# ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ моделирование в SIMULINK

Bведение в Simulink

- Технология моделирования в Simulink
- Дискретные и цифровые сигналы
- Системы цифровой фильтрации
- Системы цифровой фильтрации с фиксированной точкой
- Средства вычисления ДПФ на основе БПФ
- Многоскоростные системы





### А. И. Солонина

# ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ моделирование в SIMULINK

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» 2012

УДК 681.3.06

#### ББК 32.973.26-018.2

C60

#### Солонина А. И.

C60

#### Цифровая обработка сигналов. Моделирование в Simulink. — СПб.: БХВ-Петербург, 2012. — 432 с.: ил. — (Учебное пособие)

ISBN 978-5-9775-0686-1

Учебное пособие посвящено технологии создания Simulink-моделей и моделирования динамических систем цифровой обработки сигналов (ЦОС) с использованием базовых методов и алгоритмов ЦОС. Для облегчения работы начинающих пользователей даны основы моделирования в Simulink. Подробно описана технология создания Simulink-моделей дискретных и цифровых сигналов, специфика их обработки и анализа в процессе моделирования. Рассмотрены математические модели и Simulink-моделе дискретных систем, систем цифровой фильтрации с КИХ- и БИХ-фильтрами различных структур, в том числе с фиксированной точкой, многоскоростных систем с полифазными структурами, а также средства вычисления дискретного преобразования Фурье на основе быстрого преобразования Фурье.

Для студентов и преподавателей колледжей и вузов, инженеров-конструкторов

#### 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.

> УДК 681.3.06 ББК 32.973.26-018.2

#### Группа подготовки издания:

Главный редактор	Екатерина Кондукова
Зам. главного редактора	Евгений Рыбаков
Зав. редакцией	Григорий Добин
Редактор	Екатерина Капалыгина
Компьютерная верстка	Татьяны Олоновой
Корректор	Наталия Першакова
Дизайн серии	Инны Тачиной
Оформление обложки	Марины Дамбиевой
Зав. производством	Николай Тверских

Подписано в печать 14.03.2012. Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 34,83. Тираж 1200 экз. Заказ № "БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29. Отпечатано с готовых диапозитивов в ГУП "Типография "Наука" 199034, Санкт-Петербург, 9 линия, 12.

## Оглавление

Предисловие	11
Часть I. Знакомство с Simulink	13
Глава 1. Simulink как подсистема MATLAB	15
1.1. Средства моделирования в MATLAB. Специфика Simulink	15
1.2. Принятые обозначения	16
1.3. Установка и запуск Simulink	17
1.4. Библиотека Simulink	19
1.5. Система помощи MATLAB	23
1.5.1. Команда <i>help</i>	24
1.5.2. Пункт меню <i>Help</i> окна <i>MATLAB</i>	24
1.5.3. Пункт меню Help окна Simulink Library Browser	26
Глава 2. Технология создания S-модели системы	28
2.1. Математическая модель и S-модель системы	28
2.2. Основные этапы создания S-модели системы	31
2.3. Редактирование S-модели системы	35
2.3.1. Операции с компонентами S-модели системы	35
2.3.2. Операции по оформлению S-моделей системы	38
2.4. Исследование S-модели	41
Глава 3. Технология моделирования системы	43
3.1. Настройка S-молели системы	43
3.2. Настройка обмена данными S-модели системы с рабочим пространством	
памяти Workspace	50
3.3. Настройка диагностики моделирования	54
3.4. Моделирование системы — запуск, пауза и останов	56
3.5. Примеры моделирования простейших систем	56
Глава 4. Сигналы	58
4.1. S-модели сигналов	58
4.2. Параметры блоков	59
4.2.1. Блок Band-Limited White Noise	60

4.2.2. Блок Chirp Signal	
4.2.3. Блок Clock	
4.2.4. Блок Constant	
4.2.5. Блок Counter Free-Running	
4.2.6. Блок Counter Limited	
4.2.7. Блок Digital Clock	
4.2.8. Блок From File	
4.2.9. Блок From Workspace	
4.2.10. Блок Ground	
4.2.11. Блок In	
4.2.12. Блок Pulse Generator	
4.2.13. Блок Ramp	
4.2.14. Блок Random Number	
4.2.15. Блок Repeating Sequence	
4.2.16. Блок Repeating Sequence Interpolated	
4.2.17. Блок Repeating Sequence Stair.	
4.2.18. Блок Signal Generator	
4.2.19. Блок Sine Wave	
4.2.20. Блок Step	
4.2.21. Блок Uniform Random Number	
4.3. Виртуальные и невиртуальные блоки	
Глава 5. Средства анализа сигналов	
5.1. S-модели средств анализа сигналов	
5.2. Параметры блоков	
5.2.1. Блок Display	
5.2.2. Блок Out	
5.2.3. Блок Scope	
5.2.4. Блок Stop Simulation	
5.2.5. Блок Floating Scope	
5.2.6. Блок Terminator	
5.2.7. Блок То File	
5.2.8. Блок To Workspace	95
	······································
5.2.9. Блок XY Graph	
5.2.9. Блок XY Graph Глава 6. Математические преобразования	
<ul> <li>5.2.9. Блок ХҮ Graph</li> <li>Глава 6. Математические преобразования</li> <li>6.1. Средства библиотеки блоков Simulink</li> </ul>	
<ul> <li>5.2.9. Блок ХҮ Graph</li> <li>Глава 6. Математические преобразования</li> <li>6.1. Средства библиотеки блоков Simulink</li> <li>6.1.1. Арифметические операции</li> </ul>	
<ul> <li>5.2.9. Блок ХҮ Graph</li> <li><b>Глава 6. Математические преобразования</b></li> <li>6.1. Средства библиотеки блоков Simulink</li> <li>6.1.1. Арифметические операции</li></ul>	98 
<ul> <li>5.2.9. Блок ХҮ Graph</li> <li><b>Глава 6. Математические преобразования</b></li> <li>6.1. Средства библиотеки блоков Simulink</li> <li>6.1.1. Арифметические операции</li></ul>	98 
5.2.9. Блок XY Graph         Глава 6. Математические преобразования         6.1. Средства библиотеки блоков Simulink         6.1.1. Арифметические операции         6.1.1.1. Блок Add         6.1.1.2. Блок Bias         6.1.1.3. Блок Divide	98 
5.2.9. Блок XY Graph         Глава 6. Математические преобразования         6.1. Средства библиотеки блоков Simulink         6.1.1. Арифметические операции         6.1.1.1. Блок Add         6.1.1.2. Блок Bias         6.1.1.3. Блок Divide         6.1.1.4. Блок Dot Product	98 98 99 99 99 100 103 103 103

6.1.1.6. Блок Product	106
6.1.1.7. Блок Product of Elements	108
6.1.1.8. Блок Slider Gain	
6.1.1.9. Блок Subtract	
6.1.1.10. Блок Sum	
6.1.1.11. Блок Sum of Elements	108
6.1.1.12. Блок Unary Minus	
6.1.2. Элементарные математические функции	
6.1.2.1. Блок Abs	109
6.1.2.2. Блок Math Function	110
6.1.2.3. Блок Sign	111
6.1.2.4. Блок Trigonometric Function	111
6.1.3. Операции с комплексными числами	
6.1.3.1. Блок Complex to Magnitude-Angle	
6.1.3.2. Блок Complex to Real-Imag	
6.1.3.3. Блок Magnitude-Angle to Complex	113
6.1.3.4. Блок Real-Imag to Complex	113
6.1.4. Округление с помощью блока Rounding Function	
6.1.5. Операции с матрицами и векторами	
6.1.5.1. Блок Matrix Concatenate	115
6.1.5.2. Блок MinMax	
6.1.5.3. Блок Reshape	
6.1.5.4. Блок Vector Concatenate	
6.1.6. Вычисление корней уравнения и значений многочлена	
6.1.6.1. Блок Algebraic Constraint	
6.1.6.2. Блок Polynomial	
6.1.7. Операции отношения и логические операции	
6.1.7.1. Блок Compare To Constant	119
6.1.7.2. Блок Compare To Zero	
6.1.7.3. Блок Relational Operator	
6.1.7.4. Блок Logic Operator	
6.1.8. Побитовые операции	
6.1.8.1. Блок Bit Clear	
6.1.8.2. Блок Bit Set	
6.1.8.3. Блок Bitwise Operator	123
6.1.8.4. Блок Shift Arithmetic	124
6.1.9. Объекты MATLAB	
6.1.9.1. Блок Fcn	125
6.1.9.2. Блок MATLAB Fcn	126
6.1.9.3. Блок Embedded MATLAB Function	127
6.1.10. Основные характеристики матрицы	
6.2. Средства библиотеки блоков Signal Processing Blockset	
6.2.1. Специальные математические операции с матрицами	
6.2.1.1. Блок Complex Exponential	
6.2.1.2. Блок Cumulative Product	
6.2.1.3. Блок Cumulative Sum	
6.2.1.4. Блок dB Conversion	131

6.2.1.5. Блок Difference	132
6.2.1.6. Блок Normalization	133
6.2.2. Типовые матрицы	135
6.2.2.1. Блок Create Diagonal Matrix	135
6.2.2.2. Блок Extract Diagonal	136
6.2.2.3. Блок Extract Triangular Matrix	136
6.2.2.4. Блок Identity Matrix	136
6.2.2.5. Блок Toeplitz	137
6.2.3. Преобразование матриц	
6.2.3.1. Блок Matrix Concatenate	138
6.2.3.2. Блок Overwrite Values	138
6.2.3.3. Блок Permute Matrix	140
6.2.3.4. Блок Submatrix	141
6.2.4. Операции с матрицами в задачах линейной алгебры	
6.2.4.1. Блок Matrix Multiply	142
6.2.4.2. Блок Matrix Product	142
6.2.4.3. Блок Matrix Square	142
6.2.4.4. Блок Matrix Sum	143
6.2.4.5. Транспонирование и эрмитово сопряжение матрицы	143
6.2.5. Обращение матрицы	144
6.2.5.1. Блок LU Inverse	144
6.2.5.2. Блок LDL Inverse	145
6.2.5.3. Блок Cholesky Inverse	146
6.2.6. Разложение матрицы	146
6.2.6.1. Блок LU Factorization	147
6.2.6.2. Блок QR Factorization	148
6.2.6.3. Блок Cholesky Factorization	148
6.2.7. Решение систем линейных алгебраических уравнений	149
6.2.7.1. Блок LU Solver	150
6.2.7.2. Блок QR Solver	150
6.2.7.3. Блок Cholesky Solver	
6.2.8. Операции с матрицами в задачах математической статистики	
6.2.8.1. Блок Autocorrelation	
6.2.8.2. Блок Correlation	
6.2.8.3. Блок Махітит	
6.2.8.4. Блок Mean	
6.2.8.5. Блок Мілітит	
6.2.8.0. БЛОК КМS	
6.2.8. /. DJOK SOIT	
6.2.8.8. BJOK Standard Deviation	
0.2.8.9. DJOK Variance	100
Глава 7. Средства управления сигналами	160
7.1. S-модели средств управления маршрутом сигнала	160
7.1.1. Блок Mus	161
7.1.2. Блок Demux	

714 Fuor Bus Selector	105
7.1.4. DJOK DUS SCICCIOI	166
7.1.5. Блок Goto	167
7.1.6. Блок From	169
7.1.7. Блок Manual Switch	169
7.1.8. Блок Switch	169
7.1.9. Блок Selector	170
7.1.10. Блок Merge	173
7.2. S-модели средств управления атрибутами сигнала	176
7.2.1. Блок Bus to Vector	176
7.2.2. Блок Data Type Conversion	177
7.2.3. Блок Data Type Conversion Inherited	180
7.2.4. Блок Data Type Duplicate	181
7.2.5. Блок IC	181
7.2.6. Блок Probe	182
7.2.7. Блок Rate Transition	183
7.2.8. Блок Width	185
7.3. S-модели задержки и хранения сигнала	186
7.3.1. Блок Integer Delay	186
7.3.2. Блок Memory	187
7.3.3. Блок Unit Delay	188
Глава 8. Подсистемы	189
8.1. Разновилности полеистем	189
8.2 Неуправляемые полеистемы	190
8.2. Неуправляемые подсистемы	190
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 190
8.2. Неуправляемые подсистемы           8.2.1. Команда Create Subsystem           8.2.2. Блок Subsystem           8.2.3. Параметры неуправляемых полсистем	190 190 190 191 192
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 191 192 194
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 191 192 194 195
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 191 191 192 194 195 195
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 191 192 192 194 195 195 199
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 191 192 194 195 195 199 203
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 191 192 194 195 195 199 203 203
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 191 192 194 195 195 199 203 203 203
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 191 192 194 195 195 199 203 203 203 203
<ul> <li>8.2. Неуправляемые подсистемы.</li> <li>8.2.1. Команда Create Subsystem</li> <li>8.2.2. Блок Subsystem</li> <li>8.2.3. Параметры неуправляемых подсистем</li> <li>8.2.4. Иерархическая структура S-модели системы.</li> <li>8.3. Управляемые условные подсистемы.</li> <li>8.3.1. Блок Enabled Subsystem</li> <li>8.3.2. Блок Triggered Subsystem.</li> <li>8.3.3. Блок Enabled and Triggered Subsystem.</li> <li>8.4. Управляемые подсистемы, моделирующие логику управления потоком</li> <li>8.4.1. Блок If Action Subsystem.</li> <li>8.4.2. Блок Switch Case Action Subsystem.</li> <li>8.4.3. Блок For Iterator Subsystem</li> </ul>	190 190 190 191 192 194 195 195 195 199 203 203 203 208 211
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 191 192 194 195 195 195 199 203 203 203 208 211 215
<ul> <li>8.2. Неуправляемые подсистемы.</li> <li>8.2.1. Команда Create Subsystem</li> <li>8.2.2. Блок Subsystem</li> <li>8.2.3. Параметры неуправляемых подсистем</li> <li>8.2.4. Иерархическая структура S-модели системы.</li> <li>8.3. Управляемые условные подсистемы.</li> <li>8.3.1. Блок Enabled Subsystem</li> <li>8.3.2. Блок Triggered Subsystem.</li> <li>8.3.3. Блок Enabled and Triggered Subsystem.</li> <li>8.4. Управляемые подсистемы, моделирующие логику управления потоком</li> <li>8.4.1. Блок If Action Subsystem.</li> <li>8.4.2. Блок Switch Case Action Subsystem.</li> <li>8.4.3. Блок For Iterator Subsystem.</li> <li>8.4.4. Блок While Iterator Subsystem.</li> </ul>	190 190 190 191 192 194 195 195 199 203 203 203 208 211 215 220
<ul> <li>8.2. Неуправляемые подсистемы.</li> <li>8.2.1. Команда Create Subsystem</li> <li>8.2.2. Блок Subsystem</li> <li>8.2.3. Параметры неуправляемых подсистем</li> <li>8.2.4. Иерархическая структура S-модели системы</li> <li>8.3. Управляемые условные подсистемы.</li> <li>8.3.1. Блок Enabled Subsystem</li> <li>8.3.2. Блок Triggered Subsystem.</li> <li>8.3.3. Блок Enabled and Triggered Subsystem.</li> <li>8.4. Управляемые подсистемы, моделирующие логику управления потоком.</li> <li>8.4.1. Блок If Action Subsystem</li> <li>8.4.2. Блок Switch Case Action Subsystem.</li> <li>8.4.3. Блок For Iterator Subsystem.</li> <li>8.5. Маскированные подсистемы</li> </ul>	190 190 190 191 192 194 195 195 195 199 203 203 203 208 211 215 220
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 190 191 192 194 195 195 195 199 203 203 203 208 211 215 220
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 190 191 192 194 195 195 195 195 195 203 203 203 203 203 215 220 <b> 233</b>
<ul> <li>8.2. Неуправляемые подсистемы</li></ul>	190 190 190 191 192 194 195 195 195 195 195 203 203 203 203 211 220 <b> 233</b> 233 235

Глава 10. Отладка S-модели системы	240
10.1 Метол блока и никл молелирования	240
10.2. Запуск Debugger's GUI и окно отладчика	241
Часть II. Моделирование цифровой обработки сигналов средствами	240
Simulink	249
Глава 11. Дискретные сигналы	251
11.1. Дискретные сигналы: библиотека блоков Simulink	251
11.1.1. Односигнальная последовательность	252
11.1.2. Многосигнальная последовательность	254
11.1.3. Одноканальная последовательность	255
11.1.4. Многоканальная последовательность	257
11.1.5. Одноканальная последовательность фреймов	260
11.1.6. Многоканальная последовательность фреймов	261
11.2. Дискретные сигналы: библиотека блоков Signal Processing Blockset	262
11.2.1. S-модели средств анализа дискретных сигналов	262
11.2.1.1. Блок Display	262
11.2.1.2. Блок Matrix Viewer	263
11.2.1.3. Блок Signal To Workspace	264
11.2.1.4. Блок Triggered To Workspace	
11.2.1.5. Блок Time Scope	
11.2.1.6. БЛОК Vector Scope	
11.2.1.7. BJIOK Wateriali	
11.2.2. 5-модели дискретных сигналов	277
11.2.2.1. БЛОК СППР 11.2.2.2. Блок Constant	
11.2.2.2. Diok Constant Diagonal Matrix	
11.2.2.4 Блок Discrete Impulse	280
11.2.2.5. Блок Identity Matrix	
11.2.2.6. Блок Random Sources	281
11.2.2.7. Блок Signal From Workspace	282
11.2.2.8. Блок Sine Wave	283
11.2.3. Представление дискретных сигналов	286
11.2.3.1. Одноканальная последовательность	286
11.2.3.2. Многоканальная последовательность	287
11.2.3.3. Одноканальная последовательность фреймов	
11.2.3.4. Многоканальная последовательность фреймов	
11.3. Средства управления дискретными сигналами	290
11.3.1. Блок Butter	291
11.3.2. Блок Unbutter	293
11.3.3. Блок Flip	294
11.3.4. Блок Convert 1-D to 2-D	295
11.3.5. Блок Convert 2-D to 1-D	295

11.3.6. Блок N-Sample Enable	295
11.3.7. Блок N-Sample Switch	296
11.3.8. Блок Data Type Conversion	297
11.3.9. Блок Owerwrite Values	297
11.3.10. Блок Selector	297
11.3.11. Блок Submatrix	298
11.4. Операции с дискретными сигналами	298
11.4.1. Блок Constant Ramp	298
11.4.2. Блок Convolution	299
11.4.3. Блок Downsample	300
11.4.4. Блок Pad	301
11.4.5. Блок Peak Finder	303
11.4.6. Блок Repeat	305
11.4.7. Блок Upsample	306
11.4.8. Блок Window Function	306
11.4.9. Блок Zero Crossing	309
11.5. Генерирование типовых последовательностей	310
Глава 12. Линейные дискретные системы	312
12.1. Математическая молель и S-молель ЛЛС	
12.2. S-модели структур ЛЛС	315
12.2.1. Блок Discrete Filter	316
12.2.2. Блок Discrete FIR Filter	317
12.2.3. Блок Discrete State-Space	318
12.2.4. Блок Discrete Zero-Pole	320
12.3. Анализ характеристик S-модели ЛДС	322
Глава 13. Системы цифровой фильтрации	324
13.1. Проектирование цифровых фильтров	324
13.2. Свойства КИХ-фильтров	325
13.3. Задание требований к частотным характеристикам КИХ-фильтров	326
13.4. Свойства БИХ-фильтров	329
13.5. Задание требований к частотным характеристикам БИХ-фильтров	330
13.6. Структуры цифровых фильтров	330
13.7. Математическая модель и S-модель системы цифровой фильтрации	333
13.8. S-модель ЦФ и средства ее создания	334
13.8.1. S-модель ЦФ в виде блока Digital Filter Design	334
13.8.2. S-модель ЦФ в виде блока из группы блоков Filter Design Toolbox	337
13.8.2.1. Блок Lowpass Filter	337
13.9. Анализ характеристик S-модели ЦФ	340
13.10. Примеры S-моделей ЦФ	341
13.11. S-модели систем цифровой фильтрации	351

Глава 14. Системы цифровой фильтрации с фиксированной точкой	353
14.1. S-модель ЦФ с ФТ и средства ее создания	353
14.2. Создание S-модели ЦФ с ФТ средствами блока Filter Realization Wizard	353
14.2.1. Синтез или импорт из Workspace исходного ЦФ	354
14.2.2. Расстановка звеньев и масштабирование в каскадных	
структурах БИХ-фильтров	356
14.2.3. Создание S-модели исходного ЦФ	360
14.2.4. Моделирование структуры ЦФ с ФТ	363
14.2.4.1. Установка свойств ЦФ с ФТ на вкладке Coefficients	364
14.2.4.2. Установка свойств ЦФ с ФТ на вкладке Input/Output	366
14.2.4.3. Установка свойств ЦФ с ФТ на вкладке Filter Internals	369
14.2.5. Создание S-модели ЦФ с ФТ	373
14.3. Создание S-модели ЦФ с ФТ в виде блока Digital Filter	374
14.4. Параметры блока Digital Filter	375
14.5. Создание S-моделей исходного ЦФ и ЦФ с ФТ программными	
средствами МАТLАВ	379
14.6. S-модели цифровых сигналов	380
14.7. S-модели систем цифровой фильтрации с ФГ	380
Глава 15. Срелства вычисления ЛПФ	384
	201
15.2 S молони спонств в инеления ШПФ	304
15.2. Б-модели средств вычисления ДПФ	300
15.2.1. DЛОК ГГ 1 15.2.2. Блок IEET	307
15.2.2. DIOK II T I	380
15.2.5. Осооснности вычисления дне в Sintunink	300
13.5. Примеры 5-моделей систем обработки сигналов с вычислением ДттФ	590
Глава 16. Многоскоростные системы ЦОС	398
16.1. Математическая модель системы однократной интерполяции	399
16.2. Математическая модель системы однократной децимации	402
16.3. Математическая модель системы однократной передискретизации	404
16.4. Полифазные структуры многоскоростных систем и их описание	
в MATLAB	405
16.5. S-модели многоскоростных систем	408
16.5.1. Блок FIR Interpolation	409
16.5.2. Блок FIR Decimation	411
16.5.3. Блок FIR Rate Conversion	411
16.6. Моделирование многоскоростной обработки сигналов	412
Список литературы	417
Список сокращений	419
Список сокращений на английском языке	419
Список сокращений на русском языке	419
Предметный указатель	421

### Предисловие

Тенденции в области телекоммуникаций связаны с повсеместным распространением цифровой обработки сигналов (ЦОС). Одним из перспективных научнотехнических направлений является разработка новых методов и алгоритмов ЦОС для различных приложений и создание на их основе программных продуктов (softпродуктов) или реализация последних в виде устройств (hard-продуктов) на базе цифровых процессоров обработки сигналов (ЦПОС), программируемых логических интегральных схемах (ПЛИС), систем на кристалле (System on Chip — SoC) и т. п.

Сложность технологии создания подобных продуктов требует высокого уровня профессиональной подготовки бакалавров и магистров в области ЦОС. Радикально меняется характер работы инженера — она все больше сводится к компьютерному моделированию и проектированию.

Базовая подготовка по ЦОС подразумевает изучение теории ЦОС, технологий компьютерного моделирования и реализации алгоритмов ЦОС на современной цифровой элементной базе. В целом, это дает системное представление о технологии создания soft- и hard-продуктов.

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

В настоящее время к общепризнанным мировым универсальным стандартам в области компьютерных технологий относится система (программная среда) МАТLAB, предназначенная для компьютерного моделирования в самых разных областях науки и техники. Система МАTLAB постоянно расширяется и совершенствуется, к чему привлекаются лучшие специалисты разных стран. К ее основным достоинствам относятся: алгоритмический язык "сверхвысокого" уровня с матричной обработкой данных; колоссальная библиотека стандартных функций; огромное разнообразие графических средств; удобные средства создания и отладки программ; широкий набор программных средств (Toolbox), сгруппированных по специализированным приложениям, средств графического интерфейса пользователя (GUI) без использования алгоритмического языка в явном виде и средств Simulink общего и специального назначения для блочного моделирования динамических систем.

Предыдущая книга автора "Цифровая обработка сигналов. Моделирование в MATLAB" была посвящена моделированию ЦОС программными средствами MATLAB и средствами GUI.

Настоящая книга посвящена моделированию динамических систем ЦОС средствами Simulink, на основе которого в дальнейшем может выполняться проектирование систем ЦОС на цифровой элементной базе с помощью целевых программ, входящих в состав MATLAB.

Актуальная проблема подготовки современных научно-технических специалистов, владеющих навыками моделирования ЦОС в Simulink, предполагает включение соответствующих разделов в дисциплины ЦОС для бакалавров и магистров, а следовательно, обеспечение учебного процесса необходимой методической литературой.

Тема моделирования ЦОС в Simulink весьма обширна и в отечественной литературе освящена лишь фрагментарно. Предлагаемую книгу следует рассматривать, как первую попытку восполнить этот пробел. В ней затрагиваются базовые методы и алгоритмы ЦОС. Сразу оговоримся, система Simulink необъятная, и даже в названной ограниченной области автор не претендует на исчерпывающее описание возможностей Simulink. Тонкости и детали, равно как и разнообразие средств, предлагаемых для решения поставленных задач, можно постичь только на практике, привлекая мощную систему помощи Simulink. Автор постарался систематизировать и описать типовые приемы и средства моделирования ЦОС в Simulink, проиллюстрировав их на простых примерах. Для облегчения работы начинающих пользователей и целостности материала книги в ее первой части даны основы моделирования в Simulink, а во всех главах второй части — краткие теоретические справки по рассматриваемым разделам ЦОС.

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

Содержание книги включает 16 глав, которые тематически разделены на две части:

1. Знакомство с Simulink.

2. Моделирование цифровой обработки сигналов средствами Simulink.

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



## Часть I

## Знакомство с Simulink

#### Глава 1



## Simulink как подсистема MATLAB

Система MATLAB (или коротко MATLAB) — это интерактивная система, предназначенная для компьютерного моделирования практически в любой области науки и техники. Она была создана в США компанией The MathWorks, Inc., и ее разработка для персональных компьютеров была представлена в начале 80-х годов. Информация о MATLAB доступна на сайтах www.mathworks.com, www.softline.ru, www.matlab.ru и www.exponenta.ru.

Изложение материала книги ориентировано на версии MATLAB R2009b— R2010a.

Мощный программный комплекс MATLAB состоит из *четырех компонент*: MATLAB, Simulink, Toolbox и Blockset.

Компонент MATLAB представляет собой ядро системы — ядро MATLAB.

Компонент Simulink является подсистемой (расширением) МАТLAB, предназначенной для блочного моделирования, однако, благодаря своим уникальным возможностям, этот компонент часто воспринимают как самостоятельную систему и называют ядром Simulink. Слово "simulink" образовано из комбинации первых четырех букв слова "simulation" (моделирование) и "link" (соединение).

Компоненты *Toolbox* (Набор инструментов) и *Blockset* (Набор блоков) — это *пакеты расширения* MATLAB и Simulink соответственно, сгруппированные по специализированным приложениям, назначение которых отображается в их названиях.

#### 1.1. Средства моделирования в MATLAB. Специфика Simulink

Для моделирования объектов и/или процессов в MATLAB предусмотрены следующие средства:

1. Программные средства.

Моделирование *программными средствами*, под которыми понимают средства алгоритмического языка MATLAB, выполняется в режиме прямых вычислений или в программном режиме MATLAB.

Программные средства универсальны, и другие средства разработаны на их основе. Моделирование ЦОС программными средствами рассматривалось в [22].

2. Стандартные программы GUI (Graphical User Interface — Графический интерфейс пользователя).

Моделирование средствами GUI выполняется в режиме *интерактивного общения* в среде конкретного GUI.

Средства GUI созданы на основе программных средств MATLAB, но позволяют исключить использование языка MATLAB в явном виде, что существенно упрощает технологию создания моделей, в том числе и с целью их дальнейшего использования в командном или программном режиме MATLAB.

Средства GUI предполагают знакомство с программными средствами MATLAB, по меньшей мере, на концептуальном уровне.

Моделирование ЦОС средствами GUI рассматривалось в [22].

3. Средства Simulink.

Моделирование в Simulink выполняется средствами блочного моделирования.

Средства Simulink созданы на основе программных средств MATLAB, но позволяют исключить или минимизировать использование языка MATLAB в явном виде, что существенно упрощает технологию моделирования.

Специфика моделирования в Simulink определяется его основным предназначением — *моделирование динамических систем* (Dynamic Systems).

Технология моделирования в Simulink заключается в построении модели системы из стандартных блоков и позволяет следить за ее работой с помощью стандартных средств наблюдения, поэтому моделирование в Simulink часто называют "визуальным".

В большинстве приложений динамическую систему можно фактически или условно представить как *систему обработки сигналов*, полагая, что понятия "сигнал" и "система" имеют широкое толкование, а именно:

- под сигналом понимается воздействие любой физической природы или последовательность данных;
- под системой понимается физическое устройство или математическое преобразование, выполняющее требуемое преобразование воздействия.

Компьютерная модель динамической системы строится на основе математических моделей сигналов и систем.

Моделированию ЦОС в Simulink и посвящается настоящая книга.

#### 1.2. Принятые обозначения

- названия окон, например окно Simulink Library Browser;
- пункты меню и команды в пунктах меню, например команда New в пункте меню File;

- кнопки, вкладки, панели, поля, группы, раскрывающиеся списки и т. п. в окнах, например раскрывающийся список Sampling mode в окне Source Block Parameters: Constant;
- первое упоминание имени группы блоков, например группа блоков Sources, а при последующих упоминаниях — группа блоков Sources;
- параметры блоков, например параметр Decimation блока Clock;

□ Courier New — для выделения:

- объектов на языке MATLAB, например h=a+sin(b);
- папок MATLAB и файлов пользователя, например папка work;
- Courier New полужирный для описания объектов языка MATLAB при их первом упоминании, например команда simulink;
- курсив для выделения ключевых слов или определений в тексте, например "библиотека Simulink имеет иерархическую структуру, представленную в виде дерева".

MATLAB не имеет русифицированной версии, поэтому при первом упоминании названий окон, пунктов меню, команд в пунктах меню, кнопок, вставок, параметров блоков и т. п. в скобках дается их перевод.

#### 1.3. Установка и запуск Simulink

Simulink автоматически ставится при установке MATLAB.

При установке MATLAB пользователь может самостоятельно формировать состав MATLAB на своем компьютере из интересующих его пакетов расширения, пометив их флажком в окне **Product and Folder Selection** (Выбор пакетов и папок) в раскрывающемся списке **Select Products to install** (Выбор пакетов для инсталляции). Исключение составляют обязательные ядро MATLAB и ядро Simulink.

При желании можно легко расширить состав системы. Для этого следует повторить процедуру установки, пометив флажком дополнительные пакеты расширения.

Для широкого круга задач, решаемых методами цифровой обработки, помимо ядер MATLAB и Simulink, рекомендуется установить пакеты расширения MATLAB (Toolbox) и Simulink (Blockset), перечисленные в табл. 1.1. Пакеты, относящиеся к Simulink, выделены серым цветом.

Название пакета	Назначение пакета
Communications Blockset	Системы связи
Communications Toolbox	Системы связи
Control System Toolbox	Системы управления
Curve Fitting Toolbox	Приближение кривых и данных

Таблица 1.1. Рекомендуемые для установки пакеты расширения MATLAB

Окончание табл. 1.1

Filter Design Toolbox	Проектирование цифровых фильтров
Fixed Point Toolbox	Обработка данных с фиксированной точкой
Gauges Blockset	Измерительные приборы
Neural Network Toolbox	Нейронные сети
Optimization Toolbox	Методы оптимизации
Real-Time Windows Target	Целевые средства Windows для реального времени
Real-Time Workshop	Лаборатория реального времени
Signal Processing Blockset	Обработка сигналов
Signal Processing Toolbox	Обработка сигналов
Simulink Control Design	Системы управления
Simulink Fixed Point	Обработка данных с фиксированной точкой
Spline Toolbox	Сплайн-функции
Statistics Toolbox	Методы математической статистики
Symbolic Math Toolbox	Обработка символьных данных
System Identification Toolbox	Методы идентификации
Wavelet Toolbox	Вейвлет-функции

По завершении установки МАТLAВ на **Рабочем столе** автоматически создается ярлык МАТLAВ. Для запуска МАТLAВ достаточно щелкнуть на этом ярлыке.

После появления фирменной заставки автоматически открывается окно **MATLAB**, представляющее собой *комбинацию окон* (рис. 1.1), благодаря чему обеспечивается оперативность действий пользователя при работе.

Интерфейс MATLAB образуют следующие окна:

- Command Window (Командное окно) основное окно интерактивной системы МАТLAB с активизированной командной строкой;
- □ Current Folder (Текущая папка) в этом окне выводится содержимое папки, имя которой отображается в раскрывающемся списке Current Folder на панели инструментов окна MATLAB. По умолчанию текущей считается папка со стандартным именем work, предназначенная для хранения файлов и папок, создаваемых пользователем;
- □ Workspace (Рабочая область памяти) в этом окне выводится список текущих переменных, сохраняемых в рабочей области памяти Workspace до выхода из МАТLAB;
- □ Command History (История команд) в этом окне выводится построчный список объектов языка МАТLAB, вводимых в ходе текущей и предшествующих сессий.



Рис. 1.1. Окно МАТLAB

Пользователь может произвольно менять состав активных окон с помощью команд пункта меню **Desktop** (Стол). На рис. 1.1 представлена полная комбинация окон, соответствующая команде **Desktop Layout** | **Default** (Организация стола | По умолчанию).

Остальные пункты меню окна **MATLAB** типичны для интерфейса Windows и не требуют комментариев. Команды пункта меню **Debug** (Отладка) предназначены для работы в режиме программирования [22].

Вывод информации о версии MATLAB и включенных пользователем при установке пакетах расширения выполняется в окне по команде:

ver

Установка Simulink приводит к появлению кнопки Simulink на панели инструментов окна MATLAB.

Запуск Simulink осуществляется одним из следующих способов:

- 1. С помощью кнопки Simulink на панели инструментов окна MATLAB.
- 2. По команде в окне Command Window:

simulink

#### 1.4. Библиотека Simulink

При запуске Simulink с помощью кнопки Simulink или по команде simulink открывается окно Simulink Library Browser (Каталог Библиотеки Simulink) (рис. 1.2).



Описание блока

Рис. 1.2. Окно Simulink Library Browser

Интерфейс данного окна образуют следующие элементы:

□ левая панель Libraries — с каталогом Библиотеки Simulink (Simulink Library);

Библиотека Simulink имеет иерархическую структуру, представленную в виде дерева.

В иерархической структуре Библиотеки Simulink можно выделить три уровня:

 верхний — узел дерева с пиктограммой, соответствующей поименованной библиотеке блоков (Block Library).

Первая из библиотек блоков (верхний узел дерева), входящих в состав Библиотеки Simulink, имеет такое же имя — Simulink;

• промежуточный — узел дерева без пиктограммы (или ветвь дерева), соответствующий поименованной *группе блоков* (Blocks).

Узлу дерева без пиктограммы соответствует вложенная группа блоков;

- нижний, соответствующий поименованному блоку (Block);
- правая панель с тремя вкладками: Library... (Библиотека...), где вместо многоточия будет указан путь к объекту, выделенному на левой панели Libraries; Search Results (Результаты поиска); Most Frequently Used Blocks (Наиболее часто используемые блоки).

На открытой вкладке Library... отображается состав библиотеки или группы блоков, выделенной на левой панели Libraries. Пиктограммы *группы* блоков выделяются цветом, а пиктограммы *блоков* черно-белые.

Если на правой панели отображаются *группы* блоков, то состав любой из них можно открыть двойным щелчком левой кнопки мыши на ее пиктограмме.

Если на правой панели отображаются *блоки*, то двойной щелчок левой кнопки мыши на пиктограмме *блока* открывает окно его *параметров* с неактивными полями. Назначение и установка параметров блоков будет рассматриваться в рамках соответствующих тем.

На открытой вкладке Search Results отображаются результаты поиска, о котором речь пойдет далее в этом разделе;

□ главное меню с пунктами:

- File (Файл) работа с файлами моделей Simulink и с собственной библиотекой пользователя, о создании которой пойдет речь в *гл. 9*;
- Edit (Редактирование) добавление в текущую модель Simulink блока, выделенного на правой панели, и поиск блока в библиотеке Simulink;
- View (Вид) управление видом окна Simulink Library Browser и выводом состава библиотеки блоков.

Команда Layout | List (Расположение | Список) обеспечивает вывод состава библиотеки блоков в виде одного столбца, а команда Layout | Grid (Расположение | Сетка) — компактный вывод в виде нескольких столбцов (рис. 1.3, сравните с рис. 1.2);

- Help (Помощь) обращение к разделу Simulink Системы помощи МАТLAB;
- панель инструментов с кнопками, дублирующими основные команды пунктов главного меню;
- текстовое поле Block Description (Описание блока) с краткой справкой о блоке, выделенном на правой панели; активизируется при выборе в пункте меню View команды Show Block Descriptions (Показать описание блока);
- □ поле ввода Enter search term (Введите компонент поиска), кнопка Search for subsystems and block by name (Поиск подсистем и блоков по имени) и кнопка

Search options (Варианты поиска) на панели инструментов (см. рис. 1.2) — используются для поиска подсистемы или блока по имени.



Рис. 1.3. Компактный вывод состава библиотеки блоков Simulink

Для поиска блока следует в поле ввода ввести его имя (полное или любую его часть) и нажать кнопку Search options (Варианты поиска). В открывшемся меню нужно установить требуемые флаги (один или несколько): Regular expression (Устойчивое выражение); Match case (Часть слова); Match whole word (Слово целиком). Затем нажать кнопку Search for subsystems and block by name. На правой панели автоматически откроется вкладка Search Results, имя которой автоматически изменится на Found: '...' (Найден: '...'), где в апострофах дублируется текст из поля ввода на панели инструментов, и на вкладке будут выведены результаты поиска блока с указанием библиотеки блоков (но без группы) или сообщение, если блок не найден.

На рис. 1.4 представлены результаты поиска по введенному слову "Constant" при установке флага **Match whole word**. Выведены все блоки, в имени которых присутствует это слово, с указанием библиотек блоков.



Рис. 1.4. Результаты поиска по слову "Constant" при установке флага Match whole word

Более подробную информацию по Библиотеке Simulink можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  User Guide  $\rightarrow$  Simulink Graphical User Interface  $\rightarrow$  Library Browser. К этому разделу можно обратиться автоматически по команде меню Help | Library Browser Help в окне Simulink Library Browser (*см. разд. 1.5.3*).

#### 1.5. Система помощи MATLAB

Помощь по Simulink организована в рамках системы помощи MATLAB, которая представляет собой совокупность следующих средств помощи пользователю: электронные справочные системы, а именно:

встроенная справочная система — формируется автоматически при установке системы МАТLAB, исходя из ее состава, определяемого пользователем. Она является информативно наиболее краткой и содержит иллюстративные примеры, которые можно копировать и выполнять;

 справочная система в формате HTML (Hyper Text Markup Language язык гипертекстовой маркировки) — автономна по отношению к системе MATLAB и содержится на отдельном CD-диске. Ее состав определяется пользователем при установке, подобно системе MATLAB, и, вообще говоря, может не совпадать с составом установленных пакетов расширения.

Эта справочная система информативно более полная, чем встроенная справка. В нее также включены иллюстративные примеры, которые можно копировать и выполнять в окне **Command Window**;

 справочная система в формате PDF (Portable Document Format — формат переносного документа) — также автономна по отношению к системе MATLAB. В комплекте MATLAB, начиная с версии 7.0, она на CD-диске не предоставляется, но доступна на сайте www.mathworks.com. Для чтения в формате PDF необходимо установить бесплатную программу Acrobat Reader.

Эта справочная система информативно наиболее полная и представляет собой так называемые "электронные книги", а точнее — серии книг, где каждая серия посвящена какой-то общей теме, а входящие в ее состав книги — конкретным вопросам в рамках данной темы. Подобно обычным книгам, электронные книги разбиты на главы и параграфы. Там, где это необходимо, справке по MATLAB с обязательными иллюстративными примерами предшествуют сведения теоретического характера со ссылками на литературу;

🛛 демонстрационные примеры.

В состав MATLAB включен значительный список демонстрационных примеров, в том числе относящихся к Simulink, которые позволяют составить представление о возможностях MATLAB и Simulink в решении самых разных задач.

#### 1.5.1. Команда help

Команда help обеспечивает обращение к встроенной справочной системе.

Список команд Simulink можно вывести по команде:

```
>> help simulink
```

а справку о конкретной команде — по ее имени, например:

>> help load\_system

#### 1.5.2. Пункт меню Help окна MATLAB

Пункт меню **Help** в окне **MATLAB** обеспечивает обращение к *справочной системе в формате HTML* и демонстрационным примерам MATLAB.

Назначение команд определяется их названием. В целом, организация помощи по данным командам достаточно простая и легко осваивается самостоятельно.

Остановимся коротко на одной, главной, команде этого пункта меню и поясним необходимые действия для получения справки.

По команде меню **Help** | **Product Help** открывается окно **Help** — окно справочной системы в формате HTML, содержащее две панели (рис. 1.5):

левую панель с двумя вкладками:

- Contents (Содержание) на этой вкладке выводится дерево, узлами которого являются разделы справочной системы MATLAB, при этом на правой панели отображается HTML-страница, соответствующая разделу, выделенному в дереве;
- Search Results (Результаты поиска) на этой вкладке можно получить справку по ключевому слову (фразе), которое набирается на левой панели в поле ввода над вкладками, после чего нужно нажать клавишу <Enter>. На этой вкладке отобразится список разделов, где набранное слово (фраза) встречается в справочной системе MATLAB, а на правой панели будет выведена HTML-страница выделенного на вкладке раздела (рис. 1.6);
- правую панель для отображения HTML-страницы со справкой по разделу, выделенному на левой панели.

В верхней части правой панели имеется поле, в котором выводится путь к разделу, выделенному на левой панели.



Рис. 1.5. Окно Help справочной системы MATLAB в формате HTML с открытой вкладкой Contents



Рис. 1.6. Окно Help справочной системы MATLAB в формате HTML с открытой вкладкой Search Results и результатами поиска по ключевому слову "Source"

#### 1.5.3. Пункт меню Help окна Simulink Library Browser

Пункт меню **Help** в окне **Simulink Library Browser** (см. рис. 1.2) обеспечивает обращение к *справочной системе MATLAB в формате HTML* непосредственно по разделу Simulink, включая демонстрационные примеры, и содержит следующие команды:

□ Help for the Selected Block (Помощь по выделенному блоку).

По этой команде происходит автоматическое обращение к справочной системе MATLAB в формате HTML, при этом в окне **Help** будет открыта HTMLстраница со справкой по блоку, выделенному на правой панели окна **Simulink Library Browser** (рис. 1.7);

#### □ Library Browser Help (Помощь по каталогу библиотеки Simulink).

По этой команде происходит автоматическое обращение к HTML-странице со справкой по каталогу Библиотеки Simulink (Library Browser);

**□** Simulink Help (Помощь по Simulink).

По этой команде происходит автоматическое обращение к справочной системе MATLAB в формате HTML по всему разделу Simulink (рис. 1.8).



Рис. 1.7. Окно Help справочной системы MATLAB в формате HTML, открытое по команде Help | Help for the Selected Block из окна Simulink Library Browser, со справкой о блоке Inport

dit View Go Favorites Desktop Window Help		
۶ • Q •	💠 🔹 🖑 🔹 Simulak 🕨	
nts Search Results	C: 1: 1:	
Search Result      Creating of Model      Creating Continued Subsystems      Performation of Subsystems      Referencing a Model      Working with Block Libraries      Working with Block Libraries      Working with Block Sees      Working with Block Sees      Working with Block      Using Concesses Signals      Working with Data      Using Concesses Signals      Working with Data      Using Concesses      Working with Data	Simulink <sup>®</sup> Blocks: Functions: BC Category Alphabetical List Bocumentation Set Getting Started Introduces Simulink tutorials and comprehensive product information User's Guide Provides Simulink tutorials and comprehensive product information Writing S-Functions Explain how to use S-functions to create custom blocks Explain how to use S-functions to create custom blocks Explain in Documentation List major examples in the Simulink documentation	
al: Simumic Veebugger al: Acceler along Models Carcening Models Carcening Charlow Models Carcening Custom Biolds Carcening Custom Biolds Carcening Custom Biolds Print/Frame Editor Simular Araphical User Interface Carcening Customation Harmeters Dialog Box Carcening Customation Harmeters Chalog Box Carcening Configuration Parameters Chalog Box Carcening Customation Harmeters Carcening Customation Ca	Product Demos  Simulink Demos  Presents a collection of demos that you can run from the Help browser to help you learn the product  What's New  Release Notes Summarizes new features, bur fixes, upgrade issues, etc.	

Рис. 1.8. Окно Help справочной системы MATLAB в формате HTML, открытое по команде Help | Simulink Help из окна Simulink Library Browser



#### Глава 2

## Технология создания S-модели системы

Компьютерную модель системы, созданную средствами Simulink, будем называть *S-моделью системы* — сокращение от Simulink-модель.

В русскоязычной литературе в теории аналоговых, дискретных и цифровых систем под "моделированием системы" по умолчанию подразумевают моделирование процесса ее работы.

В справочной системе MATLAB по Simulink понятие "моделирование системы" (Modeling a System) относится скорее к *созданию* S-модели системы (т. е. к описанию *статической* системы средствами Simulink), а для моделирования *процесса работы* системы добавляют уточняющий термин — "моделирование *динамической* системы" (Modeling a Dynamic System).

Во избежание путаницы будем пользоваться русскоязычной терминологией, и в технологии моделирования системы в Simulink выделим две части:

1. Создание S-модели системы.

2. Моделирование системы.

Основные этапы технологии создания S-модели рассматриваются в этой главе, а моделирования системы — в *гл. 3*.

## 2.1. Математическая модель и S-модель системы

Ранее (см. разд. 1.1) система была условно определена как система обработки сигналов.

Напомним [23], что системой обработки сигналов называется объект, выполняющий требуемое преобразование входного сигнала (воздействия) в выходной (реакцию).

*Математической моделью системы* называют ее соотношение вход/выход, которое устанавливает связь между входным и выходным сигналами, и в общем случае имеет вид операторного уравнения (рис. 2.1, *a*):

$$Y = F(X), \tag{2.1}$$

где, X и Y — векторы входных и выходных сигналов соответственно (воздействий и реакций), в частном случае — один входной и один выходной сигналы;

*F* — оператор, определяющий математическое преобразование (линейное или нелинейное, алгебраическое или дифференциальное и т. д.).



б

Рис. 2.1. К определению математической модели системы (а) и линейной системы (б)

Вид математической модели (2.1) определяют два фактора:

1. Тип системы.

Тип системы определяется типом обрабатываемых сигналов. Различают три основных типа систем [23]:

• аналоговые системы, преобразующие аналоговые входные сигналы в аналоговые выходные сигналы.

Аналоговым называют сигнал, непрерывный по времени и по состоянию. Такой сигнал описывается непрерывной или кусочно-непрерывной функцией;

 дискретные системы, преобразующие дискретные входные в дискретные выходные сигналы.

Дискретным называют сигнал, дискретный по времени и непрерывный по состоянию. Такой сигнал описывается числовой последовательностью (коротко — последовательностью);

 иифровые системы, преобразующие цифровые входные в цифровые выходные сигналы.

Цифровым называют сигнал, дискретный по времени и квантованный по состоянию. Такой сигнал описывается квантованной последовательностью.

2. Математическая форма представления оператора F.

Математическая форма представления оператора *F* определяется типом системы — *аналоговая* или *дискретная*, а также тем, *линейная* она или *нелинейная*.

В линейных аналоговых системах оператор *F* имеет две универсальные формы представления — свертка (интеграл) или линейное дифференциальное преобразование, а математическая модель (2.1) принимает соответственно вид

формулы свертки (интегрального уравнения) или линейного дифференциального уравнения.

В линейных дискретных системах оператор *F* имеет также две универсальные формы представления — свертка (сумма) или линейное разностное преобразование, а математическая модель (2.1) принимает соответственно вид формулы свертки (линейного уравнения) или линейного разностного уравнения.

В зависимости от вида математической модели *линейной* системы параметры оператора *F* однозначно определяются:

- импульсной характеристикой для математической модели в виде формулы свертки;
- *передаточной функцией* для математической модели в виде дифференциального или разностного уравнения.

В *нелинейных* аналоговых и дискретных системах оператор *F* имеет несколько универсальных форм представления [23], а математическая модель (2.1) принимает вид соответствующего нелинейного уравнения.

Наконец, речь может идти о системе, выполняющей некое *математическое преобразование* сигналов, когда оператор F не имеет универсальной формы представления, а определяется конкретным преобразованием.

S-модель системы создается на основе математической модели (2.1).

Для создания S-моделей линейных систем используется математическая модель (2.1), в которой параметры *оператора F* однозначно определяются *передаточной функцией*, а математическая модель системы (2.1) принимает вид *линейного уравнения*: дифференциального для аналоговых и разностного — для дискретных систем.

В *математической модели линейной* системы можно выделить три основные составляющие (рис. 2.1, б):

входной сигнал X;

**С** структура линейной системы, определяемая видом ее *передаточной функции*<sup>1</sup>;

**Выходной сигнал** *Y*.

В *S-модели линейной системы* им будут соответствовать три отличающиеся своим функциональным назначением компоненты:

□ S-модель входного сигнала *X*;

□ S-модель структуры линейной системы;

□ S-модель средств анализа выходного сигнала *Y*.

<sup>&</sup>lt;sup>1</sup> Или параметрами системы уравнений переменных состояния, о чем пойдет речь в разд. 12.1.

# 2.2. Основные этапы создания S-модели системы

В общем случае технология создания S-модели системы включает в себя следующие основные этапы:

1. Открытие окна S-моделей.

По команде меню File | New | Model (Файл | Новый | Модель) окна Simulink Library Browser *(см. разд.1.4)* открывается окно S-моделей untitled (безымянное) для создания новой S-модели системы (рис. 2.2).

Ту же команду можно выполнить непосредственно в окне МАТLАВ.



Рис. 2.2. Окно untitled для создания новой S-модели

Интерфейс окна S-моделей соответствует интерфейсу Windows-приложений.

Команды меню окна S-моделей мы рассмотрим по мере изложения материала.

В строке состояния окна S-моделей (нижняя строка) отображаются (слева направо) (см. рис. 2.2):

- стадия моделирования Ready (Завершено) или Running (Выполнение), а в статическом состоянии при обращении к командам меню — краткая информация о выбранной команде;
- масштаб S-модели, по умолчанию 100%;
- индикатор процесса моделирования (активен при запуске модели);

- индикатор текущего времени моделирования (активен при запуске модели);
- выбранный решатель, о котором речь пойдет в разд. 3.3.
- 2. Перетаскивание блоков (компонентов S-модели системы) из библиотеки Simulink.

В окне **untitled** новая S-модель системы создается из блоков библиотеки Simulink с помощью технологии drag-and-drop ("перетащить и отпустить"). Она заключается в последовательном перетаскивании требуемых блоков в окно S-моделей **untitled** при нажатой левой кнопке мыши с правой панели окна Simulink Library Browser (*см. разд. 1.4*).

В этой главе для примеров воспользуемся блоками из библиотеки блоков Simulink, которые подробно будут рассматриваться в следующих главах.

Перетащим в окно **untitled** блоки, которые нам потребуются для создания S-моделей двух простейших систем:

• линейной аналоговой первого порядка.

Математическая модель системы имеет вид линейного дифференциального уравнения первого порядка.

Для примера выберем простейшую структуру — *RC*- или *RL*-цепь с передаточной функцией<sup>1</sup>:

$$H(s) = \frac{1}{s+1}.$$
 (2.2)

В качестве S-модели структуры системы выберем блок Transfer Fcn из группы блоков Continuous;

• линейной дискретной первого порядка.

Математическая модель системы имеет вид линейного разностного уравнения первого порядка.

Для примера выберем простейшую структуру — базовое звено 1-го порядка с передаточной функцией:

$$H(z) = \frac{1}{1+0.5z^{-1}}.$$
 (2.3)

В качестве S-модели структуры системы выберем блок Discrete Filter из группы блоков Discrete.

В качестве *S-модели входного сигнала* выберем блок Step из группы блоков Sources, генерирующий *единичный скачок* для аналоговой системы и *цифровой единичный скачок* для дискретной. При перетаскивании двух блоков Step для создания в одном окне **untitled** одновременно двух S-моделей систем, аналоговой и дискретной, один из блоков будет иметь имя Step, а второй — Step1.

<sup>&</sup>lt;sup>1</sup> В формуле использовано обозначение оператора Лапласа, принятое в Simulink.

В общем случае имена всех блоков S-модели должны быть уникальными. По умолчанию одинаковые блоки различаются номерами, которые автоматически добавляются к имени в порядке перетаскивания блоков.

В качестве S-моделей средств анализа входного и выходного сигналов выберем блок Scope (осциллограф) из группы блоков Sinks. Аналогично, при перетаскивании двух блоков Scope для создания двух S-моделей систем, аналоговой и дискретной, один из блоков будет иметь имя Scope, а второй — Scope1.

Блоки, подготовленные для создания S-моделей аналоговой и дискретной систем, представлены на рис. 2.3.



Рис. 2.3. Окно untitled с блоками для создания S-моделей аналоговой и дискретной систем

3. Задание параметров блоков.

Параметры блока, определяющие вид сигнала на его выходе или настройку средства анализа сигнала, задаются в окне **Parameters** (Параметры), которое открывается двойным щелчком левой кнопки мыши на пиктограмме блока.

Подробно параметры блока Step будут рассматриваться в разд. 4.2.20.

Параметры блока Step для S-модели аналоговой системы зададим по умолчанию, а один из параметров блока Step1 для S-модели дискретной системы изменим: период дискретизации **Sample Time:** 0.2.

Параметры блока Transfer Fcn зададим согласно передаточной функции (2.2), а остальные — по умолчанию.

Параметры блока Discrete Filter зададим согласно передаточной функции (2.3), период дискретизации **Sample Time:** –1, что соответствует наследованию периода дискретизации от блока Step1, остальные — по умолчанию. Подробнее параметры блока Discrete Filter будут рассматриваться в *разд. 12.2.1*.

Параметры блока Scope будут рассматриваться в *разд. 5.2.3*. Пока для каждого из блоков Scope и Scope1 установим следующие параметры:

- Number of axis (Количество осей) равным 2 для отображения входного и выходного сигналов на двух независимых графиках;
- У-тіп и У-тах (Границы осей ординат) равными –1 и 2 соответственно.
- 4. Соединение блоков.

Соединение блоков можно выполнять вручную или автоматически.

При *ручном* соединении блоков нужно подвести курсор к выходу соединяемого блока, и после того как появится крестик, фиксирующий местоположение выхода, при нажатой левой кнопке мыши провести прямую линию к входу соответствующего блока. При проведении ломаной линии кнопку мыши в точках изгиба нужно отпускать. При нажатой кнопке мыши местоположение входа (или узла) автоматически фиксируется сдвоенным крестиком.

При первом соединении автоматически выводится окно подсказки Automatic Block Connection Tip (Подсказка по автоматическому соединению блоков), которое можно закрыть.

При *автоматическом* соединении блоков выделите блок (щелчком левой кнопки мыши), содержащий вход, и при нажатой клавише <Ctrl> выделите блок, содержащий выход, после чего автоматически будет выполнено соединение блоков.

5. Сохранение содержимого окна S-моделей.

Сохранение содержимого окна S-моделей в файл выполняется по команде меню File | Save (Файл | Сохранить). В открывающемся окне Save as (Сохранить как) файлу S-моделей присваивается имя с расширением mdl.

*Имя* файла может состоять из любой последовательности латинских букв, цифр и символа подчеркивания, начинающейся с буквы.

Файлы S-моделей с расширением mdl должны храниться в папке work или в папке пользователя, вложенной в work.

Создание собственной папки в папке work выполняется с помощью контекстного меню в окне Current Folder (см. рис. 1.1) окна MATLAB, а сохранение пути к ней — по команде контекстного меню Add Path (Добавить путь) или команде главного меню File | Set Path (Файл | Установить путь).

Нашему файлу, хранящему S-модели линейных аналоговой и дискретной систем с соединением блоков, присвоим имя analog\_discrete (рис. 2.4).



Рис. 2.4. Окно analog\_discrete с S-моделями аналоговой и дискретной систем

#### 2.3. Редактирование S-модели системы

Редактирование S-модели системы выполняется в окне S-моделей.

Открытие окна S-моделей с данными из файла выполняется двойным щелчком левой кнопки мыши на его пиктограмме или имени в окне Current Folder или перетаскиванием пиктограммы файла из окна Current Folder в окно Command Window (см. рис. 1.1).

Операции по редактированию S-модели системы удобно объединить в две группы:

- 1. Операции с компонентами S-модели системы.
- 2. Операции по оформлению S-модели системы.

#### 2.3.1. Операции с компонентами S-модели системы

Компонентами S-модели системы будем называть блоки и соединения.

Группой компонентов будем называть их компактную комбинацию.

Группой блоков будем называть их компактную комбинацию без соединений.

Основные операции в окне S-моделей с компонентами, группами компонентов и блоков — *операндами* — представлены в табл. 2.1. Как обычно, основные команды главного меню дублируются в контекстном меню и кнопками на панели инструментов.

Редактирование компонента (группы компонентов или блоков) выполняется после его *выделения* щелчком левой кнопки мыши на его изображении.

*Группа компонентов* или *блоков* выделяется обведением их курсором при нажатой левой кнопке мыши.

*Все содержимое окна S-моделей* выделяется по команде меню Edit | Select All (Редактирование | Выделить все).

Соответствующее выделение отображается появлением *маркеров* по углам блоков и в изломах соединений.

Отмена любого выделения выполняется щелчком левой кнопки мыши на свободном поле окна S-моделей.

N⁰	Операция	Операнд	Выполнение
1.	Удаление	Компонент Группа компонентов или блоков	Нажатием клавиши <b>Delete</b> (Удалить) или по команде меню <b>Edit   Delete</b> (Редактирование   Удалить)
2.	Перемещение	Блок Группа компонентов или блоков	Курсором при нажатой левой кнопке мыши
3.	Дублирование	Блок Группа компонентов или блоков	Курсором при нажатой правой кнопке мыши. Дублируемые блоки автоматически именуются с номерами 1, 2,
4.	Вставка в соединение	Блок с одним вхо- дом/выходом	Перетаскиванием блока из библиотеки Simulink (или дублированием блока) и установкой его на соединение
5.	Удаление (вырезание) в буфер MATLAB	Компонент Группа компонентов или блоков	По команде меню Edit   Cut (Редактирование   Вырезать)
6.	Копирование в буфер MATLAB	Компонент Группа компонентов или блоков	По команде меню Edit   Сору (Редактирование   Копировать)
7.	Вставка из буфера MATLAB	Компонент Группа компонентов или блоков	По команде меню <b>Edit   Paste</b> (Редактирование   Вставить)
8.	Копирование в буфер Windows	Содержимое окна S-моделей	По команде меню Edit   Copy model to Clipboard (Редактирование   Копирование в буфер обмена). В качестве графического изображения доступно для различных Windows-приложений: Word, Visio и др.

Таблица 2.1. Основные операции с компонентами, группами компонентов и блоков
#### Окончание табл. 2.1

N⁰	Операция	Операнд	Выполнение
9.	Отмена опера- ции	Компонент Группа компо- нентов или блоков	По команде меню <b>Edit   Undo</b> <sup>1</sup> (Редактирование   Возврат). Не отменяет операции изменения <i>размера</i>
10.	Восстановле- ние отменен- ной операции	Компонент Группа компо- нентов или блоков	По команде меню <b>Edit   Redo</b> (Редактирование   Повтор)
11.	Вращение на 90° по часовой стрелке	Блок Группа блоков	По команде меню Format   Rotate Block (Редакти- рование   Вращение блока)
12.	Зеркальное отображение	Блок Группа блоков	По команде меню Format   Flip Block (Формат   Зер- кальное отображение блока)
13.	Изменение размера	Блок	Курсором, подведенным к одному из маркеров бло- ка, после появления двунаправленной диагональ- ной стрелки при нажатой левой кнопке мыши.
			Размер имени блока не меняется
14.	Добавление тени	Блок Группа компо- нентов или блоков	По команде меню Format   Show Drop Shadow (Формат   Показ отбрасываемой тени)
15.	Удаление тени	Блок Группа компо- нентов или блоков	По команде меню Format   Hide Drop Shadow (Формат   Скрытие отбрасываемой тени)
16.	Изменение конфигурации линии	Соединение	Курсором, подведенным к одному из маркеров со- единения, после автоматического обведения мар- кера кружком при нажатой левой кнопке мыши
17.	Смещение узла	Соединение	Курсором, подведенным к узлу на горизонтальной линии, после автоматического обведения маркера узла кружком при нажатой левой кнопке мыши
18.	Изменение толщины линий соединения	Соединение	По команде меню Format   Port Signal/Display   Wide Nonscalar Lines (Формат   Порт/Изображение сигнала   Толщина нескалярных линий).
			Эта операция доступна для соединений, по которым передаются данные — векторы и матрицы (см. разд. 4.2).
			При закрытии окна S-моделей утолщение соедине- ний не сохраняется

<sup>&</sup>lt;sup>1</sup> В списке команд отменяемая или восстанавливаемая после отмены операция конкретизируется: Undo Delete, Redo Move и т. д. Если операция не может быть выполнена, то активизируется команда Can't Undo (Невозможно отменить).

## 2.3.2. Операции по оформлению S-моделей системы

Операции по оформлению S-моделей системы можно, в свою очередь, объединить в следующие три группы:

1. Операции с текстом.

Основные операции с текстом представлены в табл. 2.2.

Редактирование текста, в том числе имени блока, выполняется после его активации щелчком левой кнопки мыши в поле текста.

При вводе текста допускается использование кириллицы в версиях, начиная с Simulink 6, однако оно не в полной мере отслеживается, поэтому, во избежание сбоев, рекомендуется использовать буквы латинского алфавита.

Редактирование текстов в соответствующем поле ввода выполняется теми же средствами, что и в Word. Для выхода из режима редактирования текста следует щелкнуть левой кнопкой мыши на свободном поле окна S-моделей.

N⁰	Операция	Выполнение
1.	Ввод текста	После активизации поля текста на свободном месте двойным щелчком левой кнопки мыши
2.	Перемещение текста	Курсором при нажатой левой кнопке мыши
3.	Дублирование текста	Курсором при нажатой правой кнопке мыши
4.	Выбор типа и размера шрифта для текста	По команде меню Format   Font (Формат   Шрифт) в окне Set Font (Шрифт)
5.	Выравнивание текста	По команде меню Format   Text Alignment (Формат   Выравни- вание текста)
6.	Изменение имени блока	В поле имени блока
7.	Перемещение имени блока	При выделенном блоке при размещении имени под/над блоком по команде меню Format   Flip Name (Формат   Перенос имени)
8.	Скрытие имени блока	При выделенном блоке по команде меню Format   Hide Name (Формат   Скрытие имени)
9.	Восстановление скры- того имени блока	При выделенном блоке по команде меню Format   Show Name (Формат   Показ имени)
10.	Создание метки (име- ни) сигнала на соеди-	После активизации поля текста двойным щелчком левой кнопки мыши точно на соединении.
	нении	Созданная метка (имя) сигнала автоматически дублируется на всех соединениях, где присутствует помечаемый сигнал.
		Если этого не происходит, то дублировать метки можно вручную путем перемещения созданной метки при нажатой клавише <ctrl>.</ctrl>

Таблица 2.2. Операции с текстом

#### Окончание табл. 2.2

N⁰	Операция	Выполнение
		При удалении метки удаляются ее дубликаты.
		Метки автоматически перемещаются с перемещением соединения
11.	Распространение ме- ток (имен) сигналов на соединениях при их объединении и разде- лении <sup>1</sup>	После создания меток (имен) сигналов на том соединении, где пользователь хочет вывести распространение меток, в поле текста соединения, которое активизируется двойным щелчком левой кнопки мыши <i>точно на соединении</i> , ставятся треугольные скобки <>, и в них будет автоматически выведен список меток (имен) объединяемых сигналов, перечисленных через запятую, или метки (имена) разделяемых сигналов

На рис. 2.5 изображены S-модели систем (см. рис. 2.4), где выполнено следующее редактирование: перемещены блоки; изменены конфигурации линий; добавлены заголовки; изменены имена блоков; созданы метки входных сигналов. После редактирования файл сохранен под именем analog\_discrete\_1.



Рис. 2.5. Окно analog\_discrete\_1 с S-моделями аналоговой и дискретной систем после редактирования

2. Операции с цветом.

Основные операции с цветом представлены в табл. 2.3.

<sup>&</sup>lt;sup>1</sup> В блоках Мих, Demux, Bus Creator и Bus Selector, о которых речь пойдет в *разо.* 7.1.

Таблица 2.3. Операции с цветом

N⁰	Операция	Выполнение
1.	Выбор цвета линий для блока	При выделенном блоке по команде меню Format   Fore- ground Color (Формат   Цвет переднего плана)
2.	Выбор цвета фона блока	При выделенном блоке по команде меню Format   Back Color (Формат   Цвет заднего плана)
3.	Выбор цвета фона окна S- моделей	При выделенном блоке по команде меню Format   Screen Color (Формат   Цвет экрана)
4.	Выделение цветом всех блоков (и связанных с ними соединений), у которых па- раметр <b>Sample Time</b> боль- ше нуля	По команде меню Format   Port Signal/Display   Sample Time (Формат   Порт/Изображение сигнала   Интервал времени). Параметр Sample Time будет рассматриваться в <i>разд. 4.2</i>

3. Операции по выводу дополнительной информации.

Основные операции по выводу дополнительной информации представлены в табл. 2.4.

Nº	Операция	Выполнение
1.	Вывод типа данных	По команде меню Format   Port/Signal Display   Port Data Types (Формат   Порт/Изображение сигнала   Типы данных порта) тип данных выводится на выходе блоков.
		При закрытии окна S-моделей отображение типа данных не сохраняется
2.	Вывод размерности данных	По команде меню Format   Port/Signal Display   Signal Di- mensions (Формат   Порт/Изображение сигнала   Размерность сигнала).
		Операция доступна для сигналов в виде данных — векторов и матриц.
		При закрытии окна S-моделей отображение размерности дан- ных не сохраняется
3.	Вывод индексов блоков в порядке их исполне-	По команде меню Format   Block Display   Selected Order (Формат   Изображение блока   Выбранный порядок).
	ния	Индекс блока (Block ID) задается двумя цифрами, разделен- ными двоеточием — s:b, где цифра s — номер подсистемы <sup>1</sup> (subsystem), a b — номер блока (block) в подсистеме.
		Нумерация ведется, начиная с <i>нуля</i> .
		Если в окне S-моделей содержится несколько S-моделей, то нумерация блоков будет сквозной.
		При закрытии окна S-моделей номера не сохраняются

Таблица 2.4. Операции по выводу дополнительной информации

<sup>&</sup>lt;sup>1</sup> О подсистемах речь пойдет далее в *гл. 8.* Если S-модель не содержит подсистем, то первой цифрой всегда будет 0.

На рис. 2.6 изображены S-модели систем (см. рис. 2.5) с выводом дополнительной информации о типе данных и индексах блоков в порядке их исполнения.



Рис. 2.6. Окно analog\_discrete\_1 с S-моделями аналоговой и дискретной систем и выводом дополнительной информации

Сообщение об ошибке при создании S-модели системы выводится в автоматически открывающемся окне с комментариями об ошибке и одновременно отображается визуально в окне S-моделей.

Более подробные сведения о создании и редактировании S-моделей можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  Getting Started  $\rightarrow$  Creating a Simulink Model (Simulink  $\rightarrow$  Bведение  $\rightarrow$  Cоздание модели Simulink).

## 2.4. Исследование S-модели

Окно S-моделей предоставляет удобные средства для исследования S-модели системы, в том числе средства оперативного просмотра и редактирования параметров блоков и параметров настройки, о которых речь пойдет в *гл. 3*.

Исследование S-модели выполняется с помощью команды меню View | Model Explorer (Вид | Исследователь модели).

При обращении к этой команде открывается окно Model Explorer. На рис. 2.7 оно показано для созданного ранее файла S-моделей analog\_discrete\_1 (см. рис. 2.5). (К окну Model Explorer можно также обратиться по команде контекстного меню Explore, выделив в S-модели интересующий блок.)



Рис. 2.7. Окно Model Explorer для файла S-моделей analog\_discrete\_1

Окно Model Explorer содержит три панели:

- □ Model Hierarchy (Иерархия модели) корневой каталог (Simulink Root), представленный в виде дерева;
- Contents of (Содержимое) содержимое выделенного на панели Model Hierarchy раздела;
- □ Model Properties (Свойства модели) информация о разделе, выделенном на панели Model Hierarchy или Contents of.



## Глава 3

# Технология моделирования системы

После создания S-модели системы (см. гл. 2) приступают к моделированию системы, основные этапы технологии которого включают в себя:

- настройку S-модели системы;
- настройку обмена данными S-модели системы с рабочим пространством памяти Workspace;
- □ настройку диагностики моделирования;
- моделирование системы запуск, пауза и останов.

Моделирование сложных систем предполагает их предварительную отладку, средства которой будут рассматриваться в *гл.* 10.

## 3.1. Настройка S-модели системы

Для знакомства с настройкой S-модели системы откроем безымянное окно untitled *(см. разд. 2.2)* и далее, по команде меню Simulation | Configuration Parameters (Моделирование | Параметры конфигурации), — окно Configuration Parameters (Параметры конфигурации) (рис. 3.1).

Компоненты настройки S-модели системы в окне Configuration Parameters представлены на левой панели Select (Выбор) в виде дерева.

Настройка S-модели системы заключается в задании параметров моделирования в компоненте Solver (Решатель), выделенной на левой панели Select по умолчанию.

Параметры моделирования задаются на правой панели, разделенной на четыре группы:

- 1. В группе Simulation time (Время моделирования) указывается интервал моделирования, задаваемый его границами:
  - в поле ввода Start time (Начальное время) начальное время моделирования в секундах, по умолчанию равное нулю;

Select:	Simulation time					
Solver Data Import/Export	Start time: 0.0			Stop time: 10.0		
Optimization     Optimization     Optimization     Option     Option	Solver options Type: Max step size: Min step size: Initial step size: Number of consect Tasking and sample Tasking mode for p Automatically	Variable-step auto auto auto utive min steps: e time options eriodic sample times: handle rate transition for data trans	Score	Nver: elative tolerance: osolute tolerance: hape preservation: Auto	ode45 (Dormand-Prince) 1e-3 auto Disable all 1	
Symbols Custom Code	Zero-crossing optic	ons				
Debug	Zero-crossing cont	rol: Use local settings		Algorithm:	Nonadaptive	~
∃-HDL Coder	Time tolerance:	10*128*eps		Signal threshold:	auto	
Global Settings Test Bench EDA Tool Scripts	Number of consecu	tive zero crossings:			1000	

Рис. 3.1. Окно Configuration Parameters при обращении к компоненте Solver для решателя с переменным шагом моделирования (Variable-step)

• в поле ввода **Stop time** (Конечное время) — конечное время моделирования в секундах.

*Конечное* время выбирается, исходя из конкретной задачи, по умолчанию оно равно 10 с.

Если конечное время заранее неизвестно, то можно установить значение inf (машинную бесконечность), а затем, после запуска моделирования, остановить процесс моделирования или сделать в нем паузу (см. разд. 3.4).

Интервал моделирования не отражает реального времени и зависит от многих факторов: сложности S-модели, шага моделирования, быстродействия компьютера и др.

Если начальное и конечное время заданы одинаковыми, то будет выполнен *один* шаг моделирования (см. далее в группе Solver options).

На практике, как правило, начальное время моделирования равно нулю, поэтому конечное время устанавливается непосредственно в поле ввода **Stop time** *окна S-моделей* и автоматически дублируется в одноименном поле ввода окна **Configuration Parameters**.

2. В группе Solver options (Параметры решателя) выбираются параметры решателя. При моделировании аналоговых систем, соотношение вход/выход которых описывается обыкновенным дифференциальным уравнением (ОДУ<sup>1</sup>), решателем (Solver) называют функцию MATLAB, реализующую метод численного интегрирования ОДУ [22].

При моделировании *дискретных* систем, соотношение вход/выход которых описывается *разностным уравнением*, непосредственно описывающим *алгоритм* вычисления реакции системы [22], *решателем* называют средства Simulink (скрытые от пользователя), реализующие данный алгоритм.

К параметрам решателя относятся:

• **Туре** (Тип) — тип решателя;

Различают *два* типа решателей — с *фиксированным* (Fixed-step) и *переменным* (Variable-step) шагом моделирования, который выбирается в раскрывающемся списке **Туре**.

При моделировании *аналоговых* систем принципиально можно выбирать любой тип решателя. Предпочтительным, однако, является выбор решателя с переменным шагом моделирования (Variable-step) (по умолчанию). Это обеспечит автоматическое изменение шага моделирования в зависимости от характера поведения решений ОДУ. Шаг будет выбираться меньшим для "быстрых" решений и бо́льшим — для "медленных"<sup>2</sup>.

При моделировании *дискретных* систем, как правило, выбирают решатель с *фиксированным* шагом моделирования (Fixed-step) (рис. 3.2).

Выбор следующих параметров моделирования зависит от *типа* решателя.

При выборе решателя с *переменным* шагом моделирования (Variable-step) для *аналоговых систем* задаются следующие параметры (см. рис. 3.1):

- Max step size (Величина максимального шага) максимальный шаг моделирования в секундах;
- Min step size (Величина минимального шага) минимальный шаг моделирования в секундах;
- Initial step size (Величина начального шага) начальный шаг моделирования в секундах;
- Number of consecutive min steps (Количество последовательных минимальных шагов) — допустимое количество следующих друг за другом уменьшений минимального шага моделирования Min step size.

Если в процессе моделирования указанное допустимое количество будет превышено, выведется сообщение об ошибке или предупреждение;

<sup>&</sup>lt;sup>1</sup> ODE (Ordinary Differential Equations).

<sup>&</sup>lt;sup>2</sup> Быстро или медленно изменяющихся решений (функций) или их фрагментов (возмущений). Это особенно важно для так называемых жестких систем, которые трудно распознать заранее.

🐞 Configuration Parameter	s: untitled/Configuration (Active)			×
Select:	Simulation time			^
Select: - Solver - Data Import/Export - Optimization - Diagnostics - Data Validity - Type Conversion - Connectivity - Compatibility - Model Referencing - Saving - Hardware Implementation - Model Referencing - Saving - Compatibility - Compatibility - Saving - Savi	Simulation time Start time: 0.0 Solver options Type: Fixed-step Fixed-step size (fundamental sample time): auto Tasking and sample time options Periodic sample time constraint: Tasking mode for periodic sample times: Automatically handle rate transition for data transfer Higher priority value indicates higher task priority	Stop time: 10.0  Solver: discrete (no continuous states)  Unconstrained Auto		
0		OK <u>C</u> ancel <u>H</u> elp	Apply	

Рис. 3.2. Окно Configuration Parameters для решателя с фиксированным шагом моделирования (Fixed-step)

 Relative tolerance (Относительная погрешность) — относительная погрешность метода численного интегрирования ОДУ.

Относительная методическая погрешность соответствует локальной ошибке усечения d<sub>i</sub> на *i*-м шаге интегрирования [22]:

$$d_i = |x_{i+1} - x_i(t_{i+1})|,$$

где  $x_{i+1}$  — решение (последовательно вычисляемые значения искомой функции), а  $x_i(t_{i+1})$  — теоретически точное решение в предположении, что локальные ошибки усечения на предыдущих шагах отсутствуют, т. е. решение, вычисленное на предыдущем шаге, совпадает с теоретически точным для этого шага —  $x_i = x_i(t_i)$ .

По умолчанию локальная ошибка усечения равна 1е-3;

• Absolute tolerance (Абсолютная погрешность) — абсолютная погрешность метода численного интегрирования ОДУ.

Абсолютная методическая погрешность соответствует *глобальной ошибке усечения*  $e_i$  на интервале  $[t_0, t_i]$ , равной разности между вычисленным и точным решением:

$$e_i = \left| x_i - x(t_i) \right|.$$

По умолчанию (auto) глобальная ошибка усечения равна 1е-6.

Абсолютная ошибка контролируется, и если в процессе моделирования наблюдается ее превышение, автоматически уменьшается шаг интегрирования;

• Shape preservation (Сохранение формы) — сохранение переменных состояний с возможными значениями Disable all (Запретить все) или Enable all (Разрешить все).

Переменными состояниями (States) S-модели аналоговой системы называют текущие значения выходного сигнала на каждом шаге моделирования, зависящие от его предыдущих значений и определяющие последующие значения.

При выборе Disable all (по умолчанию) переменные состояния не сохраняются, а при Enable all — сохраняются и могут быть использованы для автоматического повышения точности при наличии "быстрых" решений или возмущений.

При выборе решателя с *фиксированным* шагом моделирования (Fixed-step) (см. рис. 3.2) в поле ввода **Fixed-step size** (Размер фиксированного шага) *шаг моделирования* задается или устанавливается по умолчанию (auto) равным значению (**Stop time – Start time**)/50 с;

• Solver — решатель.

В раскрывающемся списке Solver решатели с *переменным* шагом моделирования (Variable-step) для *аналоговых систем* (см. рис. 3.1) представлены функциями MATLAB (табл. 3.1), реализующими методы численного интегрирования ОДУ [22].

При моделировании *дискретных* систем выбирается решатель с обобщенным именем discrete (no continuous states).

Функция (решатель)	Метод численного интегрирования	Для каких систем ОДУ рекомен- дуется
ode45	Одношаговый явный метод Рунге— Кутты 4-го порядка	Для не жестких систем. Классический метод, рекоменду- ечмый для первой пробы решения
ode23	Одношаговый явный метод Рунге— Кутты 2-го порядка	Для не жестких систем. Проще, чем ode45, но обеспечива- ет меньшую точность
ode113	Многошаговый метод предсказания- коррекции Адамса—Башфорта— Маултона	Для не жестких систем. Обеспечивает высокую точность

Таблица 3.1. Функции МАТLAB для решения систем ОДУ

Окончание табл. 3.1

Функция (решатель)	Метод численного интегрирования	Для каких систем ОДУ рекомен- дуется
ode15s	Многошаговый метод ФДН <sup>1</sup>	Для жестких систем.
		Рекомендуется для второй попытки решения, если ode45 дает сбой и есть подозрение, что система жесткая
ode23s	Одношаговый неявный метод Рунге—	Для жестких систем.
	Кутты—Розенброка 2-го порядка	Проще, чем ode15s, но обеспечи- вает меньшую точность
ode23t	Одношаговый неявный метод трапеций	Для умеренно жестких систем.
	2-го порядка	Обеспечивает высокую точность
ode23tb	Комбинация метода трапеций и ФДН 2-	Для жестких систем.
	го порядка	Проще, чем ode15s, но обеспечи- вает меньшую точность

3. В группе **Tasking and sample time options** (Параметры управления задачами и интервалом времени) задаются параметры, контролирующие интервал времени и процесс моделирования.

Для *дискретных систем* выбирается решатель discrete (no continuous states) с *фиксированным* шагом моделирования (Fixed-step) и задаются следующие параметры (см. рис. 3.2):

- Periodic sample time constraint (Ограничение на периодичность интервала времени) контроль ограничения интервала времени Sample time (о параметре Sample time речь пойдет далее в *разд. 4.2.1*) в блоках S-модели системы с возможными значениями:
  - Unconstrained (Ограничений нет) для S-моделей систем с недостаточно строгой синхронизацией по интервалу времени Sample time, не влияющей на конечный результат (по умолчанию);
  - Ensure sample time independent (Обеспечить интервал времени независимым) — для S-моделей систем, в которых интервал времени Sample time наследуется от соседних блоков;
  - Specified (Указанный) для S-моделей систем с приоритетными значениями интервалов Sample time в многозадачных режимах, о которых речь пойдет в следующем параметре;
- Tasking mode for periodic sample times (Режим управления задачами для периодического интервала времени) контроль моделирования в зависи-

<sup>&</sup>lt;sup>1</sup> Формула дифференцирования назад.

мости от значений интервала времени Sample time с возможными значениями:

- SingleTasking (Однозадачный) для S-моделей систем с одинаковыми значениями интервала времени Sample time в блоках или с недостаточно строгой синхронизацией;
- MultiTasking (Многозадачный) для S-моделей систем с различными с значениями интервала времени Sample time в блоках при необходимости их синхронизации;
- Auto (по умолчанию) для автоматического перехода в многозадачный режим (MultiTasking) при обнаружении в S-модели системы блоков с различными значениями интервала времени Sample time;
- Automatically handle rate transition for data transfer (Автоматическое управление преобразованием скорости передачи данных) флаг управления преобразованием периода дискретизации (скорости передачи данных) в *мно-госкоростных системах* посредством автоматической вставки скрытых блоков Rate Transition (Преобразование скорости) из группы блоков Signal Attributes (Атрибуты сигнала) между блоками с различными периодами дискретизации (различными скоростями).

При установке флага появляется раскрывающийся список с запросом о вставке скрытых блоков Rate Transition посредством выбора одного из значений:

- Whenever possible (Везде, где возможно) (по умолчанию);
- Always (Везде);
- Never (minimum delay) (Нигде (минимальная задержка));
- **Higher priority value indicates higher task priority** (Значение более высокого приоритета указывает на задачу с более высоким приоритетом) флаг управления приоритетом задач в системах, работающих в реальном времени, при реализации асинхронной передачи данных.

При выборе решателя с *переменным* шагом моделирования (Variable-step) для *аналоговых систем* в группе **Tasking and sample time options** могут использоваться те же флаги.

4. В группе **Zero crossing options** (Параметры для пересечения нуля) выбираются параметры метода "пересечение нуля".

При выборе решателя с *переменным* шагом моделирования (Variable-step) для *аналоговых систем* может использоваться метод "пересечение нуля", обеспечивающий контроль монотонности сигнала на каждом шаге моделирования и позволяющий максимально точно определять моменты резкого изменения значений сигнала (скачков).

В группе **Zero crossing options**, которая активизируется при выборе *переменного* шага моделирования (Variable-step), задаются или выбираются по умолчанию следующие параметры (см. рис. 3.1):

- Zero crossing control (Контроль пересечения нуля) контроль точки пересечения нуля с возможными значениями:
  - Use local setting (Использовать локальные установки) при определении пересечений нуля по принципу от блока к блоку (по умолчанию);
  - Enable all (Доступны все) при определении пересечений нуля во всех блоках S-модели системы;
  - Disable all (Недоступны все) при отсутствии контроля за пересечением нуля;
- **Time tolerance** (Погрешность времени) показатель точности, контролирующий близость последовательных точек пересечения нуля, по умолчанию равный 10\*128\*eps, где eps = 2.2204e–016;
- Algorithm (Алгоритм) алгоритм для определения местоположения пересечений нуля, неадаптивный (Nonadaptive) или адаптивный (Adaptive);
- Signal threshold (Порог сигнала) зона нечувствительности, в которой скачки сигнала будут автоматически считаться точками пересечения нуля; активизируется при выборе Algorithm: Adaptive.

Может задаваться вещественным числом, большим либо равным нулю. По умолчанию (auto) определяется автоматически;

• Number of consecutive zero crossing (Количество последовательных точек пересечения нуля) — поле ввода для количества последовательных точек пересечения нуля, при превышении которого будет выдаваться предупреждение об ошибке.

В Simulink также предоставлены средства оперативного просмотра и редактирования параметров настройки S-модели системы в окне Model Explorer (см. разд. 2.4).

Более подробную информацию можно получить в справочной системе MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  User Guide  $\rightarrow$  Simulink Graphical User Interface  $\rightarrow$  Configuration Parameters Dialog Box  $\rightarrow$  Solver Pane (Simulink  $\rightarrow$  Руководство пользователя  $\rightarrow$  GUI Simulink  $\rightarrow$  Диалоговый блок конфигурации параметров  $\rightarrow$  Панель решателя).

## 3.2. Настройка обмена данными S-модели системы с рабочим пространством памяти Workspace

Настройка обмена данными S-модели системы с рабочим пространством памяти Workspace заключается в задании *параметров взаимодействия с Workspace* с целью экспорта/импорта данных в процессе моделирования и производится при обращении к компоненте **Data Import/Export** (Импорт/Экспорт данных) на левой панели **Select** окна **Configuration Parameters** (рис. 3.3).

🍓 Configuration Parameters	s: untitled/Config	uration	(Active)			X
Select:	-Load from workspa	ce				^
Solver	Input:	[t, u]				
Optimization	Initial state:	×Initial				
Diagnostics     Sample Time     Data Validity	Save to workspace					
- Type Conversion	Time:	tout				
Connectivity	States:	xout				
	Output:	yout				
-Saving	Final states:	xFinal			Save complete SimState in final state	
	Signal logging:	logsout			Inspect signal logs when simulation is paused/stopped	-
	Save options  Limit data point Format: Return as single	s to last: e object:	1000 Array out	~	Decimation: 1	
0					QK <u>Cancel H</u> elp <u>Apply</u>	

Рис. 3.3. Окно Configuration Parameters при обращении к компоненте Data Import/Export для решателя с фиксированным шагом моделирования (Fixed-step)

Параметры взаимодействия с Workspace задаются на правой панели, разделенной на три группы:

- 1. В группе Load from workspace (Загрузка из Workspace) выбираются параметры, контролирующий импорт из Workspace *neped* моделированием:
  - Input (Входные данные) флаг, управляющий импортом данных из Workspace.

При установке флага **Input** в поле ввода указываются имена импортируемых из Workspace входных данных, которые могут быть представлены в следующих видах:

- матрица [t, u] (по умолчанию), где t вектор-столбец возрастающих значений времени, а u — матрица, векторы-столбцы которой соответствуют значениям сигналов: *i*-й столбец соответствует значениям *i*-го сигнала в моменты времени, заданные вектором времени t;
- структура (массив записей), заданная своим именем;
- вектор времени t;
- выражение MATLAB в апострофах, представляющее собой функцию времени, например 'exp(t)+3', где аргументу t в процессе моделирования будут автоматически присваиваться значения времени.

Данные, указанные в поле ввода **Input**, импортируются из Workspace с помощью блока In (см. разд. 4.2.11).

Импорт данных может также осуществляться с помощью блока From Workspace (см. разд. 4.2.9) или блока Signal From Workspace (см. разд. 11.2.2.7). В этом случае настройка обмена с Workspace выполняется посредством установки параметров блока;

• Initial state (Начальное состояние) — флаг, управляющий импортом *начального состояния* (начальных условий) из Workspace.

При установке флага начальное состояние импортируется из Workspace также с помощью блока In.

- 2. В группе **Save to workspace** (Сохранение в Workspace) выбираются параметры, управляющие экспортом данных в Workspace *после* моделирования:
  - **Time** (Время) флаг, управляющий экспортом в Workspace вектора значений времени.

При установке флага **Time** (по умолчанию) в поле ввода указывается имя вектора (по умолчанию tout) с экспортируемыми значениями времени;

• States (Переменные состояния) — флаг, управляющий экспортом переменных состояний.

При установке флага **States** в поле ввода указывается имя переменной (по умолчанию xout) для экспортируемых в Workspace переменных состояний S-модели системы.

Вид представления экспортируемых переменных состояний указывается в поле ввода Format группы Save options (см. далее);

• **Output** (Выходные данные) — флаг, управляющий экспортом выходных данных в Workspace.

При установке флага **Output** (по умолчанию) в поле ввода указывается имя переменной для экспортируемых в Workspace данных (по умолчанию yout).

Данные, указанные в поле ввода **Output**, экспортируются в Workspace с помощью блока Out *(см. разд. 5.2.2)*. Вид представления экспортируемых в Workspace данных указывается в поле ввода **Format** группы **Save options** *(см. далее)*.

Экспорт данных может также осуществляться с помощью блока To Workspace (см. разд. 5.2.8) или блока Signal To Workspace (см. разд. 11.2.1). В этом случае настройка обмена с Workspace (экспорт) выполняется посредством установки параметров блока;

• Final states (Конечные состояния) — флаг, управляющий экспортом в Workspace переменных состояний на последнем шаге моделирования.

При установке флага Final states в поле ввода указывается имя переменной для экспортируемых в Workspace конечных состояний (по умолчанию xFinal) и активизируется флаг Save complete SimState in final state (Сохранить полностью состояние моделирования в последнем положении), управляющий сохранением текущего состояния моделирования;

- Signal logging (Регистрация сигнала) флаг, управляющий регистрацией сигнала в Workspace в процессе моделирования.
- При установке флага (по умолчанию) в поле ввода Signal logging указывается имя регистрируемого в Workspace сигнала (по умолчанию logsout) и активизируется флаг Inspect signal logs when simulation is paused/stopped (Проверка регистрации сигнала по завершении моделирования или во время паузы), контролирующий регистрацию сигнала в Workspace в процессе моделирования и по его завершении.
- 3. В группе Save options (Сохранение параметров) определяются следующие параметры:
  - Limit data points to last (Граничное число данных для сохранения) флаг, управляющий числом и видом экспортируемых данных.

При установке флага Limit data points to last (по умолчанию) в поле ввода указывается граничное число значений (по умолчанию 1000) для экспортируемых в Workspace данных (длина вектора для сигнала);

• **Decimation** (Децимация) — коэффициент децимации (прореживания) для экспортируемого в Workspace сигнала — положительное целое число.

При **Decimation:** 1 (по умолчанию) экспортируются все значения (без прореживания), при **Decimation:** 2 — каждое второе значение и т. д.;

- Format (Формат) вид представления экспортируемых в Workspace данных. Экспортируемые в Workspace данные могут представляться в одном из трех видов:
  - Аггау (Массив) матрица;
  - Structure (Структура) структура (массив записей) с пустым полем time;
  - Structure with time (Структура с временем) структура (массив записей) с полем time.

Подробнее эти виды представления данных будут рассматриваться далее для блока To Workspace (см. разд. 5.2.8);

• **Output options** (Параметры выхода) — дополнительный параметр для экспортируемых в Workspace данных.

Используется только для решателей с переменным шагом моделирования (Variable-step) и может принимать значения:

• **Refine output** (Скорректированный вывод) (по умолчанию).

При выборе этого значения в поле ввода **Refine factor** (Коэффициент коррекции) можно задать дополнительные точки (значения времени) внутри шага моделирования, в которых будут выводиться экспортируемые в Workspace данные.

При **Refine factor:** 1 (по умолчанию) дополнительные точки не задаются, при **Refine factor:** 2 внутри шага моделирования выводится одно дополнительное значение и т. д.;

• Produce additional output (Произвести дополнительный вывод).

При выборе этого значения в поле ввода **Refine factor** можно задать дополнительные точки (значения времени), в которых будут выводиться экспортируемые в Workspace данные, в виде вектора в квадратных скобках;

Produce specified output only (Произвести только определенный вывод).

При выборе этого значения в поле ввода **Refine factor** можно задать произвольные точки (значения времени) в виде вектора в квадратных скобках, и экспортируемые в Workspace данные будут выводиться только в этих точках.

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  User Guide  $\rightarrow$  Simulink Graphical User Interface  $\rightarrow$  Configuration Parameters Dialog Box  $\rightarrow$  Data Import/Export Pane (Simulink  $\rightarrow$  Руководство пользователя  $\rightarrow$  GUI Simulink  $\rightarrow$  Диалоговый блок конфигурации параметров  $\rightarrow$  Панель Импорта/Экспорта данных).

## 3.3. Настройка диагностики моделирования

Настройка диагностики моделирования заключается в задании *параметров диагностики* и производится при обращении к компоненте Diagnostics (Диагностика) на левой панели **Select** окна **Configuration Parameters** и ее вложенным компонентам (рис. 3.4):

□ Sample Time (Интервал времени);

□ Data Validity (Достоверность данных);

□ Туре Conversion (Преобразование типов данных);

□ Connectivity (Соединимость);

- □ Compatibility (Совместимость);
- □ Model Referencing (Ссылки на модель);

□ Saving (Сохранение).

Параметры диагностики представляют собой выбираемую пользователем реакцию на диагноз — ошибку или проблемную ситуацию при моделировании системы, а также противоречие между параметрами блоков S-модели системы и параметрами настройки.

🖏 Configuration Parameter	s: untitled/Configuration (Active)			×
Select:	Solver		-	>
Solver Data Import/Export	Algebraic loop:	warning	~	
Optimization	Minimize algebraic loop:	warning	~	
Diagnostics     Sample Time	Block priority violation:	warning	~	
Data Validity	Min step size violation:	warning	~	
Type Conversion Connectivity	Sample hit time adjusting:	none	~	
Compatibility	Consecutive zero crossings violation:	error	~	
Saving	Unspecified inheritability of sample time:	warning	~	
Hardware Implementation Model Referencing	Solver data inconsistency:	none	~	
E-Simulation Target	Automatic solver parameter selection:	warning	~	1
Custom Code	Extraneous discrete derivative signals:	error	~	
E-Real-Time Workshop	State name clash:	warning	~	
Comments	SimState interface checksum mismatch:	warning	~	
Symbols Custom Code				
Debug				
E-HDL Coder				
Global Settings Test Bench				
EDA Tool Scripts				
				~
0		OK Cancel Help	3pply	

Рис. 3.4. Окно Configuration Parameters при обращении к компоненте Diagnostics

Со списком многочисленных параметров диагностики рекомендуется познакомиться самостоятельно с помощью справочной системы.

Реакция на диагноз представлена следующими стандартными вариантами:

none — игнорировать;

□ warning — выдать предупреждение и продолжить моделирование;

□ error — выдать сообщение об ошибке и остановить моделирование.

Предупреждение (warning) выводится в окне **Command Window**, а сообщение об ошибке (error) — в автоматически открывающемся окне **Simulation Diagnostics** (Диагностика моделирования) с именем S-модели системы и одновременно фиксируется визуально в окне S-моделей.

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  User Guide  $\rightarrow$  Simulink Graphical User Interface  $\rightarrow$  Configuration Parameters Dialog Box  $\rightarrow$  Diagnostics Pane (Simulink  $\rightarrow$  Руководство пользователя  $\rightarrow$  GUI Simulink  $\rightarrow$  Диалоговый блок конфигурации параметров  $\rightarrow$  Панель диагностики).

## 3.4. Моделирование системы — запуск, пауза и останов

Запуск моделирования системы осуществляется нажатием кнопки Start Simulation (Запуск моделирования) на панели инструментов окна S-моделей или по команде меню Simulation | Start.

Пауза в процессе моделирования системы осуществляется нажатием кнопки **Pause Simulation** (Пауза в моделировании) на панели инструментов или по команде меню **Simulation** | **Pause**, а продолжение процесса моделирования — нажатием на кнопку **Continue Simulation** (Продолжение моделирования) на панели инструментов или по команде меню **Simulation** | **Continue**.

*Останов* (завершение) моделирования системы осуществляется нажатием кнопки **Stop Simulation** (Останов моделирования) на панели инструментов или по команде меню **Simulation** | **Stop**.

## 3.5. Примеры моделирования простейших систем

Выполним моделирование двух простейших линейных систем, аналоговой и дискретной, созданных ранее и сохраненных в файле S-моделей analog\_discrete\_1.mdl (см. рис. 2.6).

Учитывая различный тип и специфику настройки S-моделей данных систем, сохраним S-модели аналоговой и дискретной систем в разных файлах: *аналоговую* систему — в файле analog.mdl (рис. 3.5), а *дискретную* — в файле discrete.mdl (рис. 3.6).



Рис. 3.5. Окно analog с S-моделью простейшей аналоговой системы и результаты моделирования



Рис. 3.6. Окно discrete с S-моделью простейшей дискретной системы и результаты моделирования

Выполним настройку S-модели аналоговой системы (см. разд. 3.1):

- □ в поле ввода **Stop time** окна S-моделей **analog** установим конечное время моделирования 5.0 (см. рис. 3.5);
- □ в поле ввода **Туре** выберем тип решателя с переменным шагом моделирования (Variable-step);

□ в поле ввода Solver выберем решатель ode45 (по умолчанию).

Остальные параметры оставим по умолчанию.

Запустим процесс моделирования *(см. разд. 3.4)*, результатами которого будут единичный скачок (воздействие) и переходная характеристика (реакция), представленные на рис. 3.5.

Выполним настройку S-модели дискретной системы (см. разд. 3.1):

- □ в поле ввода **Stop time** окна S-моделей **discrete** установим конечное время моделирования 5.0 (см. рис. 3.6);
- □ в поле ввода **Туре** выберем тип решателя с фиксированным шагом моделирования (Fixed-step);

□ в поле ввода Solver выберем решатель discrete (no continues states).

Остальные параметры оставим по умолчанию.

Запустим процесс моделирования, результатами которого будут цифровой единичный скачок (воздействие) и переходная характеристика (реакция), представленные на рис. 3.6.

## Глава 4



## Сигналы

Теперь мы переходим к рассмотрению состава S-модели системы (см. разд. 2.1). В этой главе приводятся S-модели входных сигналов, представленные в группе блоков **Sources** (Источники) библиотеки блоков Simulink.

## 4.1. Ѕ-модели сигналов

S-модели входных сигналов — блоки в группе блоков Sources (табл. 4.1), которые отвечают условиям:

□ не имеют входов<sup>1</sup>;

🗅 имеют один выход.

N⁰	Блок	Назначение
1.	Band-Limited White Noise (mask)	Генерирует нормальный белый шум с равномерной финитной СПМ <sup>2</sup> и заданным временем корреляции <sup>3</sup> для <i>аналоговых</i> систем
2.	Chirp Signal (mask)	Генерирует синусоиду с линейно возрастающей мгновенной частотой
3.	Clock	Выдает аналоговый сигнал текущего времени моделирования (с ша- гом моделирования)
4.	Constant	Выдает сигнал в виде данных — численных или логических
5.	Counter Free-Running (mask)	Формирует сигнал на выходе счетчика с коэффициентом пересчета 2 <sup>N</sup>

Таблица 4.1. Группа блоков Sources

<sup>&</sup>lt;sup>1</sup> Исключение составляют блоки Pulse Generator, Signal Generator, Sine Wave, в которых может автоматически создаваться вход для внешнего сигнала.

<sup>&</sup>lt;sup>2</sup> Спектральная плотность мощности (Power Spectral Density — PSD).

<sup>&</sup>lt;sup>3</sup> Белый шум теоретически определяется как случайный процесс с равномерной бесконечной СПМ. На практике допустимо говорить о белом шуме с финитной СПМ. Время корреляции такого шума будет отличным от нуля [19].

#### Окончание табл. 4.1

N⁰	Блок	Назначение
6.	Counter Li- mited (mask)	Формирует сигнал на выходе счетчика с произвольным коэффициен- том пересчета
7.	Digital Clock	Выдает дискретный сигнал текущего времени моделирования (с пе- риодом дискретизации)
8.	From File	Импортирует сигналы из файла данных с расширением mat
9.	From Work- space	Импортирует сигналы из рабочей области памяти Workspace
10.	Ground	"Заглушка" для неподключенного входа в блоке с несколькими входа- ми — нулевой сигнал. В отсутствие "заглушки" при моделировании выдается предупреждение
11.	In	Входной порт для сигналов, импортируемых из Workspace, и для под- систем
12.	Pulse Genera- tor	Генерирует периодические прямоугольные импульсы
13.	Ramp (mask)	Генерирует линейно изменяющийся сигнал
14.	Random Num- ber	Генерирует нормальный белый шум с заданными математическим ожиданием и дисперсией
15.	Repeating Se- quence (mask)	Генерирует периодический сигнал по заданным векторам времени и значений сигнала с линейной интерполяцией между соседними значе- ниями сигнала
16.	Repeating Se- quence Interpo- lated (mask)	Генерирует периодическую последовательность по заданным векто- рам времени и значений сигнала с различными способами вычисления отсчетов в промежуточных точках
17	Repeating Se- quence Stair (mask)	Генерирует периодическую последовательность по заданному вектору значений отсчетов и периоду дискретизации
18.	Signal Genera-	Генерирует четыре вида сигналов:
	tor	• гармонический (синусоида);
		• периодический трапецеидальный;
		<ul> <li>периодический пилообразный;</li> </ul>
		• случайный
19.	Sine Wave	Генерирует гармонический сигнал (синусоиду)
20.	Step	Генерирует скачок
21.	Uniform Ran- dom Number	Генерирует равномерный белый шум с заданными максимальным и минимальным значениями случайной величины

## 4.2. Параметры блоков

Параметры блоков задаются в окне **Parameters**, которое открывается двойным щелчком левой кнопки мыши на пиктограмме блока (см. разд. 2.2).

Рассмотрим параметры блоков из табл. 4.1. В качестве S-моделей средств анализа выходных сигналов будем использовать блоки Scope (Осциллограф) и Display (Дисплей для отображения численных данных), о которых подробнее речь пойдет в *гл. 5*.

## 4.2.1. Блок Band-Limited White Noise

Блок *Band-Limited White Noise* генерирует нормальный белый шум с равномерной финитной СПМ и заданным временем корреляции для *аналоговых* систем.

Для блока требуется задать следующие параметры (рис. 4.1):

□ Noise Power (Мощность шума) — значение равномерной СПМ, по умолчанию равно 0.1.

Здесь и далее значение некоторого численного параметра (скаляра) по умолчанию может указываться в квадратных скобках, которые, при желании, пользователь может снять. Подробнее о численных параметрах речь пойдет далее при описании параметра Interpret vector parameters as 1-D.

□ Sample time (Интервал времени) — интервал времени между соседними значениями сигнала в процессе моделирования.

Параметр Sample time необходимо согласовывать с шагом моделирования *(см. разд. 3.1).* 

В табл. 4.2 приведены типовые значения параметра **Sample time**, который будет присутствовать в списке параметров блоков группы Sources и других групп. Выбор конкретного значения определяется пользователем и зависит от назначения блока и решаемой задачи.

Для блока Band-Limited White Noise значение Sample time должно быть равно времени корреляции, по умолчанию 0.1 (с).

Значение	Тип сигнала	Интервал времени (в секундах)
0	Аналоговый	Символическое обозначение непрерывного времени
> 0	Дискретный	Период дискретизации Т
	Аналоговый	Интервал времени между соседними значе- ниями сигнала
	Данные	Период обновления в процессе моделирова- ния
-1	Аналоговый Дискретный	Символическое обозначение наследования значения Sample time от предшествующего блока
inf	Данные	Символическое обозначение необновляемо- сти данных в процессе моделирования
[<интервал> <смещение>]	Аналоговый Дискретный Данные	Первый элемент вектора < <i>интереал</i> > равен значению параметра <b>Sample time</b> .
		Второй элемент вектора < <i>смещение</i> > (Offset) отображает смещение начального времени моделирования и выбирается из условия: 0 ≤ Offset < <b>Sample time</b>

Таблица 4.2. Типовые значения параметра Sample time

□ Seed (Инициализатор) — начальное значение, запускающее генератор случайных чисел, по умолчанию 23341; при одном и том же начальном значении генерируются одинаковые сигналы;

□ Interpret vector parameters as 1-D (Интерпретировать вектор параметров как одномерный (1-D)) — флаг интерпретации вектора параметров.

*Вектор параметров* (vector parameters) — это условный термин для совокупности *численных* параметров блока.

Численные параметры блока могут задаваться скалярами, векторами или матрицами по правилам языка МАТLAB.

При этом, если хотя бы один из параметров задан *вектором*, остальные параметры будут интерпретироваться как векторы с одинаковыми элементами, а если хотя бы один из параметров задан *матрицей*, то параметры, заданные скалярами, будут интерпретироваться как матрицы с одинаковыми элементами, а векторами — как матрицы с одинаковыми элементами элементами.

Интерпретация вектора параметров (1-D или 2-D) будет зависеть от размерности численных параметров и установки (on) или сброса (off) флага Interpret vector parameters as 1-D, а *размерность выходного сигнала* (2-D или 3-D) — от интерпретации вектора параметров (табл. 4.3).

Действие флага Interpret vector parameters as 1-D показано далее в примерах 5.1—5.3.

Размерность численных параметров блока	Флаг Interpret vector parameters as 1-D	Интерпретация вектора параметров	Размерность вы- ходного сигнала
Скаляры	On	Вектор-столбец (1-D). Длина вектора равна коли- честву численных парамет- ров блока	Матрица (2-D) <i>L</i> ×1 (вектор-столбец), где <i>L</i> — количество зна- чений сигнала
	Off	Матрица 1×1 (2-D). Количество матриц равно количеству численных па- раметров конкретного блока	Трехмерный массив (3-D) 1×1×L (L матриц 1×1)
Векторы длины N	On	N векторов-столбцов (1-D). Длина каждого вектора рав- на количеству численных параметров конкретного блока	Матрица (2-D) L×N, где N — количество сигналов; L — количество зна- чений в каждом из них
	off	Матрица 1×N (2-D). Количество матриц равно количеству численных па- раметров конкретного блока	Трехмерный массив (3-D) 1×N×L (L матриц 1×N)

Таблица 4.3. Интерпретация вектора параметров и размерность выходного сигнала

Окончание табл 4.3

Размерность численных параметров блока	Флаг Interpret vec- tor parameters as 1-D	Интерпретация вектора параметров	Размерность вы- ходного сигнала
Матрицы <sup>1</sup> <i>М</i> × <i>N</i>	on	Матрица <i>M×N</i> (2-D). Количество матриц равно количеству численных па- раметров конкретного блока	Трехмерный массив (3-D) <i>M×N×L</i> ( <i>L</i> матриц <i>M×N</i> ) Здесь <i>M×N</i> — количе- ство сигналов
	off	Матрица <i>М×N</i> (2-D)	Трехмерный массив (3-D) <i>M×N×L</i> ( <i>L</i> матриц <i>M×N</i> )

Более подробную информацию о параметре флага Interpret vector parameters as 1-D можно получить в справочной системе MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  User Guide  $\rightarrow$  Working with Signals  $\rightarrow$  Determinate Output Signals Dimensions (Simulink  $\rightarrow$  Руководство пользователя  $\rightarrow$  Работа с сигналами  $\rightarrow$  Определение размерности выходного сигнала).



Рис. 4.1. Параметры и генерируемый сигнал блока Band-Limited White Noise

## 4.2.2. Блок Chirp Signal

Блок *Chirp Signal* генерирует синусоиду с линейно возрастающей мгновенной частотой и имеет следующие параметры (рис. 4.2):

**П Initial frequency (Нz)** (Начальная частота (Гц));

<sup>&</sup>lt;sup>1</sup> В том числе векторы-столбцы  $M \times 1$ .

- □ Target time (secs) (Заданное время (c)) интервал времени (в секундах), в течение которого частота синусоиды линейно возрастает от значения Initial frequency до значения Frequency at target time;
- □ Frequency at target time (Hz) (Частота на заданном времени (Гц)) конечная частота (в герцах);
- □ Interpret vector parameters as 1-D см. разд. 4.2.1.



Рис. 4.2. Параметры и генерируемый сигнал блока Chirp Signal

## 4.2.3. Блок Clock

Блок *Clock* выдает аналоговый сигнал текущего времени моделирования (с шагом моделирования) и имеет следующие параметры (рис. 4.3):

□ Display time (Показать время) — флаг отображения времени внутри пиктограммы блока: при установке флага пиктограмма имеет вид прямоугольника с отображением времени моделирования, а при сбросе — вид круга без отображения времени.

При фиксированном шаге моделирования (Fixed-step) (см. разд. 3.1) внутри пиктограммы блока отображается счетчик времени с инкрементом, определяемым параметром **Decimation**, а при *переменном* (Variable-step) — конечное время моделирования;

Decimation (Децимация) — инкремент времени — интервал, с которым значения времени отображаются внутри пиктограммы при фиксированном шаге моделирования.

Например, при шаге  $10^{-3}$  и **Decimation:** 1000 инкремент времени составит 1 с.



Рис. 4.3. Параметры и генерируемый сигнал блока Clock

## 4.2.4. Блок Constant

Блок *Constant* выдает сигнал в виде данных — численных или логических. Параметры блока задаются на двух вкладках — **Main** (Главная) и **Signal Attributes** (Атрибуты сигнала) (рис. 4.4).

	Source Block Parameters: C	onstant 🛛 🛛 🛛 🛛 🕅	
[1.378 -5.777]	Constant Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.		
📓 Source Block Parameters: Constant	Main Signal Attributes		
Constant	Output minimum:	Output maximum:	
Output the constant specified by the 'Constant value'	٥	0	
matrix with the same dimensions as the constant value	Output data type: fixdt(1,4)	▼ <<	
Main Signal Attributes	Data Type Assistant		
Constant value:	Mode: Fixed point 😪 Sign: Sig	ned 🗸 Word length: 4	
[1.378 -5.777]	Scaling: Be	st precision 😽	
Interpret vector parameters as 1-D	Fixed-point details		
Sampling mode: Sample based			
Sample time:			
inf			
		OK Cancel Help	
	OK <u>C</u> ancel	Help	

Рис. 4.4. Параметры и данные на выходе блока Constant

На вкладке Main задаются параметры:

**Constant value** (Значение константы).

Данные (константы) записываются по правилам языка MATLAB [22] и могут быть представлены в виде *скаляра*, *вектора*<sup>1</sup>, *матрицы* и вычисляемого арифметического или логического *выражения*.

Поддерживаемые типы вводимых данных приведены в табл. 4.4.

По умолчанию данные относятся к типу double. В противном случае они вводятся как аргумент соответствующей функции преобразования типа данных (см. третий столбец в табл. 4.4), например single([1 2 3]).

Числовые данные могут быть вещественными или комплексными.

Для размещения *вводимых* данных внутри пиктограммы блока Constant можно изменить его размер (см. табл. 2.1).

Символическое обозначение типа данных	Тип данных	Функция MATLAB
double	Вещественные двойной точности с ПТ <sup>2</sup>	double(C)
Single	Вещественные одинарной точности с ПТ	single(C)
int8	Целые 8-битовый со знаком с ФТ <sup>3</sup>	int8(C)
uint8	Целые 8-битовый без знака с ФТ	uint8(C)
int16	Целые 16-битовый со знаком с ФТ	int16(C)
uint16	Целые 16-битовый без знака с ФТ	uint16(C)
int32	Целые 32-битовый со знаком с ФТ	int32(C)
uint32	Целые 32-битовый без знака с ФТ	uint32(C)
boolean	Логические. Принимают значения 1 (true) или 0 (false)	boolean(C)

Таблица 4.4. Типы вводимых данных

Здесь с — константа типа double;

- □ Interpret vector parameters as 1-D *cm. pa3d. 4.2.1*;
- □ Sampling mode (Режим дискретизации) представление (интерпретация) данных как дискретного сигнала в одном из двух видов (см. разд. 11.1):
  - Sample based последовательности;

<sup>&</sup>lt;sup>1</sup> В том числе в виде N:Step:M, где Step = 1 может не указываться.

<sup>&</sup>lt;sup>2</sup> С плавающей точкой.

<sup>&</sup>lt;sup>3</sup> С фиксированной точкой.

• Frame based — последовательности фреймов. Период фрейма (в секундах) задается в поле ввода **Frame period**.

Параметр активен при сбросе флага Interpret vector parameters as 1-D;

□ Sample time — см. разд. 4.2.1.

На вкладке Signal Attributes задаются параметры (см. рис. 4.4):

□ **Output minimum** — минимальное значение данных на выходе, по умолчанию равно [], что эквивалентно – Inf (машинная бесконечность со знаком минус);

□ **Output maximum** — максимальное значение данных на выходе, по умолчанию равно [], что эквивалентно Inf (машинная бесконечность со знаком плюс).

Параметры Output minimum и Output maximum используются для контроля диапазона значений данных в процессе моделирования, а также для автоматического масштабирования данных с  $\Phi$ T;

**Оиtput data type** (Тип данных на выходе) — тип данных на выходе.

В табл. 4.5 приведен расширенный список типов данных<sup>1</sup> на выходе для различных блоков;

□ Lock output data type setting against changes by the fixed-point tools (Блокирование установки типа выходных данных от изменений средствами фиксированной точки) — флаг, при установке которого блокируется автоматическое масштабирование данных типа fixdt (табл. 4.5).

Этот флаг активен для всех типов данных, кроме fixdt, поэтому на рис. 4.4 он отсутствует.

Для контроля и редактирования типа выходных данных предусмотрена кнопка << (см. рис. 4.4), при нажатии на которую тип данных отображается в группе **Data Type Assistant** (Помощник по типу данных).

Тип данных на выходе	Способ формирования
Inherit from ''	Наследуется от указанного в апострофах параметра блока
Inherit as input	Наследуется от сигнала на входе
Inherit as first input	Наследуется от сигнала на первом входе
Inherit as second input	Наследуется от сигнала на втором входе
Inherit via back propagation	Наследуется от сигнала на выходе предыдущего блока

Таблица 4.5. Типы данных на выходе

<sup>&</sup>lt;sup>1</sup> Тип данных "enumerated" в книге не рассматривается. При необходимости с ним можно познакомиться самостоятельно с помощью справочной системы MATLAB.

67

Тип данных на выходе	Способ формирования
Inherit via internal rule	Наследуется согласно внутреннему правилу без переполнений с максимальной точностью
double, single, int8, uint8, int16, uint16, int32, uint32, boolean	Указывается явно (см. табл. 4.4)
<data expression="" type=""></data>	Указывается символическое имя типа данных в апострофах для константы в поле ввода <b>Constant</b> value, представленной выражением, например, 'single' и т. п.
fixdt	Формируются с помощью объекта fixdt для данных с ФТ (см. табл. 4.6)

При выборе в списке **Output data type** типа данных fixdt данные с ФТ формируются с помощью объекта MATLAB fixdt, различным форматам которого соответствуют различные режимы (способы) масштабирования данных, отображаемые в свойстве DataTypeMode данного объекта (табл. 4.6).

Таблица 4.6. Типы данных fixdt

Тип данных с ФТ и формат объекта fixdt	Входные параметры объекта fixdt	Свойство DataTypeMode и режим масштабирования
fixdt(S,N)	Signed — признак S нали-	Fixed-point: unspecified scal-
fixdt(Signed, Wordlength)	со знаком (Signed, когда	(Неопределенное масштабирование <sup>1</sup> )
	S = 0 — положительное без знака (Unsigned, когда старший бит — значащий);	<i>Масштабирование</i> — автоматиче- ское с максимальной точностью (Best precision)
	WordLength — длина сло- ва в битах (N)	
fixdt(S,N,M) FractionLength — Дли		Fixed-point: binary point scal-
fixdt(Signed, WordLength, Fraction-	битах (M)	(Масштабирование двоичной точки)
Length)		Масштабирование — в соответствии с длинами слова и его дробной части
fixdt(S,N,2^(-M),Bias)	Slope — коэффициент	Fixed-point: slope and bias
fixdt(Signed, WordLength, Slope, Bias)	Bias — Смещение	(Масштабирование с коэффициентом масштабирования и смещением)
		<i>Масштабирование</i> — в соответствии с входными параметрами объекта fixdt (см. комментарии после табл.)

<sup>&</sup>lt;sup>1</sup> Распределение битов в слове.

Представление данных с ФТ предполагает знакомство со следующими эквивалентами числа.

Двоичным эквивалентом числа с ФТ называют его побитовую запись в заданном формате, включая знаковый бит.

*Десятичное* (вещественное) число (RWV — Real Word Value), соответствующее двоичному эквиваленту, определяется по правилам перевода двоичного числа в десятичное.

Целочисленный эквивалент (SI — Stored Integer), соответствующий двоичному эквиваленту, определяется по правилам перевода двоичного числа в десятичное целое.

При выборе типа данных fixdt(S,N,M) (см. табл. 4.6) *десятичное число* (RWV) при S = 1 определяется с учетом того, что старший бит знаковый и вес младшего —  $2^{-M}$ , а *целочисленный эквивалент* — с учетом того, что старший бит знаковый и вес младшего —  $2^{0}$ .

При выборе типа данных fixdt(S,N,2<sup>(-M)</sup>,Bias) (см. табл. 4.6) коэффициент масштабирования slope, управляющий распределением битов в слове, будет равен:

Slope = 
$$2^{-FractionLength}$$
. (4.1)

Длина дробной части FractionLength будет равна M, а остальные *старшие* значащие биты будут интерпретироваться как целая часть двоичного числа.

Десятичное число (RWV) *а*, соответствующее двоичному эквиваленту, определяется по формуле:

$$a = a_{int} \times \text{Slope} + \text{Bias},$$
 (4.2)

где смещение Bias — целое положительное или отрицательное число, а *a*<sub>int</sub> — целочисленный эквивалент (SI), равный:

$$a_{\rm int} = (a - {\rm Bias}) / {\rm Slope}$$
 (4.3)

Примеры данных с ФТ типа fixdt приводятся далее для блока Data Type Conversion (см. разд. 7.2.2).

## 4.2.5. Блок Counter Free-Running

Блок *Counter Free-Running* формирует сигнал на выходе счетчика с коэффициентом пересчета  $2^N$  и имеет следующие параметры (рис. 4.5):

- □ Number of Bits (Количество битов) целое положительное число *N*, задающее коэффициент пересчета двоичного счетчика 2<sup>*N*</sup>; значения на выходе изменяются от 0 до 2<sup>*N*</sup> − 1 с интервалом времени Sample time;
- □ Sample time см. разд. 4.2.1.

Counter Scope Free-Running	
Source Block Parameters: Counter Free-Running	
Counter Free-Running (mask) (link)	
This block is a free-running counter that overflows back to zero after it has reached the maximum value possible for the specified number of bits. The count is always initialized to zero. The output is normally an unsigned integer with the specified number of bits. Parameters Number of Bits 3	
Sample time:	
0.2	
QK <u>C</u> ancel <u>H</u> elp	Time offset: 0

Рис. 4.5. Параметры и генерируемый сигнал блока Counter Free-Running

## 4.2.6. Блок Counter Limited

Блок *Counter Limited* формирует сигнал на выходе счетчика с произвольным коэффициентом пересчета и имеет следующие параметры (рис. 4.6):

- □ Upper limit (Верхняя граница) максимальное значение N на выходе счетчика; значения изменяются от 0 до N с интервалом времени Sample time;
- □ Sample time *см. разд. 4.2.1.*



Рис. 4.6. Параметры и генерируемый сигнал блока Counter Limited

## 4.2.7. Блок Digital Clock

Блок *Digital Clock* выдает дискретный сигнал текущего времени моделирования с периодом дискретизации. Единственный параметр **Sample time** задает период дискретизации текущего дискретного времени моделирования (рис. 4.7).



Рис. 4.7. Параметры и генерируемый сигнал блока Digital Clock

## 4.2.8. Блок From File

Блок *From File* импортирует сигналы из файла данных с расширением mat и имеет следующие параметры (рис. 4.8):

□ File name (Имя файла) — имя файла данных с расширением mat, хранящего значения импортируемого сигнала (сигналов).

Файлы данных с расширением mat, называемые МАТ-файлами, хранятся в папке work или в папке пользователя, вложенной в папку work. В первом случае следует просто указать имя файла, а во втором — путь к файлу, например: Step/fim

где Step — имя папки пользователя, а fim — имя МАТ-файла.

Данные, сохраняемые в файле, должны быть представлены в виде матрицы:

$$\mathsf{B} = [\mathsf{T}; \mathsf{U}], \tag{4.4}$$

где в — имя матрицы;

т — строка возрастающих значений времени;

и — матрица, строки которой соответствуют значениям сигналов в моменты времени, заданные вектором т.

#### Пример 4.1

В окне **Command Window** (см. рис. 1.1) сформировать и сохранить на диске в папке work файл со значениями *двух* сигналов в виде (4.4):

```
>> T=[0:pi/4:pi]
```

```
т =
```

```
Глава 4. Сигналы
```

```
0
               0.7854
                          1.5708
                                     2.3562
                                                3.1416
>> U=[sin(T);cos(T)]
U =
         0
               0.7071
                          1.0000
                                     0.7071
                                               0.0000
    1.0000
               0.7071
                          0.0000
                                    -0.7071
                                              -1.0000
>> B=[T;U]
в =
         0
               0.7854
                          1.5708
                                     2.3562
                                               3.1416
         0
               0.7071
                          1.0000
                                     0.7071
                                               0.0000
    1 0000
               0 7071
                          0.0000
                                    -0.7071
                                               -1.0000
>> save matrix B
```

межуточных точках вычисляются посредством линейной интерполяции между

**name:** matrix (см. рис. 4.8).

времени.

соседними точками. При этом если заданный начальный момент времени больше, а конечный — меньше соответствующих начального и конечного времени моделирования, *незаданные* значения сигнала вычисляются посредством линейной экстраполяции по двум первым или двум последним точкам.

Матрица в содержит три строки: первая (вектор т) хранит моменты времени, вторая и третья (матрица U) — значения первого и второго сигналов в моменты времени, заданные вектором т. При импорте этих сигналов следует указать **File** 

На выходе блока From File выдаются только значения сигналов без значений

Согласование интервала между соседними моментами времени и шагом моделирования происходит автоматически: *незаданные* значения сигнала в про-

□ **Sample time** — *см. разд. 4.2.1*.

	From File Scope
Source Block Parameters: From File	Scope
From File Read time and output values from the first matrix in the specified MAT file. The matrix must contain time values in row one. Additional rows correspond to output elements. Interpolates between columns.	
Parameters File name: matrix	05
Sample time:	0.5
QK <u>C</u> ance <u>H</u> elp	.1 0 0.5 1 1.5 2 2.5 3 Time offset: 0

Рис. 4.8. Параметры и импортируемый сигнал блока From File

## 4.2.9. Блок From Workspace

Блок *From Workspace* импортирует сигналы из рабочей области памяти Workspace и имеет следующие параметры (рис. 4.9):

□ Data (Данные) — имя *матрицы* (по умолчанию) или *структуры* (массива записей), импортируемой из Workspace и хранящей значения сигнала (сигналов).

Импортируемая из Workspace матрица должна быть представлена в виде:

$$A=[T' U'],$$
 (4.5)

где А — имя матрицы;

т' — *столбец* возрастающих значений времени;

U' — матрица, столбцы которой соответствуют значениям сигналов в моменты времени, заданные вектором т.

Матрица A (4.5) является транспонированной по отношению к матрице в (4.4): A=B'.

#### Пример 4.2

В окне **Command Window** (см. рис. 1.1) сформировать сохраняемую в Workspace матрицу A (4.5) для *двух* сигналов:

```
>> T=[0:pi/4:pi]'
т =
         0
    0.7854
    1.5708
    2.3562
    3.1416
>> U=[sin(T) cos(T)]
U =
         0
               1.0000
    0.7071
              0.7071
    1.0000
              0.0000
    0.7071
              -0.7071
    0.0000
              -1.0000
>> A=[T U]
A =
         0
                    0
                          1.0000
    0.7854
              0.7071
                          0.7071
    1.5708
               1.0000
                          0.0000
    2.3562
               0.7071
                         -0.7071
    3.1416
               0.0000
                         -1.0000
```
Матрица A содержит три столбца: первый (вектор т) хранит моменты времени; второй и третий (матрица U) — значения первого и второго сигналов в моменты времени, заданные вектором т. Для импорта матрицы A из Workspace следует указать **Data:** A (см. рис. 4.9).

📓 Source Block Parameters: From Workspace 🛛 🛛 🔀	
From Workspace Read data values specified in array or structure format from MATLAB's workspace. Array (or matrix) format: 1-D signal: var=[TimeValues DataValues] For 2-D signal use structure format Structure format: var.time=TimeValues]	A From Scope Workspace
var.signals.values=[DataValues] var.signals.dimensions=[DimValues] Select interpolation to interpolate or extrapolate at time steps for which data does not exist.	
Parameters Data: A	
Sample time:	
Interpolate data     Enable zero crossing detection     Form output after final data value by: Extrapolation	
<u>QK</u> <u>C</u> ancel <u>H</u> elp	Time offset: 0

Рис. 4.9. Параметры блока From Workspace при импорте сигнала в виде матрицы

Структура, импортируемая из Workspace, должна содержать два поля [22]:

- time вектор-столбец возрастающих значений времени;
- signals вложенная структура с двумя полями:
  - values матрица, столбцы которой соответствуют значениям сигналов в моменты времени, заданные в поле time;
  - dimensions количество сигналов (скаляр).

#### Пример 4.3

В окне **Command Window** (см. рис. 1.1) сформировать сохраняемую в Workspace структуру с для трех сигналов: U1, U2 и F:

3.1416

```
>> U1=sin(T); U2=cos(T); F=sqrt(T);
>> C.signals.values=[U1 U2 F];
>> C.signals.dimensions=3;
>> C
C. =
       time: [5x1 double]
    signals: [1x1 struct]
>> C.signals.values
ans =
         0
              1.0000
                              0
    0.7071
                         0.8862
             0.7071
    1.0000
             0.0000
                        1.2533
    0.7071
           -0.7071
                         1.5350
    0.0000
             -1.0000
                        1.7725
```

Структура с описывает три сигнала U1, U2 и F в моменты времени, заданные в поле time. Для импорта структуры из Workspace следует указать **Data:** С (рис. 4.10).

Для вывода в окне Command Window значений одного сигнала, например второго во втором столбце, следует ввести:

```
>> C.signals.values(:,2)
ans =
    1.0000
    0.7071
    0.0000
    -0.7071
    -1.0000
```

Для вывода значений *всех* сигналов в фиксированный момент времени, например соответствующий третьему элементу вектора т в поле time, следует ввести:

- □ Sample time *см. разд. 4.2.1*;
- Interpolate data (Интерполяция данных) флаг интерполяции данных, управляющий согласованием интервала между соседними моментами времени и шагом моделирования.

При установке флага (по умолчанию) незаданные значения сигнала в промежуточных точках вычисляются посредством линейной интерполяции между соседними точками. Если начальный момент времени больше начального времени моделирования, незаданные начальные значения сигнала вычисляются посредством линейной экстраполяции по первым двум точкам. При *сбросе* флага *незаданные* значения сигналов в промежуточных точках дублируют его предыдущее значение. Если *начальный* момент времени больше начального времени моделирования, *незаданные* начальные значения сигнала дублируют его первое значение;

□ Enable zero crossing detection (Доступно обнаружение пересечения нуля) — флаг контроля монотонности сигнала на каждом шаге моделирования.

Установка флага (по умолчанию) позволяет определять моменты скачков сигнала (резкого изменения значения) и автоматически уменьшать шаг моделирования при выборе типа решателя с переменным шагом (Variable-step) *(см. разд. 3.1)*;

□ Form output after final data value by (Формировать сигнал на выходе после последнего значения данных посредством) — метод вычисления незаданных *конечных* значений сигнала.

Если *конечный* момент времени меньше конечного времени моделирования, то *незаданные* конечные значения сигнала будут вычисляться по-разному, в зависимости значения этого параметра, а именно:

- Extrapolate (Экстраполяция) посредством линейной экстраполяции;
- Setting to zero (Установка нуля) добавлением нулей;
- Hold final data value (Удерживание последнего значения) дублированием последних значений сигнала;
- Cyclic repetition (Циклическое повторение) периодическим повторением значений сигнала.

Source Block Parameters: From Workspace	
From Workspace Read data values specified in array or structure format from MATLAB's workspace. Array (or matrix) format:     1-D signal:     var=[TimeValues DataValues]     For 2-D signal use structure format Structure format:     var.time=[TimeValues]     var.signals.values=[DataValues]     var.signals.dimensions=[DimValues]     Select interpolation to interpolate or extrapolate at time steps for which data does not     exist.	C From Scope Workspace
Parameters         Data:         [c]         Sample time:         0         ✓ Interpolate data         ✓ Enable zero crossing detection         Form output after final data value by: Extrapolation	

Рис. 4.10. Параметры блока From Workspace при импорте сигнала в виде структуры

### 4.2.10. Блок Ground

Блок *Ground* представляет собой "заглушку" для неподключенного входа в блоке с несколькими входами — нулевой сигнал. В отсутствие "заглушки" при моделировании выдается предупреждение. Этот блок не содержит параметров.

### 4.2.11. Блок In

Блок *In* — это входной порт для сигналов, импортируемых из Workspace, и для подсистем. Параметры блока задаются на двух вкладках — Main и Signal Attributes (рис. 4.11).

🖬 Sour	ce Block Parame	eters: In1	🗑 Sour	ce Block Parameters: In1
Provide For Trigg value of For Fun block's of The oth	an input port for a s gered Subsystems, 'l f the subsystem inpu ction-call Subsystem output to a buffer be er parameters can b	ubsystem or model. .atch input by delaying outside signal' pro t at the previous time step. s, 'Latch input by copying inside signal' cop fore the contents of the subsystem are e e used to explicitly specify the input signal	Inport Provide an input port for a subsystem or model, For Triggered Subsystems, Latch input by delaying outside signal' produces the value of the subsystems, Latch input by copying inside signal' copies the Inport block's output to a buffer before the contents of the subsystem are executed. The other parameters can be used to explicitly specify the input signal attributes.	
Main	Signal Attributes	γ	Main	Signal Attributes
Port num	nber:		Spec	ify properties via bus object
1			Bus obje	ct for validating input bus:
Icon disp	lay: Port number		BusObject	
Latch input by delaying outside signal     Latch input by copying inside signal     Vaternolate data		Output as nonvirtual bus Port dimensions (-1 for inherited): -1		
		Sample time (-1 for inherited): -1		
		Minimum Data typ Signal ty Sampling	: Maximum:	
		<u>OK</u> <u>C</u> ancel		QK <u>C</u> ancel <u>H</u> elp

Рис. 4.11. Параметры блока In

При нумерации блоков In соблюдаются следующие правила:

- 🗅 порты имеют сквозную нумерацию, начиная с 1;
- □ добавление или удаление порта сопровождается автоматической перенумерацией;
- при нарушении сквозной нумерации ее следует исправить вручную, в противном случае при моделировании системы будет выдано сообщение об ошибке.

На вкладке Main задаются параметры:

- **□** Port Number (Номер порта);
- □ Icon display (Текст внутри пиктограммы) варианты текста внутри пиктограммы:
  - номер порта (Port number);

- имя сигнала (Signal name);
- номер порта и имя сигнала (Port number and signal name);
- □ Latch input by delaying outside signal (Защелка входа посредством задержки выходного сигнала) флаг, активный, если блок In является входным портом Т-подсистем (см. разд. 8.3.2). При установке флага сигнал на выходе блока In появляется с задержкой на шаг моделирования;
- Latch input by coping inside signal (Защелка входа посредством копирования входного сигнала) — флаг, активный, если блок In является входным портом подсистем запроса функции (Function-call subsystem), которые в настоящей книге не рассматриваются;
- □ Interpolate data см. разд. 4.2.9.

На вкладке Signal Attributes задаются параметры:

Specify properties via bus object (Определить свойства через объект шины) флаг определения структуры шины, создаваемой блоком In, через объект шины (Bus Object);

При сбросе флага (по умолчанию) шина формируется автоматически и является виртуальной.

В общем случае шины таких блоков как In (см. разд. 4.2.11), Out (см. разд. 5.2.2) и Bus Creator (см. разд. 7.1.3) могут быть виртуальными или невиртуальными.

Если блоки, генерирующие сигналы, подсоединяются к *виртуальной* шине, то в процессе моделирования чтение входных и запись выходных сигналов шины осуществляются путем непосредственного обращения к областям памяти данных сигналов, и виртуальная шина является лишь удобным графическим средством отображения взаимодействия сигналов при их объединении или разделении.

Если блоки, генерирующие сигналы, подсоединяются к *невиртуальной* шине, то в процессе моделирования чтение и запись сигналов шины организуются посредством обращения к копиям данных сигналов, для чего создается промежуточная область памяти шины.

При *установке* флага активизируется поле ввода **Bus object for validating input bus** (Объект шины для подтверждения входной шины) и флаг **Output as nonvirtual bus** (Выход как невиртуальная шина), формирующие свойства шины. При необходимости с этими параметрами можно познакомиться самостоятельно.

В дальнейшем параметры блоков In, Out и Bus Creator будем рассматривать только при *сброшенном* флаге Specify properties via bus object.

#### **□** Port dimensions (Размерность порта).

Значение –1 (по умолчанию) соответствует наследованию размерности от импортируемого сигнала. Для вектора задается его длина N, а для матрицы — [N M], где N, М — число строк и столбцов;

- □ **Sample time** *см. разд. 4.2.1*;
- □ **Minimum** аналогичен параметру **Output minimum** блока Constant *(см. разд. 4.2.4)*;

□ **Махітит** — аналогичен параметру **Output maximum** блока Constant *(см. разд. 4.2.4)*;

□ Data type — тип данных на выходе (см. табл. 4.5); значение Inherit: auto соответствует наследованию типа от импортируемого сигнала.

Для контроля и редактирования типа данных на выходе предусмотрена кнопка >> справа от списка **Data type**;

- Signal type (Тип сигнала) вещественный, комплексный или наследуемый от импортируемого сигнала (auto);
- □ Sampling mode см. разд. 4.2.4.

Значение auto соответствует наследованию значения Sample based или Frame based от импортируемого сигнала.

### 4.2.12. Блок Pulse Generator

Блок *Pulse Generator* генерирует периодические прямоугольные импульсы и имеет следующие параметры (рис. 4.12):

□ Pulse type (Тип импульса) — тип импульсного сигнала:

- Time based аналоговый;
- Sample based дискретный.

Для дискретного сигнала активизируется параметр Sample time (см. разд. 4.2.1);

□ Time (t) (Время) — способ задания текущего времени моделирования:

- Use simulation time (Использование времени моделирования) текущее время моделирования определяется соответствующими параметрами настройки в окне Configuration Parameters (см. разд. 3.1); для дискретных сигналов оно определяется шагом моделирования — параметром Sample time;
- Use external signal (Использование внешнего сигнала) текущее время моделирования определяется параметрами внешнего блока на автоматически создаваемом входе блока Pulse Generator; для дискретных сигналов оно определяется шагом моделирования — параметром Sample time внешнего блока Digital Clock;
- Amplitude (Амплитуда) амплитуда импульса;
- □ Period (secs) (Период (c)) период сигнала в секундах; для дискретного сигнала в количестве отсчетов на периоде;
- Pulse Widht (% of period) (Ширина импульса в процентах от периода) ширина импульса в процентах от периода; для дискретного сигнала — в количестве отсчетов;
- □ Phase delay (secs) (Фазовая задержка (c)) задержка сигнала в секундах; для дискретного сигнала в количестве отсчетов;
- □ Interpret vector parameters as 1-D см. разд. 4.2.1.

🗑 Source Block Parameters: Pulse Generator	3
Pulse Generator	
Output pulses:	
if (t >= PhaseDelay) && Pulse is on Y(t) = Amplitude else Y(t) = 0 end Pulse type determines the computational technique used. Time-based is recommended for use with a variable step solver, while Sample-based is recommended for use with a fixed step solver or within a discrete portion of a model	
using a variable step solver.	Generator
Parameters	
Pulse type: Time based	Scope
Time (t): Use simulation time	
Amplitude:	
1	2
Period (secs):	1.5
2	
Pulse Width (% of period):	أختبين وانتبتن والتنتيز والبيب والتبير وا
25	0.5
Phase delay (secs):	
0	-0.5
✓ Interpret vector parameters as 1-D	
QK Cancel Help	Time offset: 0

Рис. 4.12. Параметры и генерируемый сигнал блока Pulse Generator

### 4.2.13. Блок Ramp

Блок Ramp генерирует линейно изменяющийся сигнал:

$$x(t) = at + b$$

и имеет следующие параметры (рис. 4.13):

- □ **Slope** (Наклон) коэффициент *a*;
- □ Start time (Начальное время) задержка сигнала;
- □ Initial output (Начальное значение) начальное значение сигнала *b*;
- □ Interpret vector parameters as 1-D см. разд. 4.2.1.

🗑 Source Block Parameters: Ramp 🛛 🛛 🔀	Ramp Scope
Ramp (mask) (link)	🛃 Scope 📃 🗖 🔀
Output a ramp signal starting at the specified time.	60 / × × A B B B 4 + ·
Parameters	
Slope:	5
0.4	4
Start time:	3
0	
Initial output:	
0	1
Interpret vector parameters as 1-D	0
	4
QK <u>C</u> ancel <u>H</u> elp	0 2 4 6 8 10 Time offset: 0

Рис. 4.13. Параметры и генерируемый сигнал блока Ramp

### 4.2.14. Блок Random Number

Блок *Random Number* генерирует нормальный белый шум с заданными математическим ожиданием и дисперсией и имеет следующие параметры (рис. 4.14):

- □ Mean (Среднее) математическое ожидание (среднее значение);
- □ Variance (Дисперсия) дисперсия;
- □ Initial seed (Инициализатор) аналогичен параметру Seed (см. разд. 4.2.1);
- □ Sample time *см. разд. 4.2.1*;
- □ Interpret vector parameters as 1-D *cm. pa3∂. 4.2.1.*

Random Number Output a normally (Gaussian) distributed random signal. Output is	Random Scope Number
Parameters Mean:	
þ Variance:	
Initial seed:	
Sample time: 0	
✓ Interpret vector parameters as 1-D       QK     Cancel	-3 -3 0 2 4 6 8 10 Time offset: 0

Рис. 4.14. Параметры и генерируемый сигнал блока Random Number

### 4.2.15. Блок Repeating Sequence

Блок *Repeating Sequence* генерирует периодический сигнал по заданным векторам времени и значений сигнала с линейной интерполяцией между соседними значениями сигнала. Для блока задаются следующие параметры (рис. 4.15):

- □ **Time values** (Значения времени) вектор возрастающих значений времени на периоде сигнала;
- Output values (Значения на выходе) вектор значений сигнала на периоде, между которыми выполняется линейная интерполяция; для формирования периодического сигнала последний элемент вектора должен совпадать с первым.

Repeating Scope Sequence	
Source Block Parameters: Repeating Sequence	
Repeating table (mask) (link)	
Output a repeating sequence of numbers specified in a table of time- value pairs. Values of time should be monotonically increasing.	$^{2}$
Parameters	
Time values:	
[0 1 3 5]	of
Output values:	
[0 2 -1 0]	.1
<u>OK</u> <u>C</u> ancel <u>H</u> elp	-2_0

Рис. 4.15. Параметры и генерируемый сигнал блока Repeating Sequence

## 4.2.16. Блок Repeating Sequence Interpolated

Блок *Repeating Sequence Interpolated* генерирует периодическую последовательность по заданным векторам времени и значений сигнала с различными способами вычисления отсчетов в промежуточных точках. Параметры блока задаются на двух вкладках — Main и Signal Attributes (рис. 4.16).

На вкладке Main задаются параметры:

- □ Vector of output values (Вектор выходных значений) вектор-столбец значений сигнала на периоде;
- Vector of time values (Вектор значений времени) вектор-столбец возрастающих значений времени на периоде сигнала;
- □ Lookup Method (Метод поиска) способ вычисления отсчетов в промежуточных точках между соседними значениями сигнала:
  - Interpolation-Use End values (Использование для интерполяции граничных значений) присваивается значение, полученное в результате линейной интерполяции между соседними значениями сигнала;
  - Use Input Nearest (Использование ближайшего входа) присваивается ближайшее значение сигнала;
  - Use Input Below (Использование верхнего входа) присваивается наибольшее из соседних значений сигнала;
  - Use Input Above (Использование нижнего входа) присваивается наименьшее из соседних значений сигнала;

#### □ Sample time — см. разд. 4.2.1.

На вкладке Signal Attributes параметры аналогичны параметрам блока Constant (см. разд. 4.2.4).

Source Block Parameters: Repeating Sequence Interpol X	Repeating Sequence Interpolated	Scope
Discrete time sequence is output, then repeated. Between data points, the specified lookup method is used to determine the output.           Main         Signal Attributes           Vector of output values:         Vector of output values:	Scope ● □ ○ ○ ○ ○ ▲ [	
[0 1 4 2].' Vector of time values: [0 1 2 3].'	5 4 3	
Lookup Method: Use Input Above Sample time: 0.5		6 8 10

Рис. 4.16. Параметры и генерируемая последовательность блока Repeating Sequence Interpolated

### 4.2.17. Блок Repeating Sequence Stair

Блок *Repeating Sequence Stair* генерирует периодическую последовательность по заданному вектору значений отсчетов и периоду дискретизации. Параметры блока задаются на двух вкладках — **Main** и **Signal Attributes** (рис. 4.17).

На вкладке Main определяются параметры:

- □ Vector of output values (Вектор выходных значений) вектор-столбец значений отсчетов на периоде;
- □ Sample time *см. разд. 4.2.1*.

На вкладке Signal Attributes параметры аналогичны параметрам блока Constant (см. разд. 4.2.4).



Рис. 4.17. Параметры и генерируемый сигнал блока Repeating Sequence Stair

### 4.2.18. Блок Signal Generator

Блок Signal Generator генерирует четыре вида сигналов (гармонический, периодический трапецеидальный, периодический пилообразный и случайный) и имеет следующие параметры (рис. 4.18):

□ Wave form (Форма волны) — вид сигнала:

- sine гармонический (синусоида);
- square периодический трапецеидальный;
- sawtooht периодический пилообразный;
- random случайный;
- □ **Time (t)** см. разд. 4.2.12;
- Amplitude (Амплитуда) амплитуда сигнала;
- □ Frequency (Частота) частота сигнала;
- □ Units (Единицы) единицы измерения частоты в герцах (Hertz) или в рад/с (rad/sec);
- □ Interpret vector parameters as 1-D см. разд. 4.2.1.



Рис. 4.18. Параметры и генерируемый сигнал (синусоида) блока Signal Generator

## 4.2.19. Блок Sine Wave

Блок Sine Wave генерирует гармонический сигнал (синусоиду):

 $x(t) = A\sin(\omega t + \varphi) + bias$ 

и имеет следующие параметры:

**Бие type** (Тип синусоиды):

- Time based аналоговая;
- Sample based дискретная;

- □ **Time (t)** *см. разд. 4.12*;
- □ Amplitude (Амплитуда) амплитуда *А*;
- □ Bias (Смещение) смещение *bias* (постоянная составляющая);
- □ Sample time *см. разд. 4.2.1*;
- □ Interpret vector parameters as 1-D *cm. pasd. 4.2.1.*

Для аналоговой синусоиды (Sine type: Time based) задаются параметры (рис. 4.19):

- **Γ** Frequency (rad/sec) (Частота (рад/с)) частота ω в рад/с;
- □ Phase (rad) (Фаза (рад)) фаза φ в рад.

Для *дискретной* синусоиды (Sine type: Sample based) указываются параметры (рис. 4.20):

- Samples per period (Количество отсчетов на периоде) количество отсчетов на периоде;
- □ Number of offset samples (Количество сдвигаемых отсчетов) задержка сигнала в количестве отсчетов.



Рис. 4.19. Параметры и генерируемый сигнал блока Sine Wave с параметром Sine type: Time based

Source Block Parameters: Sine Wave	1
Sine Wave	
Output a sine wave:	
O(t) = Amp*Sin(Freq*t+Phase) + Bias	
Sine type determines the computational technique used. The parameters in the two types are related through:	
Samples per period = 2"pi / (Frequency " Sample time)	
Number of offset samples = Phase * Samples per period / $(2*pi)$	
Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.	
Parameters	
Sine type: Sample based	++
Time (f): Use circulation time	Sine Wave Scope
Amplitude:	🛃 Scope 📃 🗖 🔯
Ampirude:	
- Piaer	
0	2
Samples per period:	
10	1 mapping income second paragraphic second second
Number of offset samples:	
0	
Sample time:	
0.5	-1
✓ Interpret vector parameters as 1-D	20 2 4 2 10
QK Cancel Help	0 2 4 0 0 10 Time offset 0

Рис. 4.20. Параметры и генерируемый сигнал блока Sine Wave с параметром Sine type: Sample based

## 4.2.20. Блок Step

Блок Step генерирует скачок и имеет следующие параметры (рис. 4.21):

- □ Step time (Время скачка) момент скачка в секундах;
- □ Initial value (Начальное значение) и Finel value (Конечное значение) начальное и конечное значения сигнала в момент скачка; могут быть положительными или отрицательными;
- □ **Sample time** *см. разд. 4.2.1*;
- □ Interpret vector parameters as 1-D см. разд. 4.2.1;
- □ Enable zero crossing detection см. разд. 4.2.9.

📓 Source Block Parameters: Step	
Step	Step Scope
Output a step.	
Parameters	Scope
Step time:	
1	
Initial value:	
-3	
Final value:	
4	
Sample time:	
þ	
Interpret vector parameters as 1-D	
Enable zero crossing detection	
	0 2 4 6 8 10
<u>QK</u> <u>Cancel</u> <u>H</u> elp	Time offset: 0

### 4.2.21. Блок Uniform Random Number

Блок Uniform Random Number генерирует равномерный белый шум с заданными максимальным и минимальным значениями случайной величины. Для блока задаются следующие параметры (рис. 4.22):

- □ **Minimum** и **Maximum** минимальное и максимальное значения случайной величины;
- □ Initial seed (Инициализатор) аналогичен параметру Seed (см. разд. 4.2.1);
- □ Sample time *см. разд. 4.2.1*;
- □ Interpret vector parameters as 1-D см. разд. 4.2.1.



Рис. 4.22. Параметры и генерируемый сигнал блока Uniform Rundom Number

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  Blocks  $\rightarrow$  Sources (Simulink  $\rightarrow$  Блоки  $\rightarrow$  Источники).

# 4.3. Виртуальные и невиртуальные блоки

В библиотеке Simulink различают два вида блоков.

□ Виртуальный (virtual — смоделированный) блок содержит признак "mask" в названии (см. табл. 4.1) и представляет собой маскируемую подсистему (см. разд. 8.5), составленную из нескольких блоков библиотеки Simulink.

Примером виртуального блока является Band-Limited White Noise (см. разд. 4.2.1). Соответствующую ему маскируемую подсистему можно открыть в окне S-моделей

по команде меню Edit | Look Under Mask (Редактирование | Заглянуть под маску) (рис. 4.23).



Рис. 4.23. Состав виртуального блока Band-Limited White Noise

□ *Невиртуальный* (nonvirtual) блок не содержит признака "mask" и является "закрытым" — в процессе моделирования функционирует как единый (неделимый) блок.



## Глава 5

# Средства анализа сигналов

В этой главе рассматриваются S-модели средств анализа сигналов, представленные в группе блоков Sinks (Приемники) библиотеки блоков Simulink.

## 5.1. S-модели средств анализа сигналов

S-модели средств анализа сигналов — блоки в группе блоков Sinks (табл. 5.1) отвечают следующим условиям:

🖵 имеют один или несколько входов;

□ не имеют выходов.

N⁰	Блок	Назначение
1.	Display	Имитирует дисплей — отображает численные значения
2.	Out	Выходной порт для сигналов, экспортируемых в рабочую область памяти Workspace, и для подсистем
3.	Scope	Имитирует осциллограф — отображает сигналы в процессе моделирования
4.	Stop Simulation	Прерывает процесс моделирования при поступлении на вход нену- левого сигнала
5.	Floating Scope	Имитирует плавающий осциллограф, оперативно подсоединяемый к линиям связи
6.	Terminator	"Заглушка" для неподключенного выхода в блоке с несколькими вы- ходами. В отсутствие "заглушки" при моделировании выдается предупреж- дение
7.	To File	Экспортирует сигналы в файл данных с расширением mat
8.	To Workspace	Экспортирует сигналы в Workspace
9.	XY Graph (mask)	Имитирует графопостроитель — отображает двумерные графики в декартовой системе координат

Таблица 5.1. Группа блоков Sinks

## 5.2. Параметры блоков

Рассмотрим параметры блоков из табл. 5.1.

### 5.2.1. Блок Display

Блок *Display* имитирует дисплей — отображает численные значения. Блок имеет следующие параметры (рис. 5.1):

□ Format (Формат) — формат вывода численных значений (табл. 5.2), по умолчанию — short;

Формат вывода	Назначение	Команда MATLAB
short	Короткий формат с выводом в обычной форме	format short
long	Длинный формат с выводом в обычной форме	format long
short_e	Короткий формат E с выводом в нормализо- ванной форме E	format short_e
long_e	Длинный формат E с выводом в нормализо- ванной форме E	format long_e
bank	Банковский формат с выводом в обычной фор- ме с двумя значащими цифрами в дробной части	format bank
hex (Stored Integer)	Шестнадцатеричный эквивалент <sup>1</sup> чисел типа int, uint или fixdt (см. табл. 4.5)	-
binary (Stored Integer)	Двоичный эквивалент чисел типа int, uint или fixdt (см. табл. 4.5)	_
decimal (Stored Integer)	Целочисленный эквивалент чисел типа int, uint или fixdt (см. табл. 4.5)	_

Таблица 5.2. Форматы вывода численных значений в блоке Display

□ Decimation (Децимация) — коэффициент децимации (прореживания) при выводе численных значений сигнала;

При Decimation: 1 выводятся все значения сигнала без прореживания, при Decimation: 2 — каждое второе значение и т. д.;

□ Floating display (Плавающий дисплей) — флаг перевода блока в плавающее состояние.

<sup>&</sup>lt;sup>1</sup> Шестнадцатеричным эквивалентом целого числа называют краткую запись его двоичного эквивалента, когда каждые его четыре бита, включая знаковый, представляются шестнадцатеричной цифрой.

В плавающем состоянии блок не имеет входа и подключается к выходу блока щелчком левой кнопки мыши на соответствующей линии связи.

Constant	Displa	у
Sink Block Parameters: Display	6	1
Display		
Numeric display of input values.		
Parameters		
Format: short		~
Decimation:		
1		
Floating display		

Рис. 5.1. Параметры блока Display и пример отображения матрицы в формате short

При отображении векторов и матриц исходный вид блока Display автоматически меняется, признаком чего будут появившиеся черные треугольники (один для вектора и два для матрицы) в правом нижнем углу блока. Для того чтобы увидеть вектор или матрицу полностью, следует выделить блок щелчком левой кнопки мыши, подвести курсор к правому нижнему углу блока, и после того как появится двунаправленная диагональная стрелка, не отпуская левой кнопки мыши, растянуть блок.

#### 5.2.2. Блок Out

Блок *Out* — это выходной порт для сигналов, экспортируемых в рабочую область памяти Workspace, и для подсистем. Параметры блока задаются на двух вкладках — Main и Signal Attributes.

При нумерации блоков Out соблюдаются те же правила, что и при нумерации блоков In (см. разд. 4.2.11).

На вкладке Main задаются параметры (рис. 5.2):

- **□ Port Number** *см. разд. 4.2.11;*
- □ Icon display см. разд. 4.2.11;
- □ **Output when disabled** (Выход при блокировании) параметр, активный, если блок Out является выходным портом Е-подсистемы (см. разд. 8.3.1) и задает варианты формирования выходного сигнала при ее блокировании:

- reset (сброс) выходной сигнал принимает значение, заданное в параметре Initial output;
- held (удерживание) выходной сигнал принимает (удерживает) последнее значение, когда подсистема была активной;
- □ Initial output (Инициализация выхода) начальное значение выходного сигнала для управляемых подсистем при запуске моделирования или при блокировании подсистемы.

Для Е-подсистемы (см. разд. 8.3.1) при Output when disabled: reset значение Initial output присваивается при каждом блокировании Е-подсистемы.

Для Т-подсистемы (см. разд. 8.3.2) значение Initial output присваивается один раз при запуске моделирования.

По умолчанию задано Initial output: [], что соответствует независимости управляемой подсистемы (см. разд. 8.3) от начального значения выходного сигнала.

На вкладке Signal Attributes параметры аналогичны параметрам блока In *(см. разд. 4.2.11)* с той разницей, что In — входной порт, а Out — выходной.

Sink Block Parameters: Out1	🐱 Sin	k Block Parameters: Out1
<ul> <li>Outport</li> <li>Provide an output port for a subsystem or model. The 'Output wi 'Initial output' parameters only apply to conditionally executed sul conditionally executed subsystem is disabled, the output is either or set to the 'Initial output'.</li> </ul>	er Provid sy 'Initial ne conditi or set	Int le an output port for a subsystem or model. The 'Output when disabled' and output' parameters only apply to conditionally executed subsystems. When a ionally executed subsystem is disabled, the output is either held at its last value to the 'Initial output'.
Main Signal Attributes	Main	Signal Attributes
Port number:	Bus obj	eify properties via bus object ject for validating input bus:
Icon display: Port number	BusOb	oject
Output when disabled: held	Port dir	tput as nonvirtual bus in parent model mensions (-1 for inherited):
Initial output:	-1	
Lu	Sample	time (-1 for inherited):
	-1	
	Minimu	um: Maximum:
		0
	Data ty Signal t	ype: Inherit: auto >> type: auto v
	Samplin	ng mode: auto
<u>OK</u> <u>C</u> ancel <u>H</u> el	5	QK <u>C</u> ancel <u>H</u> elp Apply

Рис. 5.2. Параметры блока Out

## 5.2.3. Блок Scope

Блок *Scope* отображает сигналы в окне **Scope**, которое открывается двойным щелчком левой кнопки мыши на пиктограмме блока на *любом* этапе моделирования.

Окно Scope содержит панель инструментов из 11 кнопок (рис. 5.3).

Sine Wave	
Lock/Unlock axes selection Parameters Floating Scope Signal select Scope	Stion   General Data history   Tip: try right clicking on axes   Axes   Number of axes:   Time range:   auto   Tick labels:   bottom axis only
2 0 .2 Sine Wave	Sampling Decimation 1 OK Cancel Help Apply Scope' parameters
-20246810 Time offset: 0	General Data history Tip: try right clicking on axes
Scoper properties: Step       Y-min:       -2       Y-max:       Title ('% <signallabel>' replaced by signal name):       Step       OK     Cancel</signallabel>	Variable name: ScopeData Format: Structure with time OK Cancel Help Apply

Рис. 5.3. Параметры блока Scope и пример отображения двух сигналов

Параметры блока Scope задаются в окне 'Scope' parameters с двумя вкладками General (Общие) и Data history (История данных), которое открывается при помощи кнопки Parameters на панели инструментов в окне Scope (рис. 5.3).

На вкладке General задаются параметры:

- □ Number of axis (Число осей) количество входов блока Scope, задающее количество независимых графиков для отображения различных сигналов, выделяемых цветом, при одинаковой оси абсцисс;
- □ floating scope (Плавающий осциллограф) флаг, при *сбросе* которого блок Scope работает в режиме осциллографа, а при *установке* в режиме плавающего осциллографа.

Для работы блока Scope в режиме плавающего осциллографа следует предварительно скопировать его на свободное поле окна S-моделей, после чего выполнить действия, описанные далее для блока Floating scope; □ **Time range** (Диапазон времени) — интервал времени (в секундах) по оси абсцисс; значение auto соответствует интервалу моделирования.

При **Time range**, большем или равном интервалу моделирования, сигнал в окне **Scope** отображается полностью и под графиком выводится сообщение Time offset: 0 (Сдвиг времени равен нулю).

В противном случае в окне **Scope** отображается сигнал на *последнем* отрезке времени, меньшем или равном **Time range**, а в строке **Time offset** выводится предшествующее (скрытое) время. Например, если **Time range**: 1.7 (с) при интервале моделирования 10 (с), то в окне **Scope** отображается сигнал на последнем отрезке времени, равном 1.5, и выводится **Time offset**: 8.5, где 8.5 — 5 полных интервалов по 1.7 (с) ( $5 \times 1.7 = 8.5$ ), а 1.5 — остаток (10 - 8.5 = 1.5);

□ Tick labels (Пометить оси) — варианты задания меток на осях:

- all (по умолчанию) задаются на всех осях;
- none отсутствуют на всех осях;
- bottom axis only имеются на всех осях ординат и только на одной (нижней) оси абсцисс;

Sampling (Дискретизация) — варианты задания интервалов времени между выводимыми в окне Scope значениями сигнала:

- Decimation (Децимация) см. блок Display (см. разд. 5.2.1);
- Sample time см. табл. 4.2.

На вкладке Data History задаются параметры:

Limit data points to last (Предельное число точек для сохранения) — флаг, при установке которого задается предельное число сохраняемых значений сигнала, а при сбросе — сохраняются все значения;

□ Save data to workspace (Сохранение данных в Workspace) — флаг, при установке которого сигналы сохраняются в Workspace и активизируются параметры Variable name и Save format, которые рассматриваются далее для блока To Workspace (см. разд. 5.2.8).

Назначение остальных кнопок панели инструментов окна Scope легко осваивается интуитивно.

Контекстное меню окна Scope открывается щелчком правой кнопки мыши на свободном поле графика. По команде Axes properties (Свойства осей) открывается окно 'Scope' properties: axes N (Свойства блока Scope: оси N), где N — номер графика (см. рис. 5.3). Для выделенного графика в полях Y-min и Y-max можно указать границы по оси ординат, а в поле Title — заголовок графика.

### 5.2.4. Блок Stop Simulation

Блок *Stop Simulation* прерывает процесс моделирования при поступлении на вход ненулевого сигнала. Этот блок не содержит параметров.

#### 5.2.5. Блок Floating Scope

Блок *Floating Scope* оперативно отображает сигналы в окне **Floating Scope** в *любой* точке S-модели системы и на *любом* этапе моделирования.

Окно Floating Scope открывается двойным щелчком левой кнопки мыши на пиктограмме блока, и его интерфейс аналогичен интерфейсу окна Scope (см. рис. 5.3) при нажатой кнопке floating scope, когда автоматически активизируются кнопки Lock/Unlock axes selection (Активировать/Дезактивировать выбор осей) и Signal selection (Выбор сигнала).

Параметры блока Floating Scope аналогичны параметрам блока Scope.

В отличие от блока Scope (осциллографа), блок Floating Scope (плавающий осциллограф) не имеет входов, и оперативность отображения сигналов обеспечивается путем автоматического, скрытого от пользователя, подсоединения к выходу соответствующего блока.

Для этого необходимо выполнить следующие действия:

- 1. Открыть окно Floating Scope.
- На панели инструментов нажать кнопку Signal selection, в открывшемся одноименном окне отметить флажком имя интересующего блока и закрыть окно в окне S-моделей будет автоматически создано соответствующее, скрытое от пользователя, подсоединение.
- Активизировать график (даже, если он один), в котором будет отображаться интересующий сигнал (щелчком левой кнопкой мыши на свободном поле), график будет обрамлен жирной синей линией.
- 4. Запустить моделирование в активизированном графике отобразится график сигнала.

### 5.2.6. Блок Terminator

Блок *Terminator* является "заглушкой" для неподключенного выхода в блоке с несколькими выходами. В отсутствие "заглушки" при моделировании выдается предупреждение. Блок не содержит параметров.

### 5.2.7. Блок То File

Блок *To File* экспортирует сигналы в файл данных с расширением mat. Для блока определяются следующие параметры (рис. 5.4):

□ Filename (Имя файла) — имя файла данных с расширением mat, хранящего значения экспортируемого сигнала (сигналов) в виде (4.4).

Если файл данных хранится в папке work, то достаточно указать его имя, а если в папке пользователя, то следует указать путь к файлу, например:

Step/fim

```
где Step — имя папки пользователя, а fim — имя файла с расширением mat.
```

- □ Variable name (Имя переменной) имя экспортируемой переменной (матрицы в (4.4));
- **Decimation (Децимация)** см. блок Display (см. разд. 5.2.1);
- **□** Sample time см. табл. 4.2.

📓 export_import_fi	ile * 📃 🗖 🔀
Eile Edit View Simula	ation F <u>o</u> rmat <u>T</u> ools <u>H</u> elp
	6 ⓑ 🛍   🗢 ⇒ ↔   오 오   ► = 10.0 Normal 💽
Sink Block Parameters: To File	
<ul> <li>To File</li> <li>Write time and input to specified MAT file in row format. Time is in row</li> </ul>	1. Sine Wave To File From File Scope
- Parameters Filename:	
signal.mat	
Variable name:	Impor From File signal
S Decimation:	
1	
Sample time (-1 for inherited):	
-1	
QK <u>C</u> ancel <u>H</u> elp	Apply -2 4 6 8 10
	Time offset: 0

Рис. 5.4. Параметры блока То File и примеры экспорта сигнала с именем S в файл данных с именем signal.mat и импорта того же сигнала с помощью блока From File

#### 5.2.8. Блок To Workspace

Блок *To Workspace* экспортирует сигналы в Workspace. Блок имеет следующие параметры (рис. 5.5):

- □ Variable name имя переменной, экспортируемой в Workspace (матрицы в (4.5));
- Limit data points to last (Предельное число точек для сохранения) предельное количество значений сигнала, сохраняемых в процессе моделирования; значение inf (по умолчанию) соответствует сохранению всех значений сигнала на интервале моделирования.

Если предельное количество значений сигнала будет меньше его длины (длины вектора), например, 10 при длине вектора 50, то в Workspace сохраняются *последние* 10 значений сигнала;

- **Decimation (Децимация)** см. блок Display (см. разд. 5.2.1);
- □ Sample time см. табл. 4.2;

- □ Save format (Формат сохранения) типы массивов для экспортируемого сигнала:
  - Structure With Time структура (массив записей) с тремя полями:
    - time вектор-столбец возрастающих значений времени;
    - signals вложенная структура с тремя полями: values (матрица, столбцы которой соответствуют значениям сигналов в моменты времени, заданные в поле time); dimensions (количество сигналов (скаляр)); label (зарезервированное поле для хранения информации пользователя в виде символической константы в апострофах);
    - blockName имя блока То Workspace в S-модели;
  - Structure (по умолчанию) структура, аналогичная Structure with, но с пустым полем time пустым вектором значений времени [];
  - Аггау массив значений сигнала;
- □ Log fixed point data as an fi object (Представление данных с фиксированной точкой в виде объекта fi) флаг, при установке которого данные (значения сигнала) будут представлены в виде объекта fi [22], а при сбросе (по умолчанию) отнесены к типу double.

exsport_signal_1	
Eile Edit View Simulation Format	<u>T</u> ools <u>H</u> elp
0 😂 🖬 🚭 🕺 🖻 🛍 🕅	⊨⇒☆   Ω 으   ▶ = 10.0 Nomal _
Sine Wave To Workspace	<ul> <li>Sink Block Parameters: To Workspace</li> <li>To Workspace</li> <li>Write input to specified array or structure in MATLAB's main workspace. Data is not available until the simulation is stopped or paused.</li> <li>Parameters</li> <li>Variable name:         <ul> <li>Y</li> <li>Limit data points to last:                  inf</li> <li>Decimation:                  1</li></ul></li></ul>
Ready 100%	ode45

Рис. 5.5. Параметры блока То Workspace и пример экспорта в Workspace сигнала с именем ч

На примере экспорта сигнала у с помощью блока То Workspace поясним, как будет меняться его размерность в зависимости от состояния флага Interpret vector parameters as 1-D в окне параметров блока Sine Wave.

#### Пример 5.1

Для блока Sine Wave (см. рис. 5.5) численные параметры зададим *скалярами*: Sine type: Time based; Amplitude: 1; Frequency: 1; Sample time: 0.

По завершении моделирования определим размерность сигнала у в зависимости от состояния флага Interpret vector parameters as 1-D (см. табл. 4.3):

□ при установке флага вектор параметров интерпретируется как одномерный (1-D), и сигнал у представляется в виде матрицы L×1, где L — количество значений сигнала (при заданном интервале моделирования L = 51):

```
>> size(Y)
ans =
51 1
```

При сбросе флага вектор параметров интерпретируется как матрица (2 - D), и сигнал у представляется в виде *трехмерного массива*  $1 \times 1 \times L$ :

```
>> size(Y)
ans =
1 1 51
```

#### Пример 5.2

Для блока Sine Wave (см. рис. 5.5) численные параметры зададим *векторами*: Sine type: Time based; Amplitude: [1 2 3]; Frequency: [1 3 5]; Sample time: 0.

По завершении моделирования определим размерность сигнала у в зависимости от состояния флага Interpret vector parameters as 1-D (см. табл. 4.3):

□ при установке флага каждый из трех (N = 3) векторов параметров интерпретируется как одномерный (1-D), и сигнал у представляется в виде матрицы L×N = L×3, где N = 3 — количество сигналов, а L — количество значений в каждом из них (при заданном интервале моделирования L = 51):

```
>> size(Y)
ans =
51 3
```

□ при сбросе флага три вектора параметров интерпретируются как матрица  $1 \times N = 1 \times 3$  (2-D), и сигнал у представляется в виде трехмерного массива  $1 \times N \times L = 1 \times 3 \times L$ : >> size(Y)

```
ans =
```

1 3 51

#### Пример 5.3

Для блока Sine Wave численные параметры зададим *матрицами*: Sine type: Time based; Amplitude: [1 2 5; 3 4 8]; Frequency: [1 3 1; 5 7 2]; Sample time: 0.

По завершении моделирования определим размерность сигнала у в зависимости от состояния флага Interpret vector parameters as 1-D (см. табл. 4.3).

В этом случае, независимо от *установки/сброса* флага, вектор параметров интерпретируется как матрица  $M \times N = 2 \times 3$  (2-D), и сигнал у представляется в виде *трехмерного массива*  $M \times N \times L = 2 \times 3 \times L$ , где  $M \times N = 2 \times 3$  — количество сигналов, а L — количество значений в каждом из них:

```
>> size(Y)
ans =
2 3 51
```

#### 5.2.9. Блок XY Graph

Блок *XY Graph* имитирует графопостроитель — отображает двумерные графики в декартовой системе координат. Блок имеет следующие параметры (рис. 5.6):

□ **х-тіп и х-тах** — границы по оси абсцисс;

**у-тіп и у-тах** — границы по оси ординат;

□ Sample time — *см. разд. 4.2.1.* 

Блок имеет два входа: верхний — для ввода значений аргумента, нижний — для ввода значений функции. График выводится автоматически с помощью функции MATLAB plot [22].

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  Blocks  $\rightarrow$  Sinks (Simulink  $\rightarrow$  Блоки  $\rightarrow$  Приемники).



Рис. 5.6. Параметры блока XY Graph и график сигнала на выходе блока Band-Limited White Noise

# Глава 6



# Математические преобразования

В этой главе рассматриваются средства Simulink, входящие в состав библиотек блоков Simulink и Signal Processing Blockset и предназначенные для моделирования математического преобразования сигналов, когда оператор F в (2.1) определяется конкретным преобразованием (см. разд. 2.1).

# 6.1. Средства библиотеки блоков Simulink

В этом разделе познакомимся со средствами моделирования математического преобразования из библиотеки блоков Simulink.

#### 6.1.1. Арифметические операции

S-модели *арифметических операций* представлены в группе блоков Math Operations (Математические операции) (табл. 6.1).

N⁰	Блок	Операция	Операция в MATLAB
1.	Add	Сложение матричное или поэлементное	+
2.	Bias	Добавление к входному сигналу постоянной составляющей (смещения)	+
3.	Divide	Деление матричное или поэлементное <sup>1</sup>	./ или /
4.	Dot Prod- uct (mask)	Скалярное произведение векторов	sum(conj(u1).*u2) u1 — <b>верхний вход</b> ; u2 — <b>нижний вход</b>
5.	Gain	Поэлементное или матричное умножение на коэффициент (скаляр, вектор или матрицу)	.* или *
6.	Product	Умножение матричное или поэлементное <sup>2</sup>	.* ИЛИ *

Таблица 6.1. Группа блоков Math Operations

<sup>&</sup>lt;sup>1</sup> Может использоваться для умножения.

<sup>&</sup>lt;sup>2</sup> Может использоваться для деления.

Окончание табл. 6.1

N⁰	Блок	Операция	Операция в MATLAB
7.	Product of Elements	Произведение элементов матрицы	prod((prod(A)))
8.	Slider Gain	Умножение на константу, величина которой задается положением ползунка	.* или *
9.	Subtract	Вычитание матричное или поэлементное	-
10.	Sum	Сложение матричное или поэлементное	+
11.	Sum of Elements	Сумма элементов матрицы	sum((sum(A)))
12.	Unary Mi- nus (mask)	Унарный минус	-

#### 6.1.1.1. Блок Add

Блок *Add* выполняет матричное или поэлементное сложение. Параметры блока задаются на двух вкладках — Main и Signal Attributes (рис. 6.1).

На вкладке Main определяются параметры:

- □ Icon shape (Форма пиктограммы) вид пиктограммы: прямоугольник (rectangular) или окружность (round);
- □ List of signs (Список знаков) последовательность знаков слагаемых плюс (+) и минус (-), между которым, при желании, можно ставить разделитель ( | );
- □ Sample time см. разд. 4.2.1.

На вкладке Signal Attributes определяются параметры:

- Require all inputs to have the same data type (Потребовать, чтобы входные сигналы принадлежали к одинаковому типу данных) флаг, при установке которого входные сигналы (слагаемые) должны принадлежать к одинаковому типу данных, в противном случае будет выдано сообщение об ошибке;
- Accumulator data type (Тип данных для аккумулятора) тип данных для суммы (см. табл. 4.5);
- □ Output minimum и Output maximum см. разд. 4.2.4;
- □ Output data type (Тип данных на выходе) тип данных на выходе (см. табл. 4.5);
- □ Lock output data type setting against changes by the fixed-point tools *cm. pa3∂. 4.2.4*;
- □ Integer rounding mode (Режим целочисленного округления) режим округления при преобразовании вещественных данных в целые и округлении данных с ФТ:
  - Floor округление в направлении <sub>-∞</sub> округление до ближайшего целого в сторону уменьшения.

```
Эквивалентно функции MATLAB floor:
>> floor([8.2 8.5 8.7 -8.2 -8.5 -8.7])
ans =
```

```
8 8 8 -9 -9 -9
```

Для данных с ФТ положительное число усекается, а для отрицательного берется значение ближайшего нижнего уровня квантования;

• Selling — округление в направлении ∞ — округление до ближайшего целого в сторону увеличения.

Эквивалентно функции MATLAB ceil:

Для данных с ФТ отрицательное число усекается, а для положительного берется значение ближайшего верхнего уровня квантования;

• Convergent — округление до ближайшего четного целого. При дробной части числа, равной 0.5, — в сторону ближайшего четного числа.

Эквивалентно функции MATLAB convergent:

```
>> convergent([8.2 8.5 8.7 -8.2 -8.5 -8.7])
ans =
8 8 9 -8 -8 -9
```

Для данных с ФТ выбирается значение ближайшего уровня квантования, и если оно попадает точно на границу между соседними уровнями квантования, то выбирается значение ближайшего уровня, соответствующего четному числу;

Nearest — округление до ближайшего целого. При дробной части числа, равной 0.5, — в сторону увеличения.

Эквивалентно функции MATLAB nearest:

Для данных с ФТ выбирается значение ближайшего уровня квантования, и если оно точно попадает на границу между соседними уровнями, берется значение ближайшего верхнего уровня;

• Round — округление до ближайшего целого. При дробной части, равной 0.5, — в сторону увеличения модуля числа.

Эквивалентно функции MATLAB round:

Для данных с ФТ 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;

- Simplest автоматический выбор оптимального режима округления посредством комбинации режимов округления для данных с ФТ;
- Zero округление в направлении нуля усечение дробной части.

```
Эквивалентно функции MATLAB fix:
>> fix([8.2 8.5 8.7 -8.2 -8.5 -8.7])
ans =
8 8 8 -8 -8 -8 -8
```

Для данных с ФТ выбирается значение ближайшего нижнего уровня квантования (тождественно усечению);

□ Saturate on integer overflow (Насыщение до целого при переполнении) — флаг, при установке которого используется арифметика насыщения для данных с ФТ: при переполнении результат автоматически заменяется максимально возможным (по модулю) для формата слово.



Рис. 6.1. Параметры блока Add и пример сложения двух матриц

#### 6.1.1.2. Блок Bias

Блок *Bias* добавляет к входному сигналу *и* постоянную составляющую и формирует выходной сигнал *у* со смещением *bias*:

$$y = u + bias.$$

Параметры блока включают в себя (рис. 6.2):

□ **Bias** (Смещение) — смещение *bias*;

□ Saturate on integer overflow — *см. разд. 6.1.1.1.* 



Рис. 6.2. Параметры и пример использования блока Bias при смещении, равном 1.5

#### 6.1.1.3. Блок Divide

Блок *Divide* выполняет матричное или поэлементное деление. Параметры блока задаются на двух вкладках — Main и Signal Attributes.

На вкладке Main определяются параметры (рис. 6.3, *a*):

- □ Number inputs (Количество входов) количество входов блока; при выполнении операции деления с двумя операндами задается (по умолчанию) как "\*/";
- □ Multiplication (Умножение) способ деления (умножения): поэлементное левое Element-wise (.\*) и матричное левое Matrix (\*);

В MATLAB *поэлементное* левое деление эквивалентно алгебраическому поэлементному делению, а *матричное* левое — умножению u1\*inv(u2), т. е. операции u1\*u2<sup>-1</sup>, где u1, u2 — сигналы (матрицы) на верхнем и нижнем входах, а  $u2^{-1}$  — матрица, обратная к матрице u2;

□ Sample time — см. разд. 4.2.1.

На вкладке Signal Attributes параметры подобны параметрам блока Add (см. разд. 6.1.1.1).

На рис. 6.3, б, в приведены примеры исползьзования блока Divide.

Product Multiply the folk a) * or b) scala When t or divid If / is sp	ction Block Parameters: Divide it y or divide inputs. Choose element-wise or matrix product and speci- lowing: / for each input port (e.g., ""/") ar specifies the number of input ports to be multiplied there is only one input port for element-wise product, multiply de elements over all dimensions or one specified dimension specified with matrix product, compute the inverse of the correspond	fy one of		
Main	Main Signal Attributes			
Number	r of inputs:			
=/				
Multiplica	cation: Element-wise(.**)	~		
Sample	time (-1 for inherited):			
-1				
	QK <u>C</u> ancel <u>H</u> elp	Apply		



Рис. 6.3. Параметры блока Divide на вкладке Main (а) и примеры использования: поэлементное левое деление (б) и матричное левое деление (е)

#### 6.1.1.4. Блок Dot Product

Блок *Dot Product* вычисляет скалярное произведение векторов. На вкладке **Main** задается один параметр — **Sample time** (*см. разд. 4.2.1*). На вкладке **Signal Attri-butes** параметры подобны параметрам блока Add (*см. разд. 6.1.1.1*).

Пример использования блока Dot Product приведен на рис. 6.4.



Рис. 6.4. Пример вычисления скалярного произведения векторов с помощью блока Dot Product

#### 6.1.1.5. Блок Gain

Блок *Gain* выполняет поэлементное или матричное умножение на коэффициент (скаляр, вектор или матрицу). Параметры блока задаются на трех вкладках — **Main**, **Signal Attributes** и **Parameter Attributes** (Атрибуты параметра).

На вкладке Main определяются параметры (рис. 6.5, *a*):

□ Gain (Усиление) — коэффициент усиления к входного сигнала u, задаваемый в виде скаляра, вектора или матрицы;

□ Multiplication (Умножение) — варианты умножения:

- Element-wise (К.\*и) поэлементное на скаляр к;
- Matrix (K\*u) поэлементное на матрицу к;
- Matrix (u\*K) матричное u\*к;
- Matrix (K\*u) (u vector) матричное умножение вектора u на столбцы матрицы к;

□ Sample time — см. разд. 4.2.1.

На вкладке Signal Attributes параметры подобны параметрам блока Add (см. разд. 6.1.1.1).

На вкладке Parameter Attributes задаются параметры (рис. 6.5, *a*):

- □ Parameter minimum (Минимальное значение параметра) и Parameter maximum (Максимальное значение параметра) минимальное и максимальное значения коэффициента усиления к подобны параметрам Output minimum и Output maximum для блока Constant (*см. разд. 4.2.4*);
- □ Parameter data type тип данных для коэффициента к (см. табл. 4.5).

На рис. 6.5, б, в, г приведены примеры использования блока Gain.





Рис. 6.5. Параметры блока Gain (a) и примеры: поэлементного умножения на скаляр (б), поэлементного умножения на матрицу (в) и матричного умножения (г)

#### 6.1.1.6. Блок Product

Блок *Product* выполняет матричное или поэлементное умножение. Параметры блока задаются на двух вкладках — Main и Signal Attributes.

На вкладке Main определяются параметры (рис. 6.6, *a*):

□ Number of inputs (Количество входов) — количество входов, которое при выполнении операции умножения с двумя операндам равно 2 (по умолчанию).

При одном входе вычисляется произведение элементов матрицы;

- □ **Multiplication** (Операция умножения) варианты умножения: поэлементное Element-wise (.\*) и матричное Matrix (\*);
- □ Sample time *см. разд. 4.2.1*.

На вкладке Signal Attributes параметры подобны параметрам блока Add (см. разд. 6.1.1.1).

На рис. 6.6, *б*, *в*, *г* приведены примеры использования блока Product.

Product Multiply of the follow a) * or / f b) scalars When the or divide of If / is spe	r divide inputs. Ch ing: for each input port specifies the numb re is only one inpu elements over all d cified with matrix p	oose element-wise or matrix product and specify one of (e.g., ""/") er of input ports to be multiplied t port for element-wise product, multiply mensions or one specified dimension roduct, compute the inverse of the corresponding input.		
Main	Main Signal Attributes			
Number of	finputs:			
2				
Multiplicati	on: Element-wise(	.*)		
Sample tim	ne (-1 for inherited)	:		
-1				
		QK Cancel Help Apply		



2 1 3 4 19 22 Constant Matrix 43 50 Aultiply Product Display 5 6 7 8 Constant 1 1 2 24 3 4

в

г

Рис. 6.6. Параметры блока Product при открытой вкладке Main (*a*) и примеры: поэлементного умножения (б); матричного умножение (в); произведения элементов матрицы (*a*)

Display

Product

Constant

#### 6.1.1.7. Блок Product of Elements

Блок *Product of Elements* вычисляет произведение элементов матрицы и по умолчанию имеет один вход при значении параметра **Number of inputs**: \* (рис. 6.6, *a*). При нескольких входах представляет собой реализацию блока Product.

#### 6.1.1.8. Блок Slider Gain

Блок *Slider Gain* выполняет умножение на константу, величина которой задается положением ползунка. Блок имеет два параметра (рис. 6.7): **Low** (Нижний) и **High** (Верхний) — минимальное и максимальное значения коэффициента усиления.

Положение ползунка отображается в центральном поле ввода.



Рис. 6.7. Параметры и пример использования блока Slider Gain

#### 6.1.1.9. Блок Subtract

Блок *Subtract* выполняет матричное или поэлементное вычитание. Блок представляет собой реализацию блока Add при наличии символа минус (–) в списке знаков List of signs (см. разд. 6.1.1.1).

#### 6.1.1.10. Блок Sum

Блок *Sum* выполняет матричное или поэлементное сложение. Это реализация блока Add (см. разд. 6.1.1.1).

#### 6.1.1.11. Блок Sum of Elements

Блок Sum of Elements вычисляет сумму элементов матрицы. Это реализация блока Add (см. разд. 6.1.1.1) при одном входе.

#### 6.1.1.12. Блок Unary Minus

Блок Unary Minus выдает значения входного сигнала с противоположным знаком (унарный минус). Блок имеет два параметра:

```
□ Sample time — см. разд. 4.2.1;
```
□ Saturate to max or min when overflows occur (Насыщение до максимального или минимального значения при переполнении) — флаг, при установке которого используется арифметика насыщения для данных с ФТ.

Пример использования блока Unary Minus приведен на рис. 6.8.



Рис. 6.8. Пример использования блока Unary Minus

## 6.1.2. Элементарные математические функции

S-модели элементарных математических функций представлены в группе блоков Math Operations (табл. 6.2).

N⁰	Блок	Операция	Функция MATLAB
1.	Abs	Вычисление абсолютного значения (модуля) и —  и	abs (u)
2.	Math Func- tion	Вычисление элементарных математических функций	см. табл. 6.3
3.	Sign	Определение знака числа: 1 — положительное, -1 — отрицательное, 0 — нуль	sign(u)
4.	Trigonome- tric Func- tion	Вычисление тригонометрических функций	см. табл. 6.4

Таблица 6.2. Блоки элементарных математических функций

Параметры блоков из табл. 6.2 приводятся далее, а с использованием блоков предлагаем познакомиться самостоятельно.

### 6.1.2.1. Блок Abs

Блок *Abs* вычисляет абсолютное значение (модуль) значений входного сигнала. Параметры блока задаются на двух вкладках — Main и Signal Attributes.

На вкладке Main определяются параметры:

□ Enable zero crossing detection — см. разд. 4.2.9;

□ Sample time — *см. разд. 4.2.1.* 

При открытой вкладке Signal Attributes параметры подобны параметрам блока Add (см. разд. 6.1.1.1).

### 6.1.2.2. Блок Math Function

Блок *Math Function* вычисляет значения входного сигнала согласно указанной элементарной математической функции. Параметры блока задаются на трех вкладках — Main, Signal Attributes и Intermediate Attributes (Промежуточные атрибуты).

На вкладке Main определяются параметры:

□ Function (Функция) — элементарная математическая функция (табл. 6.3);

Для ряда функций активизируется вкладка Intermediate Attributes с параметрами, позволяющими выбрать итерационный метод их вычисления и число итераций, рекомендуется оставлять значение по умолчанию;

Output signal type (Тип сигнала на выходе) — тип данных на выходе: auto (совпадает с типом данных на входе), real (вещественный), complex (комплексный);

□ Sample time — см. разд. 4.2.1.

На вкладке Signal Attributes параметры подобны параметрам блока Add (см. разд. 6.1.1.1).

Значение параметра Function	Функция	Операция в MATLAB
exp	Экспонента — е <sup>и</sup>	exp (u)
log	Натуральный логарифм — In( <i>u</i> )	log (u)
10^ <i>u</i>	Степень 10 — 10 <sup>и</sup>	10.^u
log10	Десятичный логарифм — lg(u)	log10(u)
magnitude^2	Квадрат модуля —   <i>u</i>   <sup>2</sup>	(abs(u)).^2
square	Квадрат — и <sup>2</sup>	u.^2
sqrt	Корень квадратный — $\sqrt{u}$	sqrt(u)
pow	Степень — и <sup>и</sup>	u.^v
conj	Комплексно сопряженное число	conj(u)
reciprocal	Частное от деления 1/и	1./u
hypot	Корень квадратный из суммы квадратов — $\sqrt{u^2+v^2}$	sqrt(u.^2+v.^2)
rem	Остаток от деления со знаком для числа <i>и</i> по модулю <i>v</i>	rem(u,v)
mod	Число <i>и</i> по модулю <i>v</i> — mod <sub>v</sub> <i>u</i>	mod(u,v)
transpose	Транспонирование	' (апостроф) — для вещественной матрицы; .' — для комплексной матрицы
hermitian	Эрмитово сопряжение	' — для вещественной и комплексной матриц

Таблица 6.3. Элементарные математические функции блока Math Function

### 6.1.2.3. Блок Sign

Блок *Sign* определяет знак значений вещественного сигнала: 1 — для положительного числа и –1 — для отрицательного. Блок имеет следующие параметры:

□ Enable zero crossing detection — *см. разд. 4.2.9*;

□ Sample time — см. разд. 4.2.1.

### 6.1.2.4. Блок Trigonometric Function

Блок *Trigonometric Function* вычисляет значения входного сигнала согласно указанной тригонометрической функции. Блок имеет следующие параметры:

□ Function — тригонометрическая функция (табл. 6.4);

□ **Output signal type** — *см. разд. 6.1.2.2*;

□ Sample time — *см. разд. 4.2.1*.

Значение параметра Function	Функция	Функция MATLAB
sin	Синус — sin <i>x</i>	sin(X)
cos	Косинус — cos <i>x</i>	cos(X)
tan	Тангенс — tg <i>x</i>	tan(X)
asin	Арксинус — arcsin x	asin(X)
acos	Арккосинус — arccos <i>x</i>	acos (X)
atan	Арктангенс — arctg x	atan(X)
atan2	Четырехквадрантный арктангенс	atan2(X,Y)
sinh	Гиперболический синус — sh <i>x</i>	sinh(X)
cosh	Гиперболический косинус — ch x	cosh(X)
tanh	Гиперболический тангенс — th x	tanh(X)
asinh	Гиперболический ареа-синус — arsh x	asinh(X)
acosh	Гиперболический ареа-косинус — arch x	acosh(X)
atanh	Гиперболический ареа-тангенс — arth x	atanh(X)

Таблица 6.4. Тригонометрические функции блока Trigonometric Function

### 6.1.3. Операции с комплексными числами

S-модели *операций с комплексными числами* представлены в группе блоков **Math Operations** (табл. 6.5).

N⁰	Блок	Операция	Операция в MATLAB
1.	Complex to Magnitude- Angle	Вычисление модуля и аргумента комплексного числа	abs(X) angle(X)
2.	Complex to Real-Imag	Выделение вещественной и мнимой частей комплексного числа	real(X) imag(X)
3.	Magnitude-Angle to Complex	Вычисление комплексного числа по его модулю и аргументу	R.*exp(j.*F)
4.	Real-Imag to Complex	Представление комплексного числа по его вещественной и мнимой частям	complex(X1,X2)

Таблица 6.5. Блоки операций с комплексными числами

### 6.1.3.1. Блок Complex to Magnitude-Angle

Блок *Complex to Magnitude-Angle* вычисляет модуль и аргумент значений комплексного сигнала. Параметры блока следующие:

**Оиtput** (Выход) — результат вычислений:

- Magnitude and angle модуль и аргумент в радианах;
- Magnitude модуль;
- Angle аргумент в радианах;
- □ Sample time см. разд. 4.2.1.

Пример использования блока приведен на рис. 6.9.





### 6.1.3.2. Блок Complex to Real-Imag

Блок *Complex to Real-Imag* выделяет вещественную и мнимую части значений комплексного сигнала. Параметры блока следующие:

Output — результат вычислений:

- Real and imag вещественная и мнимая части;
- Real вещественная часть;
- Imag мнимая часть;

□ Sample time — см. разд. 4.2.1.

Пример использования блока приведен на рис. 6.10.



Рис. 6.10. Пример выделения вещественной и мнимой частей комплексного числа с помощью блока Complex to Real-Imag (Output: Real and imag)

### 6.1.3.3. Блок Magnitude-Angle to Complex

Блок *Magnitude-Angle to Complex* вычисляет значения комплексного сигнала по их модулю и аргументу. Параметры блока следующие:

- □ Input (Вход) см. параметр Output блока Complex to Magnitude-Angle (см. разд. 6.1.3.1);
- □ Sample time см. разд. 4.2.1.

Пример использования блока приведен на рис. 6.11.



Рис. 6.11. Пример вычисления комплексного числа по его модулю и аргументу с помощью блока Magnitude-Angle to Complex (Input: Magnitude and angle)

### 6.1.3.4. Блок Real-Imag to Complex

Блок *Real-Imag to Complex* выдает значения комплексного сигнала, представленные по их вещественной и мнимой частям. Параметры блока следующие: **Input** — см. параметр **Output** блока Complex to Real-Imag (*см. разд. 6.1.3.2*);

□ Sample time — см. разд. 4.2.1.

Пример использования блока приведен на рис. 6.12.



Рис. 6.12. Пример представления комплексного числа по его вещественной и мнимой частям с помощью блока Real-Imag to Complex (Input: Real and imag)

# 6.1.4. Округление с помощью блока Rounding Function

S-модели функций округления реализуются блоком Rounding Function в группе блоков Math Operations. Параметры блока следующие:

□ Function — функция (табл. 6.6); для матрицы выполняется поэлементно;

□ Sample time — см. разд. 4.2.1.

Значение параметра	Назначение	Функция MATLAB
floor	Округление в направлении –∞ — округление до ближайшего целого в сторону уменьшения	floor(X)
ceil	Округление в направлении ∞ — округление до ближайшего целого в сторону увеличения	ceil(X)
round	Округление до ближайшего целого — при дробной части, равной 0.5, — в сторону увеличения модуля числа	round(X)
fix	Округление в направлении нуля — усечение дробной части числа	fix(X)

Таблица 6.6. Функции округления блока Rounding Function

Пример использования блока приведен на рис. 6.13.



Рис. 6.13. Пример округления с помощью блока Rounding Function для функции floor

# 6.1.5. Операции с матрицами и векторами

S-модели *операций с матрицами и векторами* представлены в группе блоков **Math Operations** (табл. 6.7).

Таблица 6.7. Блоки операций с матрицами и векторами

N⁰	Блок	Операция
1.	Matrix Concatenate	Конкатенация <sup>1</sup> матриц и векторов
2	MinMax	Определение максимального (минимального) элемента вектора
3.	Reshape	Преобразование размерности массива
4.	Vector Concatenate	Конкатенация векторов и матриц

### 6.1.5.1. Блок Matrix Concatenate

Блок *Matrix Concatenate* выполняет конкатенацию матриц и векторов. Блок имеет следующие параметры:

□ Number of inputs (Количество входов) — количество объединяемых матриц;

□ Mode (Режим) — режим конкатенации:

- Multidimensional объединение матриц или матрицы с вектором;
- Vector объединение векторов;
- □ Concatenate dimension (Размерность конкатенации) способ конкатенации:
  - 2 горизонтальная конкатенация матриц с одинаковым числом строк;
  - 1 вертикальная конкатенация матриц с одинаковым числом столбцов.

Примеры использования блока приведены на рис. 6.14.





<sup>1</sup> Объединение.

### 6.1.5.2. Блок MinMax

Блок *MinMax* определяет максимальный или минимальный элемент вектора. Параметры блока задаются на двух вкладках — **Main** и **Signal Attributes**.

На вкладке Main задаются параметры:

- □ Function функция: max (максимум) или min (минимум);
- □ Number of input ports (Количество входных портов) количество входов.

При одном входе определяется максимальный (минимальный) элемент вектора, а при нескольких — максимальные (минимальные) элементы среди элементов векторов с одинаковыми индексами;

- □ Enable zero crossing detection *см. разд. 4.2.9*;
- □ Sample time см. разд. 4.2.1.

На вкладке Signal Attributes параметры подобны параметрам блока Add (см. разд. 6.1.1.1).

Примеры использования блока приведены на рис. 6.15.





### 6.1.5.3. Блок Reshape

Блок *Reshape* выполняет преобразование размерности массива. Параметры блока следующие:

□ Output dimensionality (Размерность на выходе) — варианты размерности на выходе:

• 1-D array (Одномерный) — растягивание матрицы в вектор по столбцам (1-D);

- Column Vector (2-D) (Вектор-столбец) растягивание матрицы в вектор по столбцам, который трактуется как *столбец* новой матрицы (2-D);
- Row Vector (2-D) (Вектор-строка) растягивание матрицы в вектор по строкам, который трактуется как *строка* новой матрицы (2-D);
- Сustomize (Произвольная) формирование массива произвольной размерности;
- □ Output dimensions (Размер выходного массива) размерность массива при выборе параметра Output dimensionality: Customize.

Примеры использования блока приведены на рис. 6.16.



Рис. 6.16. Примеры использования блока Reshape при различных значениях параметра Output dimensionality: 1-D array (a), Row Vector (б) и Customize для матрицы 2×2 (в)

# 6.1.5.4. Блок Vector Concatenate

Блок *Vector Concatenate* выполняет конкатенацию векторов и матриц. Это реализация блока Matrix Concatenate (см. разд. 6.1.5.1).

# 6.1.6. Вычисление корней уравнения и значений многочлена

S-модели вычисления корней уравнения и значений многочлена операций представлены в группе блоков Math Operations (табл. 6.8).

N⁰	Имя блока	Операция	Функция MATLAB
1.	Algebraic Constraint (mask)	Решатель алгебраического уравнения	frezo
2.	Polynomial	Вычисление значений многочлена	polyval

Таблица 6.8. Блоки вычисления корней уравнения и значений многочлена

### 6.1.6.1. Блок Algebraic Constraint

Блок Algebraic Constraint представляет собой решатель алгебраического уравнения. Он содержит один параметр **Initial guess** (Начальное приближение) начальная точка итерационной процедуры вычисления корня — ближайшее к корню значение *слева* от него.

На рис. 6.17 приведен пример решения уравнения  $z^3 - 8 = 0$  в окрестности точки 0 (Initial guess: 0). Для блока Math Function (*см. разо. 6.1.2.2*) выбрана функция **Function**: pow.

Если алгебраическое уравнение имеет несколько корней, то предварительно рекомендуется определить области, внутри которых существует один и только один корень (области, внутри которых функция меняет знак), например, построив график функции в MATLAB.



Рис. 6.17. Пример вычисления корня уравнения с помощью блока Algebraic Constraint

### 6.1.6.2. Блок Polynomial

Блок *Polynomial* вычисляет значения многочлена и содержит один параметр **Polynomial Coefficients** (Коэффициенты многочлена) — коэффициенты многочлена в порядке убывания степеней — вектор коэффициентов, по умолчанию типа double.

На рис. 6.18 приведен пример вычисления значений полинома  $0.5z^3 + 0.7z^2 - z + 8$  (**Polynomial Coefficients**: [0.5 0.7 –1 8]) в точках, заданных вектором [1 2 3] на выходе блока Constant; последние могут задаваться скаляром, вектором или матрицей.



Рис. 6.18. Пример вычисления значений полинома с помощью блока Polynomial

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  Blocks  $\rightarrow$  Math Operations.

### 6.1.7. Операции отношения и логические операции

S-модели *операций отношения и логических операций* представлены в группе блоков Logic and Bit Operations (Логические и побитовые операции) (табл. 6.9).

Nº	Блок	Операция
1.	Compare To Constant (mask)	Сравнение с константой
2.	Compare To Zero (mask)	Сравнение с нулем
3.	Relational Operator	Сравнение двух операндов
4.	Logic Operator	Логические операции

Таблица 6.9. Блоки символов операций отношения и логических операций

### 6.1.7.1. Блок Compare To Constant

Блок *Compare To Constant* выполняет сравнение с константой. Блок имеет следующие параметры:

- □ **Operator** (Оператор) символ операции отношения, такой же, как в MATLAB (табл. 6.10);
- Constant Value (Константа) константа (скаляр, вектор или матрица), с которой сравниваются значения входного сигнала и по результатам сравнения выдается 1, если условие выполняется, и 0, если не выполняется. Если константа задана вектором или матрицей, то входной сигнал должен совпадать с ней по размерности и размеру;
- Output data type mode (Режим типа данных на выходе) тип данных на выходе: uint8 или boolean (см. табл. 4.4);
- □ Enable zero crossing detection *см. разд. 4.2.9.*

Габлица 6.10.	Символы	операций	отношения
---------------	---------	----------	-----------

Символ операции	Операция
==	Равно
~=	Не равно
<	Меньше
>	Больше
<=	Меньше либо равно
>=	Больше либо равно

Пример использования блока приведен на рис. 6.19.



Рис. 6.19. Пример сравнения элементов вектора с константой 3 с помощью блока Compare To Constant

### 6.1.7.2. Блок Compare To Zero

Блок *Compare To Zero* выполняет сравнение с нулем. Он имеет те же параметры, что и блок Compare To Constant *(см. разд. 6.1.7.1)*, но параметр **Constant Value** отсутствует, т. к. сравнение по умолчанию производится с нулем.

### 6.1.7.3. Блок Relational Operator

Блок *Relational Operator* выполняет поэлементное сравнение двух операндов. Параметры блока задаются на двух вкладках — Main и Signal Attributes.

На вкладке Main определяются параметры:

□ Relational Operator (Оператор отношения) — символ операции отношения (см. табл. 6.10).

Блок имеет два входа: верхний — для операнда *слева* от операции отношения, и нижний — *справа*;

□ Enable zero crossing detection — *см. разд. 4.2.9*;

□ Sample time — см. разд. 4.2.1.

При открытой вкладке Signal Attributes параметры подобны параметрам блока Add (см. разд. 6.1.1.1).

Пример использования блока приведен на рис. 6.20.



Рис. 6.20. Пример поэлементного сравнения векторов с помощью блока Relational Operator

## 6.1.7.4. Блок Logic Operator

Блок *Logic Operator* выполняет логические операции. Параметры блока задаются на двух вкладках — Main и Signal Attributes.

На вкладке Main определяются параметры:

**Орегаtor** (Оператор) — символ логической операции (табл. 6.11).

Операндами могут быть скаляры, векторы одинаковой длины и матрицы одинакового размера, для которых логические операции выполняются *поэлементно*;

- □ Number of input ports (Количество входных портов) количество входов;
- □ Icon shape (Форма пиктограммы) вид пиктограммы: rectangular (прямоугольник) или round (окружность);

□ Sample time — см. разд. 4.2.1.

На вкладке Signal Attributes параметры подобны параметрам для блока Add (см. разд. 6.1.1.1).

Символ операции	Операция	Функция MATLAB
AND	И — истина, если оба операнда не равны нулю	and(X,Y)
OR	ИЛИ — истина, если хотя бы один из операндов не равен нулю	or(X,Y)
NAND	И-НЕ — истина, если оба операнда равны нулю	—
NOR	ИЛИ-НЕ — истина, если хотя бы один из операндов равен нулю	_
XOR	Исключающее ИЛИ — истина, если только один из операндов равен нулю	xor(X,Y)
NOT	НЕ — все операнды, не равные нулю, заменяются нулями, а все <i>равные</i> нулю — единицами	not(X)

**Таблица 6.11**. Логические операции блока Logic Operator

Пример использования блока приведен на рис. 6.21.



Рис. 6.21. Пример выполнения операции NAND с помощью блока Logic Operator

### 6.1.8. Побитовые операции

S-модели *побитовых операций* представлены в группе блоков Logic and Bit Operations (табл. 6.12).

Таблица 6.12.	Блоки побитовых	операций
---------------	-----------------	----------

N⁰	Блок	Операция	Функция MATLAB
1.	Bit Clear (mask)	Обнуление бита	_
2.	Bit Set (mask)	Установка бита	bitset(X,N), N — номер бита
3.	Bitwise Opera- tor (mask)	Побитовые операции	<pre>bitand(X,Y) bitor(X,Y) bitxor(X,Y)</pre>
4.	Shift Arithmetic (mask)	Арифметический сдвиг битов и/или двоичной точки	bitshift(X,K), К — положительное или отрицательное число, равное количеству битов, на которое происходит сдвиг, соответственно влево или вправо

Побитовые операции выполняются над *целыми беззнаковыми десятичными* числами типа uint8, uint16 или uint32 (см. табл. 4.5), и результат операции является *целым беззнаковым десятичным* числом того же типа. Операндами могут быть скаляры, векторы одинаковой длины и матрицы одинакового размера.

### 6.1.8.1. Блок Bit Clear

Блок *Bit Clear* содержит один параметр **Index of bit** (Индекс бита) — номер обнуляемого бита, считая от нуля, справа налево (нулевой номер соответствует младшему значащему биту).

В примере на рис. 6.22 в десятичном числе 5 (двоичное — 101) обнуляется второй (старший) бит. Получаем десятичное число 1 (двоичное — 001).



Рис. 6.22. Пример обнуления второго бита в числе 5 с помощью блока Bit Clear

### 6.1.8.2. Блок Bit Set

Блок *Bit Set*, подобно блоку Bit Clear, содержит один параметр **Index of bit** — номер устанавливаемого бита.

В примере на рис. 6.23 в десятичном числе 5 (двоичное — 101) устанавливается первый бит. Получаем десятичное число 7 (двоичное — 111).



Рис. 6.23. Пример установки первого бита в числе 5 с помощью блока Bit Set

### 6.1.8.3. Блок Bitwise Operator

Блок *Bitwise Operator* выполняет побитовые операции. Блок имеет следующие параметры:

- □ **Operator** (Оператор) символ побитовой операции (см. табл. 6.11, где слово "операнд" следует заменить на "бит");
- □ Use bit mask (Использовать битовую маску) флаг маскирования чисел.

При установке флага блок имеет один вход и выполняет замену десятичных чисел на входе другими числами (скрывает числа под маской — маскирует), которые формируются в результате выполнения побитовой операции с десятичным числом, указываемым в поле **Bit Mask**;

□ Number of input ports — количество входов блока.

В примере на рис. 6.24 выполняется побитовая операция AND поэлементно с векторами [5 3] и [7 5] (двоичные — [101 011] и [111 101]) при сброшенном флаге Use bit mask. Результат операции — вектор [5 1] (двоичный — [101 001]).



Рис. 6.24. Пример выполнения побитовой операции AND с помощью блока Bitwise Operator

### 6.1.8.4. Блок Shift Arithmetic

Блок *Shift Arithmetic* выполняет арифметический сдвиг битов и/или двоичной точки. Блок имеет следующие параметры:

- □ Number of bits to shift right (use negative value to shift left) (Количество битов, на которое происходит сдвиг вправо (используйте отрицательное значение для сдвига влево)); в отсутствие сдвига указывается ноль;
- □ Number of places by which binary point shift right (use negative value to shift left) (Количество позиций, на которое двоичная точка сдвигается вправо (используйте отрицательное значение для сдвига влево)); в отсутствие сдвига указывается ноль.

На примере на рис. 6.25 выполняется сдвиг вправо на один бит в числе 5 (двоичное — 101). Первый параметр равен 1, а второй — 0. Получаем десятичное число 2 (двоичное — 010).



Рис. 6.25. Пример сдвига вправо на один бит в числе 5 с помощью блока Shift Arithmetic

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  Blocks  $\rightarrow$  Logic and Bit Operations.

# 6.1.9. Объекты MATLAB

В Simulink предусмотрена возможность включения в S-модель системы следующих объектов MATLAB [22]:

□ выражение;

□ встроенная функция;

□ внешняя функция<sup>1</sup>.

Данные объекты представлены S-моделями в группе блоков User-Defined Functions (Функции, определяемые пользователем) (табл. 6.13).

N⁰	Блок	Операция
1.	Fcn	Вычисление арифметического или логического выражения
2.	MATLAB Fcn	Вычисление арифметического выражения, встроенной или внешней функции с одним выходным параметром
3.	Embedded MATLAB Function	Вычисление создаваемой в блоке внешней функции

Таблица 6.13. Блоки объектов MATLAB

<sup>&</sup>lt;sup>1</sup> Function-файл.

### 6.1.9.1. Блок Fcn

Блок Fcn вычисляет арифметическое или логическое выражение.

Входным сигналом является константа в виде скаляра, вектора или матрицы, элементы которых трактуются как *простые переменные* со стандартными именами: и — для скаляра; u(1), u(2) и т. д. — для вектора; u(1,1), u(1,2) и т. д. — для матрицы.

Для входного сигнала, поступающего из Workspace, можно указывать оригинальное имя.

*Выходным* сигналом является скаляр — результат вычисления выражения. Блок имеет следующие параметры:

Expression (Выражение) — вычисляемое арифметическое или логическое выражение.

При записи выражения соблюдаются правила, принятые для языка С, а именно:

- символы арифметических операций: +; -; \*; /; ^;
- символы операций отношения: =; != ; <; >; <=; >= (сравните с табл. 6.10);
- символы логических операций (их всего три): «« (AND); ++ (OR); + (NOT) (сравните с табл. 6.11);
- встроенные функции MATLAB ограничены следующим списком (см. табл. 6.3– 6.4): abs, acos, asin, atan, atan2, ceil, cos, cosh, exp, fabs, floor, ln, log, log10, pow, power, rem, sgn, sin, sinh, sqrt, tan, tanh.

### □ Sample time — см. разд. 4.2.1.

На рис. 6.26 приведены примеры вычислений арифметических выражений. На рис. 6.26,  $\delta$  в поле **Expression** записано выражение  $(sin(b(1,1))+cos(b(1,2)))^2$ , и входной сигнал b поступает из Workspace:

>> b = [1 2;3 4];



Рис. 6.26. Примеры вычисления арифметических выражений с помощью блока Fcn при входном сигнале, поступающем: с выхода блока Constant (а) и из Workspace (б)

### 6.1.9.2. Блок MATLAB Fcn

Блок *MATLAB Fcn* вычисляет арифметическое выражение, встроенную или внешнюю функцию с одним выходным параметром. Блок имеет следующие параметры:

□ **MATLAB function** (Функция MATLAB) — арифметическое выражение, встроенная или внешняя функция.

Арифметические выражения записываются по правилам языка МАТLAB, и ограничения на список встроенных функций МАТLAB отсутствуют. Простые переменные записываются по тем же правилам, что и для блока Fcn (см. разд. 6.1.9.1), с использованием стандартного имени и или, для переменных из Workspace, — оригинального имени.

При обращении к *встроенной* функции достаточно указать ее имя. Если входной сигнал представлен вектором или матрицей, то вычисление функции будет *по-* элементным.

Для входного сигнала, поступающего из Workspace, следует указать его оригинальное имя — аргумент *встроенной* функции в круглых скобках.

При обращении к внешней функции с одним выходным параметром необходимо указать ее имя и список входных параметров в круглых скобках с использованием стандартного имени и или, для переменных из Workspace, — оригинального имени.

Если внешняя функция хранится в собственной папке внутри папки work, то перед запуском модели следует сохранить путь к собственной папке по команде Add Path (Добавить путь) контекстного меню;

- □ **Output dimensions** (Размер выходного массива) размер выходного массива. Указывается явно или как –1 при совпадении с размером входного массива;
- Output signal type (Тип сигнала на выходе) тип данных на выходе: real (вещественный); complex (комплексный); auto (совпадает с типом данных на входе);
- □ Collapse 2-D results to 1-D (Преобразование размерности 2-D в 1-D) флаг, при установке которого матрица на входе интерпретируется как вектор-столбец, сформированный из ее столбцов;
- □ Sample time *см. разд. 4.2.1*.

На рис. 6.27 приведен пример вычисления внешней функции F3:

function z = F3(x,y) % Вычисление суммы кубов z

с указанием в поле MATLAB function: F3(u(1),u(2)).



Рис. 6.27. Пример вычисления внешней функции F3 с помощью блока MATLAB Fcn при сбросе флага Collapse 2-D results to 1-D

# 6.1.9.3. Блок Embedded MATLAB Function

Блок *Embedded MATLAB Function* вычисляет внешнюю функцию создаваемую пользователем. Блок не имеет входных параметров.

Двойной щелчок левой кнопки мыши на пиктограмме блока открывает окно **Embedded MATLAB Editor** (Встроенный редактор MATLAB), подобное окну **Editor** (Редактор) при создании внешней функции в MATLAB. Количество входных и выходных параметров внешней функции автоматически задает количество входов и выходов блока.

Ha puc. 6.28 приведено окно Embedded MATLAB Editor с созданной внешней функцией F1 и пример использования блока Embedded MATLAB при вычислении функции F1, входные параметры которой — векторы x, y — хранятся в Workspace:

>> x = [1 3]; >> y = [2 4];



Рис. 6.28. Окно Embedded MATLAB Editor с внешней функцией F1 и пример ее вычисления с помощью блока Embedded MATLAB

Помимо рассмотренных, в группу блоков User-defined Functions включены блоки, предназначенные для создания блоков пользователя — моделей расширения, дополняющих библиотеку Simulink. Необходимость разработки новых блоков возникает в тех случаях, когда блок, реализующий требуемую операцию, отсутствует, а S-модель, создаваемая из комбинации блоков, представляется излишне сложной. В подобных ситуациях удобно создать блоки пользователя, применяя технологию S-функций на основе блока S-function в группе User-defined Functions, который будет выполнять программу пользователя, составленную на языке MATLAB или другом языке высокого уровня. В рамках данной книги технология создания S-функций не рассматривается. С ней можно познакомиться в справочной системе MATLAB по Simulink в формате HTML.

## 6.1.10. Основные характеристики матрицы

К основным характеристикам матрицы A относятся (в скобках дана функция MATLAB) [22]:

□ определитель (детерминант) (det (A));

- **D** pahr (rank (A));
- HOPMa (погт (А, р));
- □ ортонормированный базис (orth (A));
- □ нуль-пространство (null (A));
- □ собственные значения v и собственные векторы D ([v, D]=eig (A));
- Cлед (trace (A));
- □ сингулярные числа (svd (A));
- □ числа обусловленности (cond (A, p)).

Основные характеристики матрицы в Simulink можно вычислить с помощью блока MATLAB Fcn (см. разд. 6.1.9.2), указывая имя функции в поле MATLAB function.

# 6.2. Средства библиотеки блоков Signal Processing Blockset

В этом разделе мы познакомимся со средствами моделирования математических преобразований из библиотеки блоков Signal Processing Blockset.

# 6.2.1. Специальные математические операции с матрицами

S-модели специальных математических операций с матрицами представлены в группе блоков Math Functions → Math Operations (Математические функции → Математические операции) (табл. 6.14).

N⁰	Блок	Операция	Операция в MATLAB
1.	Complex Exponen- tial (mask)	Разложение по Эйлеру комплексной экспоненты: <i>e<sup>ix</sup></i> = cos( <i>x</i> ) + <i>j</i> sin( <i>x</i> )	exp(j*x)
2.	Cumulative Product	Произведение с накоплением <sup>1</sup> : элемент	cumprod(u)
		предыдущих элементов столбца (строки)	cumprod(u,2)
3.	Cumulative Sum	Сумма с накоплением <sup>2</sup> : элемент столбца	cumsum(u)
		(строки) равен сумме предыдущих элементов столбца (строки)	cumsum(u,2)
4.	dB Conversion	Преобразование в децибелы	10.*log10(abs(u))
	(mask)		10.*log10(abs(u).^2)
5.	Difference	Матрица разностей: элемент столбца	diff(u)
		(строки) равен разности данного и предыдущего элементов столбца (строки)	diff(u,[],2)
6.	Normalization	Вычисление норм на основе нормы $\left\ x\right\ _2$	norm(u,2)

Таблица 6.14. Блоки специальных математических операций с матрицами

### 6.2.1.1. Блок Complex Exponential

Блок *Complex Exponential* вычисляет разложение по Эйлеру комплексной экспоненты:  $e^{jx} = \cos(x) + j\sin(x)$ . Этот блок не содержит входных параметров. Пример использования блока приведен на рис. 6.29.



Рис. 6.29. Пример разложения по Эйлеру с помощью блока Complex Exponential

### 6.2.1.2. Блок Cumulative Product

Блок *Cumulative Product* вычисляет произведение с накоплением — элемент столбца (строки) равен произведению предыдущих элементов столбца (строки). Параметры блока задаются на двух вкладках — Main и Data type attributes (Атрибуты типа данных).

На вкладке Main определяются параметры:

□ Multiply input along (Умножить входной сигнал по) — вариант вычисления:

• Channels (running product) — по каналам<sup>3</sup>;

<sup>&</sup>lt;sup>1</sup> Кумулятивное произведение.

<sup>&</sup>lt;sup>2</sup> Кумулятивная сумма.

<sup>&</sup>lt;sup>3</sup> О многоканальном представлении сигналов речь пойдет далее в гл. 11.

- Columns по столбцам;
- Rows по строкам;
- □ Reset port (Сбрасываемый порт) активизируется при выборе значения Multiply input along: Channels (running product).

На вкладке **Data type attributes** для данных с ФТ типа fixdt (см. табл. 4.5) задаются следующие параметры:

□ Rounding mode (Режим округления) — см. параметр Integer rounding mode блока Add (см. разд. 6.1.1.1);

### □ Overflow mode (Режим переполнения) — режимы переполнения для данных с ФТ<sup>1</sup>:

 Wrap — модульная арифметика: при переполнении результат автоматически заменяется значением по модулю 2 (отрицательное число, меньшее либо равное –1, суммируется с ближайшим по модулю 2<sup>N</sup>, а из положительного, большего 1, вычитается ближайшее 2<sup>N</sup>).

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 слово.

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];

- в группе Data Type типы данных для промежуточных произведений (Intermediate product), результирующего произведения (Product output), суммирования произведений (Accumulator) и на выходе (Output) выбираются в соответствии с табл. 4.5;
- □ Lock scaling against changes by the autoscaling tool (Блокирование масштабирования) флаг, при установке которого блокируется автоматическое масштабирования) флаг, типа fixdt (см. табл. 4.5).

На рис. 6.30 приведен пример вычисления произведения с накоплением по столбцам (**Multiply input along**: Columns) для данных типа double.



Рис. 6.30. Пример вычисления произведения с накоплением по столбцам с помощью блока Cumulative Product

<sup>&</sup>lt;sup>1</sup> В МАТLАВ данные с ФТ представляются в диапазоне [-1; 1]. Подробнее обработка данных с ФТ рассматривается в *гл. 14*.

### 6.2.1.3. Блок Cumulative Sum

Блок *Cumulative Sum* вычисляет сумму с накоплением — элемент столбца (строки) равен сумме предыдущих элементов столбца (строки). Параметры блока подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2).

Пример использования блока для вычисления суммы с накоплением по строкам (параметр Sum input along: Rows) для данных типа double приведен на рис. 6.31.



Рис. 6.31. Пример вычисления суммы с накоплением по строкам с помощью блока Cumulative Sum

### 6.2.1.4. Блок dB Conversion

Блок *dB* Conversion выполняет преобразование в децибелы. Блок имеет следующие параметры (рис. 6.32):

□ Convert to (Преобразовать к) — вариант преобразования в децибелы:

• dB — преобразование в децибелы по одной из формул:

$$Y = 10 \lg |X| \ \mathsf{д}\mathsf{E}; \tag{6.1}$$

$$Y = 10 \lg |X|^2 = 20 \lg |X| \ \text{дБ}$$
(6.2)

в зависимости от выбранного значения параметра Input signal;

• dBm — преобразование в децибелы по одной из формул:

$$Y = 10 \lg |X| + 30 ; (6.3)$$

$$Y = 10 \lg |X|^2 + 30 = 20 \lg |X| + 30$$
(6.4)

в зависимости от выбранного значения параметра Input signal;

□ Input signal (Входной сигнал) — преобразуемые в децибелы значения:

- Amplitude (Амплитуда) амплитуда и преобразования (6.2), (6.4);
- Power (Мощность) мощность и преобразования (6.1), (6.3);
- □ Load resistance (Резисторная нагрузка) коэффициент масштабирования *R*, активизируемый при значении параметра Input signal: Amplitude; учитывается в формулах (6.2), (6.4) следующим образом:

$$Y = 10 \lg(|X|^2 / R);$$
  
$$Y = 10 \lg(|X|^2 / R) + 30;$$

□ Add eps to input to protect against "log(0) = -inf" (Добавить погрешность ерз к входному сигналу для присваивания log(0) = -inf) — флаг, при сбросе которого (по умолчанию) логарифму нуля присваивается значение машинной бесконечности со знаком минус — -inf.

	n (mask) (link)	
Convert input power relative converting to reference pow	of watts or volts to decibels. Voltage inputs are first converted to to the specified load resistance, where $P = (V^2)/R$ . When dB, the reference power is 1 W. When converting to dBm, the ver is 1 mW.	
Parameters		
Convert to:	IB 💌	
Input signal:	Amplitude	
Load resistan	ce (ohms):	
1		
Add eps t	o input to protect against "log(0) = -inf"	
	QK Cancel Help Apply	
0	QK         Cancel         Help         Apply           dB         0	

Рис. 6.32. Параметры блока dB Conversion и пример преобразования в децибелы

### 6.2.1.5. Блок Difference

Блок *Difference* вычисляет матрицу разностей — элемент столбца (строки) равен разности данного и предыдущего элементов столбца (строки). Параметры блока подобны параметрам блока Cumulative Product *(см. разд. 6.2.1.2)*. Пример использования блока для вычисления матрицы разностей по строкам (параметр **Difference along**: Rows) для данных типа double приведен на рис. 6.33.



Рис. 6.33. Пример вычисления матрицы разностей по строкам с помощью блока Difference

### 6.2.1.6. Блок Normalization

Блок *Normalization* вычисляет нормы на основе нормы  $||x||_2$ . Параметры блока задаются на двух вкладках — Main и Data type attributes.

На вкладке Main определяются параметры (рис. 6.34):

- □ **Norm** (Норма) нормы:
  - 2-norm норма на основе  $||x||_2$ , определяемая для *столбцов* матрицы A по формуле:

$$\frac{a_{ij}}{\sqrt{\sum_{i=1}^{m} |a_{ij}|^2} + bias}, \quad i = 1, 2, ..., m; \quad j = 1, 2, ..., n,$$
(6.5)

или для строк:

$$\frac{a_{ij}}{\sqrt{\sum_{j=1}^{n} \left|a_{ij}\right|^{2} + bias}}, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n,$$
(6.6)

где:

A — матрица размером  $m \times n$ ;

*a<sub>ii</sub>* — элемент матрицы *A*, *i*, *j* — номера строки и столбца соответственно;

$$\sqrt{\sum_{i=1}^{m} |a_{ij}|^2}$$
 и  $\sqrt{\sum_{j=1}^{n} |a_{ij}|^2}$  — нормы  $||x||_2$  для столбцов и строк матрицы  $A$ ;

*bias* — положительная константа очень малой величины, не вносящая изменений в вычисления, но предотвращающая возможное деление на ноль (см. далее параметр **Normalization bias**);

Squared 2-norm — норма на основе квадрата ||x||<sub>2</sub>, определяемая для *столбцов* матрицы *A* по формуле:

$$\frac{a_{ij}}{\sum_{i=1}^{m} |a_{ij}|^2 + bias}, \quad i = 1, 2, ..., m; \quad j = 1, 2, ..., n,$$
(6.7)

или для строк:

$$\frac{a_{ij}}{\sum_{j=1}^{n} |a_{ij}|^{2} + bias}, \quad i = 1, 2, ..., m; \quad j = 1, 2, ..., n;$$
(6.8)

□ Normalization bias (Смещение нормализации) — константа *bias* в формулах (6.5)—(6.8);

- □ Normalize over (Нормализовать по) вариант нормализации:
  - Each columns (Каждый столбец) по столбцам;
  - Each rows (Каждая строка) по строкам;
  - Specified dimension (Определенного размера) для массива произвольного размера;
- Treat sample-based row input as a column (Трактовать строку дискретного сигнала как столбец) флаг, при установке которого дискретный сигнал рассматривается как столбец; устанавливается для матриц при выборе Normalize over: Each columns.

На вкладке Data type attributes параметры подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2).

Проверим в MATLAB результат вычислений на рис. 6.34, например, для первого элемента полученной матрицы. Вычислим норму  $||x||_2$  для первого столбца исходной матрицы и разделим его первый элемент на данную норму:

	Normalize the input over the or by the squared 2-norm, u	e specified dimension by the vector 2-norm, sqrt(u'u u'u+b, where b is a bias used to protect against divi	u)+b, ide-
	by-zero. Normalization is pe input signal specified in the	erformed over rows, columns, or the dimension of t 'Dimension' parameter.	he
	Main Fixed-point		
	Parameters		
	Norm: 2-norm		×
	Normalization bias:	1e-10	
	Normalize over:	Each column	~
	Treat cample based ro	uu innut na n-anhuma	
		w input as a column	
		<u>OK</u> <u>Cancel</u> <u>H</u> elp <u>A</u> p	ply
1 2 3 ]		OK         Cancel         Help         Ap           0.1231         0.2074	ply 0.2673
1 2 3 4 5 6 7 0 0		OK         Cancel         Help         Ap           0.1231         0.2074           0.4924         0.5183	ply 4 0.2673 5 0.5345

Рис. 6.34. Параметры блока Normalization и пример вычисления нормы 2-norm для столбцов с элементами типа double

# 6.2.2. Типовые матрицы

S-модели генерирования *типовых матриц* представлены в группе блоков Math Functions  $\rightarrow$  Matrices and Linear Algebra  $\rightarrow$  Matrix Operations (Математические функции  $\rightarrow$  Матрицы и линейная алгебра  $\rightarrow$  Операции с матрицами) (табл. 6.15).

N⁰	Блок	Операция	Функция MATLAB
1.	Create Diagon- al Matrix (mask)	Генерирует диагональную матрицу: все элементы, кроме расположенных на главной диагонали, равны нулю	diag (V) V — вектор диагональных элементов
2.	Extract Di- agonal (mask)	Формирует вектор из элементов главной диагонали квадратной матрицы	diag (V) V — квадратная матрица
3.	Extract Trian- gular Matrix (mask)	Выделяет нижнюю (верхнюю) треугольную матрицу: все элементы, расположенные ниже (выше) главной диагонали, равны нулю	tril(A) triu(A) А — исходная матрица
4.	Identity Matrix (mask)	Генерирует единичную матрицу	еуе (N) N — порядок матрицы
5.	Toeplitz (mask)	Генерирует матрицу Теплица — квадратную с одинаковыми элементами на любой диагонали	toeplitz(r) r — первый столбец (строка) симметричной матрицы toeplitz(c,r) c,r — первые столбец и строка несимметричной матрицы

Таблица 6.15. Блоки генерирования типовых матриц

# 6.2.2.1. Блок Create Diagonal Matrix

Блок *Create Diagonal Matrix* генерирует диагональную матрицу: все элементы, кроме расположенных на главной диагонали, равны нулю. Блок не содержит входных параметров.

Пример использования блока приведен на рис. 6.35.



Рис. 6.35. Пример генерации диагональной матрицы с помощью блока Create Diagonal Matrix

### 6.2.2.2. Блок Extract Diagonal

Блок *Extract Diagonal* формирует вектор из элементов главной диагонали квадратной матрицы. Блок не содержит входных параметров.

Пример использования блока приведен на рис. 6.36.



Рис. 6.36. Пример формирования вектора из элементов главной диагонали с помощью блока Extract Diagonal

### 6.2.2.3. Блок Extract Triangular Matrix

Блок *Extract Triangular Matrix* выделяет нижнюю (верхнюю) треугольную матрицу: все элементы, расположенные ниже (выше) главной диагонали, равны нулю. Блок содержит один входной параметр **Extract** (Выделение), принимающий значения: Upper — верхняя или Lower — нижняя треугольная матрица.

Пример использования блока приведен на рис. 6.37.



Рис. 6.37. Пример выделения верхней треугольной матрицы (Upper) с помощью блока Extract Triangular Matrix

### 6.2.2.4. Блок Identity Matrix

Блок *Identity Matrix* генерирует единичную матрицу. Параметры блока задаются на двух вкладках — **Main** и **Data Types**.

На вкладке Main определяются параметры:

- □ Inherit output attributes from input port (Наследование выходных атрибутов от входного порта) флаг, при установке которого у блока появляется вход, и все атрибуты наследуются от сигнала на данном входе;
- □ Matrix size размер единичной матрицы;
- □ Sample time см. разд. 4.2.1.

При открытой вкладке **Data Types** задается одни параметр **Output data type** (Тип данных на выходе) (см. табл. 4.5). Пример использования блока приведен на рис. 6.38.



Рис. 6.38. Пример генерации единичной матрицы для данных типа double с помощью блока Identity Matrix

### 6.2.2.5. Блок Toeplitz

Блок *Toeplitz* генерирует матрицу Теплица — квадратную с одинаковыми элементами на любой диагонали. Блок имеет следующие параметры:

□ Symmetric (Симметричная) — флаг, при установке которого генерируется симметричная, а при сбросе — несимметричная матрица Теплица.

Для симметричной матрицы блок Toeplitz имеет один вход, и матрица Теплица генерируется по вектору на этом входе.

Для несимметричной матрицы блок Toeplitz имеет два входа: Toep col и Toep row. Первый столбец матрицы Теплица генерируется по вектору на входе Toep col, а первая строка — по вектору на входе Toep row. Векторы должны иметь одинаковую длину и одинаковые первые элементы. Если эти элементы векторов будут различными, то автоматически первому элементу первой строки будет присвоено значение первого элемента на входе Toep col;

### □ Saturate on integer overflow — см. разд. 6.1.1.1.

На рис. 6.39 приведены примеры генерации матрицы Теплица.



Рис. 6.39. Примеры генерации матрицы Теплица с помощью блока Toeplitz: симметричной (*a*) и несимметричной (*б*)

# 6.2.3. Преобразование матриц

S-модели преобразования векторов и матриц представлены в группе блоков Math Functions  $\rightarrow$  Matrices and Linear Algebra  $\rightarrow$  Matrix Operations (табл. 6.16).

Таблица 6.16. Блоки формирования матриц

N⁰	Блок	Операция
1.	Matrix Concatenate	Конкатенация матриц и векторов
2.	Overwrite Values (mask)	Замена подматрицы или главной диагонали
3.	Permute Matrix (mask)	Формирование новой матрицы из столбцов (строк) исходной матрицы путем их перестановки и/или копирования
4.	Submatrix (mask)	Выделение подматрицы из матрицы

### 6.2.3.1. Блок Matrix Concatenate

Блок *Matrix Concatenate* выполняет конкатенацию матриц и векторов и дублирует одноименный блок из группы блоков **Math Operations** (см. разд. 6.1.5.1).

### 6.2.3.2. Блок Overwrite Values

Блок *Overwrite Values* выполняет замену подматрицы или главной диагонали. Блок имеет следующие параметры (рис. 6.40):

□ Overwrite (Замена) — вариант замены в матрице:

- Submatrix (Подматрица) подматрица;
- Diagonal (Диагональ) главная диагональ;
- □ Source of overwriting value(s) (Источник перезаписываемых данных) способ замены подматрицы:
  - Specify via dialog (Определить через диалог) посредством ввода в поле **Overwrite with**;
  - Second input port (Второй входной порт) посредством ввода с автоматически создаваемого второго входа;
- Overwrite with (Заменить этим) поле для ввода подматрицы, предназначенной для замены;
- □ Row span (Границы по строкам) варианты указания границ подматрицы по строкам:
  - Range of row (границы строки) замена с произвольными границами по строкам;
  - All rows (Все строки) замена во всех строках;
  - Опе row (Одна строка) замена в одной строке;
- □ Starting row (Начальная строка) порядок отсчета начальной строки:
  - First (Первый) равен единице (*i* = 1);

- Index (Индекс) имеет произвольное значение из диапазона  $1 \le i \le N$ , где N номер последней строки;
- Offset from last (Смещение относительно последней строки) смещен относительно последнего номера строки (*i* = *N* - *l*); задается смещение *l* (неотрицательное число);
- Last (Последний) равен последнему номеру строки (i = N);
- Middle (Средний) равен срединному номеру *L* = int(*N*/2+1), где функция int означает округление до ближайшего целого в сторону увеличения;
- Offset from middle (Смещение относительно среднего) смещен относительно срединного номера строки L (i = L + l); задается смещение l (целое положительное или отрицательное число);
- Ending row (Конечная строка) порядок отсчета номера конечной строки (подобен параметру Starting row).

Остальные параметры, начиная с Column span (Границы по столбцам), связаны с границами подматрицы по столбцам, которые указываются аналогично границам по строкам.



Рис. 6.40. Пример задания параметров блока Overwrite Values и результат замены компактной совокупности элементов матрицы

### 6.2.3.3. Блок Permute Matrix

Блок *Permute Matrix* формирует новую матрицу из столбцов (строк) исходной матрицы путем их перестановки и/или копирования. Блок имеет два входа: верхний A — для ввода исходной матрицы и нижний P — для ввода вектора с номерами столбцов (строк) матрицы A, из которых будет формироваться новая матрица, и содержит параметры (рис. 6.41):

□ **Permute** (Перемещение) — способ формирования новой матрицы:

- Columns (Столбцы) по столбцам;
- Rows (Строки) по строкам;
- □ Index mode (Режим для индекса) порядок отсчета номеров столбцов (строк) исходной матрицы *A*:
  - One-based (На основе единицы) столбцы (строки) отсчитываются, начиная от единицы;
  - Zero-based (На основе нуля) столбцы (строки) отсчитываются, начиная от нуля;

a vector P o hen you selec number of in	ector P of you select mber of inp	integer- the out rows		
		~		
		~		
		~		
e				
		Apply		
3	3		3	2
6	6	6	5	5
9	9	(	<u>ا</u>	8
Display	isplay			
9 Display	9		ç	9

Рис. 6.41. Пример задания параметров блока Permute Matrix для формирования новой матрицы из столбцов исходной матрицы

- □ Invalid permutation index (Недопустимый индекс при перемещении) варианты реакции на наличие недопустимого номера в векторе *P*, большего или меньшего количества столбцов (строк) исходной матрицы *A*:
  - Clip index (Усечение индекса) недопустимому номеру присваивается номер последнего столбца;
  - Clip and warn (Усечение и предупреждение) аналогично значению Clip index, но с выводом предупреждения в автоматически открываемом окне;
  - Generate error (Генерация ошибки) при наличии недопустимого номера выдается сообщение об ошибке в автоматически открываемом окне Simulation Diagnostics (см. разд. 3.3);
- □ Error when length of P is not equal to Permute dimension size (Ошибка при длине вектора P, не соответствующей размеру исходного массива) — флаг, при установке которого выдается сообщение об ошибке в случае, если длина вектора P оказывается большей или меньшей количества столбцов (строк) матрицы A.

# 6.2.3.4. Блок Submatrix

Блок *Submatrix* выделяет подматрицу из матрицы. Параметры блока задаются подобно параметрам блока Overwrite Values (см. разд. 6.2.3.2).

Пример использования блока приведен на рис. 6.42.



Рис. 6.42. Пример задания параметров блока Submatrix для выделения подматрицы из матрицы

# 6.2.4. Операции с матрицами в задачах линейной алгебры

S-модели операций с матрицами в задачах линейной алгебры представлены в группе блоков Math Functions  $\rightarrow$  Matrices and Linear Algebra  $\rightarrow$  Matrix Operations (табл. 6.17).

Nº	Блок	Операция	Символ или операция в MATLAB
1.	Matrix Multiply	Умножение матриц	*
2.	2. Matrix Product Вычисление произведения элементов столбца, строки или всей матрицы	prod(A)	
		prod(A')	
			prod(A(:))

Таблица 6.17. Блоки операций с матрицами в задачах линейной алгебры

Окончание табл. 6.17

Nº	Блок	Операция	Символ или операция в MATLAB
3.	Matrix Square (mask)	Вычисление произведения транспонированной и вещественной матриц или эрмитово сопряженной и комплексной матриц	X,*X
4.	Matrix Sum	Вычисление суммы элементов столбца,	sum(A)
		строки или всей матрицы	sum(A')
			sum(A(:))
5.	Transpose	Транспонирование вещественной и	Χ'
		комплексной матриц или эрмитово сопряжение комплексной матрицы	X.'

### 6.2.4.1. Блок Matrix Multiply

Блок Matrix Multiply выполняет умножение матриц. Это реализация блока Product (см. разд. 6.1.1.6).

### 6.2.4.2. Блок Matrix Product

Блок *Matrix Product* вычисляет произведение элементов столбца, строки или всей матрицы. Параметры блока задаются на двух вкладках — Main и Data type attributes.

На вкладке **Main** задается один параметр **Multiply over** (Умножение по) — вариант вычисляемого произведения матрицы: Columns — по столбцам, Rows — по строкам и Entire input (Полный вход) — всех элементов.

На вкладке **Data type attributes** параметры подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2). Пример использования блока приведен на рис. 6.43.





### 6.2.4.3. Блок Matrix Square

Блок *Matrix Square* имеет один вход и вычисляет произведение транспонированной и вещественной матриц или эрмитово сопряженной и комплексной матриц. Блок не содержит входных параметров.

Пример использования блока приведен на рис. 6.44.



Рис. 6.44. Пример вычисления произведения транспонированной и вещественной матриц с помощью блока Matrix Square

### 6.2.4.4. Блок Matrix Sum

Блок *Matrix Sum* вычисляет сумму элементов столбца, строки или всей матрицы. Параметры блока подобны параметрам блока Add (*см. разд. 6.1.1.1*).

Пример использования блока приведен на рис. 6.45.



Рис. 6.45. Пример вычисления суммы матрицы по столбцам с помощью блока Matrix Sum для данных типа double

### 6.2.4.5. Транспонирование и эрмитово сопряжение матрицы

Блок *Transpose* выполняет транспонирование вещественной и комплексной матриц или эрмитово сопряжение комплексной матрицы. Блок содержит один параметр — **Hermitian** — флаг, при установке которого выполняется операция эрмитова сопряжения, а при сбросе — операция транспонирования.

Примеры использования блока приведены на рис. 6.46.



Рис. 6.46. Примеры использования блока Transpose: транспонирование вещественной (а), комплексной (б) матриц и эрмитово сопряжение комплексной матрицы (в)

# 6.2.5. Обращение матрицы

S-модели *обращения* матриц<sup>1</sup> представлены в группе блоков Math Functions  $\rightarrow$  Matrices and Linear Algebra  $\rightarrow$  Matrix Inverses, среди которых выделим три (табл. 6.18).

Таблица 6.18. Блоки инверсии матриц

N⁰	Блок	Операция	Функция MATLAB
1.	LU Inverse (mask)	Обращение вещественной или комплексной матрицы на основе LU-разложения <sup>2</sup>	inv(A)
2.	LDL Inverse (mask)	Обращение положительно определенной <sup>3</sup> эрмитовой или симметричной матрицы <sup>4</sup> на основе LU- разложения	inv(A)
3.	Cholesky Inverse (mask)	Обращение положительно определенной эрмитовой или симметричной матрицы на основе разложения Холецкого	inv(A)

### 6.2.5.1. Блок LU Inverse

Блок *LU Inverse* выполняет обращение вещественной или комплексной матрицы на основе LU-разложения. Блок не содержит параметров.

Примеры использования блока приведены на рис. 6.47.



Рис. 6.47. Примеры обращения матриц с помощью блока LU Inverse: вещественной (*a*) и комплексной (*б*)

<sup>&</sup>lt;sup>1</sup> Матрица  $A^{-1}$  называется обратной к матрице A, если произведение этих матриц дает единичную матрицу.

<sup>&</sup>lt;sup>2</sup> Разложение матриц рассматривается в *разд. 6.2.6*.

<sup>&</sup>lt;sup>3</sup> Согласно критерию Сильвестра матрица является *положительно определенной* тогда и только тогда, когда положительны все ее базисные миноры. Признаками положительно определенной эрмитовой матрицы являются положительные диагональные элементы и положительный определитель [5, 22].

<sup>&</sup>lt;sup>4</sup> С представлением в MATLAB эрмитовых и симметричных матриц, а также с операциями с ними можно познакомиться в [22].
Проверим равенство (6.9) для вещественной матрицы (рис. 6.47, а):

```
>> A = [1 2:5 8];
>> B = inv(A)
B =
   -4.0000
            1,0000
    2.5000
            -0.5000
>> A*B
ans =
     1
           0
     0
           1
и комплексной матрицы (рис. 6.47, \delta):
>> C = [1+i 2-3i;5-i 8+2i]
C =
   1.0000 + 1.0000i
                     2.0000 - 3.0000i
   5.0000 - 1.0000i 8.0000 + 2.0000i
>> D = inv(C)
D =
   0.0630 - 0.2986i
                      0.1137 + 0.0699i
                      0.0356 - 0.0384i
   0.0438 + 0.1836i
>> C*D
ans =
   1.0000 - 0.0000i
                      0.0000 + 0.0000i
  -0.0000
                       1.0000 + 0.0000i
```

#### 6.2.5.2. Блок LDL Inverse

Блок *LDL Inverse* выполняет обращение положительно определенной эрмитовой или симметричной матрицы на основе LU-разложения. Блок содержит один параметр **Non-positive definite input** (Не положительно определенный вход) — варианты реакции на *не* положительно определенную матрицу на входе:

- □ Warning выдать предупреждение;
- **Ignore** игнорировать;

□ Error — выдать сообщение об ошибке.

Пример использования блока приведен на рис. 6.48.



Рис. 6.48. Пример обращения положительно определенной эрмитовой матрицы с помощью блока LDL Inverse

```
Проверим равенство (6.9) в MATLAB (см. рис. 6.48):
>> A=[2 0.5-0.5i;0.5+0.5i 1]
A =
```

```
0.5000 - 0.5000i
   2.0000
   0.5000 + 0.5000i
                       1.0000
>> B=inv(A)
B =
   0.6667
                      -0.3333 + 0.3333i
  -0.3333 - 0.3333i
                      1.3333
>> A*B
ans =
    1.0000
                      Ο
                1.0000
          0
```

#### 6.2.5.3. Блок Cholesky Inverse

Блок *Cholesky Inverse* выполняет обращение положительно определенной эрмитовой или симметричной матрицы на основе разложения Холецкого. Блок имеет тот же параметр, что и блок LDL Inverse (см. разд. 6.2.5.2).

Пример использования блока приведен на рис. 6.49 (сравните с рис. 6.48).



Рис. 6.49. Пример обращения положительно определенной эрмитовой матрицы с помощью блока Cholesky Inverse

### 6.2.6. Разложение матрицы

S-модели *разложения* (факторизации) матриц представлены в группе блоков **Math Functions**  $\rightarrow$  **Matrices and Linear Algebra**  $\rightarrow$  **Matrix Factorizations** (Математические функции  $\rightarrow$  Матрицы и линейная алгебра  $\rightarrow$  Факторизация матриц), среди которых выделим три (табл. 6.19).

Таблица 6.19. Блоки разложен	ния матриц
------------------------------	------------

N⁰	Блок	Операция	Функция MATLAB
1.	LU Factorization	LU-разложение	[L,U,P] = lu(A)
2.	QR Factorization (mask)	QR-разложение	[Q, R, P] = qr(A)
3.	Cholesky Factorization (mask)	Разложение Холецкого	R = chol(A)

#### 6.2.6.1. Блок LU Factorization

Блок LU Factorization выполняет LU-разложение матрицы.

LU-разложение может быть использовано для вещественных и комплексных квадратных матриц и дает представление матрицы в виде произведения нижней

треугольной матрицы с единичными диагональными элементами и верхней треугольной матрицы [22]:

$$PA = LU, \tag{6.9}$$

где:

А — исходная матрица;

L — нижняя треугольная матрица с единичными диагональными элементами;

U — верхняя треугольная матрица;

*Р* — матрица перестановок с элементами, равными 0 или 1, формирующая столбцы матрицы-произведения *PA* так, чтобы первым элементом каждого столбца был *главный* элемент (наибольший по модулю).

Параметры блока задаются на двух вкладках — Main и Fixed-point.

На вкладке **Main** задается один параметр **Show singularity status** (Показать сингулярность) — флаг, при установке которого автоматически создается *третий* выход, позволяющий контролировать вырожденность (особенность, сингулярность) исходной матрицы: число 0 будет признаком невырожденной, а 1 — вырожденной матрицы.

На вкладке Fixed-point параметры подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2) на вкладке Data type attributes.

Пример использования блока приведен на рис. 6.50. На верхнем выходе выводится объединение матриц L и U, а на нижнем — вектор-столбец, элементы которого равны *номеру* единичного элемента в *строке* матрицы P.

Вывести объединение матриц L и U в MATLAB можно с помощью функции: >>  ${\tt lu}\,({\tt A})$ 



Рис. 6.50. Пример LU-разложения матрицы с помощью блока LU Factorization

#### 6.2.6.2. Блок QR Factorization

Блок *QR Factorization* выполняет QR-разложение матрицы.

R-разложение может быть использовано для вещественных и комплексных квадратных матриц. В первом случае будем иметь разложение в виде произведения ортогональной и верхней треугольной матриц, во втором — в виде произведения унитарной и верхней треугольной матриц [22]:

$$AP = QR , \qquad (6.10)$$

где:

*А* — исходная матрица;

*Q* — ортогональная матрица, если *A* — вещественная, и унитарная, если *A* — комплексная матрица;

*R* — верхняя треугольная матрица;

*P* — матрица перестановок с элементами, равными 0 или 1, выстраивающая диагональные элементы матрицы *R* в порядке их уменьшения по модулю.

Блок содержит один параметр **Output size** (Размер на выходе) со значениями Economy и Full, выбор которых для *квадратных* матриц не влияет на результат.

Пример использования блока приведен на рис. 6.51. На первом выходе выводится матрица Q, на втором — матрица R, а на третьем — вектор-столбец, элементы которого равны *номеру* единичного элемента в *столбце* матрицы P.



Рис. 6.51. Пример разложения матрицы с помощью блока QR Factorization

#### 6.2.6.3. Блок Cholesky Factorization

Блок Cholesky Factorization выполняет разложение Холецкого матрицы.

Разложение Холецкого используется для положительно определенных эрмитовых и симметричных матриц [22].

Разложение Холецкого для эрмитовых матриц имеет вид:

$$A = R^* R , \qquad (6.11)$$

а для симметричных:

$$A = R'R$$

где:

А — исходная матрица;

*R* — верхняя треугольная матрица с ненулевыми вещественными диагональными элементами;

*R*<sup>\*</sup> — эрмитово сопряженная матрица;

*R'* — транспонированная матрица.

Блок содержит один параметр — Non-positive definite input, подобный одноименному параметру блока LDL Inverse (см. разд. 6.2.5.2).

Пример использования блока приведен на рис. 6.52.



Рис. 6.52. Пример разложения Холецкого положительно определенной эрмитовой матрицы с помощью блока Cholesky Factorization

### 6.2.7. Решение систем линейных алгебраических уравнений

S-модели решения систем линейных алгебраических уравнений (СЛАУ) представлены в группе блоков Math Functions  $\rightarrow$  Matrices and Linear Algebra  $\rightarrow$  Linear System Solvers (Математические функции  $\rightarrow$  Матрицы и линейная алгебра  $\rightarrow$  Решатели линейных систем), среди которых выделим три (табл. 6.20).

N⁰	Блок	Операция	Операция в MATLAB
1.	LU Solver (mask)	Решение СЛАУ на основе LU-	Y = L\(P*B');
		разложения	$X = U \setminus Y$
		в — вектор-строка	
		[L,U,P]=lu(A)	
2.	QR Solver	Решение СЛАУ на основе QR-	Y = Q'*B';
	(mask) разложения	разложения	$X = P^*(R \setminus Y)$
			[Q, R, P]=qr(A)
3.	Cholesky Solver (mask)	Решение СЛАУ на основе разложения Холецкого	X = (R'*R)\B'
			R = chol(A)

Таблица 6.20. Блоки решения СЛАУ

### 6.2.7.1. Блок LU Solver

Блок *LU Solver* представляет собой решатель СЛАУ на основе LU-разложения. Блок не содержит входных параметров и имеет два входа: верхний — для матрицы коэффициентов *A* и нижний — для вектора или матрицы<sup>1</sup> свободных членов *B*.

Пример решения СЛАУ приведен на рис. 6.53.



Рис. 6.53. Пример решения СЛАУ с помощью блока LU Solver

#### 6.2.7.2. Блок QR Solver

Блок *QR Solver* представляет собой решатель СЛАУ на основе QR-разложения. Блок не содержит входных параметров и имеет ту же структуру, что и блок LU Solver *(см. разд. 6.2.7.1)*.

Пример решения СЛАУ приведен на рис. 6.54.



Рис. 6.54. Пример решения СЛАУ с помощью блока QR Solver

### 6.2.7.3. Блок Cholesky Solver

Блок *Cholesky Solver* представляет собой решатель СЛАУ на основе разложения Холецкого. Блок содержит один параметр — Non-positive definite input, подобно

<sup>&</sup>lt;sup>1</sup> Свободные члены задаются матрицей при одновременном решении нескольких СЛАУ с одинаковой матрицей коэффициентов *A*, но разными векторами свободных членов — столбцов матрицы *B* [22].

блоку LDL Inverse (см. разд. 6.2.5.2) и имеет два входа: верхний — для коэффициентов S положительно определенной эрмитовой или симметричной матрицы и нижний — для вектора или матрицы свободных членов B.

Пример решения СЛАУ приведен на рис. 6.55.



Рис. 6.55. Пример решения СЛАУ с помощью блока Cholesky Solver

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Math Functions.

# 6.2.8. Операции с матрицами в задачах математической статистики

S-модели *операций в задачах математической статистики* представлены в группе блоков Signal Management → Statistics (Управление сигналом → Статистика) (табл. 6.21).

N⁰	Блок	Операция	Функция MATLAB
1.	Autocorrelation	Автокорреляционная функция (АКФ) последовательности	xcorr
2.	Correlation	Взаимная корреляционная функция (ВКФ) последовательностей	xcorr
3.	Maximum	Максимальный элемент по столбцам, строкам или всей матрицы	max(A)
4.	Mean	Математическое ожидание (среднее значение) по столбцам, строкам или всей матрицы	mean(A)
5.	Minimum	Минимальный элемент по столбцам, строкам или всей матрицы	min(A)
6.	RMS	Среднеквадратичное значение (RMS <sup>1</sup> ) по столбцам, строкам или всей матрицы	sqrt(sum(A.^2)/m) m — длина столбца

Таблица 6.21. Блоки операций в задачах математической статистики

<sup>&</sup>lt;sup>1</sup> Root-Mean-Square (RMS).

Окончание	табл.	6.21
-----------	-------	------

N⁰	Блок	Операция	Функция MATLAB
7.	Sort	Сортировка элементов столбцов по возрастанию (убыванию)	<b>По возрастанию</b> sort(A)
8.	Standard Devi- ation	Среднеквадратическое (стандартное) отклонение (СКО) по столбцам, строкам или всей матрицы	std(A)
9.	Variance	Дисперсия по столбцам, строкам или всей матрицы	var(A)

#### 6.2.8.1. Блок Autocorrelation

Блок *Autocorrelation* вычисляет автокорреляционную функцию (АКФ) последовательности.

АКФ  $R_x(m)$  последовательности x(n), n = 0, 1, ..., N-1, — это четная функция, центрированная относительно  $R_x(0)$ , вычисляемая по формуле:

$$R_{x}(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x(n) x(n+m), \ \left|m\right| \le (N-1).$$
(6.12)

Параметры блока задаются на двух вкладках — Main и Data type attributes. На вкладке Main определяются параметры:

- □ Compute all non-negative lags (Вычислять все неотрицательные смещения) флаг, при установке которого выводятся значения АКФ при  $m \ge 0$ , а при сбросе значения АКФ при  $m \ge 0$  с указанием количества выводимых точек в поле ввода Maximum non-negative lags (less than input length) (Максимальное количество неотрицательных смещений (меньшее, чем длина входной последовательности));
- □ Scaling (Масштабирование) варианты вычисления АКФ:
  - None (Никакой) АКФ (6.13) без множителя 1/N;
  - Based (Базовый) АКФ (6.13) (для вычисления смещенной оценки АКФ);
  - Unbased (Не базовый) с нормированием к N m (для вычисления несмещенной оценки АКФ);
  - Unity at zero-leg (Единица на нулевом смещении) с нормированием к  $R_x(0)$ ;

□ Computation domain (Область вычисления) — область, в которой вычисляется АКФ:

- Тіте (Время) временная;
- Frequency (Частота) частотная.

На вкладке **Data type attributes** параметры подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2). На рис. 6.56 приведен пример вычисления АКФ (6.13) без множителя 1/N (Scaling: None) при установке флага Compute all non-negative lags, во временной области (Computation domain: Time).



Рис. 6.56. Пример вычисления АКФ с помощью блока Autocorrelation

В МАТLAB АКФ (6.12) (без множителя 1/N) вычисляется с помощью функции хсогг (х) по модифицированной формуле:

$$R_x(m) = \sum_{n=1}^{N} x(n) x(n+m-N), \quad m = N, N+1, \dots, N+(N-1)$$
(6.13)

и центрируется относительно  $R_r(N)$  (сравните с АКФ на рис. 6.56):

#### 6.2.8.2. Блок Correlation

Блок *Correlation* вычисляет взаимную корреляционную функцию (ВКФ) последовательностей.

ВКФ  $R_{xy}(m)$  последовательностей x(n) и y(n), n = 0, 1, ..., N-1, вычисляется по формуле:

$$R_{xy}(m) = \frac{1}{N} \sum_{n=0}^{N-|m|-1} x(n) y(n+m), \ \left|m\right| \le (N-1).$$
(6.14)

Параметры блока задаются на двух вкладках — Main и Data type attributes.

На вкладке **Main** задается один параметр — **Computation domain** (Область вычисления) — область вычисления ВКФ:

1. Time — временная;

2. Frequency — частотная;

3. Fastest (Самая быстрая) — та, в которой минимизируется объем вычислений.

На вкладке **Data type attributes** параметры подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2).

На рис. 6.57 приведен пример вычисления ВКФ без множителя 1/N (Scaling: None) во временной области (Computation domain: Time).



Рис. 6.57. Пример вычисления ВКФ с помощью блока Correlation

В МАТLАВ ВКФ (6.14) (без множителя 1/N) вычисляется по модифицированной формуле:

$$R_{xy}(m) = \sum_{n=1}^{N} x(n) y(n+m-N), \ m = N, \ N+1, \dots, \ N+(N-1)$$
(6.15)

с помощью функции xcorr (сравните с ВКФ на рис. 6.57):

>> x = [1 0 0 1]; >> y = [1 2 3 4]; >> R = xcorr(x,y) R = 4.0000 3.0000 2.0000 5.0000 3.0000 2.0000 1.0000

#### 6.2.8.3. Блок Maximum

Блок *Maximum* определяет максимальный элемент по столбцам, строкам или всей матрицы. Параметры блока задаются на двух вкладках — Main и Data type attributes.

На вкладке Main задаются параметры (рис. 6.58):

□ Mode (Режим) — варианты выводимого значения:

- Value and Index (Величина и индекс) максимальный элемент столбца и номер строки;
- Value (Величина) максимальный элемент столбца;
- Index (Индекс) номер строки;
- Running (Выполнение) максимальное значение в процессе моделирования на каждом шаге;

□ Index base (Базовый индекс) — начальный индекс строки и столбца:

- One единица;
- Zero ноль;

□ Find the maximum value over (Искать максимальную величину по) — вариант вычисления максимального элемента:

- Each column (Каждый столбец) по столбцам;
- Each row (Каждая строка) по строкам;
- Entire input (Полный вход) по всей матрице;
- Specified dimension (Определенного размера) в матрице произвольного размера;
- □ Treat sample-based row input as a column см. блок Normalization (см. разд. 6.2.1.6).

На вкладке Data type attributes параметры подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2).



Рис. 6.58. Параметры блока Maximum и пример вычисления максимальных элементов столбцов с выводом номеров строк

#### 6.2.8.4. Блок Mean

Блок *Mean* определяет математическое ожидание (среднее значение) по столбцам, строкам или всей матрицы. Параметры блока задаются на двух вкладках — **Main** и **Data type attributes**.

На вкладке Main задаются параметры:

- □ **Running mean** (Текущее среднее) флаг, при установке которого среднее значение определяется в процессе моделирования;
- □ Find the mean value over (Искать среднее по) вариант вычисления среднего значения с такими же значениями, как для параметра Find the maximum value over блока Maximum (см. разд. 6.2.8.3);
- □ Treat sample-based row input as a column см. блок Normalization (см. разд. 6.2.1.6).

На вкладке Data type attributes параметры подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2).

На рис. 6.59 приведен пример вычисления среднего значения по столбцам (Find the mean value over: Each column).



Рис. 6.59. Пример вычисления среднего значения по столбцам с помощью блока Mean

#### 6.2.8.5. Блок Minimum

Блок *Minimum* определяет минимальный элемент по столбцам, строкам или всей матрицы. Параметры блока подобны параметрам блока Maximum (см. разд. 6.2.8.3).

Пример вычисления минимальных элементов столбцов (Find the maximum value over: Each column) с выводом номеров строк (Mode: Value and Index), начальный номер которых равен единице (Index base: One), приведен на рис. 6.60.



Рис. 6.60. Пример вычисления минимальных элементов столбцов с выводом номеров строк с помощью блока Minimum

### 6.2.8.6. Блок RMS

Блок *RMS* вычисляет среднеквадратичное значение (RMS) по столбцам, строкам или всей матрицы.

Среднеквадратичное значение (RMS) для столбцов матрицы вычисляется по формуле:

RMS = 
$$\sqrt{\frac{a_{1j}^{2} + a_{2j}^{2} + \dots + a_{mj}^{2}}{m}}$$
, (6.16)

где:

 $a_{ii}$  — элемент матрицы A, i = 1, 2, ..., m, j = 1, 2, ..., n;

т — длина столбцов.

Параметры блока подобны параметрам блока Mean при открытой вкладке Main (см. разд. 6.2.8.4).

Пример вычисления RMS столбцов матрицы приведен на рис. 6.61. Результат совпадает с полученным в MATLAB:

>> A = [1 2 3;4 5 6;7 8 9]; >> m = 3; >> RMS = sqrt(sum(A.^2)/m) RMS = 4.6904 5.5678 6.4807



Рис. 6.61. Пример вычисления среднеквадратичного значения для столбцов матрицы с помощью блока RMS

### 6.2.8.7. Блок Sort

Блок *Sort* сортирует элементы столбцов по возрастанию (убыванию). Параметры блока задаются на двух вкладках — **Main** и **Data type attributes**.

На вкладке Main определяются параметры:

□ Mode — см. блок Maximum (см. разд. 6.2.8.3);

□ Sort order (Порядок сортировки) — порядок сортировки элементов:

- Descending (Нисходящий) по убыванию;
- Ascending (Восходящий) по возрастанию;

□ Sort algorithm (Алгоритм сортировки) — алгоритм сортировки элементов:

- Quick sort (Быстрая сортировка);
- Insertion sort (Сортировка путем вставки).

На вкладке Data type attributes параметры подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2).

На рис. 6.62 приведен пример сортировки элементов столбцов по возрастанию (Sort order: Descending) с выводом номеров строк (Mode: Value and Index), начальный номер которых по умолчанию равен единице.



Рис. 6.62. Пример сортировки элементов столбцов по возрастанию с выводом номеров строк с помощью блока Sort

#### 6.2.8.8. Блок Standard Deviation

Блок *Standard Deviation* вычисляет среднеквадратическое (стандартное) отклонение (СКО) по столбцам, строкам или всей матрицы.

Среднеквадратическое (стандартное) отклонение (СКО) для столбцов матрицы вычисляется по формуле:

$$\sigma_{\rm cr_{\it f}} = \sqrt{\frac{(a_{1j} - \overline{a}_{j})^2 + (a_{2j} - \overline{a}_{j})^2 + \dots + (a_{mj} - \overline{a}_{j})^2}{(m-1)}}, \qquad (6.17)$$

где:

 $a_{ii}$  — элемент матрицы A, i = 1, 2, ..., m, j = 1, 2, ..., n;

 $\overline{a}_i$ , j = 1, 2, ..., n — математическое ожидание (среднее) *j*-го столбца;

σ<sub>ст *i*</sub>, *j* = 1, 2, ..., *n* — СКО *j*-го столбца;

*т* — длина столбцов.

Блок содержит параметры, подобные параметрам блока Mean при открытой вкладке **Main** (см. разд. 6.2.8.4).

На рис. 6.63 приведен пример вычисления СКО столбцов матрицы; результат совпадает с полученным в MATLAB с помощью функции std:

```
>> A = [1 2 3;1 5 8;7 3 1];
>> std(A)
ans =
    3.4641 1.5275 3.6056
```



Рис. 6.63. Пример вычисления СКО столбцов матрицы с помощью блока Standard Deviation

### 6.2.8.9. Блок Variance

Блок *Variance* вычисляет дисперсию по столбцам, строкам или всей матрицы. Дисперсия для столбцов матрицы вычисляется по формуле:

$$\sigma_{\text{cr}\,j}^{2} = \frac{\left(a_{1j} - \overline{a}_{j}\right)^{2} + \left(a_{2j} - \overline{a}_{j}\right)^{2} + \dots + \left(a_{mj} - \overline{a}_{j}\right)^{2}}{(m-1)}, \quad (6.18)$$

где  $\sigma_{\text{ст}\,j}^2$  — дисперсия *j*-го столбца (квадрат СКО *j*-го столбца); *m* — длина столбцов.

Параметры блока задаются на двух вкладках — Main и Fixed-point.

На вкладке **Main** параметры подобны параметрам блока Mean *(см. разд. 6.2.8.4)*, а на вкладке **Fixed-point** — параметрам блока Cumulative Product *(см. разд. 6.2.1.2)* на вкладке **Data type attributes**.

На рис. 6.64 приведен пример вычисления дисперсии столбцов матрицы; результат совпадает с полученным в MATLAB с помощью функции var:

```
>> A = [1 2 3;1 5 8;7 3 1];
>> var(A)
ans =
    12.0000    2.3333    13.0000
```



Рис. 6.64. Пример вычисления дисперсии столбцов матрицы с помощью блока Variance

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Signal Management  $\rightarrow$  Statistics.



# Глава 7

## Средства управления сигналами

В этой главе рассматриваются основные средства управления сигналами, представленные в библиотеке блоков Simulink.

### 7.1. S-модели средств управления маршрутом сигнала

S-модели *средств управления маршрутом сигнала* представлены в группе блоков Signal Routing (Маршрутизация сигнала). Основные блоки этой группы приведены в табл. 7.1.

N⁰	Блок	Назначение	
1.	Mux	Объединяет скаляры и/или векторы одинакового численного типа в векторный сигнал	
2.	Demux	Выделяет из векторного сигнала скаляры и/или векторы одинакового численного типа или из массива ячеек его элементы	
3.	Bus Creator	Объединяет в сигнал шины сигналы разного вида и типа данных	
4.	Bus Selector	Выделяет из сигнала шины сигналы разного вида и типа данных	
5.	Goto	Беспроводной передатчик сигнала на вход блоков From	
6.	From	Беспроводной приемник сигнала с выхода блока Goto	
7.	Manual Switch	Ручной переключатель сигналов	
8.	Switch	Автоматический переключатель сигналов	
9.	Selector	Выделяет:	
		• элементы вектора и переупорядочивает их в выходном векторе;	
		<ul> <li>элементы строк и столбцов матрицы и переупорядочивает их в выходной матрице;</li> </ul>	
		<ul> <li>сигналы в сигнале (2-D) L×N и переупорядочивает их в выходном сигнале (2-D);</li> </ul>	
		<ul> <li>сигналы в сигнале (3-D) M×N×L и переупорядочивает их в выход- ном сигнале (3-D)</li> </ul>	

Окончание табл. 7.1

N⁰	Блок	Назначение	
10.	Merge	Объединяет:	
		<ul> <li>векторы одинаковых или разных длин Li в один вектор длины L = max{ Li };</li> </ul>	
		<ul> <li>N сигналов (2-D) Li×1 одинаковых или разных длин Li в один сигнал (2-D) L×N длины L = max{ Li }</li> </ul>	

Рассмотрим параметры блоков из табл. 7.1.

### 7.1.1. Блок Mus

Блок *Mus* объединяет скаляры и/или векторы одинакового численного типа в векторный сигнал. Блок имеет следующие параметры (рис. 7.1):

□ Number of inputs (Количество входов) — количество входных сигналов.

*Входные* сигналы могут быть представлены скалярами или векторами, *одновременно* вещественными или комплексными, и принадлежать к любому, но *одинаковому* численному типу, поддерживаемому в Simulink, в том числе с ФТ.

Выходной сигнал — всегда вектор — векторный сигнал (vector signal).

Количество входных сигналов указывается одним из следующих способов:

- целым числом;
- в виде вектора, количество элементов которого равно количеству входов, а их значения — длинам входных сигналов. Например, если на входе имеем три вектора с длинами 2, 5 и 2, то параметр Number of inputs будет задаваться вектором [2 5 2];
- в виде вектора, количество элементов которого равно количеству входов, а их значения (одно, несколько или все) указываются числом –1, что удобно при неопределенных или меняющихся длинах сигналов. Например, при трех входных сигналах параметр Number of inputs может задаваться вектором [–1 –1–1];
- в виде списка меток (имен) сигналов, перечисленных через запятую. Например, параметр **Number of inputs** может задаваться списком меток сигналов s1,s2,s3.

В первых трех способах по умолчанию входным сигналам присватываются метки SignalN, где N — номер входа (входного порта) сверху вниз;

□ Display option (Вариант отображения) — вид пиктограммы:

- bar вертикальная полоса с черным фоном;
- signals вертикальная полоса без фона с метками сигналов внутри. Для того чтобы увидеть метки, следует растянуть пиктограмму по горизонтали;
- none вертикальная полоса без фона с именем блока Мих внутри.

При желании скрытое по умолчанию имя блока Мих можно восстановить по команде Format | Show Name (см. табл. 2.2).



Рис. 7.1. Параметры блока Mux и пример объединения трех векторов в векторный сигнал

### 7.1.2. Блок Demux

Блок *Demux* выделяет из векторного сигнала скаляры и/или векторы одинакового численного типа или из массива ячеек его элементы. Блок имеет следующие параметры (рис. 7.2):

- □ Bus selection mode (Выбор режима разделения) флаг представления входного векторного сигнала: при сброшенном флаге как произвольного вектора или вектора на выходе блока Mux или Bus Creator (*см. разд. 7.1.3*); при установленном флаге как массива ячеек (cell array) [22] на выходе блока Mux, элементами которого являются сигналы на входах блока Mux;
- □ Number of outputs (Количество выходов) количество выходных сигналов.

*Выходные* сигналы могут быть представлены скалярами или векторами *одинакового* численного типа. Количество выходных сигналов указывается одним из следующих способов:

• целым числом *P*.

В этом случае формирование выходных сигналов зависит от состояния флага **Bus selection mode** и длины N входного вектора.

При *сброшенном* флаге **Bus selection mode** выходные сигналы формируются следующим образом. Если значение N/P равно целому числу, т. е.  $\text{mod}_{P}(N) = 0$ , то длины L всех P выходных сигналов будут одинаковыми и равными L = N/P. Например, если N = 9 и P = 3, то L = 9/3 = 3.

Если значение *N*/*P* не равно целому числу и  $\text{mod}_{P}(N) = M$ , то длина сигнала на первом (верхнем) выходе будет равна<sup>1</sup> L = ant(N/P) + M, а на всех остальных — L = ant(N/P). Например, если N = 10, P = 3 и, соответственно,  $\text{mod}_{3}(10) = 1$ , то длина сигнала на первом выходе будет равна L = ant(10/3) + 1 = 4, а на всех остальных — L = ant(10/3) = 3.

При *установленном* флаге **Bus selection mode** входным сигналом является массив ячеек, а выходным — его элементы.

На рис. 7.2 приведен пример выделения трех векторов из *векторного сигнала* при сброшенном флаге **Bus selection mode**, а на рис. 7.3 — выделения трех элементов из массива ячеек (в фигурных скобках на выходе блока Mux) при установленном флаге;

- в виде вектора в квадратных скобках, количество элементов которого равно количеству выходов, а значения элементов — длинам выходных сигналов. Например, если на выходе формируются три вектора длиной 2, 5 и 2, то параметр Number of outputs будет задаваться вектором [2 5 2]. Независимо от представления входного сигнала, флаг Bus selection mode в этом случае должен быть сброшен;
- в виде вектора, элементы которого (один, нескольких или все) указываются как число –1. Независимо от представления входного сигнала, флаг **Bus selection mode** должен быть *сброшен*.

Если один элемент вектора задан числом -1, то длина соответствующего выходного сигнала определяется по остаточному принципу. Если все элементы равны -1, то длины выходных сигналов определяются так же, как и при задании параметра **Number of outputs** в виде целого числа. При нескольких элементах, равных -1, имеем комбинацию двух предыдущих ситуаций;

#### **Display option** — см. разд. 7.2.1.

При желании скрытое по умолчанию имя блока Demux можно восстановить по команде Format | Show Name.

 $<sup>^1</sup>$  Функция ant соответствует округлению до ближайшего целого в сторону уменьшения.







Рис. 7.3. Пример выделения элементов массива ячеек с помощью блока Demux при установленном флаге Bus selection mode

### 7.1.3. Блок Bus Creator

Блок *Bus Creator* объединяет в *сигнал шины* сигналы разного вида и типа данных. Блок имеет следующие параметры (рис. 7.4):

- способ указания имен входных сигналов (параметр в виде выпадающего списка без имени):
  - Inherit bus signal names from input ports (Наследовать имена сигналов шины от входных портов) в выходном *сигнале шины* имена (метки) сигналов наследуются от входных сигналов.

В этом случае имена входных сигналов создаются пользователем в окне S-моделей (см. табл. 2.2) или присваиваются по умолчанию и хранятся в списке Signals in bus как SignalN, где N— номер входа сверху вниз, без отображения в окне S-моделей.

При *закрытом* окне параметров имена, созданные пользователем в окне S-моделей, автоматически отображаются в списке Signals in bus (Сигналы в шине), а при *открытом* окне для этого следует нажать кнопку Refresh (Обновить).

Если входной сигнал является *сигналом шины*, то перед его именем автоматически ставится символ + (плюс).

Блок источника входного сигнала может быть *выделен цветом* в S-модели до запуска моделирования, для чего имя соответствующего сигнала следует выделить в списке Signals in bus и нажать кнопку Find (Найти);

• Require input signal names to match signals below (Потребовать, чтобы имена входных сигналов совпадали с именами ниже) — имена (метки) входных сигналов должны совпадать с выведенными в списке Signals in bus.

В этом случае пользователь может менять имена сигналов в списке Signals in bus, поочередно выделяя их, редактируя в поле Rename selected signal (Переименовать выделенный сигнал) и щелкая на старом имени в списке — оно будет заменено на новое.

Для изменения порядка следования имен удобно воспользоваться активизированными кнопками Up (Вверх) и Down (Вниз).

Все изменения, вносимые в окне параметров, необходимо синхронно *вручную* выполнить в окне S-моделей, в противном случае после запуска моделирования будет выдано сообщение об ошибке;

□ Number of inputs (Количество входов) — количество входных сигналов.

О параметрах Specify properties via bus object, Bus object и Output as nonvirtual bus речь шла ранее при описании параметров блока In (см. разд. 4.2.11). Отметим, что при сброшенном флаге Specify properties via bus object (по умолчанию) шина является виртуальной и представляет собой лишь удобное графическое средство отображения взаимодействия сигналов при их объединении с целью компактного изображения S-модели. В этом случае может использоваться уточняющий термин "сигнал виртуальной шины".

При желании скрытое по умолчанию имя блока Bus Creator можно восстановить по команде Format | Show Name.

В примере на рис. 7.4 при объединении сигналов в сигнал шины выполнено распространение имен (меток) сигналов (см. табл. 2.2).

Function Block Parameters: Bus Creator	
BusCreator	
This block creates a bus signal from its inputs.	
Parameters	
Inherit bus signal names from input ports	Sine wave Scope
Number of inputs: 3	
Find Find	Step
s2 Refresh	
55	
	5
Rename selected signal:	
Specify properties via bus object	
Bus object: BusObject Edit	
Output as nonvirtual bus	-5
OK Caprel Help Apply	0 2 4 6 8 10
	Time offset: 0

Рис. 7.4. Параметры блока Bus Creator и пример объединения сигналов в сигнал шины

### 7.1.4. Блок Bus Selector

Блок *Bus Selector* выделяет из *сигнала шины* сигналы разного вида и типа данных. Блок имеет следующие параметры (рис. 7.5):

□ Signals in the bus (Сигналы в шине) — список входных сигналов во входном *сигнале шины*.

Назначение кнопок Find и Refresh такое же, как для блока Bus Creator (см. разд. 7.1.3);

□ Selected signals — список выделяемых сигналов.

Активизированные кнопки Up и Down удобно использовать для выделения выходных сигналов в списке и изменения порядка их следования, а кнопку **Remove** (Удаление) — для удаления выделенного сигнала из списка;

□ **Output as bus** (Выход как шина) — флаг, при установке которого выделяемые сигналы объединяются в выходной сигнал шины.

При желании скрытое по умолчанию имя блока Bus Selector можно восстановить по команде Format | Show Name.

На рис. 7.5 представлены параметры и пример выделения сигналов из сигнала виртуальной шины.



Рис. 7.5. Параметры блока Bus Selector и пример выделения сигналов из сигнала виртуальной шины

### 7.1.5. Блок Goto

Блок Goto — беспроводной передатчик сигнала на вход блоков From. У блока задаются следующие параметры (рис. 7.6):

□ Goto Tag (Метка блока Goto) — метка (ярлык)<sup>1</sup> блока;

<sup>&</sup>lt;sup>1</sup> Метка может состоять из последовательности латинских букв, цифр и символа подчеркивания, начиная с буквы.

- □ **Tag Visibility** (Область видимости метки) список компонентов S-модели системы, для которых доступен блок с данной меткой:
  - local (локальная) локальная подсистема;
  - scoped (внутри) локальная подсистема и подсистемы более низкого уровня;
  - global (глобальная) любая компонента S-модели системы.

Для компонентов local и scoped метка внутри пиктограммы автоматически заключается в квадратные скобки, а для компонента global отображается без скобок.

Одному блоку Goto может соответствовать несколько блоков From, список меток которых выводится в поле **Corresponding From blocks** (Соответствующие блоки From). Щелчок на метке выделит цветом блок From, соответствующий блоку Goto, а второй щелчок — отменит это выделение.

Ссылка refresh (обновить) позволяет обновить список после выбора другой метки блока Goto в окне параметров блока From *(см. разд. 7.1.6)*;

- □ Icon Display (Текст внутри пиктограммы) варианты отображения текста внутри пиктограммы:
  - Tag метка блока Goto;
  - Signal name имя сигнала;
  - Tag and Signal name метка блока Goto и имя сигнала.

🖬 Sink Block Parameters: Goto		$\square$	A1
Goto Send signals to From blocks that have the specified tag. If tag then a Goto Tag Visibility block must be used to define the visib block icon displays the selected tag name (ocal tags are enclosed scoped tag names are enclosed in braces, (3). Parameters Goto Tag: A1. Tag Visibility: Jobb	p visibility is 'scoped', sed in brackets, [], and	[A 1] From	Goto Scope B.mat
Corresponding From blocks: Initiled/From unbled/From Icon Display: Tag	refresh Source Block From Receive signals from 'scoped' in the Goto visibility of the tag. name (local tags ar braces, (l). Parameters Goto Tag: A1 Goto Source: untitle Icon Display: Tag	From 1 Parameters: From In the Goto block with the specifie block, then a Goto Tag Visibility b After 'Update Diagram', the bloc e enclosed in brackets, [], and sc td/Goto	To File  d tag. If the tag is defined as alock must be used to define the k icon displays the selected tag coped tag names are enclosed in  Update Tags
		<u>o</u> k	<u>C</u> ancel <u>H</u> elp

Рис. 7.6. Параметры блоков Goto и From и пример их совместного использования

### 7.1.6. Блок From

Блок *From* — это беспроводной приемник сигнала с выхода блока Goto. У блока задаются следующие параметры (рис. 7.6):

□ Goto Tag — метка блока Goto, выбираемая из списка.

Если в списке указаны метки не всех блоков Goto, то для его расширения следует выбрать значение «More Tags».

Кнопка Update Tags (Обновить список) позволяет обновить список при изменении меток блоков Goto.

Щелчок на ссылке **Goto Source** (Источник Goto), активной при повторных открываниях окна параметров, выделит цветом блок Goto, соответствующий блоку From, а второй щелчок — отменит это выделение;

□ Icon Display — см. разд. 7.1.5.

### 7.1.7. Блок Manual Switch

Блок *Manual Switch* представляет собой ручной переключатель сигналов, не содержит входных параметров (пример на рис. 7.7).

Ручное управление между двумя входными сигналами осуществляется двойным щелчком левой кнопки мыши на пиктограмме блока до или в процессе моделирования.



Рис. 7.7. Пример использования блока Manual Switch

### 7.1.8. Блок Switch

Блок Switch представляет собой автоматический переключатель сигналов.

Блок имеет один выход и три входа для сигналов u1 (верхний), u2 (средний) и u3 (нижний). Управление между сигналами u1 и u3 осуществляется *управляющим* сигналом u2: если заданное для него *условие* выполняется, то на выход поступает сигнал u1, в противном случае — сигнал u3.

Параметры блока задаются на двух вкладках — Main и Signal Attributes.

На вкладке Main определяются параметры:

- □ Criteria for the first input (Критерий для первого входа) условие для управляющего сигнала u2 на среднем входе;
- □ Threshold (Порог) пороговое значение в условии Criteria for the first input;
- □ Enable zero crossing detection *см. разд. 4.2.9*;
- □ Sample time см. разд. 4.2.1.

На вкладке Signal Attributes параметры подобны параметрам блока Add (см. разд. 6.1.1.1).

На рис. 7.8 приведен пример управления сигналами u1 и u3, соответственно на выходах блоков Pulse Generator (см. разд. 4.2.12) и Sine Wave (см. разд. 4.2.19) с параметрами, заданными по умолчанию. Управляющим является сигнал u2 на выходе блока Step (см. разд. 4.2.20) с параметрами: Step time: 5; Initial value: 1; Final value: 2; Sample time: 0.

🗑 Function Block Parameters: Switch	Pulse Generator
Switch Pass through input 1 when input 2 satisfies the selected criterion; otherwise, pass through input 3. The inputs are numbered top to bottom (or left to right). The input 1 pass-through criteria are input 2 greater than or equal, greater than, or not equal to the threshold. The first and third input ports are data ports, and the second input port is the control port.	Step Switch Scope
Criteria for passing first input: U2 >= Threshold	
Image: Second	
	Time offset: 0

Рис. 7.8. Параметры блока Switch на вкладке Main и пример управления сигналами

### 7.1.9. Блок Selector

Блок Selector выделяет:

- □ элементы вектора и переупорядочивает их в выходном векторе;
- элементы строк и столбцов матрицы и переупорядочивает их в выходной матрице;

- □ сигналы в сигнале (2-D) *L*×*N* (см. табл. 4.3) и переупорядочивает их в выходном сигнале (2-D);
- □ сигналы в сигнале (3-D) *М*×*N*×*L* (см. табл. 4.3) и переупорядочивает их в выходном сигнале (3-D).

Для блока определяются следующие параметры (рис. 7.9—7.10):

□ Number of input dimensions (Количество размерностей на входе) — размерность входного сигнала: 1 — для вектора или сигнала (2-D); 2 — для матрицы или сигнала (3-D).

*Входной* сигнал может принадлежать к любому типу данных, поддерживаемому в Simulink, в том числе с ФТ, а *выходной* — дублирует тип данных входного сигнала;

- □ Index mode (Варианты индекса) см. разд. 6.2.3.3.
- □ Index Option (Варианты индекса) список для указания выделяемых элементов вектора или строк и столбцов матрицы.

При определении этого параметра необходимо иметь в виду, что входной сигнал 2-D рассматривается как *вектор* длины N (количество сигналов), а 3-D — как *матрица*  $M \times N$  (количество сигналов).

В списке для указания выделяемых элементов *вектора* отводится одна строка, а строк и столбцов *матрицы* — две строки: первая — для строк матрицы, а вторая — для ее столбцов.

В первом столбце списка Index Option в раскрывающемся списке могут быть выбраны следующие значения:

- Select all (Выделить все) для выделения всех элементов вектора или всех строк (столбцов) матрицы;
- Index vector (dialog) (Индекс вектора (диалог)) для выделения элементов вектора или строк (столбцов) матрицы.

Выделяемые элементы *вектора* указываются вектором со списком *индексов* элементов. В доступном поле **Input port size** (Размер входного порта) указывается длина входного вектора.

Выделяемые строки (столбцы) *матрицы* указываются вектором со списком *индексов* строк (столбцов). В поле **Sample time** (см. разд. 4.2.1) указывается интервал времени;

 Starting index (dialog) (Начальный индекс (диалог)) — при компактном выделении элементов входного сигнала с заданным начальным индексом и количеством выделяемых элементов.

Начальный индекс *вектора* указывается скаляром в поле Index Option, а количество выделяемых элементов — скаляром в поле Output Size (Размер выхода).

Начальный индекс строки (столбца) *матрицы* указывается скаляром в поле Index Option, а количество строк (столбцов) — скаляром в поле Output Size;

- Index vector (port) (Индекс вектора (порт)) подобен значению Index vector (dialog), но список индексов выделяемых элементов подается извне на автоматически создаваемый вход блока Selector;
- Starting index (port) (Начальный индекс (порт)) подобен значению Starting index (dialog), но начальный индекс подается извне на автоматически создаваемый вход блока Selector.

На рис. 7.10 для блока Sine Wave *(см. разд. 4.2.19)* установлены параметры: Sine type: Time based; Sample time: 0.2; Amplitude: [1 2 3 4 5].

Func	tion Bloc	:k Paramet	ers: Sele	ector		
elector elect or rom an i option" p	reorder sp nput port o parameter.	pecified eleme or this dialog.	nts of a m You can ch	ultidimensional input sign loose the indexing metho	al. The index to each of for each dimension	element is identified by using the "Index
aramet Iumber	ers of input dir	mensions: 2				
ndex m	ode: One-	based				~
In	Index Option			Index	0.	itput Size
1 Inc	lex vector	(dialog)	*	[1 2 4]	Inh	erit from "Index"
2 Ind	ex vector (	(dialog)	*	[2 3]	Inh	erit from "Index"
ample t	ime (-1 for	inherited): -	1	<u>O</u> K	<u>C</u> ancel	Help Apply
	2 5 8 11	3 6 9 12	→ U Sele	Y	2 5 11	3 6 12
	_				Display	
Constant						

Рис. 7.9. Параметры блока Selector и пример выделения строк и столбцов матрицы с их переупорядочиванием в выходной матрице



Рис. 7.10. Параметры блока Selector и пример выделения двух сигналов в сигнале (2-D) *L*×5 с их переупорядочиванием в выходном сигнале (2-D) *L*×2

### 7.1.10. Блок Merge

Блок Merge объединяет:

- $\square$  векторы одинаковых или разных длин *Li* в один вектор длины *L* = max{*Li*};
- □ *N* сигналов (2-D) *Li*×1 одинаковых или разных длин *Li* в один сигнал (2-D) *L*×*N* длины  $L = \max\{Li\}$ .

Для блока задаются следующие параметры (рис. 7.11, 7.12):

- □ Number of inputs (Количество входов) количество входов;
- □ Initial output (Начальный выход) начальные элементы векторов при объединении векторов или начальные сигналы при объединении сигналов.

При объединении *векторов* для каждого из них с помощью параметра **Input port offsets**, о котором пойдет речь далее, указывается сдвиг, согласованный с его длиной. При этом допускается вставка начальных элементов векторов, равных значению параметра **Initial output**.

При объединении *сигналов* для каждого из них с помощью параметра **Input port** offsets указывается сдвиг, согласованный с количеством сигналов. При этом допускается вставка начальных сигналов, все значения которых будут одинаковы и равны значению параметра **Initial output**.

Длина выходного сигнала будет равна *максимальной длине* входного сигнала, при этом сигналы с меньшими длинами будут автоматически продолжены до максимальной длины;

- Allow unequal port widths (Разрешить неодинаковые длины портов) флаг, который устанавливается при неодинаковых длинах объединяемых векторов или сигналов;
- □ Input port offsets (Ввести смещение портов) вектор сдвига, задающий сдвиг между объединяемыми векторами или сигналами. Нулевое значение соответствует отсутствию сдвига.



Рис. 7.11. Параметры блока Merge и пример объединения векторов без вставки начальных элементов



Рис. 7.12. Параметры блока Merge и пример объединения векторов со вставкой начальных элементов

#### Пример 7.1

На рис. 7.13 приведен пример объединения сигналов без вставки начальных сигналов. Для блока Sine Wave *(см. разд. 4.2.19)* установлены параметры: Sine type: Time based; Sample time: 0.2; Amplitude: [1 5], а для блока Sine Wave1 — Sample time: 0.4; Amplitude: 2.

Задавая время моделирования 10 с, определим размерности *входных* сигналов блока Merge (с именами × и ×1 сверху вниз) с помощью блоков То Workspace (на рис. 7.13 они отсутствуют):

```
>> size(x), size(x1)
ans =
    51    2
ans =
    26    1
```

На верхний вход блока Merge подается сигнал (2-D)  $L1 \times 1 = 51 \times 2$  — комбинация из двух сигналов одинаковой длины L1 = 51, а на нижний — сигнал (2-D)  $L2 \times 1 = 26 \times 1$  — один сигнал длины L2 = 26.

Выходной сигнал блока Merge у имеет размерность:

>> size(y) ans = **51 3** 

и объединяет в себе сигналы × и ×1 — комбинацию из трех сигналов одинаковой длины  $L = \max{Li} = LI = 51$  с автоматическим продолжением длины входного сигнала ×1.

При задании параметров Initial output: 100 и Input port offsets: [0 3] размерность выходного сигнала оказалась бы равной:

>> size(y) ans =

```
51 4
```

В этом случае третьему сигналу предшествовал бы начальный сигнал, все значения которого были бы равны 100.



Рис. 7.13. Параметры блока Merge и пример объединения сигналов без вставки начальных сигналов

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  Blocks  $\rightarrow$  Signal Routing (Simulink  $\rightarrow$  Блоки  $\rightarrow$  Маршрутизация сигнала).

### 7.2. S-модели средств управления атрибутами сигнала

S-модели *средств управления атрибутами сигнала* представлены в группе блоков Signal Attributes (Атрибуты сигнала), основные из которых приведены в табл. 7.2.

N⁰	Блок	Назначение	
1.	Bus to Vector	Преобразует сигнал виртуальной шины в векторный сигнал	
2.	Data Type Conver- sion	Преобразует тип данных входного сигнала	
3.	Data Type Conver- sion Inherited (mask)	Преобразует тип данных входного сигнала к типу данных эталонного сигнала	
4.	Data Type Duplicate (mask)	Контролирует сигналы на принадлежность к одинаковому типу данных	
5.	IC	Устанавливает начальное значение (IC — Initial Condition) сигнала на первом шаге моделирования	
6.	Probe	Исследует атрибуты сигнала	
7.	Rate Transition	Согласует передачу данных между блоками с различными скоростями (периодами дискретизации)	
8.	Width	Определяет длину вектора или количество элементов матрицы	

Таблица 7.2. Основные блоки группы Signal Attributes

Рассмотрим параметры блоков из табл. 7.2.

### 7.2.1. Блок Bus to Vector

Блок *Bus to Vector* преобразует сигнал виртуальной шины в векторный сигнал и не содержит параметров.

Преобразование в векторный сигнал возможно только в том случае, если сигнал виртуальной шины (см. блок Bus Creator в *разд. 7.1.3*) представлен комбинацией скаляров и/или векторов одинакового численного типа с одинаковым режимом дискретизации, задаваемого параметром **Sampling mode**. Векторы могут быть представлены векторами-строками (при установленном флаге Interpret vector parameters as 1-D для входных блоков), и тогда векторный сигнал будет представлен вектором-строкой (1-D), или векторами-столбцами (при сброшенном флаге Interpret vector parameters as 1-D), и тогда векторный сигнал будет представлен вектором-столбцом (2-D).

На рис. 7.14 приведен пример преобразования сигнала виртуальной шины, образованного комбинацией скаляра и двух векторов-столбцов, в векторный сигнал; выходной векторный сигнал представлен вектором-столбцом.



Рис. 7.14. Пример преобразования сигнала виртуальной шины в векторный сигнал с помощью блока Bus to Vector

### 7.2.2. Блок Data Type Conversion

Блок *Data Type Conversion* преобразует тип данных входного сигнала. Для блока определены следующие параметры (рис. 7.15):

- □ Output minimum и Output maximum см. разд. 4.2.4;
- □ Output data type (Тип данных на выходе) тип данных на выходе (см. табл. 4.5);
- $\Box$  Lock output data type setting against changes by the fixed-point tools *cm. pa30. 4.2.4*;
- □ Input and Output to have equal (Входные и выходные данные должны быть равны) варианты представления данных на выходе:
  - Real World Value (RWV) десятичное число;
  - Stored Integer (SI) десятичное число, рассчитанное по двоичному эквиваленту, полученному после округления десятичного числа до целого согласно выбранному режиму Integer rounding mode;
- □ **Integer rounding mode** *см. разд. 6.1.1.1*;
- □ Saturate on integer overflow см. разд. 6.1.1.1;
- □ Sample time см. разд. 4.2.1.

#### Пример 7.2

На рис. 7.15 приведен пример преобразования данных типа double в данные типа fixdt(1,8,7) (см. табл. 4.6) при выборе **Input and Output to have equal:** Real World

Value (RWV). Число 2.5555555, большее единицы, в заданном формате не может быть представлено. При установленном флаге Saturate on integer overflow используется арифметика насыщения, поэтому числу присвоено значение 0.9922, максимально возможное в заданном формате.

Десятичные числа (RWV), соответствующие двоичным эквивалентам, определим по правилам перевода двоичного числа в десятичное с учетом того, что старший бит знаковый, а вес младшего —  $2^{-7}$ :

```
>>1*2^(-1)+0*2^(-2)+0*2^(-3)+0*2^(-4)+1*2^(-5)+1*2^(-6)+1*2^(-7)
ans =
    0.5547
>>1*2^(-1)+1*2^(-2)+1*2^(-3)+1*2^(-4)+1*2^(-5)+1*2^(-6)+1*2^(-7)
ans =
    0.9922
```

Целочисленные эквиваленты (SI), соответствующие двоичным эквивалентам, определим по правилам перевода двоичного числа в десятичное с учетом того, что старший бит знаковый, а вес младшего —  $2^0$ :

```
>> 1*2^0+1*2^1+1*2^2+0*2^3+0*2^4+0*2^5+1*2^6
ans =
       71
>> 1*2^0+1*2^1+1*2^2+1*2^3+1*2^4+1*2^5+1*2^6
ans =
      127
                                                                                                Format : short
                                                                                                       0.5547
                                                                       sfix 8 En7 (2)
                                         double (2)
           10.5555555 2 .5555555 1
                                                         Convert
                                                                                                       0.9922
                                                 Data Type Conversion
                   Constant
                                                                                                     Display
     🖬 Function Block Parameters: Data Type Conversion
                                                                                            Format: binary (Stored Integer )
      Data Type Conversion
      Convert the input to the data type and scaling of the output.
                                                                                                        (SI) bin 0100 0111
      The conversion has two possible goals. One goal is to have the Real World Values of the
                                                                                                        (SI) bin 0111 1111
      input and the output be equal. The other goal is to have the Stored Integer Values of the
      input and the output be equal. Overflows and guantization errors can prevent the goal
                                                                                                          Display 1
      from being fully achieved.
                                                                                          Format: decimal (Stored Integer )
      Parameters
                                                                                                                      (SI)71
       Output minimum:
                                           Output maximum:
                                                                                                                     (SI) 127
       0
                                                                                                       Display 2
       Output data type: fixdt(1,8,7)
                                                                      >>
      Lock output data type setting against changes by the fixed-point tools
      Input and output to have equal: Real World Value (RWV)
      Integer rounding mode: Nearest
                                                                            ~
      Saturate on integer overflow
       Sample time (-1 for inherited):
       -1
```

Рис. 7.15. Параметры блока Data Type Conversion и пример преобразования данных типа double в данные с ФТ типа fixdt(1,8,7) при выборе Input and Output to have equal: Real World Value (RWV)

Apply

Help

Cancel

#### Пример 7.3

На рис. 7.16 приведен пример преобразования данных типа double в данные с  $\Phi$ T типа fixdt(1,8,2<sup>-5</sup>,0), где slope = 2<sup>(-5)</sup> и віаs = 0 (см. табл. 4.6). Остальные параметры блока Data Type Conversion для этого примера такие же, как на рис. 7.15. В этом случае на дробную часть будет отведено не 7, а 5 битов, и, соответственно, на целую часть — 2 бита.

Десятичные числа (RWV), соответствующие двоичным эквивалентам, определим по формуле (4.2):

```
>> Slope = 2^(-5); Bias = 0;
>> a = [18*Slope+Bias 82*Slope+Bias]
a =
0.5625 2.5625
```

а целочисленные эквиваленты (SI), соответствующие двоичным эквивалентам, — по формуле (4.3):

```
>> aint = [(0.5625-Bias)/Slope (2.5625-Bias)/Slope]
aint =
18 82
```

Это тождественно их определению по правилам перевода двоичного числа в десятичное с учетом того, что старший бит знаковый, а вес младшего — 2<sup>0</sup>:

```
>> 0*2^0+1*2^1+0*2^2+0*2^3+1*2^4+0*2^5+0*2^6
ans =
     18
>> 0*2^0+1*2^1+0*2^2+0*2^3+1*2^4+0*2^5+1*2^6
ans =
     82
                                                                         Format : shot
                                                                               0.5625
                                                                   2
                                                                                2.563
                                                                            Display
                                                                Format : binary (Stored Integer )
                                                                       (SI) bin 0001 0010
                                                   sfix 8 En 5
                           double (
                                       Convert
   [0.5555555 2 .5555555 ]
                                                                       (SI) bin 0101 0010
          Constant
                                 Data Type Conversion
                                                                         Display 1
                                                                    Format : decimal (Stored Integer )
                                                                                          (SI) 18
                                                                                          (SI) 82
```

Display 3

Рис. 7.16. Пример преобразования данных типа double в данные с ФТ типа fixdt(1,8,2<sup>^</sup>–5,0) с помощью блока Data Type Conversion при выборе Input and Output to have equal: Real World Value (RWV)

#### Пример 7.4

На рис. 7.17 приведен пример преобразования данных типа double в данные с  $\Phi$ T типа fixdt(1,8,7) при выборе **Input and Output to have equal:** Stored Integer (SI). Остальные параметры блока Data Type Conversion для этого примера такие же, как на рис. 7.15. Целочисленные эквиваленты [1 3] получены в результате округления десятичных чисел на входе. Десятичные числа на выходе определяются по правилам перевода двоичного эквивалента чисел [1 3] с учетом того, что старший бит знаковый, а вес младшего —  $2^{-7}$ :



Рис. 7.17. Пример преобразования данных типа double в данные с ФТ типа fixdt(1,8,7) с помощью блока Data Type Conversion при выборе Input and Output to have equal: Stored Integer (SI)

### 7.2.3. Блок Data Type Conversion Inherited

Блок Data Type Conversion Inherited преобразует тип данных входного сигнала к типу данных эталонного сигнала. Этот блок имеет два входа: верхний — для эталонного сигнала с наследуемым типом данных, нижний — для сигнала, наследующего тип данных эталонного сигнала. Для блока определены следующие параметры (рис. 7.18):

- □ Input and Output to have equal *см. разд. 7.2.2*;
- □ Integer rounding mode см. разд. 6.1.1.1;
- □ Saturate to max or min when overflows occurs (Насыщение до максимального или минимального значения при переполнении) аналогичен параметру Saturate on integer overflow блока Add (*см. разд. 6.1.1.1*).
| .873 double  | ∎<br>נ<br>כ  | Data Type<br>Conversion<br>Inherited  | Display   |
|--|--|---|---|
| istant 1   |  |   |   |
| 🕃 Function Block Pa  | ramete   | rs: Data Type Conversio   | n Inherited 🛛 🕅   |
|  |  |   |   |
| Conversion Inherited (ma   | ask) (link)  |   | 3   |
| <ul> <li>Conversion Inherited (ma<br/>Convert the second input</li> </ul>  | ask) (link)<br>: to the d  | lata type and scaling of the firs   | st input.   |
| Conversion Inherited (ma<br>Convert the second input<br>The conversion has two p<br>of the input and the outp<br>Values of the input and th<br>prevent the goal from be  | ask) (link)<br>to the d<br>oossible g<br>ut be eq<br>he output<br>ing fully a                      | lata type and scaling of the firs<br>goals. One goal is to have the<br>ual. The other goal is to have<br>: be equal. Overflows and qua<br>achieved.                     | st input.<br>Real World Values<br>the Stored Integer<br>nitization errors can |
| Conversion Inherited (ma<br>Convert the second input<br>The conversion has two p<br>of the input and the outp<br>Values of the input and th<br>prevent the goal from beil<br>Parameters  | ask) (link)<br>to the d<br>oossible g<br>ut be eq<br>ne output<br>ing fully a                      | lata type and scaling of the firs<br>goals. One goal is to have the<br>ual. The other goal is to have<br>: be equal. Overflows and qua<br>achieved.                     | st input.<br>Real World Values<br>the Stored Integer<br>Intization errors can |
| Conversion Inherited (ma<br>Convert the second input<br>The conversion has two p<br>of the input and the outp<br>Values of the input and th<br>prevent the goal from being<br>Parameters<br>Input and Output to have                         | ask) (link)<br>to the d<br>oossible g<br>ut be eq<br>he output<br>ing fully a<br>e equal:          | lata type and scaling of the firs<br>goals. One goal is to have the<br>ual. The other goal is to have<br>t be equal. Overflows and qua<br>achieved.<br>Real World Value | st input.<br>Real World Values<br>the Stored Integer<br>ntization errors can  |
| Conversion Inherited (ma<br>Convert the second input<br>The conversion has two p<br>of the input and the outp<br>Values of the input and th<br>prevent the goal from bei<br>Parameters<br>Input and Output to have<br>Integer rounding mode: | sk) (link)<br>to the d<br>possible g<br>ut be eque<br>e output<br>ing fully a<br>e equal:<br>Floor | lata type and scaling of the firs<br>goals. One goal is to have the<br>ual. The other goal is to have<br>t be equal. Overflows and qua<br>achieved.<br>Real World Value | st input.<br>Real World Values<br>the Stored Integer<br>Intization errors can |

Рис. 7.18. Параметры блока Data Type Conversion Inherited и пример наследования типа данных int8 от эталонного сигнала

# 7.2.4. Блок Data Type Duplicate

Блок *Data Type Duplicate* контролирует сигналы на принадлежность к одинаковому типу данных и выдает сообщение об ошибке при их несовпадении. Блок имеет один входной параметр — **Number of input ports** (Количество входных портов).

# 7.2.5. Блок ІС

Блок *IC* устанавливает начальное значение сигнала на первом шаге моделирования и имеет следующие параметры:

- □ Initial value (Начальное значение) начальное значение сигнала на первом шаге моделирования;
- □ Sample time см. разд. 4.2.1.

На рис. 7.19 приведен пример использования блока IC при Initial value: 3 и остальных параметрах блоков, заданных по умолчанию.



Рис. 7.19. Пример задания начального значения Initial value: 3 с помощью блока IC

### 7.2.6. Блок Probe

Блок *Probe* исследует атрибуты сигнала. Параметры этого блока задаются на двух вкладках — Main и Signal Attributes (рис. 7.20).

На вкладке **Main** параметры определяются состоянием пяти флагов. При *установленном* флаге на автоматически создаваемый выход поступает соответствующий сигнал, а именно:

- Probe width (Исследовать размер) длина вектора или количество элементов матрицы;
- □ Probe sample time (Исследовать интервал времени) значение параметра Sample time в виде вектора [<*интервал*> <*смещение*>] (см. табл. 4.2);
- □ Detect complex signal (Определить, является ли сигнал комплексным) единица, если сигнал комплексный, и ноль в противном случае;
- Probe signal dimensions (Исследовать размерность сигнала) размерность сигнала;
- Detect framed signal (Определить, является ли сигнал последовательностью фреймов) — единица, если сигнал — последовательность фреймов, и ноль в противном случае.

На вкладке Signal Attributes задается тип данных для приведенных параметров, по умолчанию double.



Рис. 7.20. Параметры блока Probe и пример исследования атрибутов матрицы

#### 7.2.7. Блок Rate Transition

Блок *Rate Transition* согласует передачу данных между блоками с различными скоростями (периодами дискретизации). Блок имеет следующие параметры (рис. 7.21):

- Ensure data integrity during data transfer (Гарантировать безошибочность при передаче данных) флаг, при установке которого гарантируется безошибочная передача данных в том случае, если отношение N периодов дискретизации выход/вход или вход/выход кратно целому числу; в противном случае следует сбросить флаг Ensure deterministic data transfer (maximum delay);
- □ Ensure deterministic data transfer (maximum delay) (Гарантировать кратность передачи данных (максимальную задержку)) флаг, при установке которого гарантируется отношение N периодов дискретизации выход/вход или вход/выход, кратное целому числу, и передача данных со скоростью "медленного" блока; в противном случае флаг должен быть сброшен;
- □ Initial conditions (Начальные условия) начальные условия, которые могут воспринимаются только при передаче данных от "медленного" блока к "быстрому", когда в начале передачи данные от "медленного" блока еще не поступили; в противном случае начальные условия игнорируются;

Sine Wave Sample time 0.1 Scope	Scope
Function Block Parameters: Rate Transition     RateTransition     RateTransition     Handle transfer of data between ports operating at different rates. Configuration     options allow you to trade off transfer delay and code efficiency for safety and     determinism of data transfer. The default configuration assures safe and deterministic     data transfer. The block's behavior depends on option settings and/or the sample     times of its inout and output norts. Undating the block diagram causes text on the	1 -2 0 2 4 6 8 10 Time offset: 0
block's icon to indicate its behavior as follows: ZOH: Zero Order Hold 1/z: Unit Delay Buf: Copy input to output under semaphore control Db_buf: Copy input to output, using double buffers Copy: Unprotected copy from input to output NoOp: No Operation	Scope1
Parameters   Parameters  Ensure data integrity during data transfer  Ensure deterministic data transfer (maximum delay)  Initial conditions:   O  Output port sample time options: Inherit	
QK <u>C</u> ancel <u>H</u> elp <u>Apply</u>	Time offset: 0

Рис. 7.21. Параметры блока Rate Transition и пример передачи данных от "быстрого" блока к "медленному" с отношением периодов дискретизации выход/вход *N* = 5

- □ Output port sample time options (Опции для периода дискретизации на выходе) варианты задания периода дискретизации значения параметра Sample time:
  - Specify (Установить) задается произвольно;
  - Inherit (Наследовать) наследуется от блока на выходе;
  - Multiple of input port sample time (Кратное периоду дискретизации входного сигнала) — определяется как произведение периода дискретизации входного блока на коэффициент, задаваемый значением параметра Sample time multiple >0 в доступном поле ввода.

В процессе моделирования действие блока отображается условной меткой (label) внутри его пиктограммы (табл. 7.3).

Таблица 7.3. Метки блока Rate Transition

Метка	Действие блока
ZOH	При отношении <i>N</i> периодов дискретизации выход/вход, кратном целому числу, удерживает (holds) значения <i>i*N</i> -го отсчета входного сигнала, <i>i</i> = 0, 1, … , на периоде дискретизации выходного сигнала
1/z	При отношении <i>N</i> периодов дискретизации вход/выход, кратном целому числу, задерживает (delaies) входной сигнал на период дискретизации

Окончание табл. 7.3

Метка	Действие блока
Buf	При отношении N скоростей выход/вход или вход/выход, <i>не</i> кратном целому числу, использует буфер для входного сигнала и семафор (средства синхронизации параллельных вычислений)
Db_buf	При отношении N скоростей выход/вход или вход/выход, не кратном целому числу, использует сдвоенный буфер для входного сигнала
Сору	При отношении N скоростей выход/вход или вход/выход, не кратном целому числу, использует незащищенную копию входного сигнала
NoOp	Не выполняет действий

В предпоследних трех случаях различные действия блока определяются автоматическим, скрытным от пользователя, выбором наилучшего метода передачи данных.

## 7.2.8. Блок Width

Блок *Width* определяет длину вектора или количество элементов матрицы и имеет следующие параметры (рис. 7.22):

- □ **Output data type mode** (Режим выбора типа выходных данных) способ выбора типа данных на выходе с возможными вариантами:
  - Choose intrinsic data type (Выбрать встроенный тип данных) выбирается в раскрывающемся списке **Output data type**;
  - Inherit via back propagation (Наследование через обратное распространение) — наследуется от предшествующего сигнала;
  - All ports same datatype (Все порты имеют одинаковый тип данных) совпадает с типом данных на входе;
- □ **Output data type** тип данных при выборе режима Choose intrinsic data type (см. табл. 4.4).

rs: Width	
al, using the specified outpu	ut data type.
ntrinsic data type	~
	~
	sl, using the specified outpunt

Рис. 7.22. Параметры блока Width и пример определения длины вектора

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  Blocks  $\rightarrow$  Signal Attributes (Simulink  $\rightarrow$  Блоки  $\rightarrow$  Атрибуты сигнала).

#### 7.3. S-модели задержки и хранения сигнала

S-модели задержки и хранения сигнала представлены в группе блоков Discrete (Дискретные) (табл. 7.4).

N⁰	Блок	Назначение
1.	Integer Delay	Задерживает дискретный сигнал на N периодов дискретизации
2.	Memory	Сохраняет значение сигнала на предыдущем шаге моделирования или значение дискретного сигнала на предыдущем периоде дискретизации
3.	Unit Delay	Задерживает дискретный сигнал на период дискретизации

Таблица 7.4. Блоки задержки и хранения сигнала в группе блоков Discrete

Рассмотрим параметры блоков из табл. 7.4.

#### 7.3.1. Блок Integer Delay

Блок *Integer Delay* задерживает дискретный сигнал на *N* периодов дискретизации и имеет следующие параметры (рис. 7.23):

- □ Initial condition (Начальное условие) значение выходного сигнала на первых *N* периодах дискретизации;
- □ Sample time см. разд. 4.2.1;



Рис. 7.23. Параметры блока Integer Delay и пример задержки дискретной синусоиды на пять периодов дискретизации

□ Number of delays (Количество задержек) — задержка дискретного сигнала, указываемая как *N* периодов дискретизации.

На рис. 7.23 для блока Sine Wave (см. разд. 4.2.19) установлены параметры Sine type: Sample based и Sample time: 0.1.

#### 7.3.2. Блок Memory

Блок *Memory* сохраняет значение сигнала на предыдущем шаге моделирования или значение дискретного сигнала на предыдущем периоде дискретизации. Параметры блока задаются на двух вкладках — **Main** и **State Attributes** (Атрибуты состояния) (рис. 7.24).

На вкладке Main определяются параметры:

- Initial condition (Начальное условие) начальное значение выходного сигнала до запуска моделирования;
- □ Inherit sample time (Наследовать значение интервала времени) флаг, при установке которого значение параметра Sample time наследуется от входного сигнала, в противном случае оно будет равно шагу моделирования;
- □ Treat as a unit delay when linearizing with discrete sample time (Трактовать как единичную задержку при линеаризации с дискретным интервалом времени) флаг, при установке которого блок функционирует как единичная задержка (блок Unit Delay, *см. разд. 7.3.3*) для дискретного сигнала.

На вкладке State Attributes параметры имеют отношение к генерации кода при работе в реальном времени и не влияют на процесс моделирования.

На рис. 7.24 приведен пример сохранения значений синусоиды (Sample time: 0) на предыдущем шаге моделирования при установленном флаге Inherit sample time.



Рис. 7.24. Параметры блока Memory на вкладке Main и пример сохранения значений синусоиды на предыдущем шаге моделирования

#### 7.3.3. Блок Unit Delay

Блок *Unit Delay* задерживает дискретный сигнал на период дискретизации. Параметры блока задаются на двух вкладках — Main и State Attributes.

На вкладке **Main** задаются параметры **Initial condition** и **Sample time**, подобные параметрам блока Integer Delay *(см. разд. 7.3.1)* (рис. 7.25). При открытой вкладке **State Attributes** параметры имеют отношение к генерации кода при работе в реальном времени и не влияют на процесс моделирования.

На рис. 7.25 приведен пример задержки дискретной синусоиды (Sample time: 0.1) на период дискретизации.



Рис. 7.25. Параметры блока Unit Delay и пример задержки дискретной синусоиды на период дискретизации

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  Blocks  $\rightarrow$  Discrete (Simulink  $\rightarrow$  Блоки  $\rightarrow$  Дискретные).



# Глава 8

# Подсистемы

Подсистемой (subsystem) называют S-модель, созданную из нескольких блоков и представленную в виде одного блока.

Создание подсистем позволяет решать следующие основные задачи:

- компактного представления S-моделей сложных систем;
- иерархического модульного подхода при разработке S-моделей сложных систем;
- компактного хранения типовых фрагментов, предназначенных для неоднократного использования;
- □ расширения библиотеки Simulink блоками пользователя;
- синхронизации параллельных процессов.

### 8.1. Разновидности подсистем

В Simulink различают две разновидности подсистем:

1. Неуправляемые.

Подсистему называют *неуправляемой*, если в процессе моделирования ее активизация не зависит от какого-либо условия.

Различают два вида неуправляемых подсистем:

- Виртуальная (virtual смоделированная) подсистема является "открытой" в процессе моделирования допускается связь входящих в ее состав блоков с блоками S-модели системы.
- *Невиртуальная* (nonvirtual) подсистема является "закрытой" в процессе моделирования она функционирует как единый (неделимый) блок.

Блок невиртуальной подсистемы выделяется жирной линией.

2. Управляемые.

Подсистему называют управляемой, если в процессе моделирования ее активизация зависит от некоторого условия. Управляемые подсистемы, по определению — *невиртуальные*, представлены двумя семействами:

- условные подсистемы (Conditional Subsystems);
- подсистемы, моделирующие логику управления потоком (Modeling Control Flow Logic).

#### 8.2. Неуправляемые подсистемы

Неуправляемые подсистемы могут создаваться двумя способами:

- 1. По команде Create Subsystem (Создать подсистему).
- 2. С помощью блока Subsystem (Подсистема) из группы блоков **Port & Subsystems** (Порты и подсистемы) библиотеки блоков Simulink.

#### 8.2.1. Команда Create Subsystem

Команда меню Edit | Create Subsystem (Редактирование | Создать подсистему) в окне S-моделей (или одноименная команда контекстного меню) используется для создания неуправляемой подсистемы как *фрагмента* S-модели системы.

С этой целью следует в окне S-моделей выделить требуемый фрагмент S-модели системы — группу компонентов (см. разд. 2.3.1), который после выполнения команды Create Subsystem будет автоматически заменен блоком Subsystem — подсистемой — с входными (In) и выходными (Out) портами, связывающими подсистему с S-моделью системы.

На рис. 8.1 представлена S-модель системы (математического преобразования сигналов) с выделенным фрагментом, а на рис. 8.2 — она же с заменой выделенного фрагмента блоком Subsystem. Имена входных (In1 и In2) (см. разд. 4.2.11) и выходного (Out1) (см. разд. 5.2.2) портов установлены по умолчанию.

Для просмотра и редактирования неуправляемой подсистемы достаточно дважды щелкнуть левой кнопкой мыши на пиктограмме блока Subsystem, после чего автоматически открывается *окно подсистем* с интерфейсом, дублирующим интерфейс окна S-моделей. Связь подсистемы с S-моделью системы осуществляется через входные (In1 и In2) и выходной (Out1) порты (см. рис. 8.2).

*Редактирование* подсистем ничем не отличается от редактирования S-моделей систем (см. разд. 2.3). Отредактированная подсистема после закрытия окна подсистем автоматически сохраняется под заданным именем с расширением mdl в составе S-модели системы.

Отменить команду Create Subsystem можно только *сразу после* создания подсистемы по активизированной команде Edit | Undo Create Subsystem, которая после выполнения каких-либо действий с S-моделью системы *блокируется*, поэтому ее предварительно рекомендуется копировать под другим именем.



Рис. 8.1. Исходная S-модель системы с выделенным фрагментом



Рис. 8.2. S-модель системы с блоком Subsystems

#### 8.2.2. Блок Subsystem

Блок *Subsystem* из группы блоков **Port & Subsystems** библиотеки блоков Simulink используется для создания неуправляемой подсистемы как *самостоятельного модуля*.

С этой целью следует в окно S-моделей перетащить блок Subsystem и дважды щелкнуть левой кнопкой мыши на его пиктограмме, после чего откроется окно подсистем с соединенными блоками In1 и Out1 (рис. 8.3). Удалив соединение, можно создавать подсистему по той же технологии, что и S-модель системы (см. гл. 2); входы подсистемы подсоединяются к портам In, а выходы — к портам Out.

Созданная или отредактированная подсистема после закрытия окна подсистем автоматически сохраняется под заданным именем с расширением mdl *в составе S-модели системы*.



Рис. 8.3. Окно S-моделей с блоком Subsystem и окно подсистем

#### 8.2.3. Параметры неуправляемых подсистем

Подобно любому блоку библиотеки Simulink, блок Subsystem имеет собственное окно параметров, которое открывается после *выделения блока* по команде меню Edit | Subsystem Parameters или одноименной команде контекстного меню и включает в себя следующие параметры (рис. 8.4):

- □ Show port labels (Показать метки портов) варианты изображения меток портов внутри пиктограммы блока Subsystem:
  - none метки не изображаются;
  - FromPortIcon (по умолчанию) изображение меток портов зависит от значения параметра **Icon Display** блоков In *(см. разд. 4.2.11)* и Out *(см. разд. 5.2.2)*: если оно равно Signal name или Port number and Signal name, то изображение меток портов соответствует меткам (именам) сигналов (см. табл. 2.2, п. 10), в противном случае — именам блоков In и Out;

- FromBlockName соответствуют именам блоков In и Out;
- From Signal соответствуют меткам *сигналов*, а в их отсутствие по умолчанию именам блоков In и Out;
- □ Read/Write permissions (Разрешение чтения/записи) варианты доступа к подсистеме:
  - ReadWrite (по умолчанию) разрешено чтение и редактирование с последующим сохранением;
  - ReadOnly разрешено только чтение;
  - NoReadOrWrite не разрешено ни чтение, ни редактирование;
- Name of error callback function (Имя функции, возвращающей ошибку) поле ввода для имени функции, которая может вызываться при возникновении ошибок в подсистеме в процессе моделирования; по умолчанию поле ввода является пустым и диагностика ошибок производится по общим правилам;
- □ Permit hierarchical resolution (Разрешить многоуровневое решение) варианты разрешения ссылок подсистемы на переменные Workspace:
  - All (по умолчанию) разрешены ссылки на все переменные;
  - ExplicitOnly разрешены ссылки только на переменные, используемые при задании параметров в блоках подсистемы;
  - None запрещены ссылки на все переменные;
- □ Treat as atomic unit (Трактовать как неделимую единицу) флаг, при установке которого неуправляемая подсистема Subsystem будет считаться *невиртуальной* и выделяться жирной линией, а при сбросе (по умолчанию) — виртуальной.

При *установке* флага **Treat as atomic unit** для невиртуальной подсистемы могут задаваться следующие дополнительные параметры:

- □ Minimize algebraic loop occurrences (Минимизировать алгебраические циклы) флаг, при установке которого блокируется выполнение всех арифметических циклов<sup>1</sup> в подсистеме;
- □ **Sample time** *см. разд. 4.2.1*;

Параметр **Real Time Workshop system code** относится к обработке в реальном времени, не рассматриваемой в рамках данной книги; по умолчанию устанавливается Auto.

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  User Guide  $\rightarrow$  Creating a Model  $\rightarrow$  Creating Subsystems (Simulink  $\rightarrow$  Руководство пользователя  $\rightarrow$  Создание модели  $\rightarrow$  Создание подсистем).

<sup>&</sup>lt;sup>1</sup> С фиксированным числом повторений.

Select the settings for the subsystem block. Parameters Show port labels FromPortIcon Read/Write permissions: ReadWrite Name of error callback function: Permit hierarchical resolution: All	> >
Parameters Show port labels FromPortIcon Read/Write permissions: ReadWrite Name of error callback function: Permit hierarchical resolution: All	> >
Show port labels FromPortIcon Read/Write permissions: ReadWrite Name of error callback function:	*
Read/Write permissions: ReadWrite Name of error callback function: Permit hierarchical resolution: All	~
Name of error callback function:	
Permit hierarchical resolution: All	
Permit hierarchical resolution: All	
	~
Treat as atomic unit	

Рис. 8.4. Параметры неуправляемой подсистемы — блока Subsystem

#### 8.2.4. Иерархическая структура S-модели системы

Иерархическая структура S-модели системы предполагает, что внутри имеющихся в ее составе подсистем могут создаваться *вложенные подсистемы* с неограниченным уровнем вложенности.

В левой части окна S-моделей по команде меню View | Model Browser Options | Model Browser (Вид | Варианты просмотра модели | Просмотр модели) можно увидеть иерархическую структуру S-модели системы с вложенными подсистемами, представленную в виде дерева (рис. 8.5).



Рис. 8.5. Иерархическая структура S-модели системы в окне Model Browser

1

2

3.

#### 8.3. Управляемые условные подсистемы

В группе блоков Port & Subsystems библиотеки блоков Simulink имеется три типа управляемых условных подсистем (Conditional Subsystems) (табл. 8.1).

Параметры данных блоков подобны параметрам блока Subsystem при установке флага Treat as atomic unit (см. разд. 8.2.3).

No Блок Наименование Enabled Subsystem Разрешенная подсистема — Е-подсистема с блоком Enable Triggered Subsystem Триггерная подсистема — Т-подсистема с блоком Trigger Enabled and Triggered Разрешенная и триггерная подсистема — ЕТ-подсистема с Subsystem блоками Enable и Trigger

Таблица 8.1. Управляемые условные подсистемы

#### 8.3.1. Блок Enabled Subsystem

Блок Enabled Subsystem (Разрешенная подсистема) представляет собой управляемую подсистему, коротко называемую *E-подсистемой*, которая активизируется положительным сигналом на управляющем входе.

В Е-подсистеме, для просмотра и редактирования которой следует открыть окно подсистем двойным щелчком левой кнопкой мыши на пиктограмме блока Enabled Subsystem, управляющий вход отображается наличием дополнительного блока En*able*, который может располагаться на произвольном месте.

На рис. 8.6 показано окно S-моделей с блоком Enabled Subsystem и окно подсистем с Е-подсистемой, содержащей, помимо входного и выходного портов, блок Enable.



Рис. 8.6. Окно S-моделей с блоком Enabled Subsystem и окно подсистем с блоком Enable

Параметры блока Enable включают в себя (рис. 8.7):

- □ States when enabling (Состояние при разрешении) состояние Е-подсистемы при активизации:
  - reset (сброс) начальное (исходное);
  - held (удерживание) последнее, когда подсистема была активной.

Состоянием подсистемы называют значения внутренних переменных, которые она может хранить к моменту активизации. Если такие переменные отсутствуют, то параметр **States when enabling** игнорируется;

□ Show output port (Показать выходной порт) — флаг, при установке которого у блока Enable появляется выход.

На выход блока Enable в E-подсистему поступает тот же сигнал, который имеется на ее управляющем входе, и он может быть использован для управления внутри подсистемы;

□ Enable zero crossing detection — *cm. pa*<sub>3</sub>∂. 4.2.9.

🗑 Block Parameters: Enable
Enable Port Place this block in a subsystem to create an enabled subsystem.
States when enabling: held
OK <u>C</u> ancel <u>H</u> elp Apply

Рис. 8.7. Параметры блока Enable

Выходные сигналы Е-подсистемы зависят от параметров выходных портов Out (см. разд. 5.2.2) и параметров блока Enable, а именно:

- □ при блокировании Е-подсистемы (не положительном сигнале на управляющем входе) значения выходных сигналов зависят от значений параметров Output when disable и Initial output блоков Out (см. разд. 5.2.2);
- при активизации Е-подсистемы (положительном сигнале на управляющем входе) значения выходных сигналов зависят от значения параметра State when enabling блока Enable.

На рис. 8.8 приведен пример Е-подсистемы 1 со следующим набором параметров. Для портов Out1 и Out2 *(см. разд. 5.2.2)* — параметры: **Output when disable:** reset; **Initial output:** 0. Для порта Out3 — параметры: **Output when disable:** held. Для блока Enable — параметры: **States when enabling:** held; флаг **Show output port** сброшен (по умолчанию).

В этом случае значения внутренних переменных, которые может хранить Еподсистема, отсутствуют, поэтому при ее *активизации* параметр States when enabling блока Enable игнорируется. При *блокировании* Е-подсистемы сигналы на выходах Out1 и Out2, согласно параметру Output when disable: reset, принимают значение Initial output: 0. Сигнал на выходе Out3, согласно параметру Output when disable: held, принимает (удерживает) последнее значение, когда подсистема была активной.



Рис. 8.8. Пример Е-подсистемы 1

На рис. 8.9 приведен пример Е-подсистемы 2 со следующим набором параметров. Для портов Out1, Out2 и Out3 — параметры: Output when disable: reset; Initial output: 0. Для блока Enable — параметры: States when enabling: held; флаг Show output port установлен.

В этом случае значения внутренних переменных, которые может хранить Еподсистема, также отсутствуют, поэтому при ее *активизации* параметр States when enabling блока Enable игнорируется. При *блокировании* Е-подсистемы сигналы на выходах Out1, Out2 и Out3, согласно параметру Output when disable: reset, принимают значения Initial output: 0. При установке флага Show output port у блока Enable появляется выход, который используется для управления на выходе Out3 внутри подсистемы.



Рис. 8.9. Пример Е-подсистемы 2

На рис. 8.10 приведен пример Е-подсистемы 3, выполняющей интегрирование входного сигнала со следующим набором параметров. Для порта Out1 — параметры Output when disable: reset: Initial output: 0. Для блока Enable — параметр States when enabling: reset (см. Scope слева) и States when enabling: held (см. Scope справа).

В этом случае Е-подсистема содержит внутренние переменные, поэтому при ее *активизации* и значении **States when enabling:** held (см. Scope справа) сигнал на выходе Out1 принимает (удерживает) последнее значение, при котором Е-подсистема была активной.

Е-подсистемы могут использоваться в качестве простого и удобного механизма *синхронизации параллельных процессов* в S-моделях систем. Следует иметь в виду, что при использовании в Е-подсистеме *дискретных* блоков и некратном соотношении между периодом (периодами) дискретизации и временем активизации/ блокирования подсистемы изменение выходных сигналов дискретных блоков не будет строго совпадать с моментами начала активизации/блокирования.



Рис. 8.10. Пример Е-подсистемы 3 с параметром блока Enable States when enabling: reset (см. Scope слева) и States when enabling: held (см. Scope справа)

#### 8.3.2. Блок Triggered Subsystem

Блок *Triggered Subsystem* (Триггерная подсистема) представляет собой управляемую подсистему, коротко называемую *T-подсистемой*, которая активизируется фронтом сигнала (изменением уровня) на управляющем входе.

В Т-подсистеме, для просмотра и редактирования которой следует открыть окно подсистем двойным щелчком левой кнопкой мыши на пиктограмме блока Triggered Subsystem, управляющий вход отображается наличием *блока Trigger*, который может располагаться на произвольном месте.

На рис. 8.11 показано окно S-моделей с блоком Triggered Subsystem и окно подсистем с T-подсистемой, содержащей, помимо входного и выходного портов, *блок Trigger*.



Рис. 8.11. Окно S-моделей с блоком Triggered Subsystem и окно подсистем с блоком Trigger

Параметры блока Trigger включают в себя (рис. 8.12):

- □ **Trigger type** (Тип триггера) типы запускающего события на управляющем входе, активизирующем Т-подсистему:
  - rising (возрастание) положительный фронт возрастание сигнала от нулевого или отрицательного значения до положительного значения (или нулевого при отрицательном начальном значении);
  - falling (убывание) отрицательный фронт убывание сигнала от положительного или нулевого значения до отрицательного значения (или нулевого при положительном начальном значении);
  - either (любой) положительный и отрицательный фронт (rising и falling).

При любом значении параметра **Trigger type** по завершении активизации Т-подсистемы она сохраняет последнее состояние, когда подсистема была активной, поэтому параметр **States when enabling** по умолчанию равен held (*см. разо. 8.3.1*).

□ Show output port (Показать выходной порт) — флаг, при установке которого у блока Trigger появляется выходной порт.

Через порт блока Trigger внутрь T-подсистемы поступает тот же сигнал, который имеется на ее управляющем входе. Этот сигнал может быть использован для управления внутри подсистемы.

Output data type (Тип данных на выходе) — типы данных на дополнительном выходе блока Trigger (см. табл. 4.4); параметр активизируется при установке флага Show output port; при выборе значений auto (по умолчанию) тип данных наследуется от сигнала на выходе блока Trigger.

Остальные параметры при указанных значениях параметра **Trigger type** не активизированы.

nace this block in	a subsystem to create a triggered subsystem.	
Parameters		
Trigger type: risi	ng	~
States when ena	bling: held	~
Show output	port	
Output data type	e: auto	×
Sample time type	triggered	~
Sample time:		
1		
Enable zero d	crossing detection	
Enable zero o	rossing detection	

Рис. 8.12. Параметры блока Trigger

Выходные сигналы Т-подсистемы зависят от параметра Initial output выходных портов Out, параметров блока Trigger, а также от флага Latch input by delaying outside signal входных портов In.

На рис. 8.13 приведен пример Т-подсистемы со следующими параметрами. Для портов Out1 и Out2 (см. разд. 5.2.2) — Initial output: 0. Для блока Trigger — Trigger type: either. Флаг Latch input by delaying outside signal входного порта In1 сброшен.



Рис. 8.13. Пример Т-подсистемы

На рис. 8.14 приведен пример простейшей Т-подсистемы, иллюстрирующий действие флага Latch input by delaying outside signal входного порта In1: при его установке сигнал на выходе блока In1 задерживается на один шаг моделирования.



Рис. 8.14. Действие флага Latch input by delaying outside signal для порта In1: сигнал на выходе порта при сбросе (см. Scope слева) и установке (см. Scope справа) флага

Т-подсистемы, наряду с Е-подсистемами, могут использоваться в качестве простого и удобного механизма *синхронизации параллельных процессов* в S-моделях систем. При использовании в Т-подсистеме одновременно непрерывных и дискретных блоков следует устанавливать **Sample time:** –1 (см. табл. 4.2).

# 8.3.3. Блок Enabled and Triggered Subsystem

Блок Enabled and Triggered Subsystem (Разрешенная и триггерная подсистема) представляет собой управляемую подсистему, коротко называемую *ET-подсистемой*, которая активизируется фронтом сигнала на "триггерном" управляющем входе при положительном сигнале на "разрешающем" управляющем входе.

В окне подсистем блока Triggered Subsystem "триггерный" управляющий вход ЕТ-подсистемы отображается наличием блока Trigger, а "разрешающий" — наличием блока Enable, которые могут располагаться на произвольном месте.

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  User Guide  $\rightarrow$  Creating Conditional Subsystems (Simulink  $\rightarrow$  Руководство пользователя  $\rightarrow$  Создание условных подсистем).

# 8.4. Управляемые подсистемы, моделирующие логику управления потоком

В группе блоков **Port & Subsystems** библиотеки блоков Simulink имеется четыре типа управляемых подсистем, моделирующих логику управления потоком (Modeling Control Flow Logic) (табл. 8.2).

Параметры данных блоков подобны параметрам блока Subsystem при установке флага **Treat as atomic unit** *(см. разд. 8.2.3)*.

N⁰	Блок	Наименование
1.	If Action Subsystem	Подсистема организации разветвления по условию, активизируемая блоком If
2.	Switch Case Action Subsystem	Подсистема организации разветвления в зависимости от значения входного сигнала, активизируемая блоком Switch Case
3.	For Iterator Subsystem	Подсистема выполнения арифметического цикла с помощью блока For Iterator
4.	While Iterator Subsys- tem	Подсистема выполнения итерационного цикла с помощью блока While Iterator

Таблица 8.2. Управляемые подсистемы, моделирующие логику управления потоком

#### 8.4.1. Блок If Action Subsystem

Блок *If Action Subsystem* (Подсистема, активизируемая блоком If) представляет собой управляемую подсистему, которая предназначена для организации *развет*вления по условию и активизируется управляющими сигналами, поступающими с внешнего блока *If*<sup>1</sup>.

<sup>&</sup>lt;sup>1</sup> Блок If может использоваться в S-моделях систем и самостоятельно.

Блок If, размещаемый в S-модели системы, является аналогом *оператора if* (разветвление по условию) языка MATLAB следующих форматов [22]:

1. Оператор if с одним условием:

```
if <ycловиe>
```

<фрагмент1>

else

<фрагмент2>

end

где <фрагмент> — фрагмент программы.

Условие может быть:

простым — логическим выражением типа:

<выражение1><операция отношения><выражение2>

где:

<выражение1>, <выражение2> — константы, простые переменные или арифметические выражения;

<операция отношения> — символ операции отношения (см. табл. 6.10);

 составным — совокупностью простых условий, объединенных символами логических операций (табл. 8.3).

Таблица 8.3. Символы логических операций

Символ	Операция
<b>&amp;</b>	И (AND) — "истинно", если значение условий "истинно"
Ι	ИЛИ (OR) — "истинно", если значение хотя бы одного условия "истинно"
~	НЕ (NOT) — "истинно", если значение условия "ложно" и наоборот

Действие оператора: если значение <условия> "истинно", то выполняется <фрагмент1>, если значение <условия> "ложно", то выполняется <фрагмент2>; после этого управление передается части программы, следующей за end.

2. Оператор if с несколькими условиями:

```
if <yсловие1>
<фрагмент1>
elseif <yсловие2>
<фрагмент2>
elseif <yсловие3>
<фрагмент3>
....
elseif <yсловиеN-1>
```

<фрагментN-1>

• • •

#### else

#### <фрагментN>

#### end

Действие оператора: если значение <условия1> "истинно", то выполняется <фрагмент1>, если значение <условия2> "истинно", то выполняется <фрагмент2> и т. д. вплоть до <условияN-1>; если значения всех данных условий "ложны", то выполняется <фрагментN>; после этого управление передается части программы, следующей за end.

Двум форматам *оператора* іf сопоставлены два *алгоритма работы блока* If: 1. Оператору if с *одним условием* — алгоритмы работы блока If:

```
if <ycловиe>
```

<If Action Subsystem1>

else

<If Action Subsystem2>

где <If Action Subsystem> — фактическое имя блока If Action Subsystem.

Алгоритм работы блока If: если значение <условия> "истинно", то на выходе if блока If появляется управляющий сигнал, активизирующий подсистему <*if* Action Subsystem1>. Если значение <*условия>* "ложно", то на выходе else блока If появляется управляющий сигнал, активизирующий подсистему <*if* Action Subsystem2>.

2. Оператору іf с несколькими условиями — алгоритмы работы блока If:

```
if <yсловиel>
```

<If Action Subsystem1>

elseif <*ycловиe2*>

<If Action Subsystem2>

elseif <ycлoвиe3>

<If Action Subsystem3>

- • •
- elseif *<условие*N-1>

<If Action SubsystemN-1>

... else

<If Action SubsystemN>

Алгоритм работы блока If: если значение <условия1> "истинно", то на выходе if блока If появляется управляющий сигнал, активизирующий подсистему <*if* Action Subsystem1>. Если значение <*условия2>* "истинно", то на *первом* (верхнем) выходе elseif блока If появляется управляющий сигнал, активизирующий подсистему <*if* Action Subsystem2>, и т. д. вплоть до <*условияN-1>*. Если значения всех данных условий "ложны", то на выходе else блока If появляется управляющий сигнал, активизирующий подсистему <*if* Action SubsystemN>. Если в списке *«условие2»*, *«условие3»*, ..., *«условием-1»* несколько условий имеют значение "истинно", то управляющий сигнал появляется на *первом* из них (первом из выходов elseif), и значения следующих условий *не проверяются*.

Параметры внешнего блока If включают в себя (рис. 8.15):

- □ Number of inputs (Количество входов) количество входов;
- □ If expression (Выражение if) простое или составное условие на выходе if, представленное в алгоритме работы блока If как <условие> или <условие1>;

Условие в поле **If expression** записывается по правилам языка MATLAB для оператора if с использованием тех же символов операций отношения (см. табл. 6.10) и логических операций (см. табл. 8.3).

В отличие от языка MATLAB, где по умолчанию переменные и результат вычислений считаются *матрицами*, в поле ввода **If expression** разрешается использовать только *простые переменные*, а результат вычислений должен быть *скаляром*.

Простые переменные (*скаляры* на входе блока If) должны иметь стандартные имена u1 (при одном входе), u2 и т. д. Если входным сигналом является *вектор*, то его элементы трактуются как простые переменные и обозначаются как u1(1), u1(2) и т. д. (для первого входа), u2(1), u2(2) и т. д. (для второго входа) и т. д.;

□ Elseif expressions (Выражения elseif) — простые или составные условия, записываемые через запятую, на *выходах* elseif, представленные в алгоритме работы блока If как <условие2>, … , <условие№-1>.

Запись условий в поле Elseif expressions производится по тем же правилам, что и в поле If expression. Количество условий не ограничено.

Если несколько условий имеют значение "истинно", то после первого из них значения следующих условий *не проверяются*;

- □ Show else condition (Показать условие else) флаг, при сбросе которого в блоке If будет отсутствовать выход else;
- □ Enable zero crossing detection *cm. pa3∂. 4.2.9*;
- □ Sample time см. разд. 4.2.1.

Все блоки подсистем If Action Subsystem должны иметь *одинаковые* значения параметра **Sample time**, такие же, как в блоке If.

В подсистемах If Action Subsystem, для просмотра и редактирования каждой из которых следует открыть окно подсистем двойным щелчком левой кнопки мыши на пиктограмме блока, управляющие входы отображаются наличием *блока Action port* с именем соответствующего управляющего входа if, elseif или else блока If. Блок Action port может располагаться на произвольном месте окна подсистем.

На рис. 8.16—8.17 приведены примеры комбинаций блока If с простейшими подсистемами If Action Subsystem, работающими согласно описанным для блока If алгоритмам 1 и 2.

Function Block Parameters: If	×
_ If Block	
IF expression	
Run the Action Subsystem connected to 1st output port	
Run the Action Subsystem connected to 2nd output port	
ELSE	
Run the Action Subsystem connected to last output port	
The number of Elseif output ports in the block is equal to the number of comma-separated Elseif expressions entered in the dialog. The If and Elseif expressions can use these MATLAB operators: $\langle , \langle =, =, \sim =, \rangle > \rangle = $ , $\langle , , \sim , \rangle \rangle$ , unary-minus on the input port signals named U.I. Q. U.3. etc.	
Parameters	
Number of inputs:	
1	
If expression (e.g. u1 ~= 0):	
u1~=5	
Elseif expressions (comma-separated list, e.g. u2 ~= 0, u3(2) < u2):	
Show else condition	-
Enable zero crossino detection	
Sample time (-1 for inherited):	
-1	
	_
QK <u>Cance</u> <u>H</u> elp Appl	У.

Рис. 8.15. Параметры блока If



Рис. 8.16. Пример комбинаций оператора If с простейшими подсистемами If Action Subsystem, работающими согласно алгоритму 1



Рис. 8.17. Пример комбинаций оператора If с простейшими подсистемами If Action Subsystem, работающими согласно алгоритму 2

#### 8.4.2. Блок Switch Case Action Subsystem

Блок Switch Case Action Subsystem (Подсистема, активизируемая блоком Switch Case) представляет собой управляемую подсистему, которая предназначена для организации разветвления в зависимости от значения входного сигнала и активизируется управляющими сигналами, поступающими с внешнего блока Switch Case<sup>1</sup>.

<sup>&</sup>lt;sup>1</sup> Блок Switch Case может использоваться в S-моделях систем и самостоятельно.

Блок Switch Case, размещаемый в основной S-модели, является аналогом *оператора* switch языка MATLAB следующего формата [22]:

```
switch <==parenee>
case <=navenue>
<@parment1>
case <=navenue2>
<@parment2>
...
otherwise
<@parmentN>
```

end

где <выражение> — арифметическое выражение, значение которого известно;

*<фрагмент>* — фрагмент программы.

Действие оператора: в зависимости от значения <выражения> управление передается соответствующему <фрагменту>. Если значение выражения не равно ни одному из указанных, то управление передается <фрагментуN>; после этого оно передается части программы, следующей за end.

Данному формату *оператора* switch соответствует следующий алгоритм работы блока Switch Case:

switch <входной сигнал>

```
case <sHayeHMel>
<Switch Case Action Subsysteml>
case <sHayeHMe2>
<Switch Case Action Subsysteml>
```

```
default
```

<Switch Case Action SubsystemN>

где:

<входной сигнал> — единственный входной сигнал блока Switch Case — скаляр; если его значение не является целым числом, то дробная часть автоматически отбрасывается;

<значение1>, <значение2>, ... — варианты значений <входного сигнала> — целые числа;

<switch Case Action Subsystem> — фактическое имя блока Switch Case Action Subsystem.

Алгоритм работы Switch Case: в зависимости от значения <входного сигнала> управляющий сигнал блока Switch Case активизирует соответствующую подсистему <Switch Case Action Subsystem>. Если значение <входного сигнала> не равно ни одному из указанных, то управляющий сигнал активизирует подсистему <Switch Case Action Subsystem>.

Параметры внешнего блока Switch Case включают в себя (рис. 8.18):

□ Case conditions (Условия case) — варианты значений входного сигнала (целые числа), перечисленные через запятую в фигурных скобках, например {1,2}.

Можно задавать сразу несколько перечисленных через запятую значений входного сигнала в квадратных скобках, например  $\{1,[3,5]\}$ . Тогда на первом управляющем выходе блока Switch Case сигнал будет формироваться, если значение входного сигнала равно 1, а на втором — если оно равно 3 или 5. При шаге, равном 1, значения входного сигнала можно задавать в виде  $\{M:N\}$ , например  $\{7,2:4\}$ . Тогда на первом управляющем выходе блока Switch Case сигнал будет формироваться, если значение  $\{7,2:4\}$ . Тогда на первом управляющем выходе блока Switch Case сигнал будет формироваться, если значение входного сигнала равно 7, а на втором — если оно равно 2, 3 или 4.

- □ Show default case (Показать условие default) флаг, при сбросе которого в блоке Switch Case будет отсутствовать выход default;
- □ Enable zero crossing detection *cm. pa3∂. 4.2.9*;
- □ Sample time *см. разд. 4.2.1*.

🗑 Function Block Parameters: Switch Case
SwitchCase Block
Perform a switch-case operation on the input. The input must be a scalar and its value will be truncated. The case conditions are specified using the MATLAB cell notation, where each case is a cell element. For example, entering {1,[2,3]} as the case condition implies that port 1 is run when the input is 1 and port 2 is run when the input is either 2 or 3. If the default case is shown, then port 3 is run for all other inputs.
Parameters Case conditions (e.g. {1,[2,3]}):
{1,-1}
Show default case:
<ul> <li>Enable zero crossing detection</li> </ul>
Sample time (-1 for inherited):
-1
<u>QK</u> <u>Cancel</u> <u>H</u> elp <u>A</u> pply

Рис. 8.18. Параметры блока Switch Case

В подсистемах Switch Case Action Subsystem, для просмотра и редактирования каждой из которых следует открыть окно подсистем двойным щелчком левой кнопки мыши на пиктограмме блока, управляющие входы отображаются наличием блока Action port с именем соответствующего управляющего входа case или default блока Switch Case. Блок Action port может располагаться на произвольном месте окна подсистем.

На рис. 8.19 приведен пример комбинаций блока Switch Case с простейшими подсистемами Switch Case Action Subsystem.



Рис. 8.19. Пример комбинаций блока Switch Case с простейшими подсистемами Switch Case Action Subsystem

#### 8.4.3. Блок For Iterator Subsystem

Блок For Iterator Subsystem (Подсистема выполнения цикла for) представляет собой управляемую подсистему, которая предназначена для выполнения *арифметического цикла* (с заранее известным числом повторений) в течение *шага моделирования* с количеством повторений, задаваемым с помощью блока For Iterator или от внешнего источника.

В подсистеме For Iterator Subsystem, для просмотра и редактирования которой следует открыть окно подсистем двойным щелчком левой кнопки мыши на пиктограмме блока, содержится *блок For Iterator*, который может располагаться на произвольном месте.

На рис. 8.20 показано окно S-моделей с блоком For Iterator Subsystem и окно подсистем с подсистемой, содержащей, помимо входного и выходного портов, *блок For Iterator*.



Рис. 8.20. Окно S-моделей с блоком For Iterator Subsystem и окно подсистем с блоком For Iterator

Параметры блока For Iterator включают в себя (рис. 8.21):

- □ States when starting (Состояние при запуске) состояние подсистемы перед первым циклом со значениями:
  - reset (сброс) начальное, до первого цикла;
  - held (удерживание) после последнего цикла.

*Состоянием подсистемы* называют значения внутренних параметров, которые она может хранить к моменту активизации. Если такие переменные отсутствуют, то значение параметра **States when starting** игнорируется;

- □ Iteration limit source (Источник числа повторений) источник, задающий количество повторений:
  - internal (внутренний) внутренний, задаваемый в поле ввода Iteration limit;
  - external (внешний) внешний, задаваемый в S-модели системы и подаваемый на *вход N* блока For Iterator;
- □ Iteration limit (Количество повторений) поле ввода для количества повторений при значении параметра Iteration limit source: internal;
- Set next i (iteration variable) externally (Установить следующее значение числа повторений извне) флаг, при установке которого арифметический цикл повторяется на следующем шаге моделирования, при этом число повторений

задается в S-модели системы и подается на вход Next (Следующий) блока For Iterator; активен только при установке флага Show iteration variable;

- □ Show iteration variable (Показать количество повторений) флаг, при установке которого у блока For Iterator появляется *выход* N для вывода количества повторений;
- □ Index mode (Режим для индекса) порядок отсчета количества повторений в цикле:
  - One-based (На основе единицы) начиная от единицы;
  - Zero-based (На основе нуля) начиная от нуля;
- □ Iteration variable data type (Тип данных для количества повторений) тип данных для количества повторений (см. табл. 4.4).

Source Block Pa	nameters: For Iterator	
Run the blocks in this specified either in the variable is incremente external input port, o until the iteration vari will output the current started, any states in their previous value.	subsystem for a number of iterations. The iteration limit m block's dialog or through an external input port. If the iter d externally, then the next iteration value is read in throug therwise it is incremented by one. The iteration continues able exceeds the iteration limit. If the output port is shown iteration number starting at zero or one. When the iterat the subsystem may be either reset to their initial value or	ay be ration jh an to run n, it ion is held at
Parameters		
States when starting:	held	~
Iteration limit source:	internal	~
Iteration limit:		
5		
Set next i (iteration	on variable) externally riable	
Index mode: One-ba	sed	~
Iteration variable dat	a type: int32	~
	QK <u>C</u> ancel <u>H</u>	elp

Рис. 8.21. Параметры блока For Iterator

Блок *For Iterator* является аналогом *оператора* for языка MATLAB следующего формата [22]:

```
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 он равен 1, его можно не указывать);

<тело цикла> — циклически повторяющийся фрагмент программы.

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

Данному формату *оператора* for соответствуют следующий алгоритм работы блока For Iterator при значении <шага> по умолчанию, равном единице:

```
for <нач.значение>:<кон.значение>
```

```
<For Iterator Subsystem>
```

**end** где:

<нач. значение> — отсчитывается от единицы или нуля в зависимости от значения параметра Index mode;

<кон. значение> — определяется значением параметра Iteration limit при выборе Iteration limit source: internal или задается извне при выборе Iteration limit source: external;

<For Iterator Subsystem> — циклически повторяющийся фрагмент, выполняемый подсистемой For Iterator Subsystem.

Алгоритм работы блока For Iterator: при изменении значений счетчика от <нач. значения> до <кон. значения> с шагом, равным единице, за время, равное шагу моделирования, циклически повторяется фрагмент, выполняемый подсистемой <For Iterator Subsystem>. После этого выдается сигнал на выходе подсистемы.

#### Пример 8.1

Создать подсистему For Iterator Subsystem для вычисления суммы  $\sum_{n=0}^{19} [0, 5^n + \sin(n)]$ .

Проверить результат в MATLAB.

На рис. 8.22 приведен пример организации арифметического цикла с помощью подсистемы For Iterator Subsystem и параметры блока For Iterator. Для согласования взаимодействия блоков подсистемы задано **Sample time:** –1. Для блоков Метогу (см. разд. 7.3.2) установлен флаг **Inherit sample time** и задан параметр **Initial condition:** 0.

Вычисление в MATLAB в режиме прямых вычислений дает тот же результат: >> s=0;

```
>> for n=0:19,s=s+0.5.^n+sin(n);end
>> s
s =
2.0853
```



Рис. 8.22. Пример организации арифметического цикла для вычисления суммы с помощью подсистемы For Iterator Subsystem

### 8.4.4. Блок While Iterator Subsystem

Блок *While Iterator Subsystem* (Подсистема выполнения цикла while) представляет собой управляемую подсистему, которая предназначена для выполнения итерационного цикла (с заранее неизвестным числом повторений) в течение *шага моделирования* с количеством повторений, задаваемым с помощью блока While Iterator.

В подсистеме While Iterator Subsystem, для просмотра и редактирования которой следует открыть окно подсистем двойным щелчком левой кнопки мыши на пиктограмме блока, содержится *блок While Iterator*, который может располагаться на произвольном месте.

На рис. 8.23 показано окно S-моделей с блоком While Iterator Subsystem и окно подсистем с подсистемой, содержащей, помимо входного и выходного портов, *блок While Iterator*.

Параметры блока While Iterator включают в себя (рис. 8.24):

Maximum number of iterations (Максимальное количество повторений) — максимально возможное количество повторений в итерационном цикле; при выборе значения –1 ограничение отсутствует;

- □ While loop type (Тип цикла while) разновидности итерационного цикла while и do-while, о которых пойдет речь далее;
- **Given States when starting** (см. разд. 8.4.3);
- □ Show iteration number port (Показать выход с номером повторения) флаг, при установке которого у блока While Iterator появляется выход для вывода количества повторений;
- □ Iteration variable data type (Тип данных для числа повторений) тип данных для количества повторений (см. табл. 4.4); доступен при установке флага Show iteration number port;
- □ Output data type (Тип данных на выходе) тип данных на выходе блока While Iterator (см. табл. 4.4); активизируется при установке флага Show iteration number port.



Рис. 8.23. Окно S-моделей с блоком While Iterator Subsystem и окно подсистем с блоком While Iterator

SINK DIOCK Para	meters: While Iterator	E E
While Iterator		
Run the blocks in this maximum number of it blocks in the subsyste otherwise an externa be run on the first iter iteration number start	subsystem until the while-iterator condition is irrations is reached. If the block is in do-while in will be run once before checking the while-i signal must be fed into the IC port to check it ation. If the output port is shown, it will output ing at one.	false or the e mode, all the terator condition, f the block should ut the current
Parameters		
Parameters Maximum number of i	terations (-1 for unlimited):	
Parameters Maximum number of i	terations (-1 for unlimited):	
Parameters Maximum number of i 5 While loop type: whil	terations (-1 for unlimited):	
Parameters Maximum number of i 5 While loop type: whil States when starting	terations (-1 for unlimited): e held	v
Parameters Maximum number of i 5 While loop type: whil States when starting Show iteration nu	terations (-1 for unlimited): e held mber port	×
Блок *While Iterator* является аналогом *оператора* while языка MATLAB следующего формата [22]:

```
while <ycловиe>
```

#### <тело цикла>

#### end

где *«условие»* — простое условие *(см. разд. 8.4.1)*, причем в *«теле цикла»* встречается, по крайней мере, одна переменная, входящая в *«условие»*.

Действие оператора: «тело цикла» циклически повторяется до тех пор, пока «условие» "истинно", после чего управление передается части программы, следующей за end.

Данному формату *оператора* while соответствуют *две разновидности алгоритма работы блока* While Iterator:

1. При значении параметра While loop type: do-while:

while <ycлoвиe>

<While Iterator Subsystem>

end

где:

<условие> — условие, простое или составное (см. разд. 8.4.1), подаваемое на вход cond блока While Iterator внутри подсистемы;

«While Iterator Subsystem» — циклически повторяющийся фрагмент, выполняемый подсистемой While Iterator Subsystem.

Алгоритм работы блока While Iterator: за время, равное шагу моделирования, фрагмент, выполняемый подсистемой «While Iterator Subsystem», циклически повторяется до тех пор, пока значение «условия» "истинно" и количество повторений не превосходит значения параметра Maximum number of iterations. После этого выдается сигнал на выходе подсистемы.

2. При значении параметра While loop type: while:

```
if <yсловие IC>
while <yсловие>
<While Iterator Subsystem>
end
else
```

end

где *«условие 1С»* — условие, простое или составное *(см. разд. 8.4.1)*, подаваемое на *вход* IC блока While Iterator *извне* подсистемы (из S-модели системы).

Алгоритм работы блока While Iterator: если значение <условия IC> "истинно", то за время, равное *шагу моделирования*, фрагмент, выполняемый подсистемой <while Iterator Subsystem>, циклически повторяется до тех пор, пока значение <условия> "истинно", и количество повторений не превосходит значения параметра Maximum number of iterations. После этого выдается сигнал на вы-

ходе подсистемы. Если значение *«условия IC»* "ложно", то подсистема *«while Iterator Subsystem»* блокируется.

#### Пример 8.2

Создать подсистему While Iterator Subsystem типа do-while для вычисления

суммы  $\sum_{n=0}^{\infty} [0, 5^n + \sin(n)]$  с точностью до  $10^{-2}$ . Вывести число повторений, значения

суммы и погрешности ее вычисления. Проверить результат в MATLAB.

На рис. 8.25 приведен пример организации итерационного цикла с помощью подсистемы While Iterator Subsystem и параметры блока While Iterator. Для согласования взаимодействия блоков подсистемы задано **Sample time:** –1. Для блоков Метоry (*см. разд. 7.3.2*) установлен флаг **Inherit sample time** и задан параметр **Initial condition:** 0.



Рис. 8.25. Пример организации итерационного цикла типа do-while для вычисления суммы с помощью подсистемы While Iterator Subsystem

Вычисление в MATLAB в режиме прямых вычислений дает тот же результат:

```
>> n=0;s0=0;ep=100;
>> while ep>1e-2,s=s0+0.5.^n+sin(n);,ep=abs(s-s0);s0=s;n=n+1;end
>> [n s ep]
ans =
23.0000 3.8260 0.0089
```

#### Пример 8.3

Создать подсистему While Iterator Subsystem типа *while* для вычисления суммы  $\sum_{n=0}^{\infty} [0, 5^n + \sin(n)]$  с точностью до  $10^{-2}$ , если условие на входе IC "истинно". Вывести

значение условия на входе IC, число повторений, значения суммы и погрешности ее вычисления. Проверить результат в MATLAB.

На рис. 8.26 приведен пример организации итерационного цикла с помощью подсистемы While Iterator Subsystem с параметром While loop type: while (в отличие от do-while на рис. 8.25). Для согласования взаимодействия блоков подсистемы задано Sample time: –1. Для блоков Memory (*см. разд. 7.3.2*) установлен флаг Inherit sample time и задан параметр Initial condition: 0.



Рис. 8.26. Пример организации итерационного цикла типа while для вычисления суммы с помощью подсистемы While Iterator Subsystem

Вычисление в MATLAB в режиме прямых вычислений дает тот же результат:

```
>> n=0;s0=0;ep=100;
>> if (3+(-2)<=3)while ep>1e-2,s=s0+0.5.^n+sin(n);,
...ep=abs(s-s0);s0=s;n=n+1;end,end
>> [n s ep]
ans =
```

23.0000 3.8260 0.0089

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  User Guide  $\rightarrow$  Creating a Model  $\rightarrow$  Modeling Control Flow Logic (Simulink  $\rightarrow$  Руководство пользователя  $\rightarrow$  Создание модели  $\rightarrow$  Моделирование логики управления потоком).

### 8.5. Маскированные подсистемы

Маскированием подсистемы называют процедуру Simulink, которая позволяет скрыть структуру исходной подсистемы "под маской" и представить ее в виде отдельного блока — блока маскируемой виртуальной подсистемы, обладающего всеми признаками стандартного блока библиотеки Simulink: собственными именем, пиктограммой, окном параметров, справкой Simulink в формате HTML и т. п.

На языке MATLAB аналогом маскированной подсистемы можно считать *function-файл* — внешнюю функцию [22]. Подобно тому, как разработка function-файлов есть средство расширения библиотеки функций, разработка маскированных подсистем есть средство расширения библиотеки блоков.

Создание маскированной подсистемы будем иллюстрировать на примере простейшей S-модели системы, содержащей неуправляемую *исходную подсистему Subsystem*, предназначенную для вычисления значений многочлена с помощью блока Polynomial *(см. разд. 6.1.6.2)* с последующим математическим преобразованием (рис. 8.27).

Для *создания маскированной подсистемы* необходимо выполнить следующие действия:

- Скопировать S-модель системы с исходной подсистемой в новое окно S-моделей и заменить стандартное имя подсистемы (Subsystem) именем, выбранным пользователем (Poly) (рис. 8.28).
- 2. Подготовить исходную подсистему к маскированию:
  - выбрать переменные, ассоциируемые с параметрами блоков исходной подсистемы:
    - vector с параметром Polynomial Coefficients блока Polynomial (см. разд. 6.1.6.2);
    - k с параметром Gain блока Gain (см. разд. 6.1.1.5);
    - b с параметром Constant value блока Constant (см. разд. 4.2.4).

🖬 poly_subs *
Eile Edit View Simulation Format Tools Help
🗋 😂 🛃 🚳 🖏 🕸 💼 💼 🔶 今 수 🏳 으 오 🕨 🗧 🚺 Normal 💽
(1 2 3 4 5) Constant Subsystem Display
poly_subs/Subsystem *
Elle Edit View Simulation Format Tools Help
D 📽 🖬 🚭 🐰 🖻 🖻 🔶 수 수 ↑ 🗠 오 🕨 = 10.0 Normal
$\begin{array}{c} 1 \\ In1 \\ \hline \\ P(u) \\ O(P) = 5 \\ \hline \\ Polynomial \\ \hline \\ Gain \\ Add \\ Out1 \\ \hline \\ Add \\ Out1 \\ \hline \\ Out1 \\ \hline \\ Out1 \\ \hline \\ Out1 \\ \hline \\ Constant \\ \end{array}$
Ready 100% ode45 //

Рис. 8.27. S-модель системы с исходной подсистемой Subsystem

mask_poly_subs *			
<u>File Edit View Simulation</u>	n F <u>o</u> rmat <u>T</u> ools	Help	
🗅   🛎 🖬 🚭   🐰	b <b>6</b>   4 4	· ☆   Ω ⊆   ▶ ■ 10.0 N	omal 💌
(1 2 3 4 5) Constant	Out1 Poly	Display	
Ready	100%	ode45	

Рис. 8.28. Копия S-модели системы с исходной подсистемой Poly

*Имя переменной* составляется по правилам языка MATLAB — как последовательность латинских букв, цифр и символов подчеркивания, начинающаяся с буквы; прописные и строчные буквы различаются.

Переменные будут играть роль "посредников" между параметрами маскируемой подсистемы и параметрами стандартных блоков исходной подсистемы, связывая их через переменные, сохраняемые в рабочей области памяти маскируемой подсистемы **Mask Workspace**. Область памяти **Mask Workspace** является *локальной*, что исключает конфликт с переменными в области памяти **Workspace**.

Переменная может ассоциироваться с параметрами различных блоков исходной подсистемы в том случае, если для них требуется одновременно задавать одинаковые значения;

- присвоить имена переменных параметрами блоков исходной подсистемы (рис. 8.29):
  - vector параметру Polynomial Coefficients блока Polynomial;
  - k параметру Gain блока Gain;
  - b параметру Constant value блока Constant;

<ul> <li>Function Block Par</li> <li>Polyval</li> <li>Polynomial evaluation. Cal is sorted highest order to I function.</li> <li>Parameters</li> <li>Polynomial Coefficients:</li> </ul>	ameters: Polynomial Iculates P(u) given by the polynomial o owest order, the form accepted by M/ Function Block Parameter Gain	oefficient array P. P ATLAB's polyval s: Gain
vector	Element-wise gain (y = K.*u) or ma	trix gain ( $y = K^{\infty}u$ or $y = u^{\infty}K$ ).
	Main Signal Attributes Par	Source Block Parameters: Constant
	Gain: k Multiplication: Element-wise(K. "u) Sample time (-1 for inherited): -1	Constant Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value. Main Signal Attributes
		Constant value:
		b
	<u>Ok</u>	Interpret vector parameters as 1-D Sampling mode: Sample based Sample time: Inf
		QK <u>Cancel H</u> elp

Рис. 8.29. Параметры исходной подсистемы Poly, подготовленной к маскированию

- выделить исходную подсистему, подготовленную к маскированию (рис. 8.30);
- выбрать имена параметров маскируемой подсистемы<sup>1</sup>, которые будут согласовываться с параметрами исходной подсистемы через выбранные переменные (табл. 8.4).

<sup>&</sup>lt;sup>1</sup> Во избежание сбоев рекомендуется выбирать англоязычные имена параметров.



Рис. 8.30. S-модель с выделенной, подготовленной к маскированию подсистемой Poly

<b>Габлица 8.4.</b> Выбранные имена параметров маскируемои подсистемы Ро	Габлица 8.4. Е	Зыбранные (	имена пара	аметров м	аскируемой	подсистемы	Poly
--	----------------	-------------	------------	-----------	------------	------------	------

Параметр маскируемой подсистемы	Переменная	Параметр блока исходной подсистемы
Coefficients	vector	Polynomial Coefficients
Gain Coefficient	k	Gain
Constant	b	Constant value

- 3. Выполнить команду меню Edit | Mask Subsystem (Редактирование | Маскировать подсистему) или одноименную команду контекстного меню, после чего открывается окно Mask Editor (Редактор маски) с пятью вкладками средствами создания маскируемой подсистемы (рис. 8.31):
  - Ісоп (Пиктограмма) средства создания пиктограммы;
  - **Parameters** (Параметры) средства создания окна параметров;
  - Initialization (Инициализация) средства инициализации;
  - Documentation (Документация) средства документации.

В нижней части окна имеются общие для всех вкладок кнопки:

• Unmask (Снять маску) — отменить маскирование с сохранением заданных установок и закрыть Mask Edito.

До закрытия файла с маскируемой подсистемой установки *сохраняются* и могут быть восстановлены по команде Edit | Mask Subsystem;

- **ОК** создать маскированную систему с заданными установками и закрыть окно **Mask Editor**;
- Cancel отменить заданные установки и закрыть окно Mask Editor;
- Help обратиться к справочной системе MATLAB по Simulink в формате HTML к разделу Simulink → User Guide → Simulink Graphical User Interface → Simulink Mask Editor (Simulink → Руководство пользователя → GUI Simulink → Редактор маски Simulink);
- Apply (Применить) применить заданные установки при открытом окне Mask Editor.
- 4. Открыть вкладку Parameters (см. рис. 8.31).

Средства создания окна параметров маскируемой подсистемы включают в себя две панели и четыре кнопки:

- **Dialog Parameters** (Параметры диалога) панель для установки *основных атрибутов* параметров, построчно для каждого в пяти столбцах:
  - **Prompt** (Подсказка) имя параметра;
  - Variable (Переменная) имя переменной;
  - Туре (Тип) способы задания значения параметра:
    - о edit в поле ввода;
    - о checkbox с помощью флага;
    - о popup раскрывающимся списком;
  - Evaluate (Вычислить) флаг признак принадлежности переменной к численному типу данных.

При установке флага **Evaluate** значения параметра будут восприниматься как численные (альтернатива — символьные) и задаваться в окне параметров маскируемой подсистемы *константой* или *вычисляемым выражением*, записанным по правилам языка MATLAB;

- Tunable (Настраиваемый) флаг признак возможности изменения значения переменной в процессе моделирования;
- Options for selected parameter (Опции для выбранного параметра) панель для установки *дополнительных атрибутов* выделенного параметра, включающую в себя:
  - группу In dialog (В диалоге) с двумя флагами:
    - Show parameter (Показать параметр) флаг, при установке которого (по умолчанию) параметр выводится в окне параметров маскируемой подсистемы;

- **Enable parameter** (Доступный параметр) флаг, при установке которого (по умолчанию) параметр активизируется;
- поле ввода Popups (one per line) (Значения в раскрывающемся списке (одно на линию)) поле для построчного ввода значений в раскрывающемся списке при значении параметра Туре: рорир;
- поле ввода Dialog callback (Вернуться к диалогу) поле для ввода команд и операторов MATLAB с целью контроля значений переменных в процессе моделирования;
- четыре кнопки для управления вводом параметров:
  - Add (Добавить) добавить строку;
  - **Delete** (Удалить) удалить строку;
  - Up (Вверх) переместить на строку выше;
  - **Down** (Вниз) переместить на строку ниже.

Masl	k Editor : Poly				
Icon F	Parameters Initialization	Documentation			
	Dialog parameters				
	Prompt	Variable	Туре	Evaluate Tunabl	e
1÷	Options for selected p	irameter			
	Popups (one per line):	Dialog callback:	rameter	Enable parameter	
Unma	sk		OK Can	cel Help Ap	oply

Рис. 8.31. Окно Mask Editor при открытой вкладке Parameters

5. Последовательно нажимая кнопку Add, построчно установить *основные* и *дополнительные* атрибуты каждого параметра на панели Dialog Parameters (см. табл. 8.4 и рис. 8.32).

В примере все параметры *численные* (флаг Evaluate установлен), но для их ввода предусмотрены разные средства интерфейса, а именно:

- для параметра Coefficients поле ввода;
- для параметра Gain Coefficients раскрывающийся список;
- для параметра Constant флаг.

Для параметра Gain Coefficients установлен дополнительный атрибут — раскрывающийся список в поле ввода **Popups (one per line)**.

Параметр Constant может принимать только два численных значения (по умолчанию типа double): Constant: 1 — при установке флага и Constant: 0 — при сбросе.

Поле Dialog callback оставлено пустым.

2 Mas	k Editor : Poly						
Icon	Parameters Initialization	Documenta	tion				
	Dialog parameters						
-	Prompt	Var	riable	Туре		Evaluate	Tunable
	Coefficients	vect	or	edit	~	<ul> <li>Image: A start of the start of</li></ul>	
X	Gain Coefficient	k		popup	~	<ul> <li>Image: A start of the start of</li></ul>	
	Constant	b		checkbox	~	<ul> <li>Image: A start of the start of</li></ul>	
8	Options for selected pa Popups (one per line):	rameter In dialog:	Show para	ameter [	▼ Ena	able parame	eter
Unma	4 8 Isk	callback:		OK Car	ncel	Help	Apply

Рис. 8.32. Параметры маскируемой подсистемы Poly с основными и дополнительными атрибутами

6. Нажать кнопку **Apply**.

С этого момента подсистема становится маскируемой — ее структура скрыта "под маской", и двойной щелчок левой кнопки мыши на пиктограмме открывает окно параметров, подобно блоку библиотеки Simulink.

Для того чтобы увидеть структуру маскируемой подсистемы — "снять маску", — следует выделить подсистему и выполнить команду меню Edit | Look under mask (Редактирование | Заглянуть под маску) или одноименную команду контекстного меню. Этой же командой необходимо воспользоваться, если пользователь решил изменить *имя переменной*, ассоциируемой с блоком исходной подсистемы; новое имя должно быть присвоено соответствующему параметру.

До закрытия окна Mask Editor можно *редактировать* основные и дополнительные атрибуты параметров с повторным нажатием кнопки Apply.

 Для проверки введенных параметров открыть окно параметров маскируемой подсистемы двойным щелчком левой кнопки мыши на пиктограмме блока Poly (рис. 8.33).

Параметры *маскируемой подсистемы* согласуются с параметрами *исходной подсистемы* через переменные, сохраняемые в области памяти **Mask Workspace**.

🗑 Function Block Parameters: Poly	<
Subsystem (mask)	٦
	_
Parameters	٦
Coefficients	
0	
Gain Coefficient 1	
Constant	
OK Cancel Help Apply	)

Рис. 8.33. Окно параметров маскируемой подсистемы (предварительное)

- 8. Закрыть окно параметров Function Block Parameters: Poly и продолжить создание окна параметров.
- 9. Открыть вкладку Documentation.

Средства документации включают в себя три панели<sup>1</sup> (рис. 8.34):

• Mask type (Тип маскируемой подсистемы) — панель с полем ввода типа блока, которое будет отображаться в верхней строке окна параметров маскируемой подсистемы;

Для унификации с окном параметров любого другого блока библиотеки Simulink рекомендуется вводить в поле **Mask type** *имя блока* (Poly). Тогда, при копировании маскируемого блока в окне S-моделей к его имени, подобно блоку библиотеки Simulink, будут последовательно добавляться цифры 1, 2 и т. д. (Poly1, Poly2 и т. д.).

 Mask description (Описание маскируемой подсистемы) — панель с полем ввода краткого описания назначения блока в окне параметров;

<sup>&</sup>lt;sup>1</sup> Во избежание сбоев рекомендуется вводить англоязычный текст.

• Mask help (Помощь по маскируемой подсистеме) — панель с полем ввода справки о блоке, которая будет выдаваться справочной системой MATLAB.

Рассмотренный вариант создания справки является простейшим. С другими средствами создания справки можно познакомиться самостоятельно с помощью справочной системы MATLAB по Simulink.

Ma	sk Editor :	Poly						
Icon	Parameters	Initialization	Documentatio	on				
Mas	sk type							
Po	ly							
Mas	k description							
B1	ock Poly	generate	s values	of polyno	omial wi	th gair	n and	adding
K	sk belo		101					>
Mas B1	ik help ock Poly	(mask su	ubsystem)	contains	a Polyn	nomial h	olock,	whose
Mas Blo	sk help ock Poly	(mask su	ubsystem)	contains	a Polyr	nomial b	)lock,	whose

Рис. 8.34. Окно Mask Editor при открытой вкладке Documentation с введенной информацией

10. Нажать кнопку **Apply** — создание окна параметров маскируемой подсистемы завершено (рис. 8.35).

Function Block Parameters: Poly	X
Poly (mask)	
Block Poly generates values of polynomial with gain and adding constant 1 or 0.	
Parameters	=
Coefficients	
0	
Gain Coefficient 1	-
Constant	
OK <u>C</u> ancel <u>H</u> elp Apply	

Рис. 8.35. Окно параметров маскируемой подсистемы Poly

Справка о маскируемой системе, введенная в поле **Mask help**, будет выдаваться при выделении блока в окне S-моделей маскируемой подсистемы (Poly) и выполнении команды **Help** | **Blocks** (рис. 8.36).



Рис. 8.36. Справка о маскируемой системе Poly

#### 11. Открыть вкладку Initialization.

Средства инициализации включают в себя две панели (рис. 8.37):

- **Dialog variables** (Переменные диалога) панель со списком переменных маскируемой подсистемы;
- Initialization commands (Команды инициализации) панель с окном ввода команд и операторов MATLAB, в том числе script- и function-файлов, которые при запуске моделирования позволяют устанавливать дополнительные значения параметров маскируемой системы (через ассоциируемые с ними переменные) или обеспечивать изменение вида пиктограммы в зависимости от значений параметров.

Окно Initialization commands является аналогом командного окна MATLAB, однако все используемые в нем переменные размещаются в области памяти **Mask Workspace**. Результаты выполнения команд и операторов автоматически выводятся в окне Command Window MATLAB; для нежелательного вывода следует использовать точку с запятой.

В создаваемой маскируемой подсистеме в окне Initialization commands введены команды (см. рис. 8.37), позволяющие при запуске моделирования установить дополнительные значения параметрам Gain Coefficient (переменной k) и Constant (переменной b) в том случае, если длина x параметра Coefficients (переменной vector) превосходит 10.

Mask Editor : Poly		
Icon Parameters Initializ	ation Documentation	
Dialog variables	Initialization commands x=length (vector); if x>10 k=0;b=5;end Allow library block to modify its contents	
Unmask	OK Cancel Help	Apply

Рис. 8.37. Окно Mask Editor при открытой вкладке Initialization с введенной информацией

- 12. Нажать кнопку **Apply**.
- 13. Открыть вкладку Icon (Пиктограмма).

Средства создания пиктограммы включают в себя три панели (рис. 8.38):

- Icon options (Опции для пиктограммы) панель с четырьмя раскрывающимися списками:
  - Frame (Рамка) видимая (Visible) или невидимая (Invisible) рамка пиктограммы;
  - **Transparency** (Прозрачность) прозрачная (Transparent) или непрозрачная (Opaque) пиктограмма;
  - Rotation (Вращение) вращающееся (Rotates) или фиксированное (Fixed) содержимое внутри пиктограммы;

- Units (Единицы) варианты способа масштабирования осей координат (первого квадранта) при размещении рисунка внутри пиктограммы:
  - Autoscale (Автоматическое масштабирование) координаты осей масштабируются автоматически так, чтобы рисунок имел максимальный размер внутри рамки и пропорционально изменялся с изменением ее размера;
  - Pixels (Пикселы) координаты осей указываются в пикселах и при изменении размера рамки не меняются;
  - Normalized (Нормализованные) координаты осей нормализуются относительно левого нижнего угла рамки с координатами [0; 0] и правого верхнего с координатами [1; 1];
- Drawing commands (Команды рисования) панель с полем ввода команд и операторов MATLAB для создания содержимого внутри пиктограммы;
- Examples of drawing commands (Примеры команд рисования) панель с раскрывающимся списком типовых команд для создания содержимого внутри пиктограммы — рисунка и/или текста.

МАТLAВ предлагает разнообразные средства для создания содержимого пиктограммы, с которыми предлагаем познакомиться самостоятельно. В дальнейшем ограничимся вводом *текста* с помощью оператора disp следующего формата:

disp('<текст>') -	— для вывода	текста в	апострофах	(рис.	8.38).
-------------------	--------------	----------	------------	-------	--------

con Parameters Initi	alization Documentation				
Icon options	Drawing commands				
Frame	disp('MaskSubsystem')				
Visible 💌					
Transparency					
Transparent 💌					
Rotation					
Rotates 💌					
Units					
Autoscale 💌					
Examples of drawing c Command port_lal Syntax port_label("o	ommands bel (label specific ports)				

Рис. 8.38. Окно Mask Editor при открытой вкладке Icon с введенной информацией

14. Нажать кнопку ОК — маскируемая подсистема создана (рис. 8.39).

Маскируемая подсистема по определению является *виртуальной*, т. к. в процессе моделирования осуществляется ее связь с блоками библиотеки Simulink.

Mask_poly_s	ubs *	
Eile Edit View S	imulation Format Tools Help	
0 🗳 🖬 🚭	※ 10 10 ( 中今 合 ) 12 (	≥   ► = 10.0 Normal ▼
(1 2 3 4 Constant	5) Int MaskSubsystem Out1 Poly	Display
Ready	100%	FixedStepDiscrete

Рис. 8.39. S-модель системы с маскируемой подсистемой Poly

15. Задать параметры маскируемой подсистемы Poly, запустить моделирование и проконтролировать результат (рис. 8.40).

Окно параметров открывается двойным щелчком левой кнопки мыши на пиктограмме блока Poly, как для стандартного блока библиотеки Simulink.

В примере указано значение параметра **Coefficients:**  $[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$  (vector= $[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$ ). Длина вектора превосходит 10, поэтому после запуска моделирования, с учетом команд, введенных при открытой вкладке **Initialization** на панели **Initialization commands** (см. рис. 8.37), двум другим параметрам будут присвоены значения **Coefficients Gain:** 0 (k=0) и **Constant:** 5 (b=5), после чего выведен соответствующий результат (b=5) (см. рис. 8.40).

🖬 mas	sk_poly_subs *	
Ele Ed	it <u>V</u> iew Simulation Format <u>T</u> ools <u>H</u> elp	
	출 🖬 🚭   ※ 🖻 🖻   순 수 个   으 오   🕨 = 10.0 Normal	•
	Image: Second state st	×
ready	Block Poly generates values of polynomial with gain and adding constant 1 or 0. Parameters Coefficients [11111111111] Gain Coefficient[4]	
	✓ Constant	

Рис. 8.40. Параметры маскируемой подсистемы Poly и результат моделирования

После выполненных установок и закрытия окна Mask Editor созданная маска может быть изменена по команде главного меню Edit | Edit Mask (Редактирование | Редактирование маски) или одноименной команде контекстного меню.



# Глава 9

# Собственная библиотека

Собственная библиотека — это создаваемая пользователем библиотека блоков с собственным именем, структурно подобная библиотеке блоков в библиотеке Simulink, но существующая независимо от нее.

Создание собственной библиотеки позволяет решить следующие основные задачи:

• оперативное обращение к часто используемым блокам;

• компактное хранение созданных маскируемых подсистем.

### 9.1. Создание собственной библиотеки

Для создания собственной библиотеки необходимо выполнить следующие действия:

1. Открыть окно библиотек.

По команде меню File | New | Library (Файл | Новый | Библиотека) окна Simulink Library Browser (см. разд. 1.4) открывается окно библиотек Library:untitled (Библиотека:безымянная) для создания собственной библиотеки (рис. 9.1).



Рис. 9.1. Окно библиотек Library:untitled

Интерфейс окна библиотек подобен интерфейсу окна S-моделей, но с меньшим набором команд меню и наличием в строке состояния (нижняя строка) индикатора открытости библиотеки: Unlocked (Разблокирована) или Locked (Заблокирована) (рис. 9.1).

Состояние Unlocked означает, что библиотека доступна для пополнения и редактирования, а Locked — что она доступна только для чтения (перетаскивания блоков из нее), и для выполнения иных действий необходимо предварительно разблокировать библиотеку по команде меню Edit | Unlock Library (Редактирование | Разблокировать библиотеку).

Для вновь создаваемой библиотеки индикатор открытости по умолчанию находится в состоянии Unlocked.

- Перетащить блоки из библиотеки Simulink (и/или из другой библиотеки блоков) и блоки маскированных подсистем пользователя из соответствующих окон Sмоделей в окно Library:untitled с помощью технологии "drag-and-drop" ("перетащить и отпустить").
- 3. Установить параметры блоков (необязательный этап).

После перетаскивания блоков в собственную библиотеку их связь с библиотекой Simulink и исходными маскируемыми подсистемами *разрывается*. Поэтому в собственной библиотеке при желании можно устанавливать другие значения параметров блоков, что никак не повлияет на их значения в библиотеке Simulink и в исходной маскируемой подсистеме.

4. Сохранить собственную библиотеку на диске.

Сохранение содержимого окна библиотек выполняется по команде меню File | Save (Файл | Сохранить). В открывающемся окне Save as (Сохранить как) файлу собственной библиотеки блоков, подобно файлу S-модели, присваивается имя с расширением mdl.

*Имя* файла может состоять из любой последовательности латинских букв, цифр и символа подчеркивания, начинающейся с буквы.

На рис. 9.2 представлено окно библиотек с созданной библиотекой my\_library, которая включает в себя пять блоков из библиотеки Simulink и один блок созданной ранее маскированной подсистемы Poly (см. рис. 8.39).

Файл собственной библиотеки с расширением mdl может храниться в папке work или в папке пользователя, вложенной в work.

При следующих открытиях созданной собственной библиотеки индикатор открытости библиотеки по умолчанию будет находиться в состоянии **Locked**.

Library:my_library					
<u>File Edit View Format Help</u>	8				
0 🗳 🖬 🚭 🕺 🖻 1	8   ↔ ↔ ↔	$  \Omega  $	2 🛼 🖾 🗖	5	
5 Constant Random Number	Sine Wave				
Display	Scope	>In1	MaskSystem Poly	Out1 >	
Ready				100%	Unlocked

Рис. 9.2. Собственная библиотека my\_library

### 9.2. Создание групп блоков

Создание групп блоков внутри собственной библиотеки удобно в том случае, когда блоки можно сгруппировать по какому-либо признаку.

Для создания групп блоков необходимо выполнить следующие действия:

1. Открыть файл с собственной библиотекой.

Файл с собственной библиотекой открывается, как обычно, двойным щелчком левой кнопки мыши на его имени или пиктограмме в окне Current Folder либо перетаскиванием пиктограммы файла из окна Current Folder в окно Command Window.

Давайте откроем созданную в предыдущем разделе библиотеку my\_library (см. рис. 9.2).

- 2. Разблокировать библиотеку по команде меню Edit | Unlock Library.
- 3. Выделить блоки, объединяемые в первую группу.

В первую группу объединим источники сигналов (рис. 9.3).

4. Создать неуправляемую виртуальную подсистему по команде Edit | Create Subsystem (см. разд. 8.2.1).

После выполнения команды выделенные блоки автоматически заменятся блоком Subsystem с входными (In) и выходными (Out) портами, количество которых будет равно соответственно суммарному числу входов и выходов в объединяемых блоках.



Рис. 9.3. Окно библиотеки my library с блоками, выделенными для объединения в группу

В формируемой группе источники сигналов не имеют входов, поэтому блок Subsystem содержит только три выходных порта (рис. 9.4).

🙀 Library:my_library *					
<u>File Edit View Format Help</u>	1				
0 😂 🖬 🚳   % 🖻 I	1 4 4 1	- <u>Ω</u> Ω	2 🛼 🔯 🗖	5	
Out1 > Out2 > Out3 > Subsystem					
Display	Scope	>In1	MaskSystem Poly	Out1>	
Ready				100%	Unlocked

Рис. 9.4. Окно библиотеки my\_library с подсистемой Subsystem

 Открыть окно подсистем двойным щелчком левой кнопки мыши на пиктограмме блока Subsystem и удалить в нем автоматически созданные входные и выходные порты и линии связи.

На рис. 9.5 приведены блоки группы с удаленными выходными портами и линиями связи.



Рис. 9.5. Окно подсистем блока Subsystem с удаленными портами и линиями связи

6. Закрыть окно подсистем.

После этого пиктограмма блока Subsystem изменится (рис. 9.6).



Рис. 9.6. Окно библиотеки my\_library с подсистемой Subsystem после закрытия окна подсистем

7. Выделить блок Subsystem и выполнить *маскирование* подсистемы по команде меню Edit | Mask Subsystem (см. разд. 8.5).

После выполнения команды откроется окно Mask Editor.

8. В окне Mask Editor открыть вкладку Icon и на панели Drawing commands ввести текст внутри пиктограммы блока Subsystem по команде MATLAB (см. разд. 8.5):

```
disp('<rekct>')
```

Введем текст:

```
disp('My Sources')
```

- 9. Нажать кнопки Apply и OK в окне Mask Editor маскируемая подсистема создана.
- 10. Скрыть имя блока Subsystem по команде меню Format | Hide Name (см. табл. 2.2).
- 11. Растянуть блок до удобного размера, добавить тень по команде меню Format | Show Drop Shadow (см. табл. 2.1) и выделить блок цветом по команде Format | Foreground Color (см. табл. 2.3) (пользователь может выбрать и другие команды по оформлению блока).

На рис. 9.7 представлена группа блоков му sources с тенью и выбранным синим цветом (Blue).



Рис. 9.7. Окно библиотеки my library с группой блоков My Sources

Открывается группа блоков двойным щелчком левой кнопки мыши на ее пиктограмме.

12. Аналогичным образом создать другие группы блоков.

Действия при объединении в группу блоков маскируемых подсистем пользователя ничем не отличаются от рассмотренного объединения в группу блоков библиотеки Simulink. На рис. 9.8 представлена собственная библиотека my\_library с тремя группами блоков.



Рис. 9.8. Окно библиотеки my\_library с тремя группами блоков

После создания всех групп блоков собственная библиотека сохраняется на диске (см. разд. 9.1).

Перетаскивание блоков из собственной библиотеки в окно S-моделей выполняется, как обычно, с помощью технологии drag-and-drop.

Для *пополнения* группы блоков следует открыть и разблокировать собственную библиотеку (см. разд. 9.1), открыть группу блоков, в открываемое окно перетащить требуемые блоки и закрыть окно.

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  User Guide  $\rightarrow$  Working with Block Libraries  $\rightarrow$  Creating Block Libraries (Simulink  $\rightarrow$  Руководство пользователя  $\rightarrow$  Работа с библиотеками блоков  $\rightarrow$  Создание библиотек блоков).



# Глава 10

## Отладка S-модели системы

Отладка S-модели системы обеспечивает контроль над процессом моделирования с целью поиска ошибок в ее работе.

В Simulink предусмотрены две разновидности средств отладки:

- □ программные средства Debugger's Command-line Interface (Интерфейс командной строки отладчика) набор команд отладки, вводимых в командной строке окна Command Window;
- средства GUI Debugger's GUI (GUI отладчика) программа GUI, разработанная на основе программных средств Debugger's Command-line Interface и предназначенная для отладки путем интерактивного общения без прямого доступа к командам отладки.

В этой главе рассматривается отладка S-модели системы более простыми и удобными средствами Debugger's GUI.

### 10.1. Метод блока и цикл моделирования

S-модель системы, созданная средствами блочного моделирования Simulink, в неявном виде реализуется программными средствами Simulink, разработанными на основе языка MATLAB.

Выполнение S-модели системы на каждом шаге моделирования реализуется путем выполнения стандартных функций Simulink. С этим связана принятая в Debugger's GUI терминология:

□ *метод* (method) — стандартная функция Simulink;

□ *метод блока* (block method) — метод, реализующий выполнение *невиртуального* блока библиотеки Simulink на текущем шаге моделирования.

Имя метода блока совпадает с именем блока.

Выполнение *виртуального* блока библиотеки Simulink *(см. разд. 4.3)* реализуется с помощью *группы* методов блоков, имена которых совпадают с именами блоков, входящих в состав виртуального блока.

Подсистемы, невиртуальные и виртуальные, в том числе маскируемые (см. гл. 8), реализуются с помощью группы методов блоков, имена которых совпадают с именами блоков, входящих в состав подсистемы; □ *цикл моделирования* (simulation loop) — методы блоков, реализующие выполнение S-модели системы на текущем шаге моделирования.

Выполнение S-модели системы реализуется пошаговым повторением<sup>1</sup> циклов моделирования. В процессе отладки обеспечивается возможность "проникновения внутрь" данного процесса — контроля над выполнением методов блоков — и предоставляются необходимые для этого средства.

Цикл моделирования имеет иерархическую структуру, представленную в виде дерева (см. левую панель на рис. 10.2).

Верхний уровень дерева соответствует самому циклу моделирования с именем конкретной S-модели системы.

*Промежуточные уровни* соответствуют вложенным *группам* методов блоков внутри цикла моделирования, каждая со своим именем.

В процессе отладки ветви верхнего и промежуточного уровней будут подсвечиваться желтым цветом.

Нижний уровень соответствует методу блока, имя которого совпадает с именем блока.

В процессе отладки ветви нижних уровней будут иметь вид синих гипертекстовых ссылок.

Выполняемая ветвь дерева отмечается курсором, при этом выполнению ветви нижнего уровня соответствует выполнение метода блока или коротко — выполнение блока<sup>2</sup>.

### 10.2. Запуск Debugger's GUI и окно отладчика

Debugger's GUI запускается из окна S-моделей, в котором открыт файл с отлаживаемой S-моделью системы, по команде меню Tools | Simulink Debugger (Средства | Отладчик Simulink). При этом откроется окно отладчика Simulink Debugger.

На рис. 10.1 представлено окно отладчика Simulink Debugger: discrete для созданной ранее S-модели дискретной системы discrete.mdl (см. рис. 3.6).

Интерфейс окна отладчика Simulink Debugger предоставляет разнообразные варианты пошагового и автоматизированного процесса отладки с регистрацией текущих результатов, которые управляются с помощью кнопок на панели инструментов (см. рис. 10.1):

□ Step into current method (Шаг в текущий метод) — шаг к следующей ветви дерева с автоматическим разворачиванием всех узлов и выполнением всех методов блока, независимо от наличия/отсутствия точек останова.

*Точка останова* (Break point) — это точка, в которой прерывается процесс моделирования для контроля над выполнением блока.

<sup>&</sup>lt;sup>1</sup> Здесь и далее под "шагом" понимается шаг моделирования.

<sup>&</sup>lt;sup>2</sup> По умолчанию невиртуального.

При последовательных нажатиях кнопки процесс отладки сопровождается следующими действиями в окне S-моделей и подсистем:

- выводом *имени* ветви дерева в левом верхнем углу в *красном* прямоугольнике для верхнего и промежуточного уровней и в *черном пунктирном* для нижнего уровня *(метода блока)* с индексом блока в виде s:b, где s номер подсистемы (subsystem), а b номер блока (block) в подсистеме (см. табл. 2.4);
- появлением стрелки, указывающей на выполняемый блок и его выделение цветом.

По завершении моделирования окна S-моделей и подсистем возвращаются в исходное состояние.

Процесс отладки с помощью других кнопок сопровождается аналогичными действиями.

- Step over current method (Шаг через текущий метод) шаг к следующей ветви дерева при свернутых узлах с прерыванием в точках останова, автоматическим разворачиванием узла, который содержит метод блока в точке останова, и выполнением данного и следующих за ним методов блока в соответствующей группе методов блоков (см. разд. 10.1);
- Step out of current method (Шаг из текущего метода) шаг к предыдущей ветви дерева с более высоким (относительно текущей) приоритетом при свернутых узлах с прерыванием в точках останова, автоматическим разворачиванием узла, который содержит метод блока в точке останова, и выполнением данного метода блока;
- Go to first method at start of next time step (Перейти к первому методу в начале следующего шага) — переход к циклически повторяемой ветви дерева с прерыванием в точках останова, автоматическим разворачиванием узла, который содержит метод блока в точке останова, и выполнением данного метода блока в цикле;
- □ Go to the next block method (Перейти к следующему методу блока) действия подобны предыдущей кнопке, но с выполнением всех методов блока в цикле, независимо от наличия/отсутствия точек останова;
- □ Start/Continue (Запуск/продолжение) запуск/продолжение моделирования.

Первое нажатие кнопки запускает моделирование и активизирует другие кнопки на панели инструментов;

- □ Pause (Пауза) пауза в процессе моделирования;
- □ Stop debugging (Остановка отладки) остановка моделирования с блокированием всех кнопок на панели инструментов, кроме кнопок Start/Continue, Break before selected block, Display I/O of selected block when executed и Help;
- □ Break before selected block (Прервать перед выделенным блоком) установка *точки останова* перед выделенным блоком.

Для установки точки останова следует выделить блок в окне S-моделей или подсистем и нажать данную кнопку.

Точки останова не могут устанавливаться перед *виртуальными* блоками и подсистемами, но могут ставиться перед *невиртуальными* блоками виртуальной подсистемы;

□ Display I/O of selected block when executed (Показать вход/выход выделенного блока при выполнении) — установка *точки наблюдения* перед выделенным блоком.

*Точка наблюдения* (Display point) — это точка, в которой процесс моделирования *не прерывается*, но могут быть оперативно выведены текущие результаты выполнения любого блока (метода блока) без их сохранения.

Для установки точки наблюдения следует выделить блок в окне S-моделей или подсистем и нажать данную кнопку.

Текущие результаты выполнения метода блока выводятся на вкладке **Outputs**, о которой идет речь далее, и включают в себя (см. рис. 10.2):

- текущее время моделирования тм;
- индекс блока (Block ID) в виде s:b, где s номер подсистемы (subsystem), а
   b номер блока (block) в подсистеме (см. табл. 2.4);
- имя метода блока;
- значения входных (u) и выходных (y) сигналов;
- переменные состояния: СSTATE для аналоговых и DSTATE для дискретных систем.

Точки наблюдения не могут устанавливаться перед *виртуальными* блоками и подсистемами, но могут ставиться перед *невиртуальными* блоками виртуальной подсистемы;

□ Display current I/O of selected block (Показать текущий вход/выход выделенного блока) — вывод текущих результатов выполнения выделенного блока, таких же, как в точках наблюдения (см. кнопку Display I/O of selected block when executed).

В процессе отладки данные результаты могут быть оперативно выведены для любого выделенного в окне S-моделей или подсистем блока нажатием кнопки **Display current I/O of selected block**, после чего продолжен процесс отладки.

Доступен для виртуальных и невиртуальных блоков и подсистем;

Enable/Disable Animation (Доступна/Не доступна анимация) — переход к автоматическому выполнению ветвей дерева с разворачиванием узлов и анимационным сопровождением в окнах S-моделей и подсистем (подобного описанному для кнопки Step into current method) с прерываниями в циклически повторяемых ветвях и в точках останова.

После запуска моделирования (нажатием кнопки Start/Continue) следует последовательно нажать кнопки Enable/Disable Animation и Start/Continue.

Для продолжения процесса отладки после прерывания нажимается кнопка Start/Continue.

В процессе отладки все кнопки панели инструментов, кроме Pause и Stop debugging, блокируются и активизируются только в моменты автоматического прерывания или при нажатии кнопки Pause;

- Animation delay (Задержка анимации) бегунок для регулирования задержки между выполняемыми ветвями дерева при нажатой кнопке Enable/Disable Animation;
- □ Help (Помощь) автоматический выход в справочную систему MATLAB по Simulink в формате HTML в раздел Simulink  $\rightarrow$  User Guide  $\rightarrow$  Simulink Debugger.

Кроме панели инструментов, интерфейс окна отладчика Simulink Debugger включает в себя следующие компоненты:

□ левая панель с двумя вкладками (рис. 10.1):

- Break Points (Точки останова) для отображения точек останова (Break points), точек наблюдения (Display points) и точек останова по условию (Break on condition);
- Simulation Loop (Цикл моделирования) для отображения цикла моделирования, представленного в виде дерева (рис. 10.2);

Вкладка Break Points содержит:

- группу **Break/Display points** (Прервать/Показать точки) для отображения точек останова и точек наблюдения. Содержит *три столбца*:
  - Blocks (Блоки) список установленных точек останова и точек наблюдения;
  - Break before selected block флаг, активизируемый при установке точки останова, состояние которого позволяет оперативно игнорировать/восстанавливать точки останова в процессе отладки без их удаления из списка Blocks;
  - Display I/O of selected block when executed флаг, активизируемый при установке точки наблюдения, состояние которого позволяет оперативно игнорировать/восстанавливать точки наблюдения в процессе отладки без их удаления из списка **Blocks**.

На рис. 10.1 видно, что для S-модели дискретной системы discrete мы установили одну точку останова и одну точку наблюдения, которые отмечены галочками;

- кнопку Remove selected point (Удалить выделенную точку) для удаления точки останова или наблюдения из списка Blocks;
- группу **Break on conditions** (Прервать по условиям) с флагами для установки условий в *точках останова по условию*:
  - Zero crossings (Пересечения нуля) пересечение сигналом нулевого уровня;
  - Step size limited by state (Размер шага, ограничиваемый состоянием) состояние системы (как правило, аналоговой), требующее для решателя с

переменным шагом ограничения его размера при превышении допустимого значения;

- Solver Error (Ошибка решателя) исправимая ошибка решателя; при сбросе флага решатель автоматически исправляет ошибку и процесс моделирования продолжается без сообщения об ошибке;
- NaN values (Ошибка решателя) появление не числа Nan (Not-a-number), которое присваивается неопределенности;
- Break at time (Ошибка решателя) достижение времени, заданного в поле ввода.

Вкладка Simulation Loop содержит три столбца:

- Method (Метод) для отображения цикла моделирования в процессе отладки;
- Break points флаг оперативной установки точки останова перед выполняемым блоком (методом блока);
- ID (Method ID индекс метода) список номеров (индексов) выполняемых ветвей дерева, образующего цикл моделирования, начинающийся с нуля;



Рис. 10.1. Окна S-моделей с дискретной системой discrete и отладчика Simulink Debugger: discrete при открытых вкладках Break Points и Outputs до запуска моделирования



Рис. 10.2. Окна S-моделей с дискретной системой discrete и отладчика Simulink Debugger: discrete при открытых вкладках Simulation Loop и Outputs в процессе отладки с помощью кнопки Step into current method

правая панель с тремя вкладками:

• **Outputs** (Выходы) — для отображения текущих результатов моделирования в процессе отладки.

При открытии окна отладчика **Simulink Debugger** на вкладке **Outputs** размещается приглашение к работе, подсказка по запуску моделирования и другие сведения справочного характера (см. рис. 10.1).

После запуска моделирования на вкладке **Outputs** будут автоматически выводиться и сохраняться текущие результаты моделирования, структурно такие же, как в точках наблюдения (см. кнопку **Display I/O of selected block** when executed).

При нажатии кнопки Enable/Disable Animation текущие результаты моделирования на вкладке Output выводятся только в моменты прерывания;

- Sorted List (Упорядоченный список) список индексов и методов блоков в порядке их выполнения в цикле моделирования (рис. 10.3);
- Status (Состояния) для отображения текущего состояния отладчика (рис. 10.4);

□ кнопка Close на панели инструментов — для закрытия окна отладчика.



Рис. 10.3. Окно отладчика Simulink Debugger: discrete при открытых вкладках Simulation Loop и Sorted List в процессе отладки с помощью кнопки Step into current method

📣 Simulink Debugger: discrete	
우 : 마 ● 비 ● 이 31 다 다 우리 두권	2 ※ 0
Break Points Simulation Loop	Outputs Sorted List Status
Method O ID O discrete. Initialize 10 O construction 11 O construction 11 O construction 14 O discrete. Simulation 14 O discrete. Output 15 O construction 16 Step. Out 17 Discrete 18 Scope. Out 19 O construction	Current simulation time : 0.2 (MajorTimeStep) Default command to execute on return/enter : "step over" Break at zero crossing events : disabled Break on solver error : disabled Break on failed integration step : disabled Break on non-finite (NaN,Inf) values : disabled Break on solver reset request : disabled Break on solver reset request : disabled Display level for disp, trace, probe : 1 (i/o, states) Solver trace level : 0 Algebraic loop tracing level : 0 Animation Mode : off Window reuse : not supported Execution Mode : Normal

Рис. 10.4. Окно отладчика Simulink Debugger: discrete при открытых вкладках Simulation Loop и Status в процессе отладки с помощью кнопки Step into current method Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  User Guide  $\rightarrow$  Simulink Debugger  $\rightarrow$  Using the Debugger's Graphical User Interface (Simulink  $\rightarrow$  Руководство пользователя  $\rightarrow$  Отладчик Simulink  $\rightarrow$  Использование Debugger's GUI).



# Часть II

# Моделирование цифровой обработки сигналов средствами Simulink



# Глава 11

### Дискретные сигналы

Под дискретным сигналом понимается последовательность чисел бесконечной разрядности — последовательность x(nT), где nT — дискретное время и T — период дискретизации, а под цифровым сигналом — последовательность чисел конечной разрядности — квантованная последовательность  $\tilde{x}(nT)$ .

Термины "дискретный сигнал" и "последовательность" в теории ЦОС тождественны.

Случайные (стохастические) дискретные сигналы отождествляются с последовательностями случайных чисел и называются случайными (или стохастическими) последовательностями.

Понятно, что последовательность чисел бесконечной разрядности представить в MATLAB (и Simulink) невозможно, поэтому под *дискретным сигналом* будем условно понимать последовательность чисел типа double<sup>1</sup>, а под *цифровым* — квантованную последовательность чисел с фиксированной точкой<sup>2</sup> (ФТ).

В настоящей главе рассматриваются дискретные сигналы.

# 11.1. Дискретные сигналы: библиотека блоков Simulink

В этом разделе рассматриваются S-модели дискретных сигналов из группы блоков Sources библиотеки блоков Simulink (см. табл. 4.1).

Для моделирования *дискретного* сигнала следует установить параметр Sample time, так, чтобы последний трактовался как *период дискретизации T* (см. табл. 4.2).

Дискретный сигнал на выходе блока из *группы блоков Sources* может быть представлен в одном из следующих *шести* видов:

односигнальная последовательность — Sample-based Signal;

□ многосигнальная последовательность — Sample-based Multi Signal;

□ одноканальная последовательность — Sample-based Channel Signal;

□ многоканальная последовательность — Sample-based Multichannel Signal;

<sup>&</sup>lt;sup>1</sup> Double-precision floating point.

<sup>&</sup>lt;sup>2</sup> Fixed-point.

одноканальная последовательность фреймов — Frame-based Channel Signal. Фреймом (frame) называют группу (блок, пакет) отсчетов одинаковой длины;

□ многоканальная последовательность фреймов — Frame-based Multichannel Signal.

Вид последовательности определяет специфику ее обработки и зависит от двух факторов:

🗅 размерности численных параметров блока (скаляр, вектор, матрица);

С состояния флага Interpret vector parameters as 1-D (см. табл. 4.3).

#### 11.1.1. Односигнальная последовательность

Односигнальная последовательность (Sample-based Signal) представляется в виде *матрицы L*×1 (2-D) (вектора-столбца), где *L* — длина последовательности.

Односигнальная последовательность формируется в том случае, если численные параметры блока заданы *скалярами* и флаг Interpret vector parameters as 1-D *установлен* (см. табл. 4.3).

В процессе моделирования односигнальная последовательность — матрица  $L \times 1$  — обрабатывается по элементам столбца — на каждом шаге один элемент (один отсчет).

#### Пример 11.1

Сформируем односигнальную последовательность на выходе блока Constant (см. разд. 4.2.4) с параметрами: Constant value: 1; Sample time: 1 (рис. 11.1, Sample-based Signal).

Размерность экспортируемой в Workspace последовательности Constant\_1 и ее значения приведены в табл. 11.1 (первый столбец). Односигнальная последовательность представлена в виде *матрицы*  $L \times 1 = 5 \times 1$  (2-D) (вектора-столбца), где L = 5 — длина последовательности. При времени моделирования 4 (с) и периоде дискретизации 1 (с) одноканальная последовательность генерируется в моменты времени 0:4 (с) — один отсчет на каждом шаге<sup>1</sup>.

Односигнальная (пример 11.1)	Многосигнальная (пример 11.3)						
>> size(Constant 1)	>> size(Constant N)						
ans =	ans =						
5 1	5 6						
>> Constant 1	>> Constant N						
Constant 1 =	Constant N =						
1	1 2 3 4 5 7						
1	1 2 3 4 5 7						
1	1 2 3 4 5 7						
1	1 2 3 4 5 7						
1	1 2 3 4 5 7						

Таблица 11.1. Одно- и многосигнальная последовательности на выходе блока Constant

<sup>&</sup>lt;sup>1</sup> В блоке Display выводятся значения на последнем шаге моделирования.


Рис. 11.1. Формирование последовательности на выходе блока Constant: односигнальной (Sample-based Signal) и многосигнальной (Sample-based Multi Signal)

### Пример 11.2

Сформируем *односигнальную* последовательность на выходе блока Sine Wave *(см. разд. 4.2.19)* с параметрами: Sine type: Sample based; Amplitude: 1; Samples per period: 8; Sample time: 1 (рис. 11.2).

Pasmephoctb экспортируемой в Workspace последовательности Y\_sample\_1:
>> size(Y\_sample\_1)
ans =

#### 51 1

Односигнальная последовательность представлена в виде матрицы  $L \times 1 = 51 \times 1$  (2-D) (вектора-столбца), L = 51 — длина последовательности. При времени моделирования 50 (с) и периоде дискретизации 1 (с) последовательность генерируется в моменты времени 0:50 (с) — один отсчет на каждом шаге.



Рис. 11.2. Формирование односигнальной последовательности (Sample-based Signal) на выходе блока Sine Wave

## 11.1.2. Многосигнальная последовательность

Многосигнальная последовательность (Sample-based Multi Signal) включает в себя N последовательностей и представляется в виде *матрицы*  $L \times N$  (2-D), где N — количество последовательностей, а L — количество отсчетов (длина) каждой из них.

Многосигнальная последовательность формируется в том случае, если численные параметры блока заданы *векторами длины* N и флаг **Interpret vector parameters as 1-D** *установлен* (см. табл. 4.3).

В процессе моделирования многосигнальная последовательность — матрица  $L \times N$  — обрабатывается по строкам — на каждом шаге вектор  $1 \times N$  (N отсчетов).

#### Пример 11.3

Сформируем многосигнальную последовательность на выходе блока Constant *(см. разд. 4.2.4)* с параметрами<sup>1</sup>: **Constant value:** [1 2 3 4 5 7]; **Sample time:** 1 (см. рис. 11.1, Sample-based Multi Signal).

Размерность экспортируемой в Workspace последовательности Constant\_N и ее значения приведены в табл. 11.1 (второй столбец). Многосигнальная последовательность представлена в виде *матрицы*  $L \times N = 5 \times 6$  (2-D), где N = 6 — количество последовательностей, а L = 5 — длина каждой из них. При времени моделирования 4 (c) и периоде дискретизации 1 (c) многосигнальная последовательность генерируется в моменты времени 0:4 (c) — вектор  $1 \times N = 1 \times 6$  на каждом шаге.

#### Пример 11.4

Сформируем многосигнальную последовательность на выходе блока Sine Wave (см. разд. 4.2.19) с параметрами: Sine type: Sample based; Amplitude: [1 2 3]; Samples per period: [8 16 32]; Sample time: 1 (рис. 11.3).

Pasmephoctb экспортируемой в Workspace последовательности Y\_sample\_N: >> size(Y\_sample\_N) ans =

51 3

Многосигнальная последовательность представлена в виде *матрицы*  $L \times N = 51 \times 3$  (2-D), где N = 3 — количество последовательностей, а L = 51 — длина каждой из них. При времени моделирования 50 (c) и периоде дискретизации 1 (c) последовательность генерируется в моменты времени 0:50 (c) — вектор  $1 \times N = 1 \times 3$  на каждом шаге.

<sup>&</sup>lt;sup>1</sup> Размерность выходной последовательности выведена по команде меню Format | Port/Signal Display | Signal Dimensions (см. табл. 2.4).



Рис. 11.3. Формирование многосигнальной последовательности (Sample-based Multi Signal) на выходе блока Sine Wave

## 11.1.3. Одноканальная последовательность

Одноканальная последовательность (Sample-based Channel Signal) представляется в виде *трехмерного массива*  $1 \times 1 \times L$  (3-D) (L матриц  $1 \times 1$ ), где  $1 \times 1$  — количество каналов (один), а L — количество отсчетов (длина) последовательности в канале.

Одноканальная последовательность формируется в том случае, если численные параметры блока заданы *скалярами* и флаг Interpret vector parameters as 1-D *сброшен* (см. табл. 4.3).

В процессе моделирования одноканальная последовательность — трехмерный массив  $1 \times 1 \times L$  — обрабатывается *по матрицам* — на каждом шаге один отсчет в единственном канале (матрице  $1 \times 1$ ).

#### Пример 11.5

Сформируем одноканальную последовательность на выходе блока Constant *(см. разо. 4.2.4)* с параметрами: Constant value: 1; Sample time: 1 (рис. 11.4, верхний).

Размерность экспортируемой в Workspace последовательности  $Channel_1_1$  и ее значения приведены в табл. 11.2 (первый столбец). Одноканальная последовательность представлена в виде *трехмерного массива*  $1 \times 1 \times L = 1 \times 1 \times 5$  (3-D) (5 матриц размером  $1 \times 1$ ), где  $1 \times 1$  — количество каналов (один), а L = 5 — длина последовательности. При времени моделирования 4 (с) и периоде дискретизации 1 (с) одноканальная последовательность генерируется в моменты времени 0:4 (с) — на каждом один отсчет в одном канале (матрице  $1 \times 1$ ).



Рис. 11.4. Формирование последовательности на выходе блока Constant: одноканальной (Sample-based Channel Signal) и многоканальной (Sample-based Multichannel Signal) с параметром Constant value, заданным вектором (VECTOR) и матрицей (MATRIX)

Таблица 11.2. Одно- и многоканальная последовательности на выходе блока Constant

Одноканальная (пример 11.5)	Многоканальная (пример 11.7)	Многоканальная (пример 11.8)
>>	>> size(Channel_1_N)	>> size(Channel_M_N)
size(Channel_1_1)	ans =	ans =
ans =	1 6 5	2 3 5
1 1 5	>> Channel_1_N	>> Channel_M_N
>> Channel_1_1	Channel_1_N(:,:,1) =	Channel_M_N(:,:,1) =
Channel_1_1(:,:,1)	1 2 3 4 5 7	1 2 3
=	Channel_1_N(:,:,2) =	4 5 7
1	1 2 3 4 5 7	Channel_M_N(:,:,2) =
Channel_1_1(:,:,2)	Channel_1_N(:,:,3) =	1 2 3
=	1 2 3 4 5 7	4 5 7
1	Channel_1_N(:,:,4) =	
Channel_1_1(:,:,3)	1 2 3 4 5 7	Channel_M_N(:,:,3) =
=	Channel_1_N(:,:,5) =	1 2 3
	1 2 3 4 5 7	4 5 7
		Channel_M_N(:,:,4) =
1		1 2 3
$\frac{1}{(1 + 1)}$		4 5 7
		Channel_M_N(:,:,5) =
1		1 2 3
±		4 5 7

#### Пример 11.6

Сформируем одноканальную последовательность на выходе блока Sine Wave (см. разд. 4.2.19) с такими же параметрами, как в примере 11.2 (рис. 11.5).

Pasmephoctь экспортируемой в Workspace последовательности Y\_channel\_1: >> size(Y\_channel\_1) ans =

1

51

1

Одноканальная последовательность представлена в виде *трехмерного массива*  $1 \times 1 \times L = 1 \times 1 \times 51$  (3-D) (51 матрица  $1 \times 1$ ), где  $1 \times 1$ — количество каналов (один), а L = 51— длина последовательности. При времени моделирования 50 (с) и периоде дискретизации 1 (с) последовательность генерируется в моменты времени 0:50 (с) — на каждом один отсчет в единственном канале (матрице  $1 \times 1$ ).



Рис. 11.5. Формирование одноканальной последовательности (Sample-based Channel Signal) на выходе блока Sine Wave

## 11.1.4. Многоканальная последовательность

Многоканальная последовательность (Sample-based Multichannel Signal) представляется в виде *трехмерного массива*  $M \times N \times L$  (3-D) (L матриц  $M \times N$ ), где  $M \times N$  количество каналов (количество последовательностей), а L— количество отсчетов последовательности в каждом канале.

Многоканальная последовательность формируется в том случае, если численные параметры блока заданы векторами длины N (матрицами  $1 \times N$ ) или матрицами  $M \times N$  и флаг Interpret vector parameters as 1-D сброшен<sup>1</sup> (см. табл. 4.3).

<sup>&</sup>lt;sup>1</sup> Если численные параметры заданы матрицами  $M \times N$  (M > 1), то состояние флага Interpret vector parameters as 1-D игнорируется — он считается сброшенным по умолчанию.

В процессе моделирования многоканальная последовательность — трехмерный массив  $M \times N \times L$  — обрабатывается *по матрицам* — на каждом шаге одна матрица  $M \times N$  ( $M \times N$  отсчетов в  $M \times N$  каналах).

### Пример 11.7

Сформируем многоканальную последовательность на выходе блока Constant *(см. разд. 4.2.4)* с параметрами: **Constant value:** [1 2 3 4 5 7]; **Sample time:** 1 (рис. 11.4, Sample-based Multichannel Signal, Constant value — VECTOR).

Размерность экспортируемой в Workspace последовательности channel\_1\_N и ее значения приведены в табл. 11.2 (второй столбец). Многоканальная последовательность представлена в виде *трехмерного массива*  $M \times N \times L = 1 \times 6 \times 5$  (3-D) (L = 5 матриц  $M \times N = 1 \times 6$ ), где  $M \times N = 1 \times 6$ — количество каналов (количество последовательностей), а L = 5— длина последовательности в каждом канале. При времени моделирования 4 (с) и периоде дискретизации 1 (с) многоканальная последовательность генерируется в моменты времени 0:4 (с) — на каждом шаге матрица  $M \times N = 1 \times 6$ .

#### Пример 11.8

Сформируем многоканальную последовательность на выходе блока Constant *(см. разд. 4.2.4)* с параметрами: **Constant value:** [1 2 3;4 5 7]; **Sample time:** 1 (рис. 11.4, Sample-based Multichannel Signal, Constant value — MATRIX).

Размерность экспортируемой в Workspace последовательности channel\_M\_N и ее значения приведены в табл. 11.2 (третий столбец). Многоканальная последовательность представлена в виде *трехмерного массива*  $M \times N \times L = 2 \times 3 \times 5$  (3-D) (L = 5 матриц  $M \times N = 2 \times 3$ ), где  $M \times N = 2 \times 3$  — количество каналов (количество последовательностей), а L = 5 — длина последовательности в каждом канале. При времени моделирования 4 (c) и периоде дискретизации 1 (c) последовательность генерируется в моменты времени 0:4 (c) — на каждом шаге матрица  $M \times N = 2 \times 3$ .

#### Пример 11.9

Сформируем многоканальную последовательность на выходе блока Sine Wave (см. разд. 4.2.19) с такими же параметрами, как в примере 11.4 (рис. 11.6).

Pasmephoctь экспортируемой в Workspace последовательности y\_channel\_1\_N: >> size(Y\_channel\_1\_N) ans =

1 3

51

Многоканальная последовательность представлена в виде *трехмерного массива*  $M \times N \times L = 1 \times 3 \times 51$  (3-D) (L = 51 матриц  $M \times N = 1 \times 3$ ), где  $M \times N = 1 \times 3$  — количество каналов (количество последовательностей), а L — длина последовательности в каждом канале. При времени моделирования 50 (c) и периоде дискретизации 1 (c) последовательность генерируется в моменты времени 0:50 (c) — на каждом шаге матрица  $M \times N = 1 \times 3$ .



Рис. 11.6. Формирование многоканальной последовательности (Sample-based Multichannel Signal) на выходе блока Sine Wave, параметры которого заданы векторами (Parameters — VECTOR)

#### Пример 11.10

Сформируем многоканальную последовательность на выходе блока Sine Wave *(см. разд. 4.2.19)* с параметрами: Sine type: Sample based; Amplitude: [1 2 3;4 5 6]; Samples per period: [4 6 10;8 16 32]; Sample time: 1 (рис. 11.7).

Pasmephoctb экспортируемой в Workspace последовательности Y\_channel\_M\_N: >> size(Y\_channel\_M\_N) ans =

#### 2 3 51

Многоканальная последовательность представлена в виде *трехмерного массива*  $M \times N \times L = 2 \times 3 \times 51$  (3-D) (L = 51 матриц  $M \times N = 2 \times 3$ ), где  $M \times N = 2 \times 3$  — количество каналов (количество последовательностей), а L — длина последовательности в каждом канале. При времени моделирования 50 (с) и периоде дискретизации 1 (с) последовательность генерируется в моменты времени 0:50 (с) — на каждом шаге матрица  $M \times N = 2 \times 3$ .



Рис. 11.7. S-модель формирования многоканальной последовательности (Sample-based Multichannel Signal) на выходе блока Sine Wave, параметры которого заданы матрицами (Parameters — MATRIX)

## 11.1.5. Одноканальная последовательность фреймов

В группе блоков Sources библиотеки блоков Simulink имеется *только один блок Constant (см. разд. 4.2.4)*, для которого предусмотрена возможность представления последовательности фреймов, одноканальной или многоканальной. Для этого следует установить параметр **Sampling mode:** Framed based и указать период фрейма  $T_{fr}$  (с) в поле ввода **Frame period**.

Одноканальная последовательность фреймов (Frame-based Channel Signal) представляется в виде *матрицы*  $L \times 1$  (2-D) (вектора-столбца), где L — длина последовательности фреймов, а 1 — количество каналов (один).

В общем случае длина последовательности фреймов L равна произведению

$$L = K_{fr} \times L_{fr}, \tag{11.1}$$

где  $K_{fr}$  — количество фреймов, а  $L_{fr}$  — его длина (количество отсчетов на периоде фрейма  $T_{fr}$ ).

В блоке Constant по умолчанию значение  $L_{fr} = 1$ , поэтому, согласно (11.1),  $K_{fr} = L$ и длина последовательности L (количество фреймов  $K_{fr}$ ) определяется автоматически на интервале моделирования соотношением

$$L = K_{fr} = [ant(Stop time / T_{fr}) + 1],$$
 (11.2)

где Stop time — интервал моделирования (при Start time: 0), а функция ant выполняет округление числа до ближайшего целого в меньшую сторону.

Одноканальная последовательность фреймов формируется в том случае, если численные параметры блока заданы *скалярами*, и флаг Interpret vector parameters as 1-D *сброшен*.

В процессе моделирования одноканальная последовательность фреймов — матрица  $L \times 1$  — обрабатывается по фреймам — на каждом периоде фрейма  $T_{fr}$  один фрейм (скаляр) в одном канале.

#### Пример 11.11

Сформируем одноканальную последовательность фреймов на выходе блока Constant *(см. разд. 4.2.4)* с параметрами: **Constant value:** 5; **Sampling mode:** Frame based; **Frame period:** 4 (рис. 11.8, Frame-based Channel Signal).

Pasmephoctb экспортируемой в Workspace последовательности Frame\_1: >> size(Frame\_1) ans =

1

4

Одноканальная последовательность фреймов представлена в виде матрицы  $L \times 1 = 4 \times 1$  (2-D), где L — длина последовательности фреймов на интервале моделирования, а 1 — количество каналов (один). При времени моделирования 12 (с) и периоде фрейма **Frame period:** 4 одноканальная последовательность фреймов (скаляр 5) генерируется в моменты времени 0:12 (с) с периодом 4 (с) в одном канале. Согласно (11.2) длина последовательности фреймов L равна L = 12/4 + 1 = 4.



Рис. 11.8. Формирование последовательности фреймов на выходе блока Constant: одноканальной (Frame-based Channel Signal) и многоканальной (Frame-based Multichannel Signal)

# 11.1.6. Многоканальная последовательность фреймов

Многоканальная последовательность фреймов (Frame-based Multichannel Signal) представляется в виде *матрицы*  $L \times N$  (2-D), где L — длина последовательности фреймов (11.2), а N — количество каналов для N последовательностей фреймов длины L.

В многоканальной последовательности фреймов на выходе блока Constant *по* умолчанию в каждом канале длина фрейма  $L_{fr} = 1$ , а длина последовательности L определяется соотношением (11.2).

Многоканальная последовательность фреймов формируется в том случае, если численные параметры блока заданы *векторами* и флаг Interpret vector parameters as 1-D *сброшен*. Количество каналов *N равно длине вектора*.

В процессе моделирования многоканальная последовательность фреймов — *матрица*  $L \times N$  — обрабатывается по фреймам одновременно в N *каналах* — на каждом периоде фрейма  $T_{fr}$  матрица  $1 \times N$  (вектор-строка).

#### Пример 11.12

Сформируем многоканальную последовательность фреймов на выходе блока Constant *(см. разд. 4.2.4)* с параметрами: **Constant value:** [5 1 7]; **Sampling mode:** Frame based; **Frame period:** 4 (рис. 11.8, Frame-based Multichannel Signal). Pasmephoctb экспортируемой в Workspace последовательности Frame\_N: >> size(Frame\_N) ans =

#### 4

3

Многоканальная последовательность фреймов представлена в виде матрицы размером  $L \times N = 4 \times 3$  (2-D), где L = 4 — длина последовательности фреймов на интервале моделировании, а N — количество каналов. При времени моделирования 12 (с) и периоде фрейма **Frame period:** 4 многоканальная последовательность фреймов (вектор  $1 \times N = 1 \times 3 = [5 \ 1 \ 7]$ ) генерируется в моменты времени 0:12 (с) с периодом 4 (с) в N = 3 каналах; в каждом канале фрейм представлен скаляром 5, 1 и 7. Согласно (11.2) длина последовательности фреймов L равна L = 12/4 + 1 = 4.

# 11.2. Дискретные сигналы: библиотека блоков Signal Processing Blockset

В этом разделе рассматриваются S-модели дискретных сигналов и средств их анализа и управления из библиотеки блоков Signal Processing Blockset.

# 11.2.1. S-модели средств анализа дискретных сигналов

S-модели средств анализа дискретных сигналов представлены в группе блоков Signal Processing Sinks (Приемники обработки сигналов) (табл. 11.3).

N⁰	Блок	Назначение
1.	Display	Имитирует дисплей — отображает численные значения
2.	Matrix Viewer	Отображает элементы матрицы в цвете
3.	Signal To Work- space	Экспортирует последовательности в Workspace
4.	Triggered To Workspace	Экспортирует последовательности в Workspace при активизации на управляющем входе
5.	Time Scope	Имитирует осциллограф — отображает сигналы в процессе моделирования
6.	Vector Scope	Имитирует цифровой осциллограф — отображает последовательности в процессе моделирования
7.	Waterfall	Имитирует цифровой трехмерный осциллограф ("водопадную диаграмму") — отображает изменение вектора данных во времени

Таблица 11.3.	. Основные блоки	группы блоков	Signal	Processing Sinks
---------------	------------------	---------------	--------	------------------

## 11.2.1.1. Блок Display

Блок Display дублирует одноименный блок из библиотеки блоков Simulink (см. разд. 5.2.1).

## 11.2.1.2. Блок Matrix Viewer

Блок *Matrix Viewer* отображает элементы матрицы в цвете. Параметры блока задаются на двух вкладках — **Image Properties** (Свойства изображения) и **Axis Properties** (Свойства осей) (рис. 11.9).

На вкладке Image Properties определяются параметры:

Colormap matrix (Матричная палитра) — палитра цветов (табл. 11.4) для отображения элементов матрицы в окне Matrix Viewer, автоматически открываемом после запуска моделирования; в круглых скобках указывается количество цветовых градаций;

Символическое имя палитры	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
lines	С чередованием цветов
autumn	Красный и желтый
spring	Желтый и фиолетовый
winter	Синий и зеленый
summer	Зеленый и желтый

Таблица 11.4. Стандартная палитра

- □ Minimum input value (Минимальное входное значение) минимальный элемент матрицы с минимальной градацией выбранной палитры;
- □ **Maximum input value** (Максимальное входное значение) максимальный элемент матрицы с максимальной градацией выбранной палитры;
- □ Display colorbar (Изображение шкалы цветов) флаг, при установке которого в окне Matrix Viewer выводится шкала цветов выбранной палитры.

На вкладке Axis Properties определяются параметры:

- Axis origin (Начало отсчета) начало отсчета по оси ординат: Upper left corner (Левый верхний угол) и Lower left corner (Левый нижний угол);
- □ X-axis title (Заголовок оси "X") и Y-axis title (Заголовок оси "Y") соответственно поля ввода для обозначения осей абсцисс и ординат;

- □ Colorbar title (Заголовок шкалы частот) поле ввода для обозначения шкалы частот;
- □ Figure position, [x y width height] (Позиция фигуры, [x y ширина высота]) вектор позиции (Position Vector) из четырех элементов, определяющий местоположение и размеры окна Matrix Viewer посредством указания следующих значений в пикселях: x и y расстояния соответственно от левой и нижней границ монитора, где нулевые значения указывают на левый нижний угол; "ширина", "высота" соответственно ширина и высота окна Matrix Viewer;
- □ Axis zoom (Масштаб осей) флаг, при установке которого окно Matrix Viewer полностью заполняется матрицей.

Sink Block Parameters: Matrix Viewer	Sink Block Parameters: Matrix Viewer
Matrix Viewer Display a matrix as an image, scaling the colormap to the specified inp Colormap must be an Nx3 matrix of RGB values. Type "help graph3d" MATLAB prompt for a list of predefined colormaps.	Matrix Viewer Display a matrix as an image, scaling the colormap to the specified input data range. Colormap must be an Nx3 matrix of RGB values. Type "help graph3d" at the MATLAB prompt for a list of predefined colormaps.
Image Properties Axis Properties	Image Properties Axis Properties
Parameters	Parameters
Colormap matrix: cool(256)	Axis origin: Upper left corner
Minimum input value: 1.0	X-axis title: Rows
Maximum input value: 10.0	Y-axis title: Columns
☑ Display colorbar	Colorbar title: Coler
	Figure position, [x y width height]: [100 100 320 250]
Matrix_Constant/Matrix Viewer	Axis zoom
0.5	
1	
1.5	QK <u>C</u> ancel <u>H</u> elp Apply
2.5	
3	4 5 6 Matrix
2	
3.5 1 2 3 Rows	Constant Matrix Viewer

Рис. 11.9. Параметры блока Matrix Viewer и отображение матрицы на выходе блока Constant

## 11.2.1.3. Блок Signal To Workspace

Блок *Signal To Workspace* экспортирует последовательности в Workspace и имеет следующие параметры (рис. 11.10):

□ Variable name (Имя переменной), Limit data points to last (Предельное число точек для сохранения), Decimation (Децимация) — *см. разд. 5.2.8*;

- □ Frames (Фреймы) варианты вывода последовательности фреймов:
  - Concatenate frames (2-D array) (Объединение фреймов (массив 2-D)) фреймы выводятся в виде матрицы (2-D);
  - Log frames separately (3-D array) (Раздельная регистрация фреймов (массив 3-D)) фреймы выводятся раздельно в виде трехмерного массива (3-D);
- □ Log fixed-point data as a fi object (Регистрация данных с фиксированной точкой в виде объекта fi) флаг, при установке которого данные с ФТ выводятся в виде объекта fi [22].

Sink Block Parameters: Signal To workspace	Ľ
Signal To Workspace (mask) (link)	
Write input to specified array in MATLAB's main workspace. Data is not the simulation is stopped or paused.	t available unti
Parameters	
Variable name:	
Y	
Limit data points to last:	
inf	
Decimation:	
1	
Frames: Concatenate frames (2-D array)	*
Log fixed-point data as a fi object	
QK <u>C</u> ancel <u>H</u> elp	Apply

Рис. 11.10. Параметры блока Signal To Workspace

## 11.2.1.4. Блок Triggered To Workspace

Блок *Triggered To Workspace* экспортирует последовательности в Workspace при активизации на управляющем входе и имеет следующие параметры (рис. 11.11):

□ **Trigger type** — *см. разд. 8.3.2*;

- □ Variable name см. разд. 5.2.8;
- Maximum number of rows (Максимальное число строк) предельное количество отсчетов последовательности (строк вектора или матрицы), сохраняемых в процессе моделирования.

Значение inf соответствует сохранению всех значений последовательности на интервале моделирования.

Если предельное количество отсчетов последовательности будет меньше ее длины, например 10 при длине 50, то в Workspace будут сохраняться *последние* 10 значений последовательности;

□ **Decimation** — *см. разд. 5.2.8*;

266

#### □ Log fixed-point data as a fi object — *cм. paзd. 11.2.1.3.*

На рис. 11.11 приведен пример экспорта дискретной синусоиды в Workspace, активизируемого положительным фронтом дискретного сигнала на выходе блока Pulse Generator. Для блока Sine Wave (см. разд. 4.2.19) установлены параметры: Sine type: Sample based; Amplitude: [2 3]; Samples per period: [7 17]; Sample time: 1; флаг Interpret vector parameters as 1-D установлен. Это соответствует многосигнальной последовательности — матрице  $L \times N$ . Для блока Pulse Generator — параметры: Pulse type: Sample based; Amplitude: 1; Period (number of samples): 10; Pulse width (number of samples): 5; Sample time: 1; флаг Interpret vector parameters as 1-D установлен. Это соответствует за 1-D установлен. Это соответствует за 1-D установлен. Это соответствует многосигнальной последовательности — матрице 1; Флаг Interpret vector parameters as 1-D установлен. Это соответствует за 1-D установлен. Это соответствует за 1-D установлен. Это соответствует лисе сигнальной последовательности.

На интервале моделирования 50 с пять положительных фронтов, поэтому в Workspace экспортируется матрица Yout размером  $L \times N = 5 \times 2$ :

```
>> size(Yout)
ans =
     5
            2
>> Yout
Yout =
    0.8678
              -1.5793
   -1.5637
               2.6855
    1.9499
              -2.9872
   -1.9499
               2.3941
    1.5637
              -1.0837
```



Рис. 11.11. Параметры блока Triggered To Workspace и пример экспорта дискретной синусоиды в Workspace, активизируемого положительным фронтом на управляющем входе

## 11.2.1.5. Блок Тіте Scope

Блок *Time Scope* дублирует блок Scope из библиотеки блоков Simulink (см. разд. 5.2.3).

## 11.2.1.6. Блок Vector Scope

Блок *Vector Scope* имитирует цифровой осциллограф — отображает последовательности в процессе моделирования. Параметры блока задаются на четырех вкладках — **Scope Properties** (Свойства осциллографа), **Display Properties** (Свойства дисплея), **Axis Properties** (Свойства осей) и **Line Properties** (Свойства линии) (рис. 11.12).

На вкладке Scope Properties определяются параметры:

- □ Input domain (Область входной последовательности) область представления входной последовательности:
  - Тіте временная;
  - Frequency частотная;
  - User-defined указываемая пользователем;
- □ **Time display span (number of frames)** (Интервал отображаемого времени (количество фреймов)) количество фреймов K<sub>fr</sub> в последовательности с длиной фрейма L<sub>fr</sub>.

О длине фрейма  $L_{fr}$  речь пойдет далее *(см. разд. 11.2.3.3—11.2.3.4)*, пока лишь отметим, что для последовательностей фреймов она выбирается из условия L > 1, а для других видов последовательности —  $L_{fr} = 1$ . Исключение составляет блок Constant с  $L_{fr} = 1$  *(см. разд. 11.1.5—11.1.6)*.

При выборе **Input domain:** Тіте ось абсцисс трактуется как ось *дискретного* времени nT, конечное значение которого при начальном значении nT = 0 определяется произведением

$$(K_{fr} - 1)TL_{fr}$$
, (11.3)

где *Т* — период дискретизации (параметр **Sample time**);

 $K_{fr}$  — количество фреймов (параметр **Time display span (number of frames)**).

Для удобства пользователя в окне <**Имя файла**>/Vector Scope внизу выводится автоматически рассчитываемое максимальное количество фреймов на заданном интервале моделирования — *Frame*.

При выборе Input domain: Frequency ось абсцисс трактуется как ось дискретной частоты на заданном интервале (см. далее параметр Frequency range) с *шагом*  $F_s/L_{fr}$  (Гц), где  $F_s$  — частота дискретизации.

На вкладке Display Properties задаются параметры:

- □ Show grid (Показать сетку) флаг, при установке которого на графике наносится координатная сетка;
- □ Persistence (Продолжительность) флаг, при установке которого на графике сохраняются предыдущие фреймы;
- □ Frame number (Количество фреймов) флаг, при установке которого на графике выводится количество фреймов Frames;
- □ Channel legend (Легенда канала) флаг, при установке которого на поле графика в правом верхнем углу размещается легенда<sup>1</sup> каналов;
- □ **Compact display** (Компактное изображение) флаг, при установке которого выводится поле графика без дополнительной информации;
- □ Open scope at start of simulation (Отрыть осциллограф при запуске моделирования) — флаг, при установке которого окно <Имя файла>/Vector Scope цифрового осциллографа открывается автоматически после запуска моделирования;
- □ Scope position (Размещение осциллографа) четырехэлементный вектор, определяющий местоположение и размеры окна <Имя файла>/Vector Scope (см. аналогичный параметр Figure position, [x y width height] блока Matrix Viewer (см. разд. 11.2.1.2)).

На вкладке Axis Properties при выборе Input domain: Тіте будут параметры:

- □ **Time display limits** (Границы изображения времени) границы временного интервала:
  - Auto (по умолчанию) устанавливаются автоматически;
  - User-defined (Определяемые пользователем) устанавливаются пользователем: левая в поле Minimum X-limit и правая в поле Maximum X-limit;
- □ Minimum Y-limit и Maximum Y-limit минимальное и максимальное значение по оси ординат;

□ **Y-axis title** (Заголовок оси Y) — поле ввода для обозначения оси ординат.

На вкладке Axis Properties при выборе Input domain: Frequency будут параметры:

- □ Frequency units (Единицы измерения частоты): Hertz герцы и rad/sec рад/с;
- □ Frequency range (Диапазон частот) диапазон частот при выводе графика:
  - [0... Fs/2];
  - [0... Fs];
  - [-Fs/2... Fs/2],

где Fs — частота дискретизации;

<sup>&</sup>lt;sup>1</sup> Соответствует функции MATLAB legend.

- □ Inherit sample time from input (Наследование периода дискретизации от входа) — флаг, при установке которого период дискретизации наследуется от входного сигнала, а при сбросе — указывается в поле ввода Sample time of original series;
- □ Frequency display limits (Границы изображения частоты) границы частотного интервала (см. аналогичный параметр Time display limits);
- □ Amplitude scaling (Масштабирование амплитуды) единицы измерения по оси ординат: Magnitude безразмерные и dB в децибелах;
- □ Minimum Y-limit, Maximum Y-limit и Y-axis title см. одноименные параметры при выборе Input domain: Time.

На вкладке Line Properties задаются параметры:

- □ Line visibilities (Видимость линий) видимость графиков: on (видимый) и off (невидимый); при нескольких графиках ставится разделительный символ " | "; пустому полю соответствует видимость всех графиков;
- □ Line styles (Типы линий) тип линии (табл. 11.5); при нескольких графиках ставится разделительный символ "|"; пустому полю соответствуют непрерывные линии;

Символ	Тип линии
-	Solid (непрерывная)
	Dashed (пунктир, длинный штрих)
:	Dotted (пунктир, короткий штрих)
	Dash-dot (штрихпунктир)
none	Отсутствие линии

Таблица 11.5. Символическое обозначение типов линии

□ Line markers (Маркеры линии) — тип маркера (табл. 11.6); при нескольких графиках ставится разделительный символ " | "; пустому полю соответствует от-сутствие маркеров.

Особо отметим символ stem, при котором выводятся отсчеты последовательности, подобно одноименной функции МАТLAB. В этом случае количество фреймов (Frame) в нижней строке окна **«Имя файла»/Vector Scope** автоматически увеличивается на единицу по сравнению с любым другим маркером (или его отсутствием).

Поэтому при выводе последовательности (в том числе последовательности фреймов) с маркером stem значение параметра Time display span (number of frames) должно указываться на единицу меньшим значения Frame. Их равенство приведет к задержке последовательности на период дискретизации;

Таблица 11.6.	Символическое обозн	ачение маркеров
---------------	---------------------	-----------------

Символ	Тип маркера
*	Asterisk (звездочка)
•	Point (точка)
х	Cross (крестик)
S	Square (квадрат)
d	Diamond (ромб)
stem	Вывод графика последовательности (отсчетов)

□ Line colors (Цвета линии) — цвет линии, указываемый тройкой элементов вектора (табл. 11.7).

Тройка элементов	Цвет линии
[1 1 0]	Yellow (желтый)
[1 0 1]	Magenta (фиолетовый)
[0 1 1]	Cyan (голубой)
[1 0 0]	Red (красный)
[0 1 0]	Green (зеленый)
[0 0 1]	Blue (синий)
[1 1 1]	White (белый)
[0 0 0]	Black (черный)

Таблица 11.7. Основные цвета линии

Окно **«Имя файла»**/**Vector Scope** содержит *стандартное меню* и *панель инструментов*, которые легко осваиваются самостоятельно после знакомства с параметрами блока Vector Scope.

Перед новым запуском моделирования, в том числе с измененными параметрами, окно **«Имя файла»/Vector Scope** следует закрыть.

На рис. 11.13 представлено отображение с помощью блока Vector Scope (см. рис. 11.12) *многоканальной последовательности* на выходе блока Sine Wave (см. разд. 4.2.19) с параметрами: Amplitude: [3 5 7]; Samples per period: [8 16 32]; Sample time: 1; флаг Interpret vector parameters as 1-D сброшен.

Выведем размерность экспортируемой в Workspace последовательности у: >> size(Y)

ans =

1 3 51

Sink Block Parameters: Vector Scope	Sink Block Parameters: Vector Scope
Vector Scope Display a vector or matrix of time-domain, frequency-domain, or use Each column of a 2-D input matrix is plotted as a separate data char are assumed to be a single data channel. For frequency-domain operation, input should come from a source s Magnitude FFT block, or a block with equivalent data organization.	Vector Scope      Display a vector or matrix of time-domain, frequency-domain, or user-specified data.     Each column of a 2-D input matrix is plotted as a separate data channel. 1-D inputs     are assumed to be a single data channel.     For frequency-domain operation, input should come from a source such as the     Magnitude FFT block, or a block with equivalent data organization.
Scope Properties Display Properties Axis Properties Line Parameters Input domain: Time Time display span (number of frames): 51	Scope Properties     Display Properties     Axis Properties     Line Properties       Parameters       Show grid       Persistence       V     Frame number       V     Channel legend       Compact display       V     Open scope at start of simulation       Scope position:     [100 100 320 250]
QK <u>Cancel</u> <u>Help</u> Display a vector or matrix of time-domain, frequency-domain, or use Each column of a 2-D input matrix is plotted as a separate data chan are assumed to be a single data channel. For frequency-domain operation, input should come from a source su Magnitude FFT block, or a block with equivalent data organization.	QK         Cancel         Help         Apply           Display a vector or matrix of time-domain, frequency-domain, or user-specified data.         Each column of a 2-D input matrix is plotted as a separate data channel.         1-D input sare assumed to be a single data channel.         1-D input sare assumed to be a single data channel.           For frequency-domain operation, input should come from a source such as the Magnitude FFT block, or a block with equivalent data organization.         1-D input sare assumed to be a single data channel.
Scope Properties         Display Properties         Axis Properties         Line           Parameters         Time display limits: Auto         Iminum Y-limit: 10         Iminum Y	Scope Properties       Display Properties       Axis Properties       Line Properties         Parameters       Line visibilities:
QK Cancel Help	QK Cancel Help Apply

Рис. 11.12. Параметры блока Vector Scope



Рис. 11.13. Отображение многоканальной последовательности с помощью блока Vector Scope

## 11.2.1.7. Блок Waterfall

Блок *Waterfall* имитирует цифровой трехмерный осциллограф ("водопадную диаграмму") — отображает изменение вектора данных во времени в окне осциллографа **«Имя файла»/Waterfall** (рис. 11.14), которое открывается двойным щелчком левой кнопки мыши на пиктограмме блока.

Вектор данных — это вектор длины N с символическим именем u и элементами, изменяющимися во времени. Условно обозначим его как u(n-i), где n=0,1,...,(L-1) — дискретное нормированное время, максимальное значение (L-1) которого зависит от времени моделирования и периода дискретизации T(параметр **Sample time**), а i=0, 1, ..., (M-1) — задержка по времени, максимальное значение (M-1) которой будет определять количество векторов данных, одновременно отображаемых в окне осциллографа. Всего будет отображаться Mвекторов данных, включая вектор u(n) и (M-1) его задержанных копий u(n-i), i=1, 2, ..., (M-1).

После запуска моделирования в окне осциллографа <**Имя файла**>/**Waterfall** можно наблюдать изменение *во времени nT* значений элементов всех *M* векторов данных.

Проиллюстрируем работу осциллографа Waterfall на примере *многосигнальной последовательности* (Sample-based Multi Signal) (рис. 11.15) на выходе блока Sine Wave (см. разд. 4.2.19) с параметрами: Sine type: Sample based; Amplitude: [10 2 7 1 5]; Samples per period: [7 17]; Sample time: 2; флаг Interpret vector parameters as 1-D установлен. В этом случае вектор данных u(n) имеет длину N = 5, а его элементами являются значения N = 5 последовательностей в момент дискретного нормированного времени n.

Трехмерное пространство окна **«Имя файла»/Waterfall** (см. рис. 11.14) организовано следующим образом: по оси "X" ("Amplitude" по умолчанию) откладываются индексы M векторов данных в диапазоне, устанавливаемом автоматически; по оси "Y" ("Samples" по умолчанию) — значения элементов M векторов данных, изменяющиеся во времени nT; по оси "Z" ("Time" по умолчанию) — значения задержек (-i), i = 0, 1, ..., (M-1).

Окно трехмерного осциллографа **<Имя файла>/Waterfall** (см. рис. 11.14) содержит меню и панель инструментов со следующими кнопками:

- Scope parameters (Параметры осциллографа) установка параметров (Parameters) блока Waterfall;
- □ Save scope position and view (Сохранить местоположение и вид осциллографа) сохранение местоположения и изображения осциллографа;
- Restore scope position and view (Восстановить местоположение и вид осциллографа) восстановление местоположения и изображения;
- □ Select (Выбрать) выделение одного из *M* векторов данных (повторное нажатие отменяет выделение).





Курсор ставится на шкалу времени (см. рис. 11.14), строго на линию требуемой задержки (-i), и выполняется щелчок левой кнопкой мыши, после чего вектор данных выделяется черной линией и отображается в нижней строке состояния как u(-i) (см. рис. 11.15);



Рис. 11.15. Анализ многосигнальной последовательности с помощью блока Waterfall

- Orbit camera (Двигать камеру по орбите) вращение изображения; курсор подводится к любой точке изображения и при нажатой левой кнопке мыши выполняется его вращение;
- Zoom camera (Изменять масштаб обзора камеры) изменение масштаба изображения; курсор подводится к любой точке изображения и при нажатой левой кнопке мыши выполняется изменение его масштаба;
- □ Link scopes (Соединить осциллографы) синхронизация работы осциллографов.
- Если S-модель системы содержит несколько блоков Waterfall, то при нажатых в них кнопках Link scopes действия в процессе моделирования выполняются синхронно;
- Suspend data capture (Приостановить обработку данных) останов обработки данных с фиксацией текущего изображения и момента времени в строке состояния; активна после запуска моделирования;
- Snapshot display (Моментально остановить изображение) завершение обработки данных с возвратом изображения в исходное состояние; активна после запуска моделирования;
- □ Export to workspace (Экспорт в Workspace) экспорт векторов данных в Workspace со стандартным именем ExportData;

- □ Export to SPTool (Экспорт в SPTool) экспорт векторов данных в GUI SPTool в виде массива записей (struct array) со стандартным именем Export-Data;
- □ Hide grid (Скрыть сетку) удаление координатной сетки;
- Rescale amplitude (Изменить масштаб амплитуды) изменение масштаба по оси Samples (см. рис. 11.14) — наибольшее значение становится равным нулю; активна после запуска моделирования;
- □ Fit to view (Подогнать к подходящему виду) приведение изображения к наиболее удачному расположению в трехмерном пространстве;
- Go to scope block (Идти к блоку осциллографа) выделение цветом блока Waterfall в окне S-моделей (удобно, если блоков несколько); активна после запуска моделирования.

Параметры блока Waterfall задаются при нажатой кнопке Scope parameters в открывающемся окне Parameters с пятью вкладками: Display (Изображение), Axes (Оси), Data history (История данных), Triggering (Запуск) и Transforms (Преобразование) (см. рис. 11.14).

На вкладке **Display** определяются параметры:

□ в группе Display Properties (Свойства изображения):

- **Display traces** (Изображаемые следы) количество *M* векторов данных, одновременно отображаемых в окне <**Имя файла**>/**Waterfall**;
- Update interval (Обновить интервал) количество векторов данных, сохраняемых в блоке Waterfall до их отображения в окне <Имя файла>/Waterfall;
- **Colormap** (Палитра) стандартные палитры цветов для отображения *М* векторов данных;
- □ в группе **Transparency** (Прозрачность) два движка **Newest** (Новейшие) и **Oldest** (Старейшие), регулирующие возрастание прозрачности изображения *M* векторов данных, соответственно, с уменьшением и возрастанием задержки *i* = 0, 1, ..., (*M* − 1).

На вкладке Axes определяются параметры:

□ в группе Axis Properties (Свойства оси):

- **Y-Min** и **Y-Max** граничные значения по оси "Y" ("Samples" по умолчанию);
- Axis color (Цвет оси) цвет плоскостей трехмерного пространства, указываемый в апострофах (табл. 11.8);
- □ в группе Axis Labels (Метки оси) соответственно обозначение осей "Х" (X-Axis), "Y" (Y-Axis) и "Z" (Z-Axis).

Символ	Цвет плоскости
У	Yellow (желтый)
m	Magenta (фиолетовый)
С	Cyan (голубой)
r	Red (красный)
g	Green (зеленый)
b	Blue (синий)
W	White (белый)
k	Black (черный)

Таблица 11.8. Цвет плоскостей трехмерного пространства

На вкладке Data history задаются параметры:

**В** группе **Data Review** (Просмотр данных):

- **History traces** (Прошлые следы) количество векторов данных, сохраняемых в буфере блока Waterfall;
- When buffer is full (Когда буфер заполнен) варианты действия при заполненном буфере блока Waterfall:
  - Overwrite старые данные заменяются новыми;
  - Suspend сохранение данных приостанавливается, но процесс моделирования продолжается;
  - Extend буфер расширяется для сохранения всех входных данных;

**В** группе **Export Options** (Опции экспорта):

- Data logging (Регистрация данных) варианты контроля при экспорте данных:
  - Selected экспортируется выделенный вектор данных;
  - All visible экспортируются векторы данных, отображаемые в окне <Имя файла>/Waterfall, количество которых указано параметром Display traces;
  - All history экспортируются векторы данных, отображаемые в окне <Имя файла>/Waterfall, плюс векторы данных, сохраняемые в буфере блока Waterfall, указанные параметром History traces;
- Export variable (Экспортируемая переменная) имя переменной для данных, экспортируемых в Workspace или в GUI SPTool; по умолчанию ExportData;
- Export at end of simulation (Экспорт в конце моделирования) флаг, при установке которого данные автоматически экспортируются в Workspace.

На вкладке **Triggering** задаются параметры, контролирующие начало и конец обработки данных блоком Waterfall. В простейшем случае задаются два параметра со следующими значениями:

- □ Begin recording (Начало регистрации) начало обработки данных; при выборе значения Immediately блок Waterfall начинает обработку данных одновременно с запуском моделирования;
- □ Stop recording (Конец регистрации) завершение обработки данных; при выборе значения Never блок Waterfall продолжает обработку данных до тех пор, пока продолжается процесс моделирования.

На вкладке Transforms определяются параметры:

- □ **Transform** (Преобразование) варианты преобразования отображаемых векторов данных; при **Transform:** None преобразование не выполняется;
- □ Function (Функция) окно для ввода имени стандартной функции MATLAB; активизируется при выборе Transform: User-defined fcn;
- □ Expression (Выражение) окно для ввода арифметического выражения MATLAB; активизируется при выборе Transform: User-defined expr.

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Signal Processing Sources.

# 11.2.2. Ѕ-модели дискретных сигналов

S-модели дискретных сигналов представлены в группе блоков Signal Processing Sources (Источники обработки сигналов) (табл. 11.9). При соответствующем задании параметров, некоторые S-модели могут использоваться для генерации аналоговых сигналов.

N⁰	Блок	Назначение		
1.	Chirp (mask)	Генерирует косинусоиду с качанием частоты		
2.	Constant	Выдает сигнал в виде данных — численных или логических		
3.	Constant Diagonal Matrix (mask)	Генерирует квадратную диагональную матрицу		
4.	Discrete Impulse (mask)	Генерирует цифровой единичный импульс		
5.	Identity Matrix (mask)	Генерирует единичную матрицу		
6.	Random Sources (mask)	Генерирует белый шум, равномерный или нормальный		
7.	Signal From Work- space	Импортирует последовательности из Workspace		
8.	Sine Wave	Генерирует гармонический сигнал		

Таблица 11.9. Основные блоки из группы блоков Signal Processing Sources

## 11.2.2.1. Блок Chirp

Блок *Chirp* генерирует косинусоиду с качанием частоты и имеет следующие параметры (рис. 11.16):

- □ Frequency sweep (Качание частоты<sup>1</sup>) закон качания частоты косинусоиды на периоде:
  - Linear линейный;
  - Swept cosine линейный с удвоенным изменением частоты и вычитанием ее начального значения;
  - Logarithmic логарифмический;
  - Quadratic квадратичный;
- □ Sweep mode (Режим качания) характер направленности в качании частоты на периоде:
  - Unidirectional однонаправленное;
  - Bidirectional двунаправленное;
- □ Initial frequency (Hz) (Начальная частота (Гц)) начальная частота в герцах;
- □ Target frequency (Hz) (Заданная частота (Гц)) частота в герцах на момент времени Target time (s);
- □ **Target time (s)** (Заданное время (c)) момент времени, когда мгновенная частота достигает заданного значения **Target frequency**;
- □ Sweep time (s) (Период качания (c)) период качания частоты.

На периоде качания Sweep time значения мгновенной частоты определяются автоматически согласно параметрам Frequency sweep, Target frequency (Hz) и Target time (s). Поэтому значение параметра Sweep time должно быть большим либо равным значению Target time.

При выборе двунаправленного режима качания (Sweep mode: Bidirectional) параметр Sweep time соответствует половине периода качания частоты;

- □ Initial phase (rad) (Начальная фаза (рад)) начальная фаза косинусоиды (в радианах) на периоде Sweep time;
- □ **Sample time** *см. разд. 4.2.1*;
- □ Samples per frame (Количество отсчетов во фрейме) длина фрейма;
- □ **Output data type** (Тип данных на выходе) тип данных на выходе (см. табл. 4.5).

<sup>&</sup>lt;sup>1</sup> Качанием частоты косинусоиды называют плавное изменение ее мгновенной частоты на заданном периоде по заданному закону.



Рис. 11.16. Параметры и генерируемый сигнал блока Chirp

## 11.2.2.2. Блок Constant

Блок *Constant* дублирует одноименный блок из библиотеки блоков Simulink (см. разд. 4.2.4).

## 11.2.2.3. Блок Constant Diagonal Matrix

Блок *Constant Diagonal Matrix* генерирует квадратную диагональную матрицу. Параметры блока задаются на двух вкладках — Main и Data Types.

На вкладке Main определяются параметры:

- Constant(s) along diagonal (Константа(ы) по диагонали) поле ввода элементов главной диагонали в виде вектора;
- □ Frame-based output (Выход как последовательность фреймов) флаг, при установке которого последовательность представляется в виде последовательности фреймов.

На вкладке **Data Types** задается один параметр **Output data type** — тип данных на выходе (см. табл. 4.5).

На рис. 11.17 приведен пример генерирования диагональной матрицы при Constant(s) along diagonal: [3 5].



Рис. 11.17. Пример генерирования диагональной матрицы с помощью блока Constant Diagonal Matrix

## 11.2.2.4. Блок Discrete Impulse

Блок *Discrete Impulse* генерирует цифровой единичный импульс. Параметры блока задаются на двух вкладках — Main и Data Types.

На вкладке Main определяются параметры:

- □ Delay (samples) (Задержка (отсчеты)) задержка цифрового единичного импульса в области дискретного нормированного времени *n* (эквивалентно количеству отсчетов);
- □ **Sample time** *см. разд. 4.2.1*;

□ Samples per frame (Количество отсчетов во фрейме) — длина фрейма.

На вкладке **Data Types** задается один параметр **Output data type** — тип данных на выходе (см. табл. 4.5).

На рис. 11.18 приведен пример генерирования цифрового единичного импульса при **Delay (samples):** 5, **Sample time:** 1 и **Samples per frame:** 1.



Рис. 11.18. Пример генерирования цифрового единичного импульса с помощью блока Discrete Impulse

## 11.2.2.5. Блок Identity Matrix

Блок *Identity Matrix* дублирует одноименный блок из библиотеки блоков Simulink (см. разд. 6.2.2.4).

## 11.2.2.6. Блок Random Sources

Блок *Random Sources* генерирует белый шум, равномерный или нормальный. Параметры блока зависят от заданного значения параметра **Source type** (Тип источника), который определяет вид дискретного белого шума: Uniform — равномерный; Gaussian — нормальный.

При выборе значения Source type: Uniform будут параметры (рис. 11.19):

- Minimum и Maximum минимальное и максимальное значения случайной последовательности;
- Repeatability (Повторяемость) варианты повторяемости случайного сигнала при каждом запуске моделирования:
  - Repeatable (Повторяемый) или Not repeatable (Не повторяемый) начальное значение (Initial seed), запускающее генератор случайных чисел, автоматически задается, соответственно, одинаковым или различным;
  - Specify seed повторяемый с заданным начальным значением Initial seed, указываемым в поле ввода Initial seed (Инициализатор);
- □ Inherit output port attributes (Наследовать атрибуты выходного порта) флаг, при установке которого наследуются атрибуты сигнала на выходе блока;
- □ Sample mode (Выбор режима) тип случайного сигнала: Discrete дискретный или Continuous аналоговый;
- □ Sample time см. разд. 4.2.1;
- □ Samples per frame (Количество отсчетов во фрейме) длина фрейма;
- **Оиtput data type** тип данных на выходе (см. табл. 4.5).
- □ Complexity (Комплексность) признак вещественного (Real) или комплексного (Complex) случайного сигнала.

При выборе значения Source type: Gaussian будут параметры:

- □ Method (Метод) метод генерирования случайных чисел;
- □ Mean (Среднее) математическое ожидание (среднее значение);
- □ Variance (Дисперсия) дисперсия.

Остальные параметры описаны ранее для значения параметра Source type: Uniform.



Рис. 11.19. Параметры блока Random Sources и пример генерирования дискретного равномерного белого шума

## 11.2.2.7. Блок Signal From Workspace

Блок Signal From Workspace импортирует последовательности из Workspace и содержит следующие параметры (рис. 11.20):

- Signal (Сигнал) имя последовательности (вектора или матрицы) в Workspace или выражение, ее определяющее, с использованием символов арифметических операций и стандартных функций MATLAB;
- □ Sample time *см. разд. 4.2.1*;
- □ Samples per frame (Количество отсчетов во фрейме) длина фрейма;
- □ Form output after final data value by (Формирование выхода по завершении последнего значения данных посредством) — варианты формирования значений последовательности по завершении времени моделирования:
  - Setting to zero дополняется нулями;
  - Holding final value дополняется значением последнего отсчета;
  - Cyclic repetition циклически повторяется.

На рис. 11.20 приведен пример генерирования дискретного сигнала на основе импортируемой из Workspace последовательности в виде вектора A:

>> n = 0:20; >> A = sqrt(n);

S_work *	
Eile Edit View Simulation Format Tools Help	
[] [ [ 문 문 ] 사 백 립 ( 수 수 수   그 오 ]	▶ = 30.0 Normal
sin(A) Signal From Workspace Vector Scope S_work/Vector Scope Elle Axes Channels Window Help	Source Block Parameters: Signal From Workspace     Signal From Workspace (mask) (link)     Output signal samples obtained from the MATLAB workspace at successive sample     times. A signal matrix is interpreted as having one channel per column. Signal columns     may be buffered into frames by specifying a number of samples per frame greater     than 1.     An M x N x P signal array outputs M x N matrices at successive sample times. The     samples per frame must be equal to 1 for three-dimensional signal arrays.
1 1 1 -2 -2 -2 -2 -2 -2 -2 -2 -2 -2	Parameters         Signal:         sin(A)         Sample time:         1         Samples per frame:         1         Form output after final data value by: Setting to zero         QK       Cancel

Рис. 11.20. Параметры блока Signal From Workspace и пример генерирования дискретного сигнала

#### 11.2.2.8. Блок Sine Wave

Блок *Sine Wave* генерирует гармонический сигнал одного из следующих видов: аналоговый вещественный:

$$x(t) = A\mathbf{x}\sin(2 \mathbf{f} t + \mathbf{\phi} ; \qquad (11.4)$$

аналоговый комплексный:

$$x(t) = Ae^{j(2\pi f t + \phi)};$$
 (11.5)

□ дискретный вещественный:

$$x(nT) = \pi 4 \sin(2 \int nT + \varphi ; \qquad (11.6)$$

□ дискретный комплексный:

$$x(nT) = Ae^{j(2\pi f nT + \varphi)}.$$
 (11.7)

Параметры блока задаются на двух вкладках — **Main** и **Data Types** (рис. 11.21). На вкладке **Main** определяются параметры:

□ Amplitude (Амплитуда) — амплитуда *А*;

- **Г** Frequency (Hz) (Частота ( $\Gamma$ ц)) частота f в герцах;
- □ Phase offset (rad) (Сдвиг фазы (рад)) фаза φ в радианах;
- □ Sample mode (Выбор режима) тип гармонического сигнала: дискретный (Discrete) или аналоговый (Continuous);
- □ **Output complexity** (Комплексность выхода) признак вещественного (Real) или комплексного (Complex) гармонического сигнала;
- □ Computation method (Метод вычисления) методы вычисления значений *дискретного* или *цифрового* гармонического сигнала при задании Sample mode: Discrete:
  - Trigonometric fcn вычисление дискретного сигнала (типа double или single) по формуле (11.6) или (11.7);
  - Table lookup вычисление дискретного или цифрового сигнала с ФТ с формированием его значений в виде таблицы, сохраняемой во внутренней памяти.

При выборе этого метода активизируется параметр **Optimize table for** (Оптимизация таблицы для) с двумя возможными значениями:

- Speed оптимизация по скорости, когда сигнал вычисляется одновременно для каждого *i*-го канала, и в памяти хранится таблица из k<sub>i</sub> отсчетов сигнала, которое должно быть целым числом, определяемым отношением k<sub>i</sub> = f<sub>д</sub>/f<sub>i</sub>, где f<sub>i</sub> частота *i*-й дискретной гармоники;
- Метогу оптимизация *памяти*, когда сигнал вычисляется одновременно для каждого *i*-го канала, и в памяти хранится таблица из k<sub>i</sub>/4 отсчетов сигнала, которое должно быть *целым числом*, определяемым отношением k<sub>i</sub> = f<sub>д</sub>/4f<sub>i</sub> > 2.

Для снятия ограничений на значения частот при формировании цифрового сигнала с ФТ можно сформировать дискретный сигнал (типа double) с помощью блока Sine Wave и подключить к его выходу блок Data Type Conversion *(см. разд. 7.2.2)* для преобразования данных к типу fixdt;

- Differential вычисление дискретного сигнала (типа double или single) с использованием алгоритма, определяющего значение текущего отсчета по его предыдущим отсчетам;
- □ Sample time см. разд. 4.2.1;
- □ Samples per frame (Количество отсчетов во фрейме) длина фрейма;
- Resetting states when re-enabled (Сброс состояния при повторном разрешении) — параметр используется в том случае, если блок находится в составе подсистемы Enable Subsystem, и для блока Enable выбрано значение параметра

States when enabling: reset (см. разд. 8.3.1); в противном случае параметр игнорируется.

Состояние при повторной активизации подсистемы Enable Subsystem определяется значениями:

- Restart at time zero сброс в нулевое состояние;
- Catch up to simulation time сохранение состояния на текущий момент моделирования.

На вкладке **Data Types** задается один параметр **Output data type** с возможными значениями:

□ Double, Single, Inherit via back propagation — см. табл. 4.5;

- □ Fixed-point для сигналов с ФТ с указанием длины слова и его дробной части;
- □ User-defined для выбора типа сигналов пользователем.

Для данных с  $\Phi$ T следует выбрать метод **Compilation method:** Table lookup.

Source Block Parameters: Sine Wave	X		
Sine Wave (mask) (link) Output samples of a sinusoid. To generate more t simultaneously, enter a vector of values for the A Frequency, and Phase offset parameters.	than one sinusoid mplitude,		
Amplitude:	Sin_Way *		
1	Eile Edit View Sin	nulation Format To	ols Help
Frequency (Hz):		X B RIG	
0.5		00 -00 400 4-	Sin Way/Vector Scope
Phase offset (rad):			File Axes Channels Window Help
0			Die Bies Biernes Winnen Deb
Sample mode: Discrete		Time	2
Output complexity: Real	Sine Wave	Vector	
Computation method: Trigonometric fcn		Scope	
Sample time:			
1/8			
Samples per frame:			
1			-1 8 8 8 8 8 8
Resetting states when re-enabled: Restart at time			-2 0 2 4 6 8 10 Frame: 82 Time (s)
<u>QK</u>	Ready	100%	FixedStepDiscrete

Рис. 11.21. Параметры блока Sine Wave и пример генерирования дискретного вещественного гармонического сигнала

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Signal Processing Sinks.

# 11.2.3. Представление дискретных сигналов

Дискретный сигнал<sup>1</sup> на выходе блока из *группы блоков Signal Processing Sources* может быть представлен в одном из следующих *четырех* видов:

□ одноканальная последовательность — Sample-based Channel Signal;

□ многоканальная последовательность — Sample-based Multichannel Signal;

□ одноканальная последовательность фреймов — Frame-based Channel Signal;

многоканальная последовательность фреймов — Frame-based Multichannel Signal.

Вид последовательности определяет специфику ее обработки и зависит от двух факторов:

□ размерности численных параметров блока (скаляр, вектор);

□ длины фрейма — значения параметра Samples per frame.

## 11.2.3.1. Одноканальная последовательность

Одноканальная последовательность (Sample-based Channel Signal) представляется в виде *трехмерного массива*  $1 \times 1 \times L$  (3-D) (L матриц  $1 \times 1$ ), где  $1 \times 1$  — количество каналов (один), а L — количество отсчетов (длина) последовательности в канале.

Одноканальная последовательность формируется в том случае, если численные параметры блока заданы *скалярами* и длина фрейма  $L_{fr} = 1$  — Samples per frame: 1

В процессе моделирования одноканальная последовательность — трехмерный массив  $1 \times 1 \times L$  — обрабатывается *по матрицам* — на каждом шаге один отсчет в единственном канале (матрице  $1 \times 1$ ).

## Пример 11.13

Сформируем одноканальную последовательность на выходе блока Sine Wave *(см. разд. 11.2.2.8)* с параметрами: Amplitude: 1; Frequency (Hz): 0.1; Sample mode: Discrete; Sample time: 1; Samples per period: 1 (рис. 11.22).

Paзмерность экспортируемой в Workspace последовательности: >> size(Sample\_1) ans = 1 1 51

Одноканальная последовательность представлена в виде *трехмерного массива* размером  $1 \times 1 \times L = 1 \times 1 \times 51$  (3-D) (51 матрица размером  $1 \times 1$ ), где  $1 \times 1$  — количество каналов (один), а L = 51 — длина последовательности. При времени моделирования 50 (с) и периоде дискретизации 1 (с) одноканальная последовательность генерируется в моменты времени 0:50 (с) — на каждом один отсчет в единственном канале (матрице  $1 \times 1$ ).

<sup>&</sup>lt;sup>1</sup> Формируется при выборе режима дискретизации Sample mode: Discrete.



Рис. 11.22. Формирование одноканальной последовательности (Sample-based Channel Signal) на выходе блока Sine Wave

#### 11.2.3.2. Многоканальная последовательность

Многоканальная последовательность (Sample-based Multichannel Signal) представляется в виде *трехмерного массива*  $1 \times N \times L$  (3-D) (L матриц  $1 \times N$ ), где  $1 \times N$ —количество каналов, N— количество последовательностей, а L— количество отсчетов последовательности в каждом канале.

Многоканальная последовательность формируется в том случае, если численные параметры блока заданы *векторами длины* N (матрицами  $1 \times N$ ) и длина фрейма  $L_{fr} = 1$  — Samples per frame: 1.

В процессе моделирования многоканальная последовательность — трехмерный массив  $1 \times N \times L$  — обрабатывается *по матрицам* — на каждом шаге одна матрица  $1 \times N$  ( $1 \times N$  отсчетов в  $1 \times N$  каналах).

#### Пример 11.14

Сформируем многоканальную последовательность на выходе блока Sine Wave *(см. разд. 11.2.2.8)* с параметром **Amplitude:** [1 1.5] и остальными параметрами, такими же, как в примере 11.13 (рис. 11.23).

Pasmephoctb экспортируемой в Workspace последовательности:
>> size(Sample\_N)

ans = 1

2 51

Многоканальная последовательность представлена в виде *трехмерного массива*  $1 \times N \times L = 1 \times 2 \times 51$  (3-D) (L = 51 матриц  $* N = 1 \times 2$ ), где  $* N = 1 \times 2$  — количество каналов, N = 2 — количество последовательностей, а L = 51 — длина последовательности в каждом канале. При времени моделирования 50 (c) и периоде дискретизации 1 (c) многоканальная последовательность генерируется в моменты времени 0:50 (c) — на каждом шаге одна матрица  $1 \times N = 1 \times 2$ .



Рис. 11.23. Формирование многоканальной последовательности (Sample-based Multichannel Signal) на выходе блока Sine Wave

## 11.2.3.3. Одноканальная последовательность фреймов

Одноканальная последовательность фреймов (Frame-based Channel Signal) представляется в виде *матрицы*  $L \times 1$  (2-D) (вектора-столбца), где L — длина последовательности фреймов, определяемая соотношением (11.1), а 1 — количество каналов (один).

Одноканальная последовательность фреймов формируется в том случае, если численные параметры блока заданы *скалярами*, и длина фрейма  $L_{fr} > 1$  (параметр Sam-

#### ples per frame).

В процессе моделирования одноканальная последовательность фреймов — матрица  $L \times 1$  — обрабатывается по фреймам — на каждом периоде фрейма  $T_{fr}$  один фрейм в одном канале, где значение  $T_{fr}$  определяется автоматически произведением

$$T_{fr} = \text{Sample time} \times L_{fr}.$$
(11.8)

#### Пример 11.15

Сформируем одноканальную последовательность фреймов на выходе блока Sine Wave *(см. разд. 11.2.2.8)* с параметром **Samples per frame:** 5 ( $L_{fr} = 5$ ) и остальными параметрами, такими же, как в примере 11.13 (рис. 11.24).

Размерность экспортируемой в Workspace многоканальной последовательности Frame\_1: >> size(Frame 1)

```
>> size(Frame_1
ans =
55 1
```
Одноканальная последовательность фреймов представлена в виде матрицы размером  $L \times 1 = 55 \times 1$  (2-D), где L — длина последовательности фреймов на интервале моделирования, а 1 — количество каналов (один). При времени моделирования 50 (с) и длине фрейма  $L_{fr} = 5$  длина L, согласно (11.1), равна  $L = K_{fr} \times L_{fr} = 5 \times 11 = 55$ , где количество фреймов  $K_{fr}$  выведено в нижней строке окна **sine\_frame/Vector Scope**: Frame = 11 (см. рис. 11.24). *Период фрейма*  $T_{fr}$ , согласно (11.8), равен  $T_{fr} =$  **Sample time**  $\times L_{fr} = 1 \times 5 = 5$  (с). На периоде фрейма одноканальная последовательность фреймов генерируется в моменты времени 0:50 (с) с периодом 5 (с) в одном канале — 5 отсчетов на периоде дискретизации **Sample time** = 1 (с).



Рис. 11.24. Формирование одноканальной последовательности фреймов (Frame-based Channel Signal) на выходе блока Sine Wave

#### 11.2.3.4. Многоканальная последовательность фреймов

Многоканальная последовательность фреймов (Frame-based Multichannel Signal) представляется в виде *матрицы*  $L \times N$  (2-D), где L — длина последовательности фреймов, определяемая соотношением (11.1), а N — количество каналов для N последовательностей фреймов длины L.

Многоканальная последовательность фреймов формируется в том случае, если численные параметры блока заданы *векторами*, и длина фрейма  $L_{fr} > 1$  (параметр

#### Samples per frame).

В процессе моделирования многоканальная последовательность фреймов *матрица*  $L \times N$  — обрабатывается *по фреймам* одновременно в N *каналах* — на каждом периоде фрейма  $T_{fr}$  (11.8) матрица  $1 \times N$  (вектор-строка).

#### Пример 11.16

Сформируем многоканальную последовательность фреймов на выходе блока Sine Wave (см. разд. 11.2.2.8) с параметрами: Amplitude: [1 1.5]; Samples per frame: 5 ( $L_{fr}$  = 5); остальные параметры, такие же, как в примере 11.13 (рис. 11.25).

Размерность экспортируемой в Workspace многоканальной последовательности Frame N:

>> size(Frame N) ans = 2

```
55
```

Многоканальная последовательность фреймов представлена в виде матрицы  $L \times 2 = 55 \times 2$  (2-D), где L — длина последовательности фреймов на интервале моделирования, а 2 — количество каналов (два). При времени моделирования 50 (с) и длине фрейма  $L_{fr} = 5$  длина L, согласно (11.1), равна  $L = K_{fr} \times L_{fr} = 5 \times 11 = 55$ , где количество фреймов  $K_{fr}$  выведено в нижней строке окна sine\_frame/Vector Scope: Frame = 11 (см. рис. 11.25). Период фрейма  $T_{fr}$ , согласно (11.8), равен  $T_{fr}$  = Sample time  $\times L_{fr} = 1 \times 5 = 5$  (c). На периоде фрейма многоканальная последовательность фреймов генерируется в моменты времени 0:50 (с) с периодом 5 (с) в каждом канале.



Рис. 11.25. Формирование многоканальной последовательности фреймов (Frame-based Multichannel Signal) на выходе блока Sine Wave

### 11.3. Средства управления дискретными сигналами

Дополнительно к рассмотренным в гл. 7 средствам управления сигналами в табл. 11.10 приводятся S-модели средств управления дискретными сигналами из библиотеки блоков Signal Management (Средства управления сигнала).

Nº	Блок	Группа блоков	Назначение
1.	Buffer (mask)	Buffers	Преобразует длину фрейма
2.	Unbuffer (mask)	Buffers	Преобразует одноканальную или многоканальную последовательность фреймов в одноканальную или многоканальную последовательность соответственно
3.	Flip	Indexing	Выполняет перестановку столбцов или строк матрицы в обратном порядке
4.	Convert 1-D to 2-D (mask)	Signal Attributes	Преобразует вектор (1-D) в матрицу (2-D)
5.	Convert 2-D to 1-D (mask)	Signal Attributes	Преобразует матрицу (2-D) в вектор (1-D)
6.	N-Sample Enable (mask)	Switches and Counters	Генерирует одноканальную последовательность данных логического типа: N единиц (нулей) вначале с последующими нулями (единицами)
7.	N-Sample Switch (mask)	Switches and Counters	Выполняет переключение между двумя входными сигналами после N периодов дискретизации
8.	Data Type Conversion	Signal Attributes	Преобразует тип данных входного сигнала <sup>1</sup>
9.	Owerwrite Values (mask)	Indexing	Заменяет подматрицу или главную диагональ в матрице
10.	Selector	Indexing	Выделяет:
			<ul> <li>элементы вектора и переупорядочивает их в выходном векторе;</li> </ul>
			<ul> <li>элементы строк и столбцов матрицы и пере- упорядочивает их в выходной матрице;</li> </ul>
			<ul> <li>сигналы в сигнале (2-D) L×N и переупорядо- чивает их в выходном сигнале (2-D);</li> </ul>
			<ul> <li>сигналы в сигнале (3-D) М×N×Lи переупор я- дочивает их в выходном сигнале (3-D)</li> </ul>
11.	Submatrix (mask)	Indexing	Выделяет подматрицу из матрицы

Таблица 11.10. Основные блоки из библиотеки блоков Signal Management

# 11.3.1. Блок Buffer

Блок *Buffer* преобразует длину фрейма и содержит следующие параметры: □ **Output buffer size (per channel)** (Размер буфера на выходе (на один канал)) — длина фрейма на выходе *L*<sub>fr вых</sub>.

Последовательность фреймов на выходе генерируется с задержкой

 $T_{\mathit{fr}\,\mathsf{bbix}} = T \cdot \max\left\{L_{\mathit{fr}\,\mathsf{bbix}}; L_{\mathit{fr}\,\mathsf{bx}}\right\},$ 

<sup>&</sup>lt;sup>1</sup> С остальными блоками преобразования данных группы Signal Attributes рекомендуем познакомиться самостоятельно.

где T — период дискретизации (параметр **Sample time** блока на входе), а  $L_{fr \text{ вх}}$  — длина фрейма на входе, что соответствует добавлению начальных max { $L_{fr \text{ вых}}$ ;  $L_{fr \text{ вх}}$ } нулей (по умолчанию);

□ **Buffer overlap** (Перекрытие буфера) — длина перекрытия *L<sub>copy</sub>*, задаваемая в пределах:

$$0 \le L_{copv} < L_{fr \text{ Bbix}} , \qquad (11.9)$$

где  $L_{conv} = 0$  соответствует отсутствию перекрытия.

292

Для значений  $L_{copy} = i$ ,  $i = 1, 2, ..., (L_{fr Bbix} - 1)$  формирование фрейма длины  $L_{fr Bbix}$  с перекрытием  $L_{copy} = i$  реализуется на основе *задержанной* последовательности фреймов без перекрытия ( $L_{copy} = 0$ ) следующим образом: у каждого следующего фрейма длины  $L_{fr Bbix}$ , начиная со второго, *первые i* отсчетов являются копиями *последних i* отсчетов предыдущего фрейма;

□ Initial conditions (Начальные условия) — значения начальных отсчетов на интервале задержки.

На рис. 11.26 приводится пример преобразования (увеличения) длины фрейма *без перекрытия* ( $L_{copy} = 0$ ). Для блока Sine Wave (*см. разд. 11.2.2.8*) заданы параметры: **Amplitude:** 1; **Frequency (Hz):** 0.1; **Sample mode:** Discrete; **Sample time:** 1; **Samples per period:** 2 ( $L_{fr BX} = 2$ ). Для блока Buffer — параметры: **Output buffer size (per channel):** 4 ( $L_{fr BMX} = 4$ ); **Buffer overlap:** 0 ( $L_{copy} = 0$ ).



Рис. 11.26. Пример преобразования (увеличения) длины фрейма без перекрытия с помощью блока Buffer

На рис. 11.27 приводится пример того же преобразования, но с перекрытием  $L_{conv} = 1$  — для блока Buffer изменено значение **Buffer overlap:** 1. В этом случае у каждого следующего фрейма длины  $L_{fr \text{ вых}} = 4$ , начиная со второго, первый отсчет является копией последнего отсчета предыдущего фрейма.



Рис. 11.27. Пример преобразования (увеличения) длины фрейма с перекрытием с помощью блока Buffer

#### 11.3.2. Блок Unbuffer

size(yout1)

1

2

51

>> ans -

Блок Unbuffer преобразует одноканальную или многоканальную последовательность фреймов в одноканальную или многоканальную последовательность соответственно. Блок имеет один параметр Initial conditions для значения начальных отсчетов на интервале задержки (по умолчанию задаются нули).

Последовательность на выходе генерируется с задержкой  $T_{fr BX} = L_{fr BX} T$ .

На рис. 11.28 приведен пример преобразования многоканальной последовательности фреймов в многоканальную последовательность. Для блока Sine Wave (см. разд. 11.2.8) заданы параметры: Amplitude: [1 1.5]; Frequency (Hz): 0.1; Sample mode: Discrete; Sample time: 1; Samples per frame: 4 ( $L_{fr BX} = 4$ ).

Выведем размерности последовательностей, экспортируемых в Workspace: >> size(vout) ans = 52 2

293



Рис. 11.28. Пример преобразования многоканальной последовательности фреймов в многоканальную последовательность с помощью блока Unbuffer

#### 11.3.3. Блок Flip

Блок *Flip* выполняет перестановку столбцов или строк матрицы в обратном порядке и имеет один параметр **Flip along** (Переворот вдоль) с двумя значениями: Columns — для перестановки строк и Rows — для перестановки столбцов.

На рис. 11.29 приведен пример перестановки столбцов матрицы при значении параметра **Flip along:** Rows.



Рис. 11.29. Пример перестановки в обратном порядке столбцов матрицы с помощью блока Flip

#### 11.3.4. Блок Convert 1-D to 2-D

Блок *Convert 1-D to 2-D* преобразует вектор (1-D) в матрицу (2-D) и содержит следующие параметры:

- Number of output rows (Количество строк на выходе) и Number of output columns (Количество строк на выходе) — соответственно количество строк и столбцов в матрице на выходе (2-D);
- □ Frame based output (Выход, основанный на фреймах) флаг, при установке которого на выходе формируется последовательность фреймов.

На рис. 11.30 приведен пример преобразования вектора длины 4 (1-D) в матрицу 2×2 (2-D) при сбросе флага **Frame based output** — из каждой пары элементов вектора формируются столбцы матрицы.



Рис. 11.30. Пример преобразования вектора в матрицу с помощью блока Convert 1-D to 2-D

#### 11.3.5. Блок Convert 2-D to 1-D

Блок *Convert 2-D to 1-D* преобразует матрицу (2-D) в вектор (1-D) и не имеет параметров.

На рис. 11.31 приведен пример преобразования матрицьк 22 (2 -D) в вектор длины 4 (1-D) — матрица растягивается в вектор.



Рис. 11.31. Пример преобразования матрицы в вектор с помощью блока Convert 2-D to 1-D

#### 11.3.6. Блок N-Sample Enable

Блок *N-Sample Enable* генерирует одноканальную последовательность данных логического типа: *N* единиц (нулей) вначале с последующими нулями (единицами). Для блока определяются следующие параметры (рис. 11.32):

- □ **Trigger count**, N (Триггерный счетчик, N) количество *N* начальных отсчетов;
- □ Active level (Активный уровень) значения *N* начальных отсчетов: Low (0) ноль или High (1) единица;
- Reset input (Вход сброса) флаг, при установке которого активизируется параметр Trigger type и появляется дополнительный вход Rst для сигнала, управляющего сбросом последовательности в начальное состояние, когда значения всех отсчетов совпадают с N начальными;
- □ **Trigger type** *см. разд. 8.3.2*;
- □ Sample time *см. разд. 4.2.1*;
- **Оиtput data type** тип данных на выходе: Logical или Boolean.

Независимо от выбора Logical или Boolean, данные на выходе относятся к логическому типу boolean, но при выборе Logical во внутренних вычислениях константы логического типа будут представляться численными константами с двойной точностью (double).



Рис. 11.32. Параметры блока N-Sample Enable и пример генерирования одноканальной последовательности данных логического типа

#### 11.3.7. Блок N-Sample Switch

Блок *N-Sample Switch* выполняет переключение между двумя входными сигналами после N периодов дискретизации и содержит следующие параметры (рис. 11.33):

□ Switch count, N (Счетчик переключателя, N) — количество *N* периодов дискретизации *T*, после которого происходит переключение сигналов.

На интервале времени [0; *NT*] на выход блока N-Sample Switch поступает сигнал с верхнего входа, после чего — с нижнего входа;

- Reset input (Вход сброса) флаг, при установке которого активизируется параметр Trigger type и появляется дополнительный вход Rst для сигнала, управляющего сбросом последовательности в начальное состояние, когда на выход поступает сигнал с верхнего входа;
- □ Sample time см. разд. 4.2.1.



Рис. 11.33. Параметры блока N-Sample Switch и пример переключения между двумя входными сигналами

#### 11.3.8. Блок Data Type Conversion

Блок *Data Type Conversion* дублирует одноименный блок в группе Signal Attributes библиотеки блоков Simulink (см. разд. 7.2.2).

#### 11.3.9. Блок Owerwrite Values

Блок Owerwrite Values дублирует одноименный блок в группе Math Functions  $\rightarrow$  Matrices and Linear Algebra  $\rightarrow$  Matrix Operations библиотеки блоков Signal Processing Blockset (*см. разд. 6.2.3.2*).

#### 11.3.10. Блок Selector

Блок Selector дублирует одноименный блок в группе Signal Routing библиотеки блоков Simulink (см. разд. 7.1.9).

#### 11.3.11. Блок Submatrix

Блок Submatrix дублирует одноименный блок в группе Math Functions  $\rightarrow$  Matrices and Linear Algebra  $\rightarrow$  Matrix Operations библиотеки блоков Signal Processing Blockset (см. разд. 6.2.3.4).

#### 11.4. Операции с дискретными сигналами

S-модели операций с дискретными сигналами представлены в группе блоков Signal Management → Signal Operations (Операции с сигналами), основные из которых приведены в табл. 11.11.

N⁰	Блок	Назначение
1.	Constant Ramp (mask)	Генерирует линейно изменяющуюся периодическую последовательность
2.	Convolution	Вычисляет свертку последовательностей на каждом шаге моделирования
3.	Downsample (mask)	Увеличивает период дискретизации в целое число раз
4.	Pad	Дополняет последовательность одинаковыми отсчетами заданного значения или усекает ее до заданной длины
5.	Peak Finder	Определяет локальные экстремумы в векторе
6.	Repeat (mask)	Уменьшает период дискретизации в целое число раз путем дублирования отсчетов входной последовательности
7.	Upsample (mask)	Уменьшает период дискретизации в целое число раз путем вставки промежуточных нулевых отсчетов
8.	Window Function	Генерирует весовую функцию (окно)
9.	Zero Crossing	Фиксирует пересечения сигналом нулевого уровня

Таблица 11.11. Основные блоки из группы блоков Signal Operations

#### 11.4.1. Блок Constant Ramp

Блок *Constant Ramp* генерирует линейно изменяющуюся периодическую последовательность:

$$y(nT_{\text{Bbix}}) = anT_{\text{Bbix}} + b, \ n = 0: (L-1),$$

где:

*T*<sub>вых</sub> = *T*/*L* — период дискретизации выходного сигнала;

*T* — период выходной последовательности, равный *периоду дискретизации* входного сигнала;

L — количество отсчетов на периоде T.

Параметры блока задаются на двух вкладках — Main и Data Types.

На вкладке Main определяются параметры (рис. 11.34):

- □ Ramp length equals number of (Длина линейно изменяющейся последовательности равна количеству) варианты задания количества отсчетов *L* на периоде *T*:
  - Rows количество строк матрицы или длина вектора;
  - Columns количество столбцов матрицы или длина вектора-столбца;
  - Elements in specified dimension (Элементы в определяемой размерности) количество элементов для размерности многомерного массива, указываемое в поле **Dimension**;

□ **Slope** (Наклон) — коэффициент *a*;

□ Offset (Смещение) — начальное значение последовательности *b*;

На вкладке **Data Types** задается один параметр **Output data type** — тип данных на выходе (см. табл. 4.5).

На рис. 11.34 приведен пример линейно изменяющейся периодической последовательности с периодом T = 2, равным периоду дискретизации входного сигнала — значению параметра **Sample time:** 2 блока Constant, и количеством отсчетов на периоде L = 5, равным длине вектора на выходе данного блока.



Рис. 11.34. Параметры блока Constant Ramp и пример линейно изменяющейся периодической последовательности

### 11.4.2. Блок Convolution

Блок *Convolution* вычисляет свертку последовательностей на каждом шаге моделирования. Блок имеет два входа для сворачиваемых последовательностей.

Параметры блока задаются на двух вкладках — Main и Fixed-point.

На вкладке **Main** имеется один параметр **Computation Domain** (Область вычисления) — область вычисления свертки: Time — временная, Frequency — частотная, Fastest — наискорейшая (здесь минимизируется объем вычислений).

На вкладке **Data type attributes** параметры подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2).

На рис. 11.35 приведен пример вычисления свертки векторов (соответствует функции MATLAB conv).



Рис. 11.35. Пример вычисления свертки векторов с помощью блока Convolution

#### 11.4.3. Блок Downsample

Блок *Downsample* увеличивает период дискретизации в целое число раз и содержит следующие параметры (рис. 11.36):

□ Downsample factor, К (Коэффициент уменьшения частоты дискретизации, К) — коэффициент К увеличения периода дискретизации (целое число).

Увеличение периода дискретизации достигается путем удаления (K – 1) промежуточных отсчетов входной последовательности на каждом интервале К дискретного нормированного времени; например, при K = 2 удаляется один промежуточный отсчет, при K = 3 — два и т. д.;

- □ Sample offset (0 to K-1) (Смещение частоты дискретизации (от 0 до K 1)) задержка выходной последовательности на *m* периодов дискретизации, где  $0 \le m \le K - 1$ ;
- □ Initial condition (Начальное условие) значение первых *m* отсчетов выходной последовательности;
- □ Sample-based mode (Режим для последовательности) варианты формирования выходной последовательности:
  - Allow multirate (Разрешить многоскоростной режим) период дискретизации выходной последовательности увеличивается в К раз;
  - Force single rate (Принудить к сохранению односкоростного режима) период дискретизации выходной последовательности не меняется и она форми-

руется посредством дублирования (K – 1) раз отсчетов, начиная с первого и далее каждого (K + 1)-го; например при K = 2 один раз будут дублироваться отсчеты 1, 3, 5 и т. д.;

- Frame-based mode (Режим для последовательности фреймов) варианты формирования выходной последовательности фреймов:
  - Maintain input frame size (Сохранить длину входного фрейма) период дискретизации выходной последовательности фреймов увеличивается в К раз при неизменной длине фрейма;
  - Maintain input frame rate (Сохранить скорость входного фрейма) период дискретизации последовательности на выходе не меняется при уменьшении длины фрейма в К раз.

На рис. 11.36 приведен пример увеличения в два раза периода дискретизации на выходе блока Sine Wave *(см. разд. 11.2.2.8)* с параметрами: **Amplitude:** 1; **Frequency (Hz):** 0.05; **Sample mode:** Discrete; **Sample time:** 1; **Samples per frame:** 1.



Рис. 11.36. Параметры блока Downsample и пример увеличения периода дискретизации в два раза

#### 11.4.4. Блок Pad

Блок *Pad* дополняет последовательность одинаковыми отсчетами заданного значения или усекает ее до заданной длины и содержит следующие параметры (рис. 11.37):

- Pad over (Дополнение к) варианты указания размерности последовательности: Columns — столбцы, Rows — строки, Columns and Rows — столбцы и строки и Specified dimensions — указывается пользователем в виде скаляра для количества столбцов, двухэлементного вектора — количества столбцов и строк и трехэлементный вектор — количества элементов трехмерного массива;
- □ Pad value source (Источник значения при дополнении) источники значения для дополняемых отсчетов:
  - Specify via dialog значение вводится в поле Pad value;
  - Input port значение подается на создаваемый нижний вход Pval;
- □ Pad value (Значение дополняемых отсчетов) значение дополняемых отсчетов;
- □ Specify (Определить) активизируется при выборе Pad over: Specified dimensions и формирует длину последовательности посредством:
  - Output size (Размер на выходе) ее непосредственного указания;
  - Pad size (Размер дополнения) указания длин дополнения в начале и в конце последовательности;
- Output column mode (Режим для столбца) способы формирования длины столбца на выходе: User-specified — указывается пользователем, Next power of two — определяется как ближайшая степень двойки;
- □ Pad signal at (Добавить сигнал в) варианты дополнения отсчетами последовательности: End в конце, Beginning в начале или Beginning and end в начале и в конце;
- □ Action then traction occurs (Действие при выполнении усечения) сообщения при усечении последовательности: None не выдаются, Warning выдается предупреждение, Error выдается сообщение об ошибке.

*При добавлении отсчетов к последовательности* рекомендуется учитывать следующее:

- □ в качестве источников последовательности использовать блоки, содержащие параметр Samples per frame;
- □ в качестве средств анализа последовательностей использовать блок Vector Scope *(см. разд. 11.2.1.6)*;
- время моделирования, отсчитываемое от нуля до значения Simulation time stop, которое устанавливается в окне S-моделей, рекомендуется выбирать таким, чтобы на нем укладывалось *целое* количество фреймов; в противном случае в окне Command Window будет выдано предупреждение, а время моделирования автоматически сокращено;
- целое количество фреймов гарантируется при установке значения Simulation time stop с учетом соотношения (11.3), а именно:

Simulation time stop = 
$$(K_{fr} - 1)TL_{fr}$$
; (11.10)

□ при добавлении отсчетов к *одноканальной* или *многоканальной* последовательности  $(K_{fr} = 1)$  параметру **Samples per frame** блока-источника присваивается значение,

равное длине последовательности L<sub>fr</sub> (один фрейм длины L<sub>fr</sub>), поэтому, согласно

(11.10), значение Simulation time stop устанавливается равным нулю;

 $\Box$  для последовательности фреймов отсчеты добавляются к каждому фрейму, поэтому параметру **Samples per frame** блока-источника присваивается значение, равное длине фрейма  $L_{fr}$ , а значение **Simulation time stop** (11.10) устанавливается

в зависимости от количества фреймов K<sub>fr</sub> и периода дискретизации T;

с учетом добавленных отсчетов в новой последовательности период дискретизации автоматически сокращается.

На рис. 11.37 приведен пример добавления нулей к последовательности фреймов на выходе блока Sine Wave (см. разд. 11.2.2.8) с параметрами: Amplitude: 1; Frequency: 40; Sample time: 1/1000; Samples per frame: 25 (длина фрейма  $L_{fr} = 25$ ). Для трех фреймов ( $K_{fr} = 3$ ) в окне S-моделей, согласно (11.10), установлено Simulation time stop =  $(3-1) \cdot 0.001 \cdot 25 = 0.05$  (с).



Рис. 11.37. Параметры блока Раd и пример добавления нулей к последовательности фреймов

# 11.4.5. Блок Peak Finder

Блок *Peak Finder* определяет локальные экстремумы в векторе. Параметры блока задаются на двух вкладках — **Main** и **Fixed-point** (рис. 11.38).

На вкладке Main определяются параметры:

Peak type(s) (Тип пика (пиков)) — типы локального экстремума: Maxima — локальные максимумы, Minima — локальные минимумы, Maxima and Minima — локальные максимумы и минимумы.

Количество локальных экстремумов выводится на выходе Cnt.

При выборе **Peak type(s):** Maxima and Minima на выходе Pol выводится *признак* 1 или 0, соответственно для локального максимума или минимума;

- □ **Output peaks indices** (Вывести индексы пиков) флаг, при установке которого на выходе Inx выводятся *индексы* локальных экстремумов;
- □ **Output peaks values** (Вывести значения пиков) флаг, при установке которого на выходе Val выводятся *значения* локальных экстремумов;
- □ **Maximum number of peaks to find** (Максимальное количество пиков) максимальное количество выводимых локальных экстремумов;
- □ Ignore peaks within threshold of neighboring values (Игнорировать пики внутри порога соседних величин) флаг, при установке которого не будут выводиться локальные экстремумы по соседству с порогом, задаваемым параметром Threshold (Порог).

На вкладке **Fixed-point** параметры подобны параметрам блока Cumulative Product (см. разд. 6.2.1.2) на вкладке **Data type attributes**.

На рис. 11.38 приведен пример определения четырех локальных экстремумов (максимумов и минимумов), а на рис. 11.39 — пример определения экстремумов при заданном пороге **Threshold:** 7, когда не выводятся локальные максимум и минимум вблизи этого значения.



Рис. 11.38. Параметры блока Peak Finder и пример определения локальных экстремумов



Рис. 11.39. Пример определения экстремумов с помощью блока Peak Finder при заданном пороге

### 11.4.6. Блок Repeat

Блок *Repeat* уменьшает период дискретизации в целое число раз путем дублирования отсчетов входной последовательности и содержит параметры:

□ **Repetition count** (Количество повторений) — коэффициент L уменьшения периода дискретизации (целое число).

Уменьшение периода дискретизации достигается путем *дублирования* L раз отсчета входной последовательности на периоде дискретизации. Выходная последовательность формируется с задержкой дискретного нормированного времени L;

Initial condition (Начальное условие) — значение первых L отсчетов выходной последовательности;

#### □ Frame-based mode — см. разд. 11.4.3.

На рис. 11.40 приведен пример уменьшения периода дискретизации в два раза (L = 2) на выходе блока Sine Wave *(см. разд. 11.2.2.8)* с параметрами: Amplitude: 5; Frequency (Hz): 0.05; Sample time: 2; Frame size: 1. Начальные значения первых двух отсчетов выходной последовательности равны нулю (Initial condition: 0).



Рис. 11.40. Пример уменьшения периода дискретизации в два раза с помощью блока Repeat

#### 11.4.7. Блок Upsample

Блок Upsample уменьшает период дискретизации в целое число раз путем вставки промежуточных нулевых отсчетов. Параметры подобны параметрам блока Downsample (см. разд. 11.4.3) с той разницей, что блок Downsample увеличивает, а блок Upsample уменьшает период дискретизации.

Уменьшение периода дискретизации достигается путем *вставки* (K - 1) промежуточных нулевых отсчетов между соседними отсчетами входной последовательности; например, при K = 2 вставляется один нулевой промежуточный отсчет, при K = 3 — два и т. д. Выходная последовательность формируется с задержкой дискретного нормированного времени K.

#### 11.4.8. Блок Window Function

Блок *Window Function* генерирует весовую функцию (окно). Параметры блока задаются на двух вкладках — Main и Fixed-point (рис. 11.41).

На вкладке Main определяются параметры:

**Орегаtion** (Операция) — операции с окном:

- Apply window to input (Применить окно к входному сигналу) применить окно к последовательности;
- Generate window (Генерировать окно) генерировать весовую функцию (окно) и блокировать вход блока;

- Generate and apply window (Генерировать и применить окно) применить окно к последовательности и генерировать весовую функцию (окно) на создаваемом выходе Win;
- □ Window type (Тип окна) типы окон.

Для знакомства с видом и параметрами окон удобно выбрать значение параметра **Operation:** Generate window и использовать блок Vector Scope *(см. разд. 11.2.1.6)*;

□ Sampling (Дискретизация) — способ дискретизации при вычислении ряда окон (Блэкмана, Хэмминта и др.): Symmetric — при вычислении отсчетов окна длины *N* или Periodic — при вычислении *N* + 1 отсчетов окна, из которых затем возвращается *N* отсчетов.

На вкладке **Fixed-point** для данных с ФТ типа fixdt (см. табл. 4.6) задаются параметры:

- □ Rounding mode см. параметр Integer rounding mode блока Add (см. разд. 6.1.1.1);
- □ Overflow mode *см. разд. 6.2.1.2*;
- Window (Окно) форматы для отсчетов окна:
  - Same word length as input длина слова как у входного сигнала;
  - Specify word length соответствует формату fixdt(S,N) (см. табл. 4.6);
  - Binary point scaling соответствует формату fixdt(S,N,M) (см. табл. 4.6);
  - Slope and bias scaling соответствует формату fixdt(S,N,2^(-M),Bias) (см. табл. 4.6);
- □ **Product output** (Произведение на выходе) форматы для произведения:
  - Same as input как у входного сигнала;
  - Inherit via internal rule см. табл. 4.5;
  - Остальные значения такие же, как у параметра Window;
- **Оиtput** (Выход) форматы для отсчетов на выходе:
  - Same as input как у входного сигнала;
  - Same as product output как у произведения Product output;

Остальные значения такие же, как у параметра Window;

□ Lock output data type setting against changes by fixed-point tools — *cm. pa3∂.* 4.2.4.

Для применения окна к последовательности (дискретному сигналу или цифровому сигналу с ФТ) рекомендуется учитывать следующее:

- □ в качестве источников последовательности использовать блоки, содержащие параметр Samples per frame;
- □ в качестве средств анализа последовательностей использовать блок Vector Scope *(см. разд. 11.2.1.6)*;

Function Block Parameters: Window Function	🗑 Function Block Parameters: Window Function 🛛 🛛 🔀		
Window Function Generate a window function and/or apply a window function to an i For some fixed-point modes, the fraction length or slope of the win automatically set for you to the best possible precision given the re word length of the window values. This is equivalent to the 'Best Pr wise'' scaling option used in some Simulink fixed-point blocks.	Window Function     Generate a window function and/or apply a window function to an input signal.     For some fixed-point modes, the fraction length or slope of the window values is     automatically set for you to the best possible precision given the real-world values and     word length of the window values. This is equivalent to the "Best Precision: Matrix-     wise" scaling option used in some Simulink fixed-point blocks.		
Main Fixed-point	Main Fixed-point		
Parameters Operation: Generate and apply window Window type: Hamming Sampling: Symmetric	Settings on this pane only apply when block inputs are fixed-point signals.  Fixed-point operational parameters  Rounding mode: Floor  Fixed-point data types  Mode		
	Window       Same word length as input       Image: Constraint of the second se		
OK Cancel Help	QK Cancel Help Apply		

Рис. 11.41. Параметры блока Window Function

- время моделирования, отсчитываемое от нуля до значения Simulation time stop, которое устанавливается в окне S-моделей, рекомендуется выбирать таким, чтобы на нем укладывалось *целое* количество фреймов; в противном случае в окне Command Window будет выдано предупреждение, а время моделирования автоматически сокращено;
- □ целое количество фреймов гарантируется при установке значения Simulation time stop согласно (11.10);

□ для применения окна к одноканальной или многоканальной последовательности ( $K_{fr} = 1$ ) следует параметру Samples per frame блока-источника присвоить значение, равное длине последовательности  $L_{fr}$  (один фрейм длины  $L_{fr}$ ), поэтому, согласно (11.10), значение Simulation time stop, устанавливается равным нулю;

П для последовательности фреймов окно применяется последовательно к каждому фрейму, поэтому параметру Samples per frame блока-источника присваивается значение, равное длине фрейма  $L_{fr}$ , а значение Simulation time stop (11.10)

устанавливается в зависимости от количества фреймов  $K_{fr}$  и периода дискретизации T;

при неизвестном заранее количестве фреймов значение Simulation time stop можно установить равным inf (машинная бесконечность), и после запуска моделирования остановить процесс моделирования или сделать в нем паузу. На рис. 11.42 приведен пример использования окна Хэмминга для *одноканальной* последовательности на выходе блока Sine Wave *(см. разд. 11.2.2.8)* с параметрами: **Amplitude:** 1; **Frequency:** 40; **Sample time:** 1/1000; **Samples per frame:** 40 (длина последовательности — 0:39). В окне S-моделей установлено **Simulation time stop** = 0. Параметры блока Window Function представлены на рис. 11.41.



Рис. 11.42. Пример применения окна Хэмминга для одноканальной последовательности с помощью блока Window Function

#### 11.4.9. Блок Zero Crossing

Блок Zero Crossing фиксирует пересечения сигналом нулевого уровня (нуля) и не имеет параметров.

В процессе моделирования на выходе блока Zero Crossing формируется последовательность единиц (признак изменения знака — пересечения нуля) и нулей (признак неизменности знака). Для подсчета количества пересечений нуля на выходе блока Zero Crossing следует поставить счетчик — блок Cumulative Sum (см. разд. 6.2.1.3).

На рис. 11.43 приведен пример подсчета количества пересечений нуля, фиксируемых с помощью блока Zero Crossing.



Рис. 11.43. Пример подсчета количества пересечений нуля, фиксируемых с помощью блока Zero Crossing

# 11.5. Генерирование типовых последовательностей

Для генерирования *типовых последовательностей* [22] в Simulink могут использоваться следующие блоки:

□ Discrete Impulse (см. разд. 11.2.2.4) — для цифрового единичного импульса:

$$u_0(nT) = \begin{cases} 1, \ n = 0; \\ 0, \ n \neq 0; \end{cases}$$
(11.11)

□ Step (см. разд. 4.2.20) — для цифрового единичного скачка:

$$u_1(nT) = \begin{cases} 1, n \ge 0; \\ 0, n < 0; \end{cases}$$
(11.12)

□ Constant (см. разд. 4.2.4), Digital Clock (см. разд. 4.2.7) и Math Function (см. разд. 6.1.2.2) при выборе параметра Function: ром — для дискретной экспоненты:

$$x(nT) = \begin{cases} a^{nT}, \ n \ge 0; \\ 0, \ n < 0. \end{cases}$$
(11.13)

На рис. 11.44 приведен пример генерирования дискретной экспоненты  $x(n) = 0.8^n$  при установке **Sample time:** 1;



Рис. 11.44. S-модель и результат моделирования дискретной экспоненты

□ Sine Wave (см. разд. 11.2.2.8 или 4.2.19) — для дискретного гармонического сигнала, соответственно, в видах:

$$x(nT) \oplus A\sin()nT + \varphi$$
; (11.14)

$$x(nT) = \pi A \sin(2 fnT + \varphi;$$
 (11.15)

□ Sine Wave (см. разд. 11.2.2.8) — для дискретного комплексного гармонического сигнала:

$$x(nT) = Ae^{j(2\pi f nT + \varphi)}.$$
 (11.16)



# Глава 12

# Линейные дискретные системы

В этой главе рассматриваются S-модели линейных дискретных систем (ЛДС), структура которых представлена группой блоков **Discrete** библиотеки блоков Simulink.

### 12.1. Математическая модель и S-модель ЛДС

S-модель ЛДС может создаваться на основе следующих *математических моделей (см. разд. 2.1)*:

1. Разностное уравнение (РУ):

$$y(nT) = \sum_{i=0}^{N-1} b_i x[(n-i)T] - \sum_{k=1}^{M-1} a_k y[(n-k)T], \qquad (12.1)$$

где:

*b<sub>i</sub>*, *a<sub>k</sub>* — коэффициенты уравнения — параметры ЛДС;

i > 0, k > 0 — значения задержек соответственно воздействия и реакции;

(N-1), (M-1) — максимальные значения задержек соответственно воздействия и реакции.

Две разновидности РУ определяют два типа ЛДС:

- рекурсивная с РУ (12.1), в котором хотя бы один из коэффициентов *a<sub>k</sub>* не равен нулю;
- нерекурсивная с РУ (12.1), в котором все коэффициенты *a<sub>k</sub>* равны нулю:

$$y(n) = \sum_{i=0}^{N-1} b_i x[(n-i)T].$$
(12.2)

Рекурсивные ЛДС имеют бесконечную импульсную характеристику (БИХ), а нерекурсивные — конечную (КИХ), отсюда их тождественные названия БИХ и КИХ ЛДС.

Коэффициенты РУ (12.2) *нерекурсивной* ЛДС совпадают с отсчетами ее импульсной характеристики:

$$b_i = h(nT), \ n = i.$$
 (12.3)

 Система уравнений переменных состояния, включающая матричное уравнение наблюдения ЛДС и уравнение выхода ЛДС [23]:

$$\begin{cases} s[(n+1)T] = As(nT) + Bx(nT); \\ y(nT) = Cs(nT) + Dx(nT), \end{cases}$$
(12.4)

где:

s(nT) — вектор-строка переменных состояния — сигналов на выходах элементов задержки;

s[(n+1)T — вектор-столбец будущих значений переменных состояния;

А — квадратная матрица, порядок которой равен порядку ЛДС;

*B*, *C* — вектор-столбец и вектор-строка соответственно, длины которых равны порядку матрицы *A*;

D — скаляр, равный коэффициенту  $b_0$  в РУ (12.1).

Значения элементов матрицы А и векторов В, С определяются по коэффициентам РУ (12.1) [23].

Линейные уравнения (12.1)—(12.2) решаются методом прямой подстановки при нулевых начальных условиях (ННУ), поэтому они непосредственно описывают *алгоритм* вычисления реакции ЛДС при ННУ.

*Нулевые начальные условия* (ННУ) означают, что все значения воздействия и реакции, которые может помнить ЛДС в моменты времени, предшествующие началу воздействия nT = 0, равны нулю:

$$\begin{cases} x[(n-i)T]_{|(n-i)T < 0, i=1, 2, ...} = 0; \\ y[(n-k)T]_{|(n-k)T < 0, k=1, 2, ...} = 0. \end{cases}$$
(12.5)

В математической модели ЛДС можно выделить три основные составляющие (см. рис. 2.1, б):

**П** входной дискретный сигнал x(nT);

- □ структура ЛДС, которая определяется видом ее *передаточной функции* или параметрами системы уравнений переменных состояния;
- **П** выходной сигнал y(nT).

В *S-модели* ЛДС им будут соответствовать три отличающиеся своим функциональным назначением компоненты:

**П** S-модель входного сигнала x(nT);

S-модель структуры ЛДС;

 $\Box$  S-модель средств анализа выходного сигнала y(nT).

Передаточной функцией ЛДС H(z) общего вида называют дробно-рациональную функцию, соответствующую РУ (12.1) рекурсивной ЛДС:

$$H(z) = \frac{\sum_{i=0}^{N-1} b_i z^{-i}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}}.$$
(12.6)

Передаточная функция нерекурсивной ЛДС будет иметь вид рациональной функции, соответствующий РУ (12.2):

$$H(z) = \sum_{i=0}^{N-1} b_i z^{-i} = \sum_{n=0}^{N-1} h(nT) z^{-n} .$$
(12.7)

Порядком рекурсивной ЛДС называют порядок знаменателя (M-1) ее передаточной функции (12.6), т. к. на практике соблюдается условие:

$$(N-1) \le (M-1). \tag{12.8}$$

Порядком нерекурсивной ЛДС называют порядок (N-1) ее передаточной функции (12.7).

Для моделирования структур *рекурсивных* ЛДС в MATLAB и Simulink используются следующие виды передаточной функции (12.6):

• произведение простейших множителей:

$$H(z) = b_0 \prod_{k=1}^{M-1} \frac{\left(1 - z_{*k} z^{-1}\right)}{\left(1 - z_{*k} z^{-1}\right)},$$
(12.9)

где:

 $z_{\circ k}$ ,  $z_{*k}$  — соответственно *k*-е ноль и полюс<sup>1</sup> передаточной функции (12.6); в общем случае нули и полюсы — попарно комплексно сопряженные числа;

(М – 1) — количество рекурсивных звеньев 1-го порядка;

*b*<sub>0</sub> — коэффициент усиления;

□ произведение множителей второго порядка с вещественными коэффициентами:

$$H(z) = \prod_{k=1}^{L} \frac{b_{0k} + \tilde{b}_{1k} z^{-1} + \tilde{b}_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}},$$
(12.10)

где L — количество рекурсивных звеньев 2-го порядка, равное (M-1)/2 — при четном и M/2 — нечетном порядке (M-1) ЛДС.

<sup>&</sup>lt;sup>1</sup> Нулями называют значения z, при которых передаточная функция (12.6) обращается в ноль, а *полю*сами — значения z, при которых ее знаменатель обращается в ноль.

В MATLAB и Simulink передаточная функция (12.10) имеет следующее эквивалентное представление:

$$H(z) = G \prod_{k=1}^{L} \frac{1 + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}},$$
(12.11)

где G — коэффициент усиления, равный произведению коэффициентов  $b_{0k}$  в (12.10):

$$G = b_{01} \cdot b_{02} \cdot \ldots \cdot b_{0L},$$

и соответствующие коэффициенты связаны соотношениями:

$$b_{1k} = \tilde{b}_{1k} / G ;$$
  
$$b_{2k} = \tilde{b}_{2k} / G .$$

Для моделирования структур *нерекурсивных* ЛДС, рассматриваемых в этой главе, используется представление передаточной функции (12.7) в виде:

$$H(z) = b_0 \prod_{k=1}^{N-1} (1 - z_{\circ k} z^{-1}), \qquad (12.12)$$

где (N-1) — количество нерекурсивных звеньев 1-го порядка.

Структуры ЛДС (КИХ- и БИХ-фильтров), определяемые видом передаточной функции, будут подробно рассматриваться в *гл. 13*.

# 12.2. Ѕ-модели структур ЛДС

В группе **Discrete** библиотеки блоков Simulink *S-модели структур* ЛДС представлены блоками, основные из которых приведены в табл. 12.1.

Таблица 12.1. Основные блоки группы Discrete

Nº	Блок	Назначение
1.	Discrete Filter	Моделирует структуру рекурсивной или нерекурсивной ЛДС соответственно с передаточной функцией (12.6) или (12.7)
2.	Discrete FIR Filter	Моделирует структуру нерекурсивной ЛДС с передаточной функцией (12.7)
3.	Discrete State-Space	Моделирует структуру ЛДС, описываемую системой уравне- ний переменных состояния (12.4)
4.	Discrete Zero-Pole	Моделирует структуру рекурсивной или нерекурсивной ЛДС соответственно с передаточной функцией (12.9) или (12.12)

Рассмотрим параметры блоков из табл. 12.1.

### 12.2.1. Блок Discrete Filter

Блок *Discrete Filter* моделирует структуру рекурсивной или нерекурсивной ЛДС соответственно с передаточной функцией (12.6) или (12.7). Параметры блока задаются на трех вкладках — Main, Data Types и State Attributes (Атрибуты состояния).

На вкладке Main определяются параметры (рис. 12.1):

□ Numerator coefficients (Коэффициенты числителя) — вектор коэффициентов числителя  $b_i$  передаточной функции (12.6) в порядке возрастания номеров, i = 0, 1, ..., (N-1).

Коэффициенты числителя могут задаваться *матрицей* при моделировании одновременно нескольких ЛДС, передаточные функции которых отличаются числителем;

□ Denominator coefficients (Коэффициенты знаменателя) — вектор коэффициентов знаменателя *a<sub>k</sub>* передаточной функции (12.6) в порядке возрастания номеров, *k* = 0, 1, ..., (*M* − 1), где *a*<sub>0</sub> = 1.

Для нерекурсивной ЛДС задается Denominator coefficients: [1].

Для рекурсивной ЛДС, согласно (12.8), количество коэффициентов числителя передаточной функции (12.6) не должно превышать количества коэффициентов знаменателя.

По умолчанию коэффициенты числителя и знаменателя передаточной функции относятся к данным типа double;

- □ **Sample time** *см. разд. 4.2.1*;
- □ Optimize by skipping divide by leading denomination coefficients (Оптимизировать посредством деления на начальный коэффициент знаменателя) флаг, устанавливаемый при  $a_0 = 1$ ; при  $a_0 \neq 1$  флаг сбрасывается, и коэффициенты знаменателя и реакция ЛДС делятся на  $a_0$ .

На вкладке **Data Types** значения параметров выбираются из раскрывающихся списков и связаны с выбором *muna данных* (см. табл. 4.5) для внутреннего состояния ЛДС<sup>1</sup> (States), коэффициентов числителя (Numerator), знаменателя (Denomirator), результатов арифметических операций сложения и умножения, а также выходного сигнала (Output). Помимо этого задаются параметры:

- □ Integer rounding mode и Saturate on integer overflow *см. разд. 6.1.1.1*;
- □ Lock output data type setting against changes by the fixed-point tools *cm. pa3∂. 4.2.4.*

<sup>&</sup>lt;sup>1</sup> В структуре ЛДС — это значения на выходе элементов задержки.

На вкладке State Attributes параметры имеют отношение к генерации кода при работе в реальном времени и не влияют на процесс моделирования. Их рассмотрение выходит за рамки книги.

#### Пример 12.1

На рис. 12.1 представлена S-модель ЛДС, включающая рекурсивное звено 2-го порядка (блок Discrete Filter) с передаточной функцией:

$$H(z) = \frac{0.8 + 0.7z^{-1} + 0.49z^{-2}}{1 - 0.8z^{-1} + 0.64z^{-2}},$$
(12.13)

где для блока Sine Wave *(см. разд. 4.2.19)* установлены параметры: Sine type: Sample based; Amplitude: 1; Samples per period: 10; Sample time: 1.



Рис. 12.1. Параметры блока Discrete Filter и пример S-модели ЛДС с рекурсивным звеном 2-го порядка

#### 12.2.2. Блок Discrete FIR Filter

Блок *Discrete FIR Filter* моделирует структуру нерекурсивной ЛДС с передаточной функцией (12.7). Параметры блока задаются на двух вкладках — **Main** и **Data Types**.

На вкладке Main определяются параметры (рис. 12.2):

- □ **Coefficient source** (Источник коэффициентов) источники коэффициентов *b<sub>i</sub>* передаточной функции (12.7):
  - Dialog parameters (Параметры диалога) задаются параметром Numerator coefficients;
  - Input port (Входной порт) подаются извне на автоматически создаваемый вход Num;

Часть II. Моделирование цифровой обработки сигналов средствами Simulink

□ Filter structure (Структура фильтра) — структура нерекурсивной ЛДС.

По умолчанию выбирается прямая структура Direct form. Подробнее о структурах нерекурсивных ЛДС пойдет речь далее в *разд. 13.2*;

- □ Numerator coefficients (Коэффициенты числителя) вектор коэффициентов *b<sub>i</sub>* передаточной функции (12.7) в порядке возрастания номеров, *i* = 0, 1, ..., (*N* − 1). Параметр доступен при выборе значения Coefficient source: Dialog parameters;
- □ Initial state (Начальное состояние) начальные условия, нулевые по умолчанию;

□ Sample time — см. разд. 4.2.1.

На вкладке **Data Types** задаются параметры, аналогичные параметрам блока Discrete Filter (см. разд. 12.2.1).

#### Пример 12.2

На рис. 12.2 представлена S-модель ЛДС, включающая нерекурсивное звено 2-го порядка (блок Discrete FIR Filter) с передаточной функцией

$$H(z) = 1 + 0.8z^{-1} + 1.5z^{-2}$$

Для блока Sine Wave установлены такие же параметры, как в примере 12.1.



Рис. 12.2. Параметры блока Discrete FIR Filter и пример S-модели с нерекурсивным звеном 2-го порядка

#### 12.2.3. Блок Discrete State-Space

Блок *Discrete State-Space* моделирует структуру ЛДС, описываемую системой уравнений переменных состояния (12.4). Параметры блока задаются на двух вкладках — Main и State Attributes.

318

На вкладке Main определяются параметры (рис. 12.3):

□ A, B, C, D — одноименные параметры в системе уравнений (12.4).

Параметры **B**, **C**, **D** могут задаваться матрицами при моделировании одновременно нескольких ЛДС.

По умолчанию значения A, B, C, D относятся к данным типа double;

□ Initial state (Начальное состояние) — начальные условия, нулевые по умолчанию;

□ Sample time — см. разд. 4.2.1.

На вкладке State Attributes параметры имеют отношение к генерации кода при работе в реальном времени и не влияют на процесс моделирования. Их рассмотрение выходит за рамки книги.

#### Пример 12.3

На рис. 12.3 представлена S-модель ЛДС, включающая рекурсивное звено 2-го порядка (блок Discrete State-Space) с математической моделью в виде системы уравнений переменных состояния (12.4), и блок Sine Wave с такими же параметрами, как в примере 12.1.

Методика вычисления матрицы *A* и векторов *B*, *C* в (12.4) рассматривается в [23]. Однако в MATLAB их можно вычислить автоматически по коэффициентам числителя (вектор b) и знаменателя (вектор a) передаточной функции (12.13) с помощью функции tf2ss:

С учетом этого система уравнений переменных состояния (12.4) примет вид:

$$\begin{cases} s_1[(n+1)T] = 0, 8s_1(nT) - 0, 64s_2(nT) + x(nT); \\ s_2[(n+1)T] = s_1(nT); \\ y(nT) = 1, 34s_1(nT) - 0, 022s_2(nT) + 0, 8x(nT). \end{cases}$$
(12.14)



Рис. 12.3. Параметры блока Discrete State-Space и пример S-модели ЛДС с рекурсивным звеном 2-го порядка и математической моделью в виде системы уравнений переменных состояния

#### 12.2.4. Блок Discrete Zero-Pole

Блок *Discrete Zero-Pole* моделирует структуру рекурсивной или нерекурсивной ЛДС соответственно с передаточной функцией (12.9) или (12.12). Параметры блока задаются на двух вкладках — **Main** и **State Attributes**.

На вкладке Main определяются параметры (рис. 12.4):

□ Zeros (Нули) — вектор нулей *z*<sub>∘*k*</sub> передаточной функции (12.9), задаваемых комплексно сопряженными парами.

Нули могут задаваться *матрицей* при моделировании одновременно нескольких ЛДС, передаточные функции которых отличаются числителем;

□ Poles (Полюсы) — вектор полюсов *z*<sub>\*k</sub> передаточной функции (12.9), задаваемых комплексно сопряженными парами.

Для нерекурсивной ЛДС с передаточной функцией (12.12) задается Poles: [0];

□ Gain (Усиление) — коэффициент усиления *b*<sub>0</sub> в (12.9).

По умолчанию нули, полюсы и коэффициент усиления относятся к данным типа double;

□ Sample time — см. разд. 4.2.1.

На вкладке State Attributes параметры имеют отношение к генерации кода при работе в реальном времени и не влияют на процесс моделирования. Их рассмотрение выходит за рамки книги.

#### Пример 12.4

На рис. 12.4 представлена S-модель ЛДС, включающая рекурсивное звено 2-го порядка (блок Discrete Zero-Pole) с передаточной функцией:

$$H(z) = 0.8 \cdot \frac{\left[1 - \left(-0.4375 + 0.6489j\right)z^{-1}\right]\left[1 - \left(-0.4375 - 0.6489j\right)z^{-1}\right]}{\left[1 - \left(0.4 + 0.6928j\right)z^{-1}\right]\left[1 - \left(0.4 - 0.6928j\right)z^{-1}\right]}.$$
 (12.15)

Для блока Sine Wave установлены такие же параметры, как в примере 12.1.



Рис. 12.4. Параметры блока Discrete Zero-Pole и пример S-модели ЛДС с рекурсивным звеном 2-го порядка

Передаточная функция (12.15) тождественна передаточной функции (12.13). Нули, полюсы и коэффициент усиления в (12.15) были определены по коэффициентам (12.13) с помощью функции MATLAB tf2zpk:

```
>> b=[0.8 0.7 0.49]; a=[1 -0.8 0.64];
>> [q,p,K]=tf2zpk(b,a)
q =
    -0.4375 + 0.6489i
    -0.4375 - 0.6489i
p =
        0.4000 + 0.6928i
        0.4000 - 0.6928i
K =
        0.8
```

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Simulink  $\rightarrow$  Blocks  $\rightarrow$  Discrete.

# 12.3. Анализ характеристик S-модели ЛДС

Анализ характеристик S-модели ЛДС производится с помощью программы GUI FVTool (Filter Visualization Tool — средства визуализации фильтра), входящей в состав MATLAB, обращение к которой производится с помощью функции: fvtool (b, a)

где b, а — векторы коэффициентов передаточной функции ЛДС (12.6). Для нерекурсивных ЛДС с передаточной функцией (12.2) задается a=[1].

Например, для рекурсивного звена 2-го порядка с передаточной функцией (12.13) вызов функции будет таким:

```
>> b=[0.8 0.7 0.64]; a=[1 -0.8 0.64];
>> fvtool(b,a)
```

После этого автоматически откроется окно Filter Visualization Tool, в котором можно вывести интересующие характеристики ЛДС, обращаясь к следующим командам пункта меню Analysis (Анализ):

- □ Magnitude Response (AЧX);
- **D** Phase Response ( $\Phi$ ЧX);
- □ Magnitude and Phase Responses (АЧХ и ФЧХ);
- □ Group Delay Response (ГВЗ групповое время задержки);
- □ Phase Delay (Фазовая задержка);
- □ Impulse Response (Импульсная характеристика);
- **Step Response** (Переходная характеристика);
- □ Pole/Zero Plot (Карта нулей и полюсов);
- □ Filter Coefficients (Коэффициенты фильтра коэффициенты передаточной функции);
- □ Filter Information (Информация о фильтре);
- □ Magnitude Response Estimate (Оценка АЧХ для ЦФ с ФТ<sup>1</sup>);
- □ Round-off Noise Power Spectrum (Энергетический спектр шума округления для ЦФ с ФТ).

Помимо этого в пункт меню Analysis включены дополнительные команды:

- □ Filter Specifications (Требования к АЧХ) по этой команде в поле графика выводятся требования к АЧХ;
- Overlay Analysis (Наложение характеристик) по этой команде выводится список флагов, имена которых дублируют перечисленные команды анализа. Установив соответствующий флаг, можно добавить в поле графика еще одну характеристику поверх имеющейся. По умолчанию установлен флаг None (Добавляемой характеристики нет);

322

<sup>&</sup>lt;sup>1</sup> Цифровых фильтров с фиксированной точкой.

- Analysis Parameters (Параметры анализа) по этой команде открывается одноименное окно, в котором можно изменить параметры анализируемой характеристики (список параметров зависит от характеристики), а затем нажать кнопку Apply или OK. При нажатии кнопки Apply соответствующие параметры (единицы измерения по оси абсцисс, ординат и проч.) будут автоматически изменены на выведенном графике, а при нажатии кнопки OK — они будут изменены и сохранены при последующих выводах графиков;
- □ Sampling Frequency (Частота дискретизации) по этой команде открывается одноименное окно, в котором можно изменить единицы измерения частоты дискретизации, затем нажать кнопку OK.

Команды Analysis Parameters и Sampling Frequency дублируются в контекстном меню, открываемом щелчком правой кнопкой мыши на свободном поле графика.



# Глава 13

# Системы цифровой фильтрации

В этой главе рассматриваются S-модели систем цифровой фильтрации, основным компонентом которых является цифровой фильтр.

Цифровой фильтр (ЦФ) представляет собой линейную дискретную систему, выполняющую преобразование входной последовательности в выходную по алгоритму, описываемому разностным уравнением, который отображается заданной структурой (структурой ЦФ), реализованной программно или аппаратно.

В зависимости от того, являются ли параметры структуры ЦФ неизменными или меняющимися во времени, его называют стационарным или адаптивным. По умолчанию речь пойдет о *стационарных* ЦФ.

# 13.1. Проектирование цифровых фильтров

Проектирование ЦФ выполняется в три этапа:

1. Синтез ЦФ, включающий следующие основные шаги:

• 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).
- Задание требований к характеристикам ЦФ.

Требования к характеристикам ЦФ зависят от его типа (КИХ или БИХ) и назначения ЦФ (частотно-избирательный, преобразователь Гильберта и т. д.).

По умолчанию подразумевают *частотно-избирательные* ЦФ, выполняющие селекцию спектральных составляющих входной последовательности.

Выделяют четыре основные 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).
- Выбор метода синтеза.

Метод синтеза зависит от типа Ц $\Phi$  (КИХ или БИХ), а в рамках одного типа — от специфики дополнительных требований (простоты метода, оптимальности проектируемого фильтра и др.).

- Расчет передаточной функции ЦФ.
- Выбор структуры ЦФ.
- 2. Моделирование структуры ЦФ с учетом эффектов квантования.
- 3. Реализация структуры ЦФ.

Структура ЦФ (отображающая алгоритм вычисления реакции по РУ) может быть реализована программно или аппаратно, преимущественно *программно* на базе цифрового устройства — цифрового процессора обработки сигналов (ЦПОС), системы на кристалле (System on Chip) или программируемой логической интегральной схеме (ПЛИС).

# 13.2. Свойства КИХ-фильтров

КИХ-фильтр описывается передаточной функцией (12.7):

$$H(z) = \sum_{i=0}^{N-1} b_i z^{-i} = \sum_{n=0}^{N-1} h(n) z^{-n} .$$
(13.1)

Длиной и порядком КИХ-фильтра называют соответственно число коэффициентов N и порядок R передаточной функции (13.1), равный

$$R = N - 1. (13.2)$$

КИХ-фильтры характеризуются следующими особенностями:

 возможностью обеспечить строго линейную ΦЧХ (с точностью до скачков<sup>1</sup> на π);

• устойчивостью по определению.

Линейная ФЧХ (ЛФЧХ) КИХ-фильтра обеспечивается в том и только том случае, если для его импульсной характеристики (ИХ) h(n) выполняется одно из двух условий:

• симметрии:

$$h(n) = h(N-1-n);$$
 (13.3)

антисимметрии:

$$h(n) = -h(N - 1 - n), \qquad (13.4)$$

где ось симметрии/антисимметрии ИХ h(n) проходит через точку n = R/2.

<sup>&</sup>lt;sup>1</sup> Скачок ФЧХ на  $\pi$  имеет место в тех точках, где АЧХ равна нулю.

По двум признакам — симметрии/антисимметрии ИХ и нечетности/четности длины N — выделяют четыре типа КИХ-фильтров с ЛФЧХ (табл. 13.1), на базе которых синтезируется ЦФ.

Помимо частотно-избирательных, в табл. 13.1 включены два специальных КИХфильтра — цифровой преобразователь Гильберта (ЦПГ) и цифровой дифференциатор (ЦД).

Тип КИХ-фильтра	ЛФЧХ (с точностью до скачков на π)	ЦФ
<b>Тип 1</b> (Type-1): Длина <i>N</i> — нечетная Порядок <i>R</i> — четный <i>h</i> ( <i>n</i> ) — симметричная	$\varphi(\hat{\omega}) = -\frac{\hat{\omega}R}{2}$	ΦΗϤ, ΦΒϤ, ΠΦ, ΡΦ
<b>Тип 2</b> (Туре-2): Длина <i>N</i> — четная Порядок <i>R</i> — нечетный <i>h</i> ( <i>n</i> ) — симметричная	$\varphi(\hat{\omega}) = -\frac{\hat{\omega}R}{2}$	ФНЧ, ПФ
Тип 3 (Туре-3): Длина $N$ — нечетная Порядок $R$ — четный h(n) — антисимметричная $h\left(\frac{R}{2}\right) \equiv 0$	$\varphi(\hat{\omega}) = \frac{\pi}{2} - \frac{\hat{\omega}R}{2}$	ПФ ЦПГ ЦД
<b>Тип 4</b> (Type-4): Длина <i>N</i> — четная Порядок <i>R</i> — нечетный <i>h</i> ( <i>n</i> ) — антисимметричная	$\varphi(\hat{\omega}) = \frac{\pi}{2} - \frac{\hat{\omega}R}{2}$	ФВЧ, ПФ ЦПГ ЦД

Таблица 13.1. Четыре типа КИХ-фильтров с ЛФЧХ

# 13.3. Задание требований к частотным характеристикам КИХ-фильтров

Методы синтеза частотно-избирательных КИХ-фильтров изначально предполагают ЛФЧХ (с точностью до скачков на  $\pi$ ), поэтому требования задаются к нормированной АЧХ  $\hat{A}(f)$  в основной полосе частот [0;  $f_{\pi}/2$ ] и включают в себя:

 $\Box$  частоту дискретизации  $f_{\pi}$ ;

- □ *граничные частоты* полос пропускания (ПП) и полос задерживания (ПЗ), для которых введены условные обозначения:
  - *f*<sub>γ</sub> граничная частота ПП для ФНЧ и ФВЧ;
  - *f<sub>k</sub>* граничная частота ПЗ для ФНЧ и ФВЧ;
  - *f*<sub>-γ</sub>, *f*<sub>γ</sub> 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<sub>1</sub> от единицы в ПП;
- δ<sub>2</sub> от нуля в ПЗ.

На рис. 13.1 приведены примеры требований к АЧХ для фильтров различного типа избирательности.



Рис. 13.1. Требования к АЧХ: ФНЧ (а), ФВЧ (б), ПФ (в) и РФ (г)

Требования могут задаваться к АЧХ в децибелах — к характеристике ослабления:

$$\hat{A}(f)$$
 ( $\ddot{a}\dot{A}$ ) = 20lg( $\hat{A}(f)$ ) (13.5)

или к характеристике затухания:

$$\hat{A}(f)(\ddot{a}\dot{A}) = -20\lg(\hat{A}(f)).$$
 (13.6)

В MATLAB и Simulink требования задаются к характеристике затухания (13.6).

В требованиях к характеристике затухания (13.6) вместо значений  $\delta_1$  и  $\delta_2$  задаются:

□ *a*<sub>max</sub> (дБ) — максимально допустимое затухание в ПП;

□ *a*<sub>min</sub> (дБ) — минимально допустимое затухание в ПЗ.

На рис. 13.2 приведен пример требований к характеристике затухания ФНЧ.



Рис. 13.2. Требования к характеристике затухания ФНЧ

Взаимосвязь между значениями  $\delta_1$ ,  $\delta_2$  и  $a_{max}$ ,  $a_{min}$  (дБ) соответственно устанавливается формулами:

$$a_{\max} = -20 \, \lg(1 - \delta_1) \, (\text{gB});$$
 (13.7)

$$a_{\min} = -20 \lg(\delta_2) \ (\text{gb}),$$
 (13.8)

и наоборот:

$$\delta_1 = 1 - 10^{-a_{\max}/20}; \tag{13.9}$$

$$\delta_2 = 10^{-a_{\min}/20}.$$
 (13.10)

#### Пример 13.1

0.0100

Рассчитаем в MATLAB значения  $a_{max}$  и  $a_{min}$  (amax и amin) по заданным для ФНЧ значениям  $\delta_1 = 0,05$  и  $\delta_2 = 0,01$  (d1 и d2). Решим обратную задачу.

Расчет *a*<sub>max</sub>, *a*<sub>min</sub> по формулам (13.7)—(13.8):

```
>> d1=0.05;
>> amax=-20.*log10(1-d1)
amax =
     0.4455
>> d2=0.01;
>> amin=-20.*log10(d2)
amin =
     40
Получили a<sub>max</sub> = 0,4455 дБ и a<sub>min</sub> = 40 дБ.
     Обратную задачу — расчет δ<sub>1</sub> и δ<sub>2</sub> — выполним по формулам (13.9)—(13.10):
>> d1=10.^(-amin/20)
d1 =
```

```
328
```

Получили исходные значения  $\delta_1$  и  $\delta_2$  (d1 и d2).

## 13.4. Свойства БИХ-фильтров

БИХ-фильтр описывается передаточной функцией общего вида (12.6):

$$H(z) = \frac{\sum_{i=0}^{N-1} b_i z^{-i}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}}$$
(13.11)

и при  $(N-1) \le (M-1)$  (по умолчанию) имеет порядок R, равный

$$R = (M - 1). \tag{13.12}.$$

Помимо общего вида (13.11), используется представление передаточной функции БИХ-фильтра в виде произведения (12.11):

$$H(z) = G \prod_{k=1}^{L} \frac{1 + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}}.$$
 (13.13)

В МАТLАВ представление передаточной функции общего вида (13.11) в виде (13.13) выполняется с помощью функции:

[s,G]=tf2sos(b,a)

где:

ь, а — векторы коэффициентов передаточной функции (13.11):  $b_i$ , i = 0, 1, ..., N-1, и  $a_k$ , k = 1, 2, ..., M-1; а (1) =1;

G — коэффициент усиления G в (13.13);

s — вещественные коэффициенты звеньев 2-го порядка в H(z) (13.13), представленные в виде *матрицы*:

$$\begin{bmatrix} 1 & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ 1 & b_{12} & b_{22} & 1 & a_{12} & a_{22} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & b_{1L} & b_{2L} & 1 & a_{1L} & a_{2L} \end{bmatrix}.$$
(13.14)

БИХ-фильтры характеризуются следующими особенностями:

**О** существенно нелинейной ФЧХ;

□ необходимостью проверки на устойчивость.

## 13.5. Задание требований к частотным характеристикам БИХ-фильтров

При синтезе частотно-избирательных БИХ-фильтров с существенно нелинейной ФЧХ последняя обычно не контролируется, и требования задаются к АЧХ. Они не отличаются от требований к АЧХ КИХ-фильтров (см. разд. 13.3), за тем исключением, что значение АЧХ в полосе пропускания не должно превышать единицы. Кроме того, для БИХ-фильтров принято задавать требования не к АЧХ, а к характеристике затухания (13.6) — АЧХ (дБ).

Таким образом, требования к АЧХ БИХ-фильтра включают в себя:

- $\Box$  частоту дискретизации  $f_{\pi}$  (Гц);
- □ *граничные частоты* полос пропускания (ПП) и полос задерживания (ПЗ), для которых введены условные обозначения:
  - *f*<sub>γ</sub> граничная частота ПП для ФНЧ и ФВЧ;
  - *f<sub>k</sub>* граничная частота ПЗ для ФНЧ и ФВЧ;
  - $f_{-\chi}$ ,  $f_{\chi}$  левая и правая граничные частоты ПП для ПФ и РФ;
  - $f_{-k}$ ,  $f_k$  левая и правая граничные частоты ПЗ для ПФ и РФ;

**П** *допустимые отклонения* от  $\hat{A}(f)$  (дБ) (13.6):

- *a*<sub>max</sub> (дБ) максимально допустимое затухание в ПП;
- *a*<sub>min</sub> (дБ) минимально допустимое затухание в ПЗ.

# 13.6. Структуры цифровых фильтров

Структура ЦФ отображает алгоритм вычисления реакции по разностному уравнению (12.1) и определяется видом передаточной функции.

В MATLAB структура ЦФ описывается в виде объекта dfilt (от англ. Discretetime filter object) [22]:

#### Hd=dfilt.structure(input1,...)

где:

нd — имя объекта dfilt;

dfilt — тип объекта;

structure — функция, задающая конкретную структуру объекта нd;

input1,... — список параметров функции structure.

Выходными параметрами созданного объекта dfilt являются свойства созданного объекта на, зависящие от конкретной структуры и типа арифметики.

*Список свойств* объекта dfilt выводится по имени объекта нd или по команде MATLAB:

>> get <имя объекта dfilt>

*Тип арифметики*, под которым понимают форму представления данных в структуре ЦФ, отображается свойством Arithmetic со следующими возможными значениями:

- □ Arithmetic: 'double' представление данных с плавающей точкой (ПТ) с удвоенной точностью типа double;
- □ Arithmetic: 'single' представление данных с ПТ с одинарной точностью типа single;
- □ Arithmetic: 'fixed' представление данных с фиксированной точкой (ФТ).

В Simulink, наряду с описанием структуры ЦФ в виде *объекта* dfilt, используется ее графическое представление в виде *структурной схемы* с условными обозначениями операций умножения, задержки и суммирования.

Основные *структуры* ЦФ, поддерживаемые Simulink и описываемые в виде объектов dfilt, приведены в табл. 13.2, а *структурные схемы* — на рис. 13.3 и 13.4.

Структура structure	Функция structure	Параметры функции structure
Direct-Form I (прямая, рис. 13.3, <i>а</i> )	df1(b,a)	b, а — векторы коэффициентов передаточной функции (13.11)
Direct-Form I, SOS <sup>1</sup> (каскадная из звеньев 2-го порядка с прямой структурой)	dflsos (s,G)	s, G— матрица коэффициентов в виде (13.14) и вектор коэффициентов усиления передаточной функции (13.13)
Direct-Form I Transposed (прямая транспонированная, рис. 13.3, б)	dflt(b,a)	b, а — векторы коэффициентов передаточной функции (13.11)
Direct-Form I Transposed, SOS (каскадная из звеньев 2-го порядка с прямой транспонированной структурой)	df1tsos(s,G)	s, G— матрица коэффициентов в виде (13.14) и вектор коэффициентов усиления передаточной функции (13.13)
Direct-Form II (прямая каноническая, рис. 13.3, е)	df2(b,a)	b, а — векторы коэффициентов передаточной функции (13.11)
Direct-Form II, SOS (каскадная из звеньев 2-го порядка с прямой канонической структурой)	df2sos (s,G)	s, G— матрица коэффициентов в виде (13.14) и вектор коэффициентов усиления передаточной функции (13.13)
Direct-Form II Transposed (прямая каноническая транспонированная, рис. 13.3, <i>г</i> )	df2t(b,a)	b, а — векторы коэффициентов передаточной функции (13.11)
Direct-Form II Transposed, SOS (каскадная из звеньев 2-го порядка с прямой канонической транспониро- ванной структурой)	df2tsos(s,G)	s, G— матрица коэффициентов в виде (13.14) и вектор коэффициентов усиления передаточной функции (13.13)

Таблица 13.2. Структуры ЦФ и их описание в виде объекта dfilt

<sup>&</sup>lt;sup>1</sup> Сокращение от Second-Order Sections.

#### Окончание табл. 13.2

Структура structure	Функция structure	Параметры функции structure
Direct-Form FIR (прямая КИХ, рис.13.4, <i>а</i> )	dffir(b)	<ul> <li>b — вектор коэффициентов</li> <li>передаточной функции (13.1)</li> </ul>
Direct-Form FIR Transposed (прямая транспонированная КИХ, рис. 13.4, <i>б</i> )	dffirt(b)	<ul> <li>b — вектор коэффициентов</li> <li>передаточной функции (13.1)</li> </ul>
Direct-Form Symmetric FIR (прямая приведенная КИХ с симметричной импульсной характеристикой, рис. 13.4, <i>є</i> )	dfsymfir(b)	<ul> <li>b — вектор коэффициентов передаточной функции (13.1).</li> <li>Вектор b имеет длину N и его элементы симметричны относительно точки N/2</li> </ul>
Direct-Form Antisymmetric FIR (прямая приведенная с антисимметричной импульсной характеристикой, рис. 13.4, <i>г</i> )	dfasymfir(b)	b — вектор коэффициентов передаточной функции (13.1). Вектор b имеет длину $N$ и его элементы антисимметричны относительно точки $N/2$ , при этом, если $N$ — четное, то b ( $N/2$ )=0









Рис. 13.3. Структуры (структурные схемы) рекурсивного звена 2-го порядка: прямая (Direct-Form I) (*a*), прямая транспонированная (Direct-Form I Transposed) (*б*), прямая каноническая (Direct-Form II) (*в*) и прямая каноническая транспонированная (Direct-Form II Transposed) (*a*)



Рис. 13.4. Структуры (структурные схемы) нерекурсивного фильтра: прямая (Direct-Form I FIR) для нерекурсивного звена 2-го порядка (*a*), прямая транспонированная (Direct-Form I Transposed FIR) для нерекурсивного звена 2-го порядка (*б*), прямая приведенная с симметричной ИХ (Direct-Form Symmetric FIR) для КИХ-фильтра 1-го типа длины N = 7 (*в*) и прямая приведенная с антисимметричной ИХ (Direct-Form Antisymmetric FIR) для КИХ-фильтра 3-го типа длины N = 7 (*b*<sub>3</sub> = 0) (*e*)

# 13.7. Математическая модель и S-модель системы цифровой фильтрации

Математической моделью системы цифровой фильтрации называют соотношение вход/выход ЦФ в виде разностного уравнения (см. разд. 12.1).

Математическая модель системы цифровой фильтрации включает в себя три составляющие:

**П** входной дискретный сигнал x(n);

□ структуру ЦФ.

Структура ЦФ определяется видом передаточной функции;

□ выходной дискретный сигнал *y*(*n*).

В *S-модели системы цифровой фильтрации* им будут соответствовать три отличающиеся своим функциональным назначением компоненты:

**П** S-модель входного дискретного сигнала x(n);

□ S-модель структуры ЦФ.

Для краткости S-модель структуры ЦФ будем называть S-моделью ЦФ;

 $\Box$  S-модель средств анализа выходного дискретного сигнала y(n).

# 13.8. Ѕ-модель ЦФ и средства ее создания

*S-модель ЦФ* отображает структуру ЦФ, описанную в виде объекта dfilt со свойством Arithmetic: 'double' (см. разд. 13.6).

Средства для создания *S-моделей ЦФ* представлены в библиотеке блоков Signal Processing Blockset в группе блоков **Filtering** (Фильтрация).

*S-модель ЦФ* может создаваться:

- $\square$  в виде блока Digital Filter Design из группы блоков Filtering  $\rightarrow$  Filter Designs (Фильтрация  $\rightarrow$  Проекты фильтров);
- □ в виде блока из группы блоков Filtering → Filter Design Toolbox (Фильтрация → Средства проектирования фильтров).

## 13.8.1. S-модель ЦФ в виде блока Digital Filter Design

Блок *Digital Filter Design* предназначен для создания S-моделей ЦФ средствами, подобными GUI FDATool.

Двойной щелчок на пиктограмме блока Digital Filter Design открывает окно Block Parameters: Digital Filter Design (Блок параметров: Проектирование цифрового фильтра), подобное окну Filter Design & Analysis Tool (Средства проектирования и анализа фильтра) GUI FDATool, но с ограниченным набором кнопок на панели инструментов в нижнем левом углу (рис. 13.5). В частности отсутствует кнопка Set quantization parameters (Установка параметров квантования), т. к. по умолчанию установлен тип арифметики Filter Arithmetic: Double-precision floatingpoint, что для объекта dfilt соответствует свойству Arithmetic: 'double'.

Синтез ЦФ в окне Block Parameters: Digital Filter Design производится при нажатой кнопке Design Filter (Проектирование фильтра), расположенной на панели инструментов в нижнем левом углу (см. рис. 13.5), и включает в себя следующие *основные этапы*:

1. Выбор типа ЦФ.

Тип фильтра выбирается в группе **Design Method** (Метод синтеза) с помощью переключателя **IIR** (БИХ-фильтр) или **FIR** (КИХ-фильтр).



Рис. 13.5. Окно Block Parameters: Digital Filter Design

2. Выбор метода синтеза ЦФ.

Метод синтеза выбирается в группе **Design Method** в раскрывающихся списках **FIR** или **IIR**.

3. Задание входных параметров.

Входные параметры задаются в группе **Options** (Параметры), и их набор зависит от типа ЦФ (КИХ или БИХ), метода синтеза и переключателя в группе **Filter Order** (Порядок фильтра). Некоторые из них приводятся в *разд. 13.8.2*.

4. Задание требований к АЧХ.

Требования к АЧХ включают в себя:

тип избирательности, задаваемый в группе Response Type (Тип характеристики) с помощью переключателей Lowpass (ФНЧ), Highpass (ФВЧ), Bandpass (ПФ) или Bandstop (РФ).

В раскрывающихся списках Lowpass и Highpass присутствуют разновидности фильтров с данными типами избирательности. В последнем раскрывающемся списке специальных фильтров (без имени) представлены Hilbert Transformer (Преобразователь Гильберта), Differentiator (Дифференциатор) и др.;

• частоту дискретизации и граничные частоты ПП и ПЗ, задаваемые в группе Frequency Specifications (Требования к частотам).

Предварительно в раскрывающемся списке Units (Единицы измерения) указываются единицы измерения частот, после чего задаются частоты:

- в поле ввода **Fs** частота дискретизации;
- в соответствующих полях ввода граничные частоты ПП и ПЗ;
- допустимые отклонения АЧХ в ПП и ПЗ, задаваемые в группе Magnitude Specifications (Требования к АЧХ).

Предварительно в раскрывающемся списке Units указываются единицы измерения АЧХ:

- Linear (Безразмерный) если требования задаются к нормированной АЧХ;
- dB (дБ) если требования задаются к АЧХ (дБ) (характеристике затухания (13.6)).

После этого в соответствующих полях ввода задаются допустимые отклонения АЧХ.

5. Синтез ЦФ.

Синтез ЦФ осуществляется после нажатия кнопки **Design Filter** (Синтез фильтра), расположенной внизу по центру окна (см. рис. 13.5).

По завершении синтеза автоматически выдаются:

- в группе Magnitude Response (dB) (АЧХ (дБ)) график АЧХ (дБ) (характеристика ослабления (13.5)).
- в группе Current Filter Information (Информация о текущем фильтре):
  - Structure структура фильтра;
  - Order порядок фильтра;
  - Stable информация об устойчивости фильтра (Yes или No);
  - Source способ загрузки фильтра в результате текущего синтеза (Designed) или импорта (Imported).

По завершении синтеза ЦФ автоматически создается его описание в виде объекта dfilt со структурой, выбираемой по умолчанию.

Преобразование структуры выполняется по команде Edit | Convert Structure (Редактирование | Преобразование структуры). Открывается окно Convert Structure со списком рекомендуемых для данного ЦФ структур. Выбирается требуемая структура и нажимается кнопка OK. Новая структура фиксируется в группе Current Filter Information.

Для сохранения синтезированного Ц $\Phi$  следует закрыть окно **Block Parameters: Digital Filter Design** и открыть окно S-моделей, в котором *S-модель Ц* $\Phi$  будет представлена блоком Digital Filter Design.

Более подробную справку можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Filtering  $\rightarrow$  Filter Designs  $\rightarrow$  Digital Filter Design.

# 13.8.2. S-модель ЦФ в виде блока из группы блоков *Filter Design Toolbox*

Основные блоки из группы блоков Filtering  $\rightarrow$  Filter Design Toolbox представлены в табл. 13.3.

N⁰	Блок	Назначение
1.	Lowpass Filter	ЦФ ФНЧ
2.	Highpass Filter	ЦФ ФВЧ
3.	Bandpass Filter	ЦФ ПФ
4.	Bandstop Filter	ЦФ РФ
5.	Halfpass Filter	Полуполосный ЦФ ФНЧ или ФВЧ
6.	Nyquist Filter	ЦФ Найквиста
7.	Differentiator Filter	Цифровой дифференциатор (ЦД)
8.	Hilbert Filter	Цифровой преобразователь Гильберта (ЦПГ)

Таблица 13.3. S-модели ЦФ — блоки группы блоков Filter Design Toolbox

Параметры блоков 1–8 из табл. 13.3 по существу отображают средства GUI FDATool при установке типа арифметики Filter Arithmetic: Double-precision floating-point. Рассмотрим их для блока Lowpass Filter.

#### 13.8.2.1. Блок Lowpass Filter

Блок *Lowpass Filter* предназначен для создания S-модели ЦФ ФНЧ и список его параметров включает в себя (рис. 13.6):

□ 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 синтезируемого ЦФ:
  - Minimum (Минимальный) минимальный порядок;
  - Specify (Произвольный порядок) произвольный порядок;

При выборе Order mode: Specify order активизируется поле ввода порядка Order (Порядок);

- Filter type (Тип фильтра) активен для КИХ-фильтра (Impulse response: FIR) и отображает его разновидности:
  - Single-rate (Односкоростной) КИХ-фильтр;
  - Decimator (Дециматор) КИХ-фильтр для системы однократной децимации с указанием коэффициента децимации в поле ввода Decimation factor (Коэффициент децимации);
  - Interpolator (Интерполятор) КИХ-фильтр для системы однократной интерполяции с указанием коэффициента интерполяции в поле ввода Interpolation factor (Коэффициент интерполяции);

□ в группе Frequency specifications (Требования к частотам) — параметры:

- Frequency units (Единицы измерения частот) 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) АЧХ (дБ) (характеристика затухания (13.6));
- Apass, Astop допустимые отклонения в ПП и ПЗ;

□ в группе Algorithm (Алгоритм) — параметры:

• **Design Method** (Метод синтеза) — методы синтеза.

Для КИХ-фильтров выделим методы:

- Equiripple наилучшей равномерной (чебышевской) аппроксимации;
- Kaiser window окон с использованием окна Кайзера.

Для БИХ-фильтров — метод билинейного Z-преобразования со следующими типами аппроксимации:

- Butterworth Баттерворта;
- Chebyshev type I Чебышева 1-го рода;
- Chebyshev type II Чебышева 2-го рода;
- Elliptic Золотарева—Кауэра (эллиптический);
- Structure (Структура) структура ЦФ (см. табл. 13.2).

Отметим, что для синтеза частотно-избирательных КИХ-фильтров используются КИХ-фильтры *1-го и 2-го типа* (см. табл. 13.1), для которых рекомендуется выбрать структуру Direct-Form Symmetric FIR;

• Scale SOS filter coefficients to reduce chance to overflow (Масштабирование коэффициентов звеньев для уменьшения возможности переполнения) — флаг, активный для БИХ-фильтров с каскадной структурой.

При установке флага (по умолчанию) выполняется *масштабирование коэффициентов передаточной функции ЦФ* для предотвращения или минимизации переполнений на выходе сумматоров при последующей реализации БИХ-фильтра с ФТ.

Формирование звеньев посредством объединения полюсов (добротностей полюсов) с ближайшими нулями (добротностями нулей) и расстановка звеньев в порядке возрастания радиусов полюсов для минимизации собственных шумов в БИХ-фильтре при создании S-модели ЦФ выполняется автоматически;

 Design options (Параметры проектирования) — список входных параметров, автоматически формируемый в зависимости типа ЦФ (Impulse response), метода синтеза (Design Method) и порядка ЦФ (Order mode).

Приведем некоторые типовые входные параметры.

 Density Factor (Коэффициент плотности) — коэффициент плотности сетки частот — для КИХ-фильтров при выборе Design Method: Equiripple.

Параметр **Density Factor** тождественен параметру lgrid в функциях MATLAB firpm и firgr и используется для автоматического расчета количества точек на густой сетке интервала аппроксимации. С ростом значения Density Factor возрастает точность вычисления коэффициентов передаточной функции (13.1).

• Wpass и Wstop — веса в ПП и ПЗ — для КИХ-фильтров при выборе Design Method: Equiripple и Order mode: Specify.

Вес, равный единице, присваивается полосе с наибольшим допустимым отклонением, а веса в остальных полосах рассчитываются как отношение наибольшего отклонения к допустимому отклонению в данной полосе. Поэтому веса — всегда числа, большие или равные единице;

 Scale Passband (Масштабирование в ПП) — флаг нормирования АЧХ для КИХ-фильтров при выборе Design Method: Kaiser и Order mode: Minimum.

Параметр Scale Passband тождественен параметру normalization в функции MATLAB fir1 и используется для нормирования AЧX таким образом, чтобы обеспечить ее значение, равное единице, в центре ПП;

 Match Exactly (Согласование точно) — точное выполнение требований к АЧХ в ПП и ПЗ (Both), в ПП (Passband), в ПЗ (Stopband) — для БИХфильтров.

Параметр Match Exactly тождественен параметру MATCH в функциях butter, cheby1, cheby2 и ellip.

Function Blo	ock Parameters: Lowpa	iss Filter	
Lowpass Filter			
Design a lowpass	filter.		
- Filter specification	ne		View Filter Response
Impulse response	FIR	~	
Order mode:	Minimum	✓ Order:	
Filter type:	Single-rate	~	
Frequency specif	ications		
Frequency units:	Hz	Input Fs:	8000
Fpass:	1000	Fstop:	1500
Magnitude specifi	cations		
Magnitude units:	Linear	~	
Apass:	0.05	Astop:	0.01
Algorithm			
Design method:	Equiripple		~
Structure:	Direct-form symmetric FIR		~
Design opt	ions		
		<u>C</u> ancel	Help Apply

Рис. 13.6. Параметры блока Lowpass Filter

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Filtering  $\rightarrow$  Filter Design Toolbox.

## 13.9. Анализ характеристик S-модели ЦФ

Анализ характеристик S-модели Ц $\Phi$  в виде блока Digital Filter Design производится после синтеза Ц $\Phi$  непосредственно в окне Block Parameters: Digital Filter Design, обращаясь к командам пункта меню Analysis (см. разд. 12.3).

Для анализа характеристик S-модели ЦФ, *представленного блоками в табл. 13.3*, в окне параметров блоков предусмотрена кнопка View Filter Response (Просмотр характеристик фильтра). При ее нажатии автоматически открывается окно Filter Visualization Tool программы GUI FVTool (*см. разд. 12.3*).

Отметим, что по умолчанию выводится график характеристики ослабления (13.5).

## 13.10. Примеры S-моделей ЦФ

Создание S-моделей ЦФ в виде блоков из табл. 13.3 рассмотрим на примерах частотно-избирательных ФНЧ и ПФ, а также ЦПГ и ЦД.

Заданы требования к АЧХ ФНЧ (табл. 13.4).

Частоты (Гц)			Максимально допустимые отклонения АЧХ				
Частота дискретизации	$f_{\mathtt{A}}$	8000	-		-		
Граничная частота ПП	$f_{\chi}$	1000	В полосе пропускания (ПП)	$\delta_1$	0,05		
Граничная частота ПЗ	$f_k$	1500	В полосе задерживания (ПЗ)	δ2	0,01		

Таблица 13.4. Требования к АЧХ ФНЧ

Создадим *S-модель КИХ-фильтра* ФНЧ, синтезированного методом чебышевской аппроксимации, в виде блока Lowpass Filter (см. табл. 13.3) с параметрами, представленными на рис. 13.6.

АЧХ КИХ-фильтра ФНЧ представлена на рис. 13.7. Для проверки выполнения требований к АЧХ выведены координаты точек альтернанса в ПП и ПЗ посредством подведения курсора к требуемой точке и щелчка левой кнопки мыши. (Расставленные точки удаляются с помощью команды контекстного меню **Delete** или **Delete all**.) Как видим, требования к АЧХ выполняются.

КИХ-фильтр ФНЧ имеет длину N = 26 и структуру Direct-Form Symmetric FIR.

В окне параметров блока Lowpass Filter нажмем кнопку **ОК** — S-модель КИХфильтра ФНЧ в виде блока Lowpass Filter создана (рис. 13.8).



Рис. 13.7. АЧХ S-модели КИХ-фильтра ФНЧ



Рис. 13.8. S-модели ЦФ: КИХ-фильтра ФНЧ (блок Lowpass Filter), КИХ-фильтра ПФ (блок Bandpass Filter), ЦПГ (блок Hilbert Filter), ЦД (блок Differentiator Filter), БИХ-фильтра ФНЧ (блок Lowpass Filter1) и БИХ-фильтра ПФ (блок Bandpass Filter1)

#### Пример 13.3

Заданы требования к АЧХ ПФ (табл. 13.5).

Частоты (Гц)			Максимально допустимые отклонения АЧХ			
Частота дискретизации	$f_{\rm A}$	8000	_	_	_	
Граничная частота ПЗ1	$f_{-k}$	1000	В полосе задерживания (ПЗ1)	$\delta_2$	0,01	
Левая граничная частота ПП	$f_{-\chi}$	1400		N	0.05	
Правая граничная частота ПП	$f_{\chi}$	2000	В полосе пропускания (ппт)	01	0,05	
Граничная частота ПЗ2	$f_k$	2400	В полосе задерживания (ПЗ2)	δ2	0,01	

Таблица 13.5. Требования к АЧХ ПФ

Создадим *S-модель КИХ-фильтра* ПФ, синтезированного методом чебышевской аппроксимации, в виде блока Bandpass Filter (см. табл. 13.3) с параметрами, представленными на рис. 13.9.

АЧХ КИХ-фильтра ПФ с координатами точек альтернанса в ПП и ПЗ представлена на рис. 13.10 (об их выводе см. в примере 13.2). Как видим, требования к АЧХ выполняются. КИХ-фильтр ПФ имеет длину N = 33 и структуру Direct-Form Symmetric FIR.

В окне параметров блока Bandpass Filter нажмем кнопку **ОК** — S-модель КИХфильтра ПФ в виде блока Bandpass Filter создана (см. рис. 13.8).

View Filter Response         File view Filter Response         Order mode: Minimum of Order:         Order mode:       Minimum of Order:         Filter type:       Single-rate of Order:         Frequency specifications       Frequency specifications         Frequency units:       Hz       Input Fs:       8000         Fstop1:       1000       Fpass1:       1400         Fpass2:       2000       Fstop2:       2400         Magnitude specifications       Magnitude units:       Linear       Magnitude units:       0.01         Astop1:       0.01       Apass:       0.05       Astop2:       0.01         Agonithm       Design method:       Equiripple       View Filter Size       View Filter Response         Interview:       Direct-form symmetric FIR       View Filter Response       View Filter Response	Bandpass Filter Design a bandpas	s filter.				
Filter specifications  Impulse response: FIR  Order mode: Minimum  Corder: Filter type: Single-rate  Frequency specifications  Frequency units: Hz  Input Fs: 8000  Fstop1: 1000  Fpass1: 1400  Fpass2: 2000  Fstop2: 2400  Magnitude specifications  Magnitude units: Linear  Astop1: 0.01  Apass: 0.05  Astop2: 0.01  Agorithm  Design method: Equiripple   Filter  Filter Filter  Filter  Filter  Filter  Filter  Filter Filter  Filter  Filter Filter  Filter  Filter Filter Filter  Filter Filter  Filter Filt					View Filt	er Response
Impulse response: FIR   Order mode: Minimum  Order mode: Minimum  Filter type: Single rate  Frequency specifications  Frequency units: Hz  Frequency units: Hz  Magnitude specifications  Magnitude units: Linear  Astop 1: 0.01  Apass: 0.05  Astop 2: 0.01  Agonthm  Design method: Equiripple  Filter  Design options  Design options	Filter specification	15				
Order mode:       Minimum       ♥ Order:         Filter type:       Single-rate       ♥         Frequency specifications       ●         Frequency units:       Hz       ♥ Input Fs:       8000         Fstop1:       1000       ● Fpass1:       1400         Fpass2:       2000       ● Fstop2:       2400         Magnitude specifications       ●       ●         Magnitude units:       Linear       ♥         Astop1:       0.01       Apass:       0.05         Astop2:       0.01       ●       ●         Structure:       Direct-form symmetric FIR       ♥         ▶ Design potions       ●       ●	Impulse response	FIR	~	1		
Filter type: Single-rate  Frequency specifications  Frequency units: Hz  Frequency units: Hz  Frequency units: Hz  Frequency units: Hz  Frequency units: Linear  Magnitude specifications  Magnitude units: Linear  Astop 1: 0.01  Apass: 0.05  Astop 2: 0.01  Agorithm  Design method: Equiripple  Frequency  Design options  Design options	Order mode:	Minimum	~	Order:		1
Frequency specifications         Frequency units:       Hz       Input Fs: 8000         Fstop1:       1000       Fpass1:       1400         Fpass2:       2000       Fstop2:       2400         Magnitude specifications       Magnitude units:       Linear       Imput Fs: 8000         Astop1:       0.01       Apass:       0.05         Astop2:       0.01       Imput Fs: 8000       Imput Fs: 8000         Structure:       Direct-form symmetric FIR       Imput Fs: 8000	Filter type:	Single-rate	~			
Frequency units:       Hz       Input Fs: 8000         Fstop1:       1000       Ppass1:       1400         Fpass2:       2000       Fstop2:       2400         Magnitude specifications       Magnitude units:       Linear       Image: Additional additionadditional additeratinadditerational additionadditionadditerationadd	Frequency specifi	cations				
Fstop1:         1000         Fpass1:         1400           Fpass2:         2000         Fstop2:         2400           Magnitude specifications	Frequency units:	Hz	~ 1	nput Fs:	8000	
Fpass2:     2000     Fstop2:     2400       Magnitude specifications       Magnitude units:     Linear       Astop1:     0.01     Apass:       0.01     Apass:     0.05       Astop2:     0.01     Apass:       Design method:     Equiripple       Structure:     Direct-form symmetric FIR	Fstop 1:	1000	1	pass 1:	1400	
Magnitude specifications Magnitude units: Linear Astop 1: 0.01 Apass: 0.05 Astop 2: 0.01 Algorithm Design method: Equripple Structure: Direct-form symmetric FIR Design potions	Fpass2:	2000	F	stop2:	2400	
Magnitude units: Linear Astop 1: 0.01 Apass: 0.05 Astop 2: 0.01 Algorithm Design method: Equiripple Structure: Direct-form symmetric FIR Design potions	Magnitude specifi	cations				
Astop 1: 0.01 Apass: 0.05 Astop 2: 0.01 Agorithm Design method: Equiripple Structure: Direct-form symmetric FIR Design potions	Magnitude units:	Linear	~			
Astop2: 0.01 Algorithm Design method: Equiripple Structure: Direct-form symmetric FIR Design potions	Astop 1:	0.01	-	Apass:	0.05	
Algorithm Design method: Equiripple Structure: Direct-form symmetric FIR Design potions	Astop2:	0.01				
Design method: Equiripple  Structure: Direct-form symmetric FIR  Design potions	Algorithm					
Structure: Direct-form symmetric FIR	Design method:	auripole				~
Design options	Structure: 0	Direct-form symmetric	FIR			~
	Design opti	ions				

Рис. 13.9. Параметры блока Bandpass Filter



Рис. 13.10. АЧХ S-модели КИХ-фильтра ПФ

Заданы требования к АЧХ ЦПГ (табл. 13.6).

Таблица 13.6. Требования к АЧХ ЦПГ

Частоты (Гц)			Максимально допустимые отклонения АЧХ		
Частота дискретизации	$f_{\rm A}$	8000	_	_	
Суммарная ширина переходных полос	рная ширина переходных полос <i>Д</i> f 800		_	_	
Граничная частота ПП		—	Максимально	$\delta_1$	0.01
Граничная частота ПП	$f_{\chi}$	_	от единицы в ПП		0,01

Напомним [22], что ЦПГ может быть реализован на базе КИХ-фильтров 3-го и 4го типов (см. табл. 13.1), ФЧХ которых обеспечивает сдвиг на  $\pi/2$ . Предпочтение отдается КИХ-фильтру 3-го типа, т. к. он позволяет получить ИХ, каждый второй отсчет которой равен нулю, тем самым сокращается число арифметических операций при вычислении реакции ЦПГ.

На базе КИХ-фильтра 3-го типа можно синтезировать только ПФ, при этом специфика требований к АЧХ ЦПГ, по сравнению с требованиями к АЧХ частотноизбирательного ПФ, будет следующей.

□ АЧХ ЦПГ должна быть *симметричной* относительно середины основной полосы частот  $f_{II}/4$  для получения ИХ, *каждый второй отсчет которой равен нулю*.

Требования к АЧХ ЦПГ задаются симметричными относительно  $f_{\rm g}/4$ , что в Simulink при заданной частоте дискретизации выполняется автоматически, поэтому граничные частоты ПП не задаются, а задается *суммарная ширина переходных полос* — параметр **Transition width** (Ширина переходной полосы).

Рабочая полоса ЦПГ не превосходит полосу пропускания ПФ.

- □ Максимально допустимое отклонение в рабочей полосе  $\delta_{\text{раб}}$  не должно быть меньше максимально допустимого отклонения  $\delta_1$  в ПП. В Simulink задается  $\delta_{\text{раб}} = \delta_1$ .
- Μаксимально допустимое отклонение δ<sub>2</sub> в ПЗ нет необходимости задавать слишком жестко, так как эффективность ЦПГ оценивается в рабочей полосе. В Simulink значение δ<sub>2</sub> задается автоматически.

Создадим *S-модель ЦПГ*, синтезированного методом чебышевской аппроксимации, в виде блока Hilbert Filter (см. табл. 13.3) с параметрами, представленными на рис. 13.11.

Отметим, что на основе блока Hilbert Filter возможен синтез ЦПГ только на базе *КИХ-фильтра 4-го типа*. (При выборе **FIR type:** 3 после нажатия кнопки **OK** произойдет автоматическая замена на **FIR type:** 4). При необходимости синтеза ЦПГ на базе КИХ-фильтра 3-го типа следует выбрать блок Digital Filter Design *(см. разд. 13.8.1).* 

АЧХ ЦПГ представлена на рис. 13.12. ЦПГ имеет длину N = 16 и структуру Direct-Form Antisymmetric FIR.

В окне параметров блока Hilbert Filter нажмем кнопку **ОК** — S-модель ЦПГ в виде блока Hilbert Filter создана (см. рис. 13.8).

Design a Hilbert filte			
	f.		
			View Filter Response
Ellar marifications			Them I had the approve
- I			
Impulse response:	FIR		
Order mode:	Minimum	Order:	
Filter type:	Single-rate		
Frequency specifica	ations		
Frequency units: H	7	Input Es: 8000	
Transition width: 8	00		12
Magnitude specifica	itions		
Magnitude units: Li	near	~	
Apass: 0	.01		
Algorithm			
Design method: Eg	uiripple		~
Structure: Dir	ect-form antisymmetric FIR		~
Design option	15		

Рис. 13.11. Параметры блока Hilbert Filter



Рис. 13.12. АЧХ Ѕ-модели ЦПГ

Заданы требования к АЧХ ЦД (табл. 13.7).

Таблица 13.7. Требования к АЧХ ЦД

Частоты (Гц)			Максимально допустимые отклонения АЧХ		
Частота дискретизации	$f_{\mathtt{A}}$	8000	—	_	
Граничная частота рабочей полосы	$f_{\rm pad}$	2000	В рабочей полосе	$\delta_{\text{pad}}$	0,02
Граница нерабочей полосы	$f_{\mathrm{hepa6}}$	3000	В нерабочей полосе	$\delta_{_{Hepab}}$	0,1

Напомним [22], что ЦД синтезируется на основе КИХ-фильтров 3-го или 4-го типа по требованиям к АЧХ в рабочей полосе  $[f_{\text{раб1}}; f_{\text{раб2}}]$ , совпадающей с основной полосой [0;  $f_{\pi}/2$ ] либо расположенной внутри нее.

В зависимости от расположения рабочей полосы в основной полосе  $[0; f_{д}/2]$  различают следующие разновидности ЦД [22]:

широкополосные ЦД (ШЦД) — с рабочей полосой  $[0; f_{д}/2];$ 

□ низкочастотные ЦД (НЦД) — с рабочей полосой [0; *f*<sub>раб</sub>];

П полосовые ЦД (ПЦД) — с рабочей полосой  $[f_{\text{раб1}}; f_{\text{раб2}}];$ 

□ высокочастотные ЦД (ВЦД) — с рабочей полосой  $[f_{\text{раб}}; f_{\text{д}}/2]$ .

Так как АЧХ КИХ-фильтров 3-го типа по определению равна нулю на границах основной полосы частот, ШЦД и ВЦД можно синтезировать только на базе КИХ-фильтров 4-го типа.

Создадим *S-модель ЦД*, синтезированного методом чебышевской аппроксимации, в виде блока Differentiator Filter (см. табл. 13.3) с параметрами, представленными на рис. 13.13.

Отметим, что на основе блока Differentiator Filter возможен синтез ЦД только на базе *КИХ-фильтра 3-го типа*. При необходимости синтеза ЦД на базе *КИХ-фильтра 4-го типа* следует выбрать блок Digital Filter Design (см. разд. 13.8.1).

АЧХ ЦД представлена на рис. 13.14. ЦД имеет длину N = 15 и структуру Direct-Form Antisymmetric FIR.

В окне параметров блока Differentiator Filter нажмем кнопку **ОК** — S-модель ЦД в виде блока Differentiator Filter создана (см. рис. 13.8).

Function	Block Paramet	ers: Differen	tiator F	ilter	
Differentiator	Filter				
Design a differ	rentiator.				
				View Filter Re	sponse
Filter specifica	ations				
Order mode:	Minimum	~	Order:		
Filter type:	Single-rate	~			
Frequency sp	ecifications				
Frequency un	its: Hz	~	Input Fs:	8000	
Fpass:	2000		Fstop:	3000	
Magnitude sp	ecifications				
Magnitude un	its: Linear		•		
Apass:	0.02		Astop:	0.1	
Algorithm					
Design metho	d: Equiripple				~
Structure:	Direct-form anti	symmetric FIR			~
Design	options				
		<u>ok</u>	Cancel	Help (	Apply

Рис. 13.13. Параметры блока Differentiator Filter



Рис. 13.14. АЧХ S-модели ЦД (КИХ-фильтра ПФ 3-го типа)

Заданы требования к АЧХ ФНЧ (см. табл. 13.4).

Создадим *S-модель БИХ-фильтра*  $\Phi H \Psi$  *Баттерворта*, синтезированного методом билинейного *Z*-преобразования, в виде блока Lowpass Filter (см. табл. 13.3) с параметрами, представленными на рис. 13.15. Значения  $a_{\text{max}} = 0,4455$  (дБ) и  $a_{\text{min}} = 40$  (дБ) определены в примере 13.1.

АЧХ БИХ-фильтра ФНЧ Баттерворта представлена на рис. 13.15. БИХфильтр ФНЧ имеет порядок *R* = 12 и каскадную структуру Direct-Form II SOS.

В окне параметров блока Lowpass Filter1 нажмем кнопку **ОК** — S-модель БИХ-фильтра ФНЧ Баттерворта в виде блока Lowpass Filter1 создана (см. рис. 13.8).

Function Block Parameters: Lowpass Filter1				
	- Lowpass Filter			
Filter Visualization Tool - Figure 1: Magnitude R	Design a lowpass filter.			
<u>Eile E</u> dit <u>A</u> nalysis <u>I</u> nsert <u>V</u> iew De <u>b</u> ug <u>D</u> esktop <u>W</u> indo		View Filter Response		
$\mathbb{C} \oplus \mathbb{C}   \mathbb{C} \setminus \mathbf{Z} = \mathbb{C}   \mathbb{C}   \mathbb{C} = \mathbb{C}$	Filter specifications	The second se		
🖸 🖸 🕼 🏦 🕇 ① 🦵 🌐 😡 🖸 🗹	Impulse response: IIIR			
	Order mode: Minimum Order: Order:			
	Filter type: Single-rate V			
	Frequency specifications			
0.9	Frequency units: Hz Input Fs: 8000			
	Fpass: 1000 Fstop: 1500			
0.8	Magnitude specifications			
0.7	Magnitude units: dB			
	Apass: 0.4455 Astop: 40			
0.6 •	Algorithm			
0.5	Design method: Butterworth	~		
E E	Structure: Direct-form II SOS	~		
0.4	Scale SOS filter coefficients to reduce chance of overflow			
0.3	Design options			
0.2				
0.1	OK Cancel He	Apply		
0				
0 0.5 1 1.5	2 2.5 3 3.5 requency (kHz)			

Рис. 13.15. Параметры блока Lowpass Filter1 и АЧХ S-модели БИХ-фильтра ФНЧ Баттерворта

Заданы требования к АЧХ ПФ (см. табл. 13.5).

Создадим *S-модель БИХ-фильтра* ПФ Золотарева—Кауэра, синтезированного методом билинейного *Z*-преобразования, в виде блока Bandpass Filter1 (см. табл. 13.3) с параметрами, представленными на рис. 13.16. Значения  $a_{max} = 0,4455$  (дБ) и  $a_{min} = 40$  (дБ) определены в примере 13.1.

АЧХ БИХ-фильтра ПФ Золотарева—Кауэра представлена на рис. 13.17. БИХфильтр ПФ имеет порядок R = 14 и каскадную структуру Direct-Form II SOS.

В окне параметров блока Bandpass Filter1 нажмем кнопку **ОК** — S-модель БИХфильтра ПФ Золотарева—Кауэра в виде блока Bandpass Filter1 создана (см. рис. 13.8).

Function Blo	ck Parameters: Band	pass Filte	er1 🔀
Bandpass Filter			
Design a bandpas	s filter.		
			View Filter Response
Filter specification	IS		
Impulse response	: IIR	~	
Order mode:	Minimum	V Order:	
Filter type:	Single-rate	~	
Frequency specifi	ications		
Frequency units:	Hz	Input Fs:	8000
Fstop 1:	1000	Fpass1:	1400
Fpass2:	2000	Fstop2:	2400
Magnitude specifi	cations		
Magnitude units:	dB	~	
Astop 1:	40	Apass:	0.4455
Astop2:	40		
Algorithm			
Design method: E	Butterworth		~
Structure:	Direct-form II SOS		~
Scale SOS filt	er coefficients to reduce ch	ance of ove	rflow
Design option	ions		
	<u>o</u> ĸ	<u>C</u> ancel	Help Apply

Рис. 13.16. Параметры блока Bandpass Filter1



Рис. 13.17. АЧХ S-модели БИХ-фильтра ПФ Золотарева—Кауэра

# 13.11. S-модели систем цифровой фильтрации

Создание S-моделей систем цифровой фильтрации на основе созданных S-моделей ЦФ рассмотрим на примерах.

#### Пример 13.8

Создадим S-модель системы цифровой фильтрации, включающей (рис. 13.18):

- □ S-модель входного дискретного сигнала (2-канальной последовательности) блок Sine Wave (см. разд. 11.2.2.8) с параметрами: Amplitude: [2 3]; Frequency (Hz): [700 2500]; Sample time: 1/8000; Samples per frame: 1;
- □ S-модель КИХ-фильтра ФНЧ блок Lowpass Filter (см. пример 13.2);
- □ S-модель средства анализа последовательностей блок Scope (см. разд. 5.2.3).

На входе КИХ-фильтра ФНЧ (блока Lowpass Filter) имеем 2-канальную последовательность — две дискретные синусоиды в двух каналах, а на выходе *в установившемся режиме* — 2-канальную последовательность, в которой дискретная синусоида первого канала с частотой 700 Гц попадает в ПП, а второго канала с частотой 2500 Гц — в ПЗ (нулевой сигнал).



Рис. 13.18. S-модель системы цифровой фильтрации с S-моделью КИХ-фильтра ФНЧ

#### Пример 13.9

Создадим S-модель системы цифровой фильтрации, включающей (рис. 13.19):

□ S-модель входного дискретного сигнала — одноканальной последовательности в виде суммы двух дискретных синусоид.

S-модель входного дискретного сигнала создается с использованием двух блоков Sine Wave *(см. разд. 11.2.2.8)* с параметрами соответственно: Amplitude: 2 и Amplitude: 3; Frequency (Hz): 700 и Frequency (Hz): 2500; Sample time: 1/8000; Samples per frame: 1;

- □ S-модель БИХ-фильтра ФНЧ блок Lowpass Filter1 (см. пример 13.6);
- □ S-модель средства анализа последовательностей блок Scope (см. разд. 5.2.3).

На входе КИХ-фильтра ФНЧ (блока Lowpass Filter1) имеем одноканальную последовательность — сумму двух дискретных синусоид, а на выходе *в установившемся режиме* одну дискретную синусоиду с частотой 700 Гц, которая попадает в ПП.



Рис. 13.19. Ѕ-модель системы цифровой фильтрации с Ѕ-моделью БИХ-фильтра ФНЧ

Создадим S-модель системы цифровой фильтрации, включающей (рис. 13.20):

- □ S-модель входного дискретного сигнала (одноканальной последовательности) блок Sine Wave *(см. разд. 11.2.2.8)* с параметрами: Amplitude: 1; Frequency **(Hz)**: 1000; Sample time: 1/8000;
- □ S-модель ЦПГ блок Hilbert Filter (см. пример 13.4);

□ S-модель средства анализа последовательностей — блок Scope (см. разд. 5.2.3).

На входе ЦПГ (блока Hilbert Filter) имеем одноканальную последовательность — дискретную синусоиду, а на выходе *в установившемся режиме* — дискретную синусоиду со сдвигом фазы на  $\pi/2$ .



Рис. 13.20. S-модель системы цифровой фильтрации с S-моделью ЦПГ



# Глава 14

# Системы цифровой фильтрации с фиксированной точкой

В этой главе рассматриваются S-модели систем цифровой фильтрации с фиксированной точкой, основным компонентом которых является цифровой фильтр с фиксированной точкой.

Цифровым фильтром с фиксированной точкой (Fixed-Point Filter), или коротко  $U \Phi c \Phi T$ , называют  $U \Phi$ , в структуре которого все данные — коэффициенты передаточной функции, результаты арифметических операций, а также воздействие и реакция представляются числами с  $\Phi T$ .

# 14.1. Ѕ-модель ЦФ с ФТ и средства ее создания

*S-модель ЦФ с ФТ* отображает структуру ЦФ, описанную в виде объекта dfilt со свойством Arithmetic: 'fixed' (*см. разд. 13.6*).

Средства для создания S-моделей ЦФ с ФТ представлены в библиотеке блоков Signal Processing Blockset в группе блоков Filtering  $\rightarrow$  Filter Designs (Фильтрация  $\rightarrow$  Проекты фильтров).

*S-модель ЦФ с ФТ* может создаваться:

- средствами блока Filter Realization Wizard;
- в виде блока Digital Filter;
- программными средствами МАТLAB.

# 14.2. Создание S-модели ЦФ с ФТ средствами блока Filter Realization Wizard

Блок *Filter Realization Wizard* не имеет входов и выходов и предназначен для создания S-моделей ЦФ с ФТ, которые экспортируются в окно S-моделей в виде *виртуальных подсистем (см. разд. 8.1).* 

Процедура создания S-модели ЦФ с ФТ включает в себя следующие этапы:

1. Синтез или импорт из Workspace исходного ЦФ.

Исходным ЦФ будем называть ЦФ, в котором установлен тип арифметики Filter Arithmetic: Double-precision floating-point, что для объекта dfilt соответствует свойству Arithmetic: 'double'.

- 2. Расстановка звеньев и масштабирование в каскадных структурах БИХ-фильтров.
- 3. Создание S-модели исходного ЦФ<sup>1</sup>.
- 4. Моделирование структуры ЦФ с ФТ.
- 5. Создание S-модели ЦФ с ФТ.

## 14.2.1. Синтез или импорт из Workspace исходного ЦФ

Синтез исходного ЦФ выполняется следующим образом:

- 1. Двойным щелчком на пиктограмме блока Filter Realization Wizard открывается окно Filter Design & Analysis Tool [fwizdef.mat\*].
- 2. Нажимается кнопка **Design Filter** (Проектирование фильтра) на панели инструментов в нижнем левом углу.

Окно Filter Design & Analysis Tool – [fwizdef.mat\*] принимает вид, подобный окну Filter Design & Analysis Tool (Средства проектирования и анализа фильтра) GUI FDATool (рис. 14.1).



Рис. 14.1. Окно Filter Design & Analysis Tool – [fwizdef.mat\*] при нажатой кнопке Design Filter: по требованиям к АЧХ (дБ) синтезирован БИХ-фильтр ПФ Золотарева—Кауэра (Elliptic)

<sup>&</sup>lt;sup>1</sup> Это необязательный этап, однако часто удобно иметь S-модель исходного ЦФ для ее сравнения с S-моделью ЦФ с ФТ.

- 3. Синтезируется ЦФ *(см. разд. 13.8.1)*.
- 4. Нажимается кнопка Set quantization parameters (Установить параметры квантования) на панели инструментов в нижнем левом углу, и в раскрывающемся списке Filter arithmetic (Тип арифметики) выбирается значение Double-precision floating-point (Плавающая точка с двойной точностью).
- Нажимается кнопка Store Filter (Сохранить фильтр) в группе Current Filter Information для сохранения синтезированного ЦФ в буфере Filter Manager (Диспетчер фильтров).

Импорт исходного ЦФ из Workspace выполняется следующим образом:

1. ЦФ, синтезированный средствами блока Digital Filter Design (см. разд. 13.8.1), GUI FDATool<sup>1</sup> или программными средствами MATLAB в виде объекта dfilt со свойством Arithmetic: 'double', загружается в Workspace по команде MATLAB:

load <имя объекта dfilt>

- Выбирается команда File | Import Filter from Workspace (Файл | Импорт из Workspace), и в окне Filter Design & Analysis Tool – [fwizdef.mat\*] появляется группа Filter Coefficients (Коэффициенты фильтра).
- 3. В группе Filter Coefficients в раскрывающемся списке Filter Structure (Структура фильтра) выбирается Filter object (Фильтр как объект).
- 4. В раскрывающемся списке Units (Единицы измерения) выбирается Normalized (0 to 1) (по умолчанию).

Выбор других единиц измерения частот целесообразно производить *после* импорта Ц $\Phi$ , в противном случае последует запрос о частоте дискретизации **Fs** из Workspace.

5. Нажимается кнопка Import Filter (Импорт фильтра).

Далее исходный ЦФ экспортируется в окно S-моделей, о чем речь пойдет в *разд. 14.2.3*. В том случае, если это БИХ-фильтр с каскадной структурой, *перед* моделированием его структуры с ФТ, которое рассматривается в *разд. 14.2.4*, выполняется расстановка звеньев и масштабирование (см. разд. 14.2.2).

#### Пример 14.1

На рис. 14.1 приведены требования к АЧХ (дБ) БИХ-фильтра ПФ Золотарева— Кауэра (Elliptic). В группе **Current Filter Information** выведена информация о синтезированном ЦФ.

```
save <имя объекта dfilt>.
```

<sup>&</sup>lt;sup>1</sup> В этом случае синтезированный ЦФ предварительно экспортируется в Workspace в виде объекта dfilt, а затем по команде меню File | Export сохраняется на диске по команде:

# 14.2.2. Расстановка звеньев и масштабирование в каскадных структурах БИХ-фильтров

Если исходный БИХ-фильтр имеет каскадную структуру из звеньев 2-го порядка (см. SOS в табл. 13.2), то *до* моделирования структуры ЦФ с ФТ выполняются следующие две операции [22]:

□ расстановка звеньев — для минимизации собственных шумов в структуре ЦФ с ФТ.

Звенья формируются посредством объединения полюсов с ближайшими нулями и расставляются в порядке возрастания радиусов полюсов;

□ масштабирование — для минимизации или предотвращения переполнений на выходах сумматоров в структуре ЦФ с ФТ.

Перед выполнением этих операций рекомендуется загрузить исходный ЦФ из буфера Filter Manager и вывести параметры передаточной функции (13.13), соответствующей каскадной структуре, по команде Analysis | Filter Coefficient (Анализ | Коэффициенты фильтра). В рабочей области окна для просмотра данных параметров информация выводится по звеньям: Section #1, Section #2 и т. д., и в конце — коэффициент усиления Output Gain на выходе структуры ЦФ.

После этого приступают к расстановке звеньев и масштабированию, обращаясь к команде Edit | Reorder and Scale Second-Order Sections (Редактирование | Расстановка и масштабирование звеньев 2-го порядка).

Открывается окно Reordering and Scaling of Second-Order Sections, включающее в себя (рис. 14.2):

□ группу **Reordering** (Расстановка) — со средствами формирования и расстановки звеньев.

Расстановка звеньев, минимизирующая собственные шумы, производится автоматически при выборе переключателя Auto;

□ группу Scaling (Масштабирование) — со средствами масштабирования.

Средства масштабирования отображают параметры функции MATLAB scale и включают в себя:

- флаг Scale (Масштаб) активизирующий средства группы Scaling;
- бегунок для выбора нормы, на основе которой выполняется масштабирование;
- поле ввода Maximum Numerator (Максимальное значение числителя) максимально допустимое по модулю значение коэффициентов числителя передаточных функций звеньев *после* масштабирования;
- раскрывающийся список Numerator Constraint (Ограничение числителя) ограничение коэффициентов числителя по модулю с возможными значениями:
  - None ограничения отсутствуют, масштабирование выполняется по умолчанию;

- Unit коэффициент числителя при нулевой степени приравнивается к единице, а остальные коэффициенты пропорционально пересчитываются;
- Normalize максимальному по модулю коэффициенту числителя каждого звена присваивается значение Maximum Numerator, а остальные коэффициенты нормируются к этому значению;
- Power of Two максимальному по модулю коэффициенту числителя каждого звена по умолчанию присваивается значение степени двойки, а остальные коэффициенты нормируются к этому значению;
- раскрывающийся список **Overflow Mode** (Режим переполнения) для выбора режима с возможными значениями Saturate и Wrap, о которых речь пойдет далее *в разд. 14.2.4.3*;
- раскрывающийся список Scale Value Constraint (Ограничение коэффициента усиления) — ограничение коэффициентов усиления с возможными значениями:
  - None ограничения отсутствуют, масштабирование выполняется по умолчанию;
  - Unit максимальный коэффициент усиления равен единице.

Этому значению соответствует традиционный расчет коэффициентов усиления, когда умножители на входе *текущего* звена учитываются на входе *предыдущего* звена, после чего пересчитываются коэффициенты числителей звеньев [22].

В этом случае в структуре Direct-Form II, SOS коэффициенты усиления на *входах* всех звеньев, кроме *первого*, оказываются равными *единице*, а в структуре Direct-Form I, SOS — все, кроме *последнего* [22];

- Power of Two нормируются к степени двойки по умолчанию;
- поле ввода Max Scale Value (Максимальный коэффициент усиления) максимально допустимое значение коэффициентов усиления звеньев; блокируется при выборе Scale Value Constraint: Unit;
- кнопку Apply (Применить) для применения выбранных установок без закрытия окна Reordering and Scaling of Second-Order Sections; полученный результат можно проверить, обращаясь к команде Analysis | Filter Coefficient, и при необходимости изменить установки;
- □ кнопку **Revert to Original Filter** (Возврат к начальному фильтру) для возврата в начальное состояние;
- □ кнопку OK для применения выбранных установок с закрытием окна Reordering and Scaling of Second-Order Sections.

None     None     None     None     Not     Least selective section to most selective section     Most selective section to least selective section     Custom reordering	Scaling Scale H Linf L2	L1 lin I l	f ⊧ st SNF
Numerator Order: [1:3]	Maximum Numerator:	1	
Denominator Order	Numerator Constraint:	Normalize	~
Use Numerator Order     Specify [1:3]	Overflow Mode:	Wrap	~
Scale Values Order	Scale Value Constraint:	Unit	~
Use Numerator Order     Specify     [1:4]	Max Scale Value:	2	

Рис. 14.2. Окно Reordering and Scaling of Second-Order Sections с выполненными установками

Более подробную справку можно получить с помощью справочной системы MATLAB в формате HTML, используя поиск по ключевой фразе "Scaling Second-Order Filters" (Масштабирование в фильтрах со звеньями 2-го порядка).

#### Пример 14.2

Выполним расстановку звеньев в БИХ-фильтре ПФ Золотарева—Кауэра с каскадной структурой Direct-Form II, SOS из 4-х звеньев (см. пример 14.1).

Для наглядного отображения информации *одновременно обо всех звеньях* экспортируем исходный ЦФ в Workspace, для чего необходимо произвести следующие действия:

- 1. Выполнить команду File | Export.
- 2. В раскрывающемся списке Export to выбрать MAT-File.
- 3. В раскрывающемся списке Export as выбрать Objects.
- 4. В группе Variable Names в поле ввода Discrete Filter указать имя объекта dfilt укажем H\_IIR.
- 5. Нажать кнопку Export, после чего откроется окно Export to a MAT-File.

Если ЦФ экспортируется в текущую папку, то в поле Имя файла продублировать имя объекта dfilt — H\_IIR.

Затем выведем значения s, G — соответственно матрицу коэффициентов (13.14) и вектор коэффициентов усиления передаточной функции<sup>1</sup> (13.13)

<sup>&</sup>lt;sup>1</sup> См. структуру Direct-Form II, SOS в табл. 13.2.

объекта н\_IIR. В объекте dfilt матрица коэффициентов описывается свойством sosMatrix, а вектор коэффициентов усиления — свойством ScaleValues. Элементы вектора scaleValues равны коэффициентам усиления на *входе* звеньев, а последний элемент — коэффициенту усиления на *выходе* последнего звена (на выходе структуры исходного ЦФ). Выведенные значения s, G объекта н представлены на рис. 14.3.

```
>> load H IIR
>> H IIR
H IIR =
         FilterStructure: 'Direct-Form II, Second-Order Sections'
              Arithmetic: 'double'
               sosMatrix: [4x6 double]
             ScaleValues: [5x1 double]
        PersistentMemory: false
>> s=H IIR.sosMatrix
S =
   1.0000
             -1.1655
                         1.0000
                                   1.0000
                                            -0.8915
                                                        0.9416
    1.0000
                                                        0.9345
             0.3501
                         1.0000
                                   1.0000
                                            0.0126
    1.0000
             -1.6041
                         1.0000
                                   1.0000
                                            -0.6356
                                                        0.8056
    1.0000
              1.0947
                         1.0000
                                   1.0000
                                             -0.2179
                                                        0.7950
>> G=H IIR.ScaleValues
G =
    0.0519
    0.9706
    0.7229
    0.4316
    1.0000
```

Рис. 14.3. Матрица коэффициентов s и вектор коэффициентов усиления G БИХ-фильтра ФНЧ Золотарева—Кауэра (объекта н\_\_IIR) до масштабирования

На рис. 14.2 представлены установки для расстановки звеньев и масштабирования БИХ-фильтра ПФ Золотарева—Кауэра (см. рис. 14.1). Выбор значения Normalize в раскрывающемся списке Numerator Constraint обусловлен тем, что ряд коэффициентов числителя звеньев превосходит единицу (см. рис. 14.3). Сохраним ЦФ с новыми коэффициентами в буфере Filter Manager.

После масштабирования экспортируем ЦФ в Workspace в виде объекта dfilt с именем H\_IIR\_scale и выведем новые значения матрицы коэффициентов s\_scale и вектора коэффициентов усиления G\_scale (рис. 14.4) (сравните со значениями на рис. 14.3).

>> load H_IIR_scale							
>> s_	scale=H_	IIR_scale.s	sosMatrix				
s_sca	le =						
0	.8580	-1.0000	0.8580	1.0000	-0.8915	0.9416	
1	.0000	0.3501	1.0000	1.0000	0.0126	0.9345	
0	.6234	-1.0000	0.6234	1.0000	-0.6356	0.8056	
0	.9135	1.0000	0.9135	1.0000	-0.2179	0.7950	
>> G_	scale=H_	IIR_scale.S	ScaleValues	3			
G_sca	le =						
0	.0519						
1	.0000						
1	.0000						
1	.0000						
0	.6197						

Рис. 14.4. Матрица коэффициентов s\_scale и вектор коэффициентов усиления G\_scale БИХ-фильтра ПФ Золотарева—Кауэра (объекта H\_IIR\_scale) после масштабирования

## 14.2.3. Создание S-модели исходного ЦФ

*S-модель исходного ЦФ* создается автоматически в результате экспорта исходного ЦФ из блока Filter Realization Wizard.

Экспорт ЦФ выполняется следующим образом:

- 1. Исходный ЦФ загружается из буфера Filter Manager (выделяется курсором в том случае, если там хранится несколько ЦФ).
- 2. На панели инструментов в нижнем левом углу нажимается кнопка Realize Model (Реализовать модель) или выбирается команда меню File | Export to Simulink Model (Файл | Экспорт в модель Simulink).

После этого окно Filter Design & Analysis Tool – [fwizdef.mat\*] примет вид, представленный на рис. 14.5.

- 3. В группе **Model** (Модель) в поле ввода **Block name** (Имя блока) указывается имя S-модели исходного ЦФ.
- 4. В раскрывающемся списке **Destination** (Расположение) выбирается имя окна Sмоделей, в которое будет экспортироваться S-модель исходного ЦΦ:
  - Current (Текущее) при экспорте в текущее (открытое) окно S-моделей;
  - New (Новое) при экспорте в новое, автоматически открываемое окно Sмоделей с именем Untitled;
  - User defended (Выбранное пользователем) при экспорте в уже существующее окно S-моделей, имя которого указывается пользователем.


Рис. 14.5. Окно Filter Design & Analysis Tool – [fwizdef.mat\*] при нажатой кнопке Realize Model

При выборе значения **Destination**: Current или **Destination**: User defended активизируется флаг **Overwrite generated '<Block name>' block** (Перезапись нового блока поверх старого блока с именем **<Block name>**). При установке флага S-модель ЦФ с именем, заданным в поле ввода **Block name**, заменит имеющуюся S-модель с тем же именем.

5. Выбирается состояние флага **Build model using basic elements** (Построить модель, используя базовые элементы).

При установке флага Build model using basic elements S-модель исходного ЦФ будет представлена в виде виртуальной подсистемы, отображающей структурную схему исходного ЦФ (см. разд. 13.6) с портами входа/выхода. При этом активизируются флаги в группе Optimization (Оптимизация):

- Optimize for zero gains (Оптимизировать нулевые умножители) при установке флага в структуре исходного ЦФ будут удалены умножители на ноль вместе с соединениями;
- Optimize for unity gains (Оптимизировать единичные умножители) при установке флага в структуре исходного ЦФ будут удалены умножители на единицу с сохранением соединений;

- Optimize for negative gains (Оптимизировать умножители с отрицательным коэффициентом умножения) — при установке флага в структуре исходного ЦФ будут удалены умножители на минус единицу с сохранением соединений и заменой в сумматоре операции сложения операцией вычитания (инверсии знака);
- Optimize delay chains (Оптимизировать цепи задержки) при установке флага в структуре исходного ЦФ линии с N единичными задержками z<sup>-1</sup> будут заменены задержками z<sup>-N</sup>.

При *сбросе* флага **Build model using basic elements** S-модель исходного Ц $\Phi$  будет представлена в виде *блока Digital Filter*, а флаги в группе **Optimization** блокированы.

6. Нажимается нижняя кнопка **Realize Model** — в окно S-моделей с указанным именем экспортируется S-модель исходного ЦФ.

#### Пример 14.3

Создадим S-модель исходного ЦФ экспортируем БИХ-фильтр ПФ Золотарева—Кауэра (см. пример 14.2).

На рис. 14.5 представлено окно при нажатой кнопке Realize Model с выполненными установками.

После нажатия нижней кнопки **Realize Model** в окно S-моделей с именем **Filters** экспортируется S-модель исходного ЦФ с именем Filter1\_IIR (рис. 14.6) в виде *виртуальной подсистемы*. Двойной щелчок левой кнопки мыши на пиктограмме блока откроет окно подсистем со *структурной схемой* исходного ЦФ.



Рис. 14.6. Окно S-моделей Filters с S-моделями исходных ЦФ (Filter1\_IIR — БИХ-фильтр ПФ Золотарева—Кауэра в виде виртуальной подсистемы со структурной схемой; Filter2\_IIR — БИХ-фильтр ПФ Золотарева—Кауэра в виде блока Digital Filter Design; Filter3\_FIR — КИХ-фильтр ФНЧ в виде блока Digital Filter Design) и S-моделями ЦФ с ФТ (Filter1\_IIR\_Fix — БИХ-фильтр ПФ Золотарева—Кауэра в виде виртуальной подсистемы со структурной схемой с ФТ; Filter2\_IIR\_Fix — БИХ-фильтр ПФ Золотарева—Кауэра в виде блока Digital Filter; Filter3\_FIR\_Fix — БИХ-фильтр ПФ Золотарева—Кауэра в виде блока Digital Filter; Filter3\_FIR\_Fix — БИХ-фильтр ПФ Золотарева—Кауэра в виде блока Digital Filter; Filter3\_FIR\_Fix — КИХ-фильтр ФНЧ в виде блока Digital Filter; Filter3\_FIR\_Fix — КИХ-фильтр ФНЧ

## 14.2.4. Моделирование структуры ЦФ с ФТ

Моделирование *структуры*  $\mu \Phi c \Phi T$  производится в окне Filter Design & Analysis Tool – [fwizdef.mat\*] на основе исходного  $\mu \Phi$  следующим образом:

- 1. Исходный ЦФ загружается из буфера Filter Manager (для БИХ-фильтров с каскадной структурой — после масштабирования).
- 2. Нажимается кнопка Set quantization parameters (Установка параметров квантования) на панели инструментов в нижнем левом углу.

Нижняя половина окна будет содержать раскрывающийся список Filter Arithmetic (Тип арифметики).

3. Выбирается значение Filter Arithmetic: Fixed-point (Фиксированная точка).

Нижняя половина окна Filter Design & Analysis Tool – [fwizdef.mat\*] изменится — в правой части появятся три вкладки (рис. 14.7):

- 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 при выборе Arithmetic: 'fixed'.

- 4. Устанавливаются требуемые свойства ЦФ с ФТ на вкладках Coefficients, Input/Output и Filter Internals (см. разд. 14.2.4.1—14.2.4.3).
- 5. Нажимается кнопка **Apply** (Применить), после чего будет выполнено моделирование ЦФ с ФТ.
- 6. Анализируются характеристики ЦФ с ФТ с помощью команд пункта меню Analysis (см. разд. 12.3).

В соответствующей области окна одновременно выводятся графики характеристик ЦФ с ФТ (непрерывной линией) и исходного ЦФ (пунктирной линией).

Для размещения легенды следует выполнить команду View | Legend (Вид | Легенда)

В результате квантования коэффициентов передаточной функции АЧХ ЦФ изменяется, поэтому необходимо *контролировать выполнение требований к* АЧХ.

### 14.2.4.1. Установка свойств ЦФ с ФТ на вкладке Coefficients

На вкладке **Coefficients** устанавливаются свойства коэффициентов передаточной функции в структуре ЦФ с  $\Phi$ T, соответствующие свойствам объекта dfilt при выборе Arithmetic: 'fixed'.

Список свойств *КИХ-фильтров с ФТ* включает в себя (в скобках указывается тождественное свойство объекта dfilt):

- □ Filter precision (Точность фильтра) точность представления коэффициентов в формате слова, принимающая значения (свойство объекта dfilt orcytctByet):
  - Full максимально возможная;
  - Specify all произвольная;
- □ Numerator word length (Длина слова для числителя) длина слова для коэффициентов (свойство CoeffWordLength);
- □ Best-precision fraction length (Длина дробной части при наилучшей точности) флаг управления длиной дробной части слова для коэффициентов (свойство CoeffAutoScale).

При установке флага длина дробной части рассчитывается автоматически для представления коэффициентов с максимальной точностью без переполнений;

- □ Numerator frac. length (Длина дробной части числителя) длина дробной части слова для коэффициентов (свойство NumFracLength);
- □ Numerator range (+/-) (Диапазон значений для числителя) положительное число, определяющее диапазон, в котором автоматически определяется длина дробной части слова для коэффициентов (свойство объекта dfilt orcytctвует).

Например, при **Numerator range:** 1 длина дробной части будет автоматически выбираться, исходя из диапазона [-1; 1]<sup>1</sup>;

Use unsigned representation (Использовать беззнаковое представление) — флаг управления знаковым/беззнаковым представлением коэффициентов (свойство Signed).

При установке флага коэффициенты представляются без знака (положительные), и старший бит считается значащим, а при сбросе — со знаком, где старший бит — знаковый;

Scale the numerator coefficients to fully utilize the entire dinamic range (Масштабировать коэффициенты в соответствии с заданным диапазоном) — флаг, при установке которого коэффициенты масштабируются так, чтобы в заданном формате обеспечить динамический диапазон, равный 1, при этом значения АЧХ

<sup>&</sup>lt;sup>1</sup> В МАТLАВ данные с ФТ представляются в диапазоне [-1; 1], а не [-1; 0,999...], как в ЦПОС с ФТ в дополнительном коде.

изменяются на множитель, равный отношению максимальных коэффициентов после и до масштабирования (свойство объекта dfilt orcytcrbyer).

Список свойств *БИХ-фильтров с*  $\Phi T$  включает в себя<sup>1</sup> (в скобках указывается тождественное свойство объекта dfilt):

- □ Coefficient word length (Длина слова для коэффициентов) длина слова для коэффициентов числителя и знаменателя (свойство CoeffWordLength);
- □ Best-precision fraction length аналогично одноименному свойству КИХ-фильтров (свойство CoeffAutoScale).

При сбросе флага **Best-precision fraction length** длина дробной части определяется следующими свойствами:

- □ Numerator frac. length аналогично одноименному свойству КИХ-фильтров для коэффициентов числителя (свойство NumFracLength);
- □ Numerator range (+/-) аналогично одноименному свойству КИХ-фильтров для коэффициентов числителя (свойство объекта dfilt orcytctвyet);
- □ Denominator frac. length (Длина дробной части знаменателя) длина дробной части слова для коэффициентов знаменателя (свойство DenFracLength);
- □ Denominator range (+/-) (Диапазон значений для знаменателя) аналогично свойству Numerator frac. length для коэффициентов знаменателя (свойство объекта dfilt отсутствует);
- Scale Values frac. length (Длина дробной части слова для коэффициентов усиления) длина дробной части слова для коэффициентов усиления в каскадной структуре (свойство ScaleValueFracLength);
- □ Scale Values range (+/-) (Диапазон значений для коэффициентов усиления) аналогично свойству Numerator frac. length для коэффициентов усиления;
- □ Use unsigned representation аналогично одноименному свойству КИХфильтров для коэффициентов числителя и знаменателя (свойство signed).

#### Пример 14.4

В исходном ЦФ — ЦФ БИХ-фильтре ПФ Золотарева—Кауэра (см. пример 14.2) для моделирования его структуры с ФТ установим свойства коэффициентов передаточной функции на вкладке **Coefficients** (рис. 14.7).

<sup>&</sup>lt;sup>1</sup> Особое внимание следует обратить на представление с ФТ коэффициентов *знаменателя* передаточной функции, превышающих единицу. В этом случае можно отвести один бит на представление целой части слова.





#### 14.2.4.2. Установка свойств ЦФ с ФТ на вкладке Input/Output

На вкладке Input/Output устанавливаются свойства входного и выходного сигналов ЦФ с ФТ, соответствующие свойствам объекта dfilt при выборе Arithmetic: 'fixed'.

Список свойств *КИХ-фильтров с ФТ* включает в себя (в скобках указывается тождественное свойство объекта dfilt):

- □ Filter precision (Точность фильтра) точность представления входного и выходного сигналов в формате слово; аналогично одноименному свойству для КИХ-фильтров (см. разд. 14.2.4.1);
- □ Input word length (Длина слова на входе) длина слова для значений входного сигнала (свойство InputWordLength);
- □ Input fraction length (Длина дробной части на входе) длина дробной части слова для значений входного сигнала (свойство InputFracLength);

- □ Input range (+/-) (Диапазон значений на входе) аналогично свойству Numerator frac. length для значений входного сигнала (см. разд. 14.2.4.1) (свойство объекта dfilt отсутствует);
- □ Output word length (Длина слова на выходе) длина слова для значений выходного сигнала (свойство OutputWordLength);
- Output fraction length (Длина дробной части на выходе) длина дробной части слова для значений выходного сигнала (свойство OutputFracLength);
- □ Output range (+/-) (Диапазон значений на выходе) аналогично свойству Numerator frac. length для входного сигнала (см. разд. 14.2.4.1) (свойство объекта dfilt отсутствует).

Список свойств *БИХ-фильтров с*  $\Phi T$  включает в себя (в скобках указывается тождественное свойство объекта dfilt):

- □ Input word length аналогично одноименному свойству КИХ-фильтров (свойство InputWordLength);
- □ Input fraction length аналогично одноименному свойству КИХ-фильтров (свойство InputFracLength);
- □ Input range аналогично одноименному свойству КИХ-фильтров (свойство объекта dfilt отсутствует);
- □ Output word length аналогично одноименному свойству КИХ-фильтров (свойство OutputWordLength);
- □ Avoid overflow (Предотвращение переполнения) флаг управления переполнением для выходного сигнала (свойство объекта dfilt отсутствует).

При сбросе флага предотвращение переполнений достигается путем установки длины дробной части значений выходного сигнала с помощью свойств **Output** fraction length и **Output** range (+/-), аналогичных одноименным свойствам КИХ-фильтров.

При установке флага длина дробной части устанавливается автоматически, исходя из условия минимизации ошибок при переполнении;

- □ Section input word length (Длина слова на входе звена) длина слова для значений сигнала на входе звеньев в каскадной структуре (свойство SectionInputWordLength);
- □ Avoid overflow флаг управления переполнением для значений сигнала на входе звеньев в каскадной структуре (свойство SectionInputAutoScale).

При сбросе флага 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) (свойство SectionInputFracLength).

При установке флага длина дробной части слова устанавливается автоматически, исходя из условия минимизации ошибок при переполнении;

- Section output word length (Длина слова на выходе звена) длина слова для значений сигнала на выходе звеньев в каскадной структуре — слова (свойство SectionOutputWordLength);
- □ Avoid overflow флаг управления переполнением для значений сигнала на выходе звеньев в каскадной структуре (свойство SectionOutputAutoScale).

При сбросе флага предотвращение переполнений достигается путем установки длины дробной части для значений сигнала на входе звеньев с помощью свойства Section output fraction length (Длина дробной части слова на выходе звена) (свойство SectionOutputFracLength).

При установке флага длина дробной части устанавливается автоматически, исходя из условия минимизации ошибок при переполнении.

#### Пример 14.5

В исходном ЦФ — БИХ-фильтре ПФ Золотарева—Кауэра (см. пример 14.2) для моделирования его структуры с ФТ установим свойства коэффициентов передаточной функции на вкладке **Input/Output** (рис. 14.8).



Рис. 14.8. Окно Filter Design & Analysis Tool – [fwizdef.mat\*] с установленными на вкладке Input/Output свойствами для входного и выходного сигналов БИХ-фильтра ПФ Золотарева—Кауэра

#### 14.2.4.3. Установка свойств ЦФ с ФТ на вкладке Filter Internals

На вкладке Filter Internals устанавливаются свойства арифметических операций в ЦФ с  $\Phi$ T, соответствующие свойствам объекта dfilt при выборе Arithmetic: 'fixed'.

Список свойств *КИХ-фильтров с ФТ* включает в себя (в скобках указывается тождественное свойство объекта dfilt):

- □ Filter precision (Точность фильтра) точность арифметических операций; аналогично одноименному свойству для КИХ-фильтров (см. разд. 14.2.4.1);
- □ Rounding mode (Режим округления) метод округления при представлении данных в заданном формате с возможными значениями (свойство RoundMode):
  - Zero округление в направлении нуля: выбирается значение ближайшего нижнего уровня квантования (тождественно усечению) (значение Round-Mode: 'fix');
  - Nearest округление до ближайшего целого: выбирается значение ближайшего уровня квантования, и если оно точно попадает на границу между соседними уровнями, берется значение ближайшего верхнего уровня (значение RoundMode: 'nearest');
  - Nearest (convergent) округление до ближайшего четного целого: выбирается значение ближайшего уровня квантования, и если оно попадает точно на границу между соседними уровнями квантования, то выбирается значение ближайшего уровня, соответствующего четному числу (значение Round-Mode: 'convergent');
  - Selling округление в направлении ∞: отрицательное число усекается, а для положительного берется значение ближайшего верхнего уровня квантования (значение RoundMode: 'ceil');
  - Floor округление в направлении –∞: положительное число усекается, а для отрицательного берется значение ближайшего нижнего уровня квантования (значение RoundMode: 'floor');
  - 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 (значение Round-Mode: 'round');
- Overflow Mode (Режим переполнения) режим переполнения при представлении данных в заданном формате, который принимает значения (свойство overflowMode):
  - Wrap модульная арифметика: при переполнении результат автоматически заменяется значением по модулю 2 (отрицательное число, меньшее либо равное –1, суммируется с ближайшим по модулю 2<sup>N</sup>, а из положительного, большего 1, вычитается ближайшее 2<sup>N</sup>) (значение Overflow-Mode: 'wrap').

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 (значение OverflowMode: 'saturate').

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 mode (Режим для произведения) формат при пересылке (сохранении) результата умножения (произведения) из регистра Product в формате *двойного слова*<sup>1</sup> MSB : LSB с возможными значениями (свойство ProductMode):
  - Specify all сохранение в произвольным формате, которое отображается свойствами Product word length и Product fraction length (значение ProductMode: 'SpecifyPrecision');
  - Full precision сохранение двойного слова MSB : LSB (значение ProductMode: 'FullPrecision');
  - Keep LSB сохранение младшего слова LSB (значение ProductMode: 'KeepLSB');
  - Кеер MSB сохранение старшего слова MSB (значение ProductMode: 'Кеермsb');
- □ Product word length (Длина слова для произведения) длина двойного слова MSB : LSB для регистра Product (свойство ProductWordLength);
- □ Product fraction length (Длина дробной части для произведения) длина дробной части двойного слова MSB : LSB для регистра Product (свойство ProductFracLength);
- □ Accum. mode (Режим для суммы) формат при пересылке (сохранении) результата сложения (суммы) из регистра Accumulator (Аккумулятор) в формате *расширенного слова*<sup>2</sup> EXT : MSB : LSB (свойство AccumMode); принимает такие же значения, как в свойстве **Product mode**;
- □ Accum. word length (Длина слова для суммы) длина расширенного слова EXT : MSB : LSB для регистра Accumulator (свойство AccumWordLength);
- □ Accum. fraction length (Длина дробной части слова для суммы) длина дробной части расширенного слова EXT : MSB : LSB для регистра Accumulator (свойство AccumFracLength).

 $<sup>^1</sup>$  В двойном слове обозначение MSB соответствует старшим битам (Most Significant Bits), а LSB — младшим битам (Least Significant Bits).

<sup>&</sup>lt;sup>2</sup> В расширенном слове обозначение EXT соответствует битам расширения (Extension) для хранения целой части суммы, или, если она равна нулю, расширения знака.

Список свойств *БИХ-фильтров с*  $\Phi T$  включает в себя (в скобках указывается тождественное свойство объекта dfilt):

- □ Rounding mode аналогично одноименному свойству КИХ-фильтов (свойство RoundMode);
- □ Overflow Mode аналогично одноименному свойству КИХ-фильтов (свойство OverflowMode);
- Product mode аналогично одноименному свойству КИХ-фильтов (свойство ProductMode);
- □ Product word length аналогично одноименному свойству КИХ-фильтов (свойство ProductWordLength);
- □ Num. frac. length длина дробной части для локальных произведений в нерекурсивной части разностного уравнения (свойство NumProdFracLength);
- □ Den. frac. Length длина дробной части для локальных произведений в рекурсивной части разностного уравнения (свойство DenProdFracLength);
- □ Accum. mode аналогично одноименному свойству КИХ-фильтов (свойство AccumMode);
- □ Num. frac. length длина дробной части для суммы в нерекурсивной части разностного уравнения (свойство NumAccumFracLength);
- □ Den. frac. length длина дробной части для суммы в рекурсивной части разностного уравнения (свойство DenAccumFracLength);
- □ Cast signals before accum. (Отбрасываемые сигналы перед суммированием) флаг управления форматом слагаемых перед суммированием (свойство CastBeforeSum).

При установке флага слагаемые представляются в формате расширенного слова, а при сбросе сохраняют формат предыдущей операции.

Следующие свойства зависят от структуры БИХ-фильтра с ФТ.

Для каскадной структуры Direct-Form II, SOS (см. табл. 13.2) определяются свойства:

- □ State word length (Длина слова для состояния) длина слова для элементов задержки (ячеек памяти) (свойство StateWordLength);
- □ State fraction length (Длина дробной части для состояния) длина дробной части слова для элементов задержки (свойство StateFracLength).

Для каскадной структуры Direct-Form I, SOS (см. табл. 13.2) определяются свойства:

- □ Num. state word length (Длина слова для состояния числителя) длина слова для элементов задержки воздействия (свойство NumStateWordLength);
- □ Den. state word length (Длина слова для состояния знаменателя) длина слова для элементов задержки реакции (свойство DenStateWordLength);
- □ Num. fraction length (Длина дробной части числителя) длина дробной части слова для элементов задержки воздействия (свойство NumStateFracLength);

□ Den. fraction length (Длина дробной части знаменателя) — длина дробной части слова для элементов задержки воздействия (свойство DenStateFracLength).

Со свойствами для каскадных структур Direct-Form I Transposed, SOS и Direct-Form II Transposed, SOS (см. табл. 13.2) предлагаем познакомиться самостоятельно.

Для более подробной информации о свойствах ЦФ с ФТ следует обратиться к справочной системе MATLAB по Simulink в формате HTML к разделу Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Filtering  $\rightarrow$  Filter Designs  $\rightarrow$  Digital Filter и далее — к гипертекстовой ссылке Filter Structure Diagrams (Структурные схемы фильтров).

#### Пример 14.6

В исходном ЦФ — БИХ-фильтре ПФ Золотарева—Кауэра (см. пример 14.2) для моделирования его структуры с ФТ установим свойства арифметических операций на вкладке Filter Internals (рис. 14.9).

🛃 Fi	Iter Design & Analysis Tool - [	[fwizdef.mat *]	
Ele	<u>E</u> dit <u>A</u> nalysis Targets View <u>W</u> in	ndow <u>H</u> elp	
		[1] III N N 2 # * 1 - III N 2	
	- Current Filter Information - Structure: Direct-Form II, Second-Order Section: Order: 8 Sections: 4 Stable: Yes Source: Designed (quantized)  Store Filter Filter Manager	Magnitude Response	
	Filter arithmetic: Fixed-point	Coefficients   Input/Output   Filter Inter	rnals [
	Rounding mode	e: Nearest 🗸 Overflow Mode: Wrap 🗸	
	Product mode: Specify all Product word length: 16 Num. fraction length: 14 Den. fraction length: 14	Accum. mode:       Specify all       State word length:       8         Accum. word length:       16       16         Num. fraction length:       12       State fraction length:       7         Image: Cast signals before accum.       5       5	
Read	y	Apply	

Рис. 14.9. Окно Filter Design & Analysis Tool – [fwizdef.mat\*] с установленными на вкладке Filter Internals свойствами арифметических операций БИХ-фильтра ПФ Золотарева—Кауэра

#### Пример 14.7

Экспортируем ЦФ с ФТ — БИХ-фильтр ПФ Золотарева—Кауэра с ФТ (см. свойства на рис. 14.7—14.9) в Workspace (см. разд. 14.2.2) в виде объекта dfilt с именем H\_IIR\_quantize. Выведем квантованные значения матрицы коэффициентов s\_quantize и вектора коэффициентов усиления G\_quantize (рис. 14.10, сравните с неквантованными коэффициентами на рис. 14.4).

>> s_quantize=H_IIR_quantize.sosMatrix						
s_quantize	=					
0.8594	-1.0000	0.8594	1.0000	-0.8906	0.9453	
0.9922	0.3516	0.9922	1.0000	0.0156	0.9375	
0.6250	-1.0000	0.6250	1.0000	-0.6328	0.8047	
0.9141	0.9922	0.9141	1.0000	-0.2188	0.7969	
>> G_quanti	ze= <b>H_IIR_qu</b>	antize.Sca	leValues			
G_quantize	=					
0.0547						
1.0000						
1.0000						
1.0000						
0.6172						



## 14.2.5. Создание Ѕ-модели ЦФ с ФТ

*S-модель* Ц $\Phi c \Phi T$  создается автоматически в результате экспорта Ц $\Phi c \Phi T$  из блока Filter Realization Wizard, который выполняется точно так же, как экспорт исходного Ц $\Phi$  (см. разд. 14.2.3).

Экспортированная S-модель ЦФ с ФТ будет представлена одной из следующих разновидностей *виртуальной подсистемы*:

□ структурой ЦФ с ФТ — при установке флага Build model using basic elements;

□ соединением блоков Data Type Conversion (см. разд. 7.2.2) и Digital Filter — при сбросе флага Build model using basic elements.

Более подробную информацию можно получить с помощью системы MATLAB по Simulink в формате HTML в разделе Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Filtering  $\rightarrow$  Filter Designs.

#### Пример 14.8

Создадим S-модель ЦФ с ФТ — экспортируем БИХ-фильтр ПФ Золотарева— Кауэра с ФТ (см. примеры 14.4—14.6) при установке флага **Build model using basic**  elements в окне S-моделей Filters в виде *виртуальной подсистемы* с именем Filter1\_IIR\_Fix (см. рис. 14.6). Двойной щелчок на пиктограмме блока откроет окно подсистем со структурной схемой ЦФ с ФТ.

# 14.3. Создание S-модели ЦФ с ФТ в виде блока Digital Filter

S-модель ЦФ с ФТ может быть создана в виде блока *Digital Filter* при соответствующей установке его параметров.

Список параметров блока Digital Filter зависит от выбора одного из трех указателей в группе **Coefficient source** (Источник коэффициентов) (для экономии места окно с параметрами приведено далее на рис. 14.12—14.13):

1. **Dialog Parameters** (Параметры диалога) — при экспорте *исходного* ЦФ в виде блока Digital Filter для последующего создания на его основе ЦФ с ФТ.

Исходный ЦФ удобно создавать в виде блока Digital Filter Design (см. *разд.* 13.8.1), а затем экспортировать в виде блока Digital Filter, выполняя те же действия, что и при экспорте из блока Filter Realization Wizard<sup>1</sup> (см. *разд.* 14.2.3).

2. Input port(s) (Входной порт(ы)) — при экспорте коэффициентов передаточной функции исходного ЦФ на автоматически создаваемые входы для входного сигнала (In), коэффициентов числителя (Num) и знаменателя (Den) передаточной функции, что удобно при изменяющихся коэффициентах.

При выборе этого указателя параметры блока Digital Filter практически не отличаются от параметров при указателе **Dialog Parameters**, поэтому далее этот указатель не рассматривается.

3. Discrete-time filter object (DFILT) (ЦФ в виде объекта dfilt) — при экспорте ЦФ с ФТ в виде объекта dfilt со свойством Arithmetic: 'fixed' из Workspace.

При выборе этого указателя параметры блока Digital Filter отображают свойства объекта dfilt, поэтому далее этот указатель не рассматривается.

#### Пример 14.9

Создадим S-модель исходного ЦФ — БИХ-фильтра ПФ Золотарева—Кауэра (см. пример 14.2) — в виде блока Digital Filter Design *(см. разд. 13.8.1)* с именем Filter2\_IIR (см. рис. 14.6).

Экспортируем исходный ЦФ при *сбросе* флага **Build model using basic elements** *(см. разд. 14.2.3)* в виде блока Digital Filter с именем Filter2\_IIR\_Fix (см. рис. 14.6) для последующего создания S-модели ЦФ с ФТ путем установки параметров данного блока (см. пример 14.11).

374

<sup>&</sup>lt;sup>1</sup> Возможен экспорт непосредственно из блока Filter Realization Wizard или создание S-модели исходного ЦФ в виде блока Digital Filter на основе объекта dfilt (см. разд. 14.5).

#### Пример 14.10

Создадим S-модель исходного ЦФ — КИХ-фильтра ФНЧ (Equiripple), синтезированного по требованиям к АЧХ, представленным на рис. 14.11, — в виде блока Digital Filter Design *(см. разд. 13.8.1)* с именем Filter3\_FIR (см. рис. 14.6).



Рис. 14.11. Окно Block Parameters: Filter3\_FIR при нажатой кнопке Design Filter: по требованиям к АЧХ (дБ) синтезирован КИХ-фильтр ФНЧ

Экспортируем исходный ЦФ при *сбросе* флага **Build model using basic elements** *(см. разд. 14.2.3)* в виде блока Digital Filter с именем Filter3\_FIR\_Fix (см. рис. 14.6) для последующего создания S-модели ЦФ с ФТ путем установки параметров данного блока (см. пример 14.12).

# 14.4. Параметры блока Digital Filter

При выборе указателя Dialog Parameters параметры отображаются на двух вкладках — Main и Fixed-point.

На вкладке **Main** определяются параметры (для экономии места окно с параметрами приведено далее на рис. 14.12):

□ Transfer function type (Тип передаточной функции) — передаточная функция ЦФ:

- IIR (poles & zeros) БИХ-фильтра;
- IIR (all poles) БИХ-фильтра полюсного вида;
- FIR (all zeros) КИХ-фильтра;

□ Filter structure (Структура фильтра) — структура ЦФ (см. табл. 13.2).

Для каскадных структур БИХ-фильтров из звеньев 2-го порядка (биквадратных) добавлено определение "Biquad" (сокращение от Biquadratic).

С каждой структурой связан свой набор параметров исходного ЦФ, смысл которых поясняется в табл. 13.2;

□ Initial conditions (Начальные условия) — начальные условия, по умолчанию — нулевые (см. разд. 12.1).

На вкладке **Fixed-point** параметры указываются в двух группах (для экономии места окно с параметрами приведено далее на рис. 14.13):

- □ Fixed-point operational parameters (Рабочие параметры при фиксированной точке) с параметрами:
  - Rounding mode (Режим округления) метод округления (см. разд. 14.2.4.3);
  - Overflow mode (Режим переполнения) режим переполнения (см. разд. 14.2.4.3);
- □ Fixed-point data types (Типы данных с фиксированной точкой) параметры зависят от структуры ЦФ с ФТ и в целом включают в себя:
  - Section I/O (Вход/Выход звена) формат данных на входе/выходе звеньев в каскадных структурах БИХ-фильтров:
    - Same as input как у входного сигнала;
    - Binary point scaling соответствует формату fixdt(S,N,M) (см. табл. 4.6);
    - Slope and bias scaling соответствует формату fixdt(S,N,2^(-M),Bias) (см. табл. 4.6);
  - **Тар sum** (Сумма на ответвлении) формат для внутренних сумм в приведенных структурах КИХ-фильтров с такими же значениями, как у параметра **Section I/O**;
  - Multiplicand (Множимое) формат данных для множимого на коэффициент b<sub>0</sub> в структурах БИХ-фильтров Direct-Form I Transposed и Direct-Form I Transposed, SOS:
    - Same as output как у выходного сигнала;
    - остальные значения как у параметра Section I/O;
  - Coefficients (Коэффициенты) формат для коэффициентов передаточной функции:
    - Same word length as input как у входного сигнала;
    - Specify word length (Определить длину слова) соответствует формату fixdt(S,N) (см. табл. 4.6);
    - остальные значения как у параметра Section I/O;

- **Product output** (Выход умножителя) формат для произведения с такими же значениями, как у параметра Section I/O;
  - Accumulator (Аккумулятор) формат для суммы:
    - Same as product output как у произведения;
    - остальные значения как у параметра Section I/O;
  - State (Состояние) формат элементов линии задержки:
    - Same as accumulator как у аккумулятора;
    - остальные значения как у параметра Section I/O;
- **Output** (Выход) формат для выходного сигнала с такими же значениями, как у параметра **State**;
- Lock scaling against changes by the autoscaling tool *см. разд. 6.2.1.2.*

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML, обращаясь к разделу Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Filtering  $\rightarrow$  Filter Designs  $\rightarrow$  Digital Filter и далее — к гипертекстовой ссылке Filter Structure Diagrams.

Особое внимание следует обратить на представление с ФТ коэффициентов *знаменателя* передаточной функции, превышающих единицу. В этом случае можно отвести один бит на представление целой части слова.

Для анализа характеристик ЦФ и ЦФ с ФТ в окне параметров при открытой вкладке **Main** предусмотрена кнопка **View Filter Response**. При ее нажатии автоматически открывается окно **Filter Visualization Tool** программы GUI FVTool (см. разд. 12.3).

В результате квантования коэффициентов передаточной функции АЧХ ЦФ изменяется, поэтому необходимо контролировать выполнение требований к АЧХ.

#### Пример 14.11

Создадим S-модель ЦФ с ФТ — БИХ-фильтра ПФ Золотарева—Кауэра — в виде блока *Digital Filter* с именем Filter2\_IIR\_Fix (см. пример 14.9 и рис. 14.6) с параметрами, представленными на рис. 14.12—14.13.

Рис. 14.12. Параметры блока Digital Filter с именем Filter2\_IIR\_Fix (БИХ-фильтра ПФ) на вкладке Main

Function Blo	ock Parameters: Filter2_IIR_Fix	
Digital Filter		
ndependently fil coefficients using 'ou can also spec nformation abou	ter each channel of the input over time using a specified di g either tunable mask dialog parameters or separate input p ofy filters using discrete-time filter objects (dfilts) from the t creating these objects.	gital filter impli ports, which ar Signal Process
Coefficient source	ce	
Dialog paran	neters	
O Input port(s)	)	
O Discrete-time	e filter object (DFILT)	
O Discrete-time	e filter object (DFILT)	
Discrete-time	e filter object (DFILT) point	
Discrete-time Main Fixed-p Parameters	e filter object (DFILT) point	
Discrete-time Main Fixed-p Parameters Transfer function	e fliter object (DFILT) point	
Discrete-time	e fliter object (DFILT) point	
Discrete-time	e filter object (DFILT) point	31653477 1 .0
Discrete-time Main Fixed-p Parameters Transfer function Filter structure: SOS Matrix (Mx	e filter object (DFILT) point pon type: [IIR (poles & zeros) : Biquad direct form II (SOS) :6): \$.80558961749978;0.913527231653477 1 0.91352723	31653477 1 -0
Discrete-time Main Fixed-p Parameters Transfer function Filter structure: SOS Matrix (Mx Scale values: [[	e filter object (DFILT) point	31653477 1 -0
Discrete-time Main Fixed-p Parameters Transfer function Filter structure: SOS Matrix (Mx Scale values:	e filter object (DFILT) point	31653477 1 -0
Discrete-time Main Fixed-p Parameters Transfer function Filter structure: SOS Matrix (Mx Scale values: [[] Initial conditione	e filter object (DFILT) point point ype: IIR (poles & zeros) Biquad direct form II (SOS) (6): 0.0558961749978;0.913527231653477 1 0.91352723 0.0518525356128441;1;1;1;0.61969765575226] s: 0	31653477 1 -0

gitai riiter										
dependently fi efficients using u can also spe formation abou	Iter each channel of the i g either tunable mask dia cify filters using discrete- ut creating these objects	nput ov log pari -time filt	ier time smeteri ter obje	e using a spe s or separat ects (dflits) f	cofied digital filter imple e input ports, which are from the Signal Processi	mentation. Y useful for t ng Toolbox.	bu cz ime-v Type	n specify i arying coe "help dfilt"	fiter fficients. for more	
oefficient sour	ce									
Dialog parar	meters									
) Input port(s	a)									
Discrete-tim	e filter object (DFILT)									
ain Hoted-	point									_
ttings on this p	sane only apply when blo	ick input	ts are f	fixed-point si	ignals.					
Fixed-point ope	erational parameters									
Rounding mode	e: Nearest			~	Overflow mode: Wrap				~	
Fixed-point dat	la types									
	Mode		Signer	d Word leng	pth .	Fraction	ength			
			1.1	-		Input:	7			
section I/O	Binary point scaling	~	Yes	8		Output:	6			
						Num:		7		
	Binary point scaling	~	Yes	8		Den:		7	_	
Coefficients			2	-		Scale val	ues:	6	_	
Coefficients		~	Yes	16		14				
Coefficients Product output	Binary point scaling		-	16		12				
Coefficients Product output Accumulator	Binary point scaling Binary point scaling	~	res			10				
Coefficients Product output Accumulator State	Binary point scaling Binary point scaling Binary point scaling	~ ~	Yes	8		7				
Coefficients Product output Iccumulator Itate Dutput	Binary point scaling Binary point scaling Binary point scaling Binary point scaling	> > >	Yes Yes	8		7			_	

Рис. 14.13. Параметры блока Digital Filter с именем Filter2\_IIR\_Fix (БИХ-фильтра ПФ с ФТ) на вкладке Fixed-point

#### Пример 14.12

Создадим S-модель ЦФ с ФТ — КИХ-фильтра ФНЧ — в виде блока Digital Filter с именем Filter3\_FIR\_Fix (см. пример 14.10 и рис. 14.6) с параметрами, представленными на рис. 14.14—14.15.

Digital Filter         Independently filter each channel of the input over time using a specified digtal filter of filter other threads dialog parameters or separate input ports, coefficients.           You can also specify filters using discrete-time filter objects (difts) from the Signal Promore information about creating these objects.           Coefficient source           O Dialog parameters           Input port(s)           D biscrete-time filter object (DFILT)           Main         Fixed-point           Parameters           Transfer function type:         FIR (all zeros)           Filter structure:         Direct form symmetric           Numerator coefficients:         [0428023803219045-0.0144338672556362-0.0154417474	Fun	ction Block Parameters: Filter3_FIR_Fix	Ð
Independently filter each channel of the input over time using a specified digital filter i filter coefficients using either tunable mask dialog parameters or separate input ports, coefficients. You can also specify filters using discrete-time filter objects (difits) from the Signal Pro- more information about creating these objects. Coefficient source O Dalog parameters I pout port(s) O Discrete-time filter object (DFILT) Main Fixed-point Parameters Transfer function type: [FR (all zeros) Filter structure: [Drect form symmetric Numerator coefficients: [0/428032803219045-0.0144338672556362-0.0154417474 Initial conditions: [0	Digital	Fiter	1
Coefficient source  Disource  Disource  Disource  Birewed spont Parameters  Transfer function type: [FIR (all zeros)  Filter structure: [Direct form symmetric Numerator coefficients: [00428032803219045-0.0144338672556362-0.015447474 Initial conditions: [0]	Indepe filter co coeffici You car more in	ndently filter each channel of the input over time using a specified digt efficients using either tunable mask dalog parameters or separate rep ents. n also specify filters using discrete-time filter objects (dfits) from the Sig formation about creating these objects.	al filter i ut ports, gnal Pro
Dialog parameters     Input port(s)     Discrete-time filter object (DFILT)  Main Fixed-point Parameters  Transfer function type: [FIR (all zeros)  Filter structure: [Direct form symmetric Numerator coefficients: [00428032803219045-0.0144338672556362-0.015447474 Initial conditions: [0]	Coeffi	dent source	
Input port(s)     Discrete-time filter object (DFILT)  Main Fixed-point Parameters  Transfer function type: [FIR (all zeros)  Filter structure: [Direct form symmetric Numerator coefficients: [00428032803219045-0.0144338672556362-0.0154417474 Initial conditions: [0]	O Dia	alog parameters	
Discrete-time filter object (DFB/T)           Main         Exced-spont           Parameters         Transfer function type: [FIR (all zeros)           Filter structure: [Direct form symmetric         Numerator coefficients: [00428032803219045-0.0144338672556362-0.015447474           Initial conditions: [0         0	Oin	put port(s)	
Main         Fixed-point           Parameters         Parameters           Transfer function type:         [FIR (all zeros)]           Filter structure:         Direct form symmetric           Numerator coefficients:         [00428032803219045-0.0144338672556362-0.015447474]           Initial conditions:         [0	OD	screte-time filter object (DFILT)	
Main         Fixed-spont           Parameters         Transfer function type: [FIR (all zeros)           Filter structure: [Direct form symmetric           Numerator coefficients:         [00428032803219045-0.0144338672556362-0.015447474           Initial conditions:         [0	-		
Parameters Transfer function type: [FIR. (all zeros) Filter structure: [Direct form symmetric Numerator coefficients: [00428032803219045-0.0144338672556362-0.015447474 Initial conditions: [0]	Main	Fixed-point	
Transfer function type: [FIR (all zeros) Filter structure: [Direct form symmetric Numerator coefficients: [00428032803219045-0.0144338672556362-0.015447474 Initial conditions: [0]	Param	ieters -	
Filter structure:         Direct form symmetric           Numerator coefficients:         [00428032803219045 -0.0144338672556362 -0.015447474           Initial conditions:         [0	Trans	ifer function type: FIR (all zeros)	
Numerator coefficients: 00428032803219045 -0.0144338672556362 -0.015447474 Initial conditions: 0	Filter	structure: Direct form symmetric	
Initial conditions: 0	Norma	water coefficients: 00428032803219045 -0.0144338622556362 -0.015	447474
Inital conditions: 0			
		conditions: 0	
	Initial		
	Initial		
	Inital		

Рис. 14.14. Параметры блока Digital Filter с именем Filter3\_FIR\_Fix (КИХ-фильтра ФНЧ) на вкладке Main

ndependently f oefficients usin	iter each channel of the input o g either tunable mask dialog par	ver tim ameter	e using a specified digita s or separate input port	I filter implementation. You can specify fil s, which are useful for time-varying	lter
oefficients. Iou can also spe	cify filters using discrete-time fil	ter obi	ects (dfilts) from the Sig	nal Processing Toolbox, Type "help dfilt" f	for
ore information	about creating these objects.				
Coefficient sour	ce				
Dialog para	meters				
O Input port(	5)				
O Discrete-tim	e filter object (DFILT)				
Main Pixed-	point				
ettings on this ;	pane only apply when block inpu	ts are t	fixed-point signals.		
Fixed-point op	arational parameters				
Rounding mode	e: Nearest		Vverflow mode	Wrap	*
Fixed-point dat	ta types				
	Mode	Signe	d Word length	Fraction length	
Tap sum	Binary point scaling	Yes	8	6	]
	Binary point scaling	Yes	8	7	1
Coefficients	And a second s			14	]
Coefficients Product output	Binary point scaling	Yes	16		
Coefficients Product output Accumulator	Binary point scaling	Yes	16	12	
Coefficients Product output Accumulator Output	Binary point scaling Binary point scaling Binary point scaling	Yes Yes Yes	16 16 8	12	
Coefficients Product output Accumulator Output Lock scaling	Binary point scaling Binary point scaling Binary point scaling against changes by the autosc	Yes Yes Yes aling to	16 16 8 00	12	
Coefficients Product output Accumulator Output Lock scaling	Binary point scaling Binary point scaling Binary point scaling against changes by the autosc	Yes Yes Yes aling to	16 16 8 00	7	

Рис. 14.15. Параметры блока Digital Filter с именем Filter3\_FIR\_Fix (КИХ-фильтра ФНЧ с ФТ) на вкладке Fixed-point

# 14.5. Создание S-моделей исходного ЦФ и ЦФ с ФТ программными средствами MATLAB

S-модель исходного ЦФ может создаваться в окне S-моделей с именем Untitled на основе объекта dfilt со свойством Arithmetic: 'double' в двух видах:

Виртуальная подсистема со структурной схемой ЦФ — по команде:

realizemdl (<имя объекта dfilt>)

2. Блок Digital Filter — по команде:

```
block (<имя объекта dfilt>)
```

*S-модель ЦФ с ФТ* может создаваться на основе объекта dfilt со свойством Arithmetic: 'fixed' в виде виртуальной подсистемы двух разновидностей:

- 1. Структурная схема ЦФ с ФТ по команде realizemdl;
- 2. Соединение блоков Data Type Conversion (см. разд. 7.2.2) и Digital Filter по команде block.

При создании S-модели ЦФ с ФТ следует быть внимательными к значениям свойства объекта dfilt, т. к. не все они поддерживаются в Simulink.

# 14.6. Ѕ-модели цифровых сигналов

Процедура создания S-модели цифрового сигнала включает в себя два этапа:

- 1. Создание S-модели *дискретного* сигнала на основе *блока* в одной из следующих групп блоков:
  - Sources (см. гл. 4);
  - Signal Processing Sources (см. разд. 11.2.2).
- 2. Создание S-модели *цифрового* сигнала S-модели квантованной последовательности — одним из следующих способов:
  - выбор типа данных fixdt (см. табл. 4.6), если он предусмотрен для параметра **Output data type**;
  - выбор типа данных Fixed-point или User-defined, если они предусмотрены для параметра **Output data type**, например, в блоке Sine Wave (см. разд. 11.2.2.8);
  - подключение к выходу блока источника дискретного сигнала (типа double) блока Data Type Conversion *(см. разд. 7.2.2)* для преобразования в тип данных fixdt.

При выполнении математических преобразований с цифровыми сигналами необходимо помнить о согласовании типа данных с ФТ между соответствующими блоками.

# 14.7. S-модели систем цифровой фильтрации с ФТ

Создание S-моделей систем цифровой фильтрации с ФТ на основе созданных S-моделей ЦФ с ФТ рассмотрим на примерах.

#### Пример 14.13

Создадим S-модель системы цифровой фильтрации с ФТ, включающей (рис. 14.16):

- S-модели входного дискретного сигнала (одноканальной последовательности) блоки Sine Wave и Sine Wave1 (см. разд. 11.2.2.8) с одинаковыми параметрами: Amplitude: 0.9; Frequency (Hz): 1600; Sample time: 1/8000; Samples per frame: 1; Output data type: double;
- □ S-модель БИХ-фильтра ПФ виртуальная подсистема Filter1\_IIR (см. пример 14.3);
- □ S-модель БИХ-фильтра ПФ с ФТ виртуальная подсистема Filter1\_IIR\_Fix (см. пример 14.8);
- □ блок Data Type Conversion *(см. разд. 7.2.2)* для преобразования дискретного входного сигнала типа double в цифровой сигнал с ФТ типа fixdt(1,8,7);

- □ блоки Add (см. разд. 6.1.1.1) для вычисления разности между дискретным и цифровым сигналами;
- □ S-модели средств анализа блоки Scope и Scope1 (см. разд. 5.2.3).



Рис. 14.16. S-модель системы цифровой фильтрации с ФТ с S-моделями БИХ-фильтра ПФ и БИХфильтра ПФ с ФТ в виде виртуальных подсистем Filter1\_IIR и Filter1\_IIR\_Fix

#### Пример 14.14

Создадим S-модель системы цифровой фильтрации с ФТ (рис. 14.17), в которой в качестве S-модели БИХ-фильтра ПФ используется блок Digital Filter Design с именем Filter2\_IIR (см. пример 14.9), а в качестве S-модели БИХ-фильтра ПФ с ФТ — блок Digital Filter с именем Filter2\_IIR\_Fix (см. пример 14.11).

Остальные блоки и их параметры такие же, как в примере 14.13.

#### Пример 14.15

Создадим S-модель системы цифровой фильтрации с ФТ, включающей (рис. 14.18):

- □ S-модели входного дискретного сигнала (одноканальной последовательности) блоки Sine Wave и Sine Wave1 (см. разд. 11.2.2.8) с одинаковыми параметрами: Amplitude: 0.9; Frequency (Hz): 700; Sample time: 1/8000; Samples per frame: 1; Output data type: double.
- □ S-модель КИХ-фильтра ФНЧ блок Digital Filter Design с именем Filter3\_FIR (см. пример 14.10);
- □ S-модель КИХ-фильтра ФНЧ с ФТ блок Digital Filter с именем Filter3\_FIR\_Fix (см. пример 14.12).

Остальные блоки такие же, как в примере 14.13.



Рис. 14.17. S-модель системы цифровой фильтрации с ФТ с S-моделью БИХ-фильтра ПФ в виде блока Digital Filter Design с именем Filter2\_IIR и S-моделью БИХ-фильтра ПФ с ФТ в виде блока Digital Filter с именем Filter2\_IIR\_Fix



Рис. 14.18. S-модель системы цифровой фильтрации с ФТ с S-моделью КИХ-фильтра ФНЧ в виде блока Digital Filter Design с именем Filter3\_FIR и S-моделью КИХ-фильтра ФНЧ с ФТ в виде блока Digital Filter с именем Filter3\_FIR\_Fix



# Глава 15

# Средства вычисления ДПФ

В этой главе рассматриваются S-модели средств вычисления дискретного преобразования Фурье (ДПФ), в том числе с фиксированной точкой (ФТ).

## 15.1. Математическая модель ДПФ

Математической моделью дискретного преобразования Фурье (ДПФ) является пара взаимно однозначных преобразований:

□ прямое ДПФ (Discrete Fourier Transform — DFT):

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} , \quad k = 0, 1, ..., N-1;$$
(15.1)

 $\Box$  обратное ДПФ (ОДПФ) (Inverse Discrete Fourier Transform — IDFT):

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}, \quad n = 0, 1, \dots, N-1, \quad (15.2)$$

где:

n — дискретное нормированное время n = nT/T;

k — дискретная нормированная частота  $k = k\Delta\omega/\Delta\omega$ , а  $\Delta\omega$  — период дискретизации по частоте, который при частоте дискретизации  $\omega_{\pi} = 2\pi/T$  равен:

$$\Delta \omega = \omega_{\rm g} / N = 2\pi / NT ;$$

x(n), n = 0, 1, ..., N - 1 — N-точечная последовательность — периодическая последовательность с периодом N в области дискретного нормированного времени;

X(k), k = 0, 1, ..., N - 1 — N-точечное ДПФ (*или коротко* — ДПФ) — периодическая последовательность с периодом N в области дискретной нормированной частоты;

*N* — длина последовательности и длина (размерность) 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.

ДПФ (15.1) 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  $N \square \Phi X(k)$  (15.1) представляет собой ее *спектр* с точностью до множителя 1/N.

Модуль ДПФ |X(k)| (с точностью до множителя 1/N) называют *амплитудным*, а аргумент  $\arg\{X(k)\} - \phi$ *азовым* спектром периодической последовательности.

Для вещественной периодической последовательности амплитудный спектр будет равен модулю ДПФ |X(k)| с точностью до множителя:

$$\begin{cases} 1/N, \ k = 0; \\ 2/N, \ k \neq 0, \end{cases}$$
(15.3)

а фазовый спектр — аргументу  $\arg\{X(k)\}$  с точностью до множителя 1/N.

Для комплексных последовательностей множитель в (15.3) всегда равен 1/N.

Для конечной последовательности x(n) длины  $N \ Д\Pi \Phi \ X(k)$  (15.1) представляет собой N дискретных равноотстоящих значений ее спектральной плотности  $X(e^{j\omega T})$  на периоде  $\omega_{\pi}$ :

$$X(k) = X(e^{j\omega T}) \bigg|_{\omega = k \frac{2\pi}{NT}}, \quad k = 0, 1, ..., N-1.$$

В том случае, если начальный и конечные отсчеты последовательности существенно различаются, их периодическое продолжение с периодом N будет воспроизводить периодическую последовательность с разрывами (скачками) на границах периода N. Как следствие, возникает эффект *растекания спектра* — искажение спектрального состава последовательности.

Для периодической последовательности подобный эффект будет наблюдаться, если хотя бы для одной из входящих в ее состав дискретных гармоник на длине Nбудет укладываться *нецелое* число периодов, вследствие чего периодическое продолжение соответствующих гармоник с периодом N приведет к разрывам на границах периода.

Для уменьшения растекания спектра применяют *весовые функции* (окна) — вещественные неотрицательные последовательности, максимальные в центре и монотонно спадающие к границам периода (длины) последовательности, что позволяет ослабить влияние разрывов на границах [21].

Для улучшения различения дискретных гармоник при растекании спектра целесообразно выводить *график модуля спектральной плотности*  $X(e^{j\omega T})$ , построенный на основе ДПФ (15.1), а при близко расположенных частотах — на основе ДПФ длины L (L > N) путем добавления (N - L) нулей к исходной последовательности.

Часть II. Моделирование цифровой обработки сигналов средствами Simulink

Для повышения быстродействия ДПФ (15.1)—(15.2) рассчитывается с помощью быстрого преобразования Фурье (БПФ) (Fast Fourier Transform — FFT).

В Simulink для вычисления ДПФ используется *алгоритм БПФ Кули—Тьюки*, в котором длина *N* исходной последовательности должна быть равной

$$N = 2^{\nu}, \qquad (15.4)$$

где v — целое положительное число. Если это условие не выполняется, то исходная последовательность дополняется необходимым количеством нулей или усекается.

В БПФ Кули—Тьюки с прореживанием по времени *N*-точечное ДПФ (15.1) рассчитывается за  $\nu$  этапов, на каждом из которых текущее ДПФ определяется через ДПФ вдвое меньшей размерности по формуле [23]:

$$\begin{cases} X_{m}^{i}(k) = X_{2m}^{i-1}(k) + W_{L}^{k} X_{2m+1}^{i-1}(k); \\ X_{m}^{i}\left(k + \frac{L}{2}\right) = X_{2m}^{i-1}(k) - W_{L}^{k} X_{2m+1}^{i-1}(k); \\ i = 1, 2, ..., y; \\ m = 0, 1, ..., M - 1; \\ k = 0, 1, ..., \frac{L}{2} - 1, \end{cases}$$
(15.5)

где: *i* — номер этапа алгоритма БПФ; *m* — номер ДПФ;  $M = 2^{v-i}$  — количество *L*-точечных ДПФ;  $L = 2^{i}$  — размерность ДПФ; *k* — номер отсчета ДПФ;  $X_{m}^{i}(k)$  — *L*-точечное ДПФ;  $X_{2m}^{i-1}(k)$ ,  $X_{2m+1}^{i-1}(k)$  — L/2-точечные ДПФ соответственно четных и нечетных отсчетов.

"Бабочкой" (butterfly) называют операцию одновременного вычисления по верхней и нижней формулам (15.5) при фиксированном значении *k*.

## 15.2. Ѕ-модели средств вычисления ДПФ

*S-модели средств вычисления*  $Д\Pi \Phi$  (15.1)—(15.2) представлены двумя блоками в библиотеке блоков **Signal Processing Blockset** в группе блоков **Transforms** (Преобразования) (табл. 15.1).

N⁰	Блок	Назначение
1.	FFT	Вычисление ДПФ (15.1) с помощью БПФ
2.	IFFT	Вычисление ОДПФ (15.2) с помощью БПФ

Таблица 15.1. S-модели средств вычисления ДПФ

# 15.2.1. Блок FFT

Блок *FFT* выполняет вычисление ДПФ (15.1) с помощью БПФ. Параметры блока задаются на двух вкладках — **Main** и **Fixed-point**.

На вкладке Main определяются параметры (рис. 15.1):

- **\Box** Twiddle factor compilation (Вычисление поворачивающего множителя) методы вычисления поворачивающего множителя  $W_N^{nk}$  в (15.1):
  - Trigonometric fcn в процессе моделирования;
  - Table lookup до начала моделирования с сохранением значений во внутренней памяти.

При выборе этого метода активизируется параметр **Optimize table for** (Оптимизация таблицы для) с двумя возможными значениями:

- Speed оптимизация по скорости, когда количество одновременно вычисляемых значений поворачивающего множителя зависит от типа данных, а именно: для данных с ПТ (типа double или single) k = 3N/4 значений, где k целое число; для данных с ФТ N значений;
- Метогу оптимизация памяти только для данных с ПТ, когда одновременно вычисляются k = N/4 значений поворачивающего множителя, где k целое число.

В обоих случаях, если k — нецелое число, исходную последовательность можно дополнить нулями до длины, равной целой степени двойки, используя блок Pad (*см разд. 11.4.4*), или указать длину ДПФ в поле **FFT** length при сбросе флага Inherit FFT length from input dimensions, о котором идет речь далее;

- □ Output in bit-reversed order (Выход в бит-реверсивном порядке) флаг, при установке которого отсчеты ДПФ на выходе выдаются в бит-реверсивном порядке двоичных номеров исходной последовательности;
- □ Inherit FFT length from input dimensions (Наследовать длину БПФ от размерности входного сигнала) флаг, при установке которого длина ДПФ совпадает с длиной исходной последовательности, а при сбросе устанавливается пользователем в поле FFT length.

На вкладке **Fixed-point** для данных с ФТ типа fixdt (см. табл. 4.6) задаются параметры (рис. 15.1):

- □ Rounding mode (Режим округления) см. параметр Integer rounding mode блока Add (см. разд. 6.1.1.1);
- □ Overflow mode (Режим переполнения) *см. разд. 6.2.1.2*;
- □ Skip divide-by-two on butterfly outputs for fixed-point signals (Пропуск деления на два на выходах бабочек для сигналов с фиксированной точкой) флаг, при установке которого для последовательностей с ФТ блокируется масштабирование (деление на два) на выходах бабочек;

- □ Sine table (Таблица синусов) формат для синусов (косинусов) поворачивающего множителя:
  - Same word length as input длина слова как у входного сигнала;
  - Specify word length соответствует формату fixdt(S,N) (см. табл. 4.6);
- □ **Product output** (Произведение на выходе) формат для произведения при вычислении бабочки:
  - Inherit via internal rule см. табл. 4.5;

388

- Same as input как у входного сигнала;
- Binary point scaling соответствует формату fixdt(S,N,M) (см. табл. 4.6);
- Slope and bias scaling соответствует формату fixdt(S,N,2^(-M),Bias) (см. табл. 4.6);
- □ Accumulator (Аккумулятор) формат для суммы при вычислении бабочки со значениями, подобными параметру **Product output**;
- □ **Output** (Выход) формат для отсчетов ДПФ на выходе со значениями, подобными параметру **Product output**;
- $\Box$  Lock scaling against changes by the autoscaling tool *c.m. paso. 6.2.1.2*.

	Function Block Parameters: FFT
FFT Outputs the complex fast Fourier transform (FFT) of a real or comp computing radix-2 decimation-in-time (DIT) or decimation-in-frequer depending on block options. Uses half-length and double-signal algo inputs where possible. Computes the FFT along the vector dimension for sample-based ve Computes the FFT along each column for all other inputs. When the length from input dimensions" check box is selected, the input must of-2 width.	FFT Outputs the complex fast Fourier transform (FFT) of a real or complex input by computing radix-2 decimation-in-time (DIT) or decimation-in-frequency (DIF), depending on block options. Uses half-length and double-signal algorithms for real inputs where possible. Computes the FFT along the vector dimension for sample-based vector inputs. Computes the FFT along each column for all other inputs. When the "Inherit FFT length from input dimensions" check box is selected, the input must have a power- of-2 width.
Main       Fixed-point         Parameters       Twiddle factor computation: Table lookup         Optimize table for:       Speed         Output in bit-reversed order       Inherit FFT length from input dimensions         FFT length:       128	Main       Fixed-point         Settings on this pane only apply when block inputs are fixed-point signals.         Fixed-point operational parameters         Rounding mode:       Floor         V       Overflow mode:         Wrap       V         V       Skip divide-by-two on butterfly outputs for fixed-point signals         Fixed-point data types       Mode         Sine table       Same word length as input         Product output Inherit via internal rule       V         Output       Inherit via internal rule         Lock scaling against changes by the autoscaling tool       Image: State S

Рис. 15.1. Параметры блока FFT

# 15.2.2. Блок IFFT

Блок *IFFT* выполняет вычисление ОДПФ (15.2) с помощью БПФ. Параметры блока подобны параметрам блока FFT *(см. разд. 15.2.1)*, за исключением двух следующих на вкладке **Main**:

- □ Input is conjugate symmetric (Вход является симметрично сопряженным) флаг, при установке которого восстанавливается исходная *вещественная* последовательность, а при сбросе она формально будет содержать весьма малую *ком*-*плексную* часть;
- □ Skip scaling (Пропуск масштабирования) флаг, при установке которого ОДПФ (15.2) вычисляется без множителя 1/N.

# 15.2.3. Особенности вычисления ДПФ в Simulink

При вычислении ДПФ в Simulink рекомендуется учитывать следующее:

- □ в качестве *источников* последовательности использовать блоки, содержащие параметр Samples per frame;
- □ в качестве средств анализа последовательностей и результатов вычисления ДПФ использовать блок Vector Scope (см. разд. 11.2.1.6);
- □ для блока Vector Scope при выводе *результата вычисления ДПФ* следует установить параметры:
  - на вкладке Scope Properties Input domain: Frequency;
  - на вкладке Axis Properties Frequency units: Hertz (Герцы); Frequency range: [0... Fs] или Frequency range: [0... Fs/2] (основная полоса частот); флаг Inherit sample time from input сбросить; параметр Sample time of original series сделать равным периоду дискретизации последовательности; в противном случае значение частоты Fs будет выбрано автоматически равным количеству отсчетов ДПФ, умноженному на период дискретизации;
  - на вкладке Line Properties не рекомендуется выбирать маркер stem.

Для улучшения различения дискретных гармоник при растекании спектра целесообразно выводить *график модуля спектральной плотности*, построенный на основе ДПФ. В этом случае дискретным гармоникам будут соответствовать *пики* с максимальными амплитудами;

- время моделирования, отсчитываемое от нуля до значения Simulation time stop, которое устанавливается в окне S-моделей, рекомендуется выбирать таким, чтобы на нем укладывалось *целое* количество фреймов; в противном случае в окне Command Window будет выдано предупреждение, а время моделирования автоматически сокращено;
- □ целое количество фреймов гарантируется при установке значения Simulation time stop согласно (11.10);

- □ для вычисления ДПФ одноканальной или многоканальной последовательности (количество фреймов K<sub>fr</sub> = 1) следует параметру Samples per frame блока-источника присвоить значение, равное длине последовательности L<sub>fr</sub>. (один фрейм длины L<sub>fr</sub>), поэтому, согласно (11.10), значение Simulation time stop устанавливается равным нулю;
- □ при вычислении ДПФ последовательности фреймов ДПФ вычисляется для каждого фрейма, поэтому параметру **Samples per frame** блока-источника присваивается значение, равное *длине фрейма* L<sub>fr</sub>, а значение **Simulation time stop**

(11.10) устанавливается в зависимости от количества фреймов  $K_{fr}$  и периода дискретизации *T*;

- □ для одновременного вывода ДПФ всех фреймов необходимо в окне параметров блока Vector Scope, анализирующего ДПФ, установить флаг Persistence на вкладке Display Properties;
- □ при *неизвестном* заранее количестве фреймов значение Simulation time stop можно установить равным inf (машинная бесконечность), и после запуска моделирования остановить процесс моделирования или сделать в нем паузу.

Более подробную информацию можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Transforms.

# 15.3. Примеры S-моделей систем обработки сигналов с вычислением ДПФ

Рассмотрим примеры создания S-моделей систем обработки сигналов с вычислением ДПФ.

#### Пример 15.1

На рис. 15.2 представлена S-модель системы обработки одноканальной последовательности с вычислением модуля ДПФ, включающая:

S-модель одноканальной последовательности — блок Sine Wave (см. разд. 11.2.2.8) с параметрами: Amplitude: 1; Frequency (Hz): 40; Sample time: 1/1000; Sample per frame: 180 (длина последовательности); Output data type: double.

В этом случае в окне S-моделей следует установить окончание времени моделирования Simulation time stop *равным нулю*;

□ S-модель средства вычисления ДПФ — блок FFT с параметром **FFT length:** 256 на вкладке **Main** и параметрами на вкладке **Fixed-point**, установленными по умолчанию (для последовательности типа double);

- □ блок Complex to Magnitude-Angle (см. разд. 6.1.3.1) для вычисления модуля ДПФ;
- □ S-модель средства анализа входной последовательности блок Vector Scope (см. разд. 11.2.1.6) с параметрами на вкладке Scope Properties Input domain: Time и Time display span (number of frames): 1;
- □ S-модель средства анализа модуля ДПФ блок Vector Scope1 с параметрами: на вкладке Scope Properties — Input domain: Frequency; на вкладке Axis Properties — Frequency units: Hertz (Герцы), Frequency range: [0... Fs], флаг Inherit sample time from input сброшен и указано Sample time of original series: 1/1000 (период дискретизации исходной последовательности при частоте дискретизации Fs = 1000 Гц).



Рис. 15.2. S-модель системы обработки одноканальной последовательности с вычислением модуля ДПФ

На рис. 15.3 представлена S-модель системы обработки одноканальной последовательности с вычислением модуля ДПФ (см. пример 15.1 и рис. 15.2), где *для уменьшения растекания спектра* при вычислении ДПФ применено окно Хэмминга — добавлен блок Window Function *(см. разд. 11.4.8)*.

График исходной одноканальной последовательности такой же, как на рис. 15.2.

Для того чтобы убедиться в уменьшении растекания спектра, сравните модули ДПФ на рис. 15.3 и 15.2.



Рис. 15.3. S-модель системы обработки одноканальной последовательности с вычислением модуля ДПФ и применением окна Хэмминга

392

На рис. 15.4 представлена S-модель системы обработки одноканальной последовательности (см. пример 15.1 и рис. 15.2), в которой перед блоком FFT стоит блок Pad (см. разд. 11.4.4) для дополнения нулями длины последовательности до ближайшей степени двойки (256).

График исходной одноканальной последовательности такой же, как на рис. 15.2. Модуль ДПФ совпадает с представленным на рис. 15.2.



Рис. 15.4. S-модель системы обработки одноканальной последовательности с вычислением модуля ДПФ и включенным блоком Pad

На рис. 15.5 представлена S-модель системы обработки одноканальной последовательности (см. пример 15.1 и рис. 15.2) с ее восстановлением по отсчетам ДПФ с помощью блока IFFT *(см. разд. 15.2.2)*, в окне параметров которого установлен флаг Input is conjugate symmetric.

График исходной одноканальной последовательности такой же, как на рис. 15.2.

График восстановленной последовательности дополнен нулями до длины 256, совпадающей с размерностью ДПФ.



Рис. 15.5. S-модель системы обработки одноканальной последовательности с ее восстановлением по отсчетам ДПФ

#### Пример 15.5

На рис. 15.6 представлена S-модель системы обработки одноканальной последовательности фреймов с вычислением модуля ДПФ для *каждого фрейма*, включающая:

□ S-модель одноканальной последовательности фреймов — блок Sine Wave (см. разд. 11.2.2.8) с параметрами: Amplitude: 1; Frequency (Hz): 40; Sample time:1/1000; Sample per frame: 20 (длина фрейма); Output data type: double.

В этом случае в окне S-моделей окончание времени моделирования Simulation time stop установлено для обработки шести фреймов, которое, согласно (11.10), будет равно:

Simulation time stop =  $(K_{fr} - 1)TL_{fr} = (6-1) \cdot (1/1000) \cdot 20 = 0.1$  (c);

- □ S-модель для дополнения длины фрейма (20) нулями до ближайшей степени двойки (32) — блок Pad (*см. разд. 11.4.4*);
- □ S-модель средства вычисления ДПФ блок FFT с параметром FFT length: 32;

- □ S-модели средств анализа последовательности фреймов (исходной и дополненной нулями) — блоки Vector Scope и Vector Scope2 (см. разд. 11.2.1.6) с параметрами на вкладке Scope Properties — Input domain: Time и Time display span (number of frames): 6;
- □ S-модель средства анализа ДПФ фреймов блок Vector Scope1, в окне параметров которого для одновременного вывода ДПФ всех фреймов установлен флаг Persistence на вкладке Display Properties.

Остальные блоки такие же, как на рис. 15.2.



Рис. 15.6. S-модель системы обработки одноканальной последовательности фреймов с вычислением модуля ДПФ для каждого фрейма

#### Пример 15.6

На рис. 15.7 представлена S-модель системы обработки одноканальной последовательности фреймов (см. пример 15.5 и рис. 15.6) с вычислением модуля ДПФ, где для уменьшения растекания спектра при вычислении ДПФ применено окно Хэмминга для каждого фрейма — добавлен блок Window Function (см. разд. 11.4.8).

График исходной одноканальной последовательности такой же, как на рис. 15.6.

Для того чтобы убедиться в уменьшении растекания спектра, сравните модули ДПФ на рис. 15.7 и 15.6.



Рис. 15.7. S-модель системы обработки одноканальной последовательности фреймов с вычислением модуля ДПФ и применением окна Хэмминга для каждого фрейма

На рис. 15.8 представлена S-модель системы обработки одноканальной последовательности с ФТ с вычислением модуля ДПФ с ФТ, включающая те же блоки, что и на рис. 15.2, и два блока Data Type Conversion *(см. разд. 7.2.2)*.



Рис. 15.8. S-модель системы обработки одноканальной последовательности с ФТ и вычислением модуля ДПФ с ФТ

Для блока Sine Wave установлены такие же параметры, как в примере 15.1, поэтому в окне S-моделей установлено окончание времени моделирования Simulation time stop *равным нулю*.

Блок Data Type Conversion выполняет преобразование типа данных double к типу данных fixdt(1,4,3), блок Data Type Conversion1 — обратное преобразование перед вычислением модуля ДПФ.

Для блока FFT установлено **FFT length:** 256 и на вкладке **Fixed-point** — параметры: **Rounding mode:** Floor; **Overflow mode:** Wrap; флаг **Skip divide-by-two on butterfly outputs for fixed-point signals** установлен; **Sine table:** Same word length as input; **Product output:** Inherit via internal rule; **Accumulator:** Inherit via internal rule.

Сравните последовательность и модуль ДПФ на рис. 15.2 с последовательностью с ФТ и модулем ДПФ с ФТ на рис. 15.8.

#### Пример 15.8

На рис. 15.9 представлена S-модель системы цифровой фильтрации двухканальной последовательности с вычислением модуля ДПФ, включающая:

□ S-модель двухканальной последовательности — блок Sine Wave (см. разд. 11.2.2.8) с параметрами: Amplitude: [2 3]; Frequency (Hz): [700 2500]; Sample time: 1/8000; Sample per frame: 1000 (длина последовательности); Output data type: double.

В этом случае в окне S-моделей следует установить окончание времени моделирования Simulation time stop *равным нулю*;

- □ S-модель КИХ-фильтра ФНЧ блок Lowpass Filter (см. пример 13.2);
- □ блоки Pad (см. разд. 11.4.4) для дополнения нулями длины последовательностей до ближайшей степени двойки (1024).
- □ S-модели средств анализа модуля ДПФ исходной и отфильтрованной последовательностей — блоки Vector Scope и Vector Scope1 (см. разд. 11.2.1.6) с параметрами: на вкладке Scope Properties — Input domain: Frequency; на вкладке Axis Properties — Frequency units: Hertz (Герцы), Frequency range: [0... Fs], флаг Inherit sample time from input сброшен и указано Sample time of original series: 1/8000 (период дискретизации исходной последовательности при частоте дискретизации Fs = 8000 Гц);
- □ блоки Complex to Magnitude-Angle (см. разд. 6.1.3.1) для вычисления модуля ДПФ.

В соответствии с АЧХ КИХ-фильтра ФНЧ (см. рис. 13.7) дискретная синусоида первого канала с частотой 700 Гц попадает в ПП, а второго канала с частотой 2500 Гц — в ПЗ.


Рис. 15.9. S-модель системы цифровой фильтрации двухканальной последовательности с вычислением модуля ДПФ



# Глава 16

# Многоскоростные системы ЦОС

*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:

$$L = \frac{f'_{\mathcal{A}}}{f_{\mathcal{A}}} \implies f'_{\mathcal{A}} = Lf_{\mathcal{A}}, \qquad (16.1)$$

где  $f_{\rm d}$  и  $f'_{\rm d}$  — 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:

$$M = \frac{f_{\pi}}{f'_{\pi}} \implies f'_{\pi} = \frac{f_{\pi}}{M}, \qquad (16.2)$$

где  $f_{\rm d}$  и  $f'_{\rm d}$  — частоты дискретизации сигналов на *входе* и *выходе* системы децимации соответственно.

Системы интерполяции (децимации) называют *однократными*, если увеличение (уменьшение) частоты дискретизации выполняется за один прием — однократно.

Повышение или понижение частоты дискретизации на рациональный коэффициент L/M, называемое *передискретизацией* (resampling), реализуется последовательным соединением систем интерполяции с коэффициентом L и децимации с коэффициентом M.

*Многократными* называют системы, образованные последовательным соединением однократных, что оправдано при больших значениях *L* и *M*, т. к. требования к однократным системам будут существенно менее жесткими.

Глава 16. Многоскоростные системы ЦОС

В ЦОС преобразование частоты дискретизации выполняется средствами *цифровой фильтрации*, в результате которой формируется выходной сигнал с новой частотой дискретизации.

# 16.1. Математическая модель системы однократной интерполяции

*Математическая модель* системы однократной интерполяции, представленная на рис. 16.1, включает в себя два блока:

□ экспандер;

цифровой ФНЧ.

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):

$$w(nT') = \begin{cases} x\left(\frac{nT}{L}\right), \ n = 0, L, 2L, \dots; \\ 0 при других n \end{cases}$$
(16.3)

$$x(nT)$$

$$\downarrow L$$

$$w(nT')$$

$$H(z)$$

$$y(nT')$$

$$(mT')$$

$$(mT')$$

$$(mT')$$

$$(mT')$$

$$H(z)$$

$$(mT')$$

Рис. 16.1. Система однократной интерполяции

Соотношение между периодами дискретизации T' = T/L входного x(nT) и промежуточного w(nT') сигналов позволяет интерпретировать "новую" шкалу nT' как *растяжение* "старой" шкалы nT в L раз (рис. 16.2,  $a, \delta$ ).

В частотной области, где речь идет о спектральных плотностях данных сигналов  $X(e^{j\omega T})$  и  $W(e^{j\omega' T})$ , соотношение между частотами дискретизации  $\omega'_{d} = L\omega_{d}$  позволяет интерпретировать "новую" шкалу  $\omega'$ , напротив, как *сжатие* "старой" шкалы  $\omega$  в *L* раз (рис. 16.3, *a*, *б*).

Цифровой ФНЧ подавляет (L-1) "лишние" составляющие спектральной плотности промежуточного сигнала на интервале  $\omega'_{d}$ , его идеальная АЧХ в основной полосе частот шкалы  $\omega'$  должна удовлетворять требованиям (рис. 16.3, *в*):

$$A(\omega') = \begin{cases} L & \text{в полосе пропускания } 0 \le \omega' \le \omega'_{\pi}/2L; \\ 0 \text{при других значениях } \omega' \end{cases}$$
(16.4)



Рис. 16.2. Временная интерпретация процедуры интерполяции: входной (а), промежуточный (б) и выходной сигналы в "новой" (в) и "старой" (г) шкале времени

или в шкале частот ω:

$$A(\omega) = \begin{cases} L & \text{в полосе пропускания } 0 \le \omega \le \omega_{\text{д}}/2; \\ 0 \text{ при других значениях } . \omega \end{cases}$$
(16.5)

На рис. 16.2, в и рис. 16.3, г изображены выходной сигнал y(nT') и модуль его спектральной плотности  $Y(e^{j\omega'T})$ , а на рис. 16. 2, г и рис. 16.3,  $\partial$  — они же в "старых" шкалах времени nT и частот  $\omega$ . Период дискретизации выходного сигнала *уменьшился*, а частота дискретизации *увеличилась* в L раз.

*Модули* спектральной плотности дискретного сигнала на выходе ФНЧ (см. рис. 16.3, *г*) и спектральной плотности дискретного сигнала, которая была бы получена путем непосредственной дискретизации исходного *аналогового* сигнала с частотой  $\omega'_{\rm d} = L\omega_{\rm d}$ , совпадают с точностью до множителя *L* (который учтен в (16.5)), а *аргументы* различаются [23].



Рис. 16.3. Частотная интерпретация процедуры интерполяции: модули спектральных плотностей входного сигнала (*a*); промежуточного сигнала в "новой" шкале частот (*б*); идеальная АЧХ (*в*); выходной сигнал в "новой" (*г*) и "старой" (*д*) шкалах частот

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 (16.3) соотношение вход/выход системы интерполяции в *z*-области имеет вид:

$$Y(z) = W(z)H(z) = X(z^{L})H(z),$$
(16.6)

где:

$$W(z) = \sum_{n=0, L, 2L, \dots}^{\infty} x\left(\frac{nT}{L}\right) z^{-n} = \sum_{m=0}^{\infty} x(mT) z^{-Lm} = \sum_{m=0}^{\infty} x(m) z^{-Lm} = X(z^{L})$$

при подстановке  $m = n/L \rightarrow n = Lm$ .

# 16.2. Математическая модель системы однократной децимации

*Математическая модель* системы однократной децимации, представленная на рис. 16.4, включает в себя два блока:

цифровой ФНЧ;

□ компрессор.

*Цифровой ФНЧ* с передаточной функцией H(z) предназначен для ограничения спектральной плотности верхней частотой  $\omega'_{\rm n}/2 = \omega_{\rm n}/2M$ .



Рис. 16.4. Система однократной децимации

В частотной области, где речь идет о спектральной плотности  $X(e^{j\omega T})$  входного сигнала x(nT), соотношение между частотами дискретизации  $\omega'_{d} = \omega_{d}/M$  позволяет интерпретировать "новую" шкалу  $\omega'$  как *растяжение* "старой" шкалы  $\omega$  в M раз (рис. 16.5,  $a, \delta$ ).

В общем случае спектральная плотность  $X(e^{j\omega T})$  занимает всю основную полосу [0;  $\omega_{\rm g}/2$ ] (рис. 16.5, *a*). При этом для "новой" частоты дискретизации  $\omega'_{\rm g} = \omega_{\rm g}/M$  не будет выполняться условие теоремы Котельникова  $\omega'_{\rm g} \ge 2\omega_{\rm g}$ , что приведет к элайсингу<sup>1</sup> (см. пунктирные линии на рис. 16.5, *б*).

Идеальная АЧХ ФНЧ в основной полосе частот шкалы ω' должна удовлетворять требованиям (рис. 16.5, *в*):

$$A(\omega') = \begin{cases} 1 \text{ в полосе пропускания } 0 \le \omega' \le \omega_{\pi}^{2} / \\ 0 \text{ при других значениях } \omega' \end{cases}$$

или в шкале частот ω:

$$A(\omega) = \begin{cases} 1 \text{в полосе пропускания } 0 \le \omega \le \omega_{\pi}^2 / M \\ 0 \text{при других значениях } . \omega \end{cases}$$
(16.7)

Модуль спектральной плотности  $W(e^{j\omega'T})$  промежуточного сигнала w(nT) на выходе ФНЧ показан на рис. 16.5, *г*.

 $<sup>^1</sup>$ Элайсингом (aliasing) называют наложение спектральных плотностей в области  $\,\omega_{\pi}'/2$  .



Рис. 16.5. Частотная интерпретация процедуры децимации: модуль спектральной плотности входного сигнала в "старой" (*a*) и "новой" (*б*) шкале частот; идеальная АЧХ (*в*); модули спектральных плотностей промежуточного сигнала (*a*) и выходного сигнала в "новой" (*д*) и "старой" (*е*) шкалах частот



Рис. 16.6. Временная интерпретация процедуры децимации: промежуточный сигнал в "старой" шкале времени (*a*) и выходной сигнал в "новой" шкале времени (*б*)

Компрессор (блок со стрелкой вниз на рис. 16.4) формирует выходной сигнал y(nT') путем прореживания промежуточного сигнала w(nT), из которого берется

каждый *М*-й отсчет, в результате чего "новый" период дискретизации оказывается равным T' = MT (рис. 16.6, *a*,  $\delta$ ):

$$y(nT') = w(MnT), n = 0, 1, 2, ...$$

Модуль спектральной плотности выходного сигнала в "новой" ω' и "старой" ω шкалах частот показан на рис. 16.5, *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  $\omega'_{n} = \omega_{n}/M$ , совпадают, а *аргументы* различаются [23].

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.

Вывод соотношения вход/выход в *z*-области для системы однократной децимации не очевиден, т. к. операция выборки *M*-х отсчетов из промежуточного сигнала не имеет простого решения; с ним можно познакомиться в [23].

# 16.3. Математическая модель системы однократной передискретизации

*Математическая модель* системы однократной передискретизации, представлена на рис. 16.7.

Повышение или понижение частоты дискретизации на рациональный коэффициент L/M реализуется последовательным соединением систем интерполяции с коэффициентом L и децимации с коэффициентом M (рис. 16.7, a). В результате объединения двух каскадно включенных ФНЧ, работающих на одинаковой частоте дискретизации  $L\omega_{\rm d}$ , переходят к системе однократной передискретизации с единственным ФНЧ (рис. 16.7,  $\delta$ ). Его идеальная АЧХ, согласно (16.5) и (16.7), должна удовлетворять требованиям:



Рис. 16.7. Система однократной передискретизации с двумя ФНЧ (а) и одним ФНЧ (б)

# 16.4. Полифазные структуры многоскоростных систем и их описание в MATLAB

Полифазные структуры систем однократной интерполяции и децимации основаны на замещении одного КИХ-фильтра ФНЧ, работающего на "высокой" частоте дискретизации, эквивалентной системой из КИХ-фильтров ФНЧ, работающих на "низкой" частоте.

Идея построения полифазных структур основана на использовании двух замечательных тождеств (рис. 16.8) [20].

Рассмотрим систему однократной интерполяции (см. рис. 16.1).

Замечательному тождеству на рис. 16.8, а в *z*-области соответствует тождество для соотношения вход/выход (см. (16.6)):

$$Y(z) = X(z^{L})H(z) \equiv Y(z) = H(z^{L})X(z).$$
(16.9)

Оно означает, что можно сначала выполнить фильтрацию входного сигнала с "низкой" частотой дискретизации КИХ-фильтром с передаточной функцией  $H(z^L)$ с "низкой" частотой дискретизации, в L раз меньшей, чем у КИХ-фильтра с передаточной функцией H(z). Тем самым исключается обработка (L-1) промежуточных нулей, которые затем, при переходе к "высокой" частоте дискретизации, добавляются с помощью экспандера.



Рис. 16.8. Тождества для систем однократной интерполяции (а) и децимации (б)

В *полифазной* структуре один КИХ-фильтр длины N заменяется L более простыми КИХ-фильтрами длины G = N/L (целое число) с передаточными функциями  $H_k(z^L)$ , k = 0, 1, ..., L - 1, работающими на "низкой" частоте дискретизации.

Для формирования полифазной структуры запишем передаточную функцию КИХ-фильтра длины N :

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} = h_0 z^{-1} + h_2 z^{-2} + \dots + h_{L-1} z^{-L-1} + \dots + h_{N-1} z^{-(N-1)}.$$

Разобьем сумму из N слагаемых на G сумм из L слагаемых ( $G \times L = N$ ) и используем матричное представление H(z) в виде произведения матрицы A размера  $G \times L$  на вектор-столбец B' длины L:

$$H(z) = \begin{vmatrix} h_0 & h_1 & \dots & h_{L-1} \\ h_L z^{-L} & h_{L+1} z^{-L} & \dots & h_{L+(L-1)} z^{-L} \\ \vdots & \vdots & \vdots & \vdots \\ h_{(G-1)L} z^{-(G-1)L} & h_{(G-1)L+1} z^{-(G-1)L} & \dots & h_{(G-1)L+(L-1)} z^{-(G-1)L} \end{vmatrix} \times \begin{vmatrix} z^0 \\ z^{-1} \\ \vdots \\ z^{-(L-1)} \end{vmatrix} = AB'$$

Здесь  $h_{(G-1)L+(L-1)} = h_{N-1}$  при  $G \times L = N$ . Используя тождество из линейной алгебры:

$$AB' = BA'$$
,

перепишем H(z) в виде:

$$H(z) = \begin{vmatrix} z^{0} & z^{-1} & \dots & z^{-(L-1)} \end{vmatrix} \times \begin{vmatrix} h_{0} & h_{L}z^{-L} & \dots & h_{(G-1)L}z^{-(G-1)L} \\ h_{1} & h_{L+1}z^{-L} & \dots & h_{(G-1)L+1}z^{-(G-1)L} \\ \vdots & \vdots & \vdots & \vdots \\ h_{L-1} & h_{L+(L-1)}z^{-L} & \dots & h_{(G-1)L+(L-1)}z^{-(G-1)L} \end{vmatrix} = BA'$$

Здесь строки матрицы длины G соответствуют передаточным функциям  $H_k(z^L)$ , k = 0, 1, ..., L-1, КИХ-фильтров длины G, что позволяет перейти к записи H(z) в виде произведения векторов:

$$H(z) = \begin{vmatrix} z^{0} & z^{-1} & \dots & z^{-(L-1)} \end{vmatrix} \times \begin{vmatrix} H_{0}(z^{L}) \\ H_{1}(z^{L}) \\ \vdots \\ H_{L-1}(z^{L}) \end{vmatrix} = \sum_{k=0}^{L-1} H_{k}(z^{L}) z^{-k} , \qquad (16.10)$$

где  $H_k(z^L) - k$ -я строка матрицы A':

$$H_k(z^L) = \sum_{n=0}^{G-1} h_{Ln+k} z^{-Ln} , \ k = 0, 1, L-1.$$
 (16.11)

Представление передаточной функции H(z) в виде суммы (16.10) соответствует параллельной структуре КИХ-фильтров длины G с передаточными функциями  $H_k(z^L)$ , k = 0, 1, ..., L-1, и базовым элементом задержки  $z^{-1}$  между ними (рис. 16.9). Такое представление H(z) называют полифазным, соответствующую структуру — полифазной, ее ветви — фазами, а КИХ-фильтры с передаточными функциями  $H_k(z^L)$ , k = 0, 1, ..., L-1, — полифазными фильтрами.

Переход к "высокой" частоте дискретизации реализуется добавлением экспандеров *после* полифазных фильтров (см. рис. 16.9).

Z-изображение *реакции* равно (см. (16.6) и (16.10)):

$$Y(z) = X(z^{L})H(z) = X(z^{L})\sum_{k=0}^{L-1}H_{k}(z^{L})z^{-k} = \sum_{k=0}^{L-1}X(z^{L})H_{k}(z^{L})z^{-k} = \sum_{k=0}^{L-1}Y_{k}(z^{L})z^{-k}$$

и имеет полифазное представление, подобное (16.10).



Рис. 16.9. Полифазная структура системы однократной интерполяции

Полифазную структуру системы однократной децимации можно получить из следующих соображений. Сравнивая правые системы на рис. 16.8, a,  $\delta$ , 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. 16.9, получаем полифазную структуру системы однократной децимации (рис. 16.10). В ней полифазное представление имеет *z*-изображение воздействия:

$$X(z) = \sum_{k=0}^{M-1} X_k(z^M) z^{-k} .$$

Переход к "низкой" частоте дискретизации выходного сигнала реализуется добавлением компрессоров *перед* полифазными фильтрами (см. рис. 16.10).

Передаточная функция полифазных фильтров  $H_k(z^M)$  имеет вид (16.11) при L = M.

Более подробно с принципами построения полифазных структур однократных систем интерполяции и децимации можно познакомиться в [20, 21].

Полифазная структура системы однократной передискретизации реализуется последовательным соединением полифазных структур однократной системы интерполяции и децимации и их объединением.

В MATLAB полифазная структура, соответственно системе интерполяции, децимации и передискретизации, описывается в виде объектов mfilt (maltirate filter object):

Hi = mfilt.firinterp(L,Num)
Hd = mfilt.firdecim(M,Num)
Hid = mfilt.firsrc(L,M,Num)

где L, м — коэффициенты интерполяции и децимации, а Num — коэффициенты передаточной функции КИХ-фильтра ФНЧ, работающего на "высокой" частоте дискретизации.



Рис. 16.10. Полифазная структура системы однократной децимации

В отсутствие параметра Num автоматически синтезируется *фильтр Найквиста* с линейной ФЧХ, идеальная АЧХ которого отвечает следующим условиям:

□ для системы однократной *интерполяции*: в полосе пропускания (ПП) равна L и достигает половинного значения (−6 дБ) в шкале нормированных частот

$$\hat{f} = \frac{f}{f_{\pi}/2}$$
 на частоте 1/L;

- □ для системы однократной *децимации*: в ПП равна единице и достигает половинного значения в шкале нормированных частот на частоте 1/м;
- □ для системы однократной *передискретизации*: в ПП равна L и достигает половинного значения в шкале нормированных частот на частоте 1/max{L,M}.

Выходными параметрами объекта mfilt являются его свойства, в числе которых свойство Numerator хранит коэффициенты ФНЧ, работающего на "высокой" частоте дискретизации.

## 16.5. Ѕ-модели многоскоростных систем

*S-модели многоскоростных систем* представлены блоками в библиотеке блоков Signal Processing Blockset в группе блоков Filtering  $\rightarrow$  Multirate Filters (Фильтрация  $\rightarrow$  Многоскоростные фильтры), из которых рассмотрим три основных (табл. 16.1).

N⁰	Блок	Назначение
1.	FIR Interpolation	S-модель системы однократной интерполяции
2.	FIR Decimation	S-модель системы однократной децимации
3.	FIR Rate Conversion	S-модель системы однократной передискретизации

Таблица 16.1. S-модели многоскоростных систем — блоки группы Multirate Filters

Список параметров каждого блока из табл. 16.1 зависит от выбора одного из двух указателей в группе **Coefficient source** (Источник коэффициентов) (рис. 16.11):

- □ Dialog Parameters (Параметры диалога) при непосредственном указании параметров многоскоростной системы в группе Parameters;
- □ Multirate filter object (MFILT) (Многоскоростной ЦФ в виде объекта mfilt) при экспорте объекта mfilt (см. разд. 16.3) из Workspace.

При выборе этого указателя параметры блока FIR Interpolation отображают свойства объекта mfilt, поэтому далее этот указатель не рассматривается.

Параметры всех блоков из табл. 16.1 задаются на двух вкладках — Main и Fixed-point. На вкладке Fixed-point параметры подобны параметрам блока Digital Filter для КИХ-фильтра ФНЧ (см. разд. 14.4), поэтому далее они не рассматриваются.

#### 16.5.1. Блок FIR Interpolation

Блок *FIR Interpolation* представляет собой S-модель системы однократной интерполяции. Параметры блока на вкладке **Main** включают в себя (см. рис. 16.11):

□ FIR filter coefficients (Коэффициенты КИХ-фильтра) — поле ввода для коэффициентов КИХ-фильтра ФНЧ, работающего на "высокой" частоте дискретизации.

Коэффициенты КИХ-фильтра могут вводиться одним из следующих способов:

- именем вектора коэффициентов из Workspace;
- именем функции MATLAB при синтезе средствами блока FIR Interpolation фильтра Найквиста на базе КИХ-фильтра с линейной ФЧХ с указанием ее входных параметров в круглых скобках.

Рекомендуется выбрать функцию<sup>1</sup> fir1 и в качестве входных параметров указать *порядок* КИХ-фильтра и *частоту*, на которой идеальная АЧХ дости-

гает половинного значения. В шкале нормированных частот  $\hat{f} = \frac{f}{f_{\pi}/2}$  эта

частота не должна превышать значения 1/L, где L — коэффициент интерполяции.

В ПП значение идеальной АЧХ фильтра Найквиста равно *единице*, поэтому сигнал на выходе блока FIR Interpolation следует *умножать на L* (см. (16.5)).

Для анализа характеристик КИХ-фильтра в окне параметров всех блоков из табл. 16.1 предусмотрена кнопка View Filter Response, при нажатии которой

<sup>&</sup>lt;sup>1</sup> Функция синтеза КИХ-фильтра *методом окон* с автоматическим, скрытым от пользователя, выбором окна.

автоматически открывается окно Filter Visualization Tool GUI FVTool (см. разд. 12.3).

После анализа АЧХ параметры функции fir1 можно скорректировать.

410

Обращаясь к команде Filter Information (Информация о фильтре), можно видеть, что по умолчанию создается *полифазная структура* системы однократной интерполяции, соответствующая объекту mfilt.firinterp (см. разд. 16.3);

- □ Interpolation factor (Коэффициент интерполяции) коэффициент интерполяции *L* (16.1);
- □ Framing (Обработка фреймов) варианты интерполяции для последовательности фреймов:
  - Maintain input frame size (Поддержка размера входного фрейма) с увеличением в раз *L* длины фрейма;
  - Maintain input frame rate (Поддержка скорости входного фрейма) с увеличением в раз L частоты дискретизации фрейма.

При интерполяции одноканальных или многоканальных последовательностей необходимо выбрать Maintain input frame size;

□ Output buffer initial conditions (Начальные условия для выходного буфера) — начальные значения сигнала на выходе, сохраняемые во внутреннем буфере; по умолчанию параметр равен нулю и такие значения отсутствуют.

🗑 Function Block Parameters: FIR Interpolation 🛛 🛛 🔯		
FIR Interpolation Upsample input signal by an integer value factor, then apply an FIR filter. You can define the filter using mask dialog parameters, or by a multirate FIR interpolation filter object (mfit.frinterp) from the Filter Design Toolbox. The filter is implemented using a polyphase interpolation structure. The filter coefficients are scaled by the interpolation factor. There will be latency when this block is run in multirate, multitasking mode. For this case, an initial condition can be specified for the output buffer. Coefficient source     ① Dialog parameters     ① Multirate filter object (MFILT)		
Framing: Maintain input frame size		
Output buffer initial conditions: 0 View Filter Response		
QK <u>Cance</u> <u>H</u> elp <u>Apply</u>		

Рис. 16.11. Параметры блока FIR Interpolation на вкладке Main

#### 16.5.2. Блок FIR Decimation

Блок *FIR Decimation* представляет собой S-модель системы однократной децимации. Параметры блока на вкладке **Main** подобны параметрам блока FIR Interpolation (*см. разд. 16.4.1*) за исключением следующих двух:

- □ Decimation factor (Коэффициент децимации) коэффициент децимации *M* (16.2);
- □ Filter structure (Структура фильтра) структура КИХ-фильтра ФНЧ (см. табл. 13.2).

При указании параметра **FIR filter coefficients** с помощью функции fir1 частота, на которой идеальная АЧХ достигает половинного значения, в шкале нормированных частот  $\hat{f} = \frac{f}{f_{\pi}/2}$  не должна превышать значения 1/*M*.

После нажатия кнопки View Filter Response, обращаясь к команде Filter Information, можно видеть, что по умолчанию создается *полифазная структура* системы однократной децимации, соответствующая объекту mfilt.firdecim (см. разд. 16.3).

### 16.5.3. Блок FIR Rate Conversion

Блок *FIR Rate Conversion* представляет собой S-модель системы однократной передискретизации. Параметры блока на вкладке **Main** включают в себя:

□ Interpolation factor — коэффициент интерполяции *L* (16.1);

□ Decimation factor — коэффициент децимации *M* (16.2);

□ FIR filter coefficients — см. разд. 16.4.1.

При указании этого параметра с помощью функции fir1 частота, на которой идеальная АЧХ достигает половинного значения, в шкале нормированных час-

тот  $\hat{f} = \frac{f}{f_{\pi}/2}$  не должна превышать значения  $1/\max\{L, M\}$ .

Отметим, что в ПП значение идеальной АЧХ фильтра Найквиста будет равно не L, как в (16.8), а *единице*, поэтому сигнал на выходе блока FIR Interpolation следует *умножать на* L.

После нажатия кнопки View Filter Response, обращаясь к команде Filter Information, можно видеть, что по умолчанию создается *полифазная структура* системы однократной передискретизации, соответствующая объекту mfilt.firsrc (см. разд. 16.3).

В Simulink операция передискретизации выполняется только для *последовательности фреймов* и производится последовательно по фреймам. Длина входного фрейма должна задаваться кратной коэффициенту децимации *M*, а длина выходного фрейма умножается на коэффициент *L/M*.

При обработке одноканальных и многоканальных последовательностей их можно представить в виде последовательностей фреймов длины *M*.

Более подробную справку можно получить с помощью справочной системы MATLAB по Simulink в формате HTML в разделе Signal Processing Blockset  $\rightarrow$  Blocks  $\rightarrow$  Filtering  $\rightarrow$  Multirate Filters.

# 16.6. Моделирование многоскоростной обработки сигналов

Моделирование многоскоростной обработки сигналов рассмотрим на примерах.

#### Пример 16.1

Создадим S-модель многоскоростной обработки сигналов с повышением частоты дискретизации в два раза (L = 2), включающую (рис. 16.12):

- □ S-модель входного дискретного сигнала одноканальную последовательность в виде суммы двух дискретных гармоник, формируемой с помощью блоков:
  - Sine Wave (см. разд. 11.2.2.8) с параметрами: Amplitude: 1; Frequency (Hz): 30; Sample time: 1/1000; Sample per frame: 1;
  - Sine Wave1 с параметрами: Amplitude: 1; Frequency (Hz): 60; Sample time: 1/1000; Sample per frame: 1;
  - блок Add (см. разд. 6.1.1.1);
- □ S-модель системы *однократной интерполяции* блок FIR Interpolation *(см. разд. 16.4.1)* с параметрами: FIR filter coefficients: fir1(30,1/4); Interpolation factor: 2; Framing: Maintain input frame size; Output buffer initial conditions: 0.

АЧХ КИХ-фильтра Найквиста представлена на рис. 16.13. Частота, на которой АЧХ достигает своего половинного значения, в шкале нормированных частот  $\hat{f}$ , равна 1/4 (<1/L).

Частоты гармоник входного сигнала 30 и 60 Гц при частоте дискретизации  $f_{\rm d} = 1000$  Гц (1/**Sample time**), в шкале нормированных частот  $\hat{f}$ , равные 0.06 и 0.12, расположены в ПП фильтра;

- □ блок Gain *(см. разд. 6.1.1.5)* с параметром Gain: 2 для согласования АЧХ синтезированного КИХ-фильтра Найквиста с АЧХ (16.5);
- □ S-модели средств анализа сигналов блоки Vector Scope (см. разд. 11.2.1.6).

В установившемся режиме на выходе блока FIR Interpolation имеем дискретный сигнал с частотой дискретизации, повышенной в два раза.



414

Рис. 16.12. S-модель многоскоростной обработки сигналов с повышением частоты дискретизации в два раза



Рис. 16.13. АЧХ КИХ-фильтра Найквиста

#### Пример 16.2

Создадим S-модель многоскоростной обработки сигналов с *понижением частоты дискретизации в два раза* (M = 2) (рис. 16.14). В S-модели, представленной на рис. 16.14, удален блок Gain, и блок FIR Interpolation заменен на FIR Decimation (*см. разд. 16.4.2*) — S-модель системы *однократной децимации* — с параметрами: **FIR filter coefficients:** fir1(30,1/4); **Decimation factor:** 2; **Filter structure:** Direct form; остальные параметры такие же, как для блока FIR Interpolation (см. пример 16.1).

АЧХ КИХ-фильтра Найквиста представлена на рис. 16.13. Частота, на которой АЧХ достигает своего половинного значения, в шкале нормированных частот  $\hat{f}$  равна 1/4 (<1/M).

Частоты гармоник входного сигнала 30 и 60 Гц, в шкале нормированных частот  $\hat{f}$  равные 0.06 и 0.12, расположены в ПП фильтра.

В установившемся режиме на выходе блока FIR Decimation имеем дискретный сигнал с частотой дискретизации, пониженной в два раза.



Рис. 16.14. S-модель многоскоростной обработки сигналов с понижением частоты дискретизации в два раза

#### Пример 16.3

Создадим S-модель многоскоростной обработки сигналов с изменением частоты дискретизации на рациональный коэффициент L/M = 3/2, включающую (рис. 16.15):

- □ S-модель входного дискретного сигнала последовательность фреймов в виде суммы двух дискретных гармоник, с такими же параметрами, как в примере 16.1, но установкой длины фреймов, равной *M* при значении параметра Sample per frame: 2;
- □ S-модель системы *однократной передискретизации* блок FIR Rate Conversion *(см. разд. 16.4.3)* с параметрами: FIR filter coefficients: fir1(30,1/4); Interpolation factor: 3; Decimation factor: 2;
- □ блок Gain *(см. разд. 6.1.1.5)* с параметром Gain: 3 для согласования АЧХ синтезированного КИХ-фильтра Найквиста с АЧХ (16.8);

□ S-модели средств анализа сигналов, такие же, как в примере 16.1.

АЧХ КИХ-фильтра Найквиста представлена на рис. 16.13. Частота, на которой АЧХ достигает своего половинного значения, в шкале нормированных частот  $\hat{f}$  равна 1/4 (< 1/max {L, M}).

Частоты гармоник входного сигнала 30 и 60 Гц, в шкале нормированных частот  $\hat{f}$  равные 0.06 и 0.12, расположены в ПП фильтра.

В установившемся режиме на входе блока FIR Rate Conversion имеем длину фреймов M = 2, а на выходе —  $M \cdot L/M = 3$ .



Рис. 16.15. S-модель многоскоростной обработки сигнала (последовательности фреймов) с изменением частоты дискретизации на рациональный коэффициент 3/2

# Список литературы

- 1. M. Nuruzzaman. Modelng and Simulating for Engineers and Scientists. AUTORHOUSE, 2004.
- S. T. Karris/ Signals and Systems mith MATLAB Computing and Simulink Modeling. Third Edition. — Orhard Publications, 2007.
- Айфичер Э., Джервис Б. Цифровая обработка сигналов. М. СПб. Киев: Вильямс, 2004.
- 4. Ануфриев И. Е., Смирнов А. Б., Смирнова Е. Н. МАТLAB 7. СПб.: БХВ-Петербург, 2005.
- Воднев В. Г., Наумович А. Ф., Наумович Н. Ф. Математический словарь высшей школы. — М.: МПИ, 1989.
- 6. Воеводин В. В., Кузнецов Ю. А. Матрицы и вычисления. М.: Наука, 1984.
- Гадзиковский В. И. Методы проектирования цифровых фильтров. М.: Горячая лилия — Телеком, 2007.
- Гольденберг Л. М., Матюшкин Б. Д., Поляк М. Н. Цифровая обработка сигналов. Справочник. — М.: Радио и связь, 1985.
- 9. Дьяконов В. П. МАТLАВ и SIMULINK для радиоинженеров. М.: ДМК Пресс, 2011.
- 10. Дьяконов В. П. SIMULINK 5/6/7. Самоучитель. М.: ДМК-Пресс, 2008.
- 11. Дьяконов В. П. МАТLAB 6.5 SP1/7.0 Simulink 5/6. Обработка сигналов и проектирование фильтров. М.: СОЛОН-Пресс, 2005.
- 12. Дьяконов В. П. МАТLAB 6.5 SP1/7.0 Simulink 5/6. Обработка сигналов и проектирование фильтров. М.: СОЛОН-Пресс, 2005.
- 13. Дьяконов В. П. Simulink 5/6/7. М.: ДМК-Пресс, 2008.
- 14. Дэбни Дж., Харман Т. Simulink 4. М.: БИНОМ. Лаборатория знаний, 2003.
- 15. Икрамов Х. Д. Численное решение матричных уравнений. М.: Наука, 1984.
- 16. Кетков Ю. Л., Кетков А. Ю., Шульц М. М. МАТLAB 6.х: программирование численных методов. СПб.: БХВ-Петербург, 2004.
- 17. Корн Г., Корн Т. Справочник по математике. М.: Наука, 1984.
- 18. Лайонс Р. Цифровая обработка сигналов. М.: Бином, 2006.
- 19. Левин Б. Р. Теоретические основы статистической радиотехники. М.: Советское радио, 1969.

- 20. Оппенгейм А., Шафер Р. Цифровая обработка сигналов. М.: Техносфера, 2006.
- 21. Сергиенко А. Б. Цифровая обработка сигналов. 3-е изд. СПб.: БХВ-Петербург, 2010.
- 22. Солонина А. И., Арбузов С. М. Цифровая обработка сигналов. Моделирование в МАТLAB. СПб.: БХВ-Петербург, 2008.
- 23. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. Основы цифровой обработки сигналов. 2-е изд. СПб.: БХВ-Петербург, 2005.
- 24. Черных И. В. Simulink: среда создания инженерных приложений. М.: ДИАЛОГ-МИФИ, 2004.
- 25. Черных И. В. Моделирование электротехнических устройств в MATLAB, SimPowerSystems и Simulink. СПб.: Питер, 2008.

# Список сокращений

### Список сокращений на английском языке

DFT — Discrete Fourier Transform (Дискретное преобразование Фурье).

EXT — Extension (Расширение).

FFT — Fast Fourier Transform (Быстрое преобразование Фурье).

FIR — Finite Impulse Response (Конечная импульсная характеристика (тип фильтра)).

GUI — Graphic User Interface (Графический интерфейс пользователя).

HTML — Hyper Text Markup Language (Язык гипертекстовой разметки).

IDFT — Inverse Discrete Fourier Transform (Обратное дискретное преобразование  $\Phi$ урье).

IC — Initial Condition (Начальное значение).

IFFT — Inverse Fast Fourier Transform (Обратное быстрое преобразование Фурье).

IIR — Infinite Impulse Response (Бесконечная импульсная характеристика (тип фильтра)).

LSB — Least Significant Bits (Младшее слово).

MSB — Most Significant Bits (Старшее слово).

ODE — Ordinary Differential Equations (Обыкновенные дифференциальные уравнения).

PDF — Portable Document Format (Формат переносного документа).

PSD — Power Spectral Density (Спектральная плотность мощности).

RMS — Root-Mean-Square (Среднеквадратичное значение).

RWV — Real Word Value (Вещественное число).

SI — Stored Integer (Целочисленный эквивалент).

SOS — Second-Order Sections (Рекурсивные звенья 2-го порядка).

## Список сокращений на русском языке

- АКФ автокорреляционная функция.
- АЧХ амплитудно-частотная характеристика.
- БИХ бесконечная импульсная характеристика (тип фильтра).
- БПФ быстрое преобразование Фурье.
- ВКФ взаимная корреляционная функция.
- ВЦД высокочастотный цифровой дифференциатор.
- ДПФ дискретное преобразование Фурье.

ИХ — импульсная характеристика.

КИХ — конечная импульсная характеристика (тип фильтра).

ЛДС — линейная дискретная система.

ЛФЧХ — линейная ФЧХ.

ННУ — нулевые начальные условия.

НЦД — низкочастотный цифровой дифференциатор.

ОДПФ — обратное дискретное преобразование Фурье.

ОДУ — обыкновенные дифференциальные уравнения.

ПЗ — полоса задерживания.

ПП — полоса пропускания.

ПТ — плавающая точка.

ПФ — полосовой фильтр.

ПЦД — полосовой цифровой дифференциатор.

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.

ШЦД — широкополосный цифровой дифференциатор.

# Предметный указатель

#### Μ

MATLAB:

- ◊ интерфейс 18
- ◊ система помощи 23
- ◊ справочная система 25
- 👌 установка 17

## S

Simulink:

- ◊ справочная система 26
- 👌 установка 17
- S-модель:
- ◊ ЛДС 313
- ◊ средств вычисления ДПФ 386
- ◊ структуры ЛДС 315

## Б

- Библиотека:
- Simulink 20
- ◊ собственная 233
- ◊ создание 233
- БИХ-фильтр 329
- Блок:
- ♦ Abs 109
- ♦ Add 100
- ♦ Algebraic Constraint 118
- ♦ Autocorrelation 152
- Band-Limited White Noise 60
- In Bandpass Filter 337
- ♦ Bandstop Filter 337
- ♦ Bias 103
- It Clear 122

- ◊ цифрового сигнала 380 ◊ цифрового фильтра 334 ◊ ЦФ с ФТ 353, 373 S-модель системы: ◊ запуск моделирования 56 ◊ компоненты 30 ◊ многоскоростной 408 ◊ настройка диагностики 54 ◊ настройка обмена данными 50 ◊ объекты MATLAB 124 ◊ останов моделирования 56 ◊ отладка 240 ◊ пауза в моделировании 56 ◊ редактирование 35 ◊ цифровой фильтрации 334 ◊ цифровой фильтрации с ФТ 380 ◊ этапы создания 31
- ♦ Bit Set 123
- Bitwise Operator 123
- Buffer 291
- ♦ Bus Creator 165
- ♦ Bus Selector 166
- ♦ Bus to Vector 176
- ♦ Chirp 278
- ♦ Chirp Signal 62
- Cholesky Factorization 148
- ◊ Cholesky Inverse 146
- ◊ Cholesky Solver 151
- Clock 63
- ♦ Compare To Constant 119
- Compare To Zero 120
- ♦ Complex Exponential 129
- ◊ Complex to Magnitude-Angle 112
- ♦ Complex to Real-Imag 112

- Constant 64, 279
- Onstant Diagonal Matrix 279
- ♦ Constant Ramp 298
- Convert 1-D to 2-D 295
- ♦ Convert 2-D to 1-D 295
- ♦ Convolution 299
- ♦ Correlation 153
- Counter Free-Running 68
- Counter Limited 69
- Create Diagonal Matrix 135
- Cumulative Product 129
- Cumulative Sum 131
- ♦ Data Type Conversion 177, 291
- ♦ Data Type Conversion Inherited 180
- ♦ Data Type Duplicate 181
- ♦ dB Conversion 131
- ♦ Demux 162
- ♦ Difference 132
- O Differentiator Filter 337
- Oigital Clock 70
- Oigital Filter 374
- Oigital Filter Design 334
- Oiscrete Filter 316
- ♦ Discrete FIR Filter 317
- ♦ Discrete Impulse 280
- ♦ Discrete State-Space 318
- ♦ Discrete Zero-Pole 320
- Oisplay 89, 262
- ♦ Divide 103
- ♦ Dot Product 105
- ♦ Downsample 300
- ♦ Embedded MATLAB Function 127
- Enabled and Triggered Subsystem 203
- Enabled Subsystem 195
- Extract Diagonal 136
- ◊ Extract Triangular Matrix 136
- Fcn 125
- ♦ FFT 387
- Filter Realization Wizard 353
- ♦ FIR Decimation 411
- ♦ FIR Interpolation 409
- ♦ FIR Rate Conversion 411
- ♦ Flip 294
- ♦ Floating Scope 94
- ♦ For Iterator Subsystem 211
- ♦ From 169
- ♦ From File 70
- ♦ From Workspace 72
- ◊ Gain 105

- Goto 167
- ♦ Ground 76
- ♦ Halfpass Filter 337
- ♦ Highpass Filter 337
- ♦ Hilbert Filter 337
- ♦ IC 181
- ♦ Identity Matrix 136, 281
- ♦ If Action Subsystem 203
- ♦ IFFT 389
- ◊ In 76
- Integer Delay 186
- ♦ LDL Inverse 145
- ♦ Logic Operator 121
- ♦ Lowpass Filter 337
- ♦ LU Factorization 147
- ♦ LU Inverse 144
- ♦ LU Solver 150
- ♦ Magnitude-Angle to Complex 113
- Manual Switch 169
- ♦ Math Function 110
- ♦ MATLAB Fcn 126
- ◊ Matrix Concatenate 115, 138
- Matrix Multiply 142
- Matrix Product 142
- Matrix Square 142
- Matrix Sum 143
- Matrix Viewer 263
- ♦ Maximum 154
- ♦ Mean 156
- ♦ Memory 187
- ♦ Merge 173
- ♦ Minimum 157
- ◊ MinMax 116
- ♦ Mux 161
- ♦ Normalization 133
- ♦ N-Sample Enable 295
- N-Sample Switch 296
- ♦ Nyquist Filter 337
- ♦ Out 90
- Overwrite Values 138
- Owerwrite Values 291
- ♦ Pad 301
- ♦ Peak Finder 303
- ♦ Permute Matrix 140
- ◊ Polynomial 118
- ♦ Probe 182
- Product 106
- Product of Elements 108
- Pulse Generator 78

- QR Factorization 148
- QR Solver 150
- ◊ Ramp 79
- ◊ Random Number 80
- Random Sources 281
- ◊ Rate Transition 183
- Real-Imag to Complex 113
- Relational Operator 120
- ♦ Repeat 305
- ♦ Repeating Sequence 80
- Repeating Sequence Interpolated 81
- Repeating Sequence Stair 82
- ◊ Reshape 116
- RMS 157
- ♦ Scope 91
- Selector 170, 291
- Shift Arithmetic 124
- ♦ Sign 111
- Signal From Workspace 282
- Signal Generator 83
- Signal To Workspace 264
- Sine Wave 83, 283
- ♦ Slider Gain 108
- Sort 158
- Standard Deviation 159
- ♦ Step 85
- Stop Simulation Scope 93
- ◊ Submatrix 141, 291
- ♦ Subsystem 191
- ♦ Switch 169
- Switch Case Action Subsystem 208
- ♦ Terminator 94
- ♦ Time Scope 267
- ♦ To File 94
- To Workspace 95
- ♦ Toeplitz 137
- ♦ Transpose 143
- Triggered Subsystem 199
- Triggered To Workspace 265
- Trigonometric Function 111
- ♦ Unary Minus 108
- ♦ Unbuffer 293
- Output One of the Uniform Random Number 86
- ♦ Unit Delay 188
- ♦ Upsample 306
- ♦ Variance 160
- Vector Concatenate 117
- ♦ Vector Scope 267
- ◊ Waterfall 272

- While Iterator Subsystem 215
- Width 185
- ♦ Window Function 306
- XY Graph 98
- Zero Crossing 309
- виртуальный 86
- невиртуальный 87
- Быстрое преобразование Фурье (БПФ) 386

#### Г

Группа блоков:

- Oiscrete 186, 315
- Filtering 334, 353
- Logic and Bit Operations 119, 122
- Math Functions 128
- Math Operations 99
- Port & Subsystems 203
- Signal Attributes 176, 297
- Signal Management 151, 290
- Signal Operations 298
- ♦ Signal Processing Sinks 262
- Signal Processing Sources 277
- ♦ Signal Routing 160
- Sinks 88
- ♦ Sources 58
- ♦ Transforms 386
- User-Defined Functions 124

### Д

Дискретное преобразование Фурье (ДПФ) 384

#### К

КИХ-фильтр 325 Команда: ◊ help 24 ◊ simulink 19 ◊ ver 19

#### Л

ЛДС:
◊ нерекурсивная 312
◊ рекурсивная 312

### Μ

- Масштабирование 356
  Математическая модель:
  ◊ ЛДС 312
  ◊ системы цифровой фильтрации 333
  Матрицы:
  ◊ арифметические операции 141
  ◊ обращение 144
  ◊ основные характеристики 128
  ◊ преобразование 138
- ◊ разложение 146
- типовые 135

## Η

Нулевые начальные условия 313

## 0

Операции:

- арифметические 99
- ◊ логические 119
- ◊ математические 128
- математической статистики 151
- ◊ округления 114
- ◊ отношения 119
- ◊ побитовые 122
- элементарные математические 109, 111, 115, 117

## П

Передаточная функция 314 Подсистема:

- ◊ виртуальная 189
- ◊ невиртуальная 189
- ◊ неуправляемая 189, 190
- ◊ определение 189
- 👌 управляемая 189, 203
- Последовательность:
- 👌 квантованная 251
- ◊ многоканальная 257, 287
- ◊ многосигнальная 254
- 👌 одноканальная 255, 286
- ◊ односигнальная 252
- ◊ типовая 310

- ◊ фреймов многоканальная 261, 289
- ◊ фреймов одноканальная 260, 288

#### Ρ

Разностное уравнение 312 Расстановка звеньев 356

### С

Система:

- ◊ математическая модель 28
- ◊ многоскоростная 398
- ◊ однократной децимации 402
- ◊ однократной интерполяции 399
- ◊ однократной передискретизации 404
- ◊ определение 28
- ◊ СЛАУ 149
- ◊ составляющие 30
- ◊ тип 29
- ◊ уравнений переменных состояния 313

Спектр 385

Спектральная плотность 385

- Структура:
- ◊ полифазная 405
- ◊ ЦФ 330
- ◊ ЦФ с ФТ 363

## Т

Тип данных 65, 66

### У

- Условие:
- ◊ простое 204
- ◊ составное 204

#### Φ

- ◊ df2tsos 331
- ♦ dfasymfir 332
- $\diamond \ dffir \ 332$
- ♦ dffirt 332
- ♦ dfsymfir 332
- ♦ fvtool 322
- ♦ tf2sos 329

## Χ

Характеристика затухания 327

## Ц

Цифровой фильтр 324 Цифровой фильтр с ФТ 353