

ФНЧ — фильтр нижних частот.

ФТ — фиксированная точка.

ФЧХ — фазочастотная характеристика.

ЦД — цифровой дифференциатор.

ЦОС — цифровая обработка сигналов.

ЦПГ — цифровой преобразователь Гильберта.

ЦПОС — цифровой процессор обработки сигналов.

ЦФ — цифровой фильтр.

Предметный указатель

А

Адаптивный фильтр:

- ◇ алгоритм:
 - LMS 463
 - NLMS 465
 - RLS 467
- ◇ Винера 460
- ◇ линейность 460
- ◇ параметры 458
 - оптимальные 462
- ◇ структура 458
- ◇ сходимость в среднем 464
 - квадрате 464
- ◇ устойчивость 460
- ◇ шаг адаптации 464

АКФ 96

- ◇ ДПФ 336
- ◇ оценка:
 - несмещенная 335
 - смещенная 335
- ◇ требуемая 337

Ансамбль реализаций 98

АЧХ 126

Б

Белый шум:

- ◇ нормальный 100
- ◇ равномерный 100

БИХ-фильтр 249

- ◇ Баттерворта 252
- ◇ Золотарева—Кауэра 252
- ◇ масштабирование 256
- ◇ оптимальный 249
- ◇ расстановка звеньев 256
- ◇ требования к АЧХ 249
- ◇ Чебышева:
 - I рода 252
 - II рода 252
- ◇ эллиптический 252

В

Вектор 20

- ◇ ввод 21
- ◇ длина 35

- ◇ норма 42
 - ◇ регулярная сетка 34
- Выражение:
- ◇ арифметическое 24
 - ◇ логическое 24

Г

Гистограмма 62

Графики:

- ◇ двумерные 61
- ◇ оформление 61
- ◇ свойства 61, 65
- ◇ трехмерные 64

Д

ДПФ 143

- ◇ разрешение по частоте 143, 147

И

Идентификация системы:

- ◇ обратная 473
- ◇ прямая 472

Импульсная характеристика:

- ◇ КИХ ЛДС 122
- ◇ оценка 474

К

Карта нулей и полюсов 123

КИХ-фильтр:

- ◇ оптимальный 219
- ◇ с линейной ФЧХ 188
- ◇ точки альтернанса 220
- ◇ четыре типа 189

Команда 17

- ◇ clc 17
- ◇ clear 17
- ◇ colorbar 65, 340
- ◇ fdatool 281
- ◇ format 17, 19
- ◇ fvtool 202, 231, 261
- ◇ grid 61
- ◇ help 17

- ◇ hold on 60
- ◇ load 17, 26
- ◇ pause 76
- ◇ return 77
- ◇ save 17, 26
- ◇ sptool 383
- ◇ type 75
- ◇ ver 17
- ◇ what 17
- ◇ which 17
- ◇ who 17
- ◇ whos 17
- ◇ wintool 326

Константы 18

- ◇ логические 20
- ◇ символьные 20
- ◇ стандартные 19
- ◇ численные 18

Л

ЛДС 120

- ◇ БИХ 122
- ◇ КИХ 122
- ◇ нерекурсивная 122
- ◇ порядок 123
- ◇ рекурсивная 122
- ◇ устойчивость 123

Линейное предсказание 362

- ◇ ошибка 362
- ◇ параметры 362

М

Массив 20

- ◇ двумерный 20
- ◇ записей 53
- ◇ логический 51
- ◇ одномерный 20
- ◇ определение типа 56
- ◇ размер 21
- ◇ размерность 20
- ◇ символьный 51
- ◇ структура 53
- ◇ тип 21, 50
- ◇ трехмерный 20
- ◇ числовой 51
- ◇ ячеек 55

Матрица 20, 34

- ◇ ввод 21
- ◇ возведение в степень 39
- ◇ выделение:
 - векторов 36
 - подматриц 36

- ◇ деление 41
- ◇ диагональная 36
- ◇ единиц 36
- ◇ единичная 36
- ◇ конкатенация 37
- ◇ копирование 37
- ◇ норма 42
- ◇ нулевая 36
- ◇ обращение 41
- ◇ пустая 35
- ◇ размер 34, 35
- ◇ сложение и вычитание 38
- ◇ случайных чисел 36
- ◇ Теплица 36, 364
- ◇ тип 34
- ◇ транспонирование 39
- ◇ умножение 38
- ◇ эрмитово сопряжение 40

Метод:

- ◇ билинейного Z-преобразования 252
- ◇ окон 195
- ◇ оценки параметров АР-модели:
 - автокорреляционный 361
 - Берга 365
 - ковариационный 365
 - модифицированный ковариационный 365
 - Юла—Уолкера 364
- ◇ оценки СПМ:
 - Берга 366
 - Блэкмана—Тьюки 334
 - ковариационный 366
 - модифицированный ковариационный 366
 - периодограмм 325
 - периодограмм Бартлетта 332
 - периодограмм Даньелла 330
 - периодограмм Уэлча 333
 - Юла—Уолкера 366
- ◇ чебышевской аппроксимации 219

Н

ННУ 120

О

Объект

- ◇ adaptfilt 465
- ◇ dfilt 127, 193
- ◇ fdesign 223, 255
- ◇ mfilt 413
- ◇ spectrum 327, 334, 366

Оператор 17

- ◇ break 87
- ◇ for 85
- ◇ if 83

Оператор (*prod.*)

◇ switch 85

◇ while 87

Операции с матрицами 38

П

Параметры

◇ AP-модели 359, 481

◇ APCC-модели 359

◇ ЛДС 121

◇ линейного предсказания 481

◇ окон 339

◇ СС-модели 359

Передаточная функция

◇ БИХ-фильтра 249

◇ в виде произведения 124

◇ в виде суммы дробей 124

◇ КИХ-фильтра 188

◇ нерекурсивной ЛДС 123

◇ нули и полюсы 123

◇ рекурсивной ЛДС 123

Переменные 20

Периодограмма

◇ Бартлетта 332

◇ Даньелла 331

◇ модифицированная 326

◇ немодифицированная 325

◇ Уэлча 333

Плотность спектральная 146, 167

◇ мощности 325

Полоса частот, основная 126

Последовательность 95

◇ анализируемая 361, 362

◇ квантованная 95

◇ моделируемая 359

◇ случайная 97

Р

Равенство Парсеваля 169

Режим:

◇ командный 16

◇ программирования 72

◇ прямых вычислений 16

С

Свертка:

◇ круговая 168

◇ линейная 168

◇ секционированная 169

Свойство:

◇ аддитивности 120

◇ инвариантности во времени 99, 120

Сигнал:

◇ детерминированный 96

◇ дискретный 95

◇ нестационарный 99

◇ случайный 97

◇ среднее значение 96

◇ средняя мощность 96

◇ стационарный 98

◇ цифровой 95

◇ энергия 96

◇ эргодический 99

Система:

◇ децимации 406

◇ интерполяции 402

◇ многоскоростная 401

◇ передискретизации 409

Скаляр 21

Сохранение:

◇ М-файла 77

◇ данных на диске 26

◇ пути к папке 16

Спектр 144

◇ амплитудный 144

◇ растекание 166

◇ фазовый 144

Спектрограмма 339

СПМ 325

◇ RSME оценки 368

◇ добротность оценки 329

◇ несмещенность оценки 328

◇ оценка 325

◇ СКО оценки 330

◇ состоятельность оценки 328

Структура:

◇ БИХ-фильтра 250

◇ КИХ-фильтра 193

◇ ЛДС 127

◇ полифазная 410, 414, 435, 440

◇ рекурсивных звеньев 127

◇ ЦФ с ФТ 301, 313

У

Уравнение разностное 121

Усреднение:

◇ по времени 99

◇ статистическое 98

Ф

Формула свертки 121

Функции:

◇ внешние 73

◇ встроенные 21

◇ генерирования типовых матриц 36

- ◇ математической статистики 43
- ◇ оформления графиков 61
- ◇ построения:
 - двумерных графиков 61
 - трехмерных графиков 64
- ◇ преобразования систем счисления 23
- ◇ элементарные математические 22
- Функция:
 - ◇ abs 22
 - ◇ acos 22
 - ◇ acot 22
 - ◇ angle 22
 - ◇ arburg 365
 - ◇ arcov 365
 - ◇ armcov 365
 - ◇ aryule 365
 - ◇ asin 22
 - ◇ atan 22
 - ◇ bandpass 223
 - ◇ bandstop 223
 - ◇ bin2dec 23
 - ◇ butter 253, 255
 - ◇ butterord 253
 - ◇ ceil 23
 - ◇ cellplot 56
 - ◇ char 52
 - ◇ cheby1 253, 255
 - ◇ cheby1ord 253
 - ◇ cheby2 253, 255
 - ◇ cheby2ord 253
 - ◇ chirp 116
 - ◇ class 56
 - ◇ colormap 65, 340
 - ◇ complex 22
 - ◇ conv 122
 - ◇ convergent 23
 - ◇ cos 22
 - ◇ cot 22
 - ◇ dec2bin 23
 - ◇ dec2hex 23
 - ◇ decimate 408
 - ◇ det 41
 - ◇ df1 129, 251
 - ◇ df1sos 251
 - ◇ df1t 129, 251
 - ◇ df1tsos 251
 - ◇ df2 129, 251
 - ◇ df2sos 251
 - ◇ df2t 129, 251
 - ◇ dfasymfir 193
 - ◇ dffir 193
 - ◇ dfsymfir 193
 - ◇ diag 36
 - ◇ disp 75
 - ◇ double 50
 - ◇ ellip 253, 255
 - ◇ ellipord 253
 - ◇ equiripple 225
 - ◇ exp 22
 - ◇ eye 36
 - ◇ fft 145, 167
 - ◇ fftfilt 168, 169
 - ◇ figure 60
 - ◇ filter 122
 - ◇ fir1 196
 - ◇ firpm 222
 - ◇ firpmord 223
 - ◇ fix 23
 - ◇ floor 23
 - ◇ freqz 127, 226
 - ◇ hex2dec 23
 - ◇ highpass 223
 - ◇ hist 62
 - ◇ ifft 145
 - ◇ imag 22
 - ◇ impz 122, 226
 - ◇ input 75
 - ◇ int16 50
 - ◇ int32 50
 - ◇ int64 50
 - ◇ int8 50
 - ◇ interp 405
 - ◇ inv 41
 - ◇ kaiserord 197
 - ◇ kaiserwin 225
 - ◇ legend 61
 - ◇ length 35
 - ◇ log 22
 - ◇ log10 22
 - ◇ log2 22
 - ◇ logical 51
 - ◇ loglog 62
 - ◇ lowpass 223
 - ◇ magic 81
 - ◇ max 44
 - ◇ mean 100
 - ◇ mean 44
 - ◇ mesh 64
 - ◇ meshgrid 64
 - ◇ min 44
 - ◇ mod 22
 - ◇ norm 43
 - ◇ num2str 52
 - ◇ ones 36
 - ◇ pburg 366
 - ◇ periodogram 326
 - ◇ plot 62
 - ◇ plot3 64
 - ◇ pmcov 366
 - ◇ polyval 80

Функции (*прод.*):

- ◇ prcov 366
- ◇ prod 44
- ◇ psd 327, 334, 367
- ◇ pwelch 333
- ◇ pyulear 366
- ◇ rand 36, 100
- ◇ randn 36, 100
- ◇ real 22
- ◇ repmat 37
- ◇ resample 409
- ◇ residuez 125
- ◇ rmfield 55
- ◇ roots 90
- ◇ round 23
- ◇ semilogx 62
- ◇ semilogy 62
- ◇ sin 22
- ◇ sinc 117
- ◇ single 50
- ◇ size 35
- ◇ smooth 331
- ◇ sort 44
- ◇ spectrogram 340
- ◇ sqrt 22
- ◇ std 44
- ◇ stem 62
- ◇ strcat 75
- ◇ subplot 60
- ◇ sum 44
- ◇ surf 64
- ◇ tan 22
- ◇ tf2sos 125
- ◇ tf2zpk 125
- ◇ title 61
- ◇ toeplitz 36
- ◇ uint16 50
- ◇ uint32 50
- ◇ uint64 50
- ◇ uint8 50
- ◇ var 44, 100

- ◇ xcorr 336
- ◇ xcorr 97
- ◇ xcov 97
- ◇ xlabel 61
- ◇ xlim 61
- ◇ ylabel 61
- ◇ ylim 61
- ◇ zeros 36
- ◇ xlabel 61
- ◇ zlim 61
- ◇ zplane 125
- ◇ автоковариационная 96
- ◇ автокорреляционная 96
- ◇ передаточная 123
- ФЧХ 126

X

Характеристика:

- ◇ затухания 190
- ◇ импульсная 121
- ◇ ослабления 190
- ◇ частотная 125
 - выравнивание 478

Ц

Цифровой фильтр:

- ◇ Найквиста 413
- ◇ с ФТ 299
- ◇ этапы проектирования 187

Ч

Частота:

- ◇ дискретная нормированная 143
- ◇ нормированная 126
- ◇ разрыва 195
- ◇ среза 254