

ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

621.314(07)
Б896

О.Г. Брылина, Л.И. Цытович

**МНОГОЗОННЫЕ РАЗВЕРТЫВАЮЩИЕ
ПРЕОБРАЗОВАТЕЛИ ДЛЯ СИСТЕМ
УПРАВЛЕНИЯ ЭЛЕКТРОПРИВОДАМИ**

Учебное пособие к лабораторным работам

Челябинск
2010

Министерство образования и науки Российской Федерации
Южно-Уральский государственный университет
Кафедра «Электропривод и автоматизация
промышленных установок»

621.314(07)
Б896

О.Г. Брылина, Л.И. Цытович

**МНОГОЗОННЫЕ РАЗВЕРТЫВАЮЩИЕ
ПРЕОБРАЗОВАТЕЛИ ДЛЯ СИСТЕМ УПРАВЛЕНИЯ
ЭЛЕКТРОПРИВОДАМИ**

Учебное пособие к лабораторным работам

Челябинск
Издательский центр ЮУрГУ
2010

УДК 621.314(076.5)+681.586(076.5)+621.317(076.5)+62-83-52(076.5)
Б896

Одобрено
учебно-методической комиссией энергетического факультета

Рецензенты:
О.И. Осипов, А.С. Сарваров

Брылина, О.Г.
Б896 Многозонные развертывающие преобразователи для систем управления электроприводами: учебное пособие к лабораторным работам / О.Г. Брылина, Л.И. Цытович. – Челябинск: Издательский центр ЮУрГУ, 2010. – 120 с.

Учебное пособие посвящено виртуальному моделированию развертывающих преобразователей и систем управления на их основе в математическом пакете MatLab+Simulink в соответствии с индивидуальным заданием.

В одном из разделов учебного пособия приведено краткое руководство по работе с графическим редактором Paint, который входит в состав операционной системы Windows.

Учебное пособие предназначено для студентов дневной и заочной форм обучения по специальностям 140604 «Электропривод и автоматизация промышленных установок и технологических комплексов». Также монография будет полезна студентам других направлений электротехнического профиля.

УДК 621.314(076.5)+681.586(076.5)+621.317(076.5)+62-83-52(076.5)

ВВЕДЕНИЕ

Учебное пособие предназначено для студентов высших учебных заведений по специальности 140604 «Электропривод и автоматизация промышленных установок и технологических комплексов» при изучении курсов «Теория разветвляющихся систем», «Вентильные преобразователи», «Элементы систем автоматики», «Импульсные и релейные системы автоматического управления» и других, касающихся вопросов построения систем технологической автоматики с применением сравнительно нового класса устройств – многозонных разветвляющихся преобразователей.

Основная задача данного цикла лабораторных работ – научить студентов виртуальному моделированию разветвляющихся преобразователей и систем управления на их основе в математическом пакете MatLab+Simulink. На базе стандартных блоков и звеньев из встроенных библиотек MatLab поясняется процесс создания модели того или иного устройства или системы управления, исследуются их характеристики в соответствии с индивидуальным заданием. В результате студенты приобретают навыки работы с виртуальными электронными схемами, что способствует закреплению теоретического лекционного материала.

Индивидуальное задание к лабораторной работе предусматривает предварительный расчет параметров схемы и ее модели.

Для проверки знаний студентов перед лабораторной работой проводится коллоквиум на основе контрольных вопросов, помещенных в пособие.

После выполнения лабораторной работы студенты составляют индивидуальные отчеты.

1. КРАТКОЕ РУКОВОДСТВО ПО РАБОТЕ В MATLAB. ОСНОВНЫЕ БЛОКИ MATLAB

Виртуальное моделирование осуществляется с помощью пакета расширения Simulink, входящих в комплекс математического пакета MatLab.

Математическая модель представляет собой набор взаимосвязанных блоков, включающих в себя электрические элементы и измерительные приборы.

1.1. Запуск MatLab, создание и открытие файла

Для запуска программы, необходимо щелкнуть два раза правой кнопкой мыши на пиктограмму «MatLab»  расположенную на рабочем столе или непосредственно в папке с установленным программным обеспечением MatLab, по умолчанию это диск C:\MATLAB.

В меню «File» (Файл) выбрать пункт «New» (Новый) и далее «Model» (Модель) (рис. 1.1).

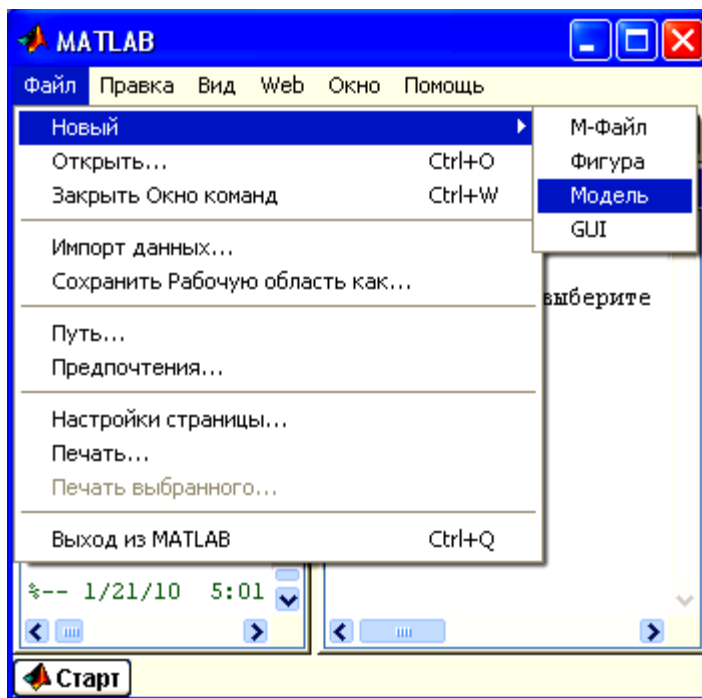


Рис. 1.1. Внешний вид окна
для создания
модели в среде
MatLab+Simulink

Для открытия ранее созданного файла в меню «File» (Файл) выбрать пункт «Open» (Открыть) (см. рис. 1.1) или нажать на кнопку «Открыть

модель» , расположенную на панели инструментов MatLab.

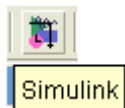
В раскрывшемся диалоговом окне необходимо найти и выбрать требуемый файл, после чего нажать кнопку «Открыть».

Открыть ранее созданный файл также можно двойным (или одинарным – в зависимости от настроек компьютера) щелчком левой кнопки мыши, если щелкнуть по выбранному файлу с расширением «.mdl».

1.2. Библиотеки Simulink, SimPowerSystems и Simulink Extras

Для открытия пакета расширения Simulink и его библиотек (рис. 1.2)

нажать на кнопку «Simulink»



расположенную, на панели инструментов окна MatLab.

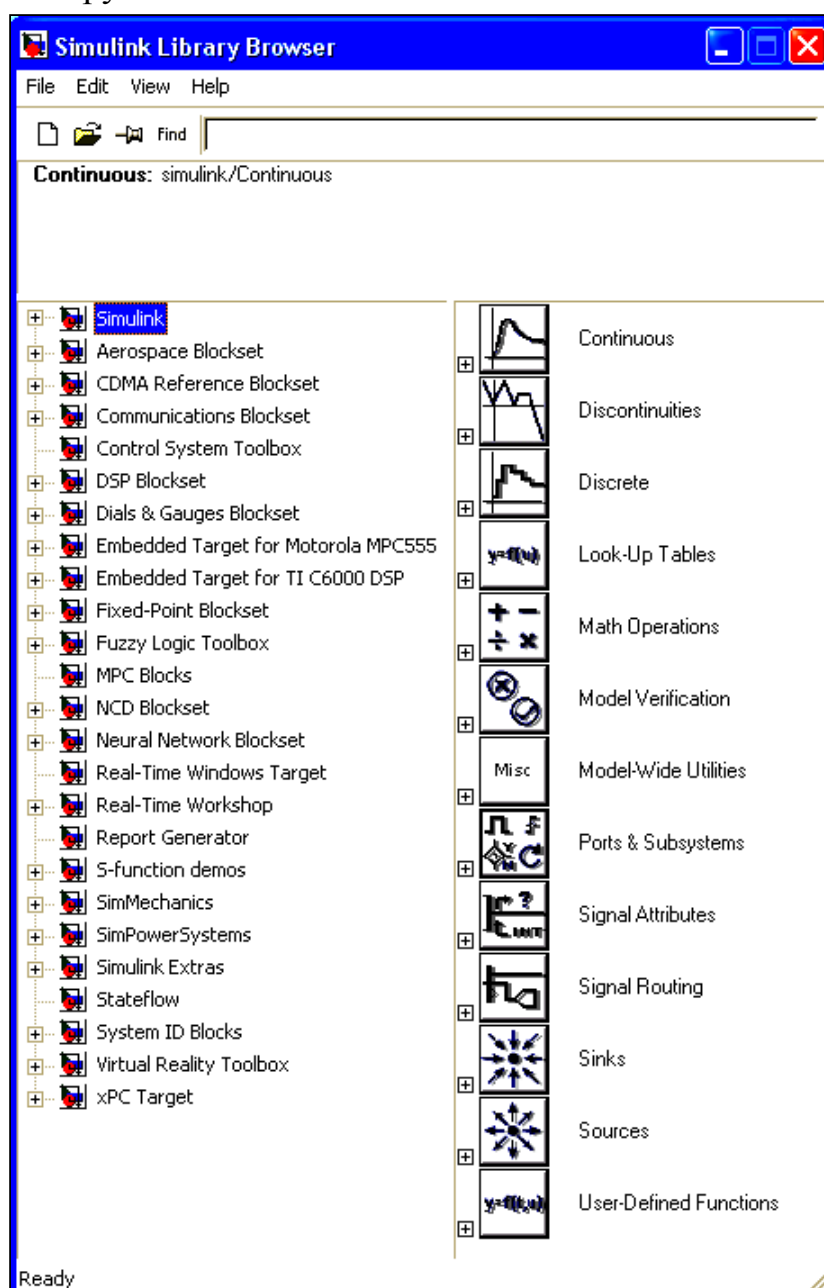


Рис. 1.2. Внешний вид библиотеки Simulink

Для создания нового файла-модели следует нажать на кнопку «Создать

модель»,  расположенную на панели инструментов Simulink Library Browser (см. рис. 1.2) или в меню «File» (Файл) выбрать пункт «New» (Новый) → «Model» (рис. 1.3) или нажать кнопки Ctrl+N.

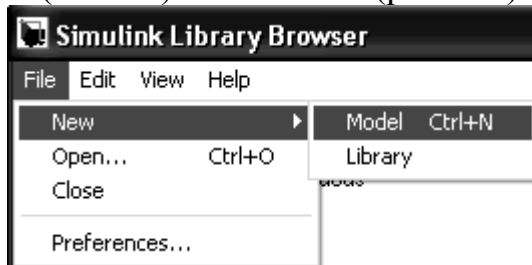


Рис. 1.3. Создание новой модели в среде MatLab+Simulink

MatLab оснащен сравнительно большим количеством библиотек, которые представлены в виде раскрывающегося списка (см. рис. 1.2). В рамках курса «Теория разветвляющихся систем» остановимся подробнее только на тех разделах библиотек (Simulink, SimPowerSystems и Simulink Extras), которые будут полезны для реализации моделей данного курса. Более подробную информацию по всем библиотекам можно получить в [1–3] или во встроенной справочной системе MATLAB. Доступ к справочной системе осуществляется нажатием «горячей» кнопки F1 или нажатием кнопок «Help» («Помощь») или , расположенных на панели инструментов основного окна MatLab (см. рис. 1.1) или в окне параметров любого стандартного блока, взятого из библиотеки.

1.3. Библиотека SIMULINK

Библиотека блоков Simulink представляет собой набор визуальных объектов. Соединяя отдельные объекты (модули) между собой линиями связей, можно составлять функциональную блок-схему любого устройства.

Для любого блока можно получать требуемое число копий и использовать каждую из них автономно. Более того, практически для всех блоков существует возможность индивидуальной настройки: пользователь может изменить как внутренние параметры блоков (например, количество входов), так и внешнее оформление (размер, цвет, имя и т.д.).

Для удобства работы пользователя библиотека блоков разбита на следующие разделы:

- Continuous – непрерывные элементы;
- Discontinuities – дискретные элементы;
- Discrete – дискретные элементы;
- Look-Up Tables – таблицы поиска;
- Math Operations – математические операторы;
- Model Verification – проверка модели;
- Model-Wide Utilities – утилиты расширения модели;

- Ports & Subsystems – входы/выходы и подсистемы;
- Signal Attributes – атрибуты сигнала;
- Signal Routing – пересылка сигнала;
- Sinks – получатели;
- Sources – источники;
- User-Defined Functions – функции определяемые пользователем.

Чтобы открыть интересующий раздел библиотеки, достаточно дважды щелкнуть на нем левой кнопкой мыши.

Каждый блок, входящий в библиотеку Simulink, имеет, по крайней мере, один параметр настройки. Задавая требуемое значение параметра (или выбирая его из предлагаемого меню), пользователь имеет возможность скорректировать функцию, реализуемую данным блоком. Чтобы открыть окно настройки параметров, нужно дважды щелкнуть левой кнопкой мыши на изображении блока. Однако возможность изменять значения параметров появляется только после того, как блок будет помещен в окно блок-диаграммы.

Окна настройки параметров всех библиотечных блоков имеют идентичную структуру и содержат краткую характеристику блока, поля ввода (или выбора) значений параметров блока и 4 кнопки управления (рис. 1.4).



Рис. 1.4. Кнопки управления окна настройки

- ОК – применить изменения и закрыть окно настроек;
- Cancel – отменить изменения и закрыть окно настроек;
- Help – вызов файла помощи в формате html;
- Apply – применить изменения, не закрывая окно настроек.

Измененные значения параметров вступают в силу только после «нажатия» кнопок Apply или ОК. Чтобы запустить модель на исполнение с новыми параметрами, закрывать окно настроек не обязательно.

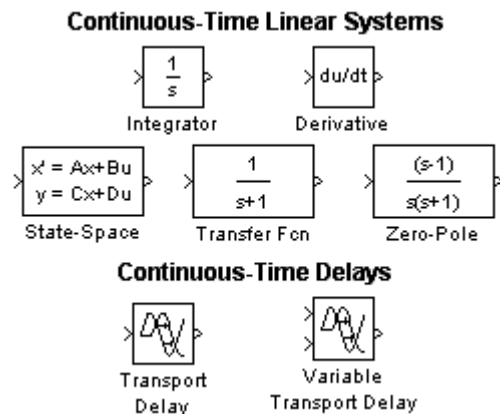
Далее описан состав и назначение блоков некоторых наиболее часто используемых разделов библиотеки Simulink.

1.3.1. Раздел Continuous

Раздел Continuous (Непрерывные элементы) содержит блоки, которые разделяются на следующие группы (рис. 1.5):

- Continuous-Time Linear Systems – блоки общего назначения (интеграторы, дифференциаторы) и блоки линейных стационарных звеньев;

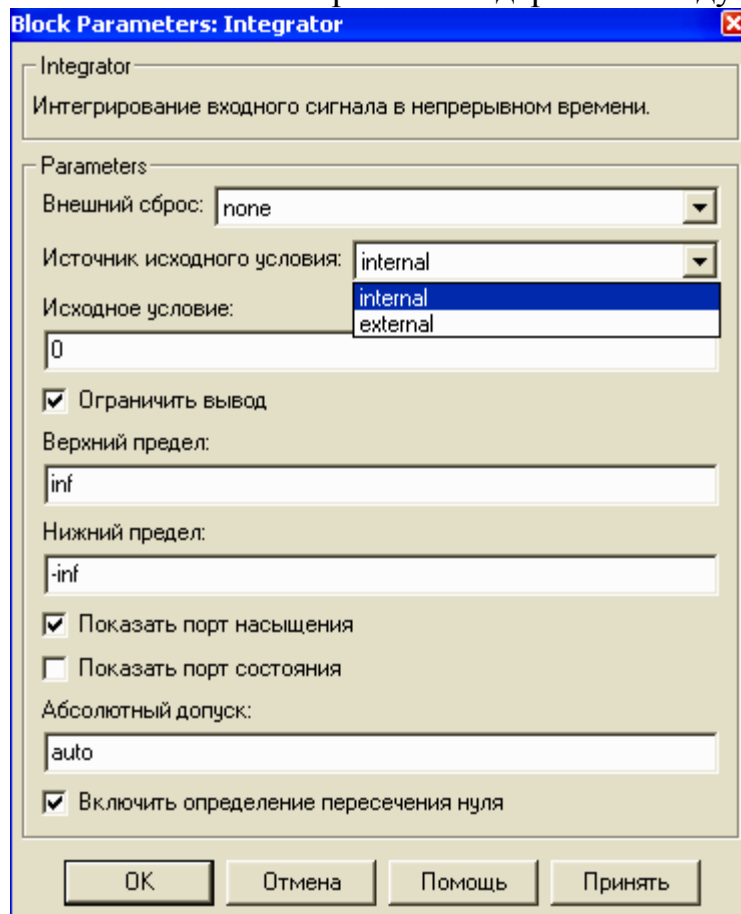
- Continuous-Time Delays – блоки задержки сигнала.



**Рис. 1.5. Внешний вид
блоков раздела
Continuous**

1. Блок Integrator позволяет осуществлять интегрирование входного сигнала в непрерывном времени.

Окно его настройки содержит следующие параметры (рис. 1.6).



**Рис. 1.6. Параметры
блока Integrator**

• External reset (Внешний сброс) – подключение дополнительного управляющего сигнала. Может принимать следующие значения:

- none – дополнительный управляющий сигнал не используется;
- rising – для управления используется нарастающий сигнал;
- falling – для управления используется убывающий сигнал;

- **either** – на работу блока влияет изменение управляющего сигнала в любом направлении.

- **Initial condition source** (Источник исходного условия) – определение источника установки начального значения выходного сигнала (внутренний – **internal** или внешний **external**). Если выбранные пользователем значения этих двух параметров предполагают наличие дополнительных входных сигналов, то на графическом изображении блока появляются дополнительные входные порты (после щелчка на кнопке **Apply** в окне настройки блока).

- **Initial condition** (Исходное условие) – начальное значение выходной величины, вводится в строке редактирования или как числовая константа либо в виде вычисляемого выражения (по умолчанию «0»).

- **Limit output** (Ограничить выход) – ограничение величины выхода (этот флажок определяет, будут ли использоваться следующие два параметра):

- **Upper saturation limit** – верхнее предельное значение выходной величины, по умолчанию не ограничено (значение «inf»);

- **Lower saturation limit** – нижнее предельное значение выходной величины (по умолчанию параметр равен «-inf»).

При установке следующих двух галочек у блока **Integrator** появляется дополнительно два вывода:

- **Show saturation port** – показать порт насыщения;

Если флажки **Limit output** и **Show saturation port** установлены, то на правой стороне изображения блока-интегратора появляется обозначение нового выходного порта и при переходе выходного значения интегратора через верхнюю или нижнюю границу на этом выходе формируется единичный сигнал.

- **Show state port** – показать порт состояния.

Установка этого флажка также приводит к появлению дополнительного выхода блока (**state port**), возникающего на верхней стороне изображения блока. Сигнал, который подается на этот порт, совпадает с главным выходным сигналом, но в отличие от него может быть использован только для прерывания алгебраического цикла или для согласования состояния подсистем модели.

- **Absolute tolerance** (Абсолютный допуск) – допустимая предельная величина абсолютной погрешности.

Для внешней установки начального условия перед блоком **Integrator** можно включить блок **IC** (**Initial Condition**). Описание этого блока приведено ниже в п. 1.3.9.

2. Блок **Derivative** (Численная производная du/dt) не имеет изменяемых параметров, осуществляет дифференцирование поданного на него сигнала по времени (du/dt).

3. Блоки State-Space, Transfer Fcn и Zero-Pole реализуют динамические звенья, описываемые линейными дифференциальными уравнениями с

Block Parameters: State-Space

State Space
Модель в пространстве состояний:
 $dx/dt = Ax + Bu$
 $y = Cx + Du$

Parameters
A:
[1]
B:
[1]
C:
[1]
D:
[1]
Начальные условия:
[0]
Абсолютный допуск:
auto

OK Отмена Помощь Принять

Рис. 1.7. Параметры блока State-Space

постоянными коэффициентами, то есть линейные стационарные звенья (см. рис. 1.5):

- State-Space – позволяет задать линейное звено путем ввода четырех матриц его пространства состояний ($dx/dt = Ax + Bu$; $y = Cx + Du$) (рис. 1.7);

- Transfer Fcn – позволяет задать линейное звено путем ввода его передаточной функции (рис. 1.8) в соответствии с выражением (1.1). Где nn и nd обозначают коэффициенты, стоящие перед s (в убывающей степени), числителя и знаменателя соответственно. Числитель может быть вектором или

$$H(s) = \frac{y(s)}{u(s)} = \frac{\text{num}(s)}{\text{den}(s)} = \frac{\text{num}(1)s^{nn-1} + \text{num}(2)s^{nn-2} + \dots + \text{num}(nn)}{\text{den}(1)s^{nd-1} + \text{den}(2)s^{nd-2} + \dots + \text{den}(nd)} \quad (1.1)$$

Block Parameters: Transfer Fcn

Transfer Fcn
Матричное выражение для числителя, векторное выражение для знаменателя. На выходе - ширина равна числу строк в числителе. Коэффициенты - для степеней s по убыванию.

Parameters
Числитель:
[1]
Знаменатель:
[1 1]
Абсолютный допуск:
auto

OK Отмена Помощь Принять

Рис. 1.8. Параметры блока Transfer Fcn

Примечание: в выражениях (1.1 и 1.2) обозначение «s» соответствует операнду p.

- Zero-Pole – позволяет задать звено посредством указания векторов значений его полюсов и нулей, а также значения коэффициента передачи, в соответствии с выражением (1.2).

Где Z представляет собой вектор нулей, P – вектор полюсов, и K – коэффициент усиления. Z

может быть вектором или матрицей, P – вектором, K – скаляром или вектором, длина которого равна числу строк в Z . Число полюсов должно быть больше или равно числу нулей.

$$H(s) = K \frac{Z(s)}{P(s)} = K \frac{(s - Z(1))(s - Z(2)) \dots (s - Z(m))}{(s - P(1))(s - P(2)) \dots (s - P(n))} \quad (1.2)$$

Если полюса и нули являются комплексными числами, они должны быть сопряженными парами комплекса.

4. Блок Transport Delay – обеспечивает задержку сигнала на заданное

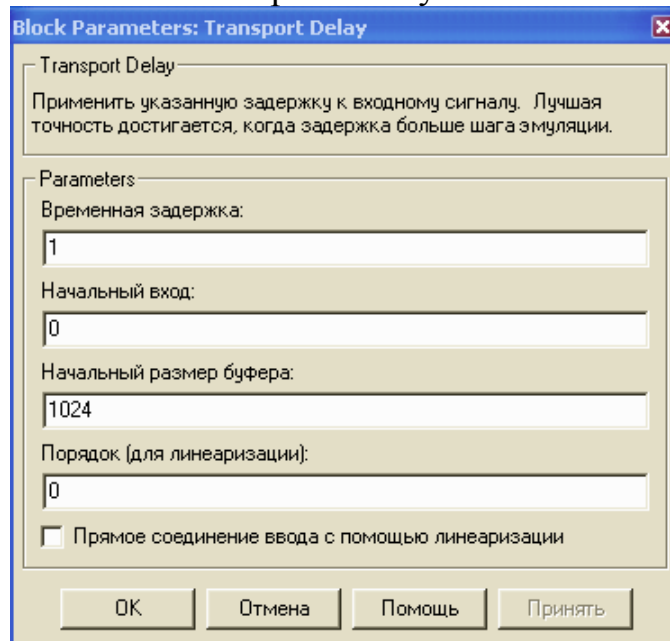


Рис. 1.9. Параметры блока Transport Delay

количество шагов (не обязательно целое число) модельного времени (рис. 1.9). Настройка блока производится с помощью трех параметров:

- Time delay (Время задержки) – количество шагов модельного времени, на которое задерживается сигнал. Может вводиться либо в числовой форме, либо в форме вычисляемого выражения;

- Initial input (Начальный вход) – значение амплитуды входного сигнала в момент инициализации блока (по умолчанию равно 0);

- Initial buffer size

(Начальный размер буфера) – объем памяти (в байтах), выделяемой в рабочей области MatLab для хранения параметров задержанного сигнала. Значение параметра должно быть кратно 8 (по умолчанию равно 1024).

- Order (Порядок) – порядок приближения для программ линеаризации.

5. Блок Variable Transport Delay (Изменяемая задержка передачи) – позволяет задавать величину задержки управляемую извне. Блок имеет дополнительный вход. Подаваемый на него сигнал определяет длительность задержки информационного сигнала, поступающего на основной вход блока.

Данный блок, как и предыдущий, имеет 3 параметра настройки:

- Maximum delay – Максимальная задержка;
- Initial input – Начальный ввод;
- Buffer size – Начальный размер буфера.

Назначение двух последних параметров идентично назначению одноименных параметров блока Transport delay.

Параметр *Maximum delay* определяет наибольшую допустимую величину задержки информационного сигнала. Если величина задержки, определяемая управляющим сигналом, превышает этот порог, то она принудительно устанавливается равной параметру *Maximum delay*. Его значение измеряется числом шагов модельного времени, может иметь дробное значение и вводиться либо в числовой форме, либо в форме вычисляемого выражения.

1.3.2. Раздел *Discontinuities*

В раздел *Discontinuities* входят блоки (рис. 1.10), с помощью которых описывается поведение нелинейных систем или систем с точкой разрыва.

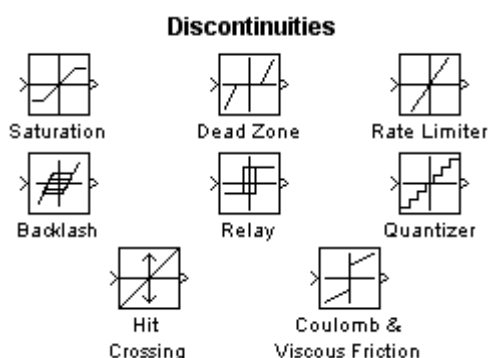


Рис. 1.10. Разделы библиотеки *Discontinuities*

1. *Saturation* – блок насыщения – ограничивает входной сигнал верхней и нижней величиной насыщения;
2. *Dead Zone* – зона нечувствительности (мертвая зона) – для входного сигнала изменяющегося в пределах зоны нечувствительности на выходе будет ноль. В остальных точках выходной равен входному сигналу;
3. *Rate Limiter* – ограничение скорости изменения (возрастания и убывания) сигнала;
4. *Backlash* – моделирование зазора, где ширина мертвой зоны указывает величину зазора в системе;
5. *Relay* – релейный элемент – переключение – данный блок работает по аналогии с обычным реле: если входной сигнал превышает некоторое предельное значение (порог переключения), то на выходе блока формируется постоянный «разрешающий» сигнал.

Блок *Relay* (рис. 1.11) имеет четыре параметра:

- *Switch on point* (Точка включения) – уровень входного сигнала (пороговое значение), при превышении которого происходит «включение» реле;
- *Switch off point* (Точка выключения) – уровень входного сигнала, ниже которого реле «выключается»;

- Output when on (Выход при включенном состоянии) – уровень выходной величины при включенном реле;
- Output when off (Выход при выключенном состоянии) – уровень выходного сигнала при выключенном реле.

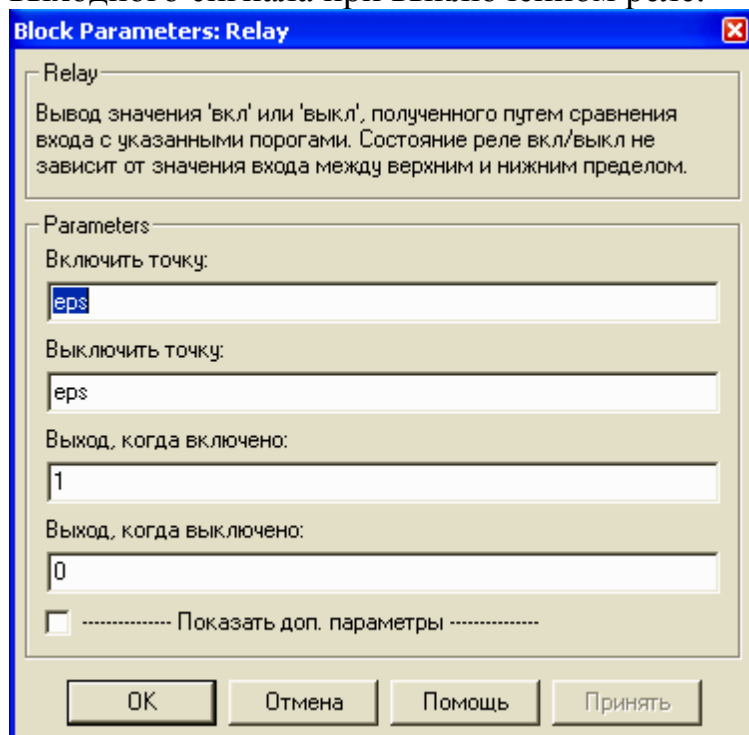


Рис. 1.11. Параметры блока Relay

Значения всех параметров блока могут вводиться либо в форме числовых констант, либо в форме вычисляемых выражений.

6. Блок Quantizer (Квантователь) позволяет осуществлять дискретизацию (квантование) входного сигнала по его величине на заданном интервале.

7. Блок Hit Crossing (Обнаружить пересечение) – позволяет зафиксировать состояние, в момент пересечения входным сигналом некоторого заданного значения.

Блок Hit Crossing имеет три параметра настройки:

- Hit crossing offset – определяет значение, «пересечение» которого необходимо идентифицировать;
- Hit crossing direction позволяет указать направление «пересечения», которое интересует разработчика модели; значение этого параметра выбирается с помощью «выпадающего» меню, которое содержит три пункта: Rising (возрастание), Falling (убывание), Either (в обоих направлениях).
- Show output port (показать выходной порт) – флажок, с помощью которого можно выбрать формат использования блока.

При одновременном выполнении условий, задаваемых параметрами Hit crossing offset и Hit crossing direction, на выходе блока формируется единичный импульс. Его длительность определяется значением параметра

Sample time блока, предшествующего в модели блоку Hit crossing. Если этот параметр отсутствует, то единичный сигнал на выходе блока Hit crossing существует до его следующего срабатывания.

8. Блок Coulumb&Viscous Friction (Кулоново и вязкое трение) реализует нелинейную зависимость типа линейной с предварительным натягом.

Блок Dead Zone (Мертвая зона) – заменяет значение входного сигнала, лежащее в заданном диапазоне, нулем.

1.3.3. Раздел Discrete

В раздел Discrete входят блоки, с помощью которых в модели может быть описано поведение дискретных систем. Различают два основных типа таких систем: системы с дискретным временем и системы с дискретными состояниями. Раздел содержит 8 блоков (рис. 1.12):

1. Блок Unit Delay предназначен для задержки сигнала на заданное число периодов (шагов модельного времени).

Параметрами настройки блока Unit Delay:

- Начальное значение сигнала (Initial condition) – значение амплитуды сигнала в момент активизации блока. Оно может быть задано либо в виде числовой константы, либо в виде вычисляемого выражения;
- Величина задержки (Sample time) – определяет число шагов модельного времени, в течение которого сохраняется значение сигнала, поступившего на вход блока Unit Delay. Значение этого параметра должно быть положительным.

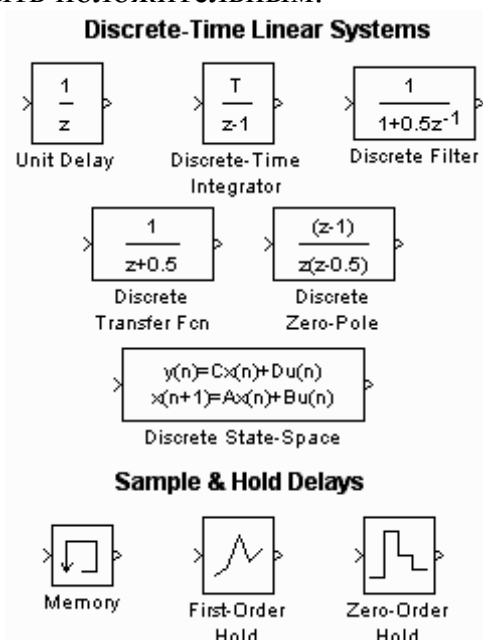


Рис. 1.12. Внешний вид блоков раздела Discrete

2. Блок Discrete-Time Integrator – дискретный сумматор (счетчик) времени. Выполняет суммирование интервалов времени между поступлениями входного сигнала. Блок может быть использован для управления логикой работы отдельных компонентов или модели в целом. В

частности, суммарная длительность работы некоторой подсистемы может служить условием «досрочного» окончания моделирования.

Блок имеет следующие параметры настройки (рис. 1.13):

- используемый метод интегрирования (Integrator method: прямой метод Эйлера (Forward Euler), обратный метод Эйлера (Backward Euler) и метод трапеций (Trapezoidal);

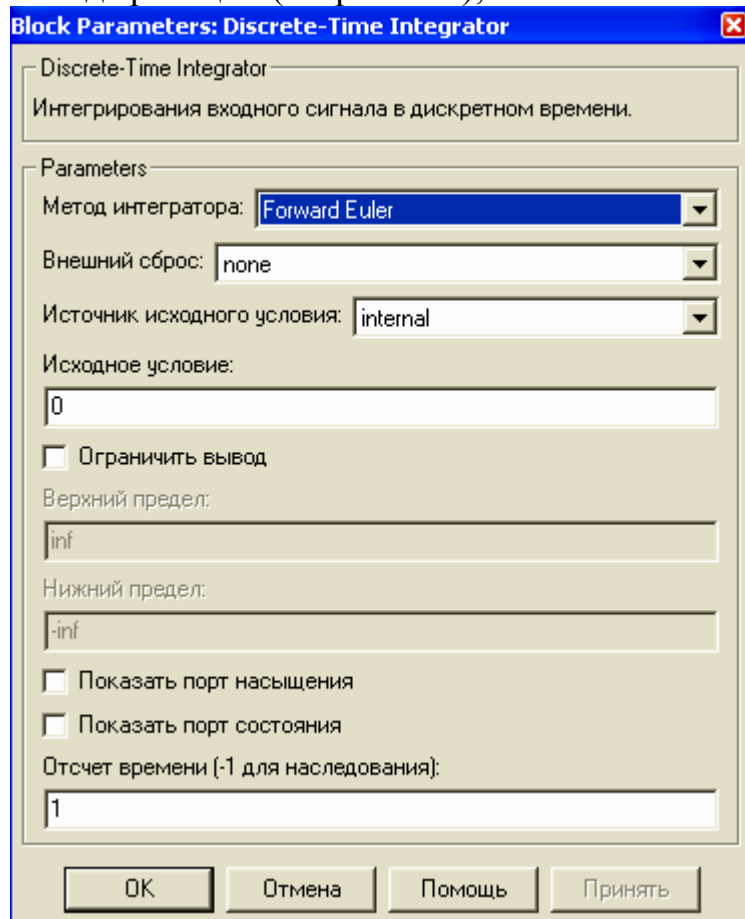


Рис. 1.13. Параметры блока Discrete-Time Integrator

- подключение дополнительного управляющего сигнала – внешний сброс (External reset):

none – дополнительный управляющий сигнал не используется;

rising – для управления используется возрастающий сигнал (по переднему фронту сигнала);

falling – для управления используется ниспадающий сигнал (по заднему фронту сигнала);

either – на работу блока влияет любое изменение амплитуды управляющего сигнала;

level – для управления используется заданный уровень сигнала.

- использование внешней установки начального значения входного сигнала (Initial condition source):

internal – используется собственная установка начального значения сумматора;

external – установка начального значения производится извне.

Внимание! Если выбранные пользователем значения двух последних рассматриваемых параметров предполагают наличие дополнительных входных сигналов, то на графическом изображении блока появляются дополнительные входные порты (после нажатия кнопки Apply в окне настроек блока).

- начальное состояние сумматора – исходное условие (Initial condition) – это значение вводится в строке редактирования либо как числовая константа, либо в виде вычисляемого выражения;

- флажок Limit output (Ограничение выходного значения сумматора) определяет, будут ли использоваться следующие ниже 4 параметра настройки:

- а) Upper saturation limit – верхнее предельное значение времени, по умолчанию – не ограничено «inf»;

- б) Lower saturation limit – нижнее предельное значение времени, по умолчанию параметр имеет значение «-inf»;

- в) Show saturation port – флажок показать порт насыщения;

- г) Show state port флажок показывает порт состояния.

Если флажок Limit output установлен, то при переходе значения сумматора через верхний или нижний предел на дополнительных выходах блока (Saturation port и State port) формируется единичный сигнал.

Чтобы этот сигнал можно было использовать для управления работой модели, флажки Show saturation port и Show state port должны быть включены (при этом на графическом изображении блока появляются обозначения портов).

Особенность порта State port состоит в том, что снимаемый с него сигнал может быть использован только для прерывания алгебраического цикла или для согласования состояния подсистем модели.

3. Блок Discrete Filter – позволяет задать дискретное звено через дискретную передаточную дробнорациональную функцию относительно z^{-1} .

4. Блок Discrete Transfer Fcn – позволяет задать линейное дискретное звено через дискретную передаточную дробно-рациональную функцию относительно z .

5. Блок Discrete Zero-Pole – используется для определения дискретного звена через указание значений нулей и полюсов дискретной передаточной функции относительно z .

6. Блок Discrete State-Space – используется для определения дискретного линейного звена путем указания матриц его состояния (блок формирования состояния системы);

7. Блок Memory – используется для задержки сигнала на один шаг модельного времени.

Блок Memory имеет два параметра настройки:

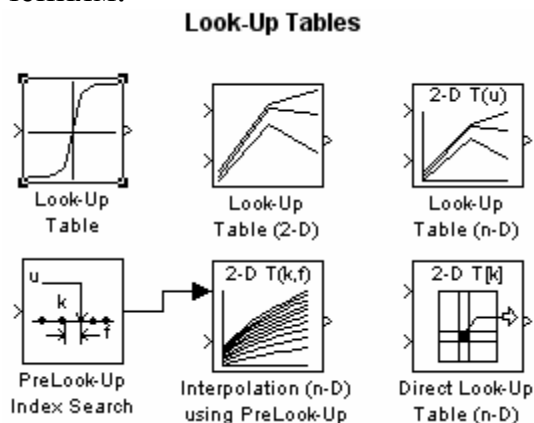
а) Initial condition (начальное условие) – значение амплитуды входного сигнала в начальный момент времени (на момент инициализации блока). В большинстве случаев целесообразно принимать равным нулю;

б) Inherit sample time (Наследование шага времени) – позволяет выбрать величину шага, на который будет производиться задержка сигнала. Если этот флажок сброшен, то используется минимальная задержка, равная 0,1 единицы модельного времени; если же он установлен, то величина задержки равна значению Sample time блока, который предшествует блоку Memory.

8. Блоки Zero-Order Hold и First-Order Hold – представляют собой экстраполяторы нулевого и первого порядка соответственно.

1.3.4. Раздел Look-Up Tables

В разделе Look-Up Tables (Табличные функции) (рис. 1.14) сосредоточены блоки, формирующие выходной сигнал, зависимость которого от входного сигнала задана с помощью таблицы соответствий. Блоки осуществляют линейную интерполяцию по содержащимся в ней значениям.



**Рис. 1.14. Внешний вид
блоков раздела Look-Up
Tables**

Блок Look-up Table (Таблица поиска) – выполняет линейную интерполяцию входного сигнала в соответствии с заданной табличной функцией.

Блок Look-up Table (2D) – производит линейную интерполяцию двумерного входного сигнала.

1.3.5. Раздел Math Operation

В разделе Math Operations (Математические операторы) (рис. 1.15) содержатся блоки, которые реализуют некоторые встроенные

математические функции системы MatLab. Они объединены в четыре группы.

Первую группу образуют блоки, реализующие элементарные математические функции. К таким блокам относятся:

1. Блок Sum – суммирует или вычитает поступающие на его входы сигналы, может использоваться в двух режимах:

- сложения входных сигналов (в том числе с разными знаками);
- суммирования элементов вектора, поступающего на вход блока.

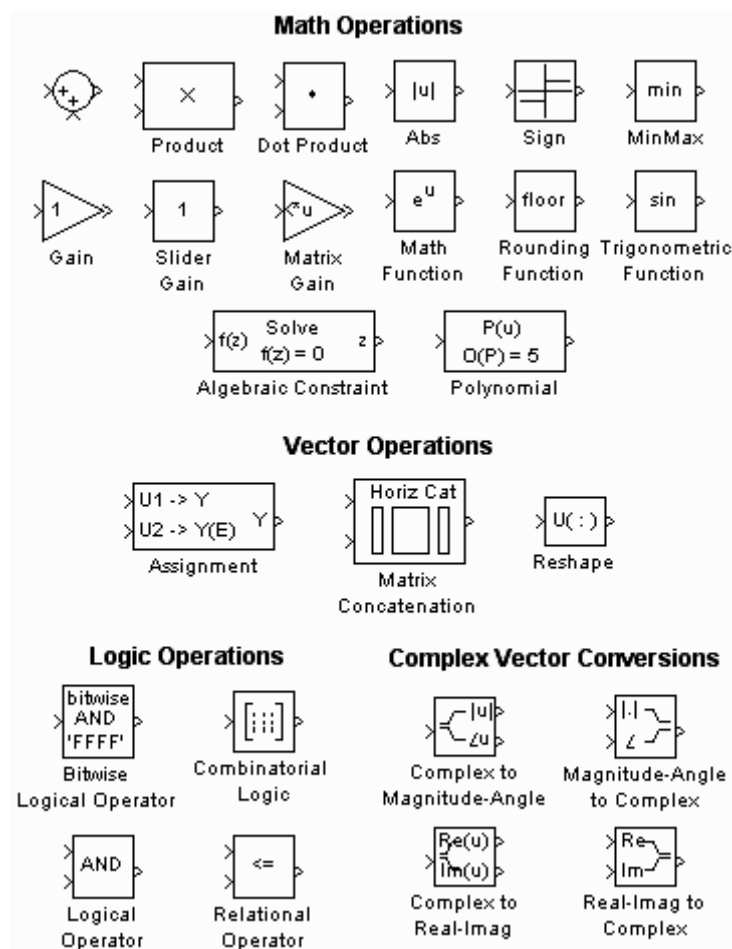


Рис. 1.15. Внешний вид блоков раздела Math Operations

Для управления режимами работы блока используются параметры Icon Shape (Форма изображения) и List of signs (Список знаков).

Первый параметр может принимать два значения: round (круглый) и rectangular (прямоугольный).

Значения второго параметра могут задаваться одним из трех способов:

- в виде последовательности знаков «+» и «-». При этом число знаков определяет число входов блока, а сам знак – полярность входного сигнала;
- в виде целого положительного числа (больше 1). Его значение равно числу входов блока, при этом все входы считаются

положительными (например, ввод константы 4 аналогичен вводу «списка знаков» в форме ++++);

- в виде символа «1», который указывает, что блок используется в режиме суммирования элементов вектора (в этом случае вход у блока один).

2. Блок Product – выполняет умножение или деление нескольких входных сигналов (величин). В качестве параметров настройки могут указываться число входов блока (Number of inputs) и вид выполняемой операции (Multiplications). Входные величины могут быть векторными или матричными. В списке Multiplications (Умножение) выбирается способ умножения входных величин: Element wise – поэлементное умножение входных векторов или матриц; Matrix – матричное умножение векторов или матриц.

Если значение параметров Number of inputs (Количество входов) является положительным числом, которое больше 1, то все входные величины перемножаются. Если в качестве значения параметра ввести «1», то будет вычисляться произведение элементов единственного входного вектора (в этом случае на изображении блока выводится символ P). В случае, когда результат выполнения должен содержать деление на некоторые входные величины, в поле Number of inputs (Количество входов) следует вводить последовательность символов «*» или «/» (по числу входов блока) в зависимости от того, умножается или делится результат на соответствующую входную величину. Значения этих параметров задаются также, как и при настройке блока Sum. Если выбрано матричное умножение, то символ «/» обозначает умножение на матрицу, обратную матрице соответствующей входной величины.

3. Блок Dot Product – осуществляет умножение двух входных величин, если они являются скалярами, или определяет сумму произведений элементов двух входных векторов одинаковой длины. Если векторы являются комплексными, то перед умножением первый вектор (верхний входной порт) заменяется комплексно сопряженным;

4. Блок Abs – формирует абсолютное значение входного сигнала (этот блок не имеет параметров настройки);

5. Блок Sign – формирует на выходе «1» для положительного входного сигнала, «-1» для отрицательного входного сигнала и «0» для нулевого.

6. Блок MinMax обеспечивает поиск минимального или максимального элемента входного вектора. Цель поиска (минимум или максимум) задается в окне настроек блока. Второй параметр настройки – число входов блока. Если входом является скалярная величина, то выходная величина совпадает с входной. Если входов несколько, отыскивается минимум/максимум среди входов;

7. Блок Gain – является линейным усилительным звеном (умножитель). Осуществляет умножение входного сигнала на постоянную величину (или

вектор), значение которой (элементы которого) задаются в диалоговом окне настройки в поле Gain (Коэффициент усиления) (рис. 1.16).

Входная величина блока (u) может быть скалярной, векторной или матричной. Если входной сигнал – вектор длиной N элементов, коэффициент усиления должен быть вектором той же длины.

В списке Multiplication (Умножение) выбирается один из следующих способов умножения входной величины на вектор K коэффициентов усиления:

- Element wise ($K \cdot u$) – поэлементное умножение входного вектора на вектор коэффициентов усиления;
- Matrix ($K \cdot u$) – матричное умножение вектора коэффициентов усиления на матрицу входной величины;
- Matrix ($u \cdot K$) – матричное умножение матрицы входной величины на вектор коэффициентов усиления;
- Matrix ($K \cdot u$)(u vector) – матричное умножение векторов K и u .

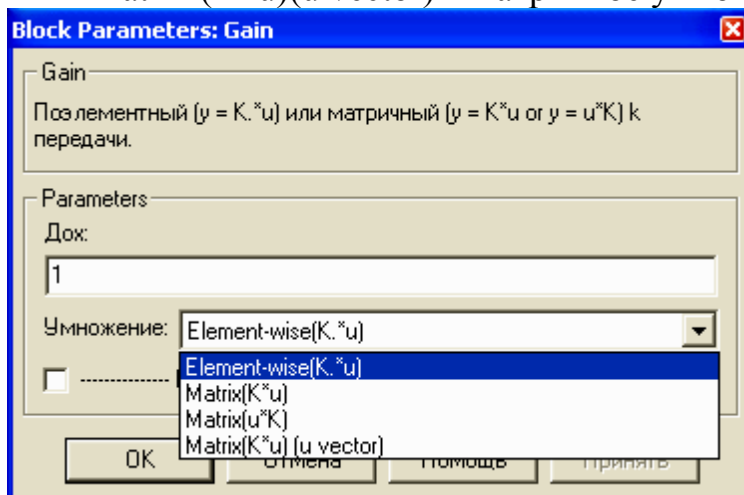


Рис. 1.16. Параметры настройки блока Gain

8. Блок Matrix Gain – отличается от блока Gain только тем, что в нем коэффициент передачи задается как матрица (многомерной системы);

9. Блок Slider Gain – предназначен для изменения коэффициента усиления. Данный блок является одним из элементов взаимодействия пользователя с моделью в процессе моделирования. В активном состоянии блок представляет собой диалоговое окно, обеспечивающее установку значения некоторого параметра модели с помощью «ползункового» регулятора (рис. 1.17).

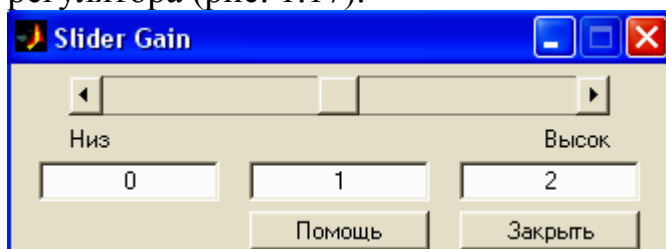


Рис. 1.17. Параметры настройки блока Slider Gain

Блок Slider Gain становится активным после того, как будет помещен в окно блок-диаграммы создаваемой модели. Чтобы открыть окно с

регулятором, необходимо дважды щелкнуть на изображении блока левой кнопкой мыши. Окно Slider Gain имеет три поля ввода:

- для указания нижнего уровня параметра (Low),
- верхнего уровня (High);
- текущего значения.

Текущее значение должно лежать внутри диапазона [Low, High]. Однако при выборе нового диапазона необходимо сначала указать новое значение параметра, а затем изменить границы диапазона.

10. Блок Math Function позволяет использовать для преобразования входного сигнала элементарные нетригонометрические функции (вычисление экспоненты, натурального и десятичного логарифмов, возведение в степень, извлечение квадратного корня: $\exp$, $\log$, 10^u , $\log_{10}$, magnitude^2 , square , sqrt , pow , conj , reciprocal , hypot , rem , mod , transpose , hermitian). Нужная функция выбирается с помощью «выпадающего» меню;

11. Блок Rounding Function содержит различные функции округления значения амплитуды входного сигнала; выбор конкретного метода округления выполняется также с помощью «выпадающего» меню;

12. Блок Trigonometric Function обеспечивает преобразование входного сигнала с помощью одной из тригонометрических функций ($\sin$, $\cos$, $\tan$, asin , acos , atan , atan2 , $\sinh$, $\cosh$, $\tanh$, asinh , acosh , atanh); выбор требуемой функции производится в окне настройки параметров блока с помощью «выпадающего» меню (оно становится доступным только после перемещения блока в поле блок-диаграммы);

13. Блок Algebraic Constraint – используется для нахождения решения алгебраического уравнения;

14. Блок Polynomial – служит для вычисления корней полинома.

Вторая группа, Vector Operations, содержит блоки, осуществляющие векторные операции. Она включает следующие блоки:

1. Блок Assignment – используется для включения одного сигнала в другой векторный сигнал;

2. Блок Matrix Concatenation – позволяет выполнить расширение матричного сигнала за счет его сцепления с другим матричным сигналом;

3. Блок Reshape – используется для преобразования входного векторного или матричного сигнала в одну из форм одномерного массива.

В состав третьей группы, Logic Operations, входят блоки, осуществляющие логические операции. Во всех блоках этой группы выходная величина является булевой, то есть может принимать лишь два значения: «1» (Истина) или «0» (Ложь). Во многих блоках этой группы булевыми должны быть и все входные величины.

1. Блок Logical Operator содержит набор основных логических операций: AND (операция логического умножения «И»), OR (логическое сложение «ИЛИ»), NAND («И-НЕ»), NOR («ИЛИ-НЕ»), XOR (сложение по модулю 2), NOT (операция логического отрицания). Выбор требуемой функции выполняется с помощью «выпадающего» меню; имя функции отображается на иконке блока.

Другим параметром настройки блока является число аргументов логической операции, то есть число входных портов блока (Number of input ports). Его значение вводится в строке редактирования и должно быть натуральным числом. Максимально допустимое число входов блока практически не ограничено;

2. Блок Relational Operator реализует операции отношения над двумя входными сигналами: $>$, $<$, $\leq$, $\geq$, $=$ (тождественно), $\neq$ (не равно). Конкретная операция выбирается при настройке параметров блока посредством «выпадающего» меню. Знак операции выводится на изображении блока.

3. Блок Combinatorial Logic обеспечивает преобразование входного сигнала в соответствии с заданной таблицей истинности. По своим функциональным возможностям это очень мощный блок. С его помощью можно описать на уровне «вход-выход» логику работы любого устройства или системы. Единственное ограничение заключается в том, что входные данные и результат должны быть представимы в форме булевых величин.

На вход блока может подаваться как скалярный, так и векторный сигнал. Скаляр в данном случае интерпретируется как одна булева величина, а вектор – как их совокупность. При этом любое ненулевое значение входного сигнала соответствует значению «истина».

Блок Combinatorial Logic имеет единственный параметр настройки – Truth table (таблица истинности), который представляет собой список возможных выходных значений блока. При задании таблицы истинности необходимо соблюдать два основных правила:

- 1) число строк таблицы должно быть равно 2^n , где n – число элементов (размерность) входного сигнала;
- 2) входы таблицы считаются заданными.

Например, если на вход блока Combinatorial Logic подается векторный сигнал с $n = 2$, то параметр Truth table представляет собой список из четырех элементов, например такой: [0; 1; 0; 1]. Таким образом, параметр Truth table описывает только значения выходного сигнала.

Разрядность выходного сигнала, а также значение каждого его разряда выбираются пользователем на основе собственных представлений о логике работы создаваемой модели. Сформированный выходной сигнал может быть или выведен в какое-либо «смотровое окно», или передан на другие блоки модели.

4. Блок Bitwise Logical Operator – является поразрядным логическим оператором.

Четвертая группа Complex Vector Conversions состоит из следующих блоков:

1. Блоки Complex to Magnitude-Angle и Complex to Real-Imag осуществляют преобразование комплексного входного сигнала в один или два действительных выходных сигнала, которые являются модулем, аргументом, действительной или мнимой частью входного сигнала.

2. Блоки Magnitude-Angle и Complex и Real-Imag to Complex преобразуют один или два выходных действительных сигнала в комплексный выходной сигнал. Количество входов или выходов определяется в окне настройки блока.

1.3.6. Раздел Model Verification

Раздел Model Verification (Проверка модели) содержит блоки, выполняющие проверку некоторых статических и динамических характеристик модели (рис. 1.18).

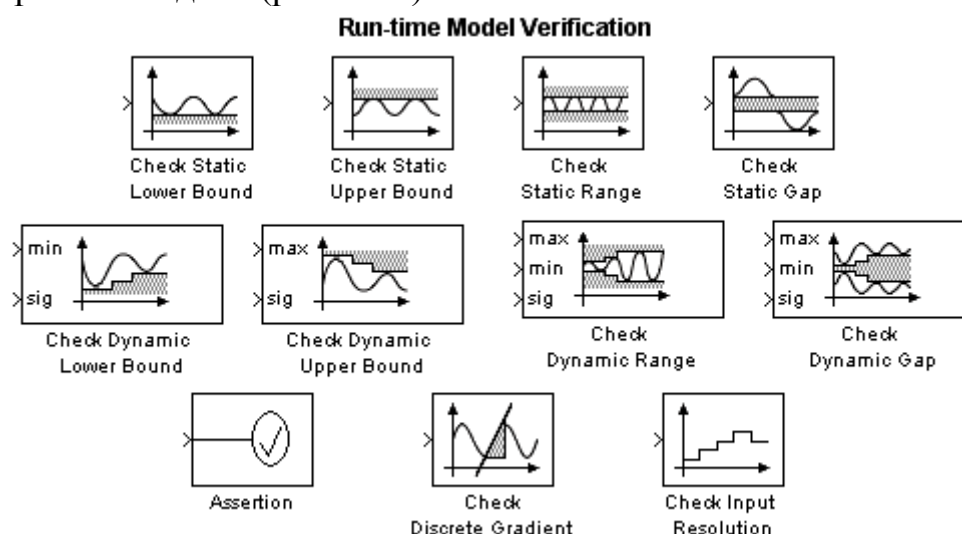


Рис. 1.18. Внешний вид блоков раздела Model Verification

1.3.7. Раздел Model-Wide Utilities

В раздел Model-Wide Utilities (Утилиты расширения модели) сосредоточены блоки, позволяющие линеаризовать динамическую модель и оформить на нее документацию (рис. 1.19).

Linearization of Running Models

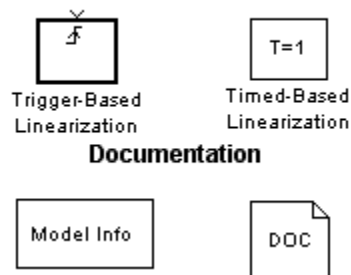


Рис. 1.19. Внешний вид блоков раздела Model-Wide Utilities

1.3.8. Раздел Ports & Subsystems

Большинство блоков раздела Ports & Subsystems (Порты и подсистемы) предназначены для разработки сложных S-моделей, содержащих модели более низкого уровня (подсистемы), и обеспечивают установление необходимых связей между несколькими S-моделями (рис. 1.20).

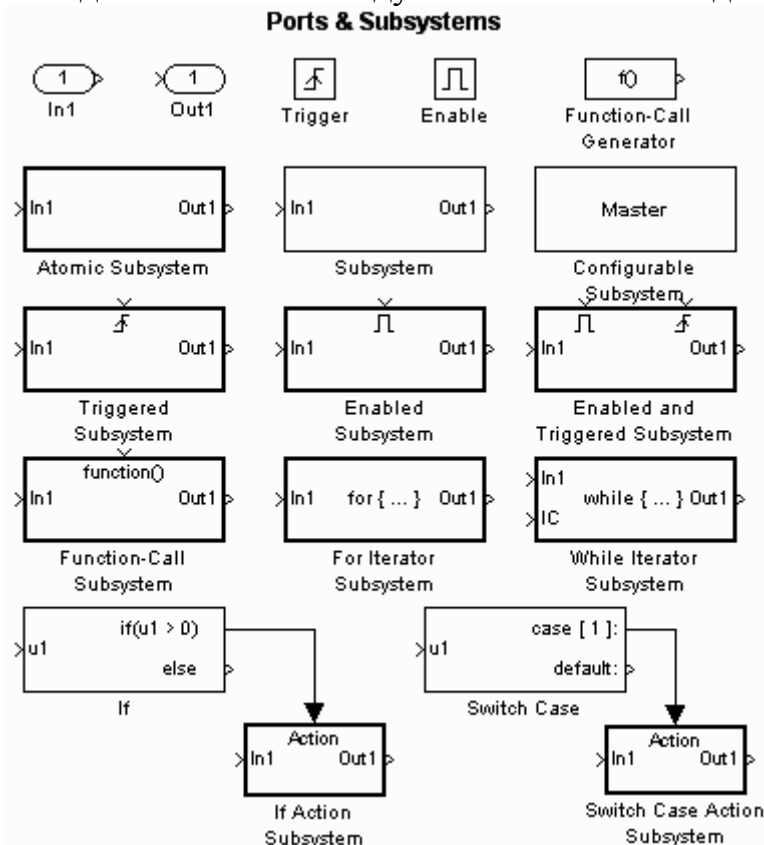


Рис. 1.20. Внешний вид блоков раздела Ports & Subsystems

Подсистемы могут функционировать лишь в составе основной S-модели, связь с которой осуществляется через входные (In) и выходные (Out) порты подсистемы.

Применение подсистем позволяет свести составление сложной S-модели к созданию совокупности вложенных простых подсистем более низкого уровня, что делает моделирование более наглядным и упрощает отладку модели.

1. Блоки In (Входной порт) и Out (Выходной порт) обеспечивают информационную «линейную» связь между подсистемой и вызывающей ее S-моделью.

2. Блоки Enable (Разрешить) и Trigger (Задвижка) предназначены для логического управления работой подсистем S-модели. Добавление того или другого блока в любую подсистему приводит к появлению на изображении блока соответствующего дополнительного управляющего входа. Блоки Enable и Trigger можно размещать только в подсистемах – попытка их переноса в окно основной системы приводит к появлению сообщения об ошибке. Место расположения блоков Enable и Trigger в подсистеме не имеет значения.

В окне параметров блока Enable можно задать следующие параметры:

- States when enable – состояние при запуске (выбор из списка: held – использовать последнее состояние; reset – использовать начальное состояние); Show output port – показать выходной порт;

- Enable zero crossing detections – фиксировать прохождение выходного сигнала через 0.

Управление подсистемой с блоком Trigger происходит, если управляющий сигнал меняет полярность. Работа такой подсистемы имеет ряд отличительных признаков:

- подсистема работает только на том шаге, на котором произошло изменение полярности управляющего сигнала;
- подсистема не возвращается в исходное состояние, и ее текущее состояние сохраняется до очередного запуска;
- отсутствует параметр Output when disabled у блока выхода;
- возможны различные виды обозначения входного порта управляющего сигнала (рис. 1.21);
- в блоках подсистемы, имеющих параметр эталонного времени Sample time, следует задавать этот параметр равным –1.

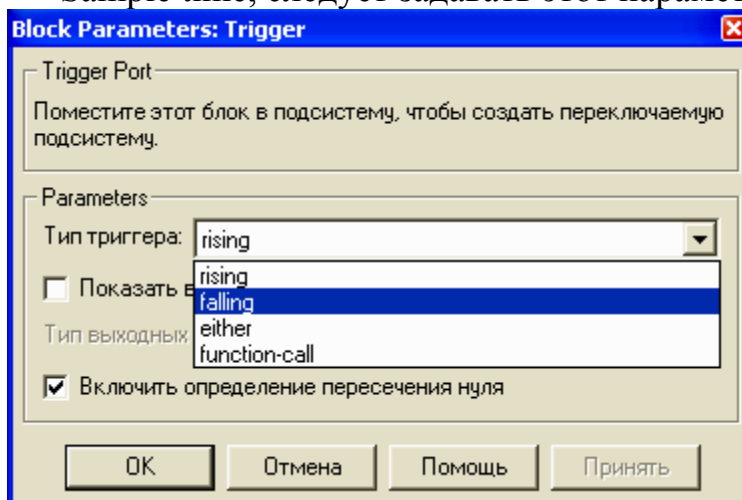


Рис. 1.21. Параметры блока Trigger

Сброс (reset), установка (held) (рис. 1.22) или запуск подсистемы (см. рис. 1.22) осуществляется под действием нарастающего (rising),

спадающего (falling) или обоих (either) фронтов внешнего сигнала управления или от внешней функции (function-call), которые подаются на соответствующий дополнительный вход подсистем с блоками Enable или Trigger.

3. Блок Subsystem (Подсистема) представляет собой «заготовку» для создания подсистемы. Подсистема – это достаточно самостоятельная S-модель более низкого уровня, которая, в свою очередь, может содержать подсистемы произвольного уровня вложенности.

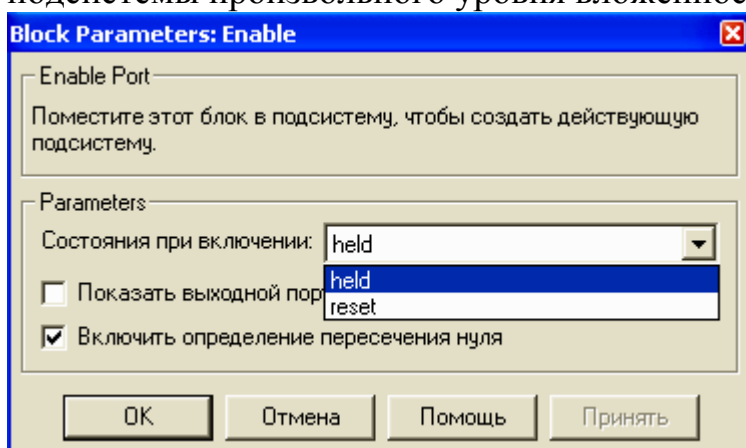


Рис. 1.22. Параметры блока Enable

Двойной щелчок на изображении блока Subsystem в блок-схеме приводит к появлению на экране окна, в котором размещена блок-схема, состоящая лишь из одного входного порта, соединенного с одним выходным портом (рис. 1.23, Subsystem1). Это служит напоминанием о том, что создаваемая пользователем подсистема должна обязательно содержать соединенные между собой (возможно через другие блоки) входные и выходные порты.

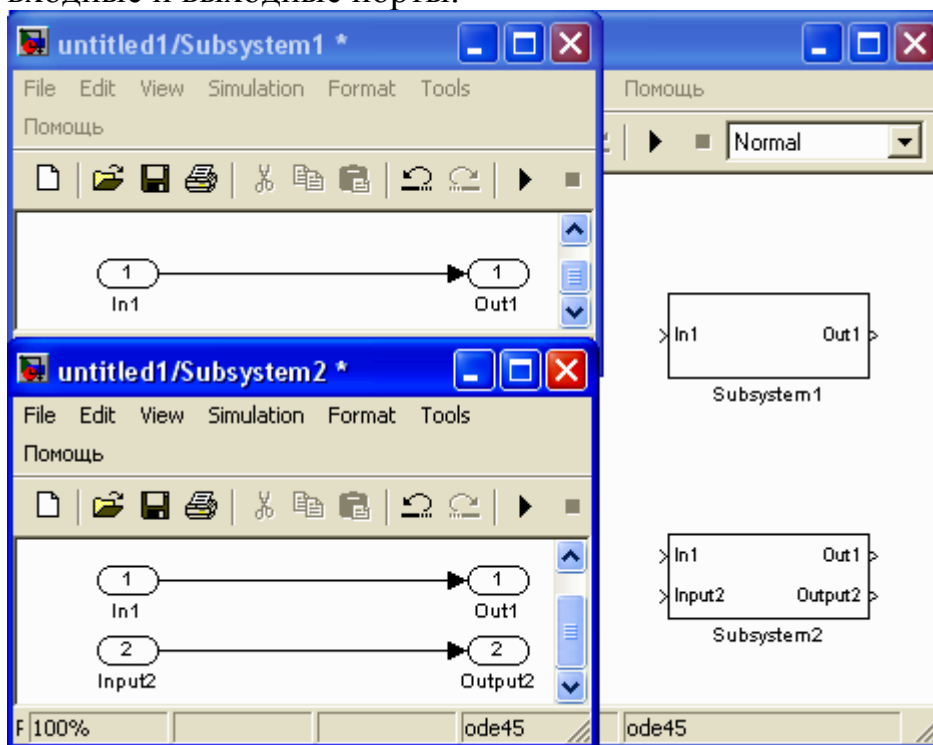


Рис. 1.23. Окна заготовки блоков Subsystem

Если число входных/выходных портов подсистемы больше одного, то соответствующим образом изменяется изображение блока Subsystem (см. рис. 1.23 Subsystem2). Переименовать любой порт можно, щелкнув по его названию один раз правой кнопкой мыши в окне подсистемы, имя порта при этом должно оставаться уникальным.

Остальные блоки, представленные на рис. 1.20, отличаются от блока Subsystem наличием уже встроенных блоков Enable и Trigger или блоков условий Function (Функция), For Iterator (Итерация), While Iterator (До тех пор, пока Итерация), If (Условие), Switch Case (До тех пор пока не выполнится условие).

4. Блок For Iterator Subsystem запускает блоки в подсистеме на заданное количество итераций. Число итераций может быть указано либо через диалог блока, либо через внешний входной порт. Если выходной порт показан, он будет выводить текущий номер итерации, начиная с единицы. Когда итерации запущены, любые состояния в подсистеме могут либо быть сброшены на начальные значения, либо оставлены в своем предыдущем значении.

5. Блок While Iterator Subsystem – выполнять блоки в этой подсистеме до тех пор, пока условие итератора While истинно и не достигнуто максимальное число итераций. Если блок находится в режиме do-while, все блоки в подсистеме будут выполнены хотя бы раз перед проверкой условия итератора While, иначе к порту IC должен быть подключен внешний сигнал для проверки, должен ли выполняться блок при первой итерации. Если выходной порт показан, он будет выводить текущий номер итерации, начиная с единицы.

6. Параметры и внешний вид блока If представлен на рис. 1.24, структурная схема блока If – на рис. 1.25.

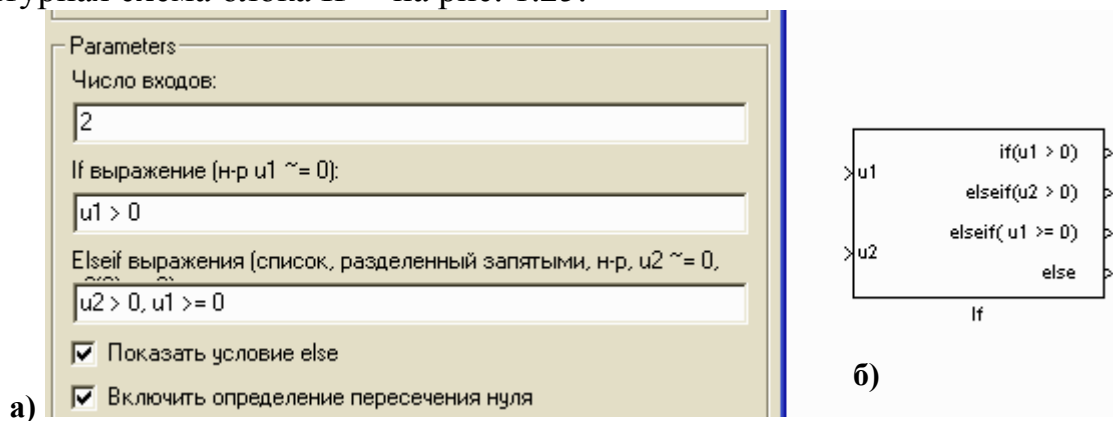


Рис. 1.24. Параметры (а) и внешний вид (б) блока If

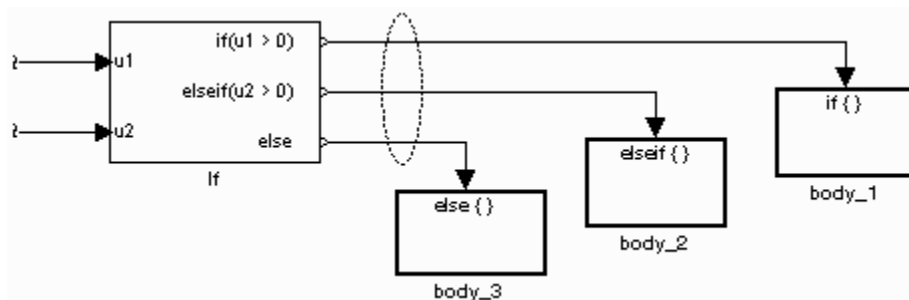


Рис. 1.25. Структурная схема, поясняющая работу блока If

Выражение If – запускает Подсистему действия, присоединенную к первому выходному порту (u1, выполняются действия body_1, см. рис. 1.25).

Выражение Elseif – запускает Подсистему действия, присоединенную ко второму выходному порту (u2). Количество выходных портов Elseif в блоке If равно числу разделенных запятыми Elseif выражений введенных в диалоге (см. рис. 1.24).

Выражение Else – запускает Подсистему действия, присоединенную к последнему выходному порту.

Работу блока If можно представить в виде следующего алгоритма:

```

if (u1 > 0) {
    body_1;
}
elseif (u2 > 0){
    body_2;
}
else {
    body_3;
}

```

If и Elseif выражения могут использовать следующие операторы MatLab: <, <=, ==, ~=, >, >=, ~ и т.п. для входных сигналов порта u1, u2, u3 и др.

7. Блок Switch Case имеет единственный вход, который проверяется на условия Case (рис. 1.26). Вход должен быть скаляром, в зависимости от этого сигнала подключается тот или иной выход, и совершаются определенные действия (body_1, body_23).

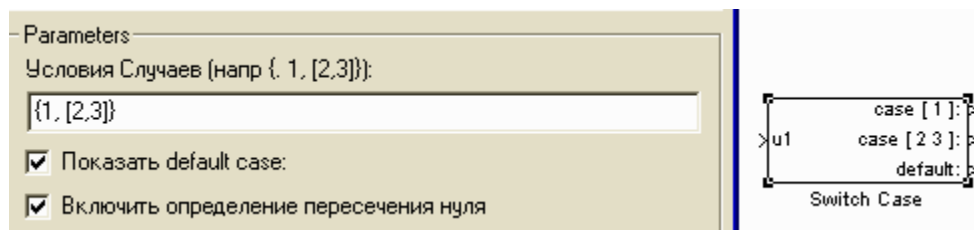


Рис. 1.26. Параметры и внешний вид блока Switch Case

Например, если в параметрах блока Switch Case в качестве условия введены числа {1, [2,3]}, это обозначает:

- если сигнал на входе $u1 = 1$, то выполняются действия, подключенные к первому выходу (case [1]);
- если $u1 = 2$ или 3 , то выполняются действия, подключенные ко второму выходу (case [2 3]);
- в противном случае выводится значение по умолчанию (default).

Работу блока Switch Case можно представить в виде следующего алгоритма (рис. 1.27):

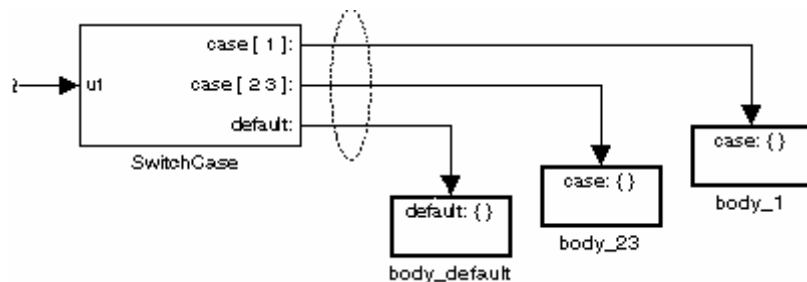


Рис. 1.27. Структурная схема, поясняющая работу блока SwitchCase

```
switch (u1) {
  case [u1=1]:
    body_1;
    break;
  case [u1=2 or u1=3]:
    body_23;
    break;
  default:
    bodydefault;
}
```

1.3.9. Раздел Signal Attributes

Раздел Signal Attributes (Атрибуты сигналов) (рис. 1.28) состоит из блоков, обеспечивающих определение либо изменение некоторых атрибутов сигнала, таких как размер (количество элементов в векторном или матричном сигнале), тип данных, начальные условия, скорость передачи данных.

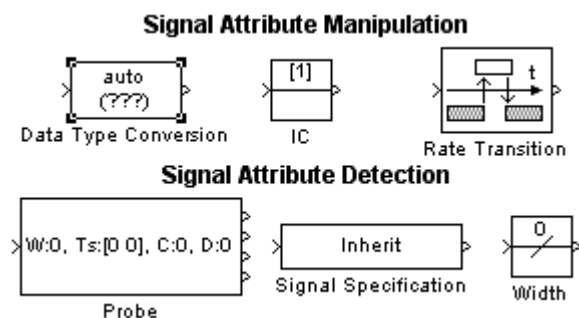


Рис. 1.28. Внешний вид блоков раздела Signal Attributes

1. Блок Data Type Conversion предназначен для изменения типа данных входного сигнала. Данный блок имеет единственный настраиваемый параметр Data Type, который позволяет задать тип данных:

- auto – автоматическая установка типа данных;
- double – число 8 байт с плавающей точкой двойной точности;
- single – число 4 байта с плавающей точкой одинарной точности;
- int8 – целое 1 байтовое число;
- uint8 – целое 1 байтовое число только для хранения и считывания;
- int16 – целое 2 байтовое число;
- uint16 – целое 2 байтовое число только для хранения и считывания;
- int32 – целое 4 байтовое число;
- uint32 – целое 4 байтовое число только для хранения и считывания;
- boolean – булев тип данных.

Входной сигнал может быть реальным или комплексным сигналом, выходной сигнал при этом будет реальным или комплексным числом соответственно.

Опция Saturate on integer overflow (Насыщать при переполнении целого) доступна только для входного сигнала целочисленного типа.

2. Блок IC (Initial Condition – начальное условие) позволяет установить произвольное начальное значение входного сигнала. Длительность пребывания системы в этом состоянии равна длительности шага моделирования. Параметром настройки блока является величина Initial Value (Значение сигнала на момент входа в блок). Данный блок применяется, например, для внешней установки начального условия перед блоком Integrator (см. п. 1.3.1).

3. Блок Rate Transition (Согласование дискретных значений) – служит для согласования дискретных блоков, имеющих разный период квантования. Окно его параметров представлено на рис. 1.29.

В окне настроек можно установить опции гарантированной целостности данных и детерминированной передачи их.

Наиболее важная установка Data rate transition (Режим передачи данных), имеет следующие значения:

- Slow to fast – передача от медленного блока к быстрому;
- Fast to slow – передача от быстрого блока к медленному.

Среди параметров блока Rate Transition есть также опция задания начального значения Initial condition (см. рис. 1.29).

4. Блок Probe (Зонд) служит для определения следующих атрибутов сигнала:

- Width – ширина сигнала, проверка длительности сигнала;

- Sample time – величина дискрета времени, проверка эталонного времени;
- Complex signal – наличие комплексных сигналов (результат проверки равен «1», если сигнал представлен в комплексном виде, и «0» – в противном случае);
- Signal dimensions – проверка размерности сигнала.

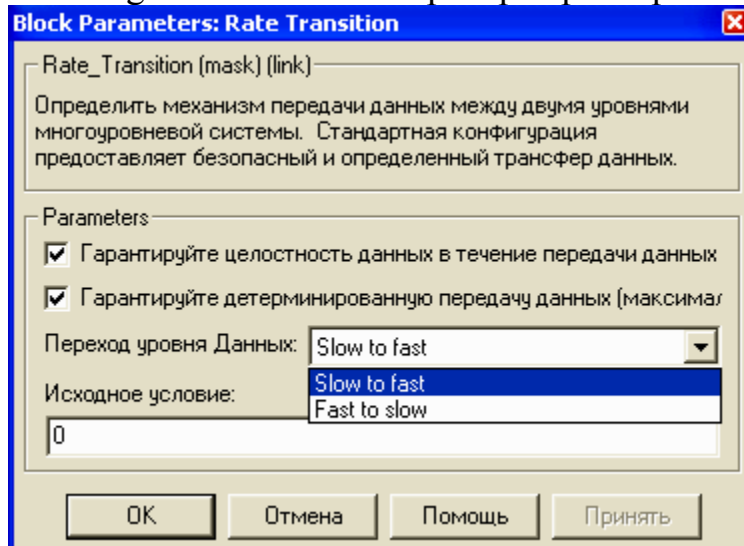


Рис. 1.29. Окно параметров блока Rate Transition

Контролируются те параметры, для которых установлены флажки. Числом отмеченных флажков задается число выходов блока.

5. Блок Signal Specification (Спецификация сигналов) служит для распознавания сигнала на его входе и выдачи спецификации сигнала (рис. 1.30).

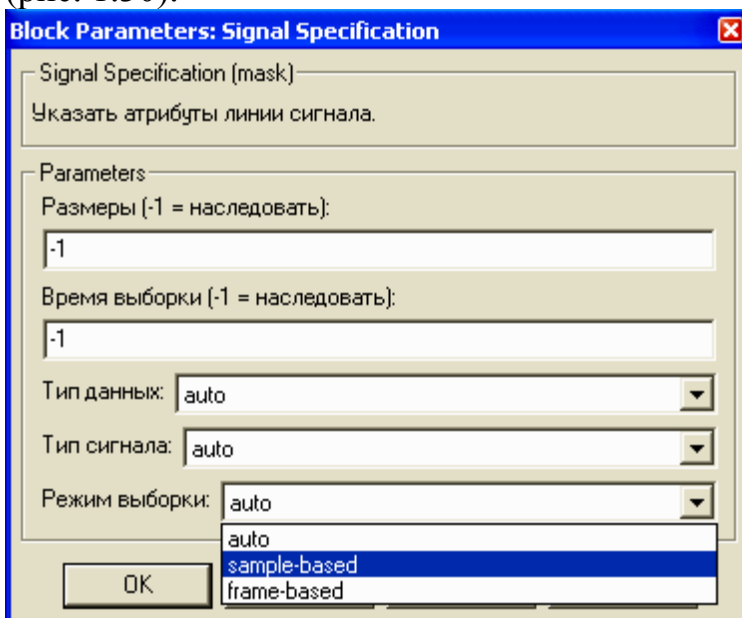


Рис. 1.30. Окно параметров блока Signal Specification

В окне параметров блока устанавливается (см. рис. 1.30):

- Dimension – размерность сигнала (в виде вектора длиной n , матрицы размерностью $[m \ n]$ или «-1» для любой размерности);
- Sample time – эталонное время;

- Data type – тип данных (auto, double, single, int8, uint8, int16, uint16, int32, uint32, boolean, specify via dialog) (см. описание типов данных выше при описании блока Data Type Conversion, п. 1.3.9);

- Signal type – тип входного сигнала (auto – любой числовой тип, real – реальное число, complex – комплексное число);

- Simpling mode – режим выборки (auto – любой режим, sample-based, frame-based).

6. Блок Width (Размер) определяет длину векторного сигнала, который поступает на его вход. Значение размерности выводятся непосредственно на изображении блока. Настраиваемых параметров блок не имеет (рис. 1.31).

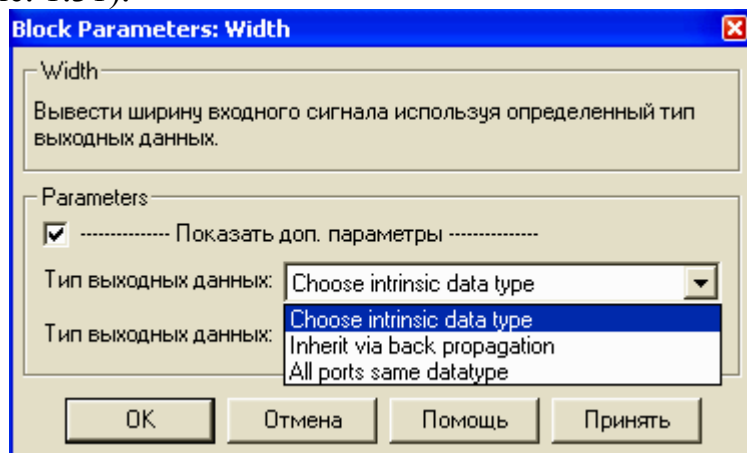


Рис. 1.31. Окно параметров блока Width

В дополнительных параметрах можно изменить тип выходных данных:

- Choose intrinsic data type – выбрать внутренний тип данных. Становится доступным вторая вкладка «Тип выходных данных», где можно выбрать любой тип (double, single, int8, uint8, int16, uint16, int32, uint32) (см. описание типов данных выше при описании блока Data Type Conversion, п. 1.3.9);

- Inherit via back propagation – тип данных наследуется по обратной связи;

- Same datatype – тип данных не изменяется.

1.3.10. Раздел Signal Routing

В раздел (Пересылка сигналов) включены блоки, обеспечивающие разного рода пересылки сигналов, такие как переключение сигналов, объединение нескольких сигналов в шину, «разводку» сигналов из шины и т.п. состав этого раздела приведен на рис. 1.32.

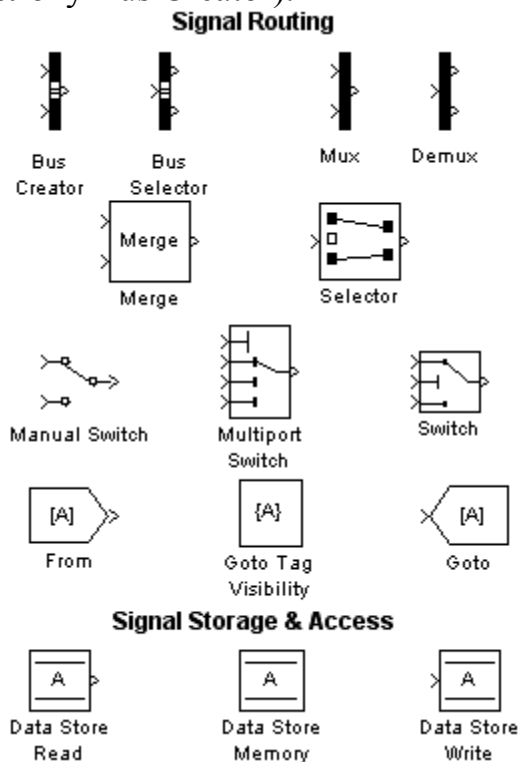
Библиотека содержит два раздела: Signal Routing и Signal Storage & Access.

В раздел Signal Routing входят следующие блоки:

1. Bus Creator (Создание шины) – служит для объединения двух и более сигналов самого различного вида в одну шину, для удобства графического

моделирования. Используется, например, для отображения в одном окне осциллографа (в одних осях координат) нескольких различных сигналов.

2. Bus Selector (Селекция сигналов шины) – обеспечивает выбор заданных сигналов из нескольких (выполняет функции противоположные блоку Bus Creator).



**Рис. 1.32. Внешний вид
блоков раздела Signal
Routing**

Окно параметров блока Bus Selector представлено на рис. 1.33.

Окно состоит из двух частей: в левой имеется список всех входных сигналов, а в правой – выбранных (селектированных) сигналов, которые будут выдаваться на выход. В данном случае выбран только первый сигнал, но можно выбрать все. Для выбора того или иного сигнала следует выделить его в левом окне (рис. 1.31) и нажать кнопку «Select >>» (Выбрать >>). Флажок Mixed output (Объединенный выход) позволяет наблюдать описание сигналов на выходе селектора. Кнопки Up (Вверх) и Down (Вниз) позволяют переносить вверх/вниз выделенный в правой части окна сигнал и, таким образом, менять порядок сигналов на выходе блока Bus Selector. Кнопка Remove удаляет выделенный сигнал из окна выбранных, а кнопка Refresh (Обновить) очищает окно выбранных сигналов.

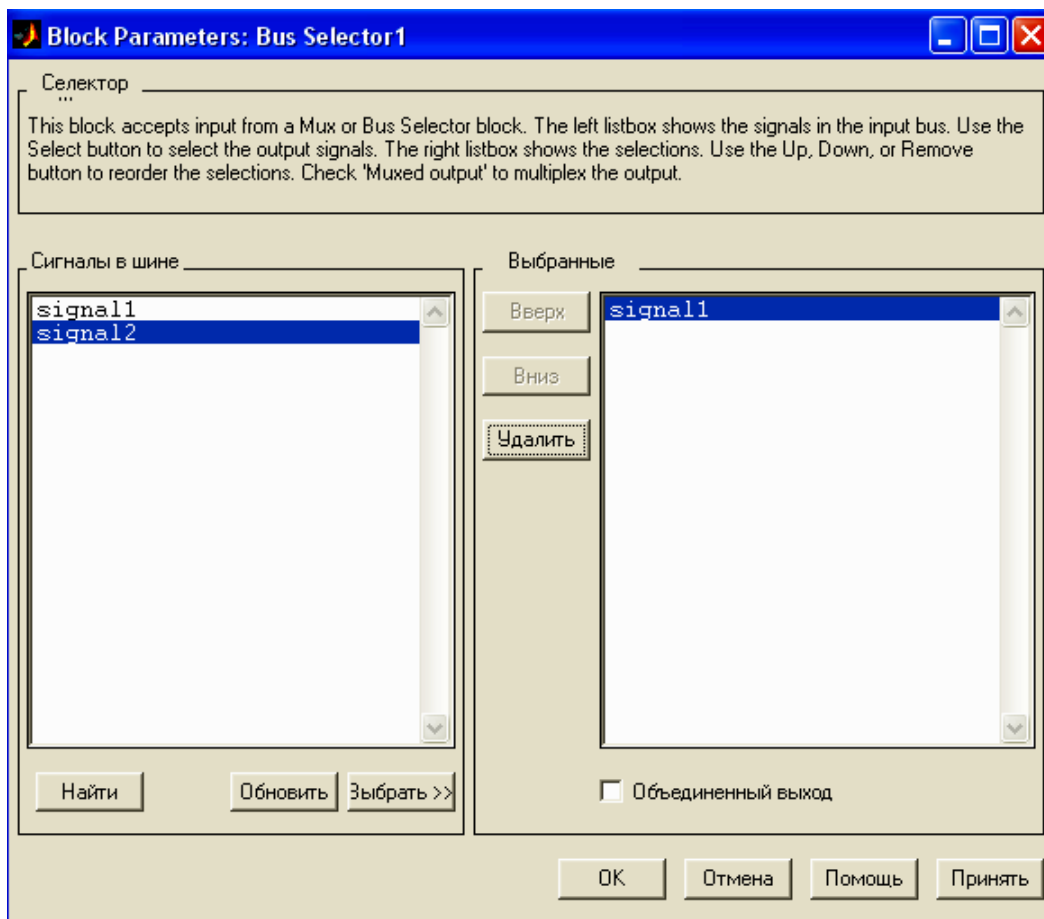


Рис. 1.33. Окно установки параметров блока Bus Selector

3. Mux (Мультиплексор) – выполняет объединение мультиплексирование входных величин в единый выходной вектор (шину). Входные величины могут быть как скалярными, так и векторными. Длина результирующего вектора равна сумме длин всех векторов. Порядок элементов в векторе выхода определяется порядком входов (сверху вниз) и порядком расположения элементов внутри каждого входа.

Блок Mux имеет два настраиваемых параметра: Number of inputs (Количество входов) и Display option (Вид изображения). Последний определяет, в каком виде блок будет изображен на блок-схеме: none (рис. 1.34, а), signals (см. рис. 1.34, б) или bar (см. рис. 1.34, в)

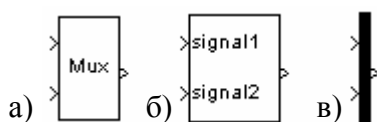


Рис. 1.34. Внешний вид блоков раздела Signal Routing

4. Dmux (Демультимплексор) – выполняет функцию обратную блоку Mux, т.е. разделяет входной вектор на заданное число компонентов.

Блок Dmux имеет два настраиваемых параметра: Number of outputs (Количество выходов) и Display option (Вид изображения). Если заданное

число выходов (N) меньше, чем размер входного вектора (M), блок формирует исходные векторы следующим образом. Первые $N-1$ выходов будут векторами с одинаковым количеством элементов, равным целой части отношения $M/(N-1)$. Последний выход будет иметь размер, равный остатку от деления. Если размер входного вектора меньше, чем число выходов ($M < N$), то при запуске модели выдается сообщение об ошибке («несоответствие размерности портов»).

5. Manual Switch – ручной переключатель сигналов. Данный блок не имеет настраиваемых параметров. У него два входа и один выход. На изображении блока с помощью переключки показано, какой именно из двух входов подключен к выходу. Блок позволяет вручную переключать входы. Для этого необходимо дважды щелкнуть левой кнопкой мыши на изображении блока (предварительно, конечно, блок должен быть помещен в поле блок-диаграммы разрабатываемой модели). В результате изменится и изображение блока – на нем выход уже будет соединен переключкой с другим входом. Переключение возможно как в процессе моделирования, так и при его остановке.

6. Multiport Switch – многопортовый (многовходовый) переключатель сигналов, обеспечивает передачу на выход сигнала, поступающего на один из информационных входов. Данный блок имеет не меньше трех входов. Первый из них (верхний) является управляющим, остальные – информационными.

Окно настройки блока содержит один параметр: Number of inputs (Количество входов) – число информационных входов. Значение параметра может вводиться в форме числовой константы или в форме вычисляемого выражения.

Номер входа, который соединяется с выходом, соответствует значению управляющего сигнала, поступающего на верхний вход. Если это значение является дробным числом, то оно округляется до целого по обычным арифметическим правилам, если это значение меньше единицы, оно считается равным единице. Когда значение управляющего сигнала больше, чем число информационных входов, то оно принимается равным наибольшему номеру (входы нумеруются сверху вниз, начиная со входа, который находится под управляющим).

Переключение возможно в динамике процесса моделирования, что позволяет формировать сложные последовательности сигналов.

7. Merge (Объединение/слияние сигналов) – служит для объединения двух и более векторных сигналов, поступающих на входы, в один векторный сигнал. Пример применения блока Merge (рис. 1.35, а) и его основные параметры представлены на рис. 1.35, б.

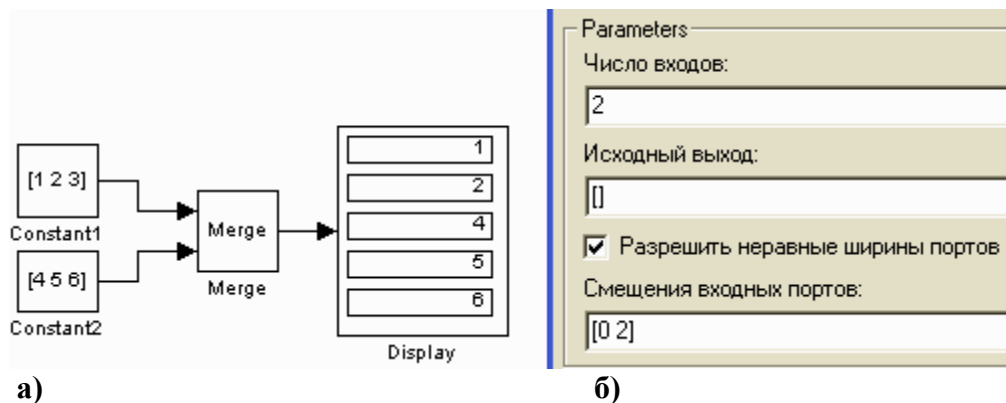


Рис. 1.35. Применения блока Merge для объединения двух векторов (а), параметры блока Merge (б)

В окне установки параметров блока Merge можно задать:

- число входов Number of inputs. Для корректной работы блока все входные сигналы должны иметь одинаковую размерность (хотя при несоблюдении этого правила сообщение об ошибке не выдается);
- начальное значение выходного сигнала Initial output – задает сигнал, на основе которого должно выполняться объединение (если параметр не задан, возвращается сигнал, заданный последним);
- разрешение на различную размерность входных портов Allow unequal port width;
- смещение входного сигнала Input Port offset – можно задавать расположение входного сигнала в выходном векторе.

8. Switch – управляемый переключатель сигналов (рис. 1.36).

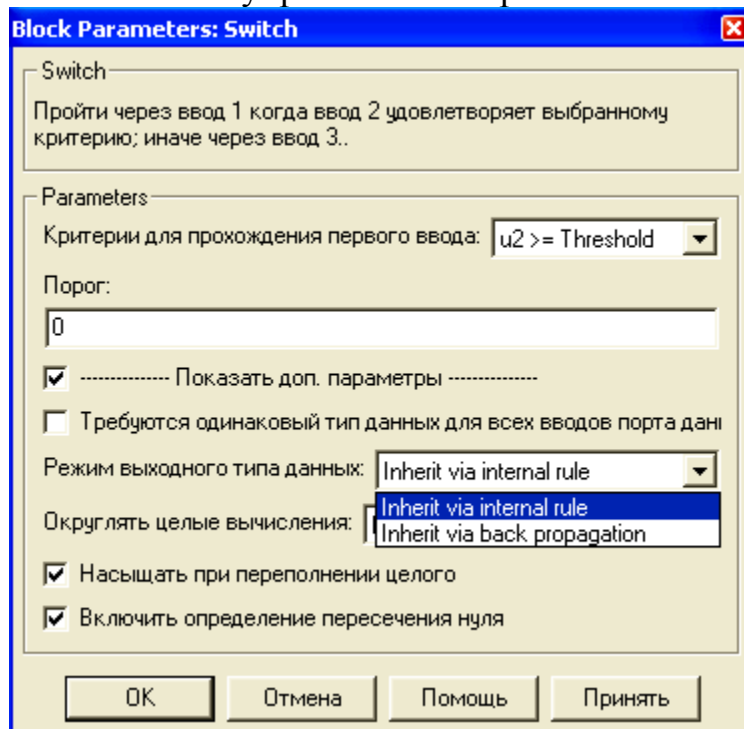


Рис. 1.36. Окно установки параметров блока Switch

Блок Switch имеет три входа и один выход. Первый и третий входы являются информационными, а второй – управляющим. Если величина (амплитуда) управляющего сигнала, поступающего на второй вход, не меньше некоторого граничного значения (больше или равно $\geq$, строго больше $>$, примерно равно нулю ≈ 0), заданного в поле Threshold (Порог), то на выход блока передается сигнал с первого входа, в противном случае – передается сигнал с третьего входа.

Блок Switch имеет один настраиваемый параметр – Threshold (Порог) – он может задаваться либо как числовая константа, либо как вычисляемое выражение.

Опция Show additional parameter (Показать дополнительные параметры) расширяет окно параметров и позволяет выполнить (см. рис. 1.36) следующие действия:

- установку идентичности всех входных данных;
- установку типа выходных данных (Inherit via internal rule – наследование по внутреннему правилу, т.е. выходной сигнал имеет тот же тип данных, что и входной, с большим положительным диапазоном; Inherit via back propagation – наследовать по обратной связи);
- способ округления чисел в данных (Zero, Nearest, Ceiling, Floor);
- корректность ограничения целочисленных данных;
- фиксацию прохождения сигналов через нуль.

Периодичность срабатывания блока Switch определяется значением параметра Sample time блока, подсоединенного к его управляющему входу. Необходимо иметь в виду, что при изменении направления передачи сигнала положение «перемычки» на иконке блока не изменяется.

9. Selector (Селектор) – выбирает во входном векторе (Vector) или матрице (Matrix) и передает на выход только те элементы, номера которых указаны в поле ввода Elements (Элементы) окна настройки блока (рис. 1.37). Изображение блока зависит от того, какой выбор элементов задан. Выбор сигнала может быть внутренним (Internal) или внешним (External). В последнем случае на изображении блока появляются выводы для управляющих сигналов.

Блок Selector имеет два параметра настройки:

- Elements (Элементы) – список номеров элементов входного вектора, подлежащих отбору;
- Input port width (Размерность входного порта) — размерность входного вектора, который может быть «просеян» с помощью данного блока; значение этого параметра обязательно должно совпадать с числом элементов входного вектора.

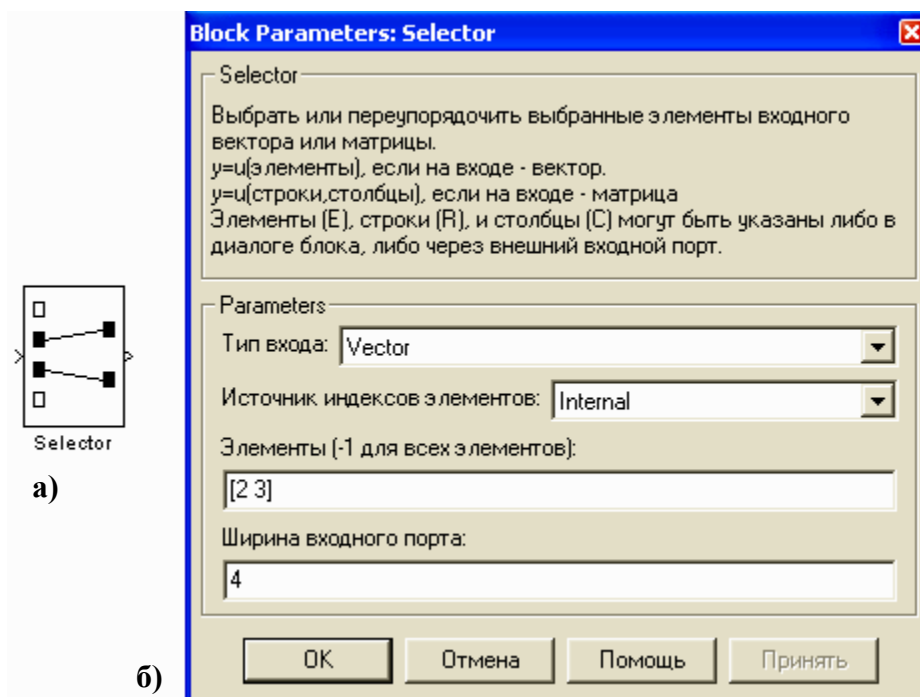


Рис. 1.37.
Внешний вид
блока Selector и
окно установки
его параметров

Если на вход блока подается матрица, то в списке Elements используется сквозная нумерация элементов, причем индекс изменяется построчно. Например, если на вход блока поступает матрица размером 3 x 4, то ее элементы будут иметь номера с 1-го по 12-й, причем номера 1 – 3 будут принадлежать элементам первой строки, номера 4 – 6 – элементам 2-й строки и т.д. Существенным достоинством блока является то, что значения его параметров настройки отображаются в графическом виде на иконке блока.

10. From – (Принять от) прием сигналов от блока Goto (см. описание далее по тексту).

11. Goto Tag Visibility (Признак видимости) – блок признака видимости сигнала;

12. Goto (Передать в) – организация «беспроводного» передатчика данных.

Блоки «беспроводной» связи From, Goto Tag Visibility и Goto используются совместно и предназначены для обмена данными между разнообразными частями S-модели с учетом досягаемости (видимости) этих данных (рис. 1.38). Они применяются для упрощения построения моделей.

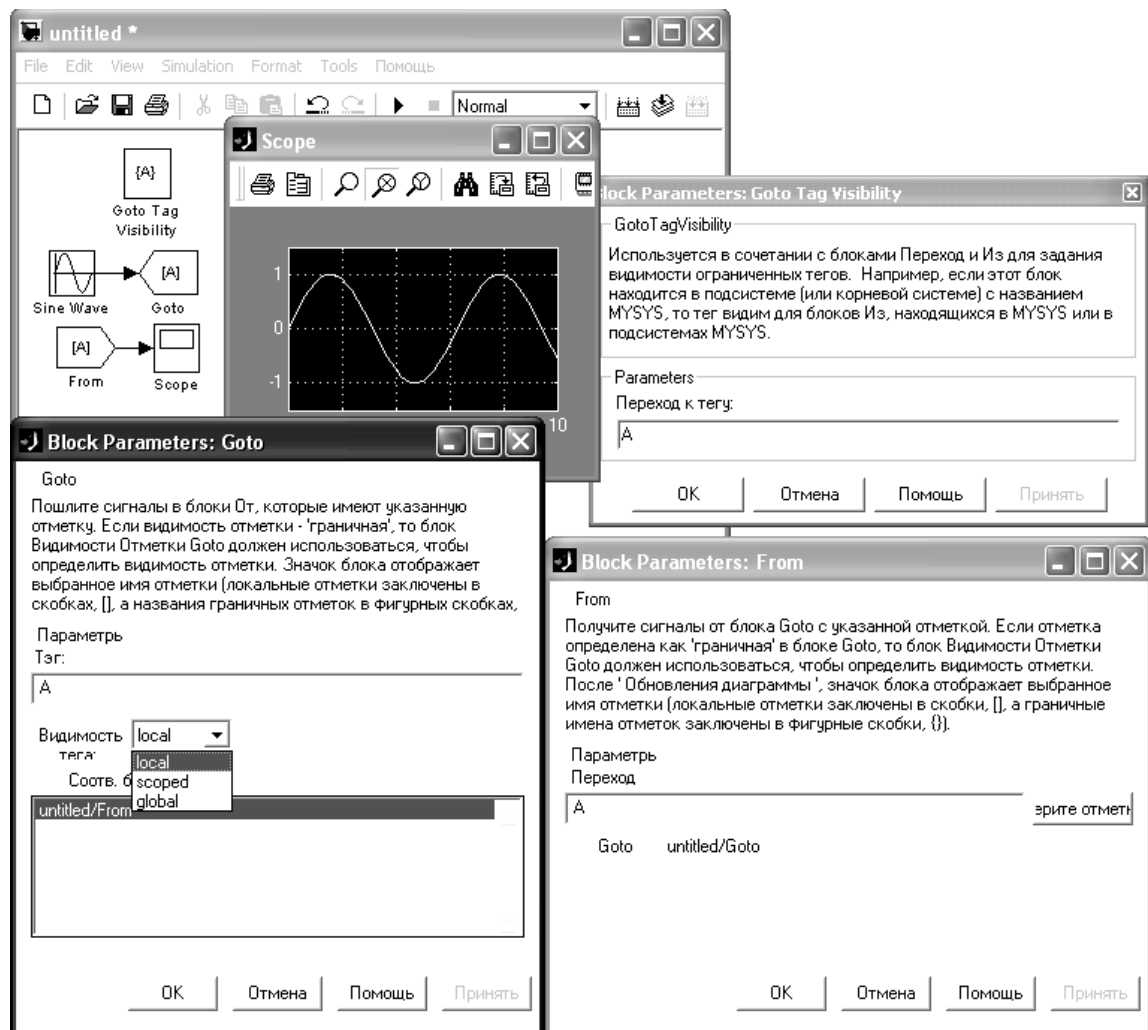


Рис. 1.38. Применение блоков «беспроводной» связи

С помощью блока Goto можно создать «беспроводной» передатчик данных, совместимых с этим блоком и имеющих имя (обычно в виде буквы). Один или ряд блоков From с таким же именем могут принимать данные от блока Goto, хотя прямых соединений между ними и блоком Goto нет.

Окно блока помимо параметра Tag (имени сигнала) имеет параметр Tag Visibility (Признак видимости), который может принимать одно из трех значений:

- local – передача сигнала в пределах данной подсистемы, где расположен блок Goto (имя сигнала задается в квадратных скобках, например [A]);
- scoped – передача сигнала в пределах данной подсистемы и подсистем более низкого уровня (имя сигнала задается в фигурных скобках, например {A});
- global – передача сигнала в пределах всей модели (имя сигнала задается непосредственно, например A).

В окне параметров блока From устанавливается единственный параметр – имя сигнала. Для этого служит список, управляемый кнопкой Select tag».

От одного блока Goto может работать несколько блоков From. Правила видимости сигналов в блоках From те же, что и у блоков Goto. Блок Goto Tag Visibility служит для установки видимости в пределах модели или подсистемы, на которую распространяется область видимости. В окне его параметров устанавливается только имя сигнала.

В разделе Signal Storage & Access входят блоки для записи и считывания данных:

1. Data Store Memory (Запоминание данных) – запись данных с заданным именем в память. Данный блок по существу резервирует под данные память;

2. Data Store Read (Чтение данных) – считывание данных с заданным именем;

3. Data Store Write (Запись данных) – запись значений сигналов с заданным именем.

Все эти блоки используются совместно и обеспечивают не только передачу данных, но и их сохранение на протяжении моделирования.

В блоке Data Store Memory предусмотрена установка трех параметров:

- Data store name – задает имя сохраняемых данных (например, A для записи переменной A);

- Initial value – задает начальное значение (по умолчанию 0) и формат сохраняемых данных (к примеру, если этот параметр задан как [[1 2] [3 4]], то это означает, что он является матрицей размера 2 x 2);

- Interpret vector parameters as 1-D – опция, задающая интерпретацию вектора параметров данных как одномерного вектора.

При использовании блоков Data Store Write и Data Store Read обязательно следует учитывать тот формат данных, который задан параметрами блока Data Store Memory.

Блок Data Store Memory может обслуживать несколько блоков Data Store Write и Data Store Read. Его расположение в диаграмме модели задает область видимости сохраняемых данных (в частности переменных). Если блок Data Store Memory размещен в диаграмме верхнего уровня, то все ее данные будут «видимы», т.е. доступными для сохранения или считывания. Если же блок Data Store Memory расположен в подмодели, то доступными будут данные только этой подмодели.

1.3.11. Раздел Sinks

Блоки раздела Sinks (Приемники) позволяют визуализировать получаемые при моделировании результаты и проконтролировать правильность работы того или иного блока и моделируемой системы в целом. В

этом разделе размещены три группы блоков (рис. 1.39). Все они имеют только входы и не имеют выходов.

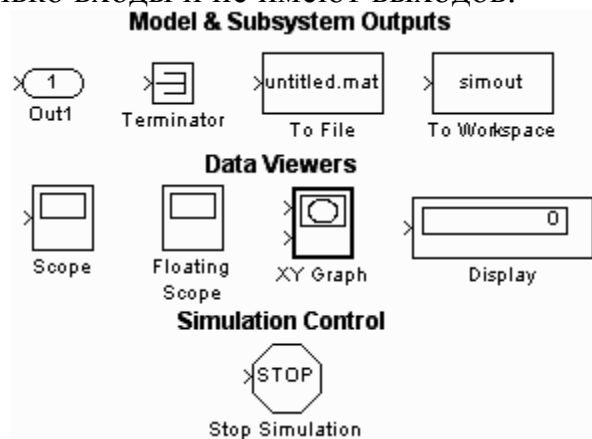


Рис. 1.39. Внешний вид блоков раздела Sinks

Группу Model & Subsystem Outputs составляют блоки, предназначенные для пересылки и сохранения результатов:

1. Out – выходной порт для вывода результатов вне модели, для организации связи с подсистемой. Напомним, что блок In служит для создания порта входа (см. рис.1.20, п. 1.3.8). При создании подсистемы блоки In и Out создаются автоматически, причем их число соответствует количеству входов и выходов подсистемы.

2. Terminator – порт для вывода результатов «в никуда». Если в модели встречаются отключенные выходы блоков, то может наступить ошибка моделирования. Для предотвращения таких ситуаций служит блок Terminator. Этот блок имеет только вход и информационное окно параметров. Данный блок можно использовать также с виртуальным осциллографом Floating Scope, где отсутствуют соединения между выводами из схемы и входами осциллографа.

3. To File – блок, обеспечивающий сохранение результатов моделирования на диске в МАТ-файле (с расширением .mat). Этот блок используется в паре с блоком From File из раздела Sources (см. п. 1.3.12).

Блок To File имеет четыре параметра настройки:

- а) File name – имя МАТ-файла, в который будут записываться данные (может быть указан полный путь доступа к файлу, по умолчанию – untitled.mat); имя файла выводится на изображении блока в блок-диаграмме;

- б) Variable name – имя переменной, по которому можно обращаться к данным, записанным в файле (для того, например, чтобы просмотреть или изменить их в командном окне MatLab); по умолчанию используется стандартное имя ans;

- в) Decimation (Прореживание) – дискретность записи данных в файл; при Decimation = 1 запись производится на каждом шаге моделирования; параметр может принимать только целочисленные значения;

г) Sample Time (Отчет времени) – величина шага моделирования для данного блока.

4. To Workspace – блок сохраняет данные, полученные в процессе моделирования, в рабочую область MatLab. Этот блок используется в паре с блоком From Workspace из раздела Sources (см. п. 1.3.12).

Данные сохраняются в виде матрицы, структура которой отличается от структуры данных в MAT-файле тем, что:

- значения сохраняемых величин расположены по строкам, а не по столбцам;
- не регистрируются значения модельного времени.

Блок To Workspace имеет 4 параметра настройки:

1. Variable name (Имя переменной) – имя, под которым данные сохраняются в рабочей области (по умолчанию – Simout);
2. Maximum number of rows (Максимальное количество строк. Только последние результаты обработки) – предельно допустимое число шагов моделирования, для которого регистрируются данные (по умолчанию задается константой «inf», то есть данные регистрируются на всем интервале моделирования);
3. Decimation (Прореживание) – дискретность регистрации данных;
4. Sample time (Отсчет времени) – величина шага (дискретность изменения) модельного времени.

Два последних параметра имеют тот же смысл, что и одноименные параметры блока To File.

В группу Data Viewers входят блоки, которые при моделировании играют роль обзорных окон. К ним относятся:

1. Scope – имитатор осциллографа – блок с одним входом, используется для вывода в графическое окно графика зависимости величины, подаваемой на его вход, от модельного времени;
2. Floating Scope – блок с одним входом, выполняющий аналогичные функции;
3. XY Graph – блок с двумя входами, который обеспечивает построение графика зависимости одной моделируемой величины (второй сверху вход) от другой (первый вход) $Y = f(X)$;
4. Display – блок с одним входом, предназначенный для отображения числовых значений входной величины.

К третьей группе относится блок управления моделированием – Stop Simulation, позволяющий прерывать моделирование при выполнении тех или иных условий. Блок срабатывает в том случае, когда на его вход поступает ненулевой сигнал.

1.3.11.1. Блок Scope

Блок Scope позволяет в ходе моделирования наблюдать графики входного сигнала в функции времени, динамику изменения процессов в модели. Если на вход прибора подается сигнал X_{BX} , то на экране представляется зависимость $X_{BX} = f(t)$. Создаваемое с помощью блока Scope «смотровое окно» напоминает экран измерительного прибора – высокоточного осциллографа с отцифрованной масштабной сеткой. Для настройки его параметров следует, после размещения изображения блока в окне блок-схемы, дважды щелкнуть на нем левой кнопкой мыши. В результате на экране появится окно Scope (рис. 1.40). Размер и пропорции окна можно изменять произвольно, используя курсор мыши.

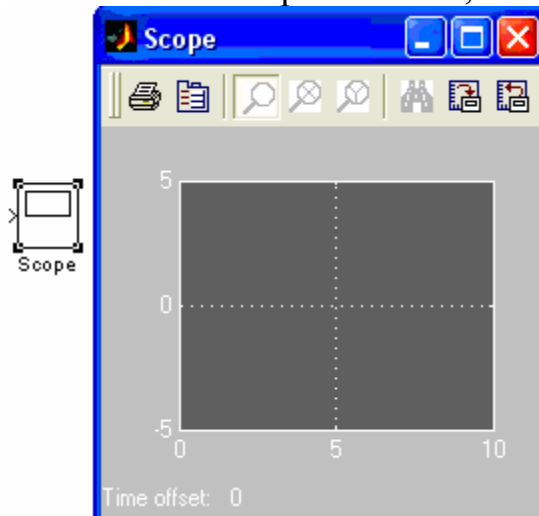


Рис. 1.40. Внешний вид окна блока Scope

По оси ординат шкалы измерений откладываются значения наблюдаемой (входной) величины, по оси абсцисс – значения модельного времени. По умолчанию для оси ординат используется диапазон $[-5; 5]$, а для оси модельного времени – $[0; 10]$.

Если входная величина блока Scope является вектором, то в окне строятся графики изменения всех элементов этого вектора. В этом случае для каждого элемента вектора в окне строится отдельная кривая, отражающая динамику его изменения. Выводимые кривые различаются цветом, который устанавливается автоматически. Одновременно в окне Scope может отображаться до 30 кривых.

Для управления параметрами окна Scope в нем имеется панель инструментов (рис. 1.41):



Рис. 1.41. Панель инструментов блока Scope

На рис. 1.41 приняты следующие обозначения кнопок управления:

1. Print – печать содержимого окна Scope;

2. Parameters – вызов диалогового окна настройки параметров блока Scope;
3. Zoom – изменение масштаба одновременно по обеим осям графика (X и Y);
4. Zoom X-axis – изменение масштаба по оси абсцисс (X);
5. Zoom Y-axis – изменение масштаба по оси ординат (Y);
6. Autoscale (Автошкалирование) – автоматическая установка оптимального масштаба осей, позволяющая наблюдать осциллограмму с максимальным размером по вертикали и отражается весь временной интервал моделирования;
7. Save current axes setting – сохранение текущего масштаба (установленных параметров) осей;
8. Restore saved current axes setting – установка ранее сохраненных настроек масштаба осей;
9. Floating Scope – переход к осциллографу вида Floating Scope (с «плавающей» границей);
10. Lock/Unlock axes selection – фиксация/разрыв связи между текущей координатной системой окна и отображаемым сигналом (только в случае «плавающего» осциллографа);
11. Signal selection – окно выбора сигналов для отображения (только для осциллографа Floating Scope).

Кнопки 3, 4, 5 являются альтернативными, т.е. в каждый момент времени может быть использована («нажата») только одна из них. Кнопки не доступны до тех пор, пока в окне Scope не отобразится график.

Кнопки 10 и 11 становятся активными при нажатии кнопки 9.

Для изменения масштаба по выбранной оси координат необходимо выполнить следующие действия:

- «нажать» соответствующую кнопку изменения масштаба (по оси X или Y);
- подвести курсор мыши к тому участку графика, который должен быть отображен в новом масштабе;
- нажать один раз левую кнопку мыши.

Первое нажатие левой кнопкой мыши приводит к четырехкратному увеличению масштаба (относительно той точки графика, куда был подведен курсор мыши), каждое последующее нажатие дает увеличение масштаба в два раза.

Можно также растянуть заданный участок графика следующим образом:

- «нажать» соответствующую кнопку изменения масштаба (по оси X или Y);
- подвести курсор мыши к тому участку графика, который должен быть отображен в новом масштабе;

- нажать левую кнопку мыши и, не отпуская ее, выделить отрезок (движением мыши вверх/вниз или влево/вправо, по оси X или Y), который необходимо растянуть.

Нажатие два раза левой кнопки мыши (в любой точке графика) приводит к возвращению автомасштаба графика, т.е. результат получается, аналогичен действию кнопки 6 (см. рис. 1.41).

При нажатии правой кнопки мыши в любой точке графика на окне Score, открывается окно, представленное на рис. 1.42. Первые три команды этого окна становятся активными только после изменения масштаба графика.

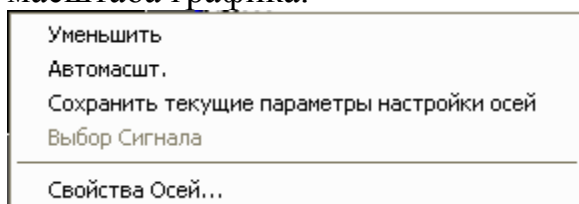


Рис. 1.42. Окно масштабирования и выбора свойств графика блока Score

Команда «Уменьшить» – при ее выборе возвращается предыдущий (последний) масштаб графика, т.е. изменяется масштаб на один шаг назад;

Действия команд «Автомасштаб», «Сохранить текущие параметры ...» и «Выбор сигнала» аналогично нажатию кнопок 6, 7 и 11 соответственно (см. рис. 1.41).

Команда «Выбор Сигнала» становится активной при работе осциллографа в режиме Floating Score (с «плавающей» границей).

Выбор команды «Свойства осей...» открывает окно представленное на рис. 1.43. В окне устанавливаются верхняя (Ymax) и нижняя (Ymin) границы оси ординат, а также заголовок графика. Заголовок будет отображаться (при нажатии кнопки ОК или Принять) непосредственно в окне Score над изображением соответствующего графика. Название графика следует написать вместо надписи «%<SignalLabel>»,

Внимание! В названии графиков рекомендуется использовать только латинские буквы!

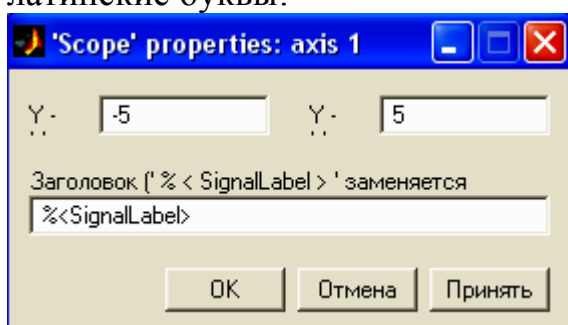


Рис. 1.43. Окно задания свойств осей блока Score

Нажатие кнопки 2 (см. рис. 1.41) открывается окно 'Score' Parameters, содержащее две вкладки (рис. 1.44):

- General (Основное) – для установки параметров осей (Axes) графика (см. рис. 1.44, а);

- Data history (Технология/представление данных) – для ввода значений дополнительных параметров блока Scope (см. рис. 1.44, б).

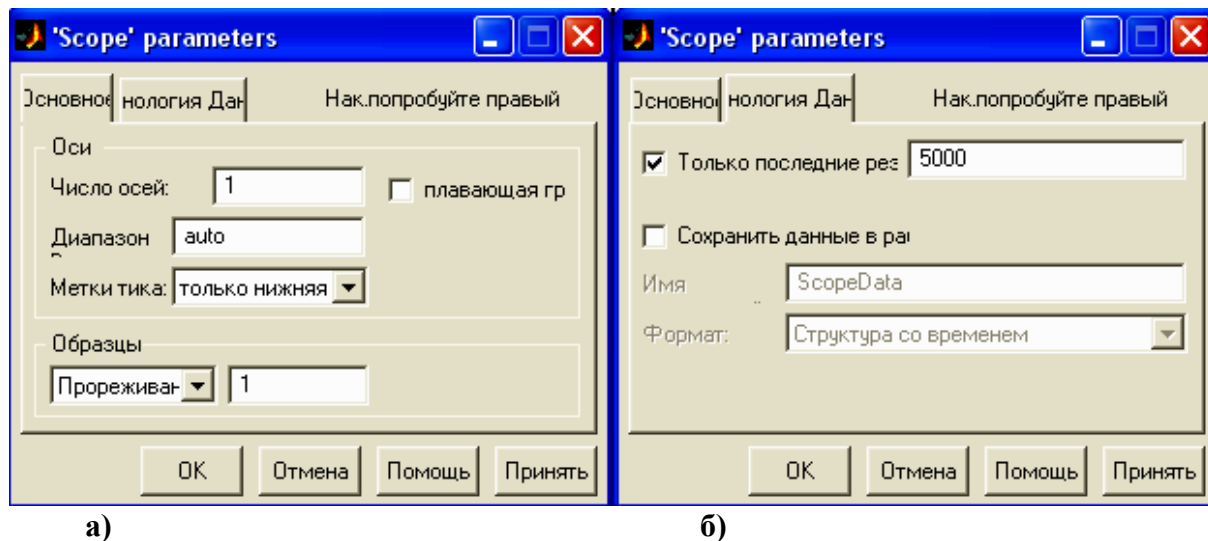


Рис. 1.44. Параметры блока Scope

В нижней части окна (см. рис. 1.44) расположены кнопки, аналогичные кнопкам окон настройки параметров других библиотечных блоков: ОК (Подтвердить установку), Cancel (Вернуться назад/Отмена), Help (Вызов помощи в формате HTML); Apply (Принять).

1. На вкладке General (Основное) имеются следующие поля (см. рис. 1.44, а):

- Numbers of axes (Число осей) – можно задать любое целое число, которое будет соответствовать количеству графических полей в окне Scope. Одновременно при этом изменяется количество входов блока Scope.

Оси Y всех создаваемых графиков являются независимыми по отношению друг к другу, а для формирования координаты X используются одни и те же моменты модельного времени.

Внимание! Для наглядности результата рекомендуется на одном осциллографе выводить не более 5 графиков!

- Time range (Диапазоны осей) – устанавливается верхняя граница модельного времени, откладываемого по оси абсцисс, по умолчанию установлено auto. В этом случае время вывода на Scope совпадает со временем заданном в параметрах моделирования (см. рис. 1.85, п. 1.6). Можно также задать любой меньший диапазон. Причем если величина заданного интервала моделирования (T_m) не превышает установленного в этом поле значения (то есть весь процесс «умещается» в окне Scope), то под графиком в строке Time offset (Сдвиг по времени) выводится значение 0.

Если же интервал моделирования превышает установленное значение, то в окне Scope отображается только график,

соответствующий последнему отрезку времени, меньшему по размеру, чем Time range и равному $T_m - n \times (\text{Time range})$, где T_m – длительность интервала моделирования, n – целое число. При этом в строке Time offset выводится величина «скрытого» интервала времени (длиной $n \times (\text{Time range})$). Например, если значение Time range равно 7, а длительность интервала моделирования составляет 16 единиц времени, то в окне Score будет выведен график моделируемого процесса за последние 2 единицы времени, а строка под графиком будет иметь вид: Time offset: 14.

- Tick labels (Метки осей) – задается вид оформления осей в графиках окна Score. Возможны варианты:

- отображаются все оси, т.е. (деления по горизонтальным осям (X) будут наноситься вдоль каждой из осей всех графиков, хотя масштаб по осям X одинаковый),
- нет осей и надписей на них, т.е. оси графика не отображаются, и график занимает всю рабочую область окна;
- отображается только одна нижняя ось X, деления по горизонтальным осям остальных графических полей (если их несколько) будут отсутствовать.

- флажок floating score – переход к осциллографу с «плавающей» границей. При установленном флажке, Score отображается как блок без входа, и если он был связан по входу с другими блоками, то эти связи «обрываются». Тот же эффект оказывает щелчок на кнопке с таким же названием, находящейся на панели инструментов блока (кнопка 9, см. рис. 1.41);

- Поле Sampling (Образцы/Дискретизация) – обеспечивает выбор дискретности измерения отображаемых величин (характеристик системы и модельного времени).

Установка дискретности измерения характеристик системы и времени выполняется раздельно. Выбор производится с помощью выпадающего меню, содержащего два пункта:

Decimation (Прореживание, кратность выхода данных) – установка дискретности измерения характеристик системы и Sample time – установка дискретности измерения модельного времени.

Для ввода требуемых значений используется строка редактирования, расположенная справа от меню.

По умолчанию для измеряемых характеристик дискретность равна 1, а для модельного времени – нулю. Это означает, что значение наблюдаемой характеристики измеряется на каждом шаге моделирования, а модельное время считается непрерывным. Если параметр Decimation установить равным, например, 3, то значение исследуемой характеристики будет определяться только 1 раз в течение

трех шагов моделирования. Параметр Decimation может принимать только целочисленные значения. Дискретность измерения модельного времени – это фактически величина шага моделирования. По умолчанию она равна 0,02. Для дискретных систем величина шага может быть задана либо в виде положительной константы, либо в форме вычисляемого выражения. Дискретность изменения модельного времени определяет момент окончания вывода данных в окно Scope: если очередной интервал заканчивается за пределами окна, то данные для него не выводятся.

Замечание! Параметр Sample time имеется практически во всех библиотечных блоках Simulink. Для каждого блока значение этого параметра устанавливается индивидуально. Если блок реализует некоторую функцию, то параметр Sample time определяет дискретность вычислений. Поэтому для корректной работы модели необходимо согласовывать установку параметра для взаимосвязанных блоков.

2. Поле Data history (Технология/История данных) (см. рис. 1.44, б) позволяет задавать максимальный объём (начиная с конца) и способ хранения отображаемых в окне Scope данных. Количество сохраняемых точек вводится в строке редактирования – Limit data point to last (соответствует числу строк матрицы, рис. 1.45, окно б).

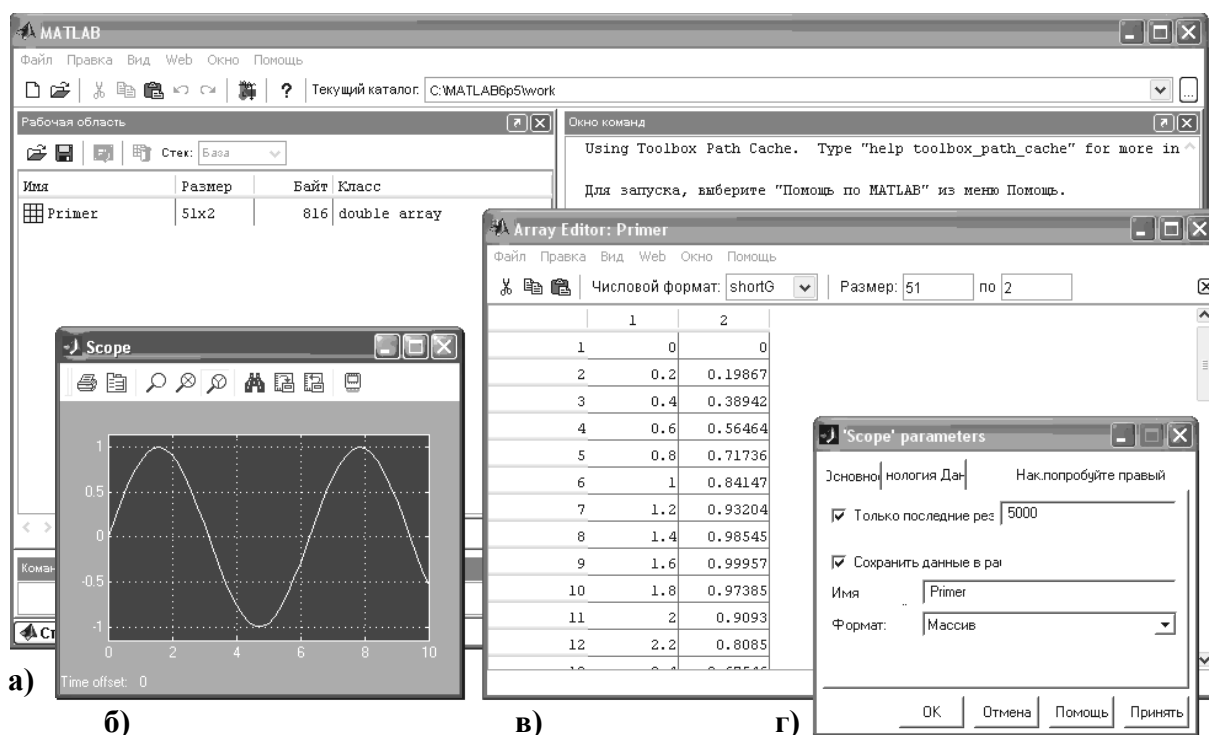


Рис. 1.45. Пример сохранения данных с блока Scope в рабочую область

Способ хранения (формат) указывается с помощью флажка Save data to workspace (Сохранить данные в рабочую область): структура со временем (Structure with time), структура (Structure) или массив/матрица (Array)

(см. рис. 1.45, окно г). Если этот флажок установлен, то изображаемые в окне Scope данные сохраняются в рабочей области MatLab в виде матрицы, аналогичной по структуре MAT-файлу (см. рис. 1.45, окно а).

Имя переменной, под которой будут сохраняться данные в рабочем пространстве системы, указывается в строке редактирования (по умолчанию присваивается имя – Scope Data, можно задать другое, причем рекомендуется писать латинскими буквами, например «Primer») (см. рис. 1.45, окно г).

1.3.11.2. Блок Floating Scope

Включение опции floating scope превращает осциллограф в «плавающий». При этом входы отключаются, а связи обозначаются пунктирными линиями. «Плавающий» осциллограф есть и как отдельный блок в составе блоков регистрирующих устройств. Он отличается от «обычного» тем, что у него отсутствуют в явном виде входы. Это упрощает построение моделей с большим числом осциллографов за счет отсутствия соединений между выходами тех или иных сигналов и входами осциллографа.

Рис. 1.46 показывает применение «плавающего» осциллографа.

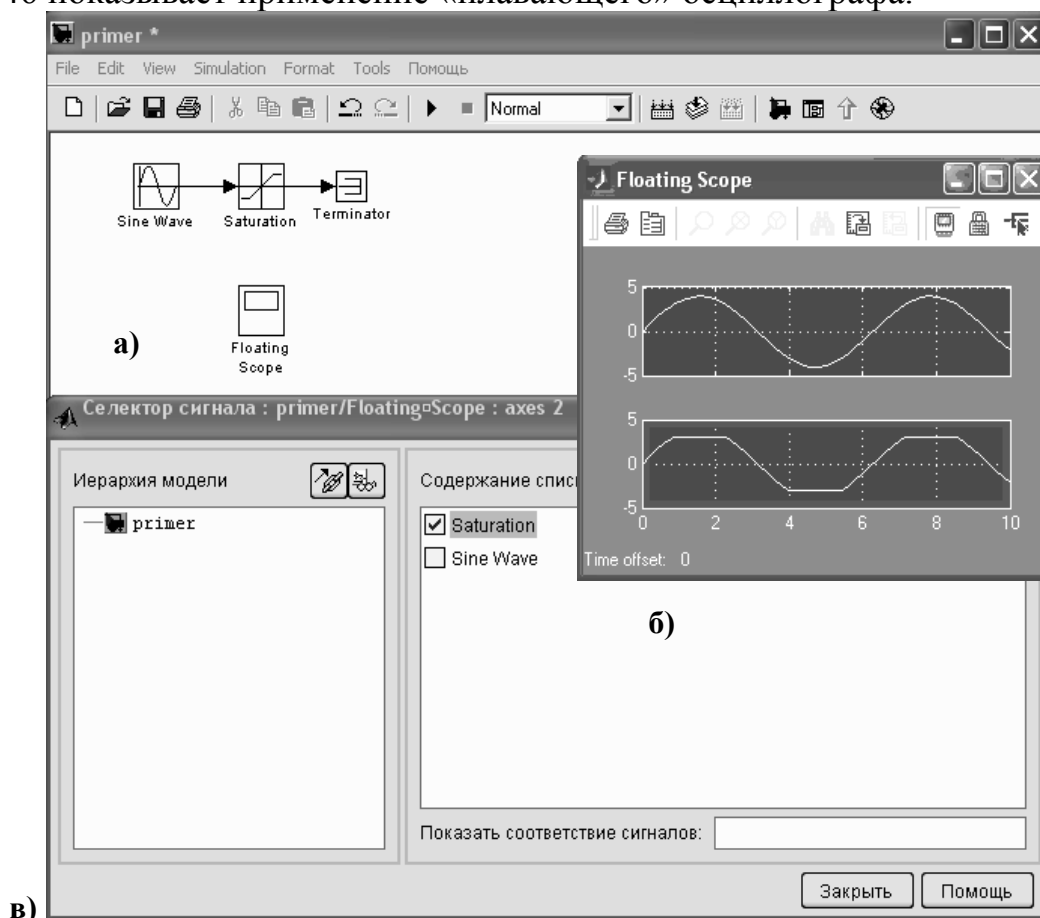


Рис. 1.46. Пример работы с плавающим осциллографом Floating Scope

Здесь моделируется ограничитель, содержащий источник синусоидального сигнала и блок его ограничения, нагруженный на терминатор – эквивалент нагрузки (см. рис. 1.39 и рис. 1.46, а). На рис. 1.46, б выведены осциллограммы блока Floating Scope. Запустив модель, можно настроить осциллограф на отображение нужных сигналов.

Для настройки осциллографа Floating Scope, прежде всего, необходимо вывести его окно параметров (см. рис. 1.46, в), наведя на блок осциллографа курсор мыши и дважды щелкнув левой кнопкой мыши. Затем выделить соответствующее окно осциллографа и выбрать сигнал, который должен быть в нем представлен. Для выделения окна необходимо подвести к нему курсор мыши и щелкнуть один раз левой кнопкой мыши. Выделенное окно обрамляется рамкой (толстой синей линией). На рис. 1.46 это показано для нижнего окна «плавающего» осциллографа.

1.3.11.3. Блок XY Graph

Блок XY Graph обеспечивает построение двумерных графиков произвольных величин (по осям X и Y) (рис. 1.47), что позволяет строить графики функции в полярной системе координат, фигуры Лиссажу, фазовые портреты и т.п.

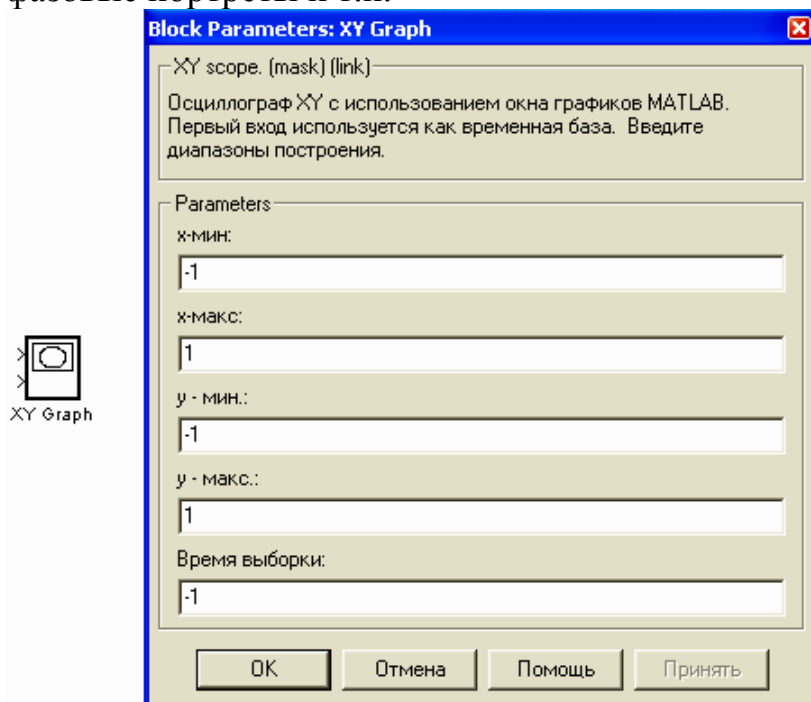


Рис. 1.47. Внешний вид окна настройки блока XY Graph

Блок XY Graph имеет два входа, первый из которых предназначен для ввода аргумента (ось X), второй – для ввода значений функции этого аргумента (ось Y).

Окно XY Graph, в котором непосредственно отображается зависимость $Y = f(X)$ открывается автоматически при запуске модели.

Двойной щелчок левой кнопкой мыши на изображении блока (предварительно помещенного из библиотеки в окно блок-схемы), открывает окно настройки параметров блока XY Graph (см. рис. 1.47).

Окно настройки параметров блока XY Graph включает в себя:

- диапазоны осей графика (X-min, X-max – для оси абсцисс и Y-min, Y-max – для оси ординат);
- шаг модельного времени (Sample time), по умолчанию его значение равно –1. Это обозначает, что величина шага блока XY Graph, совпадает с шагом установленным для модели в целом, либо со значением одноименного параметра предшествующего блока.

1.3.11.4. Блок Display

Блок Display предназначенный для отображения (вывода на экран) численных значений величин, фигурирующих в модели.

Блок имеет 4 параметра настройки (рис. 1.48):

1. Format – с помощью выпадающего меню задает формат вывода, содержащего 5 пунктов: short, long, short_e, long_e, bank.

Следующие два параметра используются так же, как и одноименные параметры блока Scope (см. п. 1.3.11.1).

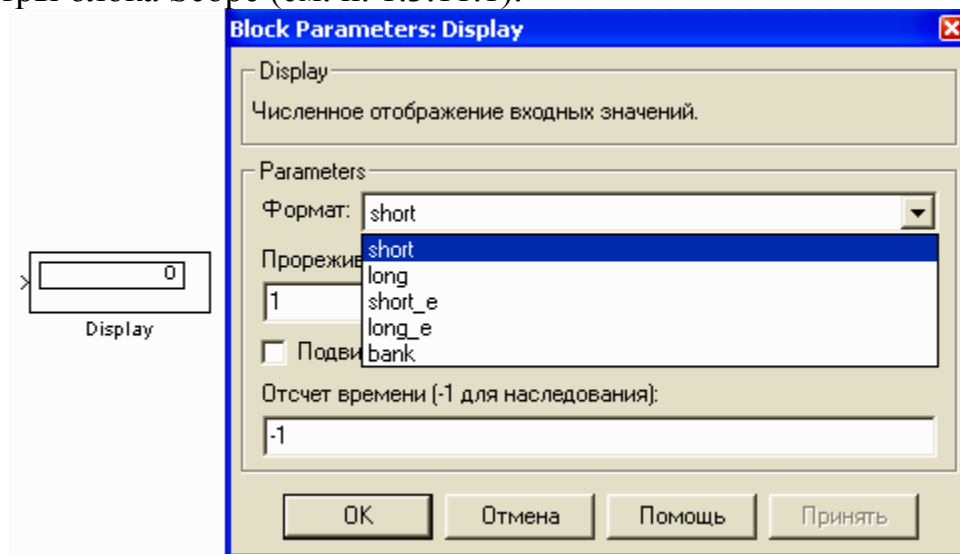


Рис. 1.48. Внешний вид окна настройки блока Display

2. Decimation (Прореживание) – определяет периодичность вывода значений в окне Display. При значении 1 блок Display выводит все данные вектора, при значении 2 – каждый второй, при 3 – каждый третий элемент вектора и т.д.;

3. Переключатель Floating display (Подвижное отображение) позволяет указывать способ использования блока Display в блок-диаграмме – аналогично блоку Floating Scope (см. П. 1.3.11.2);

4. Поле Sample Time задает величину шага модельного времени, т.е. дискретность вывода данных в окно Display. Данный параметр используется только для дискретных во времени процессов. По умолчанию его значение установлено равное «-1», для непрерывных процессов и блоков изменять не рекомендуется.

Блок Display может использоваться для вывода как скалярных, так и векторных данных. Если отображаемая величина является вектором, то исходный формат блока изменяется автоматически. Об изменении формата говорит маленький черный треугольник, появляющийся в нижнем правом углу блока. Для каждого элемента вектора создается свое мини-окно, но чтобы они стали видимы, необходимо «растянуть» изображение блока. Для этого следует выделить блок, подвести курсор мыши к одному из его углов (курсor при этом примет форму двойной стрелки), нажать левую кнопку мыши, и, не отпуская ее, растянуть изображение блока. После того как кнопка мыши будет отпущена, на экране появятся дополнительные окна с выведенными в них значениями элементов вектора. Если хотя бы один элемент вектора остался «за кадром», на изображении блока Display по-прежнему присутствует черный треугольник. В этом случае операцию «растягивания» блока следует повторить.

Дисплей обеспечивает динамическое отображение данных, то есть можно наблюдать их изменение в процессе моделирования.

1.3.12. Раздел Sources

Блоки, входящие в раздел Sources (Источники), предназначены для формирования сигналов, обеспечивающих управление работой модели в целом или отдельных ее частей. Все блоки-источники имеют по одному выходу и не имеют входов (рис. 1.49).

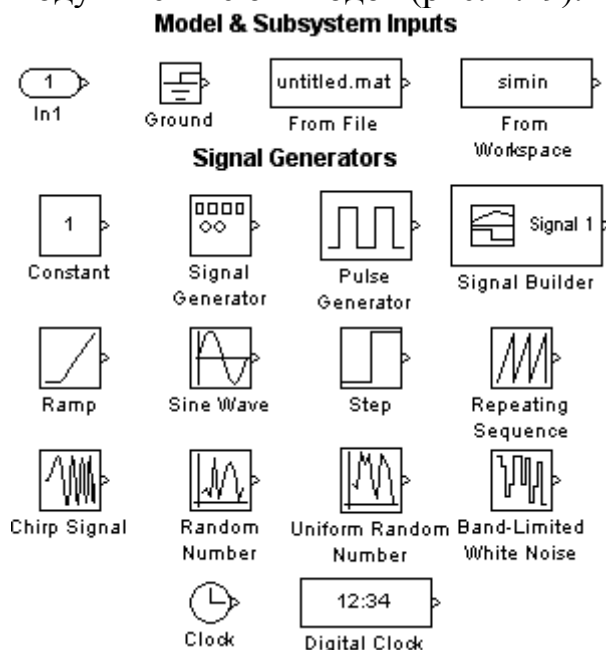


Рис. 1.49. Внешний вид блоков раздела Sources

Раздел Sources содержит две группы блоков: Model & Sybsystem Inputs (Входы моделей и подсистем) и Signal Generators (Генераторы сигналов).

В первую группу входят четыре блока, обеспечивающих поступление сигналов в S-модель или подсистему Simulink извне (из других S-моделей, рабочего пространства или МАТ-файлов):

1. In1 – входной порт S-модели или подсистемы; обеспечивает поступление в подсистему сигналов из системы более высокого порядка, формируя вход этой подсистемы (см. рис. 1.20, п. 1.3.8);
2. Ground – создает нулевой сигнал;
3. From File – предназначен для ввода в S-модель данных, хранящихся в МАТ-файле;
4. From Workspace – обеспечивает ввод в модель данных непосредственно из рабочей области MatLab.

Структура данных в МАТ-файле представляет собой многомерный массив с переменным числом строк, которое определяется числом регистрируемых переменных. Элементы первой строки содержат последовательные значения модельного времени, элементы в других строках – соответствующие им значения переменных

Последние два блока отличаются от первых тем, что обеспечивают использование в модели различных числовых данных, полученных ранее как с помощью Simulink, так и другими средствами MatLab.

Во вторую группу раздела включены блоки, формирующие выходную величину как заданную функцию времени. Они представляют собой своеобразные генераторы сигналов заданного вида. Эту группу образуют следующие блоки:

1. Constant – формирует постоянную величину (скаляр, вектор или матрицу);
2. Signal Generator – создает непрерывный колебательный сигнал одной из волновых форм (синусоидальной, прямоугольной, треугольной) или случайный сигнал;
3. Pulse Generator – представляет собой генератор непрерывных прямоугольных импульсов;
4. Signal Builder – создает один или несколько сигналов произвольной формы (const, sin, треугольной, квадратной, шум и т.п.);
5. Ramp – создает линейно возрастающий (или убывающий) сигнал;
6. Sine Wave – генерирует гармонический сигнал;
7. Step – генерирует сигнал в виде ступеньки (ступенчатый сигнал) с заданными параметрами (время начала ступеньки и ее высота).
8. Repeating Sequence – генерирует периодическую последовательность чисел, указанных в таблице пар время-значение. Значения времени должны монотонно возрастать;

9. Chirp Signal – представляет собой генератор гармонических колебаний, переменной частоты. Частота линейно изменяется во времени;

10. Random Number – служит источником дискретного сигнала, значения амплитуды которого являются случайной величиной, распределенной по нормальному закону;

11. Uniform Random Number – служит источником дискретного сигнала, значения амплитуды которого являются случайной равномерно распределенной величиной;

12. Band-Limited White Noise – является генератором «белого шума» с ограниченной полосой частот;

13. Clock – источник непрерывного временного сигнала, его работа основана на использовании аппаратного таймера компьютера);

14. Digital clock – формирует дискретный временной сигнал.

Как и другие блоки библиотеки Simulink, все блоки-источники могут быть настроены пользователем, за исключением блока Clock.

1.3.12.1. Блок Constant

Блок Constant генерирует процессы, которые являются неизменными во времени, то есть характеризуются постоянным значением (рис. 1.50). Для блока Constant настраивается один параметр – Constant value, который может быть введен и как скалярная величина, вектор-строка или матрица.

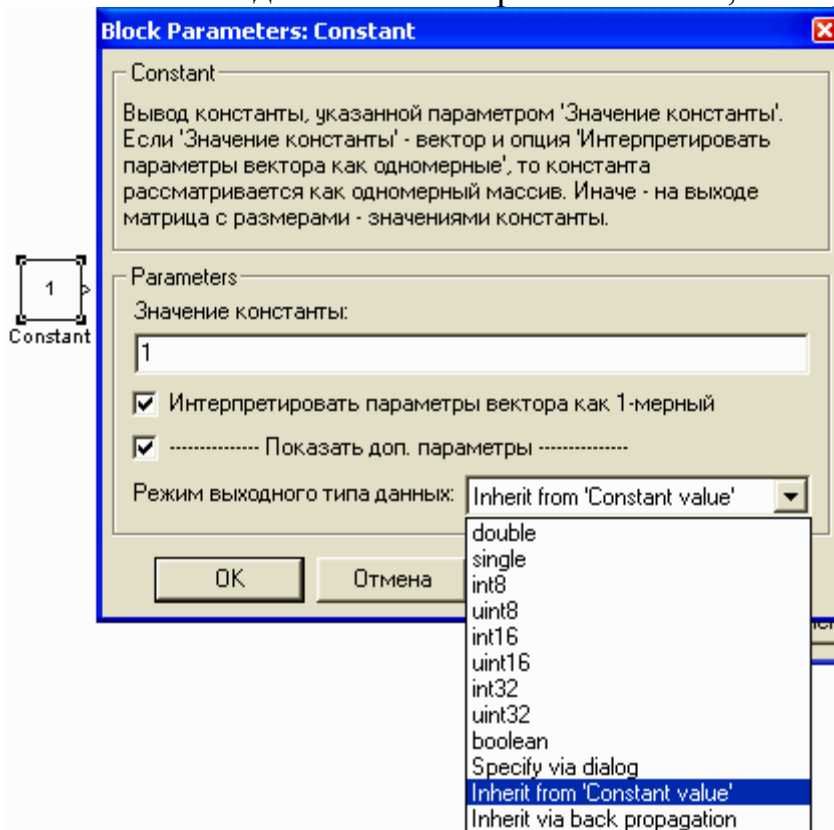


Рис. 1.50. Внешний вид и параметры блока Constant

Вектор констант в частности следует задавать в квадратных скобках: например, запись $[-1 \ 0 \ 1 \ 2 \ 3.14]$ задает вектор из пяти констант со значениями $-1, 0, 1, 2$ и 3.14 .

При установленном флажке Interpret vector parameters as 1-D вектор параметров интерпретируется как одномерный.

Блок Constant может выдавать постоянный сигнал с разным форматом его представления. Возможные форматы представлены в открытом списке на рис. 1.50. Например, если выбран формат double – данные будут представлены в формате удвоенной длины.

1.3.12.2. Блок Signal Generator

Блок Signal Generator представляет собой генератор непрерывных колебательных сигналов. Диалоговое окно настройки блока Signal Generator содержит следующие параметры (рис. 1.51):

- Wave form (форма волны) – в раскрывающемся списке можно выбрать одну из четырех форм периодического процесса: Sine (Синусоидальные волны), Square (Прямоугольные волны), Sawtooth (Треугольные волны), Random (Случайные колебания);
- Amplitude – значения амплитуды генерируемых колебаний;
- Frequency – частота колебаний;
- Units – единицы измерения частоты колебаний; с помощью раскрывающегося списка можно задать одну из единиц измерения: Hertz (Герцы), Rad/Sec (Рadiany в секунду).

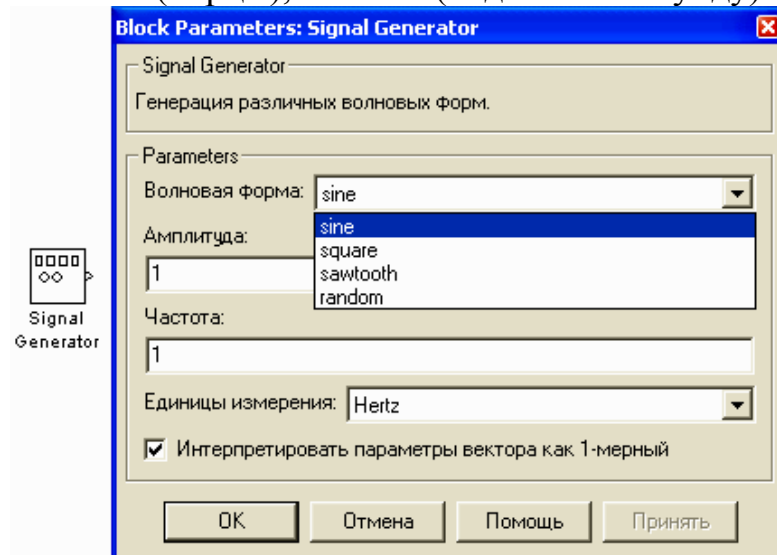


Рис. 1.51. Внешний вид и параметры блока Signal Generator

При выборе пункта Random в списке Wave form генерируется сигнал, значения которого равномерно случайно распределены в диапазоне, указанном параметром Amplitude, а значения моментов времени, в которые происходят скачкообразные изменения сигнала, разделены на величину

шага моделирования, устанавливаемого командой Simulation → Simulation Parameters → Solver.

1.3.12.3. Блок Pulse Generator

Источник дискретных импульсов Pulse Generator служит для создания прямоугольных однополярных импульсов (рис. 1.52).

Параметры источника по умолчанию обеспечивают формирование однополярных прямоугольных импульсов. Можно устанавливать следующие параметры сигнала этого источника:

- Pulse type – тип импульсов. В раскрывающемся списке можно выбрать Time based (для непрерывного сигнала, аргумент – время), Sample based (для дискретного времени, аргумент – число дискретов времени);
- Amplitude – амплитуда сигнала (высота прямоугольных импульсов);
- Period – период сигнала (кратный эталонному времени);
- Pulse width – ширина импульсов (в процентах от периода) – скважность сигнала;

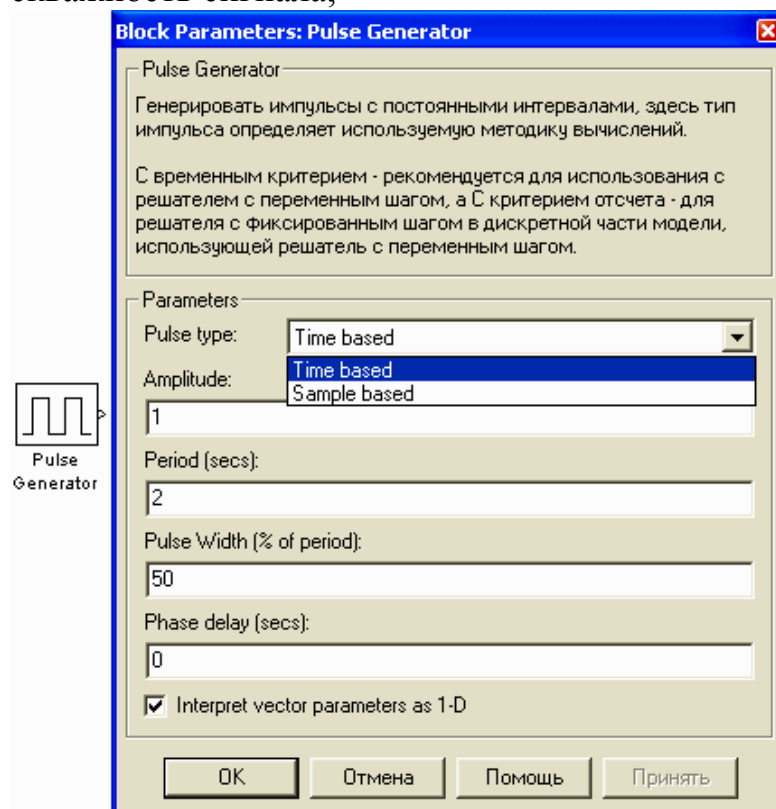


Рис. 1.52. Внешний вид и параметры блока Pulse Generator

- Phase delay – величина задержки первого импульса относительно $t = 0$ (кратная эталонному времени);
- Simple time – эталонное время.

1.3.12.4. Блок Signal Builder

Блок Signal Builder (Конструктор сигналов) – источник одного или нескольких сигналов с устанавливаемой пользователем формой сигнала. Количество портов на выходе блока соответствует числу сигналов.

По умолчанию на выходе блока Signal Builder один сигнал – прямоугольный импульс (рис. 1.53).

Форму сигнала можно изменять с помощью окна, представленного на рис. 1.54. Используя мышь, можно перемещать линии импульса, выполнив следующие действия:

- подвести курсор мыши к линии или любой точке на графике (в окне Signal Builder), внешний вид курсора при этом изменится;
- нажать и, не отпуская левую кнопку мыши, переместить линию (или точку) по горизонтальной или вертикальной оси или изменить наклон линии;
- или нажать один раз левую кнопку мыши – на концах линий при этом появляются жирные точки, нажать еще раз левую кнопку мыши – линия (или точка) выделится красным цветом. Далее можно задать координаты линии или точки в окне Signal Builder (см. рис. 1.53 строки «Левая точка», «Правая точка» «Т:», «Y:»).

Внимание! Разделять целую и дробную часть в MatLab можно только точкой.

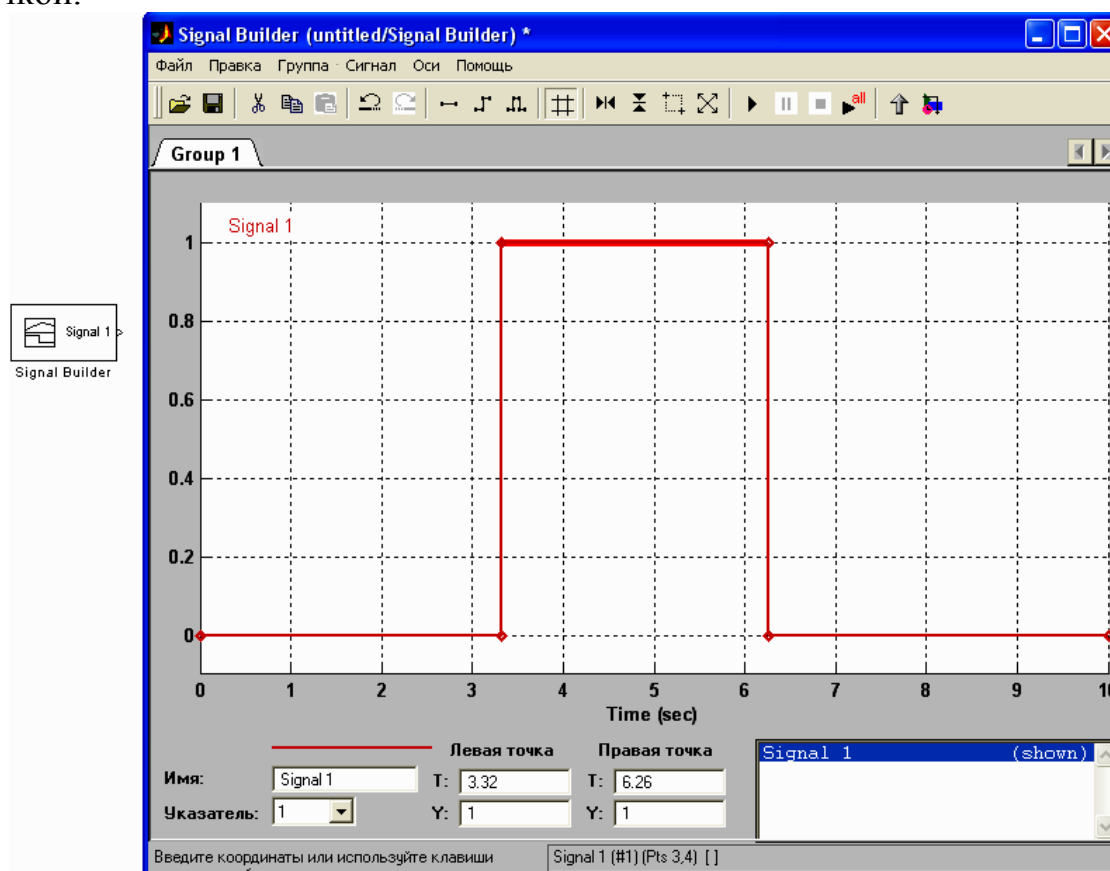


Рис. 1.53. Окно конструктора сигналов Signal Builder

Задать новый сигнал или изменить форму имеющего, можно выполнив следующие действия:

– на панели инструментов (рис. 1.54) выбрать Сигнал → Новый (или Заменить на) → Форма сигнала (Постоянный, Шаг, Импульс, Квадратная, Треугольная, Гармонический сигнал или Шум). Кнопка «Заменить на» становится активной, только если сигнал выделен, т.е. на концах линий есть жирные точки.

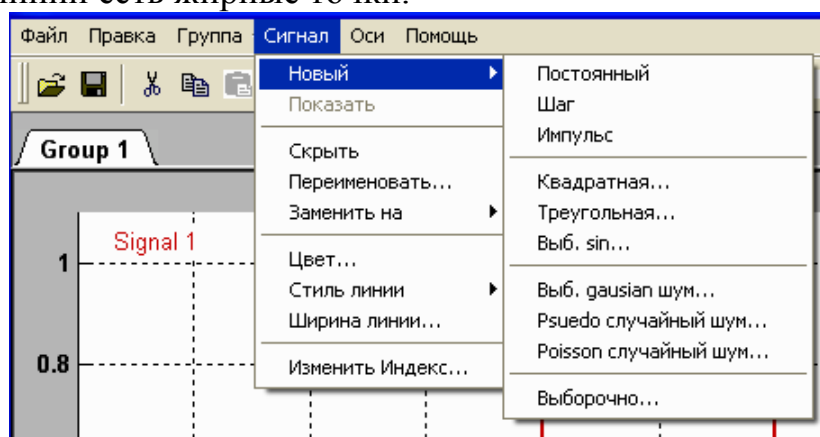
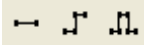


Рис. 1.54. Панель инструментов окна Signal Builder

Новый сигнал вида Постоянный, Шаг и Импульс можно также создать, нажав на одну из кнопок , расположенную на панели инструментов (см. рис. 1.53).

Выбрав один из сигналов вида «Квадратная», «Треугольная», «Выб. sin», в открывшемся диалоговом окне необходимо задать его параметры в соответствии с рис. 1.55:

- строка «Частота» – определяет количество периодов за 1 секунду модельного времени;
- строка «Амплитуда» – определяет амплитуду сигнала по оси Y ;
- строка «Смещение» – показывает сдвиг периодического сигнала относительно нулевой оси. Можно, например, задать не симметричный гармонический сигнал, с разными амплитудами в положительную и отрицательную сторону или однополярный сигнал.

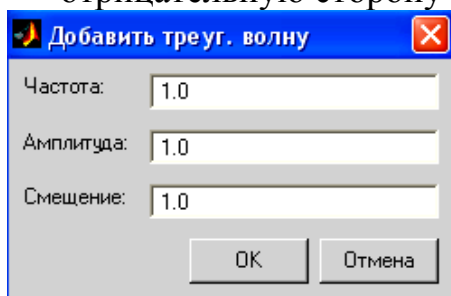


Рис. 1.55. Параметры треугольного периодического сигнала блока Signal Builder

Если в окне блока Signal Builder создано больше одного сигнала, то не нужные сигналы можно удалить. Единственный сигнал не может быть удален, его можно только заменить на другой или удалить блок Signal

Builder целиком. Для удаления сигнала его нужно предварительно выделить (на концах линий при этом появляются жирные точки) и нажать кнопку Delete (на клавиатуре) или кнопку Вырезать  на панели инструментов окна Signal Builder.

На панели инструментов во вкладке «Сигнал» можно также (см. рис. 1.54) изменить цвет (выбрать из гаммы цветов), стиль (сплошная, пунктир или точечный) и ширину (задать произвольную в открывающемся окне) линий сигнала в окне Signal Builder.

Переименовать существующий сигнал можно в строке «Имя» непосредственно в окне Signal Builder (см. рис. 1.53) или выбрав на панели инструментов вкладку «Сигнал → Переименовать» (см. рис. 1.54).

Масштаб по осям задается с панели инструментов во вкладке «Оси» (рис. 1.56).

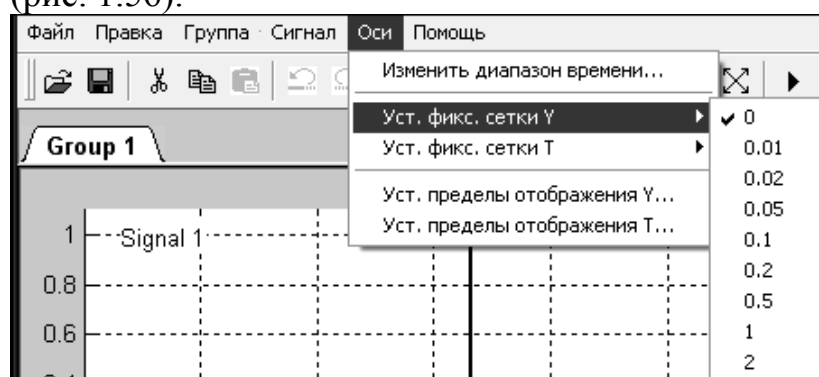


Рис. 1.56. Задание масштаба блока Signal Builder

Можно изменить диапазон времени для сигнала, отображаемого в окне Signal Builder (рис. 1.57).

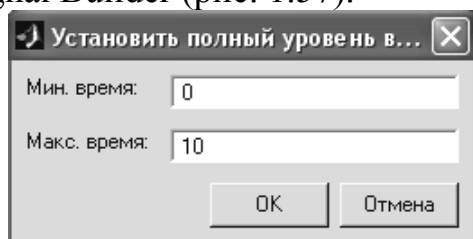


Рис. 1.57. Изменение масштаба по времени блока Signal Builder

При этом следует помнить, что максимальное время ограничено установленным модельным временем (см. рис. 1.85, п. 1.6).

Кнопки , расположенные на панели инструментов (см. рис. 1.53), по функциональному назначению аналогичны кнопкам 3, 4 и 5 блока Score (см. рис. 1.41), т.е. позволяют растягивать масштаб по одной или обеим осям X или Y графика, а также возвращать первоначальный масштаб  (автошкалирование), соответствует кнопке 6 (см. рис. 1.41).

Запустить/остановить расчет модели можно кнопками , расположенными на панели инструментов (см. рис. 1.53, также см. п. 1.6.).

Кнопка , расположенная на панели инструментов окна Signal Builder (см. рис. 1.53) позволяет перейти из окна Signal Builder непосредственно в модель (диаграмму Simulink).

Кнопка  – открывает окно библиотеки Simulink (см. п. 1.3).

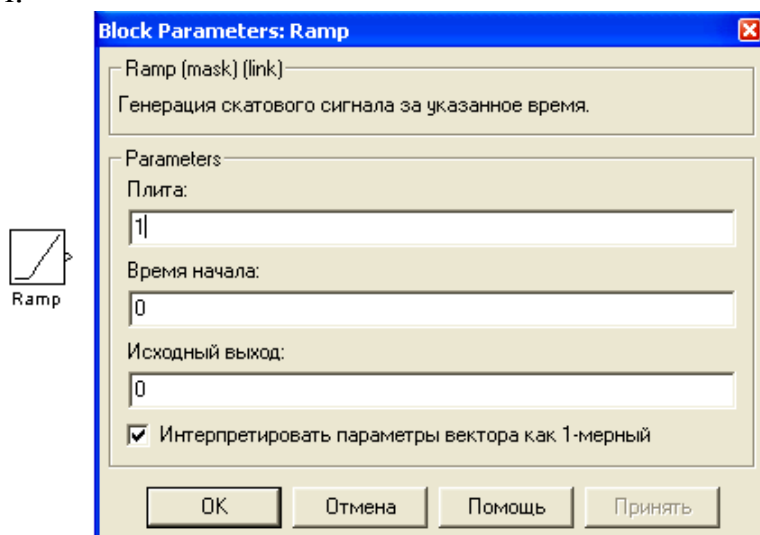
Раздел 1.3.12.5. Блок Ramp

Блок Ramp формирует непрерывно нарастающий сигнал вида (1.3).

$$F(t) = kt + i \quad (1.3)$$

В окне настройки блока Ramp задаются следующие параметры (рис. 1.58):

- Slope – значение скорости нарастания сигнала, угловой коэффициент временной зависимости k ;
- Start time – время начала нарастания сигнала;
- Initial output – значение сигнала до момента начала его нарастания i .



**Рис. 1.58. Окно
настройки
параметров блока
Ramp**

Обратите внимание, что за пределами отрезка времени моделирования (см. п. 1.6), (по умолчанию 10 секунд) воздействие не определено.

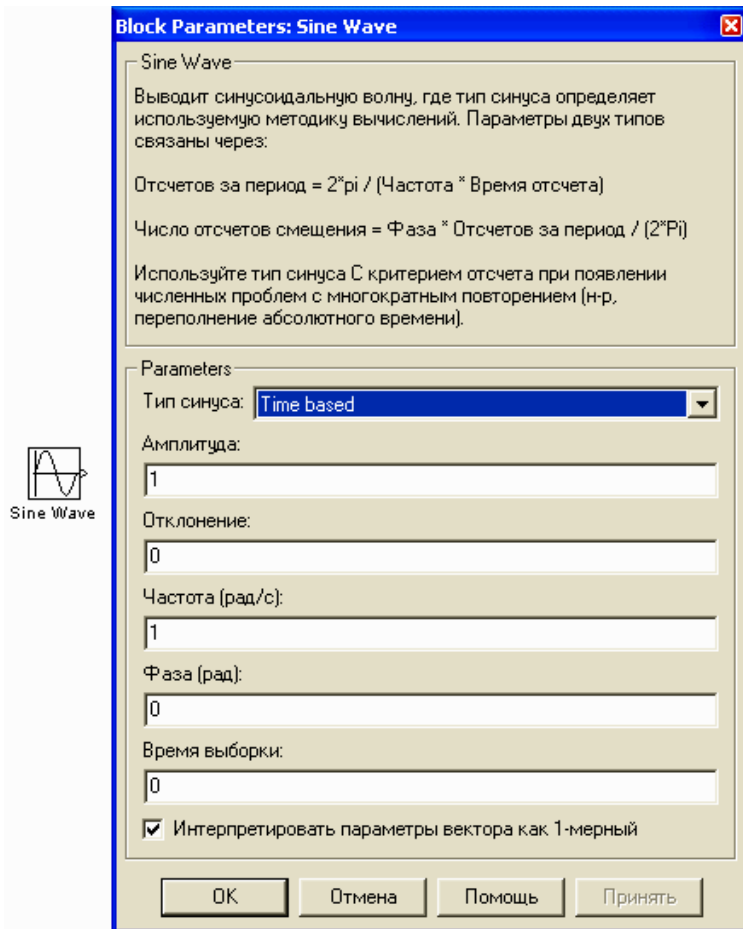
1.3.12.6. Блок Sine Wave

Блок Sine Wave является генератором гармонического сигнала (рис. 1.59).

В окне настройки блока Sine Wave задаются следующие параметры:

- Sine type – тип синусоидальной волны:
 - а) Time based – по текущему алгоритму для непрерывного сигнала или по конечно-разностному алгоритму для дискретных сигналов (аргумент – время);

б) Sample based – по такту дискретности и числу расчетных шагов за один период сигнала (аргумент – число дискретов времени).



**Рис. 1.59. Окно
настройки
параметров блока
Sine Wave**

При формировании синусоидального сигнала по текущему времени используется формула (1.4):

$$f(t) = A \sin(2\pi f t + \varphi) + b, \quad (1.4)$$

где A – амплитуда сигнала, f – частота, t – текущее время, φ – фаза и b – сдвиг сигнала (постоянная составляющая). В условиях ограниченной точности вычислений такой расчет при больших временах может приводить к заметным ошибкам.

При конечно-разностном алгоритме используется представление сигнала в соответствии с уравнением (1.5):

$$\begin{bmatrix} \sin(t + \Delta t) \\ \cos(t + \Delta t) \end{bmatrix} = \begin{bmatrix} \cos(\Delta t) \cdot \sin(\Delta t) \\ -\sin(\Delta t) \cdot \cos(\Delta t) \end{bmatrix} \cdot \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix} \quad (1.5)$$

Этот алгоритм обеспечивает меньшую погрешность округления для больших времен.

При формировании по количеству тактов за период используется соотношение (1.6):

$$f(t) = A \sin(2\pi k f T + \varphi) + b = A \sin(2\pi k + 1)/N + b, \quad (1.6)$$

где T – такт дискретности, N – количество тактов в секунду, l – начальная фаза, выраженная в тактах, $k = 0, 1, \dots, N - 1$ – номер текущего шага.

- Amplitude – амплитуда синусоидального сигнала;
- Bias – отклонение смещение (постоянная составляющая синусоиды);
- Frequency (rad/sec) – частота колебаний (в радианах в секунду);
- Phase (rad) – начальная фаза (в радианах);
- Sample time – величина шага моделирования для данного блока.

Отличия этого блок-генератора от генератора синусоидальных колебаний (блок Signal Generator см. п. 1.3.12.2) состоят в следующем: в блоке Sine Wave можно установить произвольную начальную фазу, задать средний уровень синусоиды, но нельзя задать частоту в герцах.

1.3.12.7. Блок Step

Блок Step обеспечивает формирование единичного управляющего сигнала (ступенчатого сигнала), который можно подключить в любую точку модели (рис. 1.60).

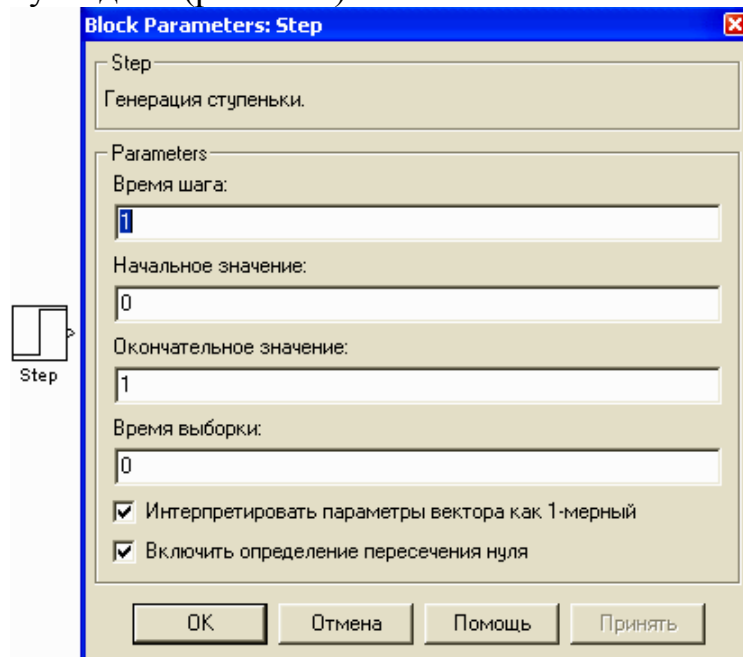


Рис. 1.60. Внешний вид и параметры блока Step

Блок Step имеет 3 параметра настройки:

- Step time (Время шага) – определяет время начала ступеньки – момент времени, в который происходит скачкообразное изменение величины сигнала, значение по умолчанию равно 1;
- Initial value (Начальное значение) – задает значение амплитуды сигнала в начальный момент времени, до скачка. Значение по умолчанию – 0.
- Final value (Конечное значение) – задает значение амплитуды сигнала после скачка, значение по умолчанию равно 1;

- Sample time (Время выборки) – величина шага моделирования.

Ступенчатый сигнал можно задать по знаку как положительный, так и отрицательный. Для задания отрицательного перепада начальное значение должно быть больше чем конечное.

1.3.12.8. Блок *Repeating Sequence*

Блок *Repeating Sequence* представляет собой генератор колебаний с периодической последовательностью. Последовательность задается векторами моментов времени и значениями сигналов в эти моменты времени. Узловые точки сигнала, заданные этими векторами, соединяются отрезками прямых, то есть между ними задается линейная интерполяция.

В окне блока *Repeating Sequence* устанавливаются два параметра (рис. 1.61):

- Time values – вектор значений моментов времени, в которые заданы значения выходной величины;
- Output values – вектор значений выходной величины, которые она должна принять в указанные в первом векторе моменты времени.

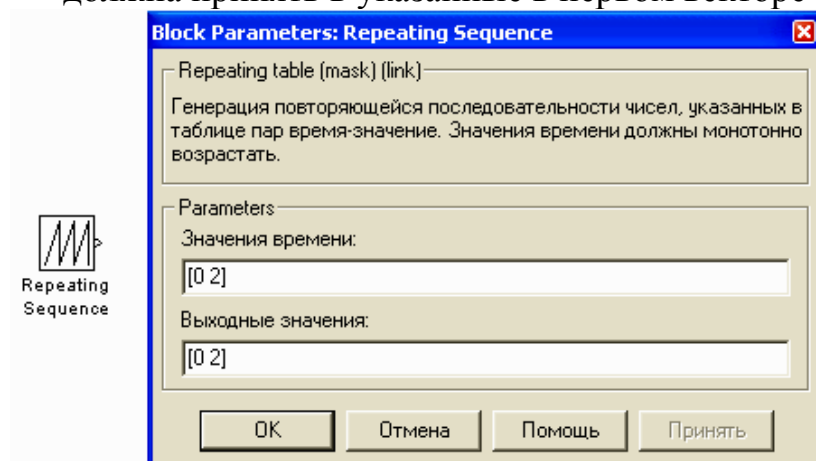


Рис. 1.61.
Внешний вид и
параметры блока
**Repeating
Sequence**

Блок обеспечивает генерирование колебаний с периодом, равным разнице между последним значением вектора Time values и значением первого его элемента. Волна внутри периода имеет форму ломаной линии, проходящей через точки с указанными в векторах Time values и Output values координатами.

Форма заданного сигнала отображается внутри пинтограммы. По умолчанию формируется пилообразный сигнал.

1.3.12.9. Блок *Chirp Signal*

Блок *Chirp Signal* генерирует синусоидальный сигнал единичной амплитуды переменной частоты, причем значения частоты колебаний изменяются с течением времени по линейному закону (рис. 1.62).

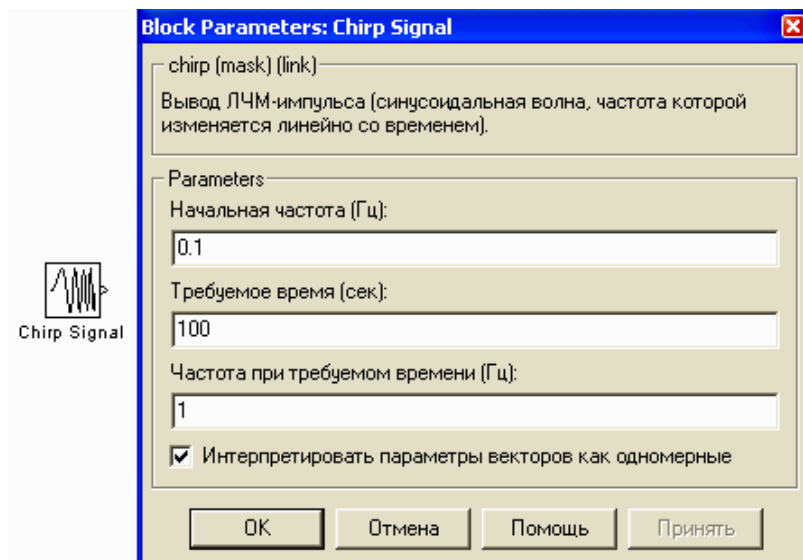


Рис. 1.62. Внешний вид и параметры блока Chirp Signal

В окне настройки блока Chirp Signal предусмотрены следующие параметры:

- Initial frequency (Hz) – начальное значение частоты (в герцах) при $t = 0$ (по умолчанию 0.1 Гц);
- Target time (secs) – второй момент времени (в секундах), время нарастания частоты. Положительная величина (по умолчанию 100 с);
- Frequency at target time (Hz) – значение частоты во второй момент времени (по умолчанию 1 Гц).

1.3.12.10. Блок Random Number

Блок Random Number обеспечивает формирование сигналов, значения которых в отдельные моменты времени являются случайной величиной, распределенной по нормальному закону с заданными параметрами.

В окне настройки этого блока (рис. 1.63) можно установить следующие параметры:

- Mean (Значение) – среднее значение генерируемого процесса;
- Variance (Рассеяние) – среднеквадратичное отклонение от этого среднего;
- Initial seed (Начальное зерно) – начальное значение базы для инициализации генератора последовательности случайных чисел; при фиксированном значении этого параметра генератор всегда вырабатывает одну и ту же последовательность;
- Sample time (Время выборки) – величина шага моделирования.

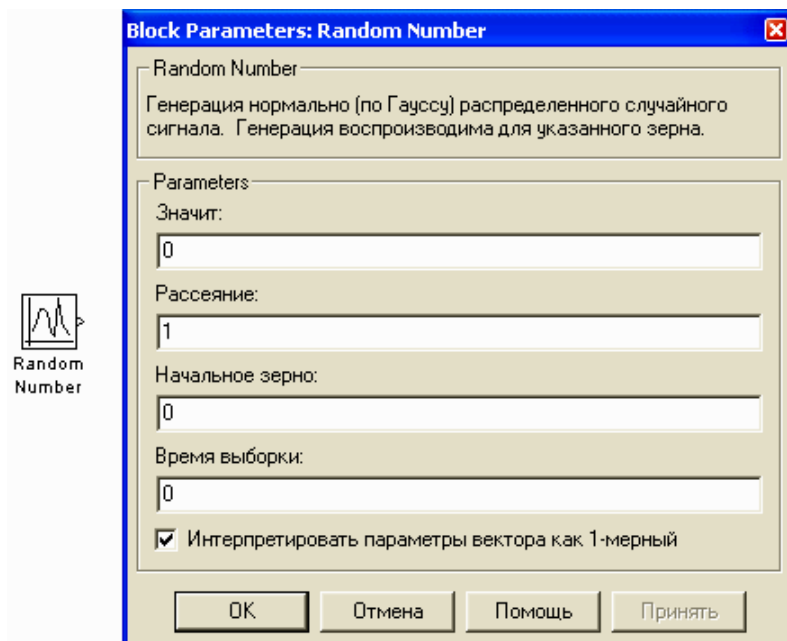


Рис. 1.63. Внешний вид и параметры блока Random Number

Следует помнить, что при повторении экспериментов осциллограмма сигнала с одного и того же блока Random Number не будет повторяться, поскольку сигнал источника является случайным.

1.3.12.11. Блок Uniform Random Number

Блок Uniform Random Number формирует сигналы, значения которых в отдельные моменты времени являются случайной величиной, равномерно распределенной в заданном интервале (рис. 1.64).

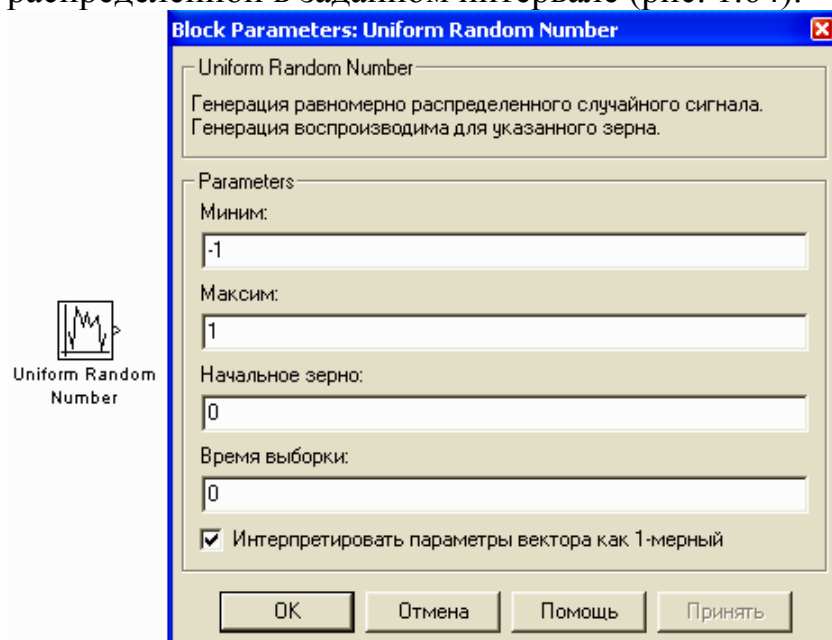


Рис. 1.64. Внешний вид и параметры блока Uniform Random Number

В окне настройки этого блока можно установить следующие параметры:

- Minimum (Минимум) – нижний предел случайной величины;
- Maximum (Максимум) – верхний предел случайной величины;

- Initial seed (Начальное зерно) – начальное значение базы генератора случайных чисел;
- Sample time (Время выборки) – величина шага моделирования.

1.3.12.12. Блок Band-Limited White Noise

Блок Band-Limited White Noise формирует процесс в виде частотно-ограниченного белого шума, с заданной мощностью, равномерно распределенной по частоте.

Данный блок имеет следующие параметры настройки (рис. 1.65):

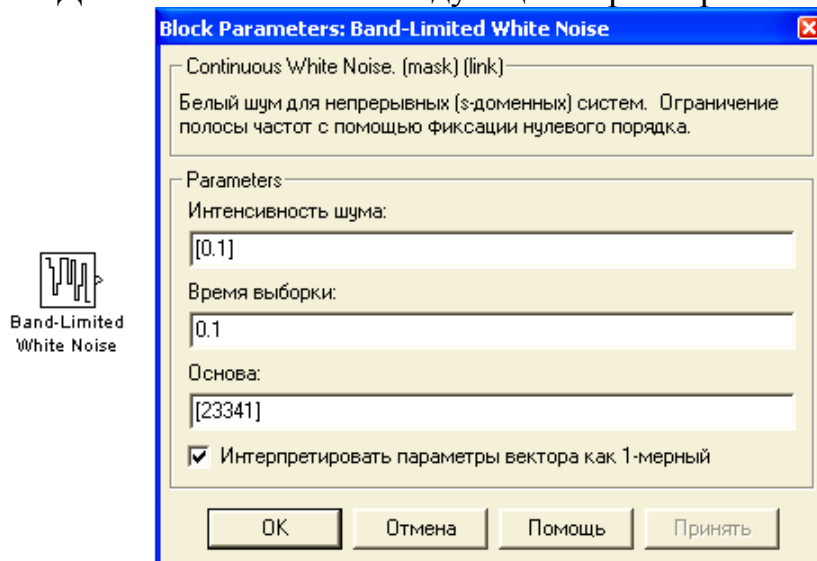


Рис. 1.65. Внешний вид и параметры блока Band-Limited White Noise

- Noise power – значение интенсивности (мощности) белого шума;
- Sample time – значение дискрета времени (определяет верхнее значение частоты процесса);
- Seed – начальное значение базы генератора случайной величины

Генератор фактически является квантователем непрерывного сигнала, представляющего белый шум.

Примечание! Следует различать шаг изменения модельного времени, устанавливаемый при проведении моделирования в окне Simulation parameters и определяющий интервалы времени, через которые производятся вычисления отдельных состояний моделируемой системы (см. п. 1.6), и параметр Sample time (Дискрета времени), определяющий интервал времени, внутри которого выходная величина блока не изменяет своего значения.

Моделируемые процессы сохраняют неизменные значения внутри интервалов времени, устанавливаемых с помощью параметра Sample time. Изменение значений процесса происходит скачкообразно на границе соседних дискретов времени.

Блок Band-Limited White Noise существенно отличается от блоков-генераторов Random Number и Uniform Random Number. Для последних,

установка значения параметра Sample time, отличного от нуля, не является обязательной. В блоке Band-Limited White Noise этот параметр определяет наибольшую частоту в спектре выходного сигнала, она равна величине (в герцах), обратной значению параметра Sample time, а потому последний не может быть равен нулю.

1.3.12.13. Блок Clock

Источник текущего времени Clock служит для генерации чисел, которые являются значениями текущего времени моделирования. Для контроля этого времени может использоваться цифровой индикатор – Display или осциллограф Scope (рис. 1.66).

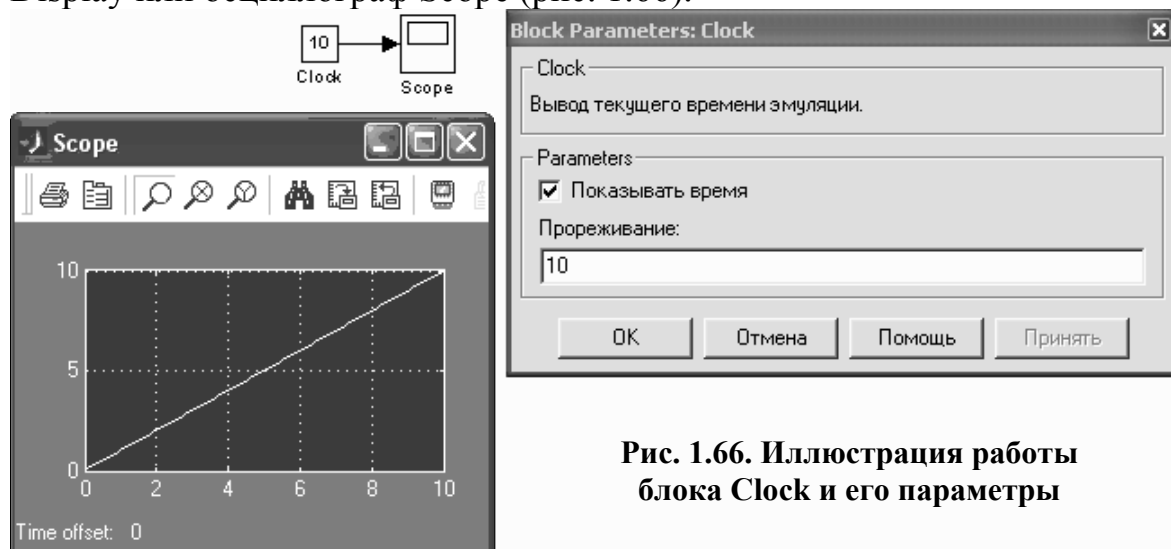


Рис. 1.66. Иллюстрация работы блока Clock и его параметры

Параметром источника является шаг Decimation (Прореживание), с которым меняются отсчеты времени. Флажок Display time (Показывать время) задает отображение времени в блоке источника.

1.3.12.14. Блок Digital Clock

Блок Digital Clock имитирует работу цифровых часов. Он имеет единственный параметр – эталонное время Sample time (по умолчанию 1 с). На рис. 1.67 показана работа источника Digital Clock.

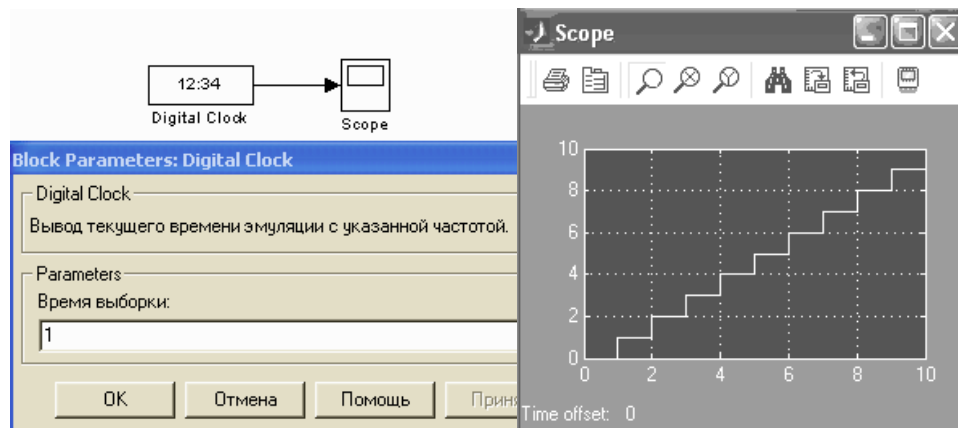


Рис. 1.67. Иллюстрация работы блока Digital Clock и его параметры

Нетрудно заметить, что при заданном по умолчанию эталонном времени источник создает линейно нарастающий ступенчатый сигнал, что соответствует такому же росту текущего времени.

1.3.13. Раздел User-Defined Functions

Раздел User-Defined Functions (Пользовательские функции) содержатся блоки, назначение которых определяет пользователь (рис. 1.68).

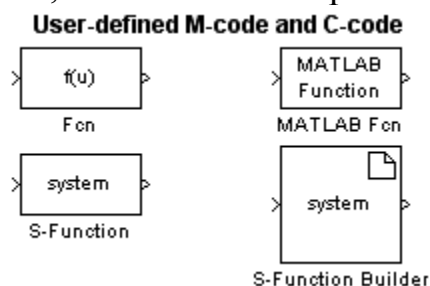


Рис. 1.68. Внешний вид блоков раздела User-Defined Functions

На основе этих блоков пользователь может создавать собственные S-блоки, выполняющие необходимые ему функции.

1.3.13.1. Блок Fcn

В блок Fcn пользователь может ввести любую скалярную функцию от одного (скалярного или векторного) или нескольких аргумента, которая выражается через стандартные, функции MatLab (рис. 1.69). Выражение функции вводится в окне настройки блока. Для обозначения входного сигнала (аргумента функции) используется символ «u» или «u(i)».

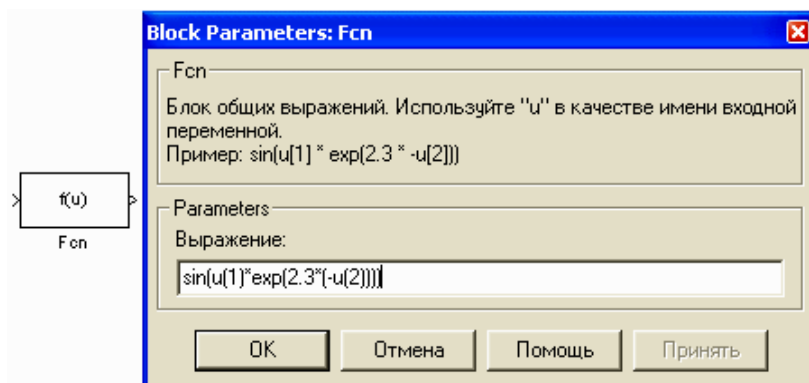


Рис. 1.69. Внешний вид и окно параметров блока Fcn

В окне параметров блока имеется поле Expression для ввода выражения, задающего нужную функцию. Это выражение составляется по правилам, принятым для описания функций на языке Си. Допустимые операторы перечислены ниже в порядке уменьшения приоритета их операций:

- круглые скобки ();
- унарные операторы – и +;
- оператор возведения в степень ^;
- оператор логического отрицания !;
- операторы арифметического умножения * и деления /;
- операторы арифметического сложения + и вычитания –;
- логические операторы отношения <, >, <= и >=;
- операторы отношения «равно» = и «не равно» !;
- оператор логического умножения && (И);
- оператор логического сложения || (ИЛИ).

Операторы отношения и логические операторы возвращают логические значения в виде логического нуля (FALSE) или логической единицы (TRUE). В выражениях для функций этого типа могут использоваться и переменные системы MatLab, находящиеся в рабочем пространстве, например математические функции: abs, acos, asin, atan, cos, cosh, exp, ln, log, log10, sgn, sin, sinh, sqrt, tan, tanh и т.п.

1.3.13.2. Блок MatLab Fcn

Блок задания функции MatLab Fcn служит для задания функций одной переменной «u» или ряда переменных «u(i)» по правилам, принятым для языка программирования базовой системы MatLab. Блок MatLab Fcn позволяет применить к входному сигналу любую подпрограмму обработки, реализованную в виде М-файла.

Это может быть как библиотечная функция пакета MatLab, так и подпрограмма, созданная разработчиком модели.

В первом случае использование блока MatLab Fcn аналогично использованию блока Fcn. Вычисляемое выражение вводится в окне настроек блока в строке MatLab function. Если оно содержит только

обращение к библиотечной функции, то ее аргумент можно явно не указывать. Например, для вычисления квадратного корня входного значения достаточно ввести имя функции `sqrt`.

Другой, более существенной, особенностью блока `MatLab Fcn` является наличие второго параметра настройки – `Output width` (Ширина выходного сигнала). Это означает, что результатом работы блока может быть не только скаляр, но и вектор. Число его элементов должно быть равно значению параметра `Output width`.

Для обозначения входного сигнала используется символ `u`. Если входной сигнал является вектором, то для операций, выполняемых над отдельными его элементами, аргумент также должен быть задан явно. Например, сложение двух элементов входного сигнала должно быть записано в таком виде: `u(1) + u(2)`.

Рис. 1.70, а демонстрирует применение блока `MatLab Fcn` для вычисления функции синуса, заданной только ее именем «`sin`». При этом на входе задается вектор значений входного сигнала, а на выходе формируется вектор выходного сигнала с той же длиной.

Окно параметров этого блока (см. рис. 1.70, б) содержит описание правил задания функции и раздел параметров `Parameters`. В разделе `MatLab function` задается выражение для функции и длина вектора выхода `Output width`, последний параметр определяет количество элементов выходного вектора. Если установить значение этого параметра равным «`-1`», то размер выходного вектора совпадет с размером входа.

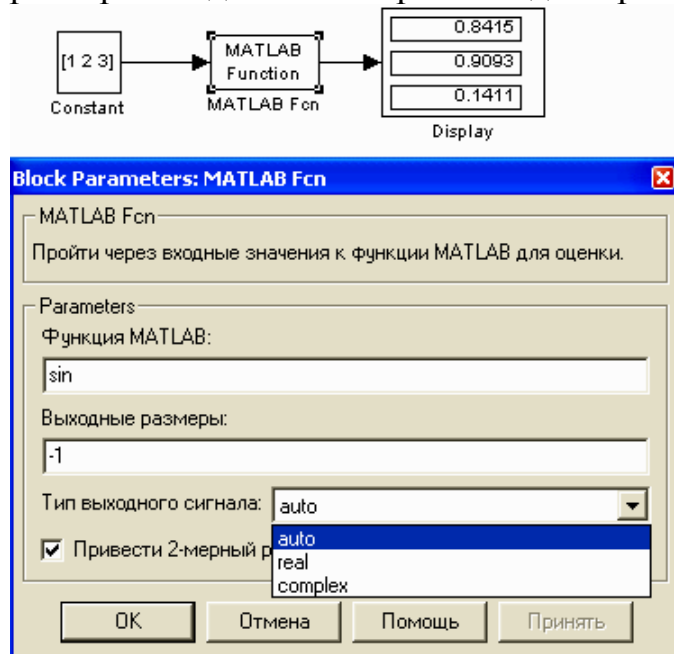


Рис. 1.70. Пример использования и параметры блока `MatLab Fcn`

Правила задания выражения для функции совпадают с правилами, рассмотренными для функции `Fcn` (см. рис. 1.69, п. 1.3.13.1). Если выражение состоит из одной встроенной функции (см. пример на рис. 1.70), то достаточно задать имя функции без входных параметров.

В раскрывающемся списке Output signal type можно выбрать тип выходного сигнала в виде вещественного числа (real), комплексного (complex) или задать автоматический выбор (auto).

Отдельный i -й элемент выходного вектора в окне параметра MatLab function задается в виде функции, которая записана на М-языке и предваряется записью « $u(i)=$ ».

1.3.13.3. Блок задания S-функций

Модели расширения Simulink в конечном итоге являются программами, называемыми S-функциями. Блок задания S-функций одновременно самый простой и самый сложный блок. Простой, потому что окно его параметров, представленное на рис. 1.71, содержит лишь поля для ввода наименования S-функции и ее параметров.

Сложность S-функций обусловлена тем, что они являются вполне самостоятельными программами, написанными на языках MatLab, Си, Ада, Фортран и т.п. и представлены файлами соответствующих форматов. Основное назначение S-функций связано, прежде всего, с созданием новых блоков и решением нестандартных задач, которые трудно решаются с помощью имеющихся библиотек пакета Simulink.

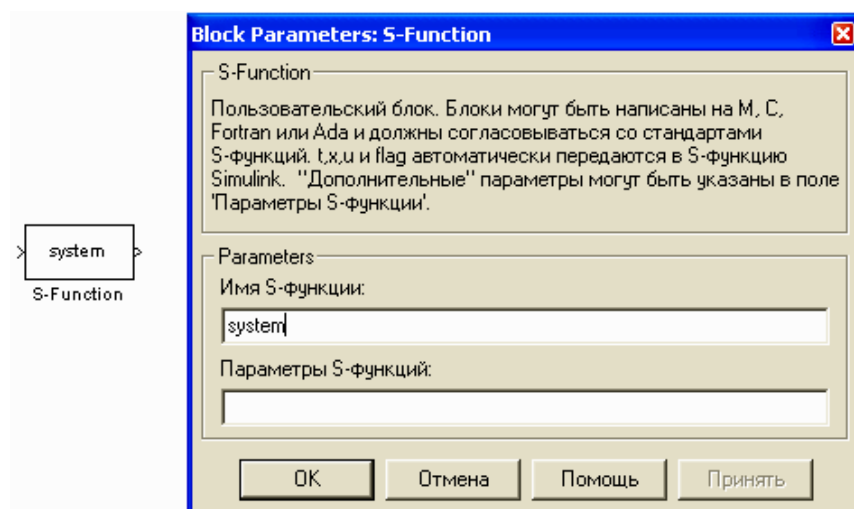


Рис. 1.71. Внешний вид и параметры блока S-Function

С помощью блока S-function пользователь может реализовать в виде Simulink-блока любую программу обработки входного сигнала, включая создание сложных моделей систем, описываемых нелинейными дифференциальными или конечно-разностными уравнениями, и обработку дискретных во времени сигналов.

1.3.13.4. Блок задания S-function Builder

Блок S-function Builder дает возможность пользователю создавать S-функцию в диалоговом режиме.

1.4. Библиотека SimPowerSystems

Для моделирования энергетических систем, включая устройства электротехники и промышленной электроники, служит пакет расширения SimPowerSystems.

Разделы библиотеки SimPowerSystems представлены на рис. 1.71.

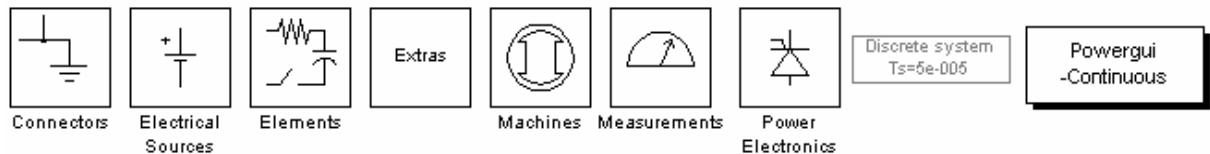


Рис. 1.71. Разделы библиотеки SimPowerSystems

В этот пакет входят библиотеки следующего назначения:

- Electrical Sources – источники электрической энергии и сигналов;
- Elements – линейные и нелинейные компоненты электротехнических и электронных устройств;
- Extra Library – специальная расширенная библиотека блоков-масок;
- Machines – электрические машины;
- Measurements – измерительные и контрольные устройства;
- Phasor Element – элемент моделирования трехфазных систем;
- Power Electronics – блоки устройств силовой электроники (преобразовательная техника);
- Powergui – графический интерфейс пользователя пакета моделирования энергетических систем.

Подробное описание всех блоков входящих в раздел SimPowerSystems дано в работе [1 – 3].

1.5. Библиотека Simulink Extras

Библиотека Simulink Extras является дополнительной библиотекой пакета Simulink. Работа с этой библиотекой ничем не отличается от работы с основной библиотекой.

Раздел библиотеки Simulink Extras имеет следующие наборы блоков:

- Additional Discrete –  дополнительные дискретные, блоки;
- Additional Linear –  дополнительные линейные блоки;

- Additional Sinks –  дополнительные получатели сигналов (регистраторы);
- Flip-Flops –  триггерные блоки;
- Linearization –  линеаризирующие блоки;
- Transformations –  блоки преобразования.

1.5.1. Дополнительные дискретные блоки *Additional Discrete*

Дополнительные дискретные блоки *Additional Discrete* представлены всего четырьмя блоками – по два варианта уже выше описанных блоков (см. рис. 1.5, п. 1.3.1) *Discrete Transfer Fcn* и *Discrete Zero Pole* (рис. 1.72).

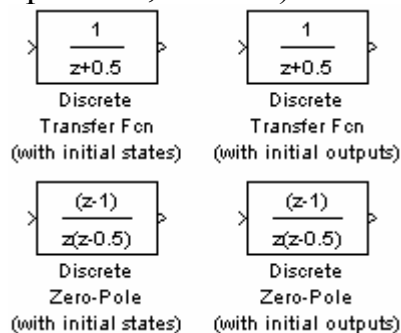


Рис. 1.72. Внешний вид блоков раздела *Additional Discrete*

Их единственным отличием от описанных ранее блоков (см. рис. 1.5, п. 1.3.1) является возможность инициализации входов и состояний, т.е. задание начальных условий и время выборки.

1.5.2. Дополнительные линейные блоки

Состав дополнительных линейных блоков раздела *Additional Linear* показан на рис. 1.73.

Блоки раздела *Additional Linear* можно разделить на две категории: PID-контроллеры и блоки типа *State-Space*, *Transfer Fcn* и *Zero-Pole*, дополненные возможностями инициализации выходных сигналов и состояний.

PID-контроллер (*PID controller*) – это довольно универсальный блок, выходной сигнал которого задается операторным выражением:

$$\text{OUT} = P + I/s + D, \quad (1.7)$$

где P – входной сигнал, I – его интеграл и D – его производная. Параметр P (по умолчанию равен 1) фактически задает компоненту выходного сигнала, пропорциональную входному сигналу, параметр I задает пропорциональность интегралу входного сигнала и, наконец, параметр D задает пропорциональность производной входного сигнала. Варьируя

параметры P, I и D, можно задавать различный вид выходного сигнала, в том числе в виде интеграла или производной от входного.

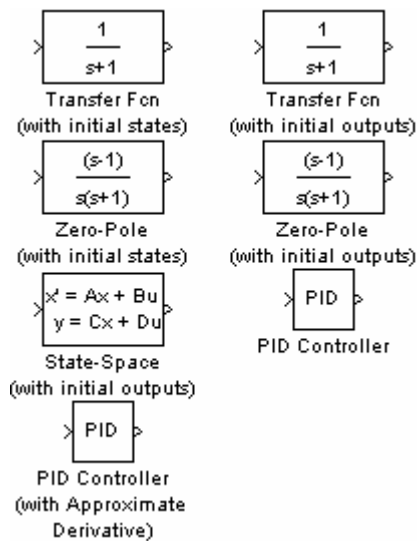


Рис. 1.73. Внешний вид блоков раздела Additional Linear

Другой блок – PID controller with Approximate Derivative (PID-контроллер с улучшенным вычислением производной) вычисляет выходной сигнал в соответствии с выражением (1.8)

$$\text{OUT} = 3 + I/s + Ds/(1/Ns+1). \quad (1.8)$$

За счет применения дополнительного параметра N улучшается вычисление производной.

Остальные блоки раздела Additional Linear библиотеки Simulink Extras повторяют по функциям уже описанные блоки раздела Continuous основной библиотеки Simulink (см. рис. 1.5, п. 1.3.1) и отличаются только возможностью инициализации выходных сигналов или состояний.

1.5.3. Дополнительные блоки Additional Sinks

Раздел дополнительных блоков Additional Sinks содержит ряд новых виртуальных регистраторов, графические «смотровые окна» (рис. 1.74).

В него входят следующие блоки:

- Power Spectral Density – анализатор спектральной плотности мощности;
- Averaging Power Spectral Density – анализатор спектральной плотности мощности с усреднением;
- Spectrum Analyser – анализатор спектра.
- Averaging Spectrum Analyzer – спектральный анализатор с усреднением;
- Cross-Correlator – кросскоррелятор (используется с пакетом Signal Processing Toolbox);

- Auto Correlator – автоматический коррелятор (используется с пакетом Signal Processing Toolbox). Служит для вычисления автокорреляционной функции сигналов;
- Floating Bar Point – утилита построения гистограмм.

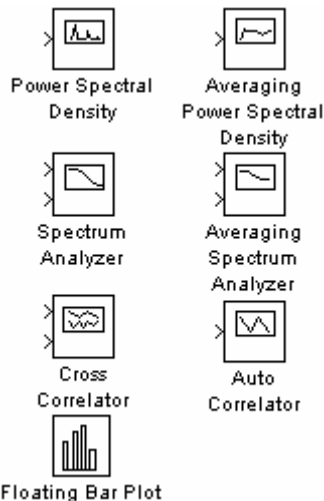


Рис. 1.74. Внешний вид блоков раздела Additional Sinks

Все блоки этого раздела относятся к двум важным разделам моделирования – статистическому анализу и анализу спектров сигналов (по уровню и по мощности).

1.5.3.1. Блоки спектрального анализа

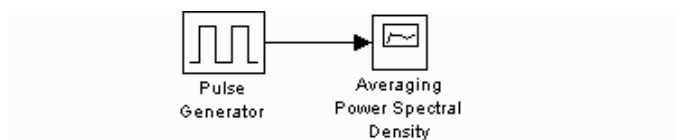
Блок Averaging Power Spectral Density служит для наглядного представления формы сигналов и их энергетического спектра (рис. 1.75). Спектр (в данном случае прямоугольных импульсов) представляется амплитудно-частотной характеристикой мощности с усреднением и фазочастотной характеристикой.

Блок имеет следующие параметры:

- Length of buffer – размер буфера;
- Number of point for fft – число точек для быстрого преобразования Фурье;
- Plot after how many points – построение графика после задания числа точек;
- Sample time – эталонное время, шаг моделирования.

На рис. 1.76 представлено окно анализатора спектра. Оно содержит три графика:

- график временной зависимости входного сигнала;
- график амплитудно-частотной характеристики мощности входного сигнала;
- график зависимости фазы от частоты.



Block Parameters: Averaging Power Spectral Density

Averaging Power Spectral Density (mask) (link)
Построить среднюю PSD входа.

Parameters

Длина буфера:
100

Число точек для FFT:
500

Построение после задания количества точек:
100

Время выборки:
0.1

OK Отмена Помощь Принять

Рис. 1.75. Пример применения блока Averaging Power Spectral Density и его параметры

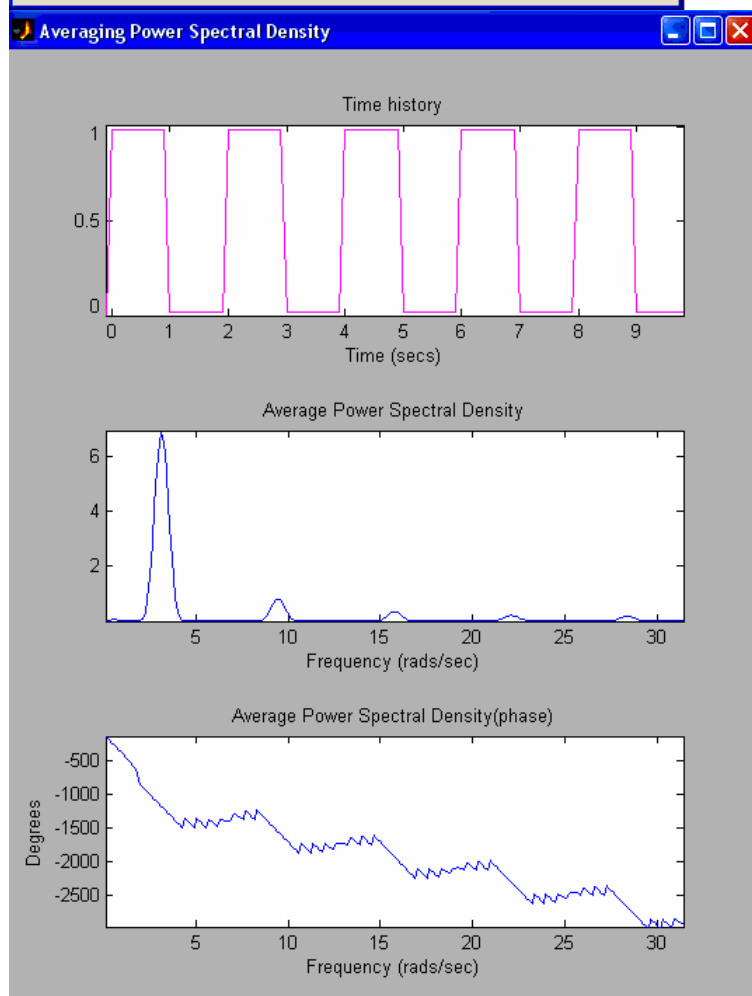


Рис. 1.76. Графики создаваемые блоком Averaging Power Spectral Density

Масштабы графиков линейные. Спектр вычисляется приближенно на основе быстрого преобразования Фурье (БПФ). Можно заметить, что

спектрограмма гораздо больше напоминает кривые, созданные реальными анализаторами спектра, чем получаемые теоретически. В частности, линии первой и высших (в данном случае нечетных) гармоник представлены колоколообразными кривыми, а не четкими вертикальными отрезками прямых. Как достоинство спектрограмм можно отметить широкий диапазон представления углов без характерных разрывов фазы, возникающих при ограничении изменения фазы, например в диапазоне углов от 0 до 360 градусов (от 0 до 2π радиан).

Остальные анализаторы спектра используются совершенно аналогично и имеют такое же окно представления результатов спектрального анализа. Их различия вытекают из наименования блоков.

1.5.3.2. Блок кросскоррелятора Cross-Correlator

Блок Cross-Correlator служит для построения графиков поданных на его вход сигналов и текущего коэффициента корреляции, отражающего степень взаимосвязи изменений сигналов. Напомним, что если оба сигнала меняются идентично, то коэффициент корреляции равен 1, а его значение 0 означает различное изменение сигналов.

Параметры этого блока имеют то же назначение, что и одноименные параметры блока Averaging Power Spectral Density.

1.5.4. Обзор раздела библиотеки Flip Flops

Раздел библиотеки Simulink Extras Flip Flops содержит генератор тактовых прямоугольных импульсов и 4 блока триггеров (рис. 1.77). Все они описаны ниже.

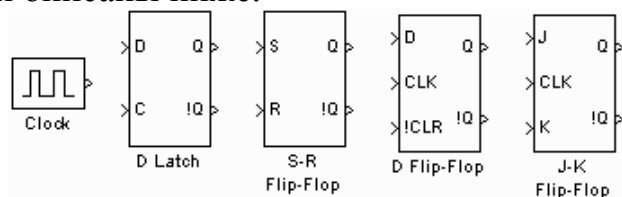


Рис. 1.77. Внешний вид блоков раздела Flip Flops

Раздел 1.5.4.1. Генератор тактовых импульсов Clock

Генератор Clock (Часы) является самым простым из источников прямоугольных импульсов. Он создает сигнал в виде импульсов единичной амплитуды при скважности равной 2. Скважность есть отношение периода импульсов к их длительности.

Единственным параметром этого источника импульсов является период сигнала (в тактах периода эталонного времени). Этот источник обычно используется для тактирования работы триггеров.

1.5.4.2. Триггерные блоки

Раздел библиотеки Flip Flops содержит 4 блока реализующих следующие типы стандартных триггерных устройств: D Flip Flop, D Latch, J-K Flip-Flops и S-R Flip-Flops (рис. 1.78, а, б). Блоки D-триггеров не имеют каких-либо устанавливаемых параметров.

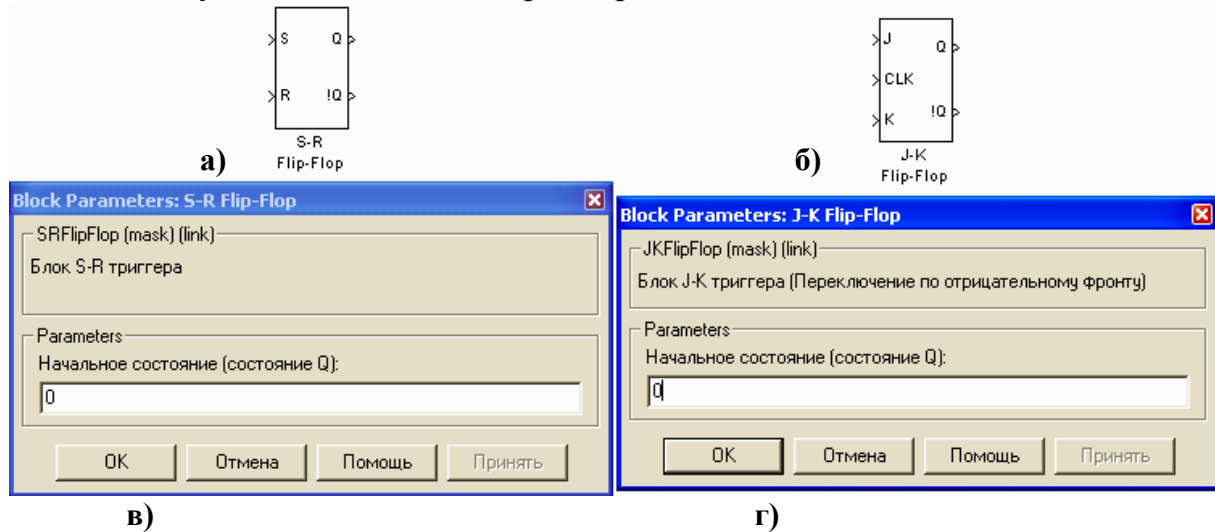


Рис. 1.78. Блоки S-R (а) и J-K-триггеров (б) и окна их параметров (в, г)

Для J-K и RS-триггеров в окне установки параметров (см. рис. 1.78 в, г) задается один параметр – начальное состояние для прямого выхода (Q).

1.5.5. Раздел Linearization

Раздел Linearization содержит всего два блока – дифференцирующий блок (Switched derivative for linearization) и блок заданной временной задержки (Switched transport delay for linearization) (рис. 1.79).

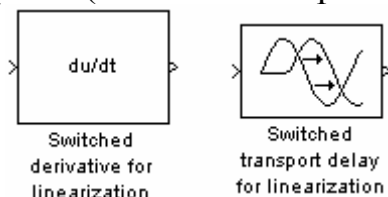


Рис. 1.79. Внешний вид блоков раздела Linearization

Блок дифференцирования (Switched derivative for linearization) имеет два параметра настройки:

- Switch value (Значение переключателя) – при установке Switch value = 1 включается линеаризация процесса дифференцирования, а установка Switch value = 0 отключает линеаризацию процесса дифференцирования;
- Derivative constant $Ns/(s+N)$ (Константа производной) – константа дифференцирования, она входит в знаменатель операторной передаточной функции блока.

Блок заданной временной задержки Switched transport delay for linearization представляет собой линейное устройство временной задержки.

Блок имеет следующие параметры:

- Time delay – длительность задержки;
- Initial input – начальное значение входного сигнала;
- Initial buffer size – начальная емкость буфера;
- Pade order (for linearization) – порядок Паде-линеаризации.

1.5.6. Блоки преобразований Transformations

Раздел преобразований Transformations содержит 8 блоков для осуществления типичных преобразований – это преобразование температуры, углов и систем координат.

Соответствующие формулы преобразования появляются в информационном поле окна этого раздела, а также в окнах параметров блоков.

Блок Polar to Cartesian служит для преобразования полярных координат точки в прямоугольные координаты на плоскости (рис. 1.80, а, в).

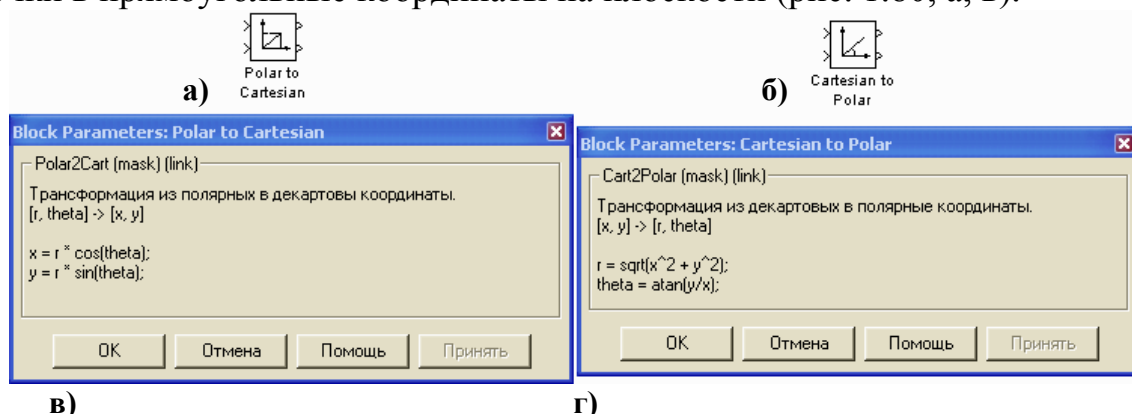


Рис. 1.80. Внешний вид блоков Polar to Cartesian (а), Cartesian to Polar (б) и их параметры (в, г)

Блок Cartesian to Polar служит для преобразования прямоугольных координат точки на плоскости в полярные координаты (см. рис. 1.80, б, г).

Блок Spherical to Cartesian служит для преобразования сферических координат точки пространства в прямоугольные координаты (рис. 1.81, а, в).

Блок Cartesian to Spherical служит для преобразования прямоугольных координат точки в пространстве в сферические координаты (см. рис. 1.81, б, г).

Блок Fahrenheit to Celsius переводит температуру из градусов Фаренгейта в градусы Цельсия (рис. 1.82, а, в).

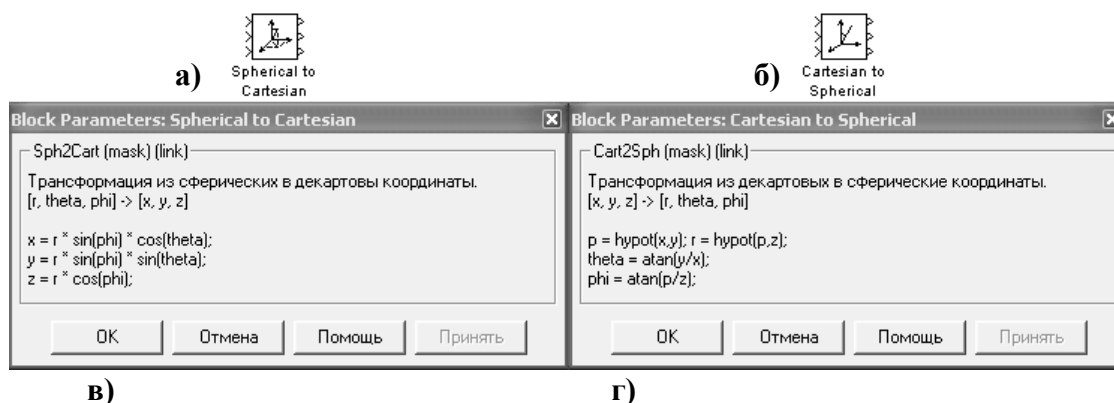


Рис. 1.81. Внешний вид блоков Spherical to Cartesian (а), Cartesian to Spherical (б) и их параметры (в, г)

Блок Celsius to Fahrenheit преобразует температуру, выраженную в градусах Цельсия, в температуру, выраженную в градусах Фаренгейта (см. рис. 1.82, б, г).

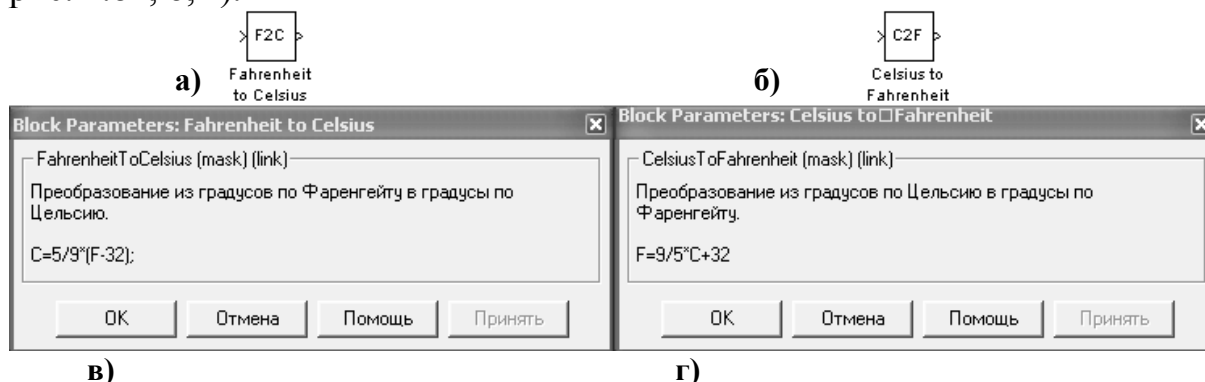


Рис. 1.82. Внешний вид блоков Fahrenheit to Celsius (а), Celsius to Fahrenheit (б) и их параметры (в, г)

Блок Degree to Radians служит для преобразования углов, выраженных в градусах, в углы, выраженные в радианах (рис. 1.83, а, в).

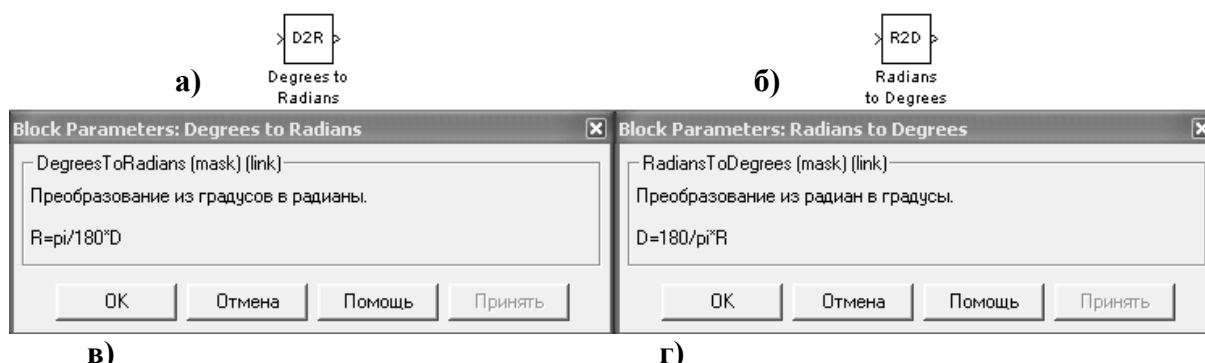


Рис. 1.83. Внешний вид блоков Degree to Radians (а), Radians to Degree (б) и их параметры (в, г)

Блок Radians to Degree служит для преобразования углов, выраженных в радианах, в углы, выраженные в градусах (см. рис. 1.83, б, г).

1.6. Запуск расчета модели и параметры расчета

Управлять процессом расчета модели можно с помощью кнопок, расположенных на панели инструментов модели или подсистемы, а также с помощью команд из раздела меню Simulation (рис. 1.84). Этот раздел доступен только из окна блок-диаграммы модели или подсистемы и заблокирован в окне библиотеки (Library).

Запустить расчет модели позволяет кнопка , расположенная на панели инструментов (или команда «Start», см. рис. 1.84). После запуска процесса моделирования данная кнопка преобразуется в кнопку – «пауза»  (или команда «Pause»¹) и становится активной расположенная рядом кнопка  (или команда «Stop»), которая позволяет завершить моделирование досрочно, то есть либо до истечения заданного интервала моделирования, либо до реализации предусмотренных условий окончания сеанса моделирования.

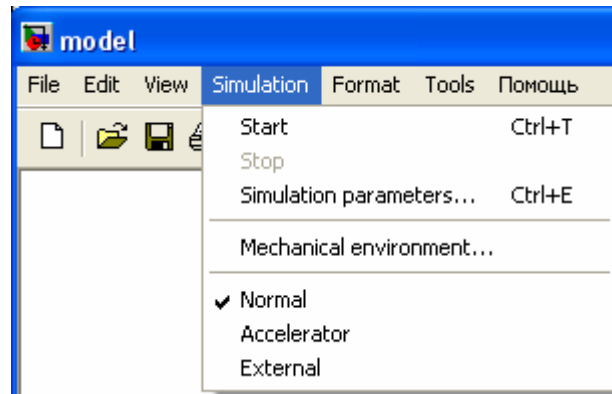


Рис. 1.84. Раздел Simulation меню пользователя

В нижней части окна моделирования при запуске модели можно увидеть индикаторы хода расчета – секундный и графический , также здесь отображается метод расчета, например, ode45 (рис. 1.85).

Раздел Simulation содержит команду «Simulation Parameters...» (см. рис. 1.84) – с ее помощью открывается диалоговое окно настроек параметров моделирования (см. рис. 1.85). Подробно остановимся только на вкладке Solver (Решатель), по остальной информации можно получить в [1 – 3] или во встроенной справочной системе MatLab.

¹ при прерывании моделирования с помощью команды Pause она заменяется альтернативной командой – Continue (Продолжить)

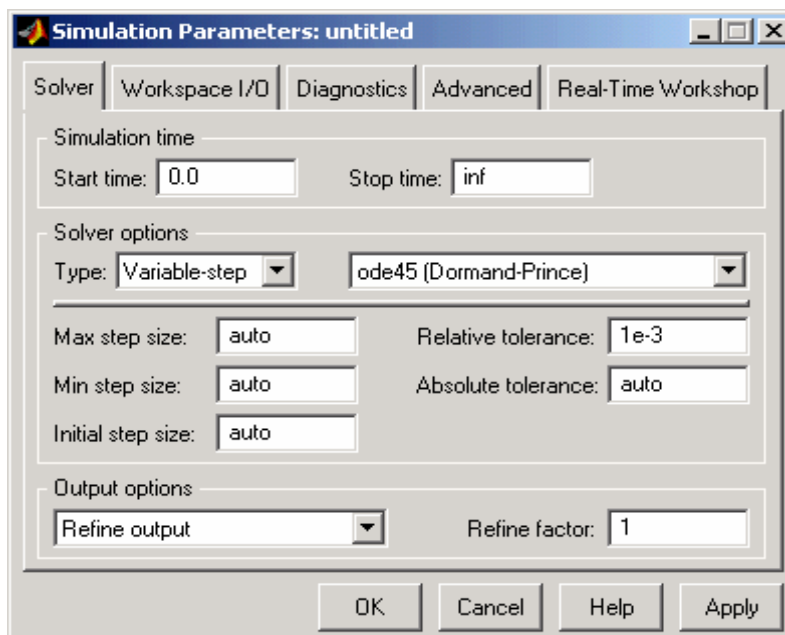


Рис. 1.85. Вкладка Solver диалогового окна установки параметров моделирования

На вкладке Solver могут быть произведены следующие установки (см. рис. 1.85):

Simulation time – (Интервал моделирования). Его величина задается посредством указания начального (Start time) и конечного (Stop time) значений модельного времени. Следует помнить, что реальное и модельное время не совпадают, последнее определяется сложностью модели и параметрами ПК на котором производится расчет модели;

Solver options – выбор метода расчета модели.

Первое меню – Type – позволяет выбрать способ изменения модельного времени, содержит два пункта:

- Variable-step – моделирование с переменным шагом, используется для непрерывных систем;
- Fixed-step – моделирование с фиксированным шагом – для дискретных систем.

Второе меню, расположенное справа от вкладки Type, позволяет выбрать метод расчета нового состояния системы. Первый пункт меню (discrete) обеспечивает расчет дискретных состояний системы (и для непрерывного, и для дискретного времени переходов из состояния в состояние).

Остальные 5 пунктов меню обеспечивают выбор метода расчета нового состояния для непрерывных систем. Эти методы различны для непрерывного времени (Variable-step) и для дискретного времени (Fixed-step), но основаны на единой методике – решении обыкновенных дифференциальных уравнений (ODE).

Подробное описание каждого из методов расчета состояний системы приведено во встроенной справочной системе MatLab (раздел matlab\funfun).

Под окном меню Type находится строка редактирования, название которой изменяется в зависимости от выбранного способа изменения модельного времени. Для Fixed-step она называется Fixed-step size и позволяет указывать величину шага моделирования.

При выборе Variable-step данная строка получает имя Max step size (Максимальная величина шага) и, соответственно, содержит предельное допустимое значение шага моделирования. В этом случае ниже также появляется строка Min step size (Минимальная величина шага).

По умолчанию величина шага моделирования для обоих способов изменения модельного времени устанавливается системой автоматически (auto) в соответствии с параметрами настройки блоков модели; требуемая величина шага может быть введена вместо значения auto либо в форме числовой константы, либо в виде вычисляемого выражения.

При моделировании непрерывных систем с использованием переменного шага (Variable step) необходимо указать точность вычислений: относительную (Relative tolerance) и абсолютную (Absolute tolerance). По умолчанию они равны соответственно -1×10^{-3} и 1×10^{-6} . Кроме того, для указанного класса систем можно задать начальное значение шага моделирования (в поле initial step size), он обязательно должен быть больше и равен Min step size, если не указано «auto».

При моделировании с переменным шагом можно задавать опции вывода (Options output) выходных параметров моделируемой системы. Соответствующие настройки выполняются с помощью двух элементов интерфейса: выпадающего меню и строки редактирования.

Меню, расположенное слева, содержит три пункта:

- Refine output (Улучшенный вывод или Очистить вывод) – при выборе этой опции можно изменять дискретность регистрации модельного времени и параметров модели (тех величин, которые сохраняются в рабочей области MatLab с помощью блока To Workspace). Установка дискретности выполняется в строке редактирования Refine factor, расположенной справа; по умолчанию значение Refine factor равно 1; это означает, что регистрация производится с шагом $\Delta t_M = 1$ (то есть для каждого очередного значения модельного времени: 0, 1, 2, 3, и т.д.); это наибольший возможный шаг регистрации. Значение Refine factor можно интерпретировать как относительную частоту регистрации: если, например Refine factor равен 2, это означает, что $\Delta t_M = 1/2$, если Refine factor равен 3, то $\Delta t_M = 1/3$ и т.д.; параметр Refine factor может быть только целым положительным числом.

- Produce additional output (создать дополнительный вывод) – обеспечивает дополнительную регистрацию параметров модели в заданные моменты времени; их значения вводятся в строке редактирования (в этом случае она называется Output times) в виде списка, заключенного в квадратные скобки; при использовании дополнительных моментов регистрации базовый шаг регистрации (Δt_M) равен 1. Значения времени в списке Output times могут быть дробными числами и иметь любую точность.

- Produce specified output only (формировать только заданный вывод) – устанавливает вывод параметров модели только в заданные моменты времени, которые указываются в списке Output times.

1.7. Построение блок-схем

Рассмотрим операции, с помощью которых можно формировать блок-схемы сложных динамических систем.

1.7.1. Выделение объектов

При создании и редактировании S-модели используются такие операции, как копирование или удаление блоков и линий. Прежде чем выполнять эти операции необходимо выделить (выбрать) объект.

Для выделения отдельного объекта нужно щелкнуть на нем один раз. В результате по углам выделенного блока или в начале и конце линии появятся маленькие черные квадратики. При этом снимается выделение со всех ранее выделенных объектов. Если щелкнуть на любой свободной области модели вне блока, все блоки и линии становятся не выделенными.

Чтобы выделить несколько объектов, следует, удерживая нажатой кнопку Shift, щелкнуть на каждом из них, затем отпустить кнопку Shift. Несколько объектов можно выделить также с помощью прямоугольной рамки. Для этого необходимо щелкнуть мышью в точке, которая будет служить углом рамки, а затем, удерживая кнопку мыши нажатой, протянуть указатель в направлении диагонали прямоугольника, т.е. обвести необходимые объекты. В результате вокруг выделяемых объектов должна появиться пунктирная рамка. Когда все нужные объекты будут охвачены рамкой, отпустить кнопку мыши.

Выделение всей модели, то есть всех объектов в активном окне блок-схемы, осуществляется одним из двух способов: путем вызова команды Edit → Select All (Правка → Выбрать все) в окне блок-схемы или посредством нажатия кнопок Ctrl+A.

1.7.2. Копирование блоков

В процессе создания и редактирования модели необходимо копировать блоки из библиотеки или из другой модели в текущую модель. Для этого достаточно открыть нужный раздел библиотеки или окно модели-прототипа и перетащить нужный блок с помощью мыши (нажать и не отпускать правую кнопку мыши) в окно создаваемой (редактируемой) модели. Скопированный блок получает те же значения настраиваемых параметров, что и блок-оригинал. Копировать блоки можно и другим способом. Для этого необходимо выполнить следующие действия:

- выделить блок, который нужно скопировать;
- на панели инструментов выбрать команду Edit → Copy (Правка → Копировать) (см. рис. 1.84) или нажать правую кнопку мыши (курсор предварительно навести на выделенный блок) и выбрать в раскрывшемся списке команду Copy (Копировать).
- сделать активным окно, в которое нужно скопировать блок;
- в этом окне на панели инструментов выбрать команду Edit → Paste (Редактирование → Вставить) или нажать на кнопку, также можно нажать правую кнопку мыши на свободном месте в окне модели и выбрать в раскрывшемся списке команду Paste (Вставить).

Чтобы создать копию блока внутри модели достаточно выполнить следующие действия:

- выделить нужный блок;
- нажать и не отпускать Ctrl + левая кнопка мыши;
- или нажать одну правую кнопку мыши, перенести блок на свободное место в окне модели (при перемещении блок обозначается пунктирной линией, справа от блока появляется знак «+»);
- или воспользоваться выше описанными командами и кнопками Copy и Paste.

Каждому из скопированных блоков автоматически присваивается имя. Первый из них будет иметь то же имя, что и блок в библиотеке, а каждый следующий блок такого типа – то же имя с добавлением порядкового номера. Пользователь может переименовать блок (см. далее).

1.7.3. Перемещение блоков в модели

Чтобы переместить блок внутри модели с одного места на другое, достаточно перетянуть его с помощью мыши, для этого необходимо: поместить курсор мыши на блок, нажать и не отпускать левую кнопку мыши, перенести блок. При этом автоматически будут перерисованы линии связей перемещенного блока с другими.

Также блок можно переместить кнопками $\uparrow$, $\downarrow$, $\leftarrow$, $\rightarrow$ расположенными на клавиатуре, предварительно блок следует выделить.

Для одновременного перемещения нескольких блоков вместе с соединительными линиями следует выделить блоки и линии рамкой, а затем перетянуть с помощью мыши один из выделенных блоков в нужное место. Остальные блоки займут новые места; все относительные расстояния между выделенными блоками и линиями при этом сохраняются.

1.7.4. Установка параметров блока

Параметры блока устанавливаются в диалоговом окне его настройки. Для того чтобы вызвать это окно, необходимо дважды щелкнуть на изображении блока в блок-схеме.

1.7.5. Удаление блоков

Чтобы удалить ненужные блоки из блок-схемы, нужно выделить эти блоки и нажать кнопку Delete или Backspace, расположенную на клавиатуре. Для удаления блоков применяется также команда Edit $\rightarrow$ Clear (Правка $\rightarrow$ Очистить) или Edit $\rightarrow$ Cut (Правка $\rightarrow$ Вырезать) (или кнопка ) окна блок-схемы. Если выбрать вторую команду, то в дальнейшем удаленные блоки можно будет скопировать обратно в модель, воспользовавшись командой Edit $\rightarrow$ Paste (Правка $\rightarrow$ Вставить) окна блок-схемы.

1.7.6. Отсоединение блока

Для отсоединения блока от линий связи следует поместить указатель на изображении блока, нажать кнопку Shift и, не отпуская ее, перетянуть блок в другое место.

1.7.7. Изменение ориентации блока

В обычном изображении блоков сигнал проходит сквозь блок слева направо (по левую сторону размещены входы блока, а по правую сторону – выходы). Для изменения ориентации блока, нужно выполнить два действия:

1. Выделить блок, который нужно повернуть;
2. Открыть меню Format (Формат) в окне блок-схемы и выбрать команду Flip Block (Перевернуть блок) или воспользоваться «горячими» кнопками «Ctrl+I» – чтобы повернуть блок на 180° или команду Rotate Block (Повернуть блок) или воспользоваться «горячими» кнопками «Ctrl+R» – для поворота блока по часовой стрелке на 90°.

1.7.8. Изменение размеров блока

Чтобы изменить размеры блока, необходимо выполнить следующие действия:

1. Выделить нужный блок;
2. Установить указатель мыши на одну из угловых меток блока. Возле этой метки должен появиться указатель в виде двунаправленной стрелки, расположенной под углом 45°;
3. Захватить эту метку мышью и перетянуть в новое положение. Противоположная метка блока останется неподвижной.

1.7.9. Изменение имен блоков и манипуляции с ними

Все имена блоков в модели должны быть уникальными и включать как минимум один символ. Если блок ориентирован слева направо, то по умолчанию имя располагается под ним, если блок ориентирован справа налево – выше блока, а если сверху вниз или снизу вверх – по правую сторону блока.

Чтобы изменить имя блока, следует щелкнуть на нем левой кнопкой мыши, а потом с клавиатуры ввести новое имя или отредактировать существующее.

Внимание! Рекомендуется писать имена блоков только латинскими буквами.

Для изменения шрифта необходимо выделить блок, нажав левую кнопку мыши, вызвать команду Format → Font (Формат → Шрифт) с панели инструментов в окне модели (или нажать правую кнопку мыши и в раскрывшемся списке выбрать аналогичную команду), затем в раскрывшемся списке выбрать нужный шрифт.

Изменить местоположение имени выделенного блока можно двумя способами: перетянуть имя на противоположную сторону с помощью мыши или воспользоваться командой Format → Flip Name (Формат → Развернуть имя) окна модели.

Удалить имя блока можно с помощью команды Format → Hide Name (Формат и Скрыть имя) окна модели. Чтобы потом восстановить отображение имени блока, следует воспользоваться командой Format → Show Name (Формат → Показать имя).

1.7.10. Проведение соединительных линий

Сигналы в модели передаются по линиям. Каждая линия может передавать либо скалярный, либо векторный сигнал. Линия соединяет выходной порт одного блока с входным портом другого или с входными портами нескольких блоков через разветвление линии.

1.7.10.1. Создание линии между блоками

Для соединения выходного порта одного блока со входным портом другого следует выполнить следующие действия:

1. Установить указатель мыши в область выходного порта первого блока. При этом указатель принимает вид крестика;
2. Удерживая нажатой левую кнопку мыши, переместить указатель ко входному порту второго блока;
3. Отпустить кнопку мыши. В результате вместо символов портов появится соединительная линия, стрелка на конце которой будет указывать направление передачи сигнала. Если между блоками появилась связь, стрелка будет сплошной и черного цвета, если связи нет – пунктирной и красного цвета.

Линии можно проводить как от выходного порта ко входному, так и наоборот. Описанным выше способом можно создать соединительные линии, включающие лишь горизонтальные и вертикальные сегменты.

1.7.10.2. Создание разветвления линии

Ответвляющаяся линия начинается от существующей и передает ее сигнал ко входному порту другого блока. Таким образом, разветвление линий дает возможность организовать передачу одного сигнала нескольким блокам.

Для создания ответвления от линии нужно выполнить следующие действия:

1. Щелкнуть в той точке линии, от которой должна ответвляться другая линия.
2. Нажав и удерживая правую кнопку мыши, провести линию ко входному порту нужного блока новую линию связи, отпустить кнопку мыши.

1.7.10.3. Создание сегмента линии

Линии могут быть нарисованы по сегментам, причем сегменты соединены под прямыми углами или углом 45^0 . В этом случае указатель устанавливается в конец сегмента и с помощью мыши рисуется следующий сегмент (аналогично п. 1.7.10.2).

1.7.10.4. Перемещение сегмента линии

Для перемещения отдельного сегмента линии необходимо выполнить следующие действия:

1. Установить указатель на сегмент, который нужно передвинуть;

2. Нажать и удерживать левую кнопку мыши; при этом указатель должен принять вид четырехнаправленной стрелки;

3. Передвинуть указатель к новому положению сегмента и отпустить кнопку мыши. Сегмент, который соединен с портом блока, передвинуть нельзя. Только если переместить сам блок.

1.7.10.5. Разделение линии на сегменты

Для того чтобы разделить линию на два сегмента, необходимо выполнить следующие действия:

1. Выделить линию и установить указатель мыши в ту ее точку, в которой линия должна быть разделена на два сегмента;

2. Удерживая нажатой кнопку Shift, нажать и удерживать левую кнопку мыши. Указатель при этом примет вид кружка, и на линии образуется излом;

3. Передвинуть указатель в новое положение и отпустить кнопку мыши и кнопку Shift.

Для того чтобы передвинуть излом линии, достаточно на блок-схеме перетянуть точку излома в новое положение.

1.7.11. Метки сигналов и комментарии

Для наглядности оформления блок-схемы и удобства пользования ею, линии можно снабжать метками протекающих по ним сигналов. Метка размещается выше или ниже горизонтальной линии либо слева или справа от вертикальной линии. Метка может быть расположена в начале, в конце линии или в любой другой ее точке.

1.7.11.1. Создание меток сигналов и манипулирование ими

Для того чтобы создать метку сигнала, нужно дважды щелкнуть левой кнопкой мыши на сегменте линии и ввести текст метки (латинскими буквами!). При выполнении щелчка следует соблюдать точность, иначе вместо метки будет создан комментарий к модели.

Для проверки точности: первый щелчок левой кнопкой мыши выделяет линию (появляются черные квадратики на ее концах), второй щелчок создает поле для ввода текста метки линии.

Для перемещения метки нужно просто перетянуть ее с помощью мыши (нажав левую кнопку мыши) в новое место.

Нажатая правая кнопка мыши копирует метку. Также скопировать метку можно нажав и, удерживая кнопку Ctrl, перетянуть метку в новое положение на линии или другой сегмент линии, на котором нужно установить копию метки.

Если необходимо отредактировать текст метки, надо щелкнуть на ней левой кнопкой мыши и потом произвести соответствующие изменения, как в обычном текстовом редакторе. Чтобы удалить метку, нажмите и удерживайте кнопку Shift, выделите метку и затем воспользуйтесь кнопкой Delete или Backspace.

1.7.11.2. Распространение меток сигналов

Распространение меток сигналов – это процесс автоматического переноса имени метки на сегменты одной линии, разорванные блоками From/Goto и Mux.

Для того чтобы распространить метки, создайте во втором и следующих сегментах линии метки с именем `< >`. После повторного запуска модели или выполнения команды Edit → Update Diagram (Правка → Обновить схему) в окне модели или нажатия кнопки Ctrl+D в указанных сегментах автоматически будут проставлены метки (рис. 1.86).

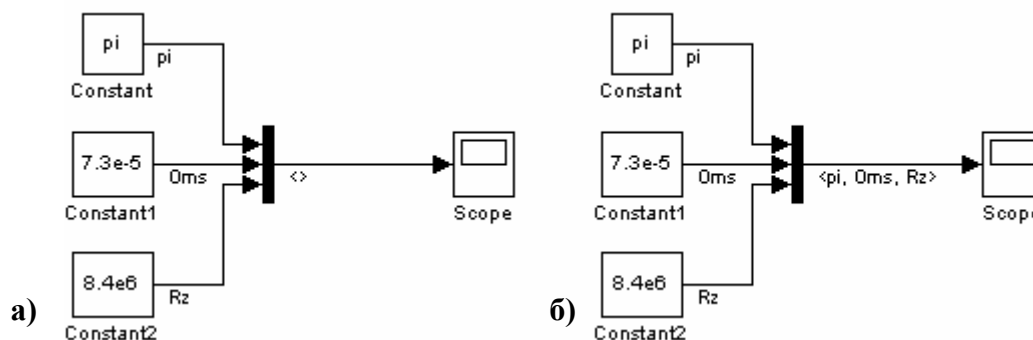


Рис. 1.86. Пример распространения меток: подготовка (а) и результат (б)

1.7.11.3. Комментарии к блок-схемам

Комментарии дают возможность снабжать блок-схемы текстовой информацией о модели и отдельных ее составляющих. Комментарии можно размещать в любом свободном месте блок-схемы.

Для создания комментария дважды щелкните в любом свободном месте блок-схемы, а потом введите комментарий в прямоугольную рамку (латинскими буквами!).

Для перемещения комментария в другое место следует перетянуть его на это место с помощью мыши, нажав левую кнопку мыши.

Чтобы скопировать комментарий, нужно нажать кнопку Ctrl и, удерживая ее, перетянуть комментарий в новое место.

Для редактирования комментария следует щелкнуть на нем левой кнопкой мыши, а потом внести нужные изменения, как в обычном текстовом редакторе. Чтобы изменить при этом шрифт, его размер или стиль, следует выделить фрагмент текста комментария, подлежащий

изменению, вызвать команду Format → Font (Формат → Шрифт) в окне блок-схемы, выбрать в появившемся диалоговом окне название шрифта, его размер, стиль и т.п., а затем щелкнуть на кнопке ОК.

Удалить комментарий можно выделив его с помощью рамки и нажать кнопку Delete или Backspace.

1.7.12. Сохранение и вывод на печать блок-схемы S-модели

Для записи модели (блок-схемы) на диск нужно вызвать команду Save (Сохранить) или Save As (Сохранить как) из меню File (Файл) окна модели. Simulink записывает в указанную директорию файл с заданным (введенным с клавиатуры) именем, присваивая ему расширение .mdl.

В названии моделей рекомендуется использовать только латинские буквы.

Чтобы распечатать модель (блок-схему) на принтере, необходимо воспользоваться командой File → Print (Файл → Печать) окна модели.

Довольно интересной является возможность вставки блок-схемы в документ текстового редактора Word. Для этого следует использовать команду Copy Model to clipboard (Копировать модель в буфер) меню Edit (Правка) окна модели, которая помещает в буфер обмена содержимое окна модели. Если после этого перейти в окно текстового редактора и нажать кнопки Ctrl+V, в открытом документе редактора появится изображение блок-схемы модели.

1.7.13. Создание подсистем

Сложную блок-схему модели можно существенно упростить путем группировки блоков. Последние можно объединить в подсистемы. Использование подсистем дает следующие преимущества: сокращается количество блоков, которые выводятся в окне модели, а также появляется возможность объединять в одну группу (подсистему) функционально связанные блоки и создавать иерархические блок-схемы.

Подсистему можно создать двумя способами: добавив блок *Subsystem* (см. п. 1.3.8) или сгруппировав существующие блоки.

Чтобы создать подсистему первым способом, необходимо выполнить следующие действия:

1. Перетащить блок Subsystem из раздела Ports & Systems (см. п. 1.3.8) в окно модели;
2. Открыть окно блок-схемы блока Subsystem, дважды щелкнув на его изображении в блок-схеме S-модели;
3. В пустом окне блок-схемы подсистемы создать подсистему, используя блоки In и Out для представления ее входов и выходов.

Если блок-схема уже содержит блоки, которые нужно объединить в подсистему, то последнюю можно образовать таким образом:

1. Выделить с помощью рамки блоки и соединяющие их линии, которые нужно включить в состав подсистемы;

2. Вызвать команду Edit → Create Subsystem (Правка → Создать подсистему) в окне модели или нажав правую кнопку мыши и в раскрывшемся списке выбрать Create Subsystem (Создать подсистему). Все выделенные блоки будут заменены одним блоком Subsystem.

Внимание! Для операции Create Subsystem недоступна команда Undo (Отмена последней операции). Поэтому перед созданием подсистемы, рекомендуется сохранить исходную модель под другим именем. Имена подсистем также рекомендуется давать латинскими буквами!

Чтобы вызвать подсистему для просмотра или модификации, достаточно навести на нее курсор мыши и дважды щелкнуть левой кнопкой, появится окно созданной подсистемы.

При создании подсистемы, любым из описанных выше способов, состав блоков и соединений в подсистеме остается неизменным, но дополнительно появляются новые блоки – порт ввода In и порт вывода Out (см. п. 1.3.8).

В любой из подсистем доступны все кнопки, что и в основном окне: пуск, стоп, вызов библиотеки и др.

Для сложных моделей, содержащих несколько подсистем, переключение между окнами удобно осуществлять через браузер (рис. 1.87).

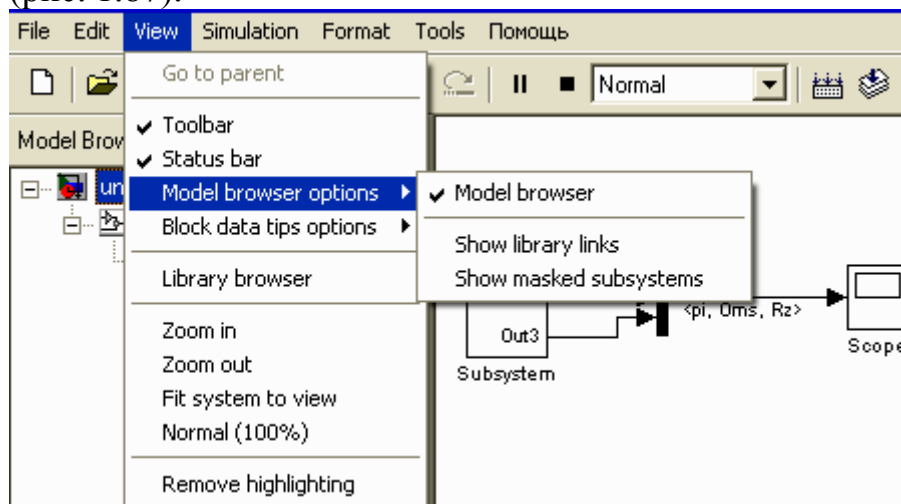


Рис. 1.87.
Пример работы с браузером модели

Доступ к нему осуществляется с панели инструментов View (Вид) → Model browser options (Опции браузера модели) → Model browser (Браузер модели). Внешний вид окна модели с подсистемами при подключенной опции представляет собой древовидную структуру (см. рис. 1.87).

1.7.13.1. Задание свойств подсистемы

Подсистема имеет окно свойств. Его можно вызвать для просмотра и редактирования с помощью команды меню File → Model Properties окна модели или с помощью контекстного меню правой кнопкой мыши.

Окно свойств подсистемы имеет три вкладки:

- Model Properties – свойства подсистемы – отражается версия подсистемы и дата ее создания. Другие поля вначале пусты. В поле Creator обычно вводится фамилия и инициалы создателя подсистемы, а в поле Model description задается ее описание. После сохранения подсистемы (командой меню File → Save в окне подсистемы) эта информация будет выводиться при вызове окна свойств подсистемы;

- Options – опции, задающие формат свойств;
- History – данные об истории подсистемы.

Обратите внимание, что окно свойств имеет один и тот же вид для всей системы и для подсистемы. Поэтому следует осторожно относиться к заданию параметров в этом окне – они могут относиться к системе в целом.

1.7.13.2. Маскирование подсистем

Для построения пользовательских блоков Simulink предлагает специальный механизм маскирования подсистем. Маскированные подсистемы это такие подсистемы, которые имеют специальный признак (маску), скрывающий их внутреннюю, порой достаточно сложную, структуру. В результате такая подсистема в деталях не видна и воспринимается как библиотечный модуль. Маскированные подсистемы обладают рядом важных достоинств:

- они имеют свои пиктограммы с уникальными изображениями;
- их можно использовать как библиотечные блоки;
- у них есть свое окно установки параметров;
- есть возможность в любой момент сбросить маску и наблюдать структуру блока;
- применение масок расширяет возможности построения сложных моделей;
- имеется возможность легко отредактировать подсистему, превращенную в маскированную;
- повышается наглядность моделей-диаграмм;
- повышается защищенность подсистемы от модификации, в том числе преднамеренной.

Для создания маскированных подсистем необходимо выполнить следующие операции:

- разработать и протестировать модель с соответствующими блоками;
- выделить часть блоков и оформить их в виде подсистемы (см. п. 1.7.13);
- задать подсистеме статус маскированной подсистемы;
- с помощью специального редактора масок создать окно установки параметров, определить средства инициализации подсистемы, создать документацию под маскированную подсистему и ее справочную систему;
- выполнить (если необходимо) тестирование основной модели с ее маскированными подсистемами и прочими блоками;
- выполнить (если необходимо) редактирование созданной маскированной подсистемы;
- демаскировать (если необходимо) подсистему.

Большая часть этих операций выполняется с помощью редактора маскированных подсистем, или просто масок.

Создание начальной модели

Рассмотрим следующий пример. Требуется создать модель, выполняющую функцию вычисления значения $y = ax^2 + b$, где a и b – константы. Для возведения x в квадрат достаточно использовать блок умножения Product, на оба входа которого подан сигнал x . Подключив к выходу этого блока масштабирующий блок Gain с коэффициентом передачи a , получаем сигнал ax^2 . К этим двум блокам необходимо добавить сумматор, на один вход которого поступает сигнал с блоков Product+Gain, а на другой выход константы b . В итоге на его выходе будет сигнал ax^2+b , что и требовалось рис. 1.88.

На рис. 1.88 показаны осциллограммы входного синусоидального сигнала и выходного сигнала при следующих параметрах: коэффициент передачи блока Gain равен 1 и константа b равна 0. После такого преобразования входного сигнала, выходной сигнал очень напоминает синусоиду, но удвоенной частоты и с добавлением постоянной составляющей.

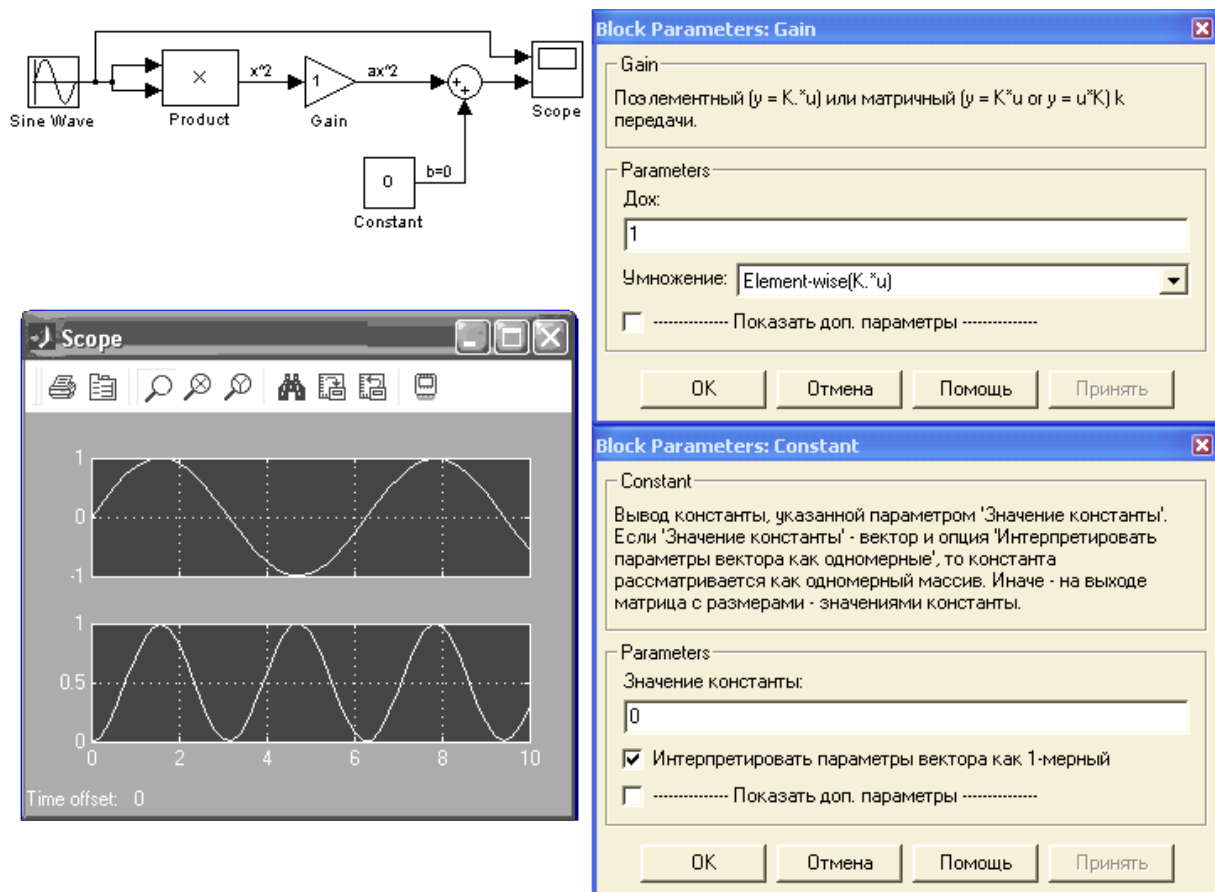


Рис. 1.88. Внешний вид исходной модели

Это говорит о том, что с помощью подобного устройства можно создать широкодиапазонный удвоитель частоты.

Подготовка к маскированию подсистемы

Если объединить в один блок источник синусоидального сигнала и осциллограф, получим функционально законченный блок, назовем его квадратором (Quadrator).

Прежде всего, необходимо перейти от конкретных параметров блоков к обобщенным. Для этого в параметрах блоков Gain и Constant следует задать коэффициент передачи масштабирующего блока Gain равным не 1, а переменной «а», которая впоследствии будет принимать любые значения. Вместо константы 0 у блока Constant – значение «b».

Кроме того, необходимо выделить отведенные под подсистему блоки и выполнив команду меню Edit → Create Subsystem, создать на базе выделенных блоков подсистему (см. п. 1.7.13). Если после этого выполнить двойной щелчок левой кнопкой мыши на блоке подсистемы, то появится окно с графической моделью подсистемы (рис. 1.89).

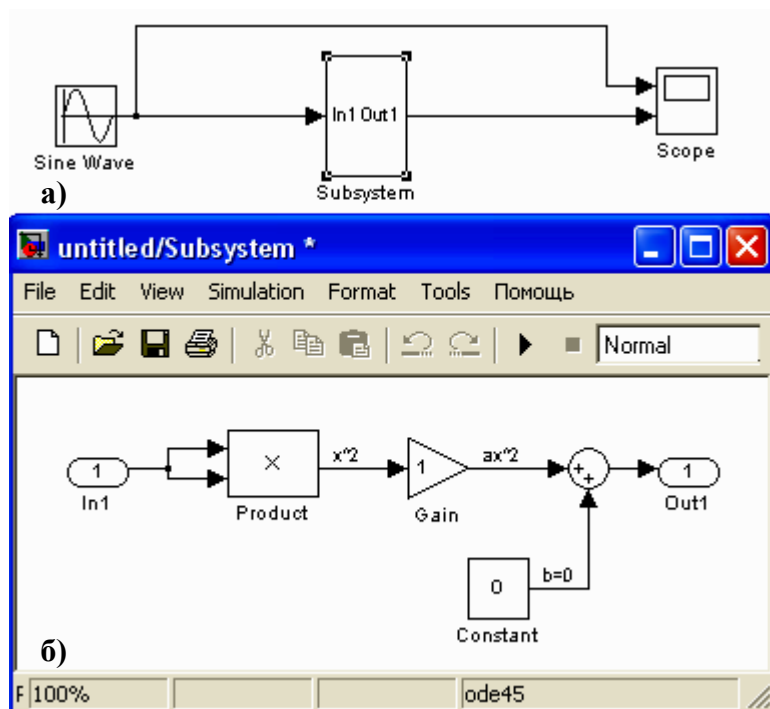


Рис. 1.89.
Преобразованная
модель (а) с
подсистемой (б)

После маскирования подсистемы графическая модель уже не будет выводиться. Двойной щелчок мышью на блоке будет выводить окно параметров блока. А для вывода графической модели используется команда Look under mask... в панели инструментов Edit меню окна Simulink.

Запуск редактора маски

Для создания маски необходимо выделить нужную подсистему, установив на ней курсор мыши и щелкнув ее левой кнопкой мыши, после чего из раскрывшегося списка выбрать команду Edit Mask... или аналогичную команду из меню Edit (рис. 1.90, а). Данная команда запускает редактор маски – появляется его окно с открытой незаполненной вкладкой Icon (Значок) (см. рис. 1.90). Редактор маски имеет четыре вкладки:

- Icon – подготовка пиктограммы (значка) блока;
- Parameters – подготовка окна параметров блока;
- Initialization – подготовка средства инициализации;
- Documentation – подготовка документации по блоку.

Внизу окна имеются общие для всех вкладок пять кнопок:

- OK – завершить установки и создать маску;
- Cancel – отказаться от создания маски и закрыть окно редактора;
- Unmask – снять маску с выделенного блока;
- Help – открыть окно справки по редактору маски;

- Apply – применить заданные установки не выходя из редактора маски.

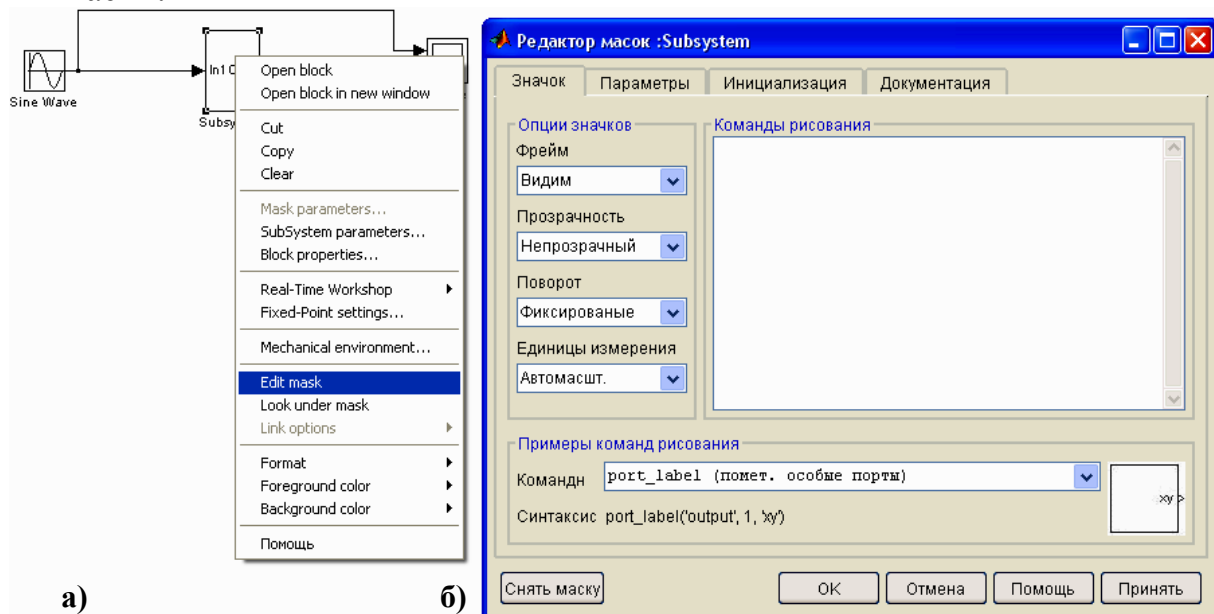


Рис. 1.90. Подготовка к запуску редактора маски

Следует помнить, что после создания маски команда Undo (Отмена последнего действия) не работает, так что в процессе задания параметров маски не стоит нажимать кнопку ОК. Делать это рекомендуется только когда все данные для маскируемой системы введены. В ходе ввода данных целесообразно пользоваться кнопкой Apply. Она вводит заданные данные и некоторые из них тут же отражаются на виде пиктограммы маски.

Создание окна параметров блока

Вторая вкладка Parameters – окно параметров блока может содержать параметры блока, а также флажки и списки. Для блока Quadrator достаточно создать окно с двумя параметрами – а и b. Редактор маски позволяет задать до полутора десятков параметров и элементов управления, причем разного вида (рис. 1.91).

При первом открытии вкладка Parameters содержит пустой список Dialog parameters (Параметры диалога):

- Prompt (Подсказка) – имена используемых в маске блоков;
- Variable (Переменные) – имена переменных, несущих значение соответствующих параметров;
- Type – типы параметров (выбираются из раскрывающегося списка: Edit – редактируемое поле ввода, Checkbox – поле для флажка, Popup – раскрывающийся список);
- Evaluate (Вычислить) – признак исполнения;
- Tunable (Настраиваемость) – признак возможности изменения.

Для создания этого списка служат четыре кнопки вкладки, расположенные слева:

- Add  – добавить в список новую позицию;
- Delete  – стереть выделенную позицию списка;
- Up  – поднять выделенную позицию списка вверх;
- Down  – опустить выделенную позицию списка вниз.

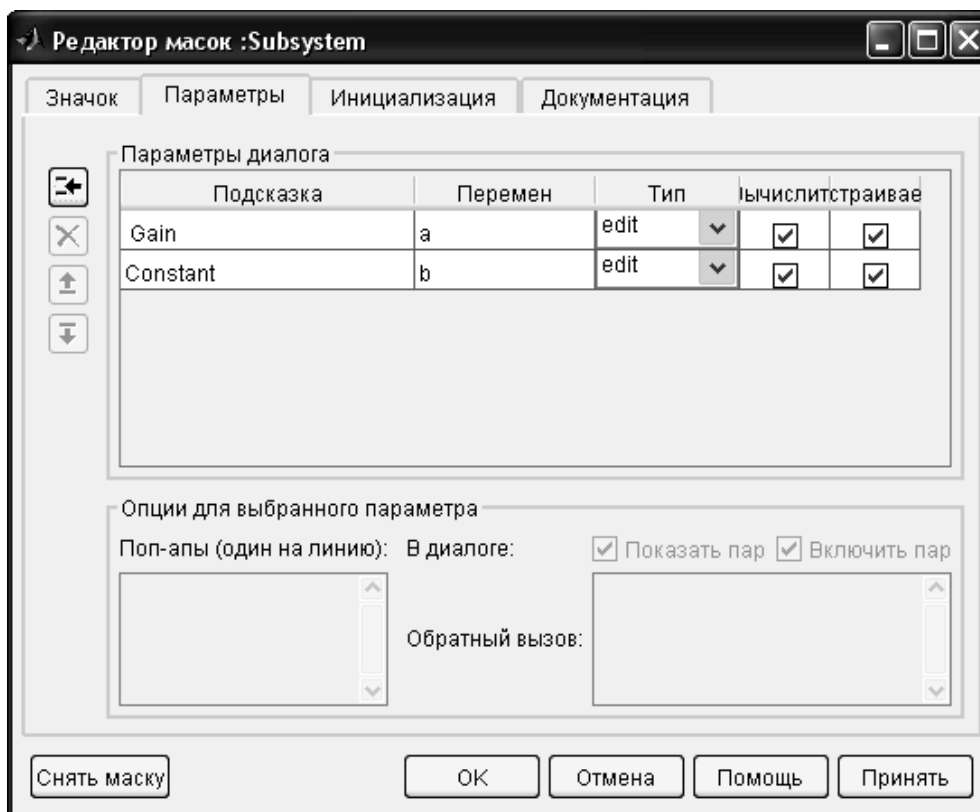


Рис. 1.91. Подготовка данных для окна задания параметров блока

В нашем случае, нажав кнопку Add, следует задать сначала имя блока Gain и переменную a, а затем имя блока Constant и переменную b. Можно использовать предложенные по умолчанию признаки исполнения и возможности изменения переменных. Для завершения этого процесса достаточно нажать кнопку Apply (Применить).

Инициализация параметров

Вкладка Initialization используется для задания средств инициализации блока. Эти средства действуют только внутри максимизированной подсистемы. Если были определены переменные параметры, то их список появляется в поле Dialog Variables (см. рис. 1.92).

В поле Initialization Commands (Команды инициализации) можно записать любые выражения на языке системы MatLab, в том числе содержащие условные выражения и переключатели. Это может потребоваться, например, для изменения вида пиктограммы создаваемого блока в зависимости от набора его параметров. После ввода нужных выражений следует зафиксировать их нажатием кнопки Apply (Применить).

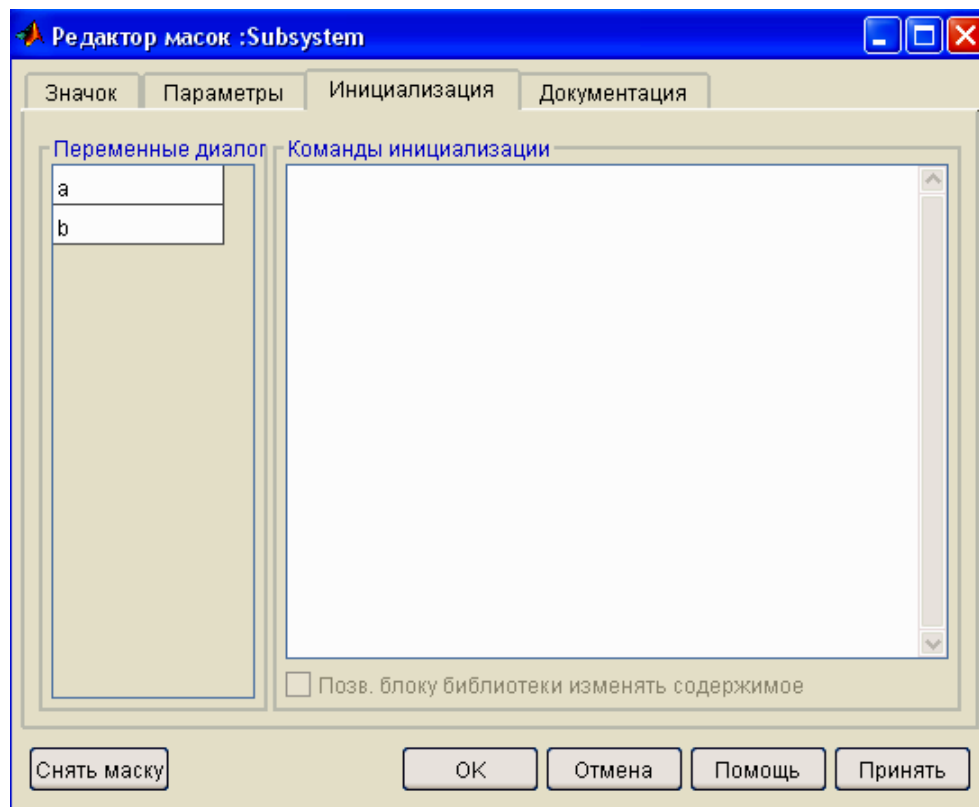


Рис. 1.92. Работа с вкладкой *Initialization*

Команды инициализации исполняются при открытии окна модели, при запуске модели на исполнение, выполнении команды Update diagram позиции Edit меню, при вращении или динамическом изменении пиктограммы.

Подготовка описания и документации блока

Библиотечные блоки Simulink имеют два основных типа описания:

- описание блока, размещенное вверху окна установки параметров;
- справочное описание блока, размещаемое в справочной системе.

Вкладка Documentation позволяет создавать описания обоих типов. Пример описания для создаваемого нами блока представлен на рис. 1.93.

Поле Block Description (Описание маски) служит для ввода текстового описания блока, которое будет размещено вверху окна параметров блока. Его размером не стоит злоупотреблять.

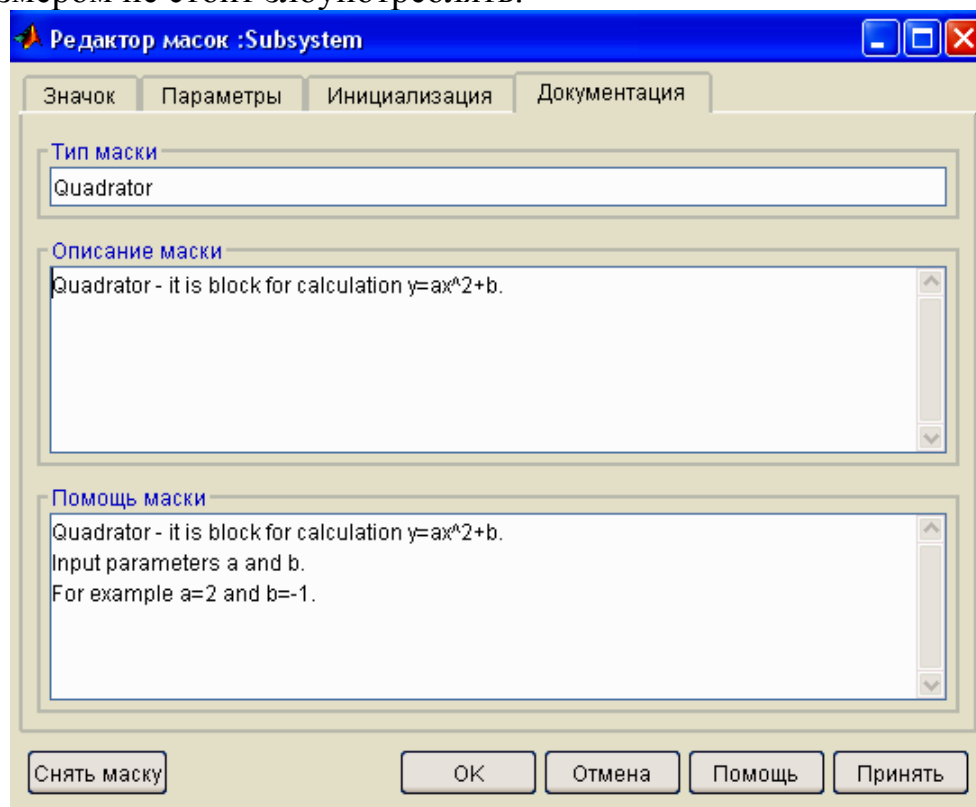


Рис. 1.93. Подготовка описания и справки по блоку

Поле Block Help (Помощь маски) служит для ввода текстовой части справки, которая будет размещена в справочной системе Simulink. Справка должна быть достаточно подробной.

Таким образом, реализуются единообразные правила описания блоков, созданных пользователем, и стандартных библиотечных блоков. Завершив ввод описаний, следует нажать кнопку Apply (Применить) в окне редактора масок.

Создание простой пиктограммы блока

Окно редактора маски появляется с открытой вкладкой Icon (Значок). Данная вкладка позволяет задать внешний вид пиктограммы блока (или его значка). Рис. 1.94 поясняет создание простейшей пиктограммы, которая представляет собой прямоугольник с надписью внутри, состоящей из двух строк: «Quadrator» и « $y = ax^2 + b$ ».

Вкладка Icon (Значок) содержит поле Drawing command (Команды рисования), в котором задаются текстовые и графические команды создания пиктограммы. Они задаются по правилам языка

программирования MatLab. Так, для создания указанной пиктограммы использована команда вывода текста посередине графического окна «disp('Текст')». Чтобы разбить строку текста (в нашем случае «Quadrator» и « $y = ax^2 + b$ ») на две строки, использован специальный символ перевода строки «\n».

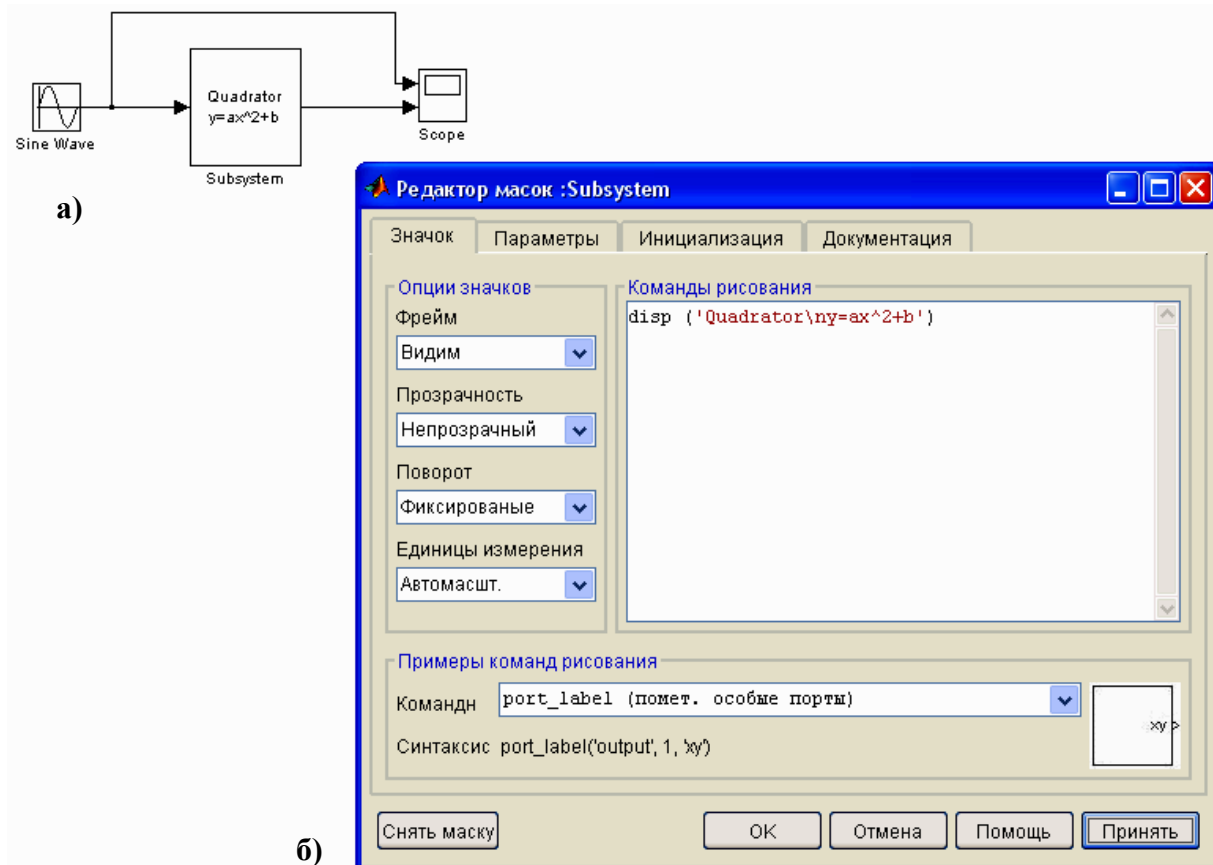


Рис. 1.94. Создание простой пиктограммы

Нажав кнопку Apply, можно наблюдать превращение блока Subsystem (см. рис. 1.89, а) в блок с новой пиктограммой (см. рис. 1.94, а). Пока наш блок имеет снизу название Subsystem.

Нам осталось завершить процесс создания маски нажатием кнопки ОК. На практике перед этим рекомендуется еще раз просмотреть все установки на всех вкладках и скорректировать замеченные неточности, если таковые есть. Нажав кнопку ОК, мы фиксируем создание маскированной подсистемы.

Проверка модели с созданной маской

Если выполнить двойной щелчок мыши на созданной маскированной подсистеме (маске) (рис. 1.95, а), то появится окно параметров маски (см. рис. 1.95, б).

Запустив модель кнопкой пуска моделирования, можно наблюдать работу модели и, в частности, появление осциллограмм (см. рис. 1.95, в). Сравнение их с осциллограммами исходной модели (см. рис. 1.88) показывает их полную идентичность, что свидетельствует о том, что созданная нами маскированная подсистема работает так, как это было задумано.

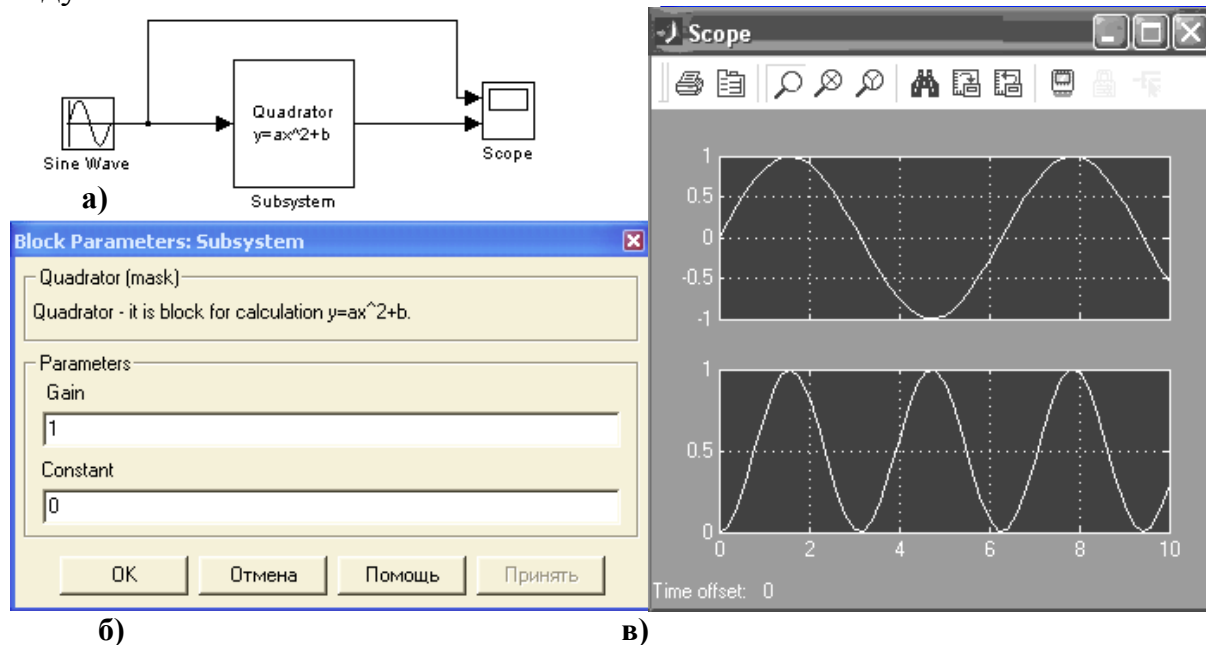


Рис. 1.95. Работа с моделью, имеющей маскированную подсистему

Вывод описания и справки маски

В верхней части окна параметров созданной маскированной подсистемы можно увидеть ранее созданное описание блока – см. рис. 1.95, б. Если нажать мышью кнопку Help в этом окне, то появится раздел стандартной справочной системы Simulink с текстом также ранее введенной справки (см. рис. 1.96, б).

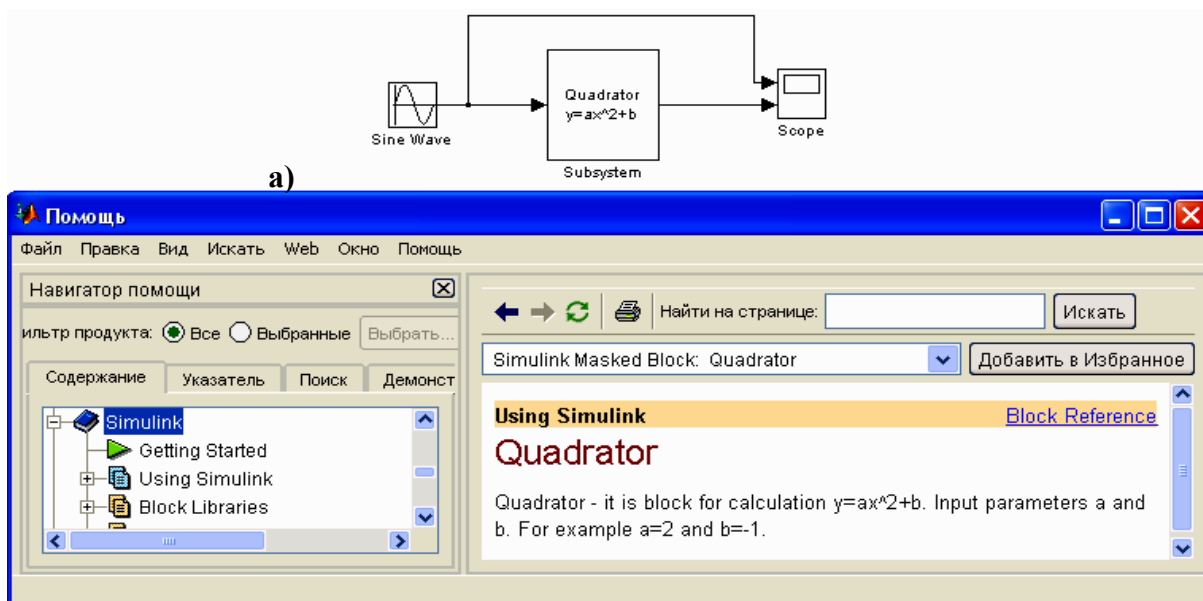


Рис. 1.96. Внешний вид (а) и пример вывода описания и справки по маскированной подсистеме (б)

Таким образом, созданная маскированная подсистема приобрела все атрибуты библиотечного блока. Однако в то время как библиотечные модули защищены от модернизации, маскированные модули остаются доступными для редактирования. Более того, их легко можно демаскировать.

Создание маски-справки

На диаграммах моделей часто можно увидеть блоки со справочными данными. Их активизация открывает окно с текстом справки или описанием диаграммы. Вы также можете создать такие блоки в виде масок. Они могут иметь пиктограмму в виде вопросительного знака или текста с пояснением правил работы с такой маской.

Обычно двойной щелчок мышью на пиктограмме такого блока открывает его окно и выводит текст справки. Можно также задать специальную функцию открытия, описав ее в поле Open function окна свойств блока Block Properties. Однако эта возможность используется довольно редко.

2. ПРИМЕР ПОСТРОЕНИЯ МОДЕЛИ В SIMULINK

Из разделов библиотеки Simulink создадим модель, описывающую поведение интегрирующего развортывающего преобразователя (РП) (рис. 2.1). Для этого поместите курсор на иконку необходимого блока, нажмите левую кнопку мыши и, не отпуская ее, «перетащите» изображение блока в поле созданной модели (см. рис. 2.1). Отпустите кнопку мыши.

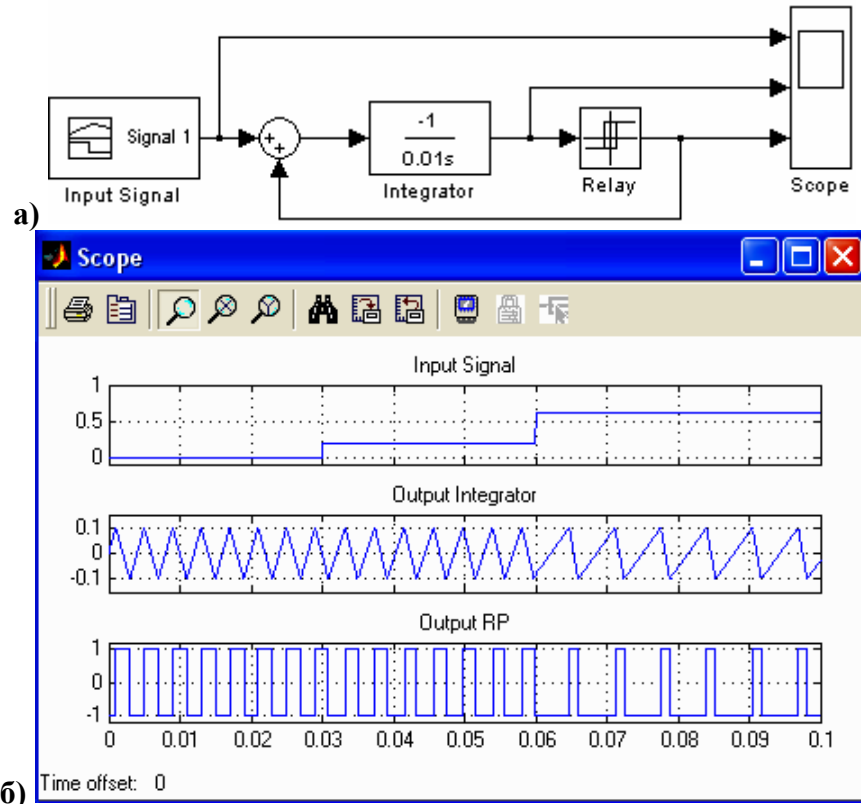


Рис. 2.1. Блок-диаграмма модели РП (а) и осциллограммы (б) поясняющие ее работу

На следующем шаге добавляют в модель элементы, соответствующие остальным элементам РП. Если в структуре встречается вторично аналогичный блок (например, блок Signal Builder из раздела библиотеки Sources), есть три пути его размещения в блок-диаграмме:

- повторить еще раз операцию перемещения блока из раздела библиотеки;
- воспользоваться командами Copy (Копировать) и Paste (Вставить) из раздела меню Edit;
- использовать дополнительные сервисные возможности Simulink.

Пойдем по третьему пути. Поместите курсор на иконку блока Signal Builder в блок-диаграмме создаваемой модели и нажмите левую кнопку

мышь. Справа от курсора появится белый крестик. Не отпуская левую кнопку мыши, переместите курсор в ту точку диаграммы, куда нужно вставить второй блок Signal Builder. Отпустив левую кнопку мыши, вы увидите его. Обратите внимание, что имя (метка) этого блока несколько изменилась: он называется Signal Builder1. Объясняется это тем, что каждый блок, включенный в блок-диаграмму, должен иметь уникальное имя. Если бы в модели использовалось пять блоков Signal Builder, то последний из них получил бы имя Signal Builder4. Аналогичным образом именуются все блоки, используемые в модели.

Чтобы создаваемая модель стала более понятной, можно заменить метки блоков другими названиями, которые соответствуют его функциональному назначению. Для этого поместите курсор на название блока и щелкните левую кнопку мыши. Название блока окажется заключенным в прямоугольную рамку, а курсор примет форму вертикальной мерцающей черты. Используя клавиатуру, удалите стандартное имя блока и замените его на новое. Щелкните по левой кнопке мыши вне текстовой области. Выделяющая рамка исчезнет.

Замечание: все блоки в блок-диаграмме обязательно должны иметь имя – либо предложенное Simulink, либо назначенное разработчиком модели; поэтому нельзя сделать блок «безымянным», используя кнопку Del или Backspace, имя блока можно только «скрыть» с помощью команды Hide Name (из раздела меню Format).

Кроме того, для корректной работы модели всем блокам рекомендуется давать название на английском языке или латинскими буквами!

Для изменения параметров блока, поместите курсор на изображение блока, и дважды щелкните левой кнопкой мыши. В открывшемся окне настроек измените значение изменяемого параметра (например, Number of input (Число входов)).

Важно помнить, что при вводе численных параметров в блоках MatLab дробная часть от числа отделяется точкой!

Нажмите кнопку Apply, затем – Close. Иконка блока изменится в соответствии с новым значением параметра настройки.

Теперь можно соединить блоки, помещенные в поле блок-диаграммы, в соответствии с логикой их взаимодействия. Для соединения двух блоков, необходимо подвести курсор мыши к выходу первого или входу второго блока, нажать и, не отпуская левую кнопку мыши, провести линию соединения. Если связь между блоками появилась, то линия будет черного цвета, при отсутствии – красная.

Для создания нескольких линий связи из одной точки (например, для вывода сигнала на осциллограф) необходимо выполнить следующие действия: соединить два блока, нажав левую кнопку мыши, затем, встав на

линию связи и нажав правую кнопку мыши, провести еще одну линию связи к другому блоку (или к осциллографу).

Перемещать линии связи (также как и блоки) можно, если предварительно выделить их (по краям при этом появляются черные квадратики), а затем двигать с помощью мыши, нажав левую кнопку мыши, или кнопками с обозначениями стрелок на клавиатуре. При перемещении блоков линии связи между ними автоматически растягиваются или сужаются.

Для изменения размеров блока необходимо выделить его, нажав левую кнопку мыши, и подведя курсор мыши к одному из углов блока растянуть/сжать его.

После соединения блоков модель будет выглядеть так, как показано на рис. 2.1.

Можно объединить несколько блоков в подсистему. С этой целью необходимо выполнить действия, описанные в п. 1.7.13.

2.1. Построение амплитудной и модуляционной характеристики РП

Внешний вид амплитудной и модуляционной характеристик РП представлен на рис. 2.2.

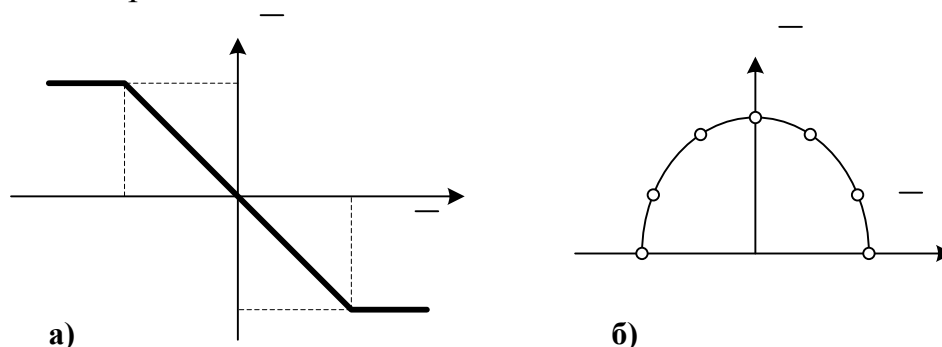


Рис. 2.2. Амплитудная (а) и модуляционная (б) характеристики РП

Амплитудная характеристика РП $\bar{Y}_0 = f(\bar{X}_{\text{ВХ}})$ представляет собой линейную зависимость с единичным наклоном и зонами насыщения при достижении величины напряжения источника питания ($\pm A$), причем входной сигнал передается на выход с инверсией (см. рис. 2.2, а).

РП относится к системам с частотно-широотно-импульсной модуляцией поэтому модуляционная характеристика $\bar{f}_0 = f(\bar{X}_{\text{ВХ}})$ представляет собой перевернутую ветвь параболы (см. рис. 2.2, б).

Характеристики построены в относительных единицах, т.е. в соответствии с уравнениями (2.1) и (2.2).

$$\bar{Y}_0 = Y_0 / A, \quad (2.1)$$

$$\bar{f}_0 = f_0 / f_{0|_{\bar{X}_{\text{ВХ}} = 0}} \quad (2.2)$$

Для построения амплитудной характеристики в среде MatLab в модели представленной на рис. 2.1 следует выполнить следующие действия:

1. Изменить внешний вид сигнала в блоке Signal Builder на линейно возрастающий или убывающий (в пределах от -1 до $+1$ и добавить участки насыщения $\geq \pm 1$).

2. Добавить блок Display (см. п. 1.3.11.4) на верхний вход которого подать сигнал с выхода РП, а на нижний – с блока Signal Builder. Выходной сигнал с РП передавать на вход дисплея через фильтр представленный, например, апериодическим звеном первого порядка (из раздела Continuous, блок Transfer Fcn см. п.1.3.1). Параметры фильтра установить оптимальными, чтобы не ухудшить значительно, динамические свойства системы и получить среднее значение на выходе РП в установившемся режиме – прямую линию.

3. Задать удобные масштабы в параметрах блока Display, для наглядности отображаемых результатов, чтобы наблюдать диапазоны изменения $\bar{Y}_0$ и $X_{ВХ}$ целиком.

Модуляционную характеристику построить по семи точкам (см. рис. 2.2, б). Для ее построения следует выполнить следующие действия:

1. Изменить внешний вид сигнала в блоке Signal Builder на постоянный или задать входной сигнал от блока Constant (см. п.1.3.12).

2. Определить период выходного сигнала РП, по методике описанной в п. 1.3.11.1, см. рис. 1.45.

3. Зная период, рассчитать частоту $f_0 = 1/T_0$ на выходе РП (для одной точки).

4. Повторить все пункты с начала для другого входного сигнала, определив, таким образом, значения f_0 в семи точках при постоянном значении $X_{ВХ}$.

2.2. Рекомендации по работе с графической средой Paint

Полученные в процессе виртуального моделирования зависимости (временные или одной величины от другой) удобно обрабатывать в графическом редакторе Paint, который является стандартным приложением среды Windows 95, 98, 2000 или XP.

Для этого необходимо выполнить следующие действия:

1. В среде MatLab преобразовать графическое изображение к необходимому виду. Например, для временных диаграмм – два полных периода. Менять внешний вид изображения следует в соответствии с рекомендациями, описанными выше (см. п. 1.3.11.1).

2. Скопировать все изображение с экрана монитора в буфер с помощью кнопки «Print Screen» расположенной на клавиатуре.

3. Открыть редактор «Paint» щелкнув на ярлык  расположенный, на рабочем столе или зайти в меню «Пуск → Программы → Стандартные → Paint».

4. Из буфера вставить в Paint изображение, выбрав команду «Правка → Вставить», расположенную в верхней части окна редактора Paint (рис. 2.3).

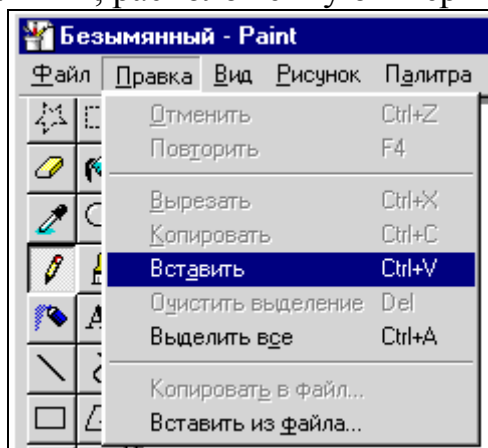


Рис. 2.3. Команды вставки в окне редактора Paint

5. Для удобства последующей печати, графическое изображение на черном фоне можно инвертировать (в первую очередь это относится к временным диаграммам, представленным в блоке Score). Для этого в верхней части окна редактора Paint выбрать команду «Рисунок → Обратить цвета» (рис. 2.4)

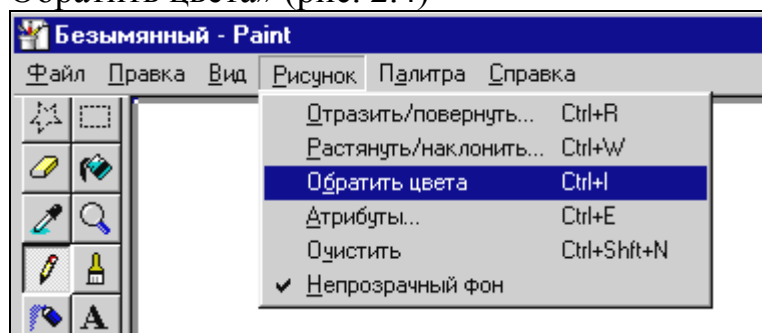


Рис. 2.4. Команды инвертирования в окне редактора Paint

6. Оставшийся серый фон также можно сделать белым, если нажать на кнопку Заливка , расположенную на боковой панели инструментов и выбрать цвет заливки – белый, щелкнув на соответствующий цвет на палитре цветов , расположенной в нижней части экрана графического редактора. Затем курсор мыши (преобразовавшийся к виду ) установить на область, цвет которой следует изменить, и щелкнуть правой кнопкой мыши.

7. Выделить необходимую часть графического изображения можно, нажав на кнопку «Выделить»  или «Выделение произвольной области» , которые расположены на боковой панели графического редактора Paint (см. рис. 2.3).

8. Скопировать выделенную часть графического изображения позволяет команда «Правка → Копировать», расположенную в верхней части окна редактора Paint.

9. Для сравнения полученных результатов в процессе разных опытов некоторые осциллограммы (по указанию преподавателя) или зависимости $Y = f(X)$ удобно располагать на одних осях координат. С этой целью оказывается удобен прозрачный фон, который можно установить в окне графического редактора Paint с помощью нажатия на нижнюю кнопку



, расположенной на боковой панели. Для возврата к белому фону следует нажать на верхнюю кнопку.

10. При отсутствии необходимой панели с инструментами на экране графического редактора ее следует активизировать, пометив галочкой в меню Вид (рис. 2.5).

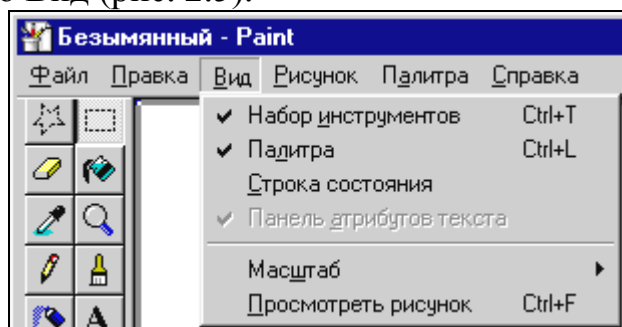


Рис. 2.5. Добавление панели инструментов в окно редактора Paint

11. Удалить часть изображения поможет кнопка «Ластик» .

При выборе какого-либо другого цвета, кроме белого, из палитры цветов, расположенной в нижней части экрана графического редактора, ластик оставляет за собой цветной след.

12. Остальные кнопки из набора инструментов имеют следующее функциональное назначение:



– выбор цветов – по выполняемым функциям это кнопка противоположна и используется в паре с кнопкой Заливка ;  – изменение масштаба,  – карандаш для рисования;  – кисть;  – распылитель;  – надпись;   – прямая и волнистая линия;   – прямоугольник и многоугольник;   – овал и скругленный прямоугольник.

3. ЗАДАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ

В соответствии с заданным вариантом в среде Matlab+Simulink создать модель многозонного развертывающего преобразователя, построить для него амплитудную $\bar{Y}_0 = f(\bar{X}_{ВХ})$ и модуляционную $f_0 = f(\bar{X}_{ВХ})$ характеристики.

Структурные схемы и условные обозначения всех блоков входящих в состав многозонных развертывающих преобразователей (рис. 3.1 – рис. 3.9), а также подробное описание их работы дано в работе [4].

Вариант 1. Многозонный развертывающий преобразователь с выборкой мгновенных значений с широтно-импульсной модуляцией первого рода (МРП-ВМЗ с ШИМ-1). Число ключевых элементов $n = 3$ (рис. 3.1).

Вариант 2. Многозонный развертывающий преобразователь с выборкой мгновенных значений с широтно-импульсной модуляцией второго рода (МРП-ВМЗ с ШИМ-2). Число ключевых элементов $n = 5$ (см. рис. 3.1).

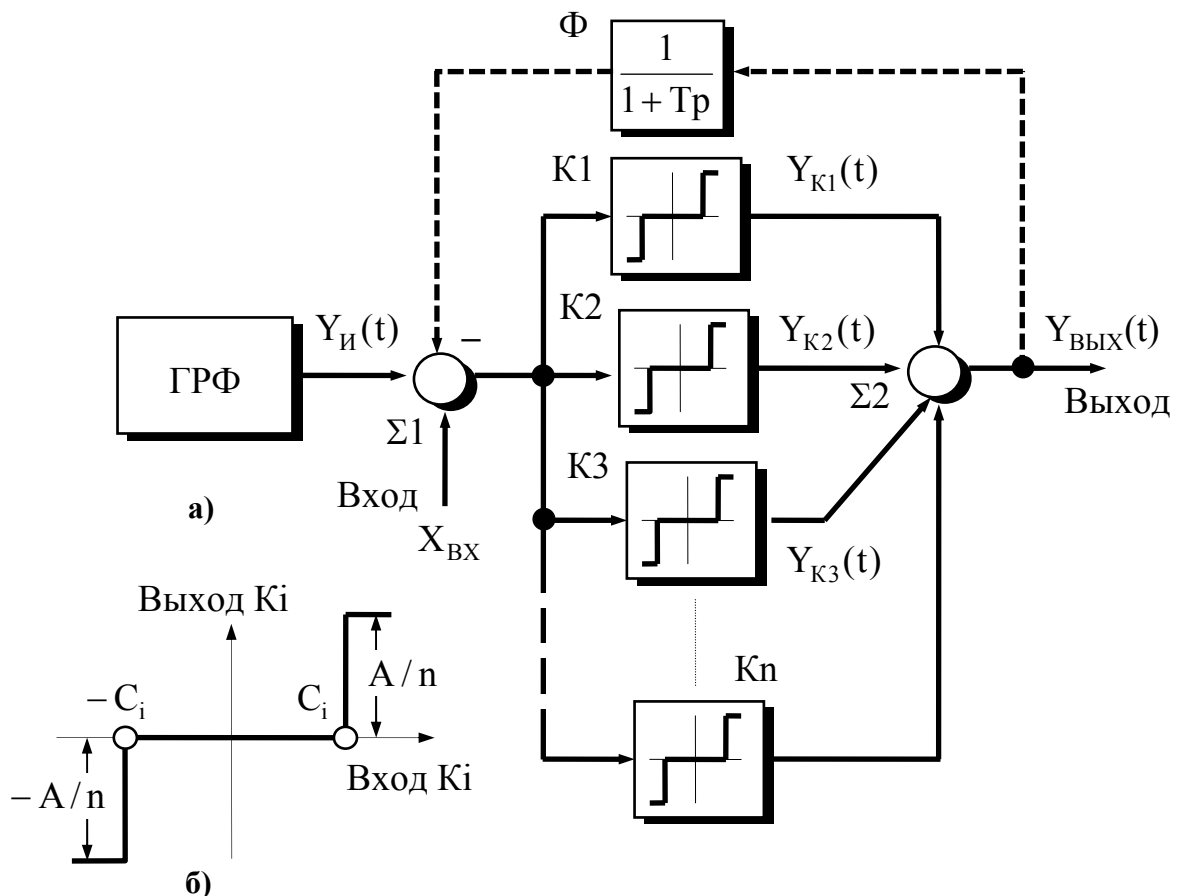


Рис. 3.1. Структурная схема МРП-ВМЗ (а) с ШИМ-1 или ШИМ-2 и статическая характеристика ключевого элемента (б)

Вариант 3. Многозонный интегрирующий развертывающий преобразователь с частотно-нулевым сопряжением модуляционных зон (МРП-НС) (рис. 3.2). Число релейных элементов $n = 3$.

Вариант 4. Многозонный интегрирующий развертывающий преобразователь с частотно-нулевым сопряжением модуляционных зон (МРП-НС) (см. рис. 3.2). Число релейных элементов $n = 5$.

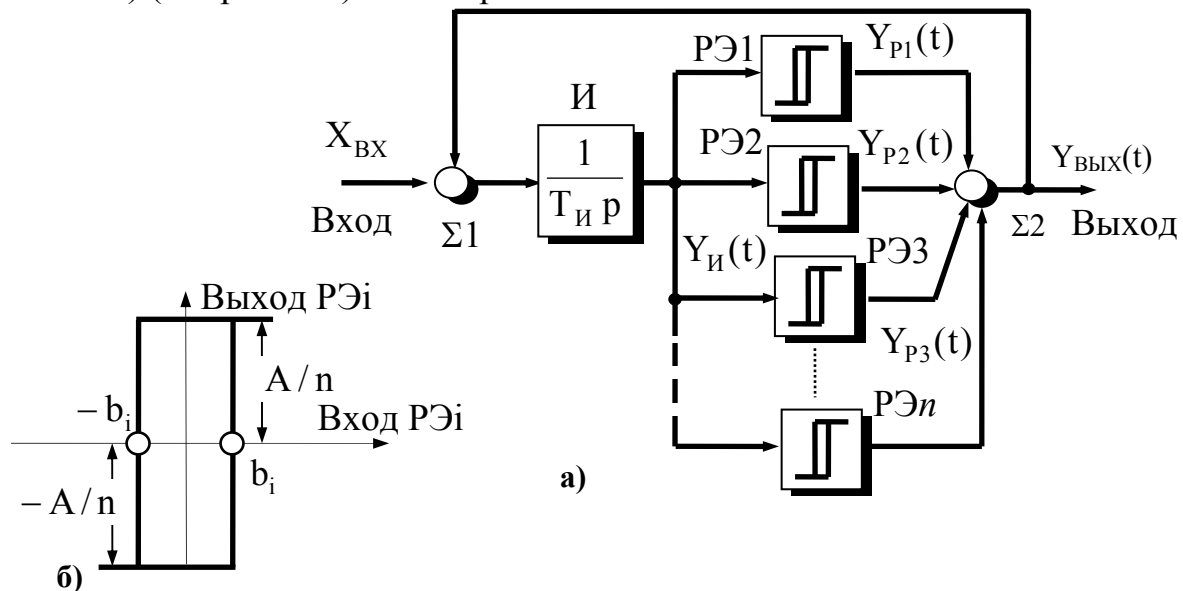


Рис. 3.2. Структурная схема МРП-НС (а) и статическая характеристика релейного элемента (б)

Вариант 5. Многозонный интегрирующий развертывающий преобразователь с частотно-нулевым сопряжением модуляционных зон (МРП-НС) (см. рис. 3.2). Число релейных элементов $n = 4$.

Вариант 6. Многозонный интегрирующий развертывающий преобразователь с частотно-нулевым сопряжением модуляционных зон (МРП-НС) (см. рис. 3.2). Число релейных элементов $n = 6$.

Вариант 7. Многозонный интегрирующий развертывающий преобразователь (МРП) с дополнительным источником опорного сигнала (рис. 3.3). Число релейных элементов $n = 4$.

Вариант 8. Многозонный интегрирующий развертывающий преобразователь (МРП) с дополнительным источником опорного сигнала (см. рис. 3.3). Число релейных элементов $n = 6$.

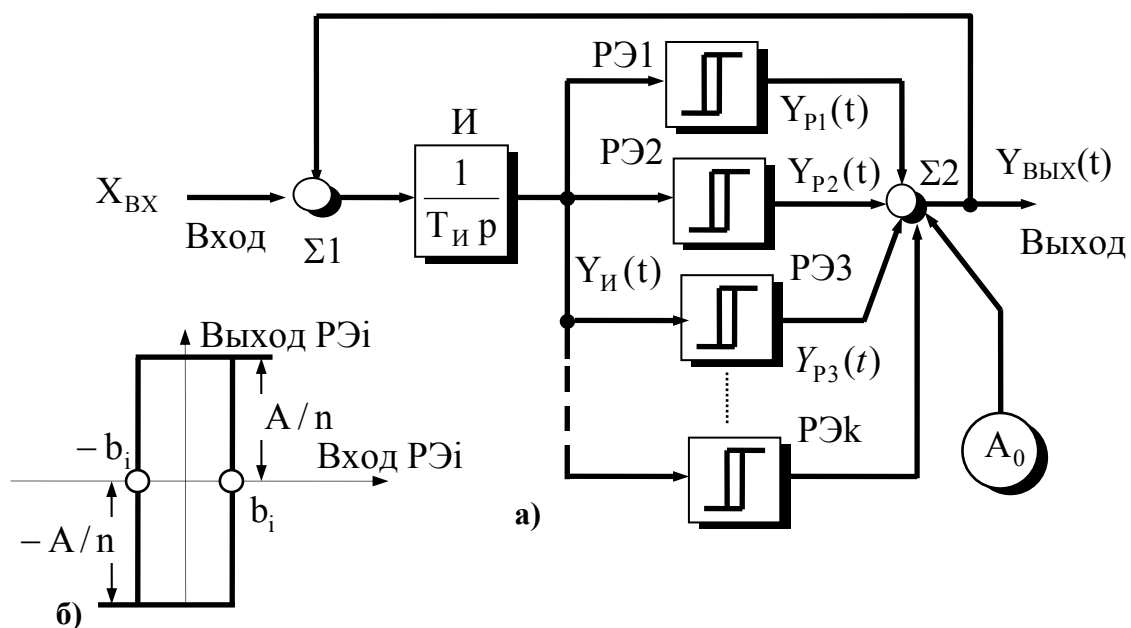


Рис. 3.3. Структурная схема МРП с дополнительным источником опорного сигнала (а) и статическая характеристика релейного элемента (б)

Вариант 9. Многозонный интегрирующий развертывающий преобразователь с частотно-нулевым сопряжением модуляционных зон (МРП-НС) с принудительной ориентацией релейных элементов при включении источника электропитания (рис. 3.4). Число релейных элементов $n = 3$.

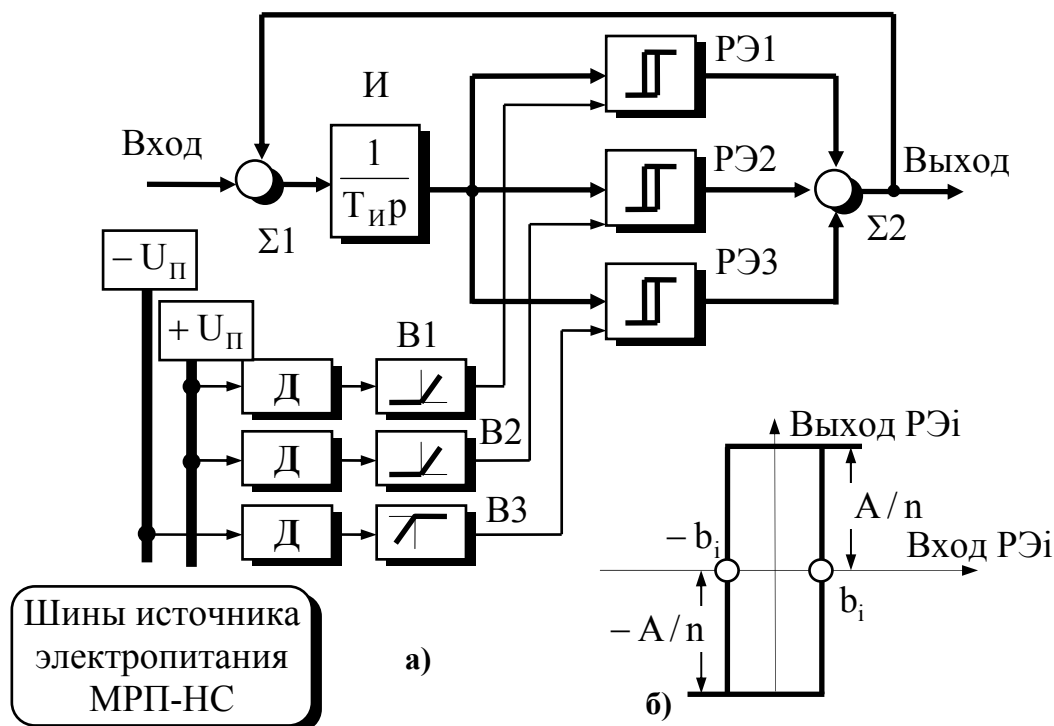


Рис. 3.4. Структурная схема МРП-НС с принудительной ориентацией релейных элементов при включении источника электропитания (а) и статическая характеристика релейного элемента (б)

Вариант 10. Многозонный интегрирующий развертывающий преобразователь с частотно-нулевым сопряжением модуляционных зон (МРП-НС) с принудительной ориентацией релейных элементов (см. рис. 3.4). Число релейных элементов $n = 5$.

Вариант 11. Многозонный интегрирующий развертывающий преобразователь с частотно-нулевым сопряжением модуляционных зон (МРП-НС) с принудительной ориентацией релейных элементов (см. рис. 3.4). Число релейных элементов $n = 4$.

Вариант 12. Многозонный интегрирующий развертывающий преобразователь с параметрическим частотно-импульсным сопряжением модуляционных зон (МРП-ЧС) (рис. 3.5). Число релейных элементов $n = 3$.

Вариант 13. Многозонный интегрирующий развертывающий преобразователь с параметрическим частотно-импульсным сопряжением модуляционных зон (МРП-ЧС) (см. рис. 3.5). Число релейных элементов $n = 5$.

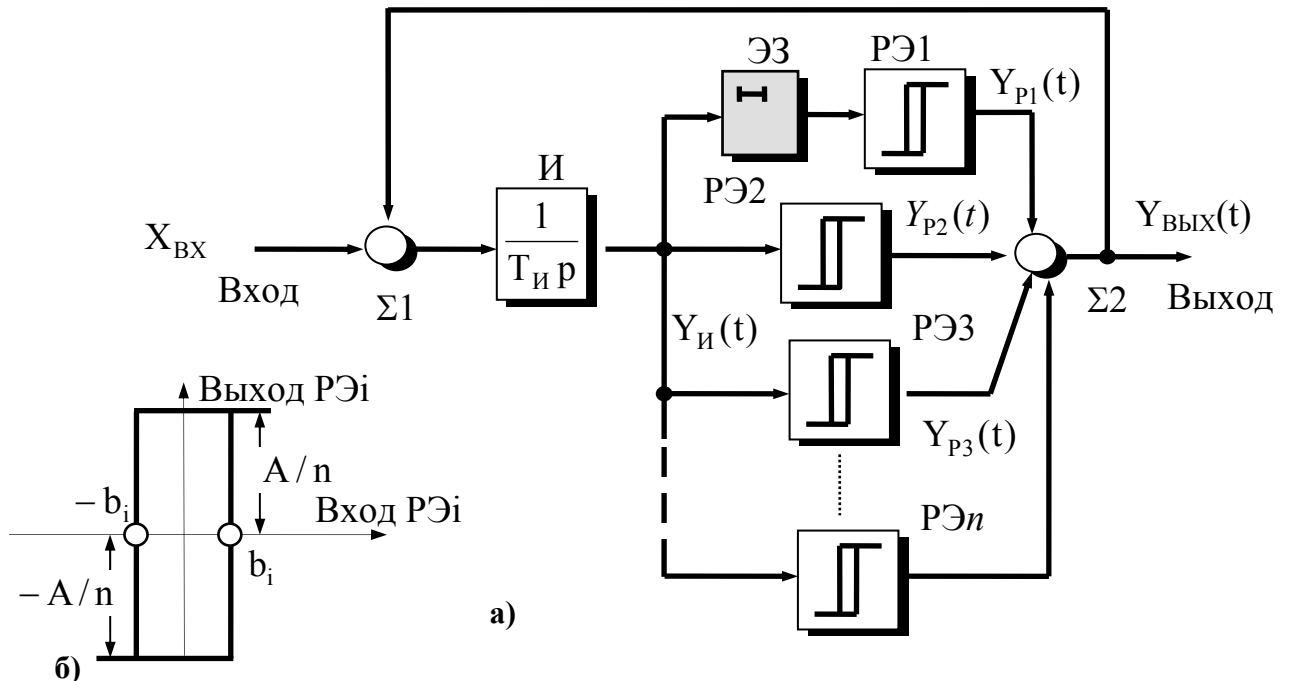


Рис. 3.5. Структурная схема МРП-ЧС (а) и статическая характеристика релейного элемента (б)

Вариант 14. МРП с параметрическим «эстафетным» алгоритмом работы релейных элементов (см. рис. 3.2). Число релейных элементов $n = 3$.

Вариант 15. МРП с параметрическим «эстафетным» алгоритмом работы релейных элементов (см. рис. 3.2). Число релейных элементов $n = 5$.

Вариант 16. МРП с параметрическим «эстафетным» алгоритмом работы релейных элементов (см. рис. 3.2). Число релейных элементов $n = 4$.

Вариант 17. МРП с интервало-временным «эстафетным» алгоритмом работы РЭ (рис. 3.6). Число релейных элементов $n = 3$

Вариант 18. МРП с интервало-временным «эстафетным» алгоритмом работы РЭ (см. рис. 3.6). Число релейных элементов $n = 4$.

Вариант 19. МРП с кодо-адресным алгоритмом перестройки динамических характеристик (рис. 3.7). Число релейных элементов $n = 3$.

Вариант 20. МРП с кодо-адресным алгоритмом перестройки динамических характеристик (см. рис. 3.7). Число релейных элементов $n = 5$.

Вариант 21. МРП с адаптируемой в функции неисправности активных компонентов структурой (рис. 3.8). Число релейных элементов $n = 3$.

Вариант 22. МРП с адаптируемой в функции неисправности активных компонентов структурой (см. рис. 3.8). Число релейных элементов $n = 4$.

Вариант 23. МРП с адаптируемой в функции неисправности активных компонентов структурой (см. рис. 3.8). Число релейных элементов $n = 5$.

Вариант 24. МРП с принудительным тестированием релейных элементов (рис. 3.9). Число релейных элементов $n = 3$.

Вариант 25. МРП с принудительным тестированием релейных элементов (см. рис. 3.9). Число релейных элементов $n = 4$.

Вариант 23. МРП с принудительным тестированием релейных элементов (см. рис. 3.9). Число релейных элементов $n = 5$.

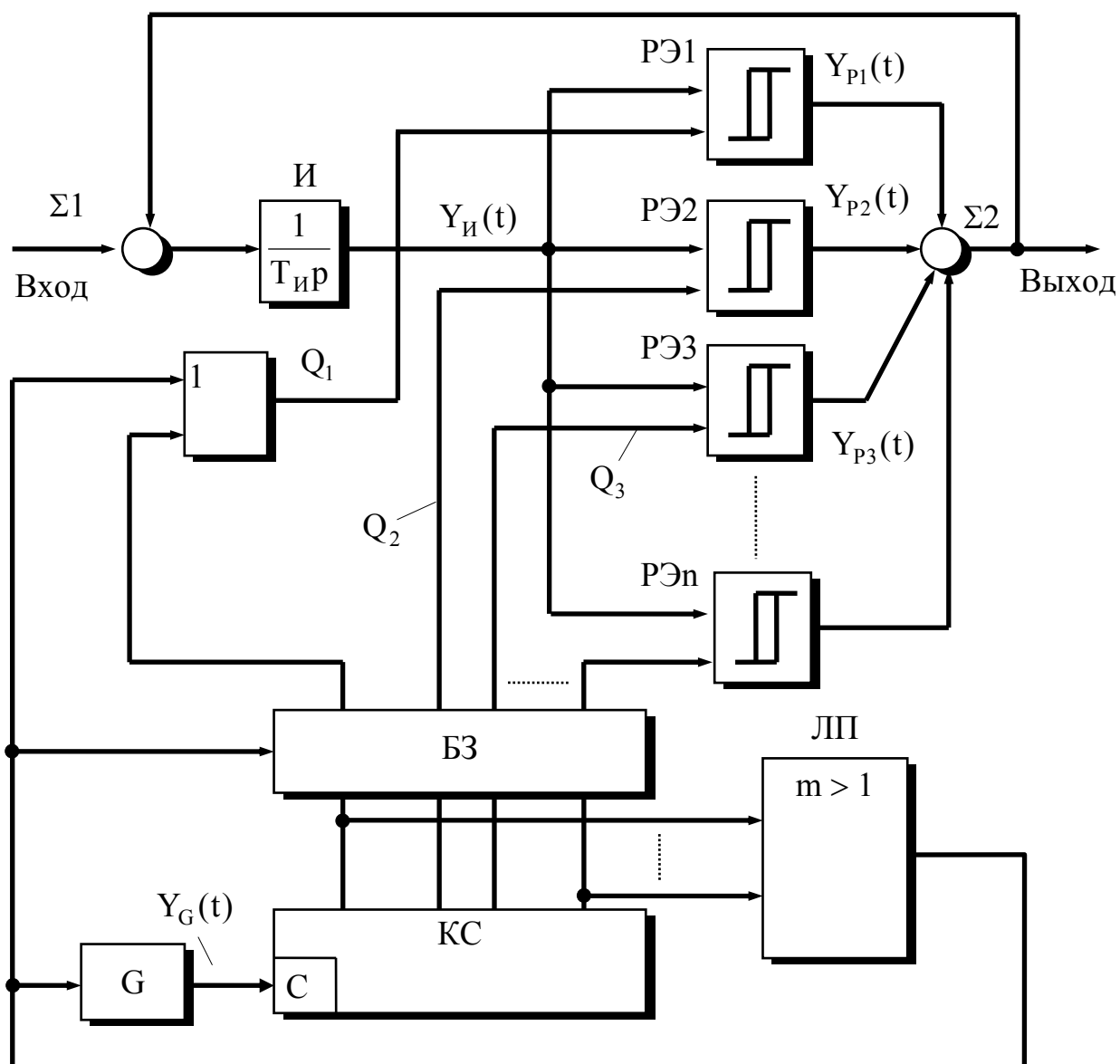


Рис. 3.6. Структурная схема МРП с интервало-временным «эстафетным» алгоритмом работы РЭ

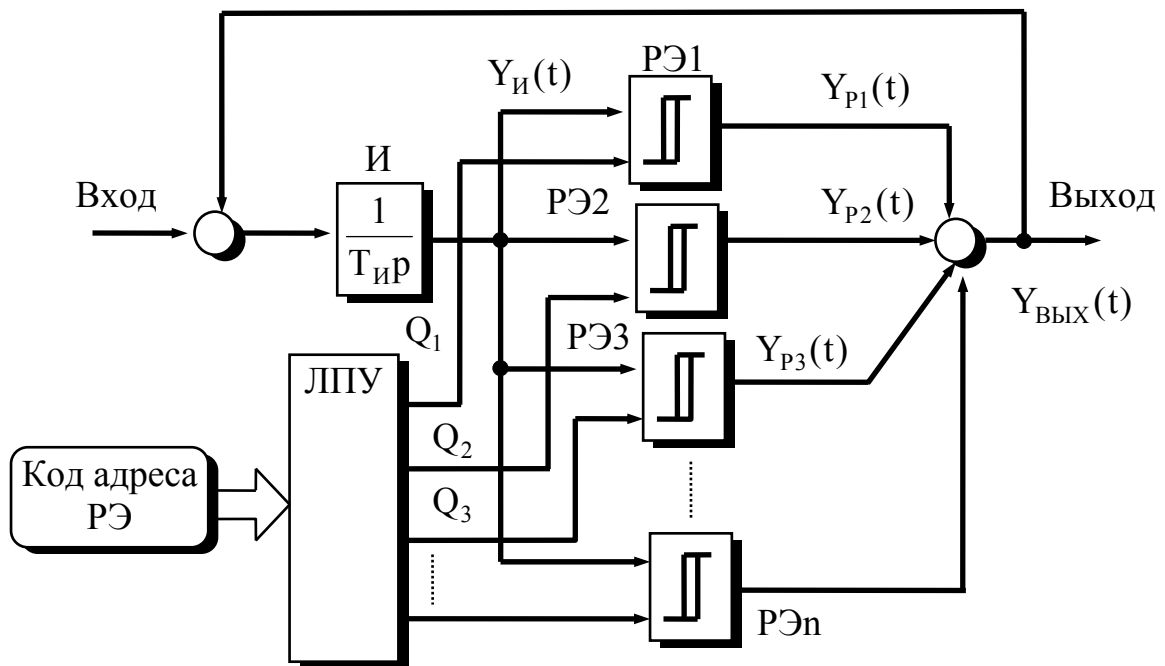


Рис. 3.7. Структурная схема МРП с кодо-адресным алгоритмом перестройки динамических характеристик

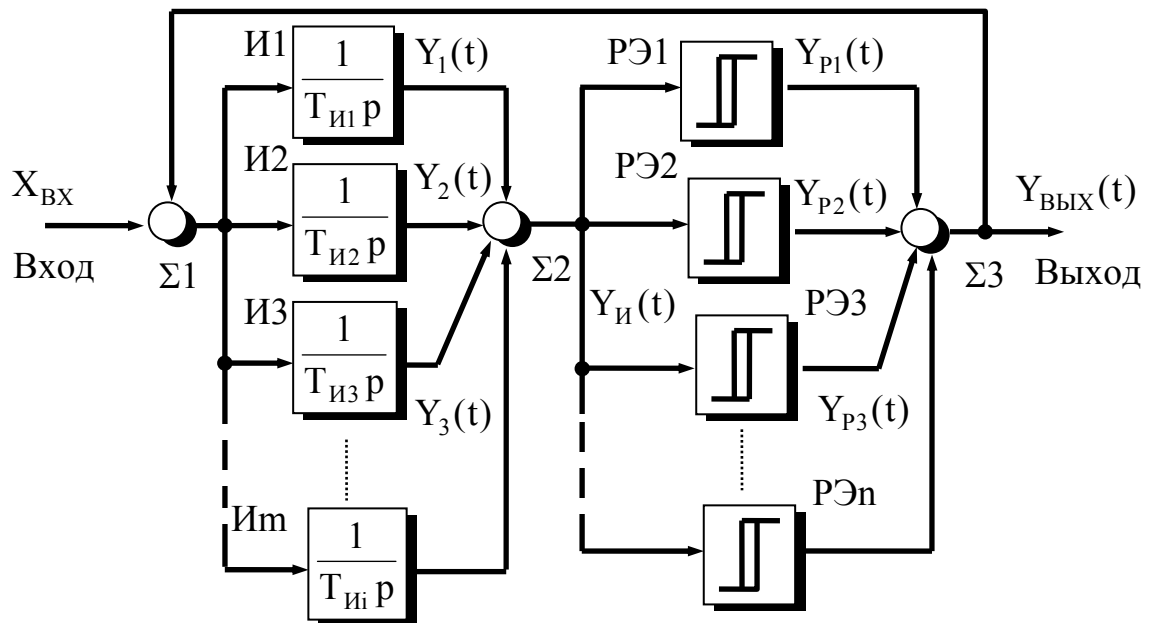


Рис. 3.8. Структурная схема МРП с адаптируемой в функции неисправности активных компонентов структурой

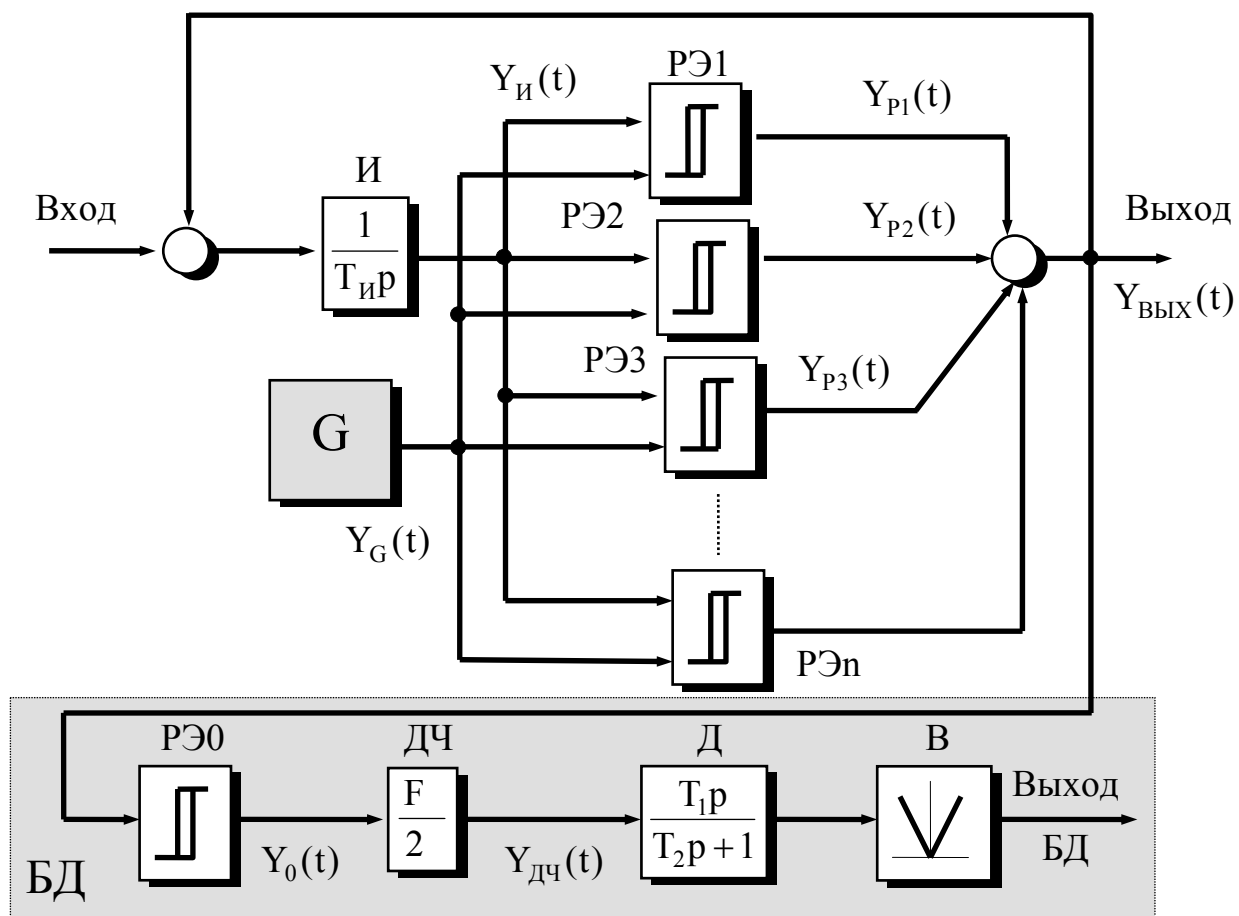


Рис. 3.9. Структурная схема МРП с принудительным тестированием релейных элементов

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем состоит принципиальное отличие развертывающего преобразования с выборкой мгновенного значения сигнала управления от интегрирующего развертывающего преобразования?
2. Чем отличается широтно-импульсная и частотно-широтно-импульсная модуляции?
3. Что такое модуляционная и амплитудная характеристики развертывающего преобразователя?
4. В чем состоят различия между однозонной и многозонной, двухтактной и одноконтной модуляций?
5. Что такое частотно-нулевое и частотно-импульсное сопряжение модуляционных зон?
6. Чем отличаются широтно-импульсная модуляция первого и второго рода?
7. В чем состоят основные недостатки многозонных развертывающих преобразователей с выборкой мгновенных значений входного сигнала?
8. Как влияет нестабильность амплитуды и частоты сигнала развертки на метрологические характеристики МРП с ШИМ-1 и ШИМ-2?
9. Как влияет нестабильность параметров ключевых элементов на метрологические характеристики МРП с ШИМ-1 и ШИМ-2?
10. При каком числе релейных элементов в МРП реализуется многозонная модуляция? Как связаны число модуляционных зон с числом релейных элементов МРП?
11. Почему амплитудная характеристика МРП-НС является линейной во всем диапазоне изменения входного сигнала?
12. Какой вид модуляции реализуется в каждой из модуляционных зон МРП-НС? В чем заключаются ее преимущества и недостатки?
13. В каких системах регулирования целесообразно применять МРП-НС?
14. В чем состоит специфика работы МРП-НС с четным числом релейных элементов?
15. Какому условию должны удовлетворять значения амплитуд выходных сигналов релейных элементов в МРП-НС с $n = 2$, $n = 4$, $n = 6$, $n = 8$?
16. Каким недостатком обладает МРП с дополнительным источником опорного сигнала и из какого условия выбирается его величина?
17. Как производится принудительная ориентация релейных элементов при включении МРП?
18. Для чего нужен режим частотно-импульсного сопряжения модуляционных зон?
19. Что такое параметрическое частотно-импульсное сопряжение модуляционных зон и каким путем оно достигается?

20. Что такое “эстафетный” алгоритм работы релейных элементов МРП и для чего он нужен?

21. Каким образом в МРП достигается параметрический “эстафетный” алгоритм управления релейными элементами? В чем его преимущества и недостатки?

22. Что такое “вакантный уровень” порогов переключения группы релейных элементов?

23. В чем состоит основной недостаток интервало-временного алгоритма управления релейными элементами?

24. На чем базируется число-импульсный “эстафетный” алгоритм управления релейными элементами, его преимущества и недостатки?

25. Как реализовать релейные элементы с дискретно перестраиваемыми порогами переключения?

26. На чем основан кодо-адресный алгоритм перестройки динамических характеристик МРП?

27. В чем состоит основной недостаток кодо-адресного алгоритма изменения динамических характеристик МРП?

28. Чему равно значение постоянной времени канала интегрирования МРП при параллельной работе нескольких интеграторов?

29. Как влияет на модуляционную и амплитудную характеристики МРП отказы интеграторов и релейных элементов?

30. Почему в многократно резервируемом МРП выходной сумматор необходимо выполнять на основе пассивного R - “сумматора” и как это влияет на метрологические показатели разворачивающего преобразователя?

31. Какое число интеграторов и релейных элементов в МРП с автоматическим резервированием следует считать оптимальным?

32. Какой вид катастрофических отказов элементов МРП является наиболее тяжелым с точки зрения их влияния на статические и динамические показатели разворачивающего преобразователя?

33. Как выбрать параметры дифференцирующего звена блока диагностирования?

34. Для чего в схему блока диагностирования вводится делитель частоты?

35. Для чего МРП снабжается контуром принудительного тестирования релейных элементов? В чем заключается недостаток подобного технического решения?

36. Назвать целесообразные области применения МРП с автоматическим диагностированием и резервированием активных компонентов схемы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дьяконов, В.П. Matlab 6.5 SP1/7 + Simulink 5/6 в математике и моделировании / В.П. Дьяконов // Серия «Библиотека профессионала». – М.: СОЛОН-Пресс, 2005. – 576 с.
2. Дьяконов, В.П. Matlab 6.5 SP1/7 + Simulink 5/6. Основы применения / В.П. Дьяконов // Серия «Библиотека профессионала». – М.: СОЛОН-Пресс, 2005. – 800 с.
3. Лазарев Ю. Моделирование процессов и систем в Matlab. Учебный курс / Ю. Лазарев. – СПб.: Питер, 2005. – 512 с.
4. Брылина, О.Г. Многозонные развертывающие преобразователи для систем управления электроприводами: монография / О.Г. Брылина, Л.И. Цытович. – Челябинск: Издательский центр ЮУрГУ, 2010. – 150 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. Краткое руководство по работе в Matlab. основные блоки Matlab	4
1.1. Запуск MatLab, создание и открытие файла	4
1.2. Библиотеки Simulink, SimPowerSystems и Simulink Extras	5
1.3. Библиотека SIMULINK	6
1.3.1. Раздел Continuous	7
1.3.2. Раздел Discontinuities	12
1.3.3. Раздел Discrete	14
1.3.4. Раздел Look-Up Tables	17
1.3.5. Раздел Math Operation	17
1.3.6. Раздел Model Verification	23
1.3.7. Раздел Model-Wide Utilities	23
1.3.8. Раздел Ports & Subsystems	24
1.3.9. Раздел Signal Attributes	29
1.3.10. Раздел Signal Routing	32
1.3.11. Раздел Sinks	40
1.3.11.1. Блок Scope	43
1.3.11.2. Блок Floating Scope	49
1.3.11.3. Блок XY Graph	50
1.3.11.4. Блок Display	51
1.3.12. Раздел Sources	52
1.3.12.1. Блок Constant	54
1.3.12.2. Блок Signal Generator	55
1.3.12.3. Блок Pulse Generator	56
1.3.12.4. Блок Signal Builder	57
1.3.12.5. Блок Ramp	60
1.3.12.6. Блок Sine Wave	60
1.3.12.7. Блок Step	62
1.3.12.8. Блок Repeating Sequence	63
1.3.12.9. Блок Chirp Signal	63
1.3.12.10. Блок Random Number	64
1.3.12.11. Блок Uniform Random Number	65
1.3.12.12. Блок Band-Limited White Noise	66
1.3.12.13. Блок Clock	67
1.3.12.14. Блок Digital Clock	67
1.3.13. Раздел User-Defined Functions	68
1.3.13.1. блок Fcn	68
1.3.13.2. блок MatLab Fcn	69
1.3.13.3. блок задания S-функций	71
1.3.13.4. блок задания S-function Builder	72
1.4. Библиотека SimPowerSystems	72

1.5. Библиотека Simulink Extras.....	72
1.5.1. Дополнительные дискретные блоки Additional Discrete	73
1.5.2. Дополнительные линейные блоки	73
1.5.3. Дополнительные блоки Additional Sinks.....	74
1.5.3.1. Блоки спектрального анализа	75
1.5.3.2. Блок кросскоррелятора Cross-Correlator.....	77
1.5.4. Обзор раздела библиотеки Flip Flops	77
1.5.4.1. Генератор тактовых импульсов Clock	77
1.5.4.2. Триггерные блоки	78
1.5.5. Раздел Linearization	78
1.5.6. Блоки преобразований Transformations.....	79
1.6. Запуск расчета модели и параметры расчета.....	81
1.7. Построение блок-схем.....	84
1.7.1. Выделение объектов.....	84
1.7.2. Копирование блоков	85
1.7.3. Перемещение блоков в модели	85
1.7.4. Установка параметров блока.....	86
1.7.5. Удаление блоков.....	86
1.7.6. Отсоединение блока.....	86
1.7.7. Изменение ориентации блока	86
1.7.8. Изменение размеров блока.....	87
1.7.9. Изменение имен блоков и манипуляции с ними.....	87
1.7.10. Проведение соединительных линий.....	87
1.7.10.1. Создание линии между блоками	88
1.7.10.2. Создание разветвления линии	88
1.7.10.3. Создание сегмента линии.....	88
1.7.10.4. Перемещение сегмента линии	88
1.7.10.5. Разделение линии на сегменты	89
1.7.11. Метки сигналов и комментарии	89
1.7.11.1. Создание меток сигналов и манипулирование ими	89
1.7.11.2. Распространение меток сигналов	90
1.7.11.3. Комментарии к блок-схемам	90
1.7.12. Сохранение и вывод на печать блок-схемы S-модели	91
1.7.13. Создание подсистем	91
1.7.13.1. Задание свойств подсистемы	93
1.7.13.2. Маскирование подсистем.....	94
2. Пример построения модели в SIMULINK.....	104
2.1. Построение амплитудной и модуляционной характеристики РП... 106	
2.2. Рекомендации по работе с графической средой Paint.....	107
3. Задания к лабораторным работам.....	110
Контрольные вопросы	118
Библиографический список.....	120

Олеся Геннадьевна Брылина, Леонид Игнатьевич Цытович

**МНОГОЗОННЫЕ РАЗВЕРТЫВАЮЩИЕ
ПРЕОБРАЗОВАТЕЛИ ДЛЯ СИСТЕМ УПРАВЛЕНИЯ
ЭЛЕКТРОПРИВОДАМИ**

Учебное пособие к лабораторным работам

Техн. редактор

Издательский центр Южно-Уральского государственного
университета

УОП Издательства. 454080, г. Челябинск, пр. им. В.И. Ленина, 76