

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ ХІМІКО-ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

**ІНСТРУКЦІЯ ДО ЛАБОРАТОРНОЇ РОБОТИ №3**  
**з дисципліни**  
**моделювання систем**  
**за темою «Знайомство з пакетом візуального моделювання Simulink»**

Дніпропетровськ, 2015

Цель лабораторной работы – ознакомиться с возможностями пакета визуального программирования SIMULINK среды MATLAB; изучить его структуру и инструментарий.

## ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Пакет MATLAB широко используется во всем мире при решении задач, связанных с матричными вычислениями. Название пакета образовано путем сокращения от MATrix LABoratory (матричная лаборатория). Операции и команды в MATLAB достаточно естественны и аналогичны математической записи формул на бумаге. MATLAB создавался как пакет программ, реализующих наиболее эффективные вычислительные алгоритмы линейной алгебры. Он организован таким образом, чтобы пользователь имел возможность применять при работе обычный математический язык.

В настоящее время пакет MATLAB представляет собой развитую интегральную программную среду, включающую собственный язык программирования. Он дает пользователю возможность быстро выполнять различные операции над векторами и матрицами, такие как умножение и обращение матриц, вычисление определителей, нахождение собственных чисел и векторов. Кроме того, в MATLAB входят операции вычисления обычных функций (алгебраических, тригонометрических, логических), решения алгебраических и дифференциальных уравнений, операции построения графиков и ряд других.

MATLAB является языком высокого уровня. По отдельным его командам можно выполнять такие сложные операции, как нахождение корней полиномов, решение линейных и нелинейных алгебраических уравнений, моделирование линейных динамических систем. Указанные операции являются элементарными функциями MATLAB.

Помимо ядра, реализующего вычислительные алгоритмы общего назначения, в пакете MATLAB реализовано несколько десятков так называемых тулбоксов (библиотек специализированных подпрограмм), предназначенных для решения разнообразных практических задач. Например, тулбокс SYMBOLIC предназначен для выполнения символьных вычислений, а тулбокс CONTROL – для расчета и моделирования систем автоматического управления

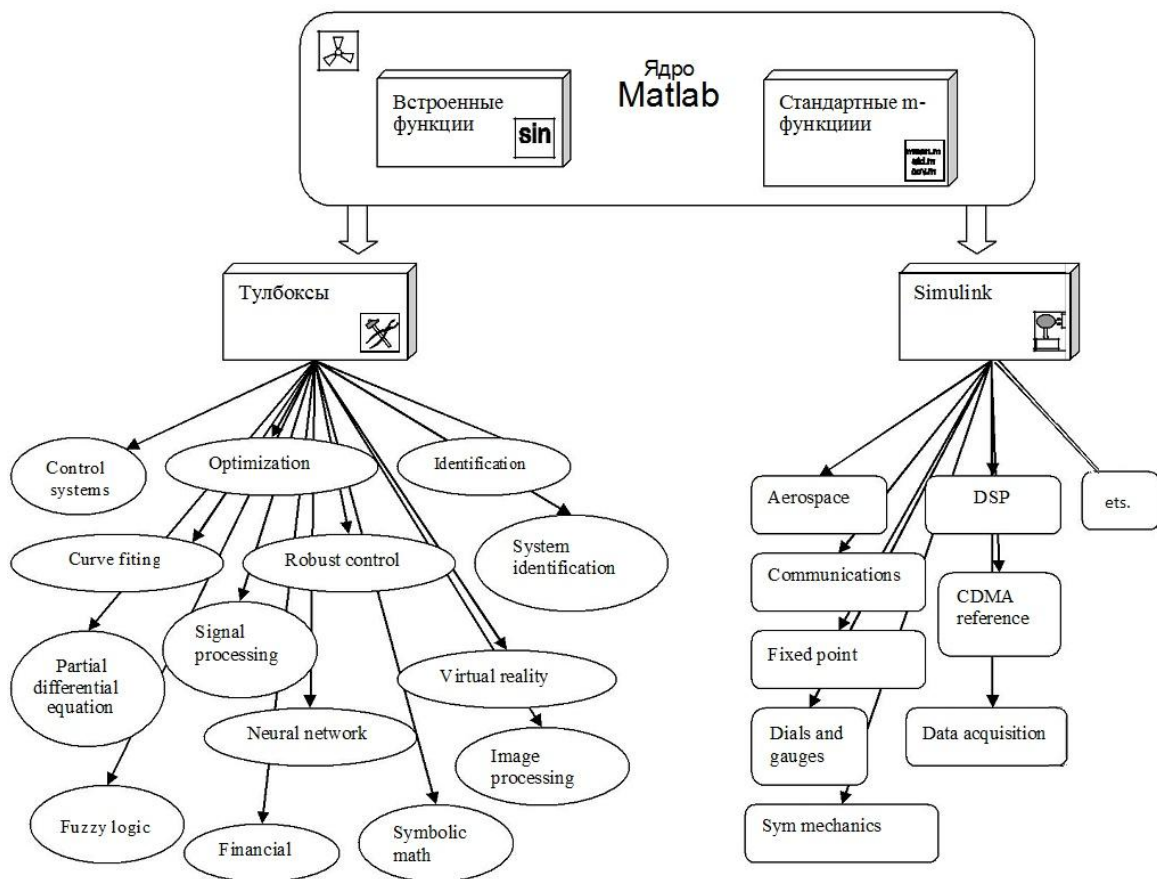



Рис. 1 Структура пакета  
Запуск и начало работы в SIMULINK

Наиболее простым и удобным средством для моделирования динамических систем в пакете MATLAB является SIMULINK. Он поставляется в виде отдельного приложения и содержит библиотеки различных блоков, из которых в рабочем окне строится структурная схема модели. Процесс программирования в явном виде отсутствует, поэтому иногда говорят о визуальном моделировании. Особенно удобен такой подход для моделирования систем автоматического управления, заданных структурными схемами, дифференциальных уравнений (схемы для них строятся по методу понижения производных), механических систем и электрических схем.

SIMULINK можно запустить, набрав команду `simulink` или нажимая на кнопку  в линейке меню. При этом открывается окно браузера библиотек.

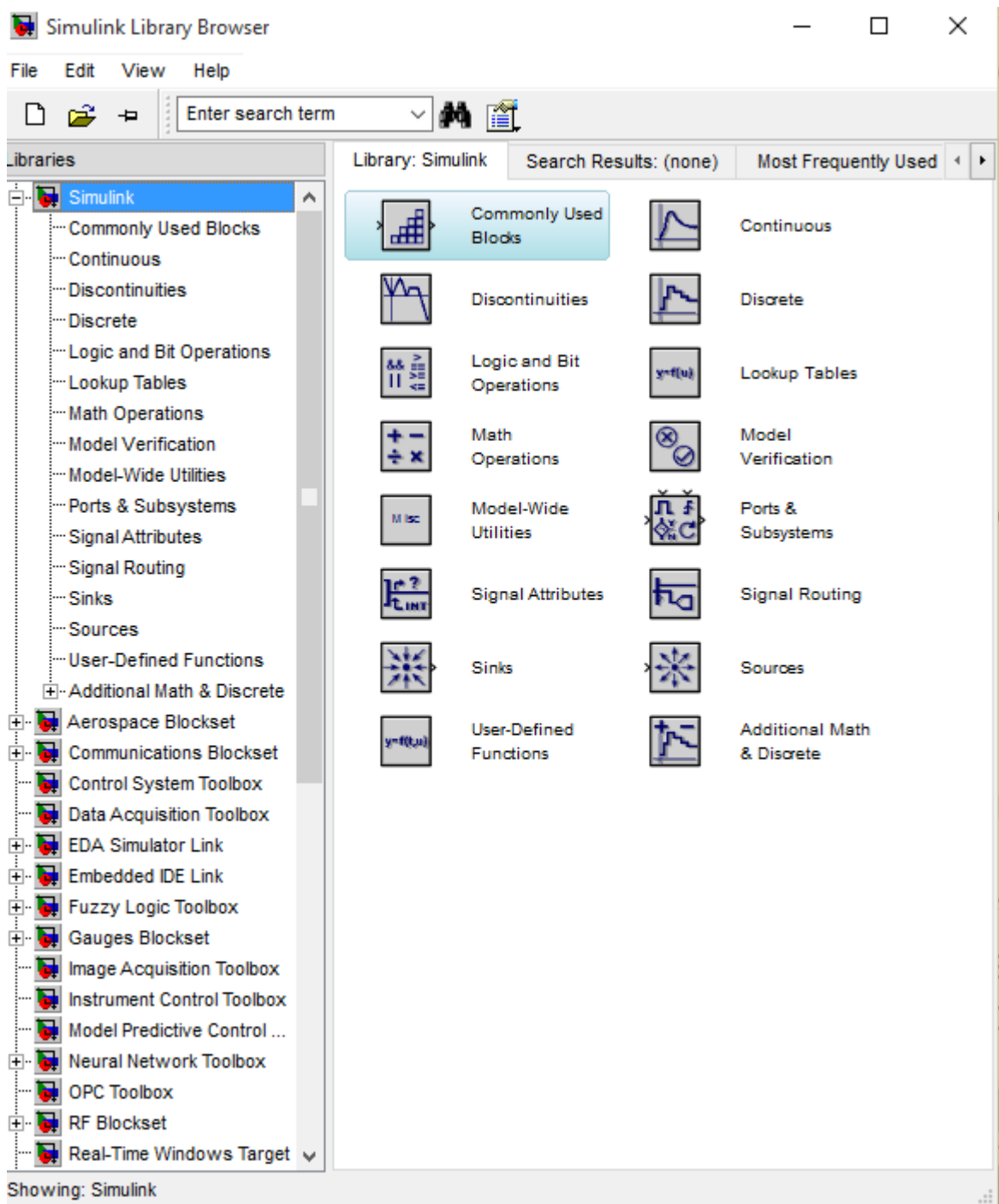



Рис. 2 Окно библиотек

## Создание модели

1. Создать новый файл модели с помощью команды File/New/Model, или используя кнопку  на панели инструментов (здесь и далее, с помощью символа “/”, указаны пункты меню программы, которые необходимо последовательно выбрать для выполнения указанного действия). Вновь созданное окно модели показано на рис. 3.

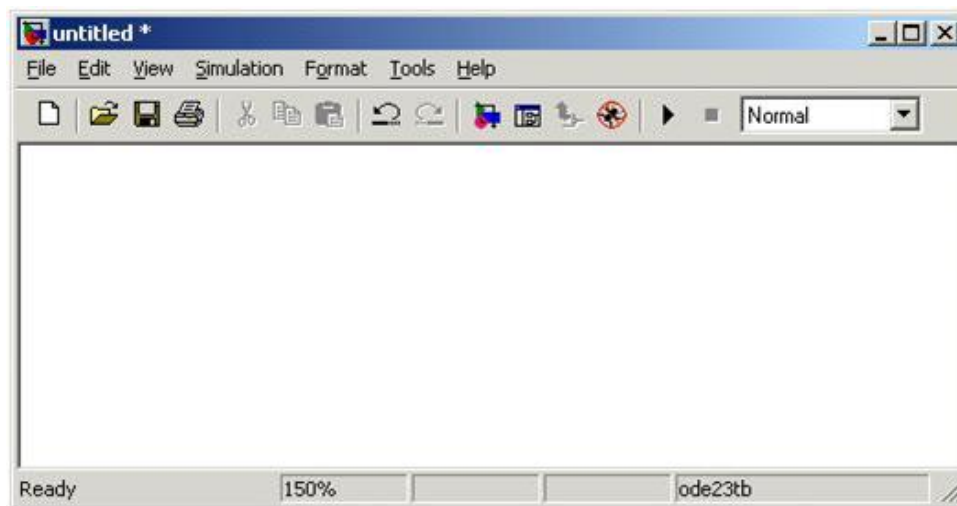


Рис 3. Пустое окно модели

2. Расположить блоки в окне модели. Для этого необходимо открыть соответствующий раздел библиотеки (Например, Sources – Источники). Далее, указав курсором на требуемый блок и нажав на левую клавишу мыши – «перетащить» блок в созданное окно. Клавишу мыши нужно держать нажатой. На рис. 4 показано окно модели, содержащее блоки.

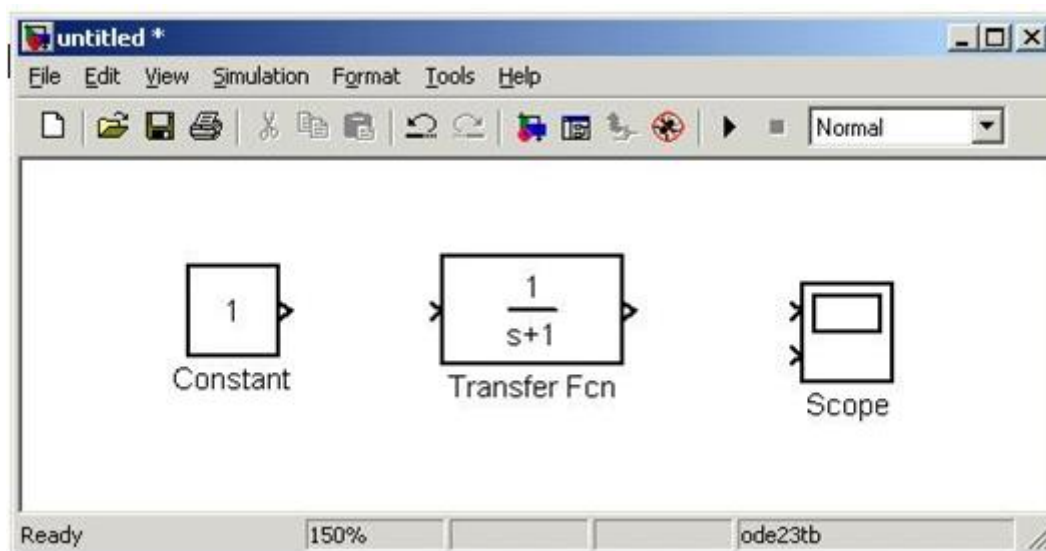


Рис 4. Окно модели, содержащее блоки

3. После установки на схеме всех блоков из требуемых библиотек, нужно выполнить соединение элементов схемы. Для соединения блоков необходимо указать курсором на «выход» блока, а затем, нажав и, не отпуская левую клавишу мыши, провести линию к входу другого блока. После чего отпустить клавишу. В случае правильного соединения, изображение стрелки на входе блока изменяет цвет. Для создания точки разветвления в соединительной линии нужно подвести курсор к предполагаемому узлу и, нажав правую клавишу мыши, протянуть линию.

4. После составления расчетной схемы необходимо сохранить ее в виде файла на диске, выбрав пункт меню File/Save As... в окне схемы и указав папку и имя файла. Следует иметь в виду, что имя файла не должно превышать 32 символа, должно начинаться с буквы и не может содержать символы кириллицы и спецсимволы. Это же требование относится и к пути файла (к тем папкам, в которых сохраняется файл). При последующем редактировании схемы можно пользоваться пунктом меню File/Save. При повторных запусках программы Simulink загрузка схемы осуществляется с помощью меню File/Open... в окне обозревателя библиотеки или из основного окна MATLAB.

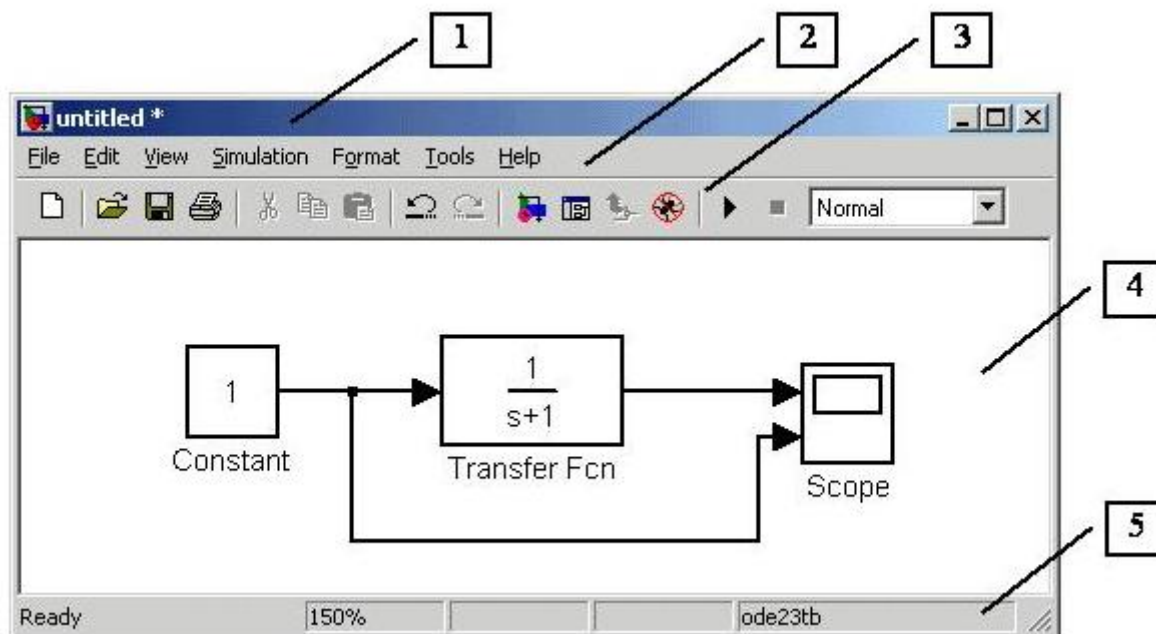


Рис 5. Схема модели

### Окно модели

Окно модели содержит следующие элементы (см.рис. 5):

1. Заголовок с названием окна. (Вновь созданному окну присваивается имя Untitled с соответствующим номером)
2. Меню с командами File, Edit, View и т.д.
3. Панель инструментов.
4. Окно для создания схемы модели.
5. Строка состояния, содержащая информацию о текущем состоянии модели.

Меню окна содержит команды для редактирования модели, ее настройки и управления процессом расчета, работы файлами и т.п

Для работы с моделью можно также использовать кнопки на панели инструментов (рис. 6)

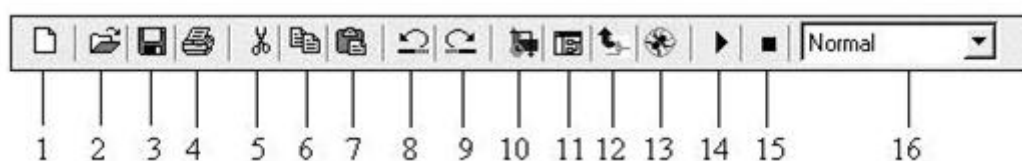



Рис 6. Панель инструментов окна модели

Кнопки панели инструментов имеют следующее назначение:

1. **New Model** – Открыть новое (пустое) окно модели.
2. **Open Model** – Открыть существующий mdl-файл.
3. **Save Model** – Сохранить mdl-файл на диске.
4. **Print Model** – Вывод на печать блок-диаграммы модели.
5. **Cut** – Вырезать выделенную часть модели в буфер промежуточного хранения.
6. **Copy** – Скопировать выделенную часть модели в буфер промежуточного хранения.
7. **Paste** – Вставить в окно модели содержимое буфера промежуточного хранения.
8. **Undo** – Отменить предыдущую операцию редактирования.
9. **Redo** – Восстановить результат отмененной операции редактирования.
10. **Library Browser** – Открыть окно обозревателя библиотек.
11. **Toggle Model Browser** – Открыть окно обозревателя модели.

12. **Go to parent system** – Переход из подсистемы в систему высшего уровня иерархии («родительскую систему»). Команда доступна только если открыта подсистема.
13. **Debug** – Запуск отладчика модели.
14. **Start/Pause/Continue Simulation** – Запуск модели на исполнение (команда **Start**); после запуска модели на изображении кнопки выводится символ , и ей соответствует уже команда **Pause** (Приостановить моделирование); для возобновления моделирования следует щелкнуть по той же кнопке, поскольку в режиме паузы ей соответствует команда **Continue** (Продолжить).
15. **Stop** – Закончить моделирование. Кнопка становится доступной после начала моделирования, а также после выполнения команды **Pause**.
16. **Normal/Accelerator** – Обычный/Ускоренный режим расчета. Инструмент доступен, если установлено приложение **Simulink Performance Tool**.

В нижней части окна модели находится строка состояния, в которой отображаются краткие комментарии к кнопкам панели инструментов, а также к пунктам меню, когда указатель мыши находится над соответствующим элементом интерфейса. Это же текстовое поле используется и для индикации состояния **Simulink: Ready** (Готов) или **Running** (Выполнение).

- масштаб отображения блок-диаграммы (в процентах, исходное значение равно 100%),
- индикатор степени завершенности сеанса моделирования (появляется после запуска модели),
- текущее значения модельного времени (выводится также только после запуска модели),

На рис. 7 показана простейшая схема моделирования. Она содержит три блока: **Sine wave** (генератор синусоиды) из группы **Sources** (генераторы входных сигналов), **Scope** (осциллограф) из группы **Sinks** (регистрация выходных сигналов) и **Integrator** (интегратор) из группы **Continuous** (непрерывные модели). Блоки можно соединить, протянув линию от одного блока к другому при нажатой левой кнопке мыши. Чтобы удалить связь, достаточно выделить ее мышью, а затем нажать клавишу DEL. Блоки можно удалять точно таким же способом.

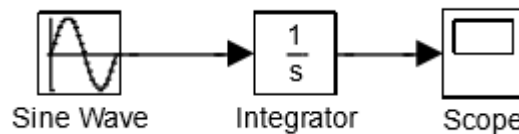


Рис. 7 Простейшая схема моделирования

## Обзор библиотеки блоков Simulink

Далее приводится краткая информация о содержании основных блоков библиотеки **Simulink**.

**Continuous** – блоки аналоговых (непрерывных) сигналов.

**Discrete** - блоки дискретных (цифровых) сигналов.

**Function&Tables** - блоки для формирования функций и таблиц.

**Math** - блоки для реализации математических операций.

**Nonlinear**- нелинейные элементы.

**Signal&Systems** - сигналы и системы.

**Sinks** - приемники и измерители сигналов.

**Sources** - источники сигналов.

**Subsystems**- подсистемы.

Далее будут рассмотрены основные библиотеки.

### Sources – источники сигналов

Источник постоянного сигнала **Constant**

Назначение: задает постоянный по уровню сигнал.

Параметры: **Constant value** – постоянная величина. **Interpret vector parameters as 1-D** – Интерпретировать вектор параметров как одномерный (при установленном флажке). Данный параметр встречается у большинства блоков библиотеки **Simulink**. В дальнейшем он рассматриваться не будет.

Значение константы может быть действительным или комплексным числом, вычисляемым выражением, вектором или матрицей.

Рис. 8 иллюстрирует применение этого источника и измерение его выходного сигнала с помощью цифрового индикатора **Display**.

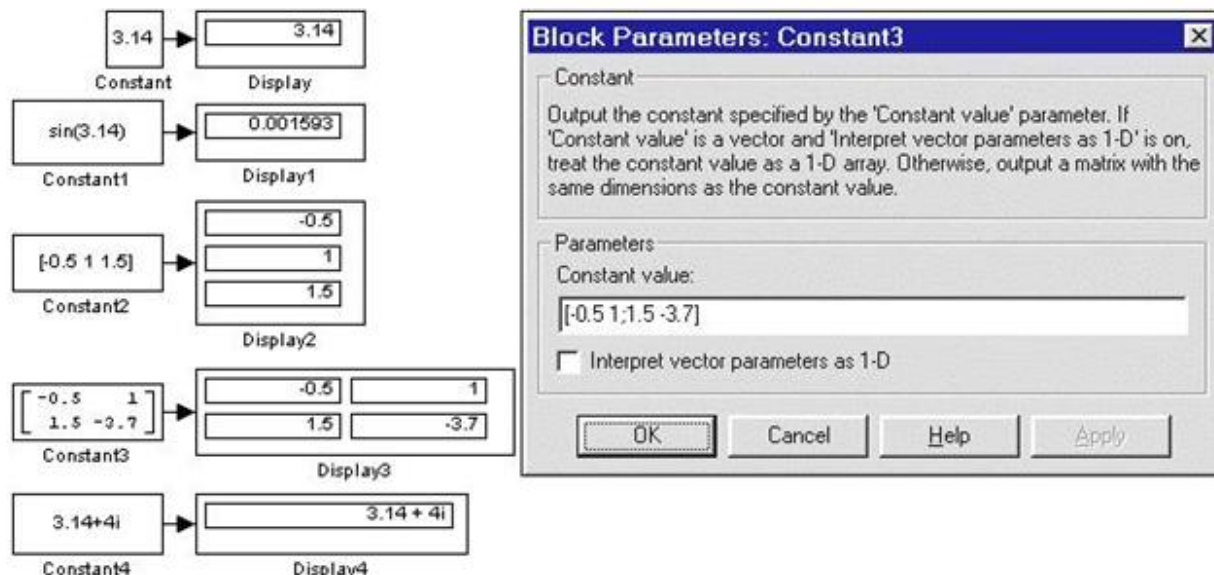


Рис. 8. Источник постоянного воздействия Constant

## Генератор ступенчатого сигнала **Step**

Назначение: формирует ступенчатый сигнал.

Параметры: **Step time** – время наступления перепада сигнала (с). **Initial value** – начальное значение сигнала. **Final value** – конечное значение сигнала.

Перепад может быть как в большую сторону (конечное значение больше чем начальное), так и в меньшую (конечное значение меньше чем начальное). Значения начального и конечного уровней могут быть не только положительными, но и отрицательными (например, изменение сигнала с уровня – 5 до уровня – 3).

На рис. 9 показано использование генератора ступенчатого сигнала.



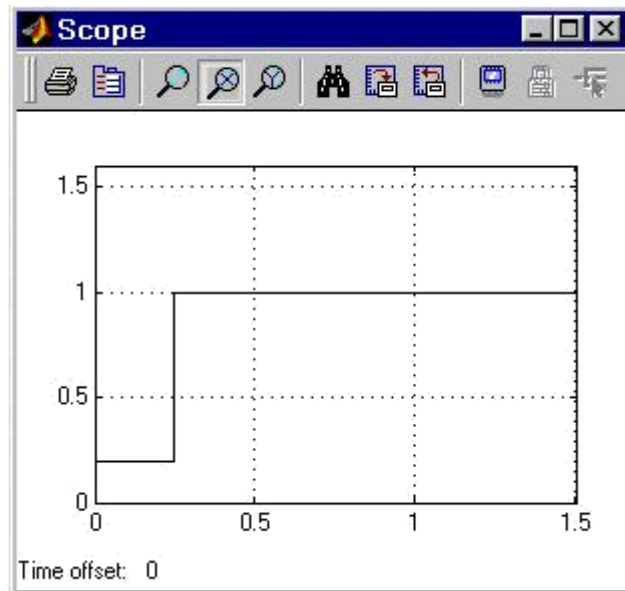
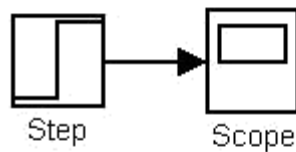


Рис. 9. Блок **Step**

### Генератор сигналов **Signal Generator**

Назначение: формирует один из четырех видов периодических сигналов:

**sine** – синусоидальный сигнал.

**square** – прямоугольный сигнал.

**sawtooth** – пилообразный сигнал.

**random** – случайный сигнал.

Параметры:

**Wave form** – вид сигнала.

**Amplitude** – амплитуда сигнала.

**Frequency** – частота (рад/с).

**Units** – единицы измерения частоты. Может принимать два значения:

**Hertz** – Гц;

**rad/sec** – рад/с.

На рис. 10. показано применение этого источника при моделировании прямоугольного сигнала.

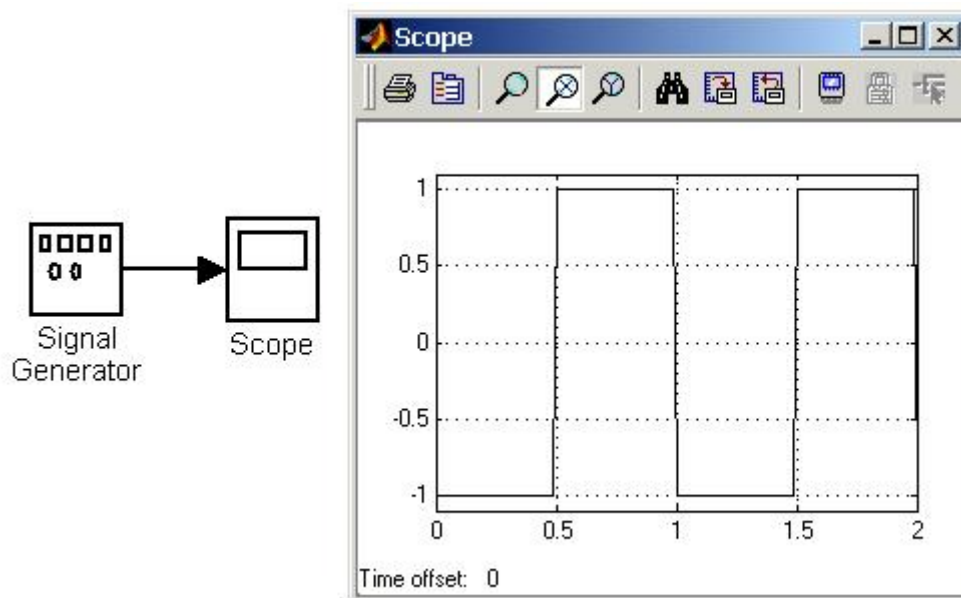


Рис. 10. Блок генератора сигналов  
Источник импульсного сигнала **Pulse Generator**

Назначение: формирование прямоугольных импульсов.

Параметры:

**Pulse Type** – способ формирования сигнала. Может принимать два значения:

**Time-based** – по текущему времени;

**Sample-based** – по величине модельного времени и количеству расчетных шагов.

**Amplitude** – амплитуда.

**Period** – период. Задается в секундах для **Time-based Pulse Type** или в шагах модельного времени для **Sample-based Pulse Type**.

**Pulse width** – ширина импульсов. Задается в процентах по отношению к периоду для **Time-based Pulse Type** или в шагах модельного времени для **Sample-based Pulse Type**.

**Phase delay** – фазовая задержка. Задается в секундах для **Time-based Pulse Type** или в шагах модельного времени для **Sample-based Pulse Type**.

**Sample time** – шаг модельного времени. Задается для **Sample-based Pulse Type**.

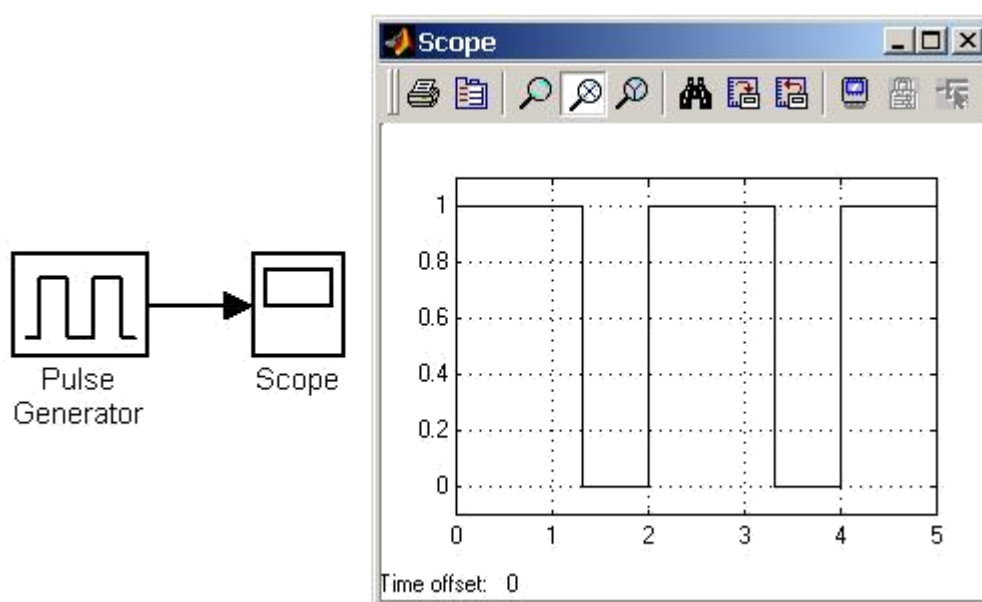


Рис. 11. Источник прямоугольных импульсов

## Источник временного сигнала **Clock**

Назначение: формирует сигнал, величина которого на каждом шаге расчета равна текущему времени моделирования.

Параметры:

**Decimation** – шаг, с которым обновляются показания времени на изображении источника (в том случае, если установлен флажок параметра **Display time**). Параметр задается как количество шагов расчета. Например, если шаг расчета модели в окне диалога **Simulation parameters** установлен равным 0,01 с, а параметр **Decimation** блока **Clock** задан равным 1000, то обновление показаний времени будет производиться каждые 10 с модельного времени.

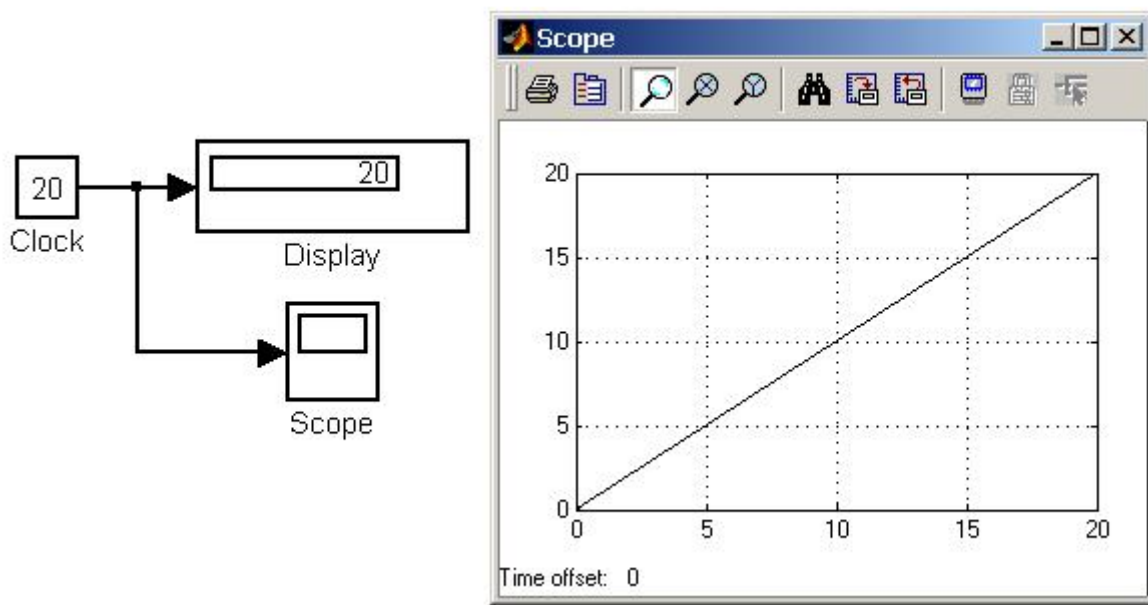


Рис. 12 Источник временного сигнал

**Display time** – отображение значения времени в блоке источника. На рис. 12 показан пример работы данного источника.

## Цифровой источник времени **Digital Clock**

Назначение: формирует дискретный временной сигнал.

Параметр:

**Sample time** – шаг модельного времени (с).

На рис. 13 показана работа источника **Digital Clock**.

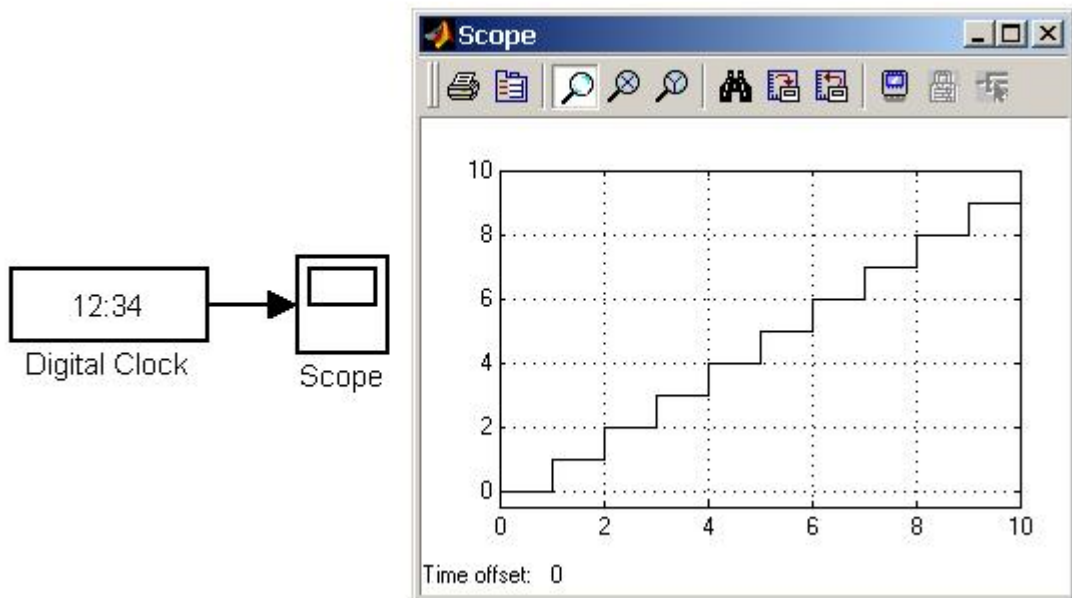


Рис. 13. Цифровой источник временного сигнала

### **Sinks – приемники сигналов** **Осциллограф Scope**

Назначение: строит графики исследуемых сигналов в функции времени. Позволяет наблюдать за изменениями сигналов в процессе моделирования.

Изображение блока и окно для просмотра графиков показаны на рис. 14.

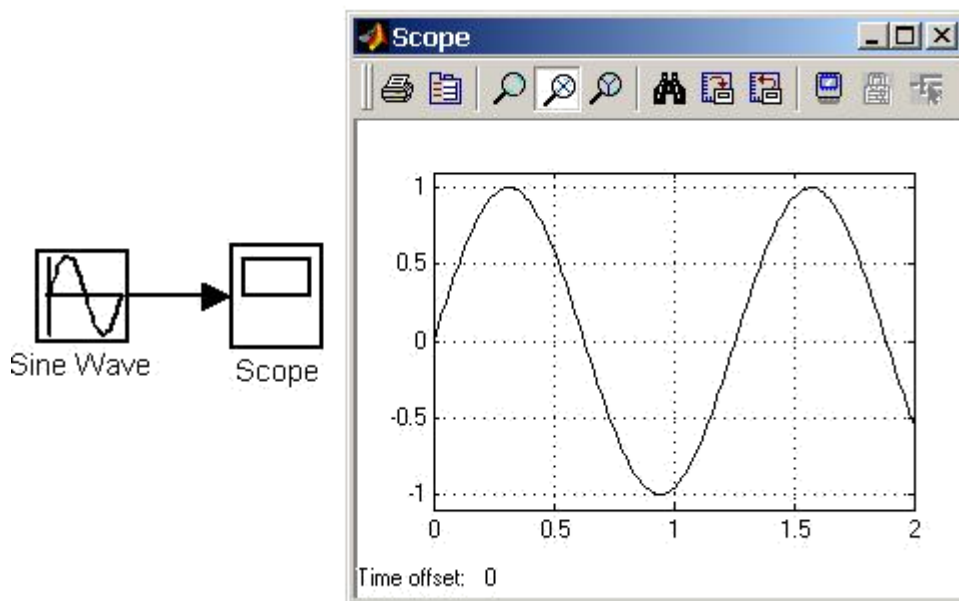


Рис. 14 Осциллограф Scope

Для того чтобы открыть окно просмотра сигналов, необходимо выполнить двойной щелчок левой клавишей мыши на изображении блока. Это можно сделать на любом этапе расчета (как до начала расчета, так и после него, а также во время расчета). В том случае, если на вход блока поступает векторный сигнал, то кривая для каждого элемента вектора строится отдельным цветом.

Настройка окна осциллографа выполняется с помощью панелей инструментов (рис. 15).

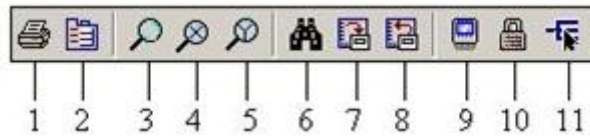


Рис. 15 Панель инструментов блока **Scope**

Панель инструментов содержит 11 кнопок:

**Print** – печать содержимого окна осциллографа;

**Parameters** – доступ к окну настройки параметров;

**Zoom** – увеличение масштаба по обеим осям;

**Zoom X-axis** – увеличение масштаба по горизонтальной оси;

**Zoom Y-axis** – увеличение масштаба по вертикальной оси;

**Autoscale** – автоматическая установка масштабов по обеим осям;

**Save current axes settings** – сохранение текущих настроек окна;




**Restore saved axes settings** – установка ранее сохраненных настроек окна;




**Floating scope** – перевод осциллографа в «свободный» режим;

**Lock/Unlock axes selection** – закрепить/разорвать связь между текущей координатной системой окна и отображаемым сигналом. Инструмент доступен, если включен режим **Floating scope**;

**Signal selection** – выбор сигналов для отображения. Инструмент доступен, если включен режим **Floating scope**.

Изменение масштабов отображаемых графиков можно выполнять несколькими способами:

Нажать соответствующую кнопку ( ,  или  ) и щелкнуть один раз левой клавишей мыши в нужном месте графика. Произойдет 2,5 кратное увеличение масштаба.

Нажать соответствующую кнопку ( ,  или  ) и, нажав левую клавишу мыши, с помощью динамической рамки или отрезка указать область графика для увеличенного изображения. Рис. 16 поясняет этот процесс.

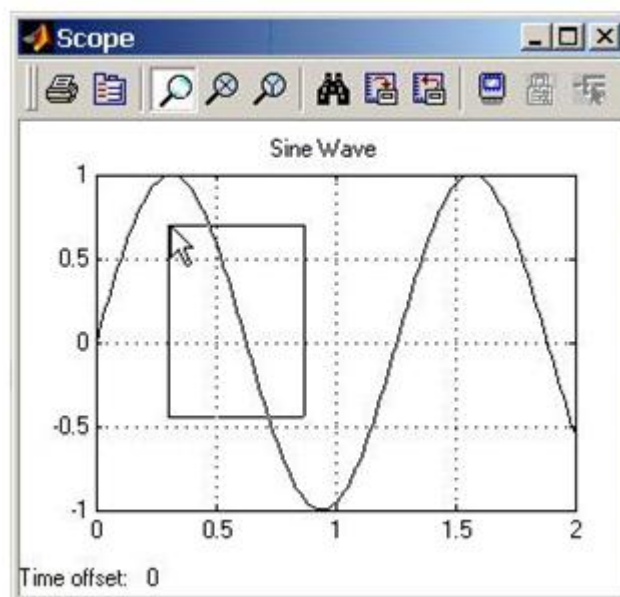



Рис. 16 Увеличение масштаба графика

Щелкнуть правой клавишей мыши в окне графиков и выбрать команду **Axes properties...** в контекстном меню. Откроется окно свойств графика, в котором с помощью параметров *Y-min* и *Y-max* можно указать предельные значения вертикальной оси. В этом же окне можно указать заголовок графика (**Title**), заменив выражение %<SignalLabel> в строке ввода. Окно свойств показано на рис. 17.



Рис. 17 Окно свойств графика

Параметры:

Параметры блока устанавливаются в окне '**Scope**' parameters, которое открывается с помощью инструмента  (**Parameters**) панели инструментов. Окно параметров имеет две вкладки:

**General** – общие параметры.

**Data history** – параметры сохранения сигналов в рабочей области **MATLAB**.

Вкладка общих параметров показана на рис. 18.



Рис. 18 Вкладка общих параметров **General**

На вкладке **General** задаются следующие параметры:

**Number of axes** – число входов (систем координат) осциллографа. При изменении этого параметра на изображении блока появляются дополнительные входные порты;

**Time range** – величина временного интервала, для которого отображаются графики. Если время расчета модели превышает заданное параметром **Time range**, то вывод графика производится порциями, при этом интервал отображения каждой порции графика равен заданному значению **Time range**.

**Tick labels** – вывод/скрытие осей и меток осей. Может принимать три значения (выбираются из списка):

**all** – подписи для всех осей,

**none** – отсутствие всех осей и подписей к ним,

**bottom axis only** – подписи горизонтальной оси только для нижнего графика;

**Sampling** – установка параметров вывода графиков в окне. Задает режим вывода расчетных точек на экран. При выборе **Decimation** кратность вывода устанавливается числом, задающим шаг выводимых расчетных точек;

**floating scope** – перевод осциллографа в «свободный» режим (при установленном флажке).

На вкладке **Data history** (рис. 19) задаются следующие параметры:

**Limit data points to last** – максимальное количество отображаемых расчетных точек графика. При превышении этого числа начальная часть графика обрезается. В том случае, если флажок параметра **Limit data points to last** не установлен, то **Simulink** автоматически увеличит значение этого параметра для отображения всех расчетных точек;

**Save data to workspace** – сохранение значений сигналов в рабочей области **MATLAB**.

**Variable name** – имя переменной для сохранения сигналов в рабочей области MATLAB.

**Format** – формат данных при сохранении в рабочей области MATLAB. Может принимать значения:

**Array** – массив,

**Structure** – структура,

**Structure with time** – структура с дополнительным полем «время».

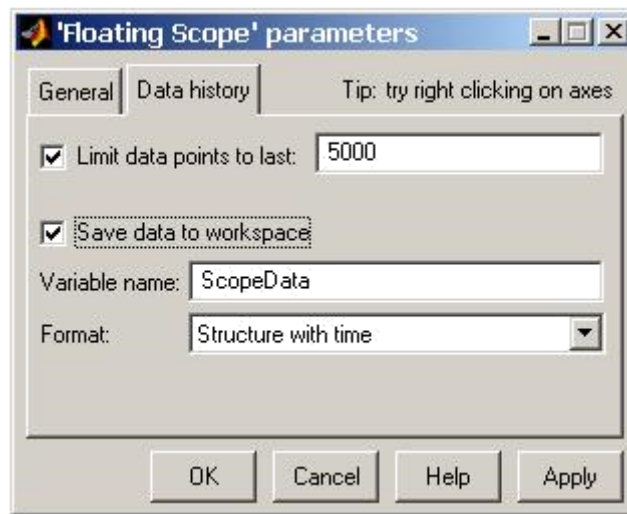


Рис. 19 Вкладка **Data history**

## Графопостроитель XY Graph

Назначение: строит график одного сигнала в функции другого (график вида  $Y(X)$ ).

Параметры:

**x-min** – минимальное значение сигнала по оси X;

**x-max** – максимальное значение сигнала по оси X;

**y-min** – минимальное значение сигнала по оси Y;

**y-max** – максимальное значение сигнала по оси Y;

**Sample time** – шаг модельного времени.

Блок имеет два входа. Верхний вход предназначен для подачи сигнала, который является аргументом (X), нижний – для подачи значений функции (Y).

На рис. 20, в качестве примера использования графопостроителя, показано построение фазовой траектории колебательного звена.

Графопостроитель можно использовать и для построения временных зависимостей. Для этого на первый вход следует подать временной сигнал с выхода блока **Clock**. Пример такого использования графопостроителя показан на рис. 21.

## Цифровой дисплей Display

Назначение: отображает значение сигнала в виде числа.

Параметры:

**Format** – формат отображения данных. Параметр **Format** может принимать следующие значения:

- **short** – 5 значащих десятичных цифр;
- **long** – 15 значащих десятичных цифр;
- **short\_e** – 5 значащих десятичных цифр и 3 символа степени десяти;
- **long\_e** – 15 значащих десятичных цифр и 3 символа степени десяти;



- **bank** – «денежный» формат. Формат с фиксированной точкой и двумя десятичными цифрами в дробной части числа.

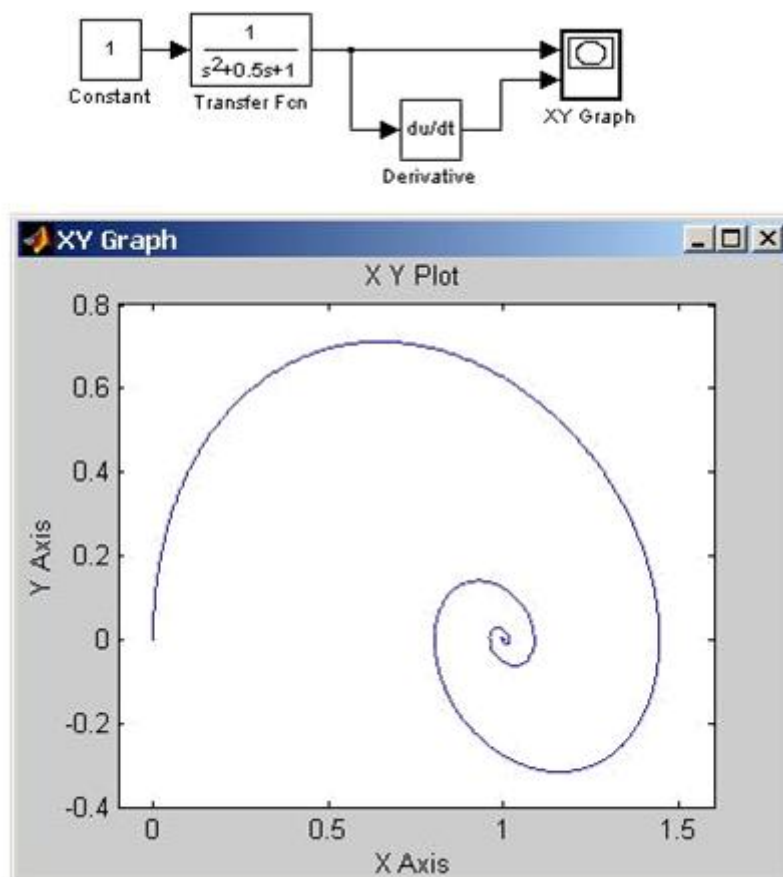


Рис. 20 Пример использования графопостроителя XY Graph

**Decimation** – кратность отображения входного сигнала. При *Decimation* = 1 отображается каждое значение входного сигнала, при *Decimation* = 2 – каждое второе значение, при *Decimation* = 3 – каждое третье значение и т.д.;

**Sample time** – шаг модельного времени. Определяет дискретность отображения данных;

**Floating display** (флажок)– перевод блока в «свободный» режим. В данном режиме входной порт блока отсутствует, а выбор сигнала для отображения выполняется щелчком левой клавиши мыши на соответствующей линии связи. В этом режиме для параметра расчета **Signal storage reuse** должно быть установлено значение off (вкладка **Advanced** в окне диалога **Simulation parameters...**). На рис. 21 показано применение блока **Display** с использованием различных вариантов параметра Format.



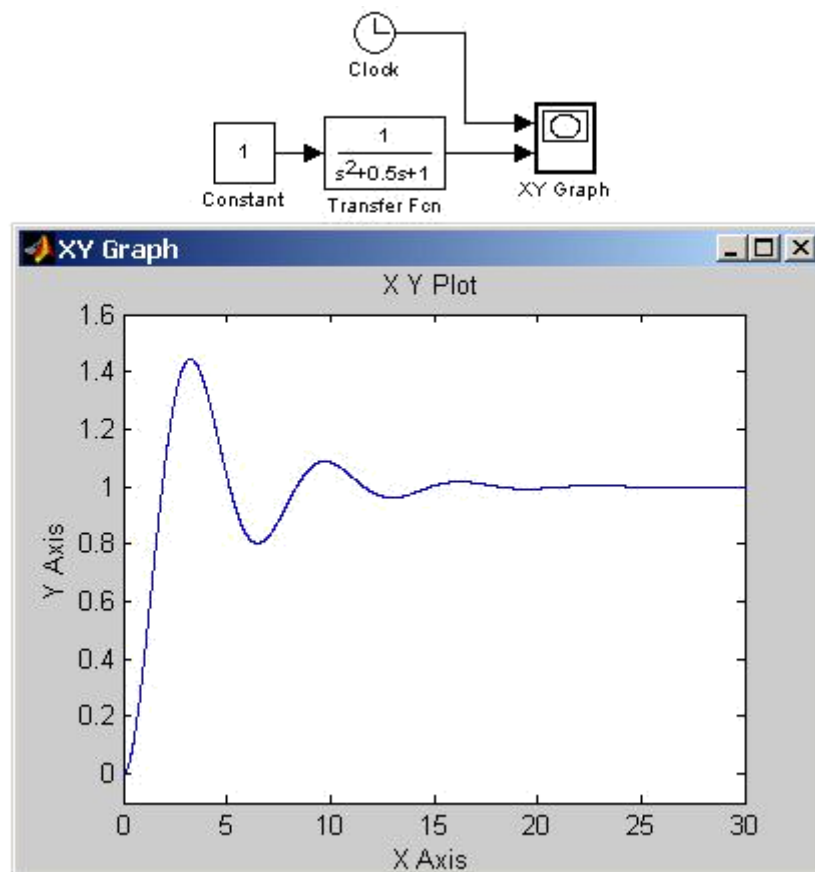


Рис. 21 Пример использования блока **XY Graph** для отображения временных зависимостей

Блок **Display** может использоваться для отображения не только скалярных сигналов, но также векторных, матричных и комплексных. Рис. 23 иллюстрирует это. Если все отображаемые значения не могут поместиться в окне блока, в правом нижнем углу блока появляется символ ↗, указывающий на необходимость увеличить размеры блока (см. блок **Display** на рис. 23).

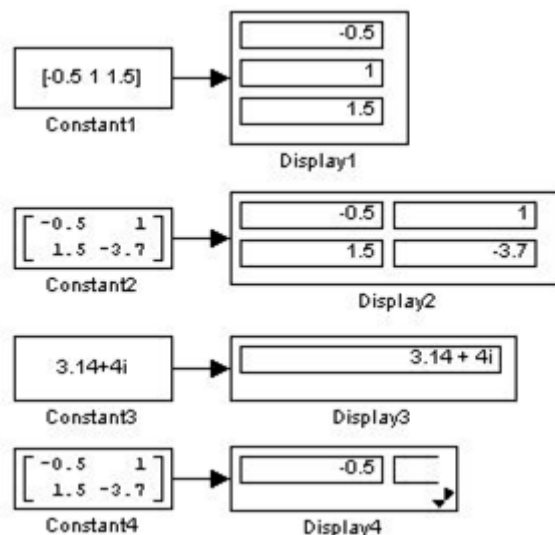


Рис. 22 Применение блока **Display** с использованием различных вариантов параметра **Format**

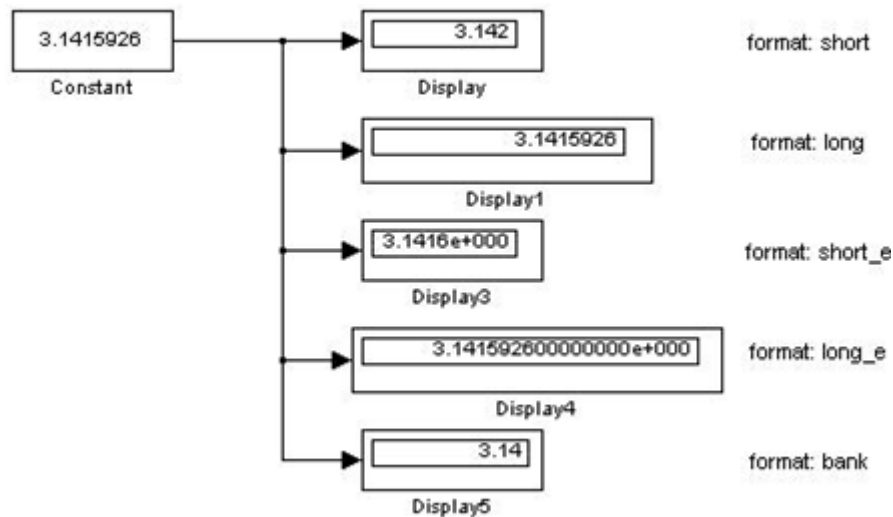


Рис. 23 Применение блока Display для отображения векторных, матричных и комплексных сигналов

### Блок остановки моделирования Stop Simulation

Назначение: обеспечивает завершение расчета, если входной сигнал блока становится не равным нулю.

Параметры: Нет.

При подаче на вход блока ненулевого сигнала Simulink выполняет текущий шаг расчета, а затем останавливает моделирование. Если на вход блока подан векторный сигнал, то для остановки расчета достаточно, чтобы один элемент вектора стал ненулевым. На рис. 24 показан пример использования данного блока. В примере остановка расчета происходит, если выходной сигнал блока Transfer Function становится большим или равным 0,99.

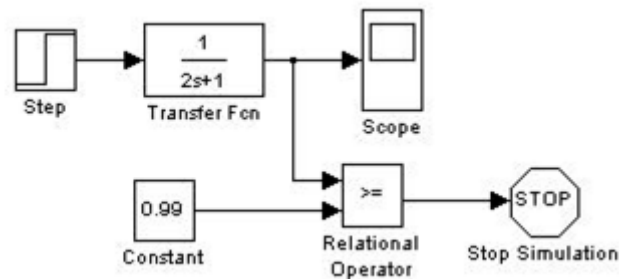


Рис. 24 Применение блока Stop Simulation

### Continuous – непрерывные элементы

Раздел содержит элементы трех групп:

- блоки общего назначения (интеграторы, дифференциаторы);
- блоки задержки сигнала;
- блоки линейных стационарных звеньев.

### Math блоки математических операций

Блок вычисления модуля Abs

Назначение: выполняет вычисление абсолютного значения величины сигнала.

Параметры:

**Saturate on integer overflow** (флажок) –подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.

Пример использования блока **Abs**, вычисляющего модуль текущего значения синусоидального сигнала, показан на рис. 25.

Блок **Abs** может использоваться также для вычисления модуля сигнала комплексного типа.

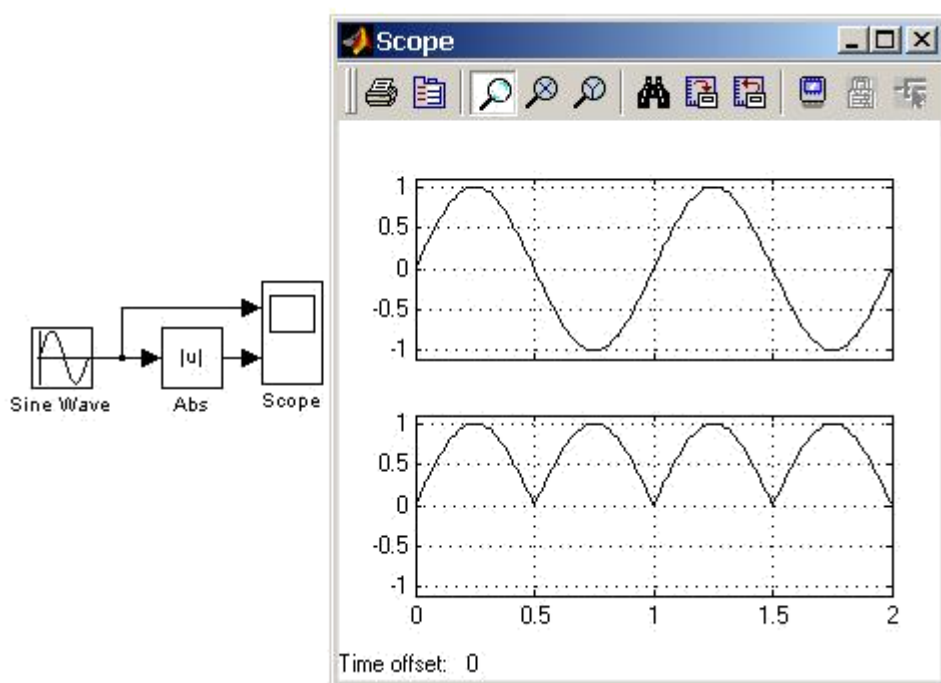


Рис. 25 Пример использования блока **Abs**

### Блок вычисления суммы **Sum**

Назначение: выполняет вычисление суммы текущих значений сигналов.

Параметры:

**Icon shape** – форма блока. Выбирается из списка.

- **round** – окружность,
- **rectangular** – прямоугольник.

**List of sign** – список знаков. В списке можно использовать следующие знаки: + (плюс), - (минус) и | (разделитель знаков).

**Saturate on integer overflow** (флажок) – подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.

Количество входов и операция (сложение или вычитание) определяется списком знаков параметра **List of sign**, при этом метки входов обозначаются соответствующими знаками. В параметре **List of sign** можно также указать число входов блока. В этом случае все входы будут суммирующими.

Если количество входов блока превышает три, то удобнее использовать блок **Sum** прямоугольной формы.

Блок может использоваться для суммирования скалярных, векторных или матричных сигналов. Типы суммируемых сигналов должны совпадать. Нельзя, например, подать на один и тот же суммирующий блок сигналы целого и действительного типов.

Если количество входов блока больше, чем один, то блок выполняет поэлементные операции над векторными и матричными сигналами. При этом количество элементов в матрице или векторе должно быть одинаковым.

Если в качестве списка знаков указать цифру 1 (один вход), то блок можно использовать для определения суммы элементов вектора.

Примеры использования блока **Sum** показаны на рис. 26.

Блок умножения **Product**

Назначение: выполняет вычисление произведения текущих значений сигналов.

Параметры:

**Number of inputs** – количество входов. Может задаваться как число или как список знаков. В списке знаков можно использовать знаки «×» (умножить) и «/» (разделить).

**Multiplication** – способ выполнения операции. Может принимать значения (из списка):

- **Element-wise** – поэлементный;
- **Matrix** – матричный.

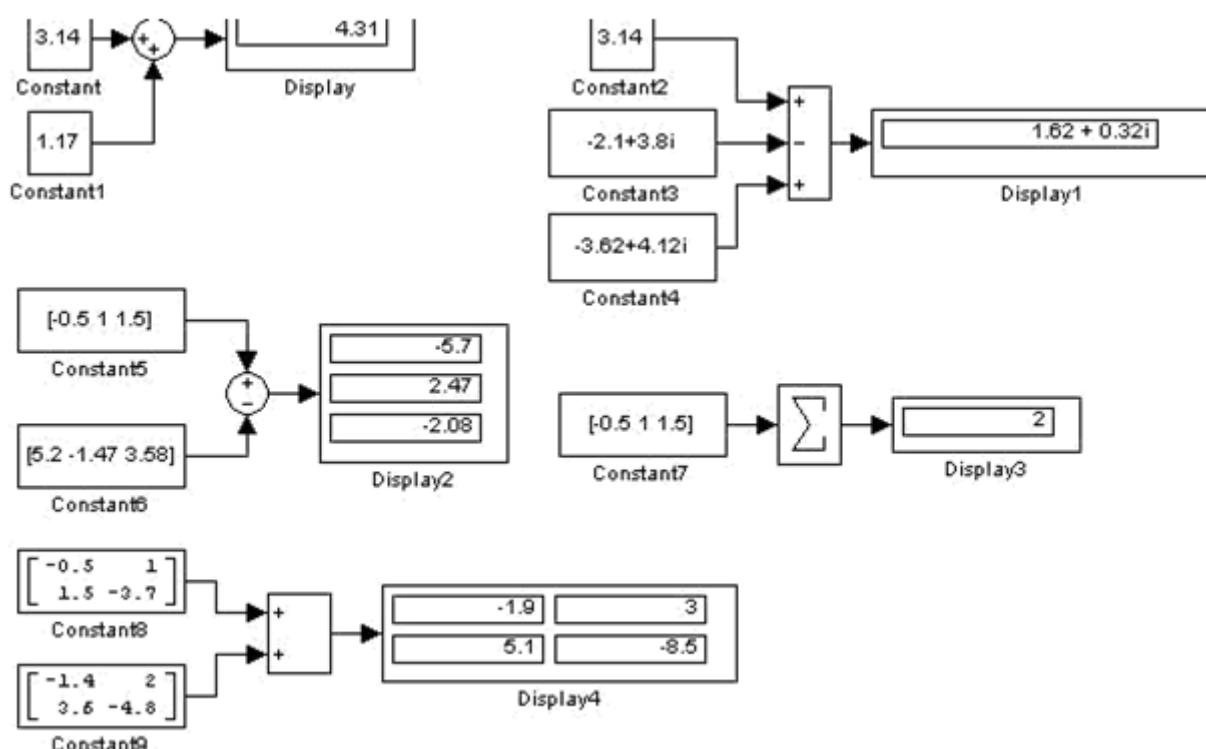


Рис. 26 Примеры использования блока **Sum**

**Saturate on integer overflow** (флажок) – подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.

Если параметр Number of inputs задан списком, включающим кроме знаков умножения также знаки деления, то метки входов будут обозначены символами соответствующих операций.

Блок может использоваться для операций умножения или деления скалярных векторных или матричных сигналов. Типы входных сигналов блока должны совпадать. Если в качестве количества входов указать цифру 1 (один вход), то блок можно использовать для определения произведения элементов вектора.

Примеры использования блока **Product** при выполнении скалярных и поэлементных операций показаны на рис. 27.

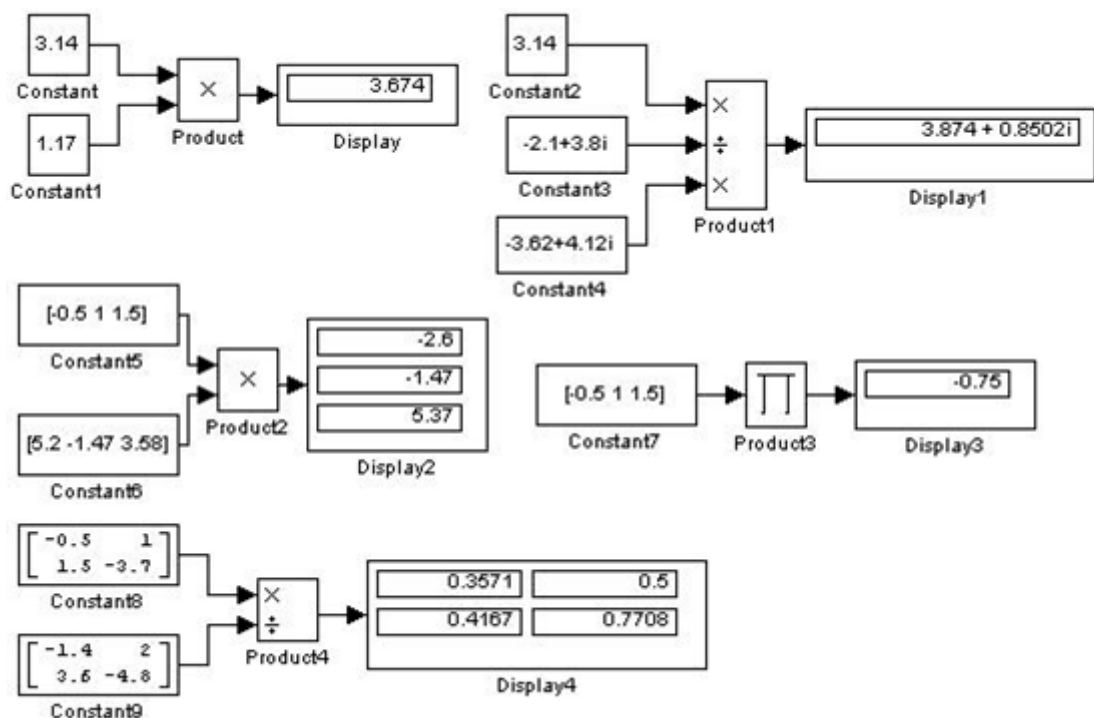
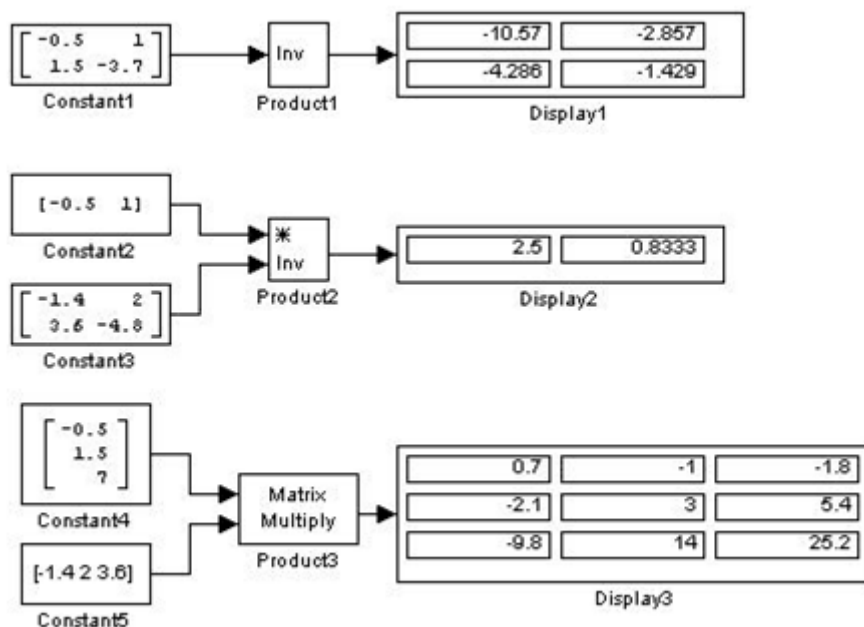


Рис. 27 Примеры использования блока **Product** при выполнении скалярных и поэлементных операций

При выполнении матричных операций необходимо соблюдать правила их выполнения. Например, при умножении двух матриц необходимо, чтобы количество строк первой матрицы равнялось количеству столбцов второй матрицы. Примеры использования блока **Product** при выполнении матричных операций показаны на рис. 28. В примере показаны операции формирования обратной матрицы, деление матриц, а также умножение матриц.



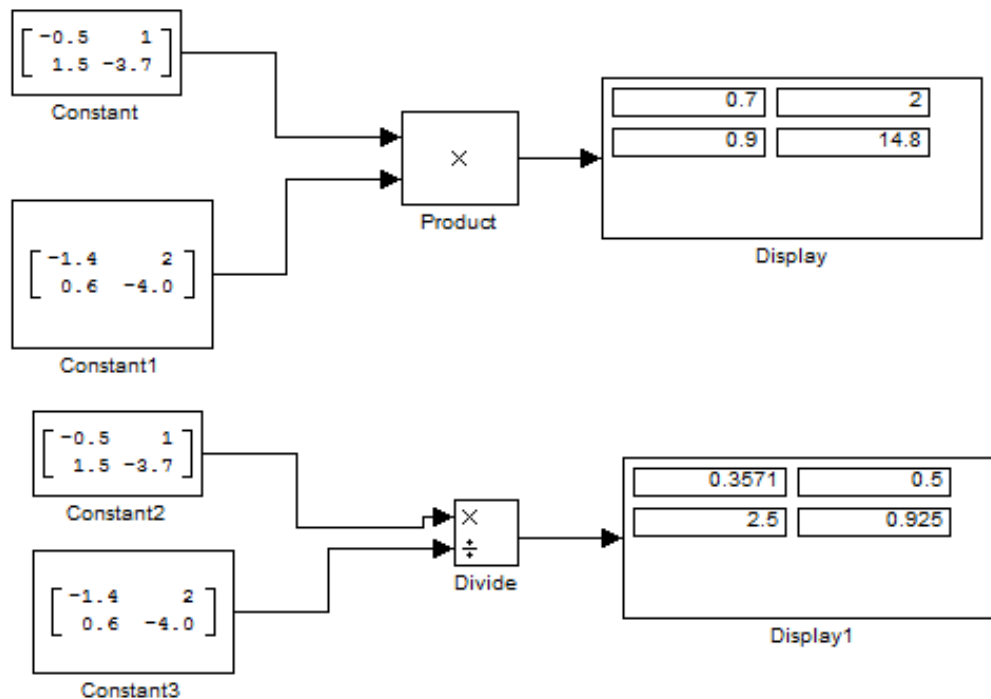


Рис. 28 Примеры использования блока Product при выполнении матричных операций

### Усилители **Gain** и **Matrix Gain**

Назначение: выполняют умножение входного сигнала на постоянный коэффициент.

Параметры:

**Gain** – коэффициент усиления.

**Multiplication** – способ выполнения операции. Может принимать значения (из списка):

- **Element-wise  $K*u$**  – поэлементный;
- **Matrix  $K*u$**  – матричный. Коэффициент усиления является левосторонним операндом;
- **Matrix  $u*K$**  – матричный. Коэффициент усиления является правосторонним операндом.

**Saturate on integer overflow** (флажок) – подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.

Параметр блока **Gain** может быть положительным или отрицательным числом, как больше, так и меньше 1. Коэффициент усиления можно задавать в виде скаляра, матрицы или вектора, а также в виде вычисляемого выражения. В том случае если параметр **Multiplication** задан как **Elementwise  $K*u$** , то блок выполняет операцию умножения на заданный коэффициент скалярного сигнала или каждого элемента векторного сигнала. В противном случае блок выполняет операцию матричного умножения сигнала на коэффициент заданный матрицей. По умолчанию коэффициент усиления является действительным числом типа **double**. Для операции поэлементного усиления входной сигнал может быть скалярным, векторным или матричным любого типа, за исключением логического (**boolean**). Элементы вектора должны иметь одинаковый тип сигнала. Выходной сигнал блока будет иметь тот же самый тип, что и входной сигнал. Параметр блока **Gain** может быть скаляром, вектором или матрицей любого типа, за исключением логического (**boolean**). При вычислении выходного сигнала блок Gain использует следующие правила:

- Если входной сигнал действительного типа, а коэффициент усиления комплексный, то выходной сигнал будет комплексным;
- Если тип входного сигнала отличается от типа коэффициента усиления, то **Simulink** пытается выполнить приведение типа коэффициента усиления к типу входного сигнала. В том случае, если такое приведение невозможно, то расчет будет остановлен с выводом сообщения об ошибке. Такая ситуация может возникнуть, например, если входной сигнал есть беззнаковое целое (**uint8**), а параметр Gain задан отрицательным числом.

Примеры использования блока **Gain** при выполнении скалярных и поэлементных операций показаны на рис. 29. Для операций матричного усиления (матричного умножения входного сигнала на заданный коэффициент) входной сигнал и коэффициент усиления должны быть скалярными, векторными или матричными значениями комплексного или действительного типа **single** или **double**. Примеры использования блока **Matrix Gain** при выполнении матричных операций показаны на рисунке 29.

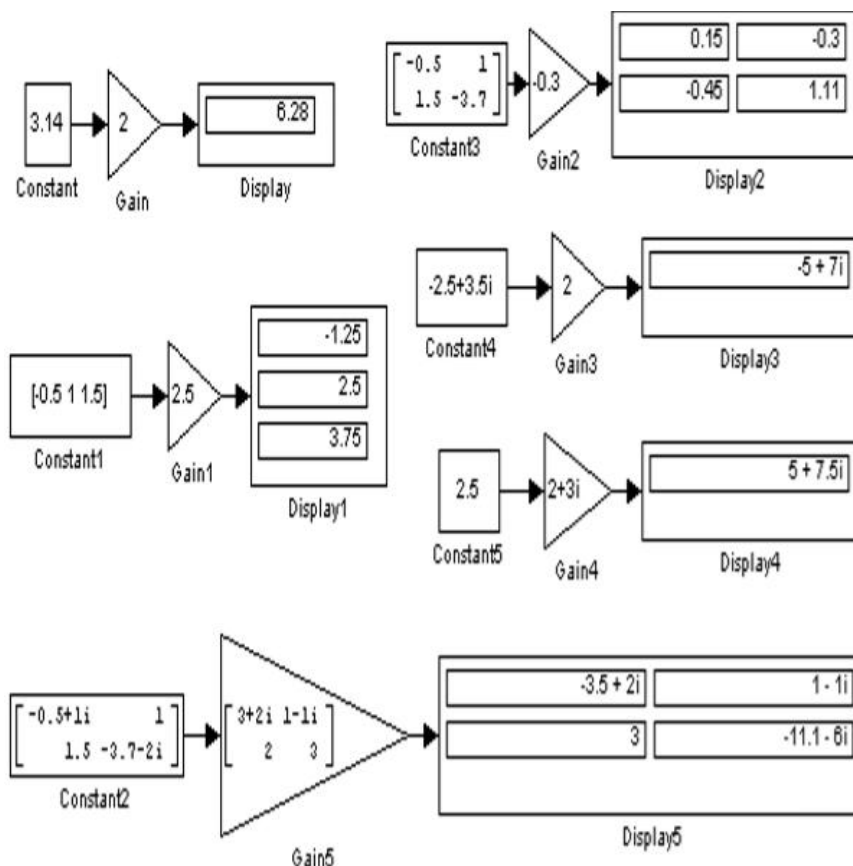


Рис. 29 Примеры использования блока Gain

### Блок вычисления математических функций **Math Function**

Назначение: выполняет вычисление математической функции.

Параметры:

**Function** – вид вычисляемой функции (выбирается из списка):

- **exp** – экспоненциальная функция
- **log** – функция натурального логарифма
- **10^u** – вычисление степени 10
- **log10** – функции логарифма
- **magnitude^2** – Вычисление квадрата модуля входного сигнала
- **square** – вычисление квадрата входного сигнала
- **sqrt** – квадратный корень
- **pow** – возведение в степень
- **conj** – вычисление комплексно-сопряженного числа
- **reciprocal** – вычисление частного от деления входного сигнала на 1
- **hypot** – вычисление корня квадратного из суммы квадратов входных сигналов

(гипотенузы прямоугольного треугольника по значениям катетов)

- **rem** – функция, вычисляющая остаток от деления первого входного сигнала на второй
- **mod** – функция, вычисляющая остаток от деления с учетом знака
- **transpose** – транспонирование матрицы
- **hermitian** – вычисление эрмитовой матрицы.

**Output signal type** – тип выходного сигнала (выбирается из списка):

- **auto** – автоматическое определение типа;
- **real** – действительный сигнал;
- **complex** – комплексный сигнал.

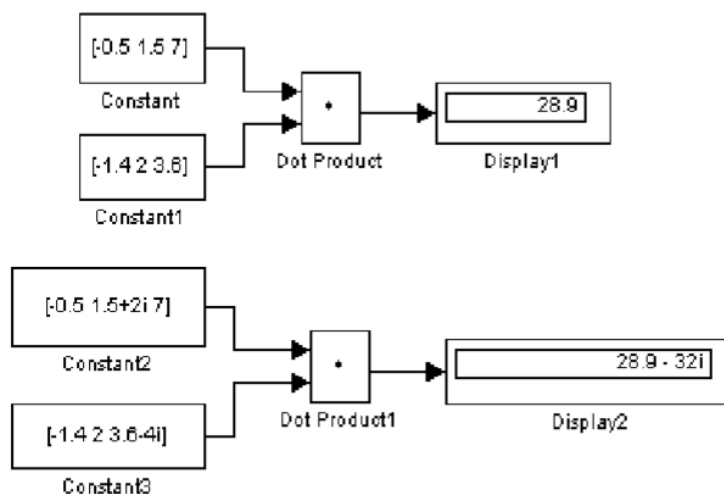


Рис. 30. Примеры использования **Dot Product**

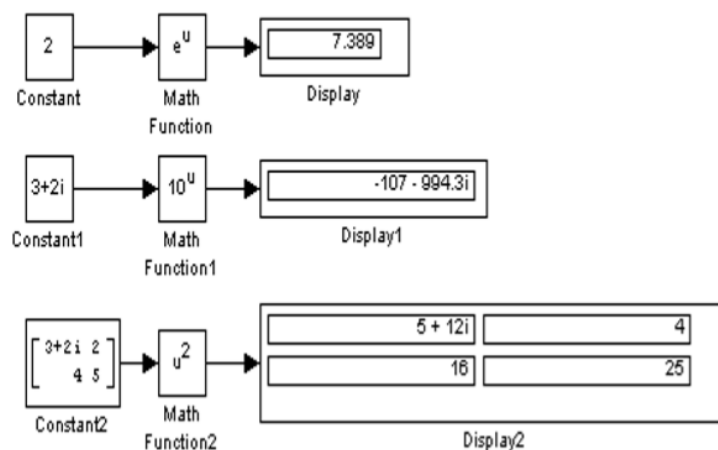


Рис. 31. Примеры использования **Math Function**

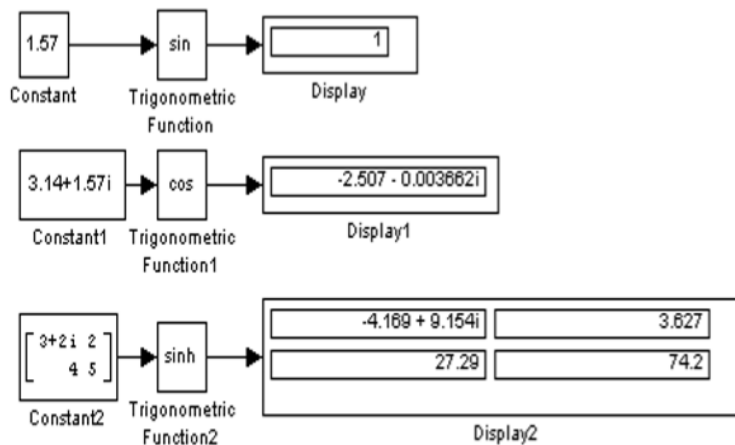


Рис. 32. Примеры использования блока **Trigonometric Function**

## ПРАКТИЧЕСКАЯ ЧАСТЬ

Предполагает:

1. Использование приведенных примеров в качестве знакомства с пакетом.



2. Выполнение элементов моделирования.
3. Оформление отчета в соответствии с требованиями по кафедре ИС.
4. Устные ответы на вопросы преподавателя по данному и лекционному материалу.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. По текущим темам лекционного курса.
2. по практической части, приведенного ранее материала.

## ЛИТЕРАТУРА

1. Ануфриев И. Е. Самоучитель MatLab 5.3/6.x. -СПб.: БХВ-Петербург, 2003. 736 с.
2. Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н. MATLAB 7. – СПб.: БХВ – Петербург, 2005. – 1104 с.
3. Васильев В.В., Симак Л.А., Рыбникова А.М. Математическое и компьютерное моделирование процессов и систем в среде MATLAB/SIMULINK. Учебное пособие для студентов и аспирантов / В.В. Васильев, Л.А. Симак, А.М. Рыбникова. – К.: НАН Украины, 2008. – 91 с.
4. Дьяконов В., Круглов В. Математические пакеты расширения MATLAB. -СПб.: Питер, 2001. 560 с.
5. Дьяконов В.П. MATLAB 6/6.1/6.5 + SIMULINK 4/5 в математике и моделировании. -М.: Солон-Пресс, 2003. 576 с.
6. Дзэни Дж., Хароган Т. SIMULINK 4. Секреты мастерства. -М.: Бином, Лаборатория знаний. 2003. 403 с.
7. Кетков Ю., Кетков А., Шульц М. MATLAB 6.x: программирование численных методов. - СПб.: БХВ-Петербург, 2004. 742 с.
8. Мироновский Л. А. Моделирование линейных систем. Учеб. пособие с грифом УМО. СПб. ГУАП, 2009. – 244 с.
9. Лазарев Ю.Ф. MatLAB 5.x. – К.: Издательская группа BHV, 2000. – 384 с.
10. Мироновский Л.А., Петрова К.Ю. Введение в MATLAB: Учеб. пособие. СПб., ГУАП. 2006. 163с.
11. Конев В.Ю., Мироновский Л.А. Основные функции пакета MATLAB. Учеб. пособие. СПб. ГААП.1992,75с.; 1994. 79с.
12. Мироновский Л.А. Моделирование динамических систем. Учеб. пособие. СПб., ГААП. 1992. 92 с.
13. Наместников А.М. Разработка имитационных моделей в среде MATLAB: Методические указания для студентов специальностей 01719, 351400 / Ульяновск, УлГТУ, 2004. – 72 с.
14. Половко А.М., Бутусов П.Н. Matlab для студентов. - СПб: БХВ-Петербург, 2005. 320 с.
15. Потемкин В. Г. MATLAB 6: среда проектирования инженерных приложений. М.: Диалог-МИФИ, 2003. 448 с.
16. Сергиенко А. Цифровая обработка сигналов. -СПб.: Питер, 2002. 606 с.
17. Терёхин В.В. Моделирование в системе MATLAB: Учебное пособие Часть 1. Основы работы в MATLAB /Кемеровский государственный университет. – Новокузнецк: Кузбассвуиздат, 2005. -376с.
18. Терёхин В.В. Моделирование в системе MATLAB: Учебное пособие Часть 2. Simulink /Кемеровский государственный университет. – Новокузнецк: Кузбассвуиздат, 2004. -376 с.