



О.І. Михальов, К.Ю. Островська

**МЕТОДИЧНІ ВКАЗІВКИ**  
**до виконання контрольних робіт з дисципліни**

**“Комп’ютерне моделювання виробничих процесів”**

для студентів зі спеціальності  
122 «Комп’ютерні науки»  
галузі знань 12 «Інформаційні технології»

УДК 681.3.06+519.68

Методичні вказівки до виконання лабораторних та контрольних робіт з дисципліни “Комп’ютерне моделювання виробничих процесів”. Для студентів зі спеціальності 122 «Комп’ютерні науки» галузі знань 12 «Інформаційні технології». - Укл.: О.І. Михальов, К.Ю. Островська. Під ред. О.І. Михальова. – Дніпро: НМетАУ, 2019. – 48 с.

Методичні вказівки комплексом навчально-методичних матеріалів з дисципліни “Комп’ютерне моделювання виробничих процесів”, в якій наведені теоретичні матеріали та розглянуті приклади по застосуванню графічного інтерфейсу GUI та пакету SIMULINK для комп’ютерного моделювання виробничих процесів. Практична частина містить завдання для самостійного виконання лабораторних робіт.

Методичні вказівки призначені для студентів зі спеціальності 122 «Комп’ютерні науки» галузі знань 12 «Інформаційні технології», а також для слухачів курсів підвищення кваліфікації, студентів і аспірантів інших спеціальностей.

Укладачі: О.І. Михальов, д-р техн. наук, проф.  
К.Ю. Островська, канд. техн. наук, доц.

Відповідальний за випуск О.І. Михальов, д-р техн. наук, проф.  
Рецензент В.І. Корсун, д-р техн. наук, проф., (НГУ)

Друкується за авторською редакцією.

Затверджено на засіданні кафедри інформаційних технологій і систем, протокол № 1 від 29.08.2019.

Підписано до друку 30.08.2019. Формат 60x84 1/19. Папір типогр. Друк  
різограф. Облік.-вид. арк 3,93. Умов. друк. арк. 3,44.  
Тираж 100 пр. Замовл. № 01/19.

Національна металургійна академія України.  
49600, Дніпро-5, пр. Гагаріна, 4

---

ДНВП “Системні технології”

## ЗМІСТ

<b><u>Лабораторна робота №1</u></b>	
<b>Simulink. Ознайомлення з інтерфейсом. Загальні відомості. Опис бібліотек.</b>	4
<b><u>Лабораторна робота №2</u></b>	
<b>Побудова моделі з нелінійним елементом в цепі зворотного зв'язку.</b>	36
<b><u>Лабораторна робота №3</u></b>	41
<b>Моделювання роботи ідеального обмежника сигналів.</b>	
<b><u>Завдання на контрольну роботу</u></b>	44

## Лабораторная работа №1 SIMULINK. ОЗНАЙОМЛЕННЯ З ІНТЕРФЕЙСОМ. ЗАГАЛЬНІ ВІДОМОСТІ. ОПИС БІБЛІОТЕК.

**Цель:** Изучить основные возможности ППП SIMULINK

**Simulink** является приложением к пакету MATLAB. При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым, пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты. При этом, в отличие от классических способов моделирования, пользователю не нужно досконально изучать язык программирования и численные методы математики, а достаточно общих знаний требующихся при работе на компьютере и, естественно, знаний той предметной области в которой он работает.

Simulink является достаточно самостоятельным инструментом MATLAB и при работе с ним совсем не требуется знать сам MATLAB и остальные его приложения. С другой стороны доступ к функциям MATLAB и другим его инструментам остается открытым и их можно использовать в Simulink. Часть входящих в состав пакетов имеет инструменты, встраиваемые в Simulink (например, LTI-Viewer приложения Control System Toolbox – пакета для разработки систем управления). Имеются также дополнительные библиотеки блоков для разных областей применения (например, Power System Blockset – моделирование электротехнических устройств, Digital Signal Processing Blockset – набор блоков для разработки цифровых устройств и т.д). При работе с Simulink пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков.

При моделировании пользователь может выбирать метод решения дифференциальных уравнений, а также способ изменения модельного времени (с фиксированным или переменным шагом). В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки Simulink. Результаты моделирования могут быть представлены в виде графиков или таблиц.

Преимущество Simulink заключается также в том, что он позволяет пополнять библиотеки блоков с помощью подпрограмм написанных как на языке MATLAB, так и на языках C ++, Fortran и Ada.

### ***Запуск SIMULINK***

Для запуска программы необходимо предварительно запустить пакет MATLAB. Основное окно пакета MATLAB показано на Рис. 1. Там же показана подсказка появляющаяся в окне при наведении указателя мыши на ярлык Simulink в панели инструментов.

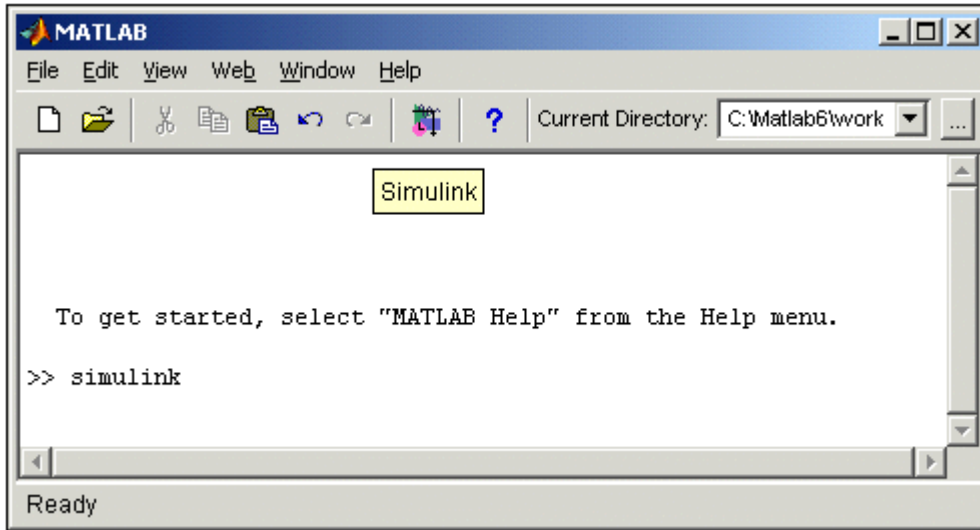



Рис 1. Основное окно программы MATLAB

После открытия основного окна программы MATLAB нужно запустить программу Simulink. Это можно сделать одним из трех способов:

- Нажать кнопку  (Simulink) на панели инструментов командного окна MATLAB.
- В командной строке главного окна MATLAB напечатать Simulink и нажать клавишу Enter на клавиатуре.
- Выполнить команду Open... в меню File и открыть файл модели (mdl - файл).

Последний вариант удобно использовать для запуска уже готовой и отлаженной модели, когда требуется лишь провести расчеты и не нужно добавлять новые блоки в модель. Использование первого и второго способов приводит к открытию окна обозревателя разделов библиотеки Simulink (рис. 2).

### ***Обзор разделов библиотеки SIMULINK***

Окно обозревателя библиотеки блоков содержит следующие элементы (Рис. 2):

1. Заголовок, с названием окна – **Simulink Library Browser**.
2. Меню, с командами **File, Edit, View, Help**.
3. Панель инструментов, с ярлыками наиболее часто используемых команд.
4. Окно комментария для вывода поясняющего сообщения о выбранном блоке.
5. Список разделов библиотеки, реализованный в виде дерева.
6. Окно содержимого раздела библиотеки (список вложенных разделов библиотеки или блоков).
7. Строка состояния, содержащая подсказку по выполняемому действию.

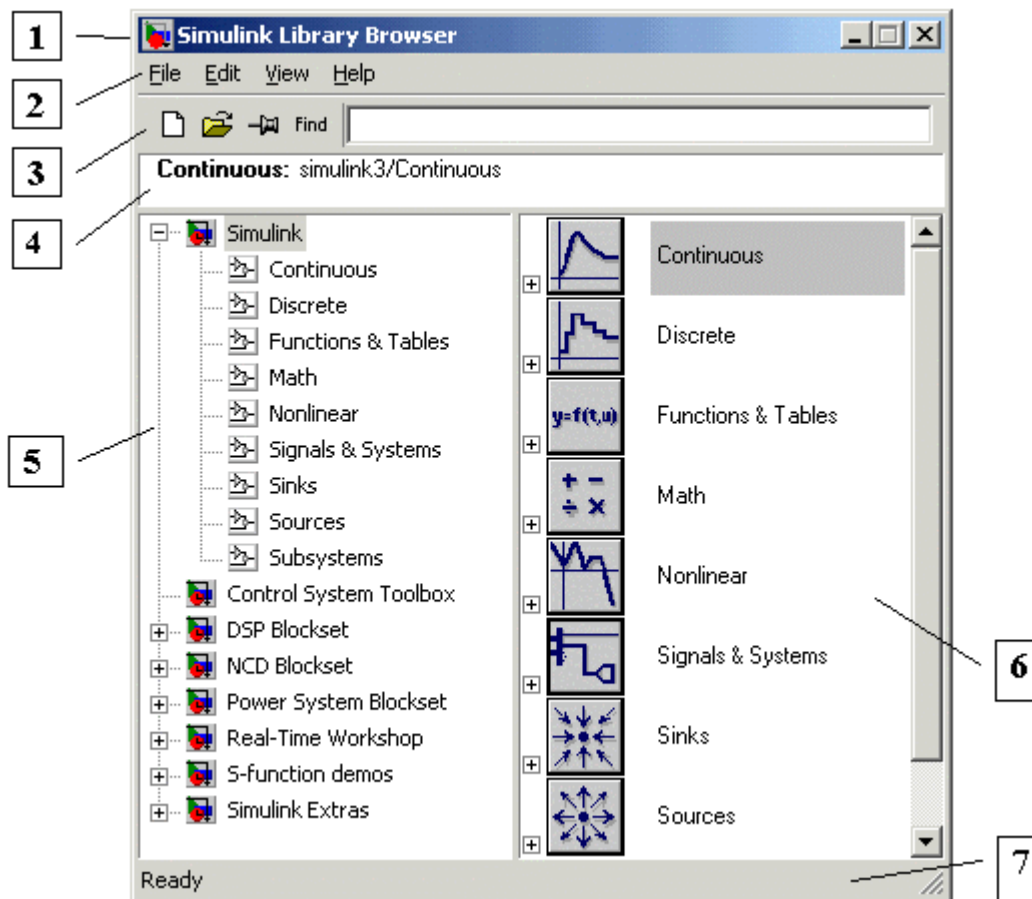


Рис 2. Окно обозревателя разделов библиотеки **Simulink**

На рис. 2 выделена основная библиотека Simulink (в левой части окна) и показаны ее разделы (в правой части окна).

Библиотека Simulink содержит следующие основные разделы:

1. **Continuous** – линейные блоки.
2. **Discrete** – дискретные блоки.
3. **Functions & Tables** – функции и таблицы.
4. **Math** – блоки математических операций.
5. **Nonlinear** – нелинейные блоки.
6. **Signals & Systems** – сигналы и системы.
7. **Sinks** - регистрирующие устройства.
8. **Sources** — источники сигналов и воздействий.
9. **Subsystems** – блоки подсистем.

Список разделов библиотеки Simulink представлен в виде дерева. При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое (Рис. 3).

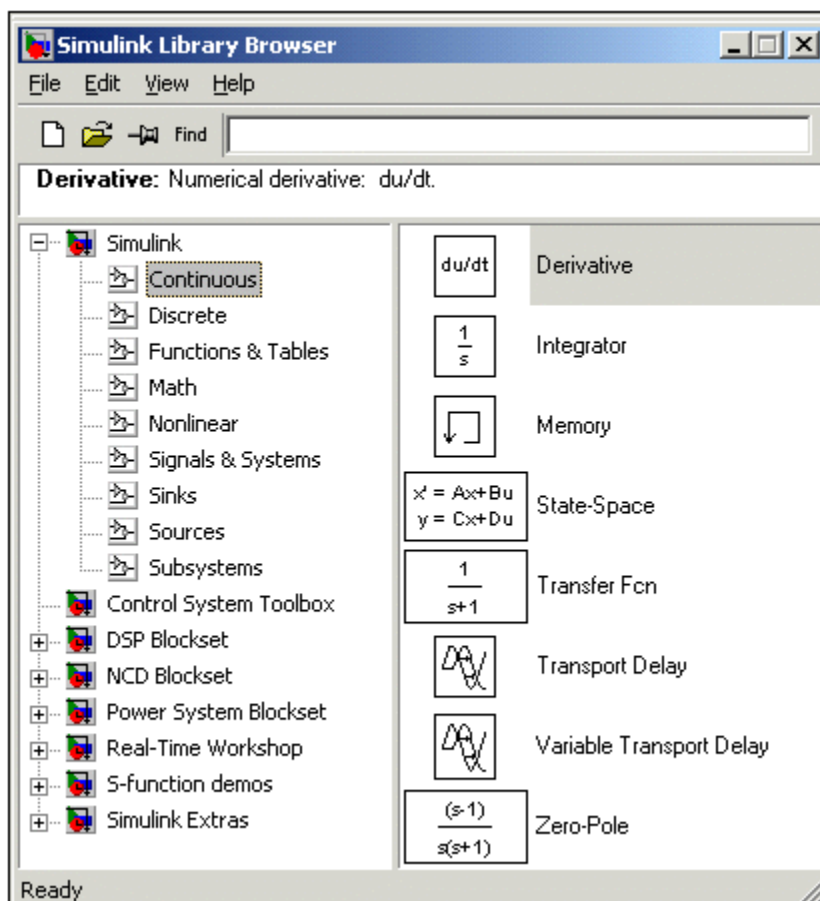


Рис 3. Окно обозревателя с набором блоков раздела библиотеки

Для работы с окном используются команды собранные в меню. Меню обозревателя библиотек содержит следующие пункты:

- **File (Файл)** — Работа с файлами библиотек.
- **Edit (Редактирование)** — Добавление блоков и их поиск (по названию).
- **View (Вид)** — Управление показом элементов интерфейса.
- **Help (Справка)** — Вывод окна справки по обозревателю библиотек.

Для работы с обозревателем можно также использовать кнопки на панели инструментов (Рис.6.4).



Рис 4. Панель инструментов обозревателя разделов библиотек


Кнопки панели инструментов имеют следующее назначение:

1. Создать новую S-модель (открыть новое окно модели).
2. Открыть одну из существующих S-моделей.
3. Изменить свойства окна обозревателя. Данная кнопка позволяет установить режим отображения окна обозревателя "поверх всех окон". Повторное нажатие отменяет такой режим.

4. Поиск блока по названию (по первым символам названия). После того как блок будет найден, в окне обозревателя откроется соответствующий раздел библиотеки, а блок будет выделен. Если же блок с таким названием отсутствует, то в окне комментария будет выведено сообщение **Not found <имя блока>** (Блок не найден).

### Создание модели

Для создания модели в среде SIMULINK необходимо последовательно выполнить ряд действий:

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. 5.

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 6 показано окно модели, содержащее блоки.

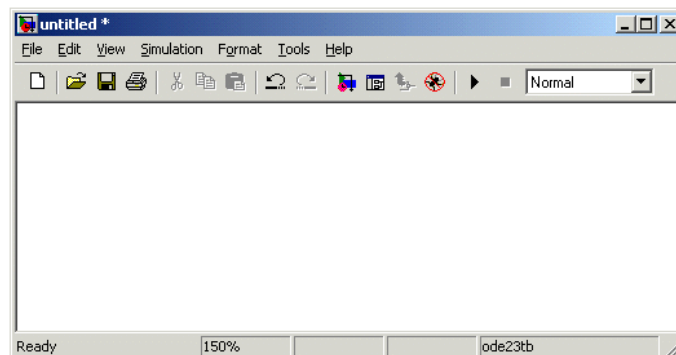


Рис 5. Пустое окно модели

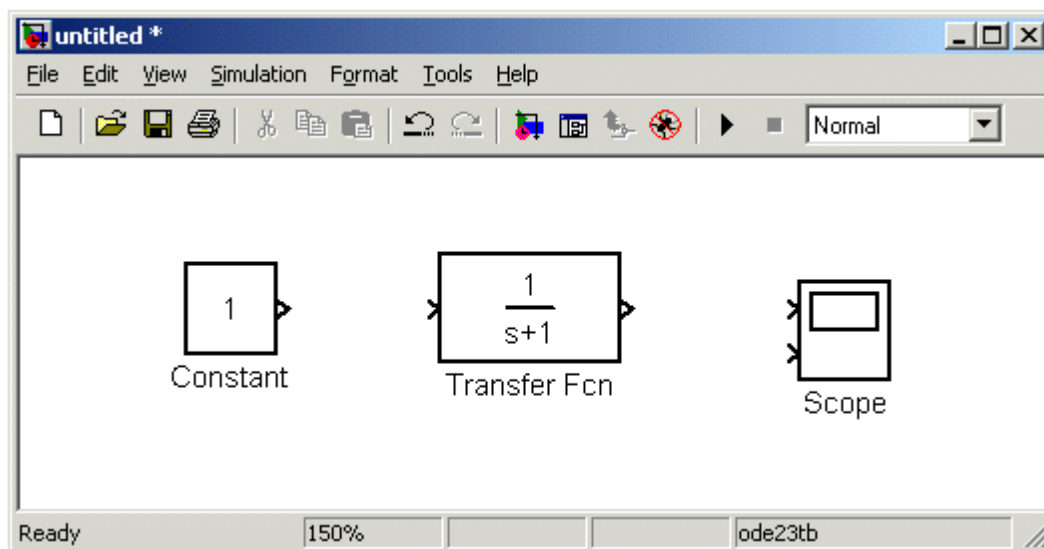


Рис 6. Окно модели, содержащее блоки



Для удаления блока необходимо выбрать блок (указать курсором на его изображение и нажать левую клавишу “мыши”), а затем нажать клавишу **Delete** на клавиатуре.

Для изменения размеров блока требуется выбрать блок, установить курсор в один из углов блока и, нажав левую клавишу “мыши”, изменить размер блока (курсор при этом превратится в двухстороннюю стрелку).

3. Далее, если это требуется, нужно изменить параметры блока, установленные программой “по умолчанию”. Для этого необходимо дважды щелкнуть левой клавишей “мыши”, указав курсором на изображение блока. Откроется окно редактирования параметров данного блока. При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая. После внесения изменений нужно закрыть окно кнопкой **ОК**. На рис.6.7 в качестве примера показаны блок, моделирующий передаточную функцию и окно редактирования параметров данного блока.

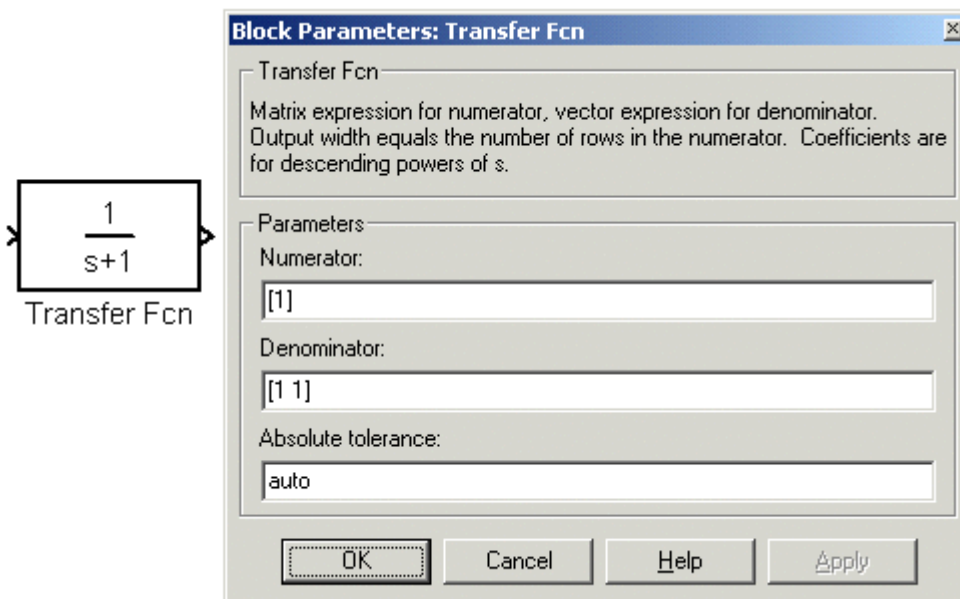


Рис 7. Блок, моделирующий передаточную функцию и окно редактирования параметров блока

4. 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. Для удаления линии требуется выбрать линию (так же, как это выполняется для блока), а затем нажать

клавишу **Delete** на клавиатуре. Схема модели, в которой выполнены соединения между блоками, показана на Рис. 6.8.

5. 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.

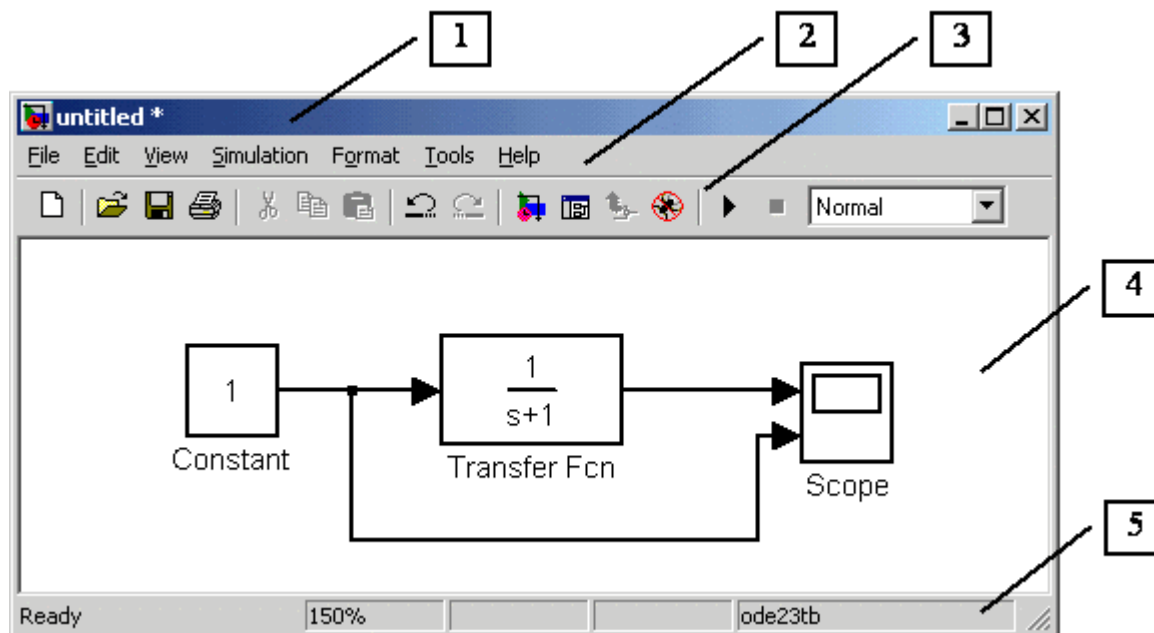


Рис 8. Схема модели

## **Библиотека блоков SIMULINK**

### **Sources - источники сигналов**

#### **Источник постоянного сигнала Constant**

*Назначение:*

Задаёт постоянный по уровню сигнал.

*Параметры:*

1. **Constant value** - Постоянная величина.
2. **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. 9 иллюстрирует применение этого источника и измерение его выходного сигнала с помощью цифрового индикатора **Display**.

#### **Источник синусоидального сигнала Sine Wave**

*Назначение:*

Формирует синусоидальный сигнал с заданной частотой, амплитудой, фазой и смещением.

Для формирования выходного сигнала блоком могут использоваться два алгоритма. Вид алгоритма определяется параметром **Sine Type** (способ формирования сигнала):

- **Time-based** – По текущему времени.
- **Sample-based** – По величине шага модельного времени.

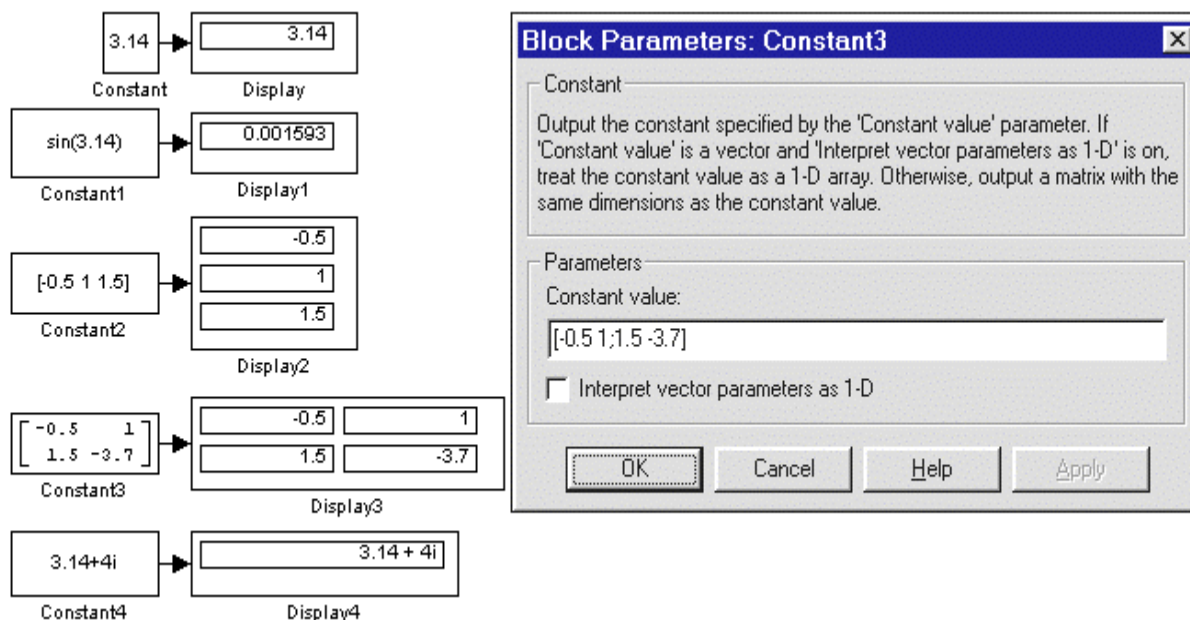


Рис. 9. Источник постоянного воздействия **Constant**

### Формирование выходного сигнала по текущему значению времени для непрерывных систем

Выходной сигнал источника в этом режиме соответствует выражению:

$$y = Amplitude * \sin(frequency * time + phase) + bias.$$

Параметры:

1. **Amplitude** - Амплитуда.
2. **Bias** – Постоянная составляющая сигнала.
3. **Frequency (rads/sec)** - Частота (рад/с).
4. **Phase (rads)** - Начальная фаза (рад).
5. **Sample time** – Шаг модельного времени. Используется для согласования работы источника и других компонентов модели во времени. Параметр может принимать следующие значения: **0** (по умолчанию) – Используется при моделировании непрерывных систем. **> 0** (положительное значение) – Задается при моделировании дискретных систем. В этом случае шаг модельного времени можно интерпретировать как шаг квантования по времени выходного сигнала. **-1** – Шаг модельного времени устанавливается таким же, как и в предшествующем блоке, т.е. блоке, откуда

приходит сигнал в данный блок. Этот параметр может задаваться для большинства блоков библиотеки **Simulink**. В дальнейшем он рассматриваться не будет.

При расчетах для очень больших значений времени точность расчета выходных значений сигнала падает вследствие значительной ошибки округления.

### Формирование выходного сигнала по текущему значению времени для дискретных систем

Алгоритм определения значения выходного сигнала источника для каждого последующего шага расчета определяется выражением (в матричной форме):

$$\begin{bmatrix} \sin(t + \Delta t) \\ \cos(t + \Delta t) \end{bmatrix} = \begin{bmatrix} \cos(\Delta t) & \sin(\Delta t) \\ -\sin(\Delta t) & \cos(\Delta t) \end{bmatrix} \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix},$$

где  $\Delta t$  – постоянная величина, равная значению **Sample time**.

В данном режиме ошибка округления для больших значений времени также уменьшает точность расчета.

### Формирование выходного сигнала по величине модельного времени и количеству расчетных шагов на один период

Выходной сигнал источника в этом режиме соответствует выражению:

$$y = \text{Amplitude} * \sin[(k + \text{Number of offset samples}) / \text{Samples per period}] + \text{bias},$$

где  $k$  – номер текущего шага расчета.

*Параметры:*

1. **Amplitude** - Амплитуда.
2. **Bias** – Постоянная составляющая сигнала.
3. **Samples per period** – Количество расчетных шагов на один период синусоидального сигнала:

$$\text{Samples per period} = 2\pi / (\text{frequency} * \text{Sample time})$$

4. **Number of offset samples** – Начальная фаза сигнала. Задается количеством шагов модельного времени:

$$\text{Number of offset samples} = \text{Phase} * \text{Samples per period} / (2\pi).$$

5. **Sample time** – Шаг модельного времени.

В данном режиме ошибка округления не накапливается, поскольку Simulink начинает отсчет номера текущего шага с нуля для каждого периода.

На Рис. 10 показано применение блока с разными значениями шага модельного времени (**Sample time** = 0 для блока **Sine Wave 1** и **Sample time** = 0.1 для блока **Sine Wave 2**). Для отображения графиков выходных сигналов в модели использован виртуальный осциллограф (**Scope**).

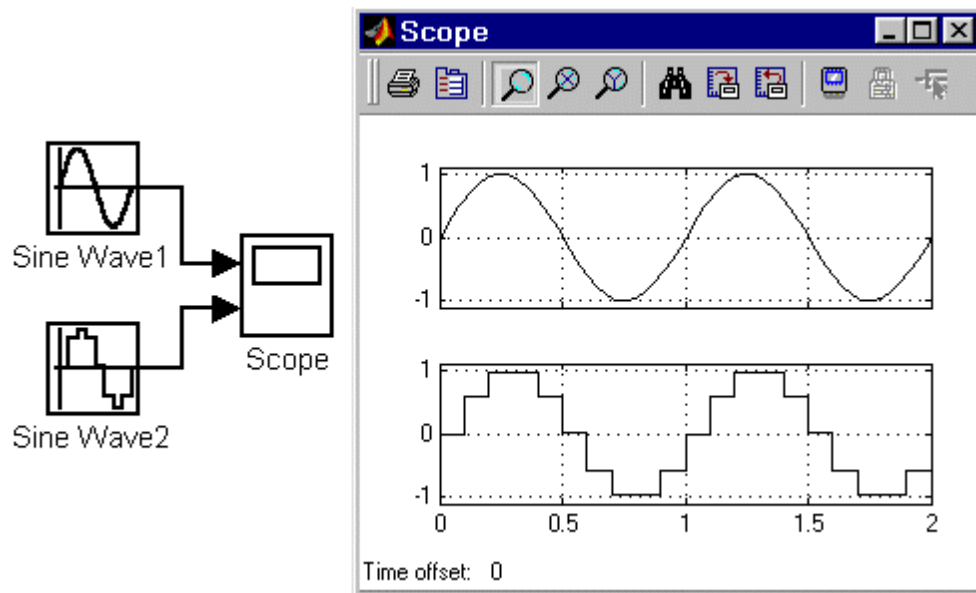


Рис. 10. Блок **Sine Wave**

### Источник линейно изменяющегося воздействия **Ramp**

*Назначение:*

Формирует линейный сигнал вида  $y = \textit{Slope} * \textit{time} + \textit{Initial value}$ .

*Параметры:*

1. **Slope** — Скорость изменения выходного сигнала.
2. **Start time** — Время начала формирования сигнала.
3. **Initial value** — Начальный уровень сигнала на выходе блока.

На Рис. 6.11 показано использование данного блока.

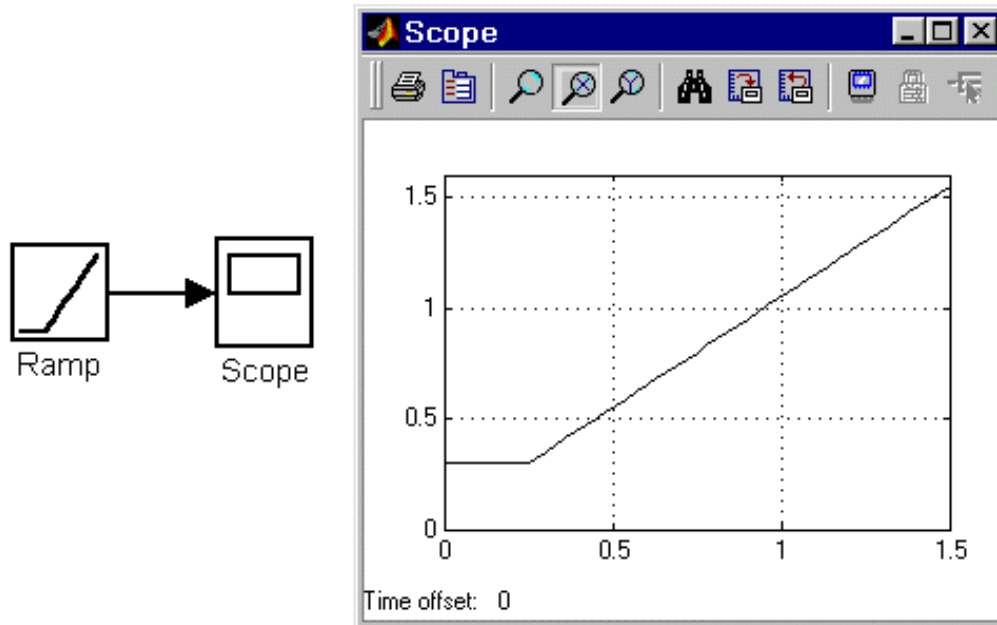
### Генератор ступенчатого сигнала **Step**

*Назначение:*

Формирует ступенчатый сигнал.

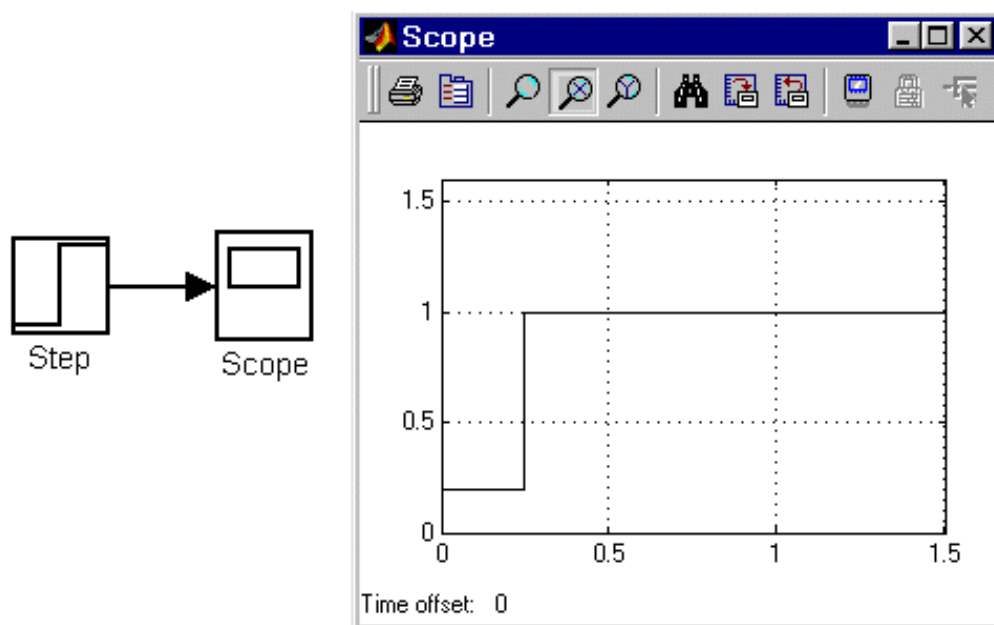
*Параметры:*

1. **Step time** - Время наступления перепада сигнала (с).
2. **Initial value** - Начальное значение сигнала.
3. **Final value** - Конечное значение сигнала.

Рис. 11. Блок **Ramp**

Изменение 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. 12. показано использование генератора ступенчатого сигнала.

Рис. 12. Блок **Step**

## Генератор сигналов **Signal Generator**

*Назначение:*

Формирует один из четырех видов периодических сигналов:

1. **sine** — Синусоидальный сигнал.

2. **square** — Прямоугольный сигнал.
3. **sawtooth** — Пилообразный сигнал.
4. **random** — Случайный сигнал.

*Параметры:*

1. **Wave form** – Вид сигнала.
2. **Amplitude** – Амплитуда сигнала.
3. **Frequency** - Частота (рад/с).
4. **Units** – Единицы измерения частоты. Может принимать два значения:
  - **Hertz** - Гц.
  - **rad/sec** – рад/с.

На Рис. 13 показано применение этого источника при моделировании прямоугольного сигнала.

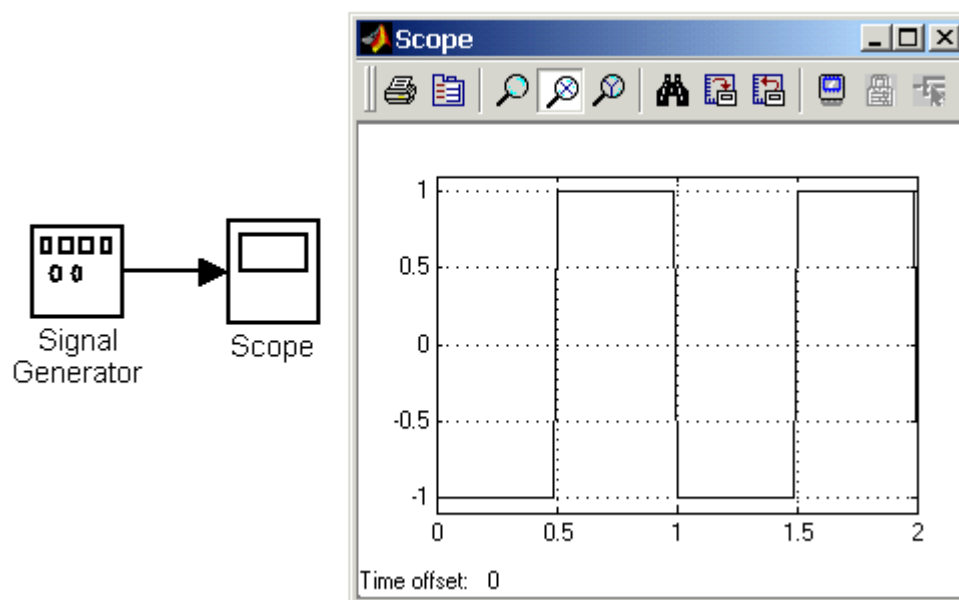


Рис. 13. Блок генератора сигналов

### Источник случайного сигнала с равномерным распределением

#### Uniform Random Number

*Назначение:*

*Формирование случайного сигнала с равномерным распределением.*

*Параметры :*

1. **Minimum** – Минимальный уровень сигнала.
2. **Maximum** – Максимальный уровень сигнала.
3. **Initial seed** – Начальное значение.

Пример использования блока и график его выходного сигнала представлен на Рис. 14.

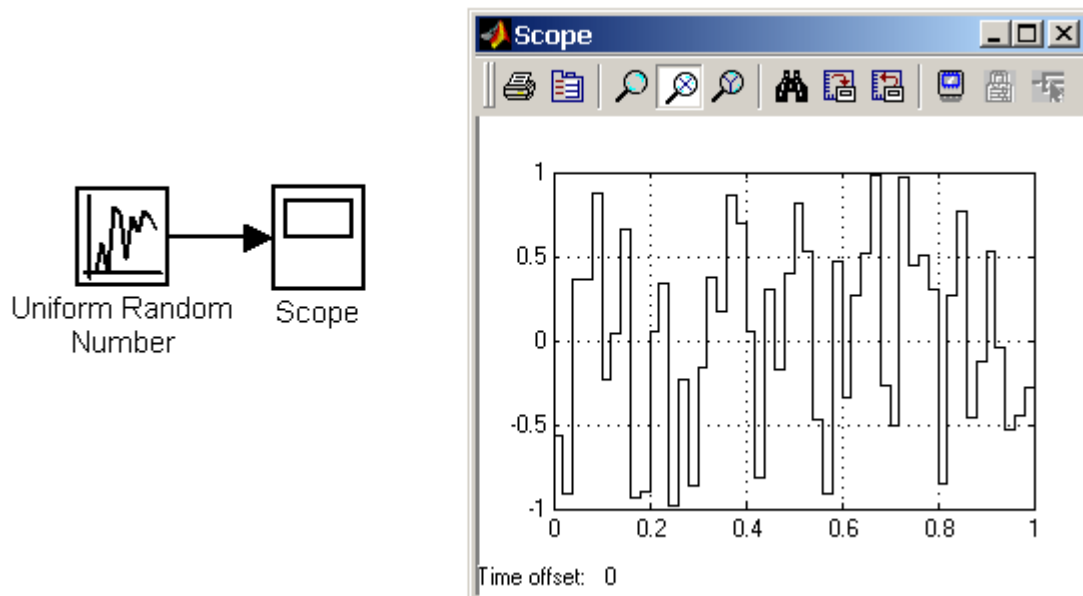


Рис. 14. Источник случайного сигнала с равномерным распределением

### Источник случайного сигнала с нормальным распределением

#### Random Number

*Назначение:*

*Формирование случайного сигнала с нормальным распределением уровня сигнала.*

*Параметры:*

1. **Mean** - Среднее значение сигнала
2. **Variance**- Дисперсия (среднеквадратическое отклонение).
3. **Initial seed** – Начальное значение.

### Источник импульсного сигнала Pulse Generator

*Назначение:*

*Формирование прямоугольных импульсов.*

*Параметры:*

1. **Pulse Type** – Способ формирования сигнала. Может принимать два значения:
  - **Time-based** – По текущему времени.
  - **Sample-based** – По величине модельного времени и количеству расчетных шагов.
2. **Amplitude** — Амплитуда.
3. **Period** — Период. Задается в секундах для **Time-based Pulse Type** или в шагах модельного времени для **Sample-based Pulse Type**.
4. **Pulse width** — Ширина импульсов. Задается в % по отношению к периоду для **Time-based Pulse Type** или в шагах модельного времени для **Sample-based Pulse Type**.
5. **Phase delay** — Фазовая задержка. Задается в секундах для **Time-based Pulse Type** или в шагах модельного времени для **Sample-based Pulse Type**.
6. **Sample time** — Шаг модельного времени. Задается для **Sample-based Pulse Type**.

Пример использования **Pulse Generator** показан на рис. 16.



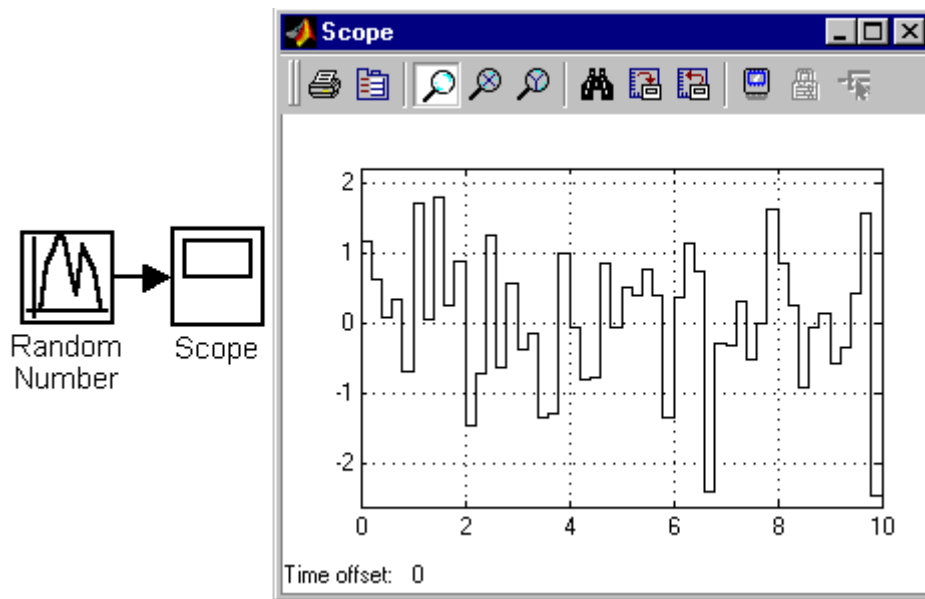


Рис. 15. Источник случайного сигнала с нормальным распределением

### Генератор линейно-изменяющейся частоты Chirp Generator

*Назначение:*

Формирование синусоидальных колебаний, частота которых линейно изменяется.

*Параметры:*

1. **Initial frequency** — Начальная частота (Гц);
2. **Target time** — Время изменения частоты (с);
3. **Frequency at target time** — Конечное значение частоты (Гц).

Пример использования блока показан на Рис. 17.

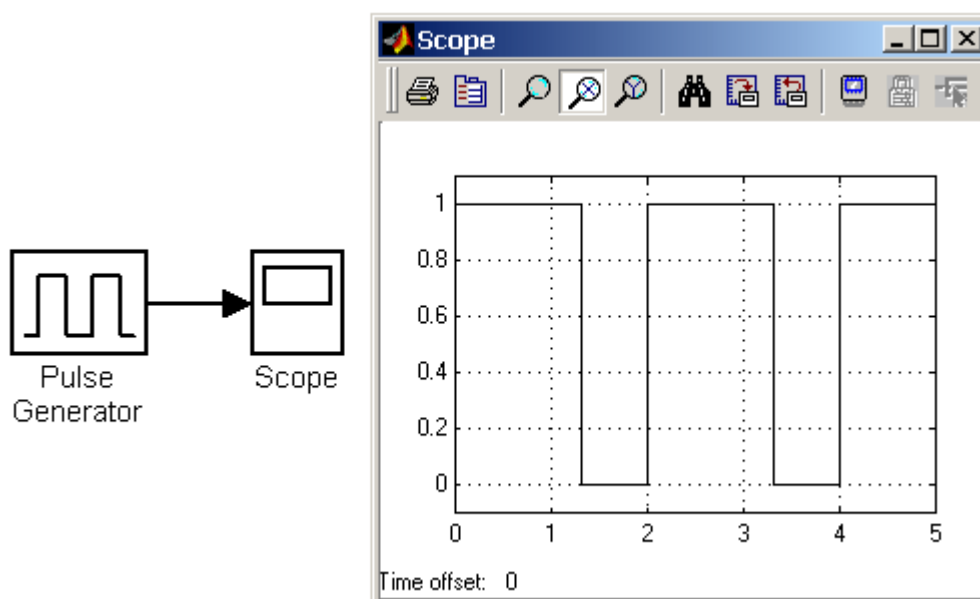


Рис.16. Источник прямоугольных импульсов

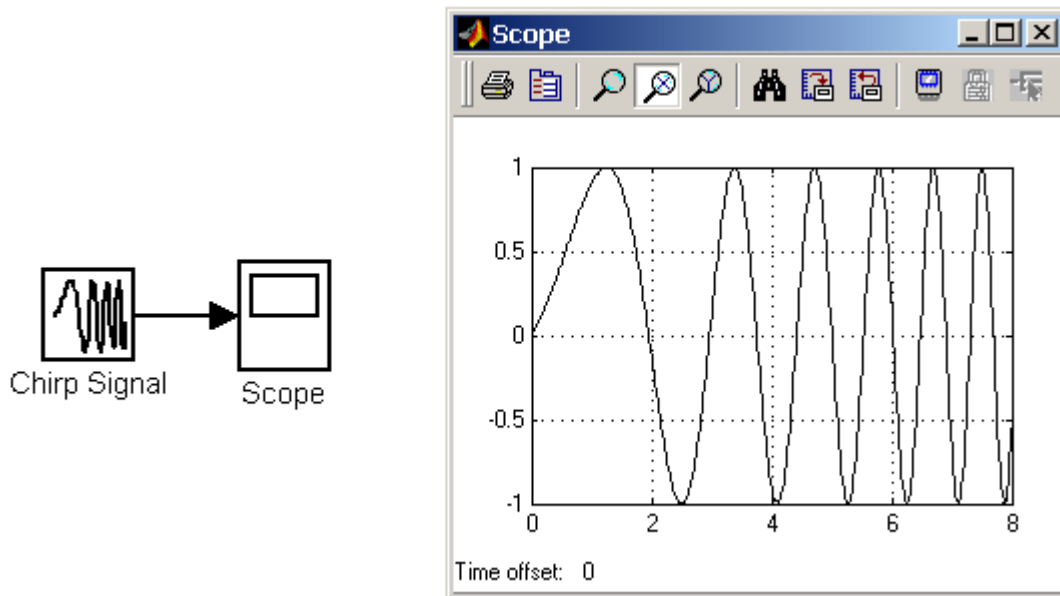


Рис.17. Генератор линейно-изменяющейся частоты

### Генератор белого шума Band-Limited White Noise

*Назначение:*

Создает сигнал заданной мощности, равномерно распределенной по частоте.

*Параметры:*

1. **Noise Power** – Мощность шума.
2. **Sample Time** – Модельное время.
3. **Seed** - Число, необходимое для инициализации генератора случайных чисел.

Рис. 18 показывает работу этого генератора.

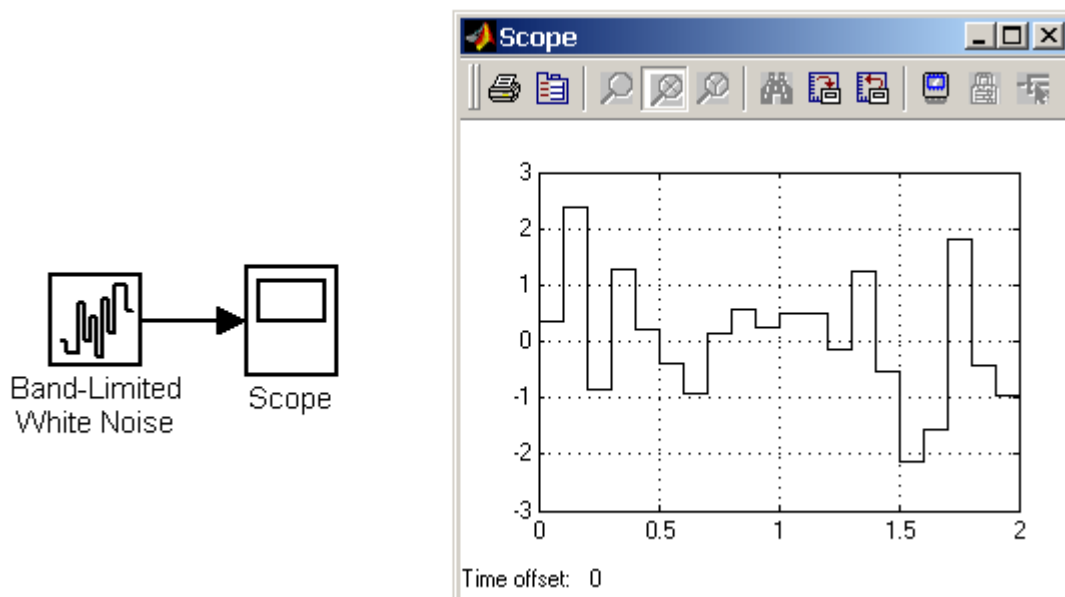


Рис.18. Генератор белого шума

### 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.
- Display time** - Отображение значения времени в блоке источника.

На рис. 19 показан пример работы данного источника.

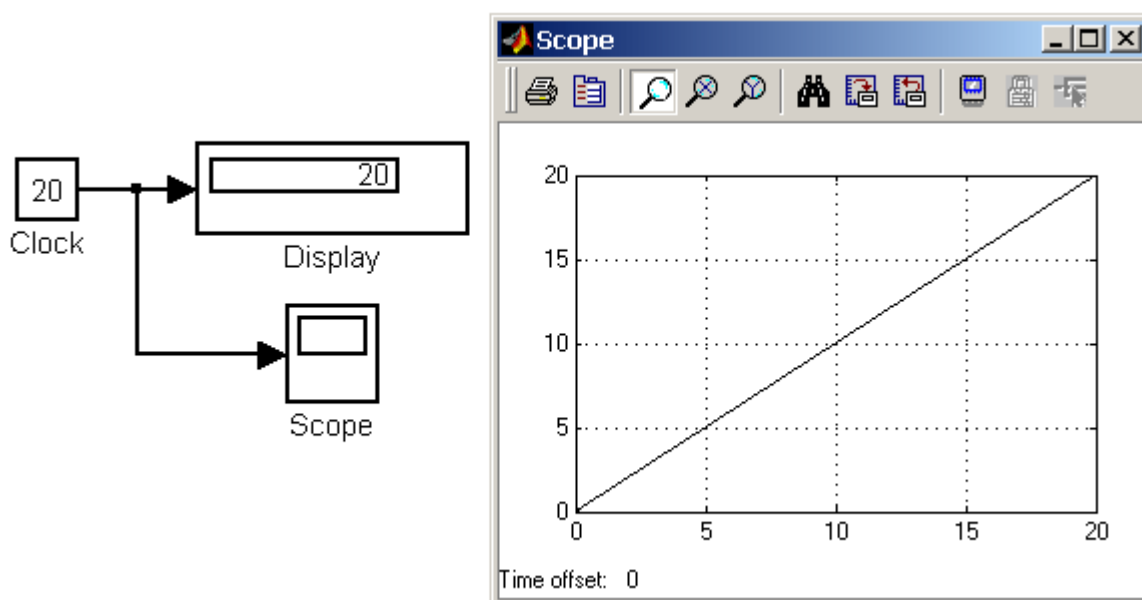


Рис.19. Источник временного сигнала

### Цифровой источник времени **Digital Clock**

*Назначение:*

Формирует дискретный временной сигнал.

*Параметр:*

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

На Рис 20 показана работа источника **Digital Clock**.

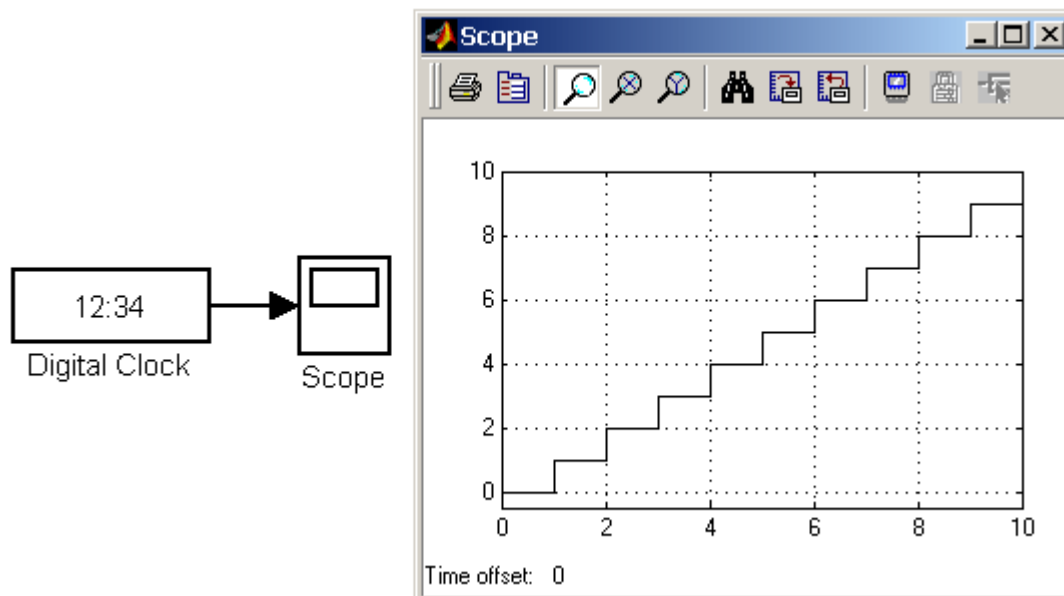


Рис.20. Цифровой источник временного сигнала

### Блок считывания данных из файла From File

*Назначение:*

Получение данных из внешнего файла.

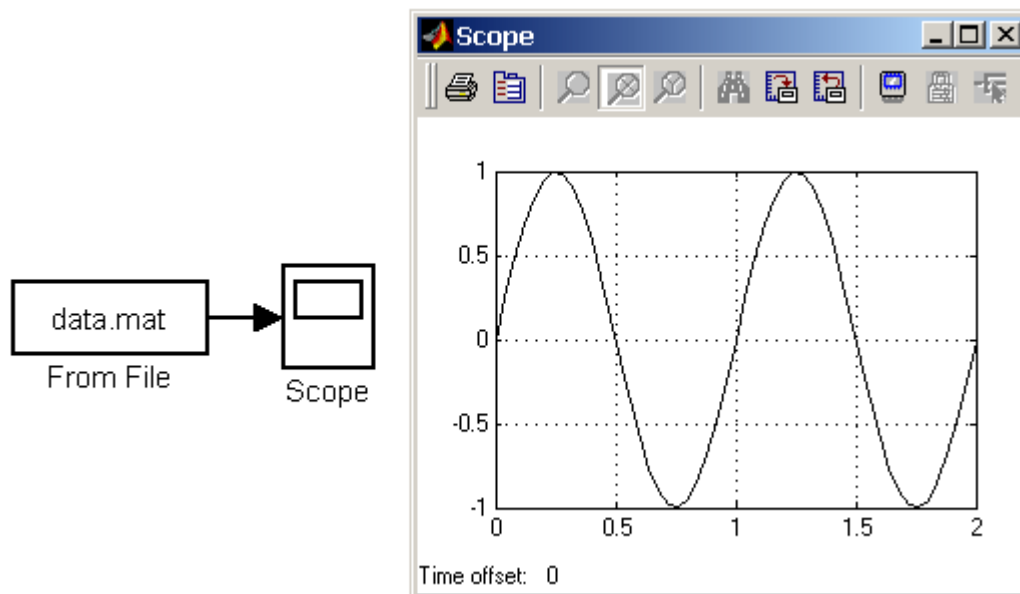
*Параметры:*

1. **File Name** - Имя файла с данными.
2. **Sample time** - Шаг изменения выходного сигнала блока.

Данные в файле должны быть представлены в виде матрицы.

Матрица должна состоять, как минимум, из двух строк. Значения времени записаны в первой строке матрицы, а в остальных строках находятся значения сигналов, соответствующие данным моментам времени. Значения времени должны быть записаны в возрастающем порядке. Выходной сигнал блока содержит только значения сигналов, а значения времени в нем отсутствуют. Если шаг расчета текущей модели не совпадает с отсчетами времени в файле данных, то Simulink выполняет линейную интерполяцию данных.

Файл данных (**mat**-файл), из которого считываются значения, не является текстовым. Структура файла подробно описана в справочной системе MATLAB. Пользователям Simulink удобнее всего создавать **mat**-файл с помощью блока **To File** (библиотека **Sinks**). На рис. 21 показан пример использования данного блока. Из файла **data.mat** считываются значения синусоидального сигнала.

Рис.21. Блок **From File**

### Блок считывания данных из рабочего пространства **From Workspace**

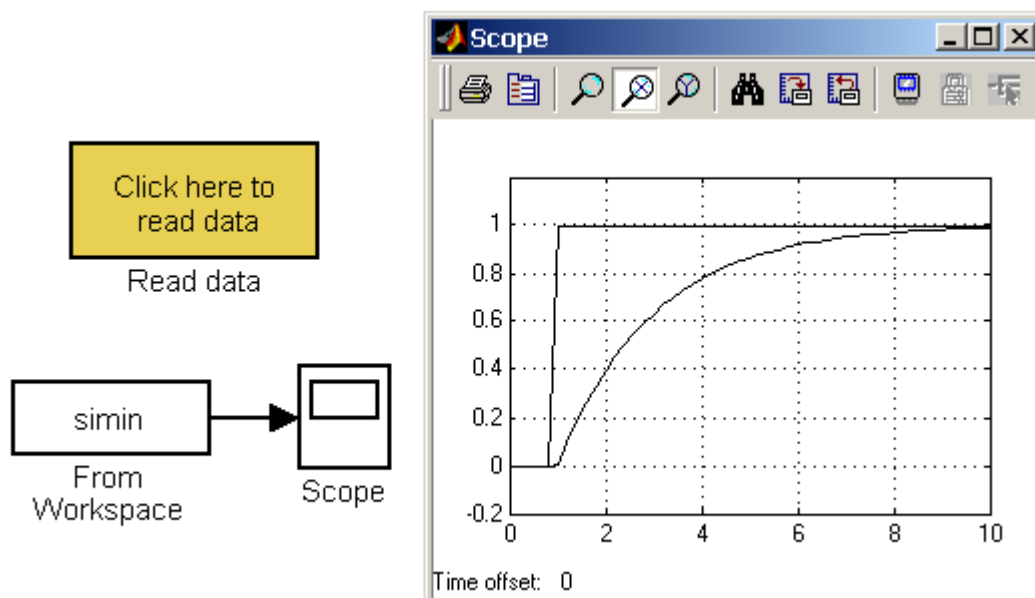
*Назначение:*

Получение данных из рабочего пространства MATLAB.

*Параметры:*

1. **Data** – Имя переменной (матрицы или структуры) содержащей данные.
2. **Sample time** - Шаг изменения выходного сигнала блока.
3. **Interpolate data** — Интерполяция данных для значений модельного времени не совпадающих со значениями в переменной **Data**.
4. **Form output after final data value by** – Вид выходного сигнала по окончании значений времени в переменной **Data**:
  - **Extrapolate** – Линейная экстраполяция сигналов.
  - **SettingToZero** – Нулевые значения сигналов.
  - **HoldingFinalValue** – Выходные значения сигналов равны последним значениям.
  - **CyclicRepetition** – Циклическое повторение значений сигналов. Данный вариант может использоваться, только если переменная **Data** имеет формат **Structure without time**.

На рис. 22 показан пример использования данного блока. Данные в переменную **simin** рабочей области MATLAB загружаются из файла с помощью блока **Read data**.

Рис. 22. Блок **From File**

### Блок сигнала нулевого уровня **Ground**

*Назначение:*

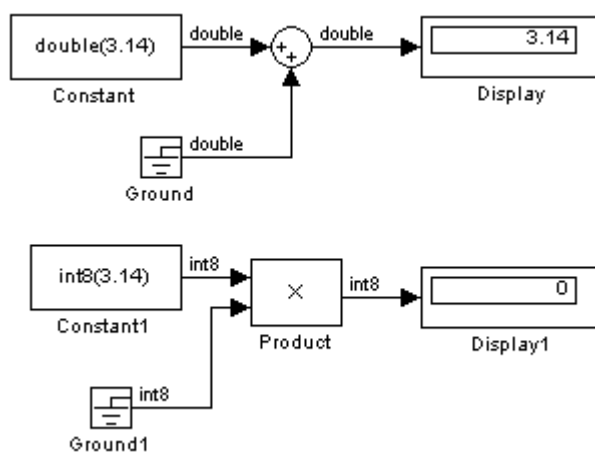
Формирование сигнала нулевого уровня.

*Параметры:*

Нет.

Если какой-либо вход блока в модели не подсоединен, то при выполнении моделирования в главном окне MATLAB появляется предупреждающее сообщение. Для устранения этого на неподключенный вход блока можно подать сигнал с блока **Ground**.

На рис. 23 даны примеры использования блока. В первом случае сигнал с блока **Ground** поступает на один из входов сумматора, а во втором на один из входов блока умножения. Показания блоков **Display** подтверждают, что вырабатываемый блоком **Ground** сигнал имеет нулевое значение. Из рисунка также видно, что тип выходного сигнала блока устанавливается автоматически, в соответствии с типами сигналов, подаваемых на другие входы блоков (в данном случае – на входы блоков **Sum** и **Product**).

Рис.23. Применение блока **Ground**

## Блок периодического сигнала **Repeating Sequence**

*Назначение:*

Формирование периодического сигнала.

*Параметры:*

1. **Time values** – Вектор значений модельного времени.
2. **Output values** – Вектор значений сигнала для моментов времени заданных вектором **Time values**.

Блок выполняет линейную интерполяцию выходного сигнала для моментов времени не совпадающих со значениями заданными вектором **Time values**. На рис. 24 показан пример использования блока для формирования пилообразного сигнала. Значения модельного времени заданы вектором **[0 3]**, а значения выходного сигнала вектором **[0 2]**.

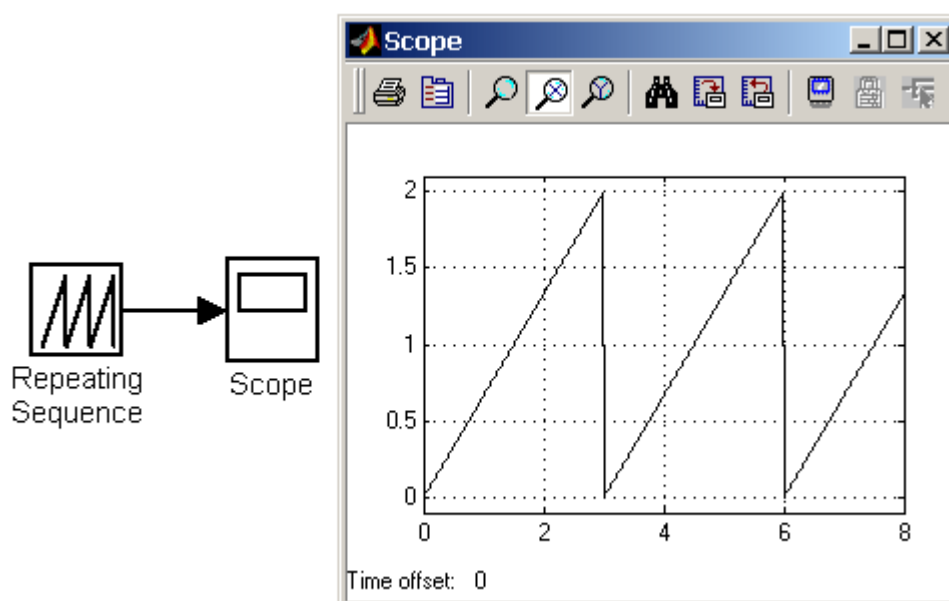


Рис. 24. Использование блока **Repeating Sequence**

## Блок входного порта **Inport**

*Назначение:*

Создает входной порт для подсистемы или модели верхнего уровня иерархии.

*Параметры:*

- **Port number** – Номер порта.
- **Port dimensions** – Размерность входного сигнала. Если этот параметр равен **-1**, то размерность входного сигнала будет определяться автоматически.
- **Sample time** – Шаг модельного времени.
- **Data type** – Тип данных входного сигнала: **auto**, **double**, **single**, **int8**, **uint8**, **int16**, **uint16**, **int32**, **uint32** или **boolean**.
- **Signal type** – Тип входного сигнала:
  1. **auto** – Автоматическое определение типа.
  2. **real** – Действительный сигнал.

### 3. complex- Комплексный сигнал.

• **Interpolate data** (флажок) – Интерполировать входной сигнал. В случае, если временные отсчеты входного сигнала считываемого из рабочей области **MATLAB** не совпадают с модельным временем, то блок будет выполнять интерполяцию входного сигнала. При использовании блока **Inport** в подсистеме данный параметр не доступен.

#### Использование блока **Inport** в подсистемах

Блоки **Inport** подсистемы являются ее входами. Сигнал, подаваемый на входной порт подсистемы через блок **Inport**, передается внутрь подсистемы. Название входного порта будет показано на изображении подсистемы как метка порта.

При создании подсистем и добавлении блока **Inport** в подсистему Simulink использует следующие правила:

1. При создании подсистемы с помощью команды **Edit/Create subsystem** входные порты создаются и нумеруются автоматически начиная с 1.
2. Если в подсистему добавляется новый блок **Inport**, то ему присваивается следующий по порядку номер.
3. Если какой либо блок **Inport** удаляется, то остальные порты переименовываются таким образом, чтобы последовательность номеров портов была непрерывной.
4. Если в последовательности номеров портов имеется разрыв, то при выполнении моделирования Simulink выдаст сообщение об ошибке и остановит расчет. В этом случае необходимо вручную переименовать порты таким образом, чтобы последовательность номеров портов не нарушалась.

На рис. 25 показана модель, использующая подсистему и схема этой подсистемы.

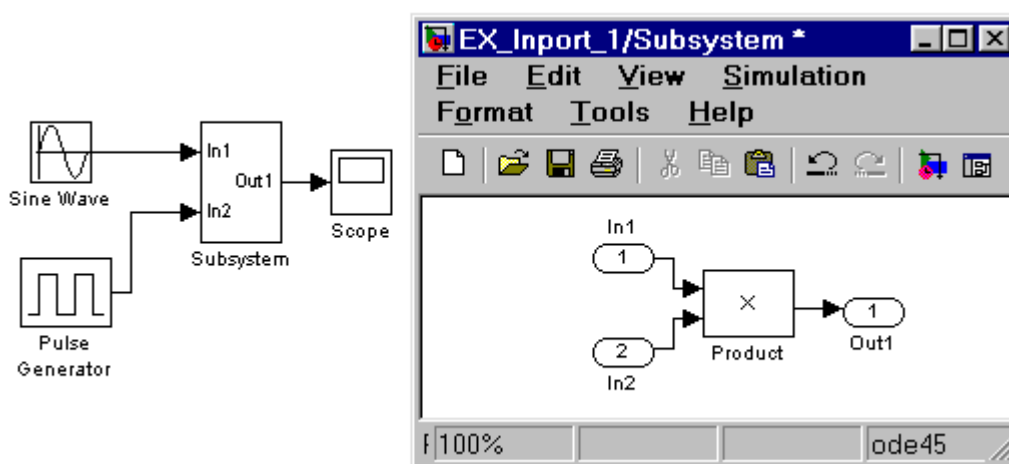


Рис. 25. Использование блока **Inport** в подсистеме

#### Использование блока **Inport** в модели верхнего уровня

Входной порт в системе верхнего уровня используется для передачи сигнала из рабочей области **MATLAB** в модель.

Для передачи сигнала из рабочего пространство **MATLAB** требуется не только установить в модели входной порт, но и выполнить установку параметров ввода на вкладке



**Workspace I/O** окна диалога **Simulation parameters...** (должен быть установлен флажок для параметра **Input** и задано имя переменной, которая содержит входные данные). Тип вводимых данных: **Array** (массив), **Structure** (структура) или **Structure with time** (структура с полем "время") задается на этой же вкладке.

На рис. 26 показана модель, считывающая входной сигнал из рабочего пространства MATLAB. Подсистема **Load Data** обеспечивает ввод данных из файла в рабочую область MATLAB.

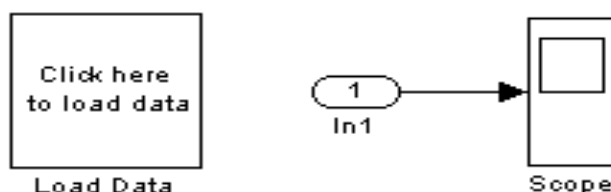


Рис. 26. Модель, считывающая входной сигнал из рабочего пространства MATLAB с помощью блока **Input**.

### **Отладчик SIMULINK моделей**

Отладчик Simulink является инструментом для поиска и диагностирования ошибок в моделях Simulink. Он дает возможность точно определить проблемы, выполняя моделирование постепенно с отображением значений входных и выходных сигналов любого из интересующих блоков модели. Simulink-отладчик имеет и графический, и интерфейс пользователя командной строки. Графический интерфейс позволяет наиболее удобно использовать основные возможности отладчика. Интерфейс командной строки дает способ обращаться ко всем возможностям отладчика. Пользователь, как правило, работает с графическим интерфейсом отладчика и обращается к интерфейсу командной строки по мере необходимости.

### **Графический интерфейс отладчика Simulink моделей**

Запуск графического интерфейса отладчика возможен одним из двух способов:

1. С помощью команды меню **Tools/Debugger** окна Simulink модели.
2. С помощью кнопки  панели инструментов окна Simulink модели.

После запуска отладчика на экране появится его окно (см. рис. 27).

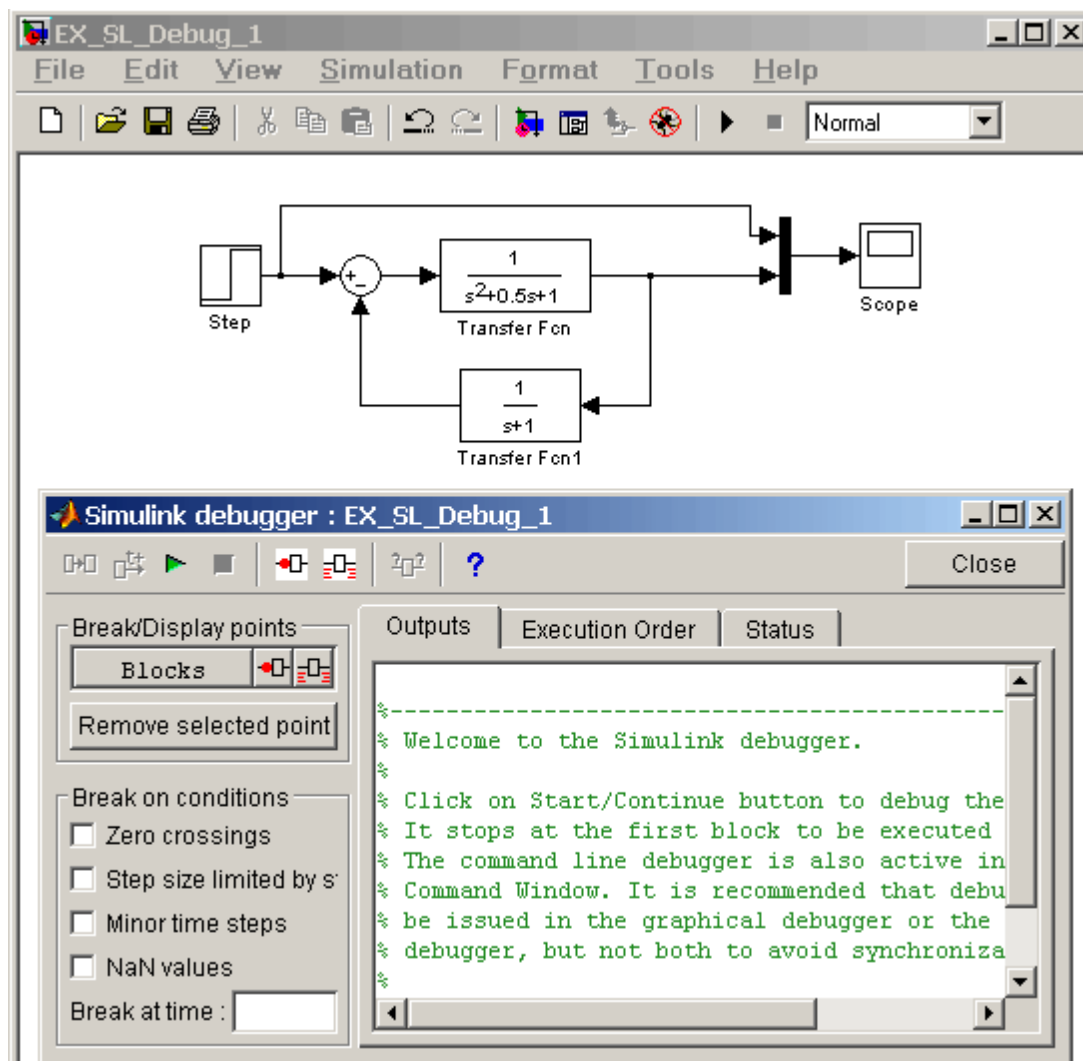


Рис. 27. Окно отладчика Simulink-модели.

Окно отладчика содержит следующие элементы:

- Панель инструментов
- Список контрольных точек **Break/Display points**
- Панель задания точек прерывания по условию **Break on conditions**
- Главное окно отладчика

### Панель инструментов

Общий вид панели инструментов показан на рис.28.

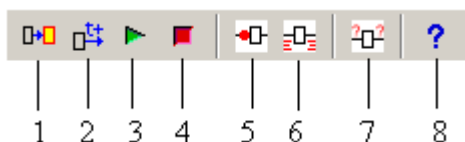


Рис. 28. Панель инструментов отладчика

Эта панель имеет следующие кнопки:

1. **Next Block** – Переход к следующему блоку. С помощью данной кнопки осуществляется режим отладки с остановкой моделирования после каждого выполненного блока.

2. **Next Time Step** – Переход к следующему временному шагу. С помощью данной кнопки выполняется режим отладки с остановкой моделирования после каждого выполненного временного шага.
3. **Start/Continue** – Начало/Продолжение отладки. Нажатие данной кнопки после запуска отладчика приводит к началу процесса моделирования и остановке его перед первым исполняемым блоком. Данная кнопка служит также для продолжения процесса отладки при установленных точках прерывания (остановки). Если точки прерывания установлены в модели, то нажатие данной кнопки позволяет продолжить моделирование и, затем, остановить его в заданной точке прерывания. Повторное нажатие кнопки возобновляет процесс моделирования и вызывает остановку на следующей точке прерывания. Если на текущем временном шаге все точки прерывания пройдены, то происходит переход на следующий временной шаг и остановка в первой точке прерывания на новом временном шаге.
4. **Stop** – Остановка отладки.
5. **Break before selected block** – Установка точки прерывания перед выделенным блоком. Точка прерывания не может быть установлена для виртуального блока, т.е. такого блока, чья функция является исключительно графической. Список неvirtуальных блоков можно посмотреть на вкладке **Execution Order** окна отладчика или вывести его командой **slist** при работе с отладчиком в режиме командной строки. При попытке установить контрольную точку для виртуального блока отладчик выведет на экран предупреждающее сообщение.
6. Для установки точки прерывания достаточно выделить блок в окне модели и, затем, нажать данную кнопку. Название блока немедленно появится в окне **Break/Display points** отладчика, как это показано на рис. 29.
7. **Display I/O of selected block with executed** – Отображать значения входных и выходных сигналов. Данная кнопка позволяет установить режим просмотра входных и выходных сигналов какого-либо неvirtуального блока. Режим может быть установлен как для блока, для которого установлена точка прерывания, так и для блока для которого контрольная точка не установлена. Значения входных и выходных сигналов отображаются на вкладке **Outputs** окна отладчика. Входные сигналы отображаются идентификаторами **U1, U2, U3** и т.д., а выходные – **Y1, Y2, Y3** и т.д. Дополнительно здесь же отображаются значения переменных состояния – **CSTATE** (см. рис. 29).

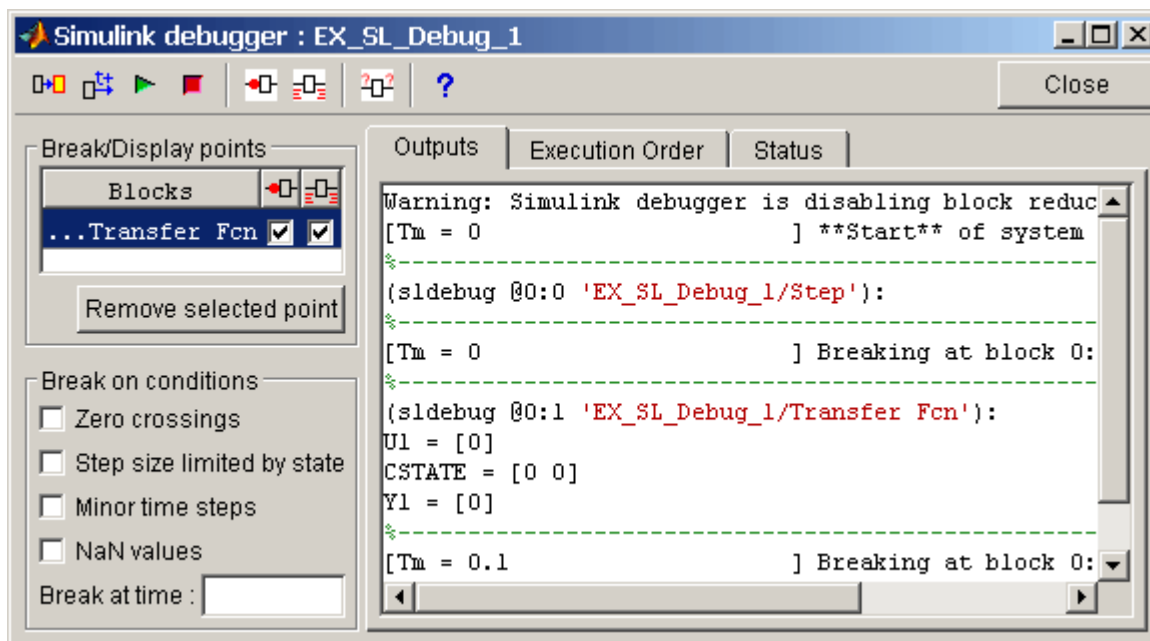





Рис. 29. Окно отладчика с установленной точкой прерывания для блока **Transfer Fcn**

**8. Display current I/O of selected block** – Показать значения входных и выходных сигналов для выделенного блока. Данной кнопкой удобно пользоваться, когда необходимо просмотреть значения сигналов какого-либо блока в момент остановки. Например, если для рассмотренной модели установлена точка прерывания перед выполнением блока **Transfer Fcn** и остановка моделирования в этой точке произошла, то просмотреть значения сигналов сумматора можно выделив этот блок в окне модели и нажав кнопку  отладчика.

**9. Help** – Вызов справки по отладчику.

### Список контрольных точек Break/Display points

Окно списка контрольных точек (рис. 30) содержит список блоков, для которых установлены контрольные точки (графа **Blocks**), а также свойства этих точек указанные с помощью флажков. Пользователь снимая или устанавливая флажки может изменять свойства контрольной точки, а именно: задавать/убирать точку прерывания на входе блока (графа ) или включать/выключать режим отображения значений сигналов блока (графа )

Удалить контрольную точку можно выделив ее в списке, и нажав кнопку **Remove selected point** (убрать выделенную точку).

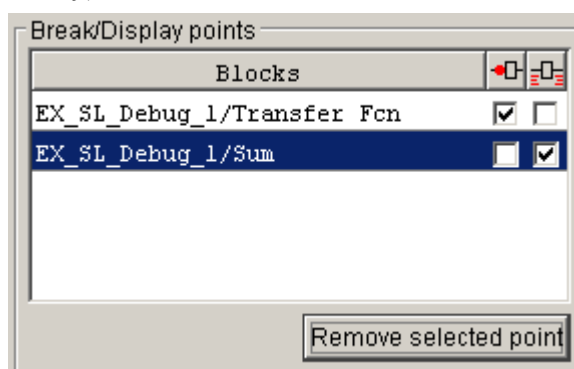


Рис. 30. Окно списка контрольных точек **Break/Display points**

### Панель задания точек прерывания по условию Break on conditions

Панель (рис. 31) содержит список условий, при наступлении которых расчет должен быть остановлен.

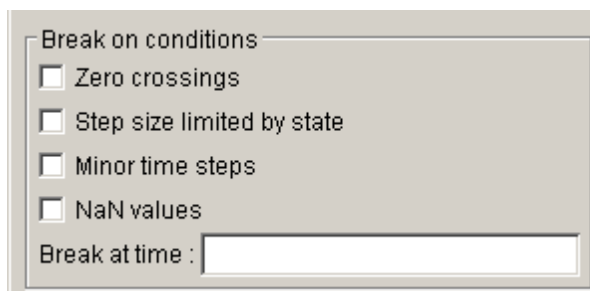


Рис. 31. Список условий прерывания расчетов

Список включает в себя:

1. **Zero crossings** – Переход сигнала через нулевой уровень при его скачкообразном изменении. На рис. рис. 32 показан пример модели, в которой имеет место такая ситуация. На рисунке хорошо видно, что в момент времени  $t = 5$  с происходит скачкообразное изменение сигнала с пересечением нулевого уровня.
2. **Step size limited by state** – Состояние ограничивающее шаг расчета. Опция заставляет отладчик останавливать моделирование, когда модель использует решатель с переменным шагом и решатель сталкивается с состоянием требующим ограничения размера шага расчета. Эта опция полезна при отладке моделей, требующих, как кажется, чрезмерно много расчетных шагов.

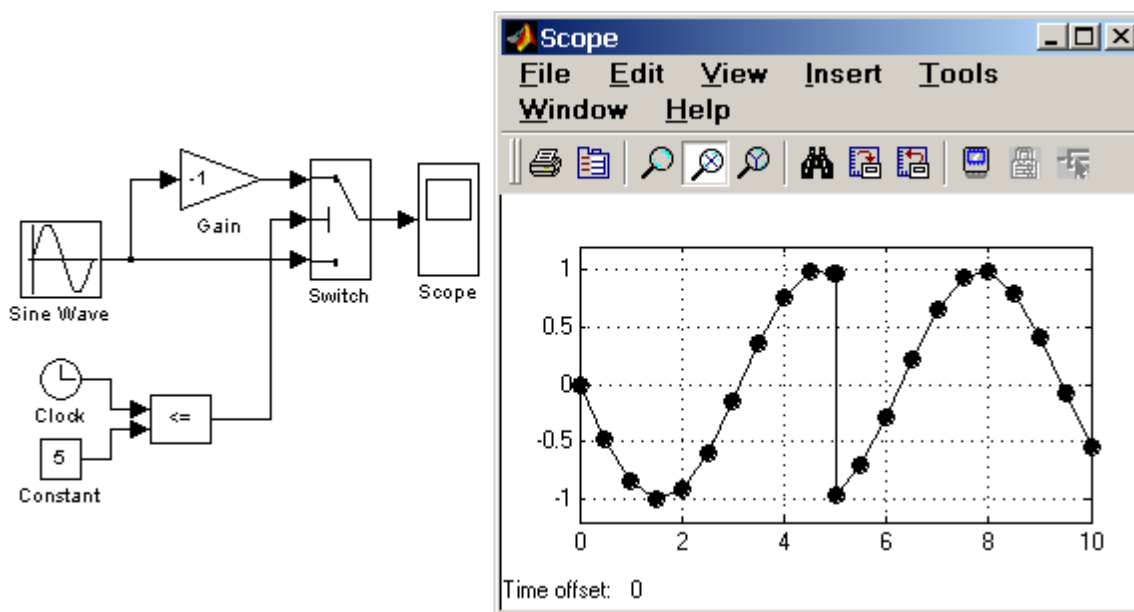


Рис. 32. Пример модели с изменением полярности сигнала

3. **Minor time steps** – режим отладки с использованием внутренних (малых) шагов. При выполнении расчетов **Simulink** может уменьшать заданный шаг расчета для

достижения нужной точности. Для того, чтобы увидеть и эти малые (внутренние) шаги необходимо установить опцию **Minor time steps**.

4. **NaN values** – Не числовое значение. Расчет будет прерван, когда вычисленное значение бесконечно или лежит вне диапазона значений, которые могут быть представлены компьютером, выполняющим моделирование (слишком малые или слишком большие значения). Эта опция полезна для точного определения вычислительных погрешностей в модели **Simulink**.
5. **Break at time** – Остановка в заданный момент времени. Параметр позволяет задать время до которого модель рассчитывается в обычном режиме. По достижении заданного времени расчет будет остановлен. Далее расчет необходимо возобновить с использованием возможностей отладчика. Данный режим удобен, если ошибка возникает не в начале интервала моделирования, а в какой-либо более поздний момент. В этом случае можно установить время остановки непосредственно перед появлением ошибки, а затем продолжить расчет в пошаговом режиме.

### Главное окно отладчика

Главное окно содержит три вкладки:

1. **Outputs** – Отображение результатов работы в режиме отладки. На данной вкладке (рис. 33) отображается текущее модельное время **Tm** (или **Ti** для внутренних шагов), индекс контролируемого блока в виде **@s:b**, где **s** – номер модели (подсистемы), **b** – номер блока, а также имя блока с указанием полного пути к блоку. На этой же вкладке выводятся значения входных (**U**) и выходных (**Y**) сигналов блока, если соответствующая опция установлена.

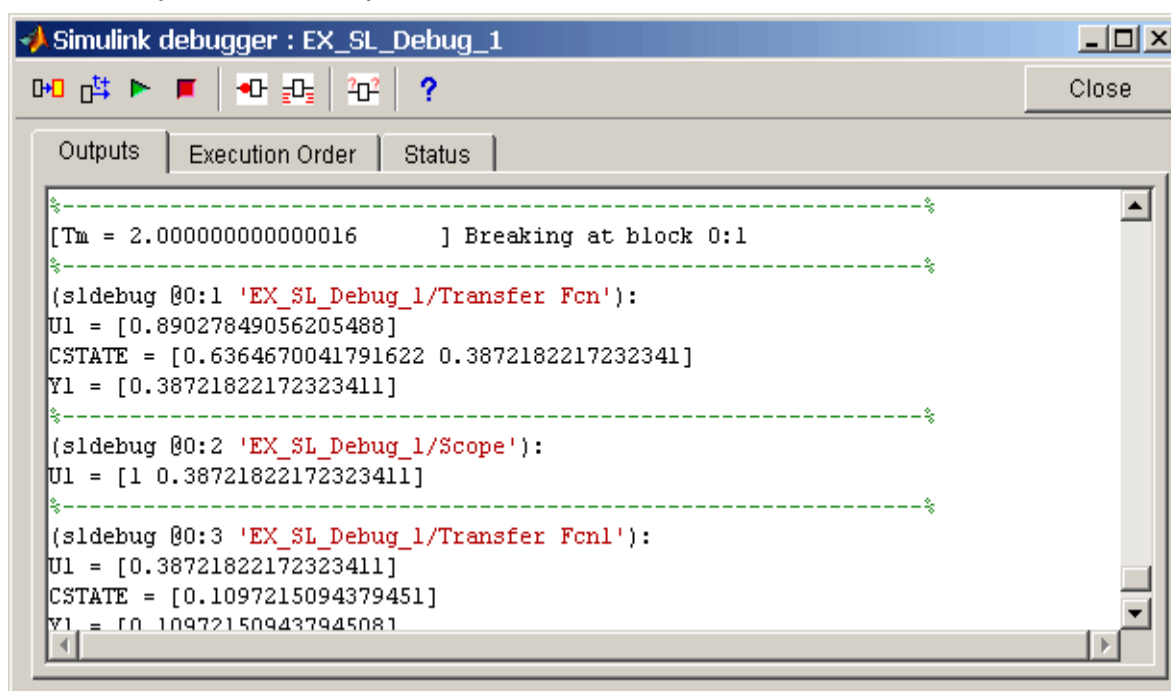


Рис. 33. Вкладка **Outputs** главного окна отладчика

2. **Execution Order** – Порядок выполнения. На вкладке отображается список неvirtуальных блоков в порядке их выполнения. Блоки, расположенные в начале

списка выполняются раньше, чем блоки, расположенные в конце списка. На рис. 34 показан пример данной вкладки для рассматриваемой выше модели.

- Status** – Статус отладчика. На данной вкладке (рис. 35) отображается информация о настройках и текущем состоянии отладчика: значение текущего временного шага, количество точек прерывания, информация об установке точек прерывания по условию и т.п.

Таким образом, запустив отладчик в графическом режиме, пользователь может провести пошаговую (по блокам или по временным шагам) отладку модели, установив при необходимости нужные контрольные точки.

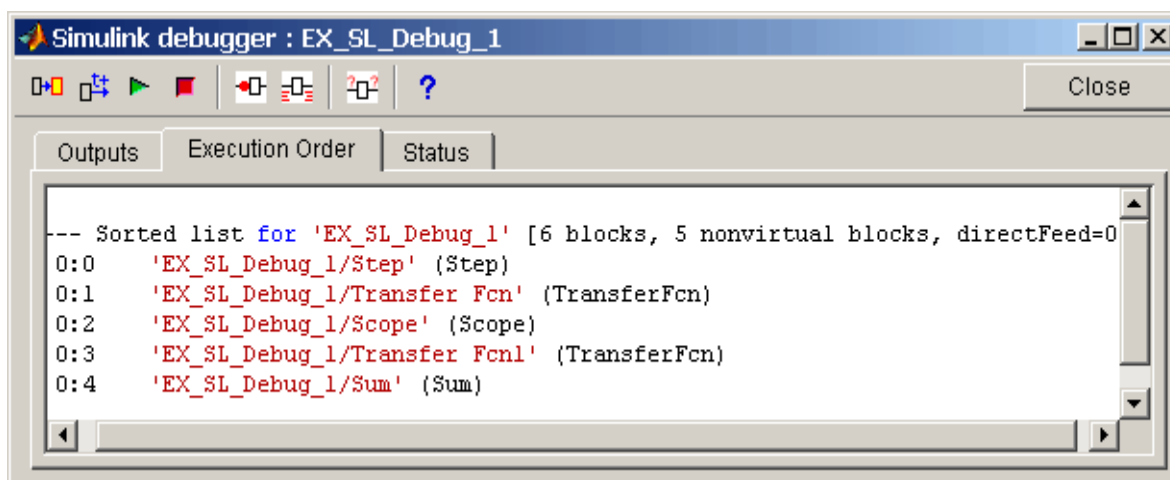


Рис. 34. Вкладка **Execution Order** главного окна отладчика

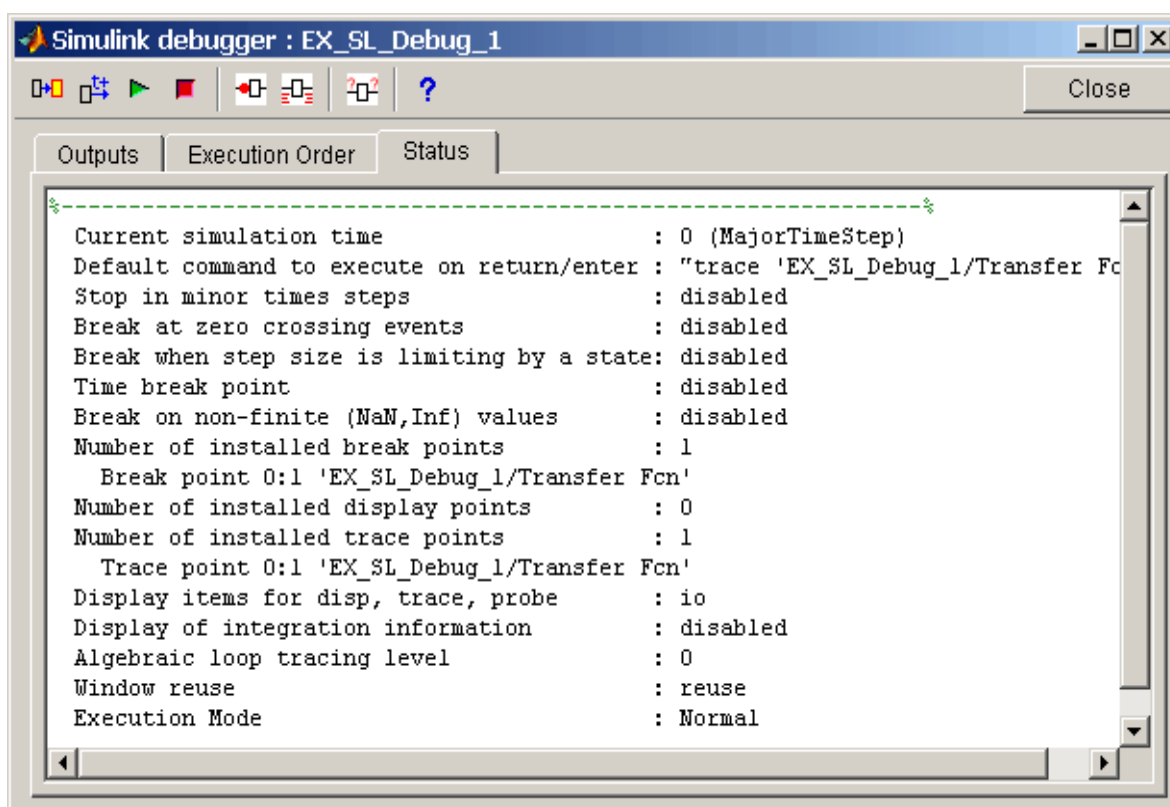


Рис. 35. Вкладка **Status** главного окна отладчика

Интерфейс командной строки дает пользователю доступ ко всем возможностям отладчика. Запуск отладчика в режиме командной строки возможен с помощью команды, задаваемой в рабочем окне MATLAB:

**sldebug('My\_model')** , где **My\_model** – имя отлаживаемой модели.

Для работы с отладчиком требуется вводить команды в главном окне MATLAB. Список команд приведен в Таблице 1.

Таблица 1

Команда	Краткая форма	Повтор	Назначение
<b>step</b>	<b>s</b>	<b>да</b>	Переход к следующему блоку
<b>next</b>	<b>n</b>	<b>да</b>	Переход к следующему временному шагу
<b>disp [s:b   gcb]</b>	<b>d</b>	<b>да</b>	Показ входных и выходных сигналов блока при остановке
<b>undisp &lt;s:b   gcb&gt;</b>	<b>und</b>	<b>да</b>	Удаление блока из списка отображаемых
<b>trace &lt;s:b   gcb&gt;</b>	<b>tr</b>	<b>да</b>	Показ входных и выходных сигналов блока во время выполнения
<b>untrace &lt;s:b   gcb&gt;</b>	<b>unt</b>	<b>да</b>	Удаление блока из списка трассировки
<b>probe [s:b   gcb]</b>	<b>p</b>	<b>нет</b>	Показ входных и выходных сигналов указанного блока
<b>break &lt;s:b   gcb&gt;</b>	<b>b</b>	<b>нет</b>	Вставка точки остановки при входе в блок
<b>bafter &lt;s:b   gcb&gt;</b>	<b>ba</b>	<b>нет</b>	Вставка точки остановки при выходе из блока
<b>bshow s:b</b>	<b>bs</b>	<b>нет</b>	Показ указанного с помощью индекса блока
<b>clear &lt;s:b   gcb&gt;</b>	<b>cl</b>	<b>нет</b>	Удаление точки останова
<b>zcbreak</b>	<b>zcb</b>	<b>нет</b>	Прерывание при обнаружении скачкообразного перехода сигналом нулевого уровня (непредусмотренное пересечение нуля)
<b>zclist</b>	<b>zcl</b>	<b>нет</b>	Список блоков дающих непредусмотренное пересечение



			нуля
<b>xbreak</b>	<b>x</b>	<b>нет</b>	Прерывание при переменном шаге расчета в состоянии требующем ограничения шага расчета
<b>tbreak [t]</b>	<b>tb</b>	<b>нет</b>	Установка/удаление остановки в указанный момент времени
<b>nanbreak</b>	<b>na</b>	<b>нет</b>	Установка/удаление остановки при обнаружении не числового ( <b>NaN, Inf</b> ) значения
<b>continue</b>	<b>c</b>	<b>да</b>	Продолжение моделирования
<b>run</b>	<b>r</b>	<b>нет</b>	Окончание режима отладки и продолжение расчета в обычном режиме
<b>stop</b>	<b>sto</b>	<b>нет</b>	Остановка моделирования
<b>quit</b>	<b>q</b>	<b>нет</b>	Прерывание моделирования
<b>status [all]</b>	<b>stat</b>	<b>нет</b>	Показ параметров отладчика
<b>states</b>	<b>state</b>	<b>нет</b>	Показ текущих значений переменных состояния
<b>systems</b>	<b>sys</b>	<b>нет</b>	Показ списка неvirtуальных подсистем
<b>slist</b>	<b>sli</b>	<b>нет</b>	Список неvirtуальных блоков
<b>minor</b>	<b>m</b>	<b>нет</b>	Режим отладки с использованием внутренних (малых) шагов
<b>ishow</b>	<b>i</b>	<b>нет</b>	Включение/выключение режима показа информации об интегрирующих блоках
<b>emode</b>	<b>e</b>	<b>нет</b>	Вывод информации о текущем режиме моделирования (обычный или ускоренный)
<b>probe level {all}   io</b>		<b>нет</b>	Установить уровень подробности показа сигналов блоков (все либо только входные и выходные)
<b>atrace level</b>	<b>at</b>	<b>нет</b>	Установка уровня отображения информации при трассировке алгебраических контуров ( <b>0</b> – ничего, <b>4</b> - все)

<b>ashow &lt;gcb   s:b&gt;</b>	<b>as</b>	<b>нет</b>	Показ алгебраического контура, содержащего указанный блок
<b>ashow s#n</b>	<b>as</b>	<b>нет</b>	Показ алгебраического контура с номером <b>n</b> в подсистеме (модели) <b>s</b>
<b>ashow clear</b>	<b>as</b>	<b>нет</b>	Отменить показ алгебраического контура

Часть команд приведенных в таблице требуют указания индекса блока. При использовании таких команд вместо имени блока можно указывать команду **gcb** (получить путь текущего блока), предварительно выделив нужный блок в окне модели.

Пример командного окна MATLAB в процессе отладки модели показан на рис. 36.

```

Command Window
U1 = [0.51686683568034941]
CSTATE = [0.1721147312588621]
Y1 = [0.17211473125886206]
%-----%
(sldebug @0:4 'EX_SL_Debug_1/Sum'): s
U1 = [1]
U2 = [0.17211473125886206]
Y1 = [0.82788526874113799]
[Tm = 2.3000000000000017 ] **Start** of system 'EX_SL_Debug_1' outputs
%-----%
(sldebug @0:0 'EX_SL_Debug_1/Step'): s
Zc =
NO]
Y1 = [1]
%-----%
(sldebug @0:1 'EX_SL_Debug_1/Transfer Fcn'): |

```

Рис. 36. Командное окно MATLAB в процессе отладки модели

### Задание

1. Используя справочную информацию о пакете, провести моделирование всех схем моделирования нелинейных элементов:

- Модель из трех нелинейных устройств (Saturation, Dead Zone, Quantizer, Relay) соединенных параллельно.
- Модель интегратора, подключенного к выходу источника прямоугольных импульсов (Discrete Pulse Generator)
- Модель дифференцирующего устройства
- Модель блока ограничения (Saturation)
- Модель блока с зоной нечувствительности (Dead Zone)
- Модель релейного устройства (Relay)
- Модель блока с ограничением скорости (Rate Limiter)
- Модель блока следящего квантования (Quantizer)
- Модель блока фрикционных эффектов (Coulomb&Viscous Friction)
- Модель люфта (Backlash)

- Описать сигналы на входе и выходе системы.
  - Описать закон функционирования системы
2. Исследовать работу всех схем для различных параметров (не менее трех различных вариаций параметров для каждой приведенной схемы).
- Управляемый переключатель (Switch)
  - Последовательное и параллельное соединение управляемых переключателей с выводом данных.
  - Многоходовый переключатель
  - Описать постановку задачи для решения ее переключателями и обосновать полученные результаты.
3. Провести моделирование системы с последовательно/параллельно соединенными нелинейными элементами.
4. Исследовать схему с двумя и более нелинейностями (добавить блок люфта).

#### **ЛИТЕРАТУРА:**

1. Данилов А. Компьютерный практикум по курсу "Теория управления". Simulink-моделирование в среде Matlab. МГУИЭ, 2002.
2. Дьяконов В.П. Simulink. Специальный справочник. – СПб: Питер, 2002. – 528с.
3. Дьяконов В.П. MATLAB 6/6.1/6.5 + Simulink 4/5. Основы применения. Полное руководство пользователя. СОЛОН-Пресс, 2004.
4. Дьяконов В.П., Круглов В. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. Питер, 2001.
5. Поршнев С. Компьютерное моделирование физических процессов в пакете MATLAB. Горячая Линия – Телеком, 2003.
6. Чен К., Джиблин П., Ирвинг А. MATLAB в математических исследованиях. М.: Мир, 2001.

## Лабораторная работа № 2

### Использование пакета Simulink для описания динамики колебательных систем с несколькими степенями свободы

13.00.02 – теория и методика обучения и воспитания (информатизация образования)

**A**

В статье описано использование программы моделирования динамических и событийно-управляемых систем Simulink для проектирования и анализа динамических систем, а также систем автоматического управления.

Для проектирования и анализа динамических систем, а также систем автоматического управления в настоящее время широко используются средства вычислительной техники. К последним, в частности, относится пакет моделирования динамических и событийно-управляемых систем Simulink [1], входящий в состав системы MATLAB. Пакет Simulink позволяет сокращать сроки проектирования, повышать качество разработки моделей физических систем и моделирования процессов, протекающих в данных системах.

В использовавшемся до недавнего времени традиционном подходе сначала создавалась математическая модель (как правило, в виде структурной схемы), далее создавалась программная реализация данной модели на каком-либо из универсальных языков программирования. При этом неизбежно возникало дублирование описания элементов системы и связей между ними. В пакете Simulink принципиально изменён характер требований, предъявляемых к математическому обеспечению: для управления всем ходом вычислительного процесса разработаны графические модули, которые используются для составления структурных схем исследуемых систем. Однако, как показывает анализ вузовского опыта преподавания различных естественнонаучных дисциплин, данное программное средство не получило пока широкого распространения. Целью статьи является демонстрация возможностей пакета Simulink на примере описания динамики колебательных систем с несколькими степенями свободы в пространстве состояний [2].

Напомним, что в пространстве состояния любая многомерная динамическая система описывается системой дифференциальных уравнений первого порядка в явной форме

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t), \quad (1)$$

где  $\mathbf{x}(t)$  –  $n$ -мерный вектор-столбец, компонентами которого являются переменные состояния,  $\mathbf{u}(t)$  –  $r$ -мерный вектор-столбец, координаты которого содержат значения выходных переменных,  $t$  – независимая переменная, время. Уравнение (1) часто называют уравнением состояния. Выходная переменная может быть представлена следующим образом

$$\mathbf{y} = g(\mathbf{x}, \mathbf{u}, t). \quad (2)$$

Уравнение (2) называется уравнением выходной переменной.

Для многомерных линейных систем уравнения (1), (2) соответственно принимают вид

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \quad (3)$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t), \quad (4)$$

где  $\mathbf{A}(t)$ ,  $\mathbf{B}(t)$ ,  $\mathbf{C}(t)$ ,  $\mathbf{D}(t)$  – матрицы размера  $(n \times n)$ ,  $(n \times r)$ ,  $(k \times n)$ ,  $(k \times k)$  соответственно.

Если матрицы  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  не зависят от времени, то такая система называется многомерной стационарной системой.

Для описания динамики системы в пространстве состояний необходимо также задать вектор начальных условий

$$\mathbf{x}(t_0) = \mathbf{x}_0. \quad (5)$$

Найдем значения матриц  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{D}$  для колебательной системы, имеющей одну степень свободы, уравнение движения которой имеет вид

$$\ddot{x} = -\omega^2 x + f(t), \quad (6)$$

где  $\omega^2$  – циклическая частота,  $x$  – смещение колеблющейся материальной точки от положения равновесия,  $f(t)$  – вынуждающая сила, отнесенная к массе колеблющегося тела.

В качестве компонентов вектора состояния выберем

$$\begin{aligned} x_1 &= x, \\ x_2 &= \dot{x}. \end{aligned} \quad (7)$$

Подставляя выражение (7) в (6), получаем уравнения состояния

$$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -\omega^2 x_1 + f(t). \end{cases} \quad (8)$$

Стандартная форма уравнений состояния (8) в векторно-матричных обозначениях имеет вид

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (9)$$

где  $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ ,  $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}$ ,  $\mathbf{u} = f(t)$ ,  $\mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ .

Если выходом рассматриваемой системы является смещение колебательной системы от положения равновесия, то  $\mathbf{y} = x_1$ , тогда  $\mathbf{C} = [1 \ 0]$  и  $\mathbf{D} = 0$ .

Если выходом рассматриваемой системы является её скорость, то  $\mathbf{y} = x_2$ ,  $\mathbf{C} = [0 \ 1]$ .

Если выходами рассматриваемой системы являются и смещение колебательной системы относительно положения равновесия, и её скорость, то  $\mathbf{y} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ ,  $\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ .

Таким образом, задача построения модели в пространстве состояний не имеет однозначного решения.

Рассмотрим более сложную линейную динамическую систему, представленную на рис. 1.

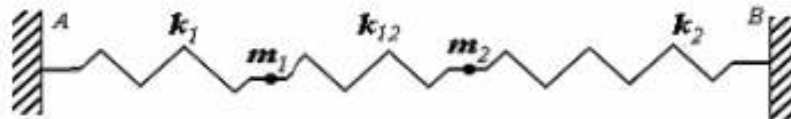


Рис. 1. Схема исследуемой колебательной системы

Система уравнений движения данной системы имеет вид

$$\begin{cases} \ddot{x}_1 = -\frac{k_1}{m_1} x_1 - \frac{k_{12}}{m_1} (x_1 - x_2) + \frac{F_1(t)}{m_1}, \\ \ddot{x}_2 = -\frac{k_2}{m_2} x_2 - \frac{k_{12}}{m_2} (x_1 - x_2) + \frac{F_2(t)}{m_2}. \end{cases} \quad (10)$$

В качестве компонентов вектора состояния выберем

$$w_1 = x_1, w_2 = \dot{x}_1, w_3 = x_2, w_4 = \dot{x}_2.$$

В выбранных переменных уравнения движения (10) принимают вид

$$\begin{cases} \dot{w}_1 = w_2, \\ \dot{w}_2 = -\frac{k_1}{m_1} w_1 - \frac{k_{12}}{m_1} (w_1 - w_3) + \frac{F_1(t)}{m_1}, \\ \dot{w}_3 = w_4, \\ \dot{w}_4 = -\frac{k_2}{m_2} w_3 - \frac{k_{12}}{m_2} (w_3 - w_1) + \frac{F_2(t)}{m_2}. \end{cases} \quad (11)$$

Из системы (11) находим

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_1 + k_{12}}{m_1} & 0 & \frac{k_{12}}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_{12}}{m_2} & 0 & -\frac{k_2 + k_{12}}{m_2} & 0 \end{pmatrix}, \quad \mathbf{u} = \begin{bmatrix} F_1(t) \\ F_2(t) \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{1}{m_1} & 0 \\ 0 & 0 \\ 0 & \frac{1}{m_2} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

После вычисления матриц **A**, **B**, **C**, **D**, необходимо запустить пакет Simulink, создать с помощью Simulink Library Browser, поддерживающего технологию визуального проектирования, блок-схему математической модели (рис. 2). Далее ввести в окно «Block Parameters: Sine Wave» параметры внешних сил, приложенных к колебательной системе (рис. 3). Далее в окне «Block Parameters: State-Space» задать матрицы **A**, **B**, **C**, **D**, а также начальные условия (рис. 4). После выполнения перечисленных выше действий оказывается достаточным запустить модель (меню Simulation => пункт Start) и открыть окно отображения зависимости внешних сил, действующих на колебательную систему от времени, а также выходных параметров, дважды щелкнув на блок-схеме по приборам Scope и Scope 1 (рис. 5, 6).

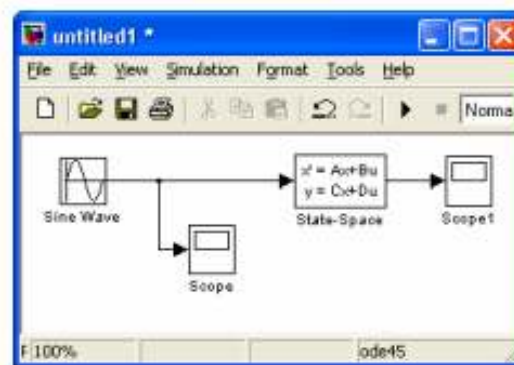


Рис. 2. Блок-схема математической модели колебательной системы в пространстве состояний



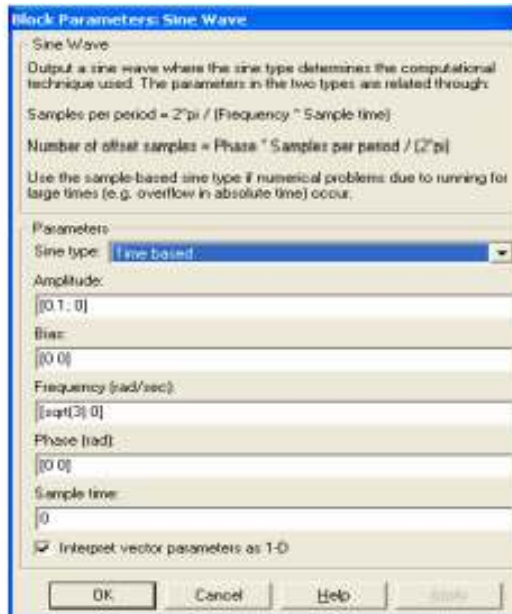


Рис. 3. Окно для ввода параметров внешней силы, приложенной к колебательной системе

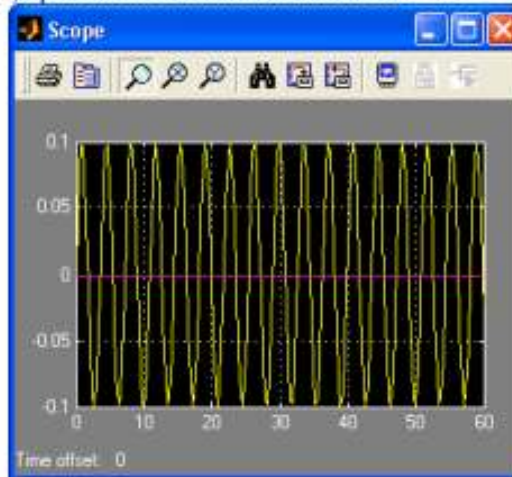


Рис. 5. Зависимость мгновенных значений внешних сил, приложенных к колебательной системе от времени



Рис. 4. Окно для задания матриц A, B, C, D и начальных условий

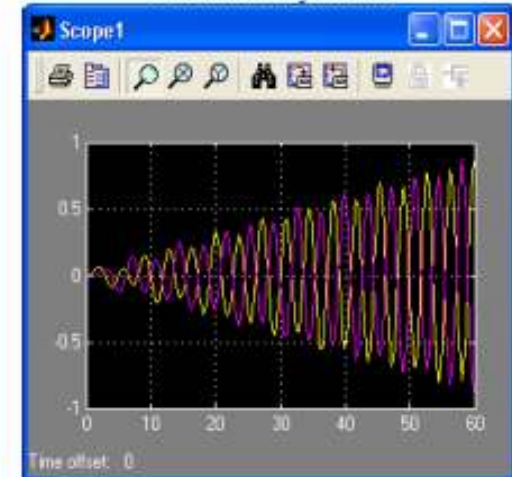


Рис. 6. Зависимость мгновенных смещений тел колебательной системы от времени

Реализацию традиционного подхода, основанного на реализации математической модели в виде программы, написанной на m-языке MATLAB, заинтересованный читатель может найти в [3, 4].

Отметим, два важных обстоятельства. Во-первых, в пакете Simulink имеется возможность считывать значения переменных, находящихся в рабочем пространстве MATLAB (Workspace). Для этого используется блок чтения/записи входных/выходных переменных, представленных на рис. 7.

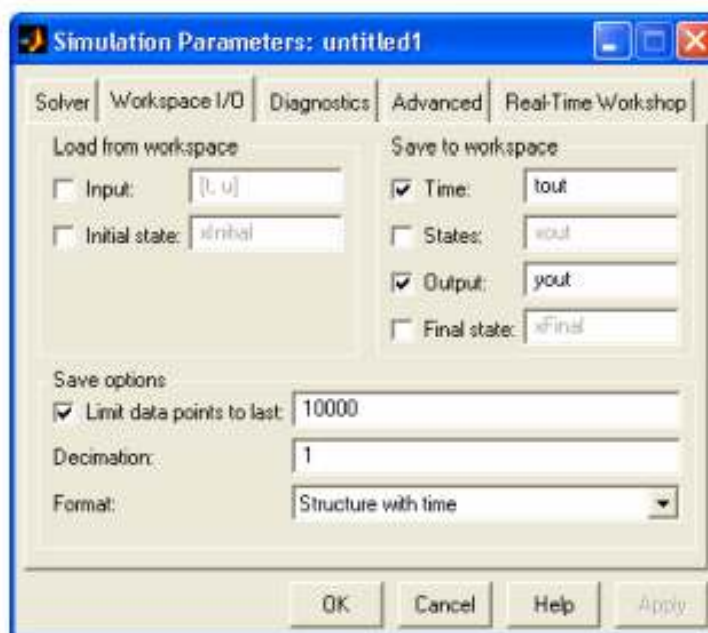


Рис. 7. Блок чтения/записи данных из Workspace

Во-вторых, процедура вычисления элементов матрицы  $A$  для произвольного числа тел колебательной системы достаточно легко алгоритмуется [3]. Сказанное в полной мере относится и к матрицам  $B$ ,  $C$ ,  $D$ . Это даёт возможность использовать описанную выше блок-схему математической модели колебательной системы для исследования динамики системы с произвольным числом тел. Действительно, достаточно создать соответствующие функции, размещающие матрицы  $A$ ,  $B$ ,  $C$ ,  $D$  в рабочем пространстве MATLAB, откуда они будут считываться перед началом выполнения расчётов.

Отмеченные свойства, в сочетании с другими не менее полезными свойствами, делают пакет Simulink гибким инструментом исследования динамических систем, а также систем управления, который целесообразно использовать при разработке методической поддержки дисциплин естественнонаучного блока, изучаемых в высшей школе. Знакомство с основами работы с данным пакетом представляется целесообразным как для студентов технических, так и педагогических университетов.

## Библиография

1. Дэбни Дж., Харман Т. Simulink 4. Секреты мастерства. М.: Бином; Лаборатория базовых знаний, 2003. 403 с.
2. Пантелеев А.В., Боргаковский А.С. Теория управления в примерах и задачах. М.: Высшая школа, 2003. 583 с.
3. Поршнев С.В. Компьютерное моделирование физических процессов в пакете MATLAB. М.: Горячая линия – Телеком, 2003. 592 с.
4. Поршнев С.В. Исследование в MATLAB особенностей вынужденных колебаний цепочки связанных гармонических осцилляторов // Труды Второй Всероссийской научной конференции «Проектирование инженерных и научных приложений в среде MATLAB». М.: ИПУ РАН, 2004. С. 665-681.



## Лабораторная работа № 3

# Simulink как средство исследования дифференциальных моделей

В математической теории управления распространены так называемые структурные схемы, которые являются средством моделирования систем. Аналогом этих схем являются Simulink-модели. В статье приводятся примеры использования Simulink при решении задач управления, включая нестационарные и нелинейные задачи. Предлагаемый подход упрощает и сокращает время решения инженерных задач в этой области. Статья предназначена для специалистов в области управления, находящихся в состоянии поиска инструментальных средств.

Достаточно часто задается вопрос о том, как построить фазовый портрет системы, как увидеть траекторию движения и т. п. Ответом на эти вопросы и служит эта заметка.

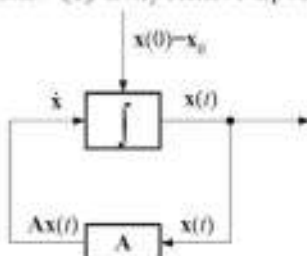
В основе построения дифференциальных моделей лежат так называемые структурные схемы, которые очень популярны в учебниках по теории автоматического управления. Рассмотрим «простые» модели теории управления — линейные стационарные динамические автономные системы. С математической точки зрения — это линейные дифференциальные однородные уравнения с постоянными коэффициентами, т. е. модели вида:

$$\begin{aligned} \frac{dx_1}{dt} &= a_{11}x_1(t) + a_{12}x_2(t) + \dots + a_{1n}x_n(t); \\ \frac{dx_2}{dt} &= a_{21}x_1(t) + a_{22}x_2(t) + \dots + a_{2n}x_n(t); \\ &\dots \\ \frac{dx_n}{dt} &= a_{n1}x_1(t) + a_{n2}x_2(t) + \dots + a_{nn}x_n(t), \end{aligned}$$

или, в матричном представлении:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t), \quad (1)$$

где  $\mathbf{x}$  — вектор состояний системы,  $\mathbf{A}$  — матрица размерности  $n \times n$ . Считается, что известны начальные значения параметров  $\mathbf{x}(0) = \mathbf{x}_0$ . В теории управления модель (1) задается в виде схемы, приведенной на рис. 1. На рисунке понятно, что вектор параметров  $\mathbf{A}$  умножается на матрицу  $\mathbf{A}$  — получается правая часть уравнения. Согласно (1) полученное произведение равно



▲ Рис. 1.

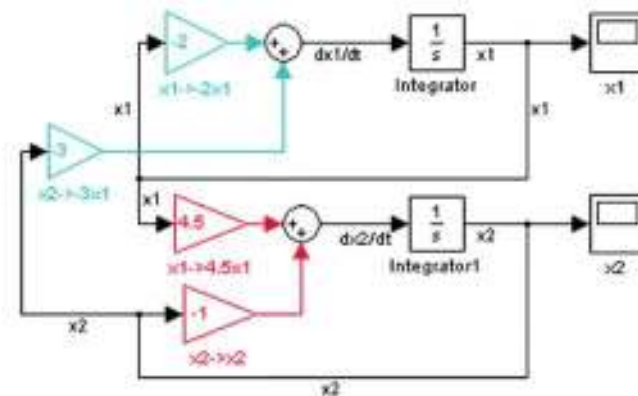
$dx/dt$ , и включив в схему интегратор, получим вектор  $\mathbf{x}$ , который и выдается на вход  $\mathbf{A}$ . Начальные условия необходимы для интегрирования.

Приведенная схема аналогично будет выглядеть и в виде Simulink-модели, которую можно исследовать, получая, например, динамические характеристики.

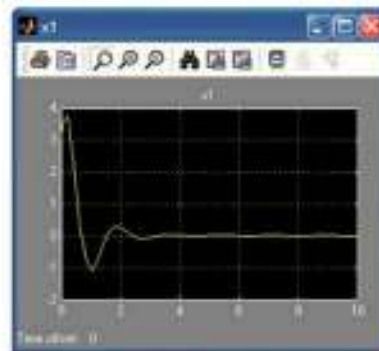
Пусть исследуемая система задается так:

$$\begin{aligned} \frac{dx_1}{dt} &= -2x_1(t) + 3x_2(t), \\ \frac{dx_2}{dt} &= 4.5x_1(t) + x_2(t), \\ x_1(0) &= 3, \quad x_2(0) = -6. \end{aligned} \quad (2)$$

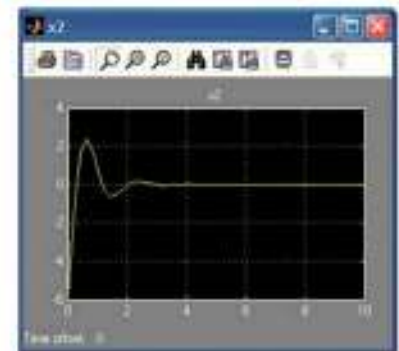
На рис. 2, а показана ее Simulink-модель. Цветом «cyan» выделено сформированное первое уравнение системы, цветом «red» — второе. Собственно, начальные условия можно установить, дважды щелкнув мышью на интеграторе и указав в поле начальных значений необходимое число. Рассмотрим, например, автономное движение системы, т. е. выведем графики  $x_1$  и  $x_2$  с помощью блока Scope. Результат показан на рис. 2, б и 2, в.



▲ Рис. 2, а.



▲ Рис. 2, б.



▲ Рис. 2, в.

Построим модель управляемой системы (рис. 3) вида:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{x}(0) &= \mathbf{x}_0. \end{aligned}$$

где  $\mathbf{u}(t)$  — вектор управлений,  $\mathbf{B}$  — матрица.

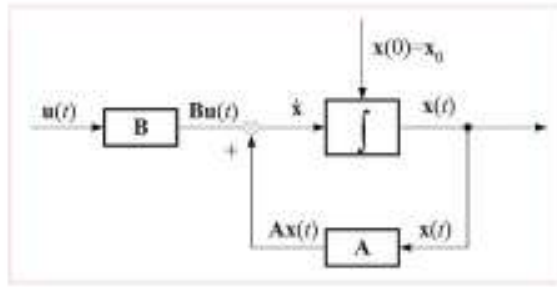


Рис. 3.

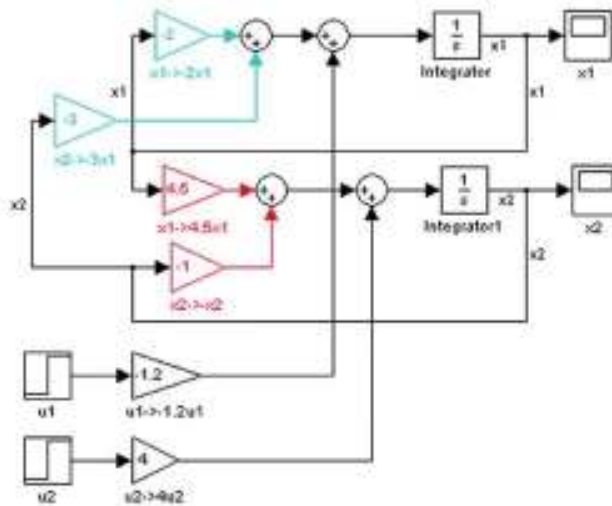


Рис. 4, а.



Рис. 4, б.

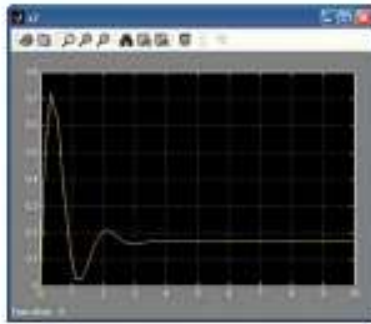


Рис. 4, в.

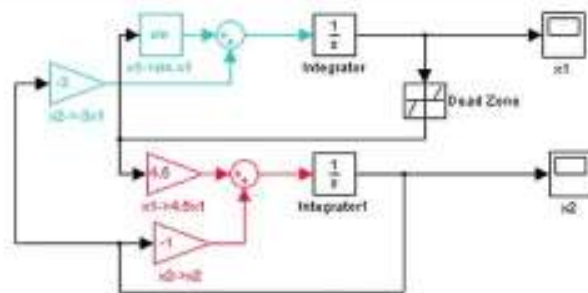


Рис. 5, а.

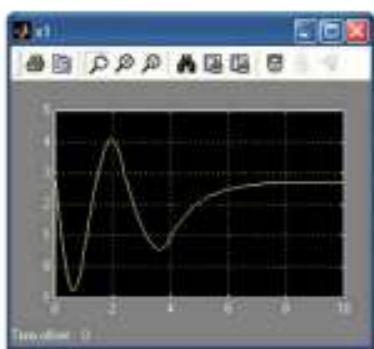


Рис. 5, б.

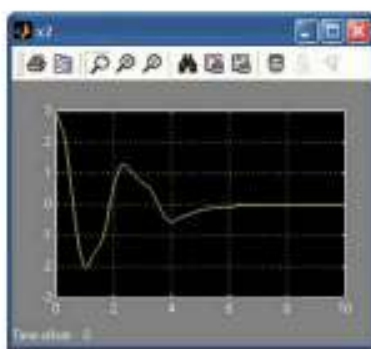


Рис. 5, в.

Добавим к предыдущей модели (2) входные воздействия в соответствии с системой

$$\frac{dx_1}{dt} = -2x_1(t) + 3x_2(t) - 1.2u_1,$$

$$\frac{dx_2}{dt} = 4.5x_1(t) + x_2(t) + 4u_2.$$

Полученная модель показана на рис. 4, а. Если в качестве входа использовать единичный ступок и взять нулевые условия, то на выходе будет переходная характеристика системы. Все это и было проделано. Полученный результат — рис. 4, б, в.

Конечно, для рассмотренных «классических» моделей необходимости в таком построении, собственно, нет. В Simulink существует специальный блок, где необходимо указать только матрицы **A** и **B**. Этот подход становится мощным инструментом при исследовании нестационарных и нелинейных моделей [1, 2].

Simulink предоставляет большой набор нелинейностей и специальных функций. На рис. 5, а показана модель нелинейной системы вида

$$\frac{dx_1}{dt} = -5\sin(\sigma(x_1(t))) - 3x_2(t),$$

$$\frac{dx_2}{dt} = 4.5\sigma(x_1(t)) - x_2(t)$$

с начальными условиями  $x_1(0) = 3, x_2(0) = 3$ . Здесь  $\sigma$  — нелинейность типа «мертвая зона». Автономное поведение параметров системы иллюстрируют рис. 5, б и в.

Если выводить выходные процессы в блок XY Graph, то в результате можно получить фазовую траекторию (рис. 6, а и б), меняя начальные условия — фазовый портрет системы.

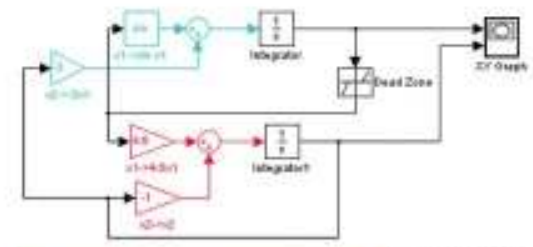
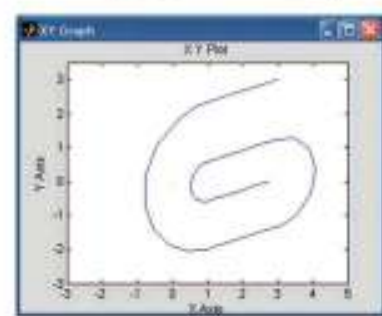


Рис. 6, а.  
Рис. 6, б.



Simulink-модель нестационарной управляемой системы

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2t & t \\ -9 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -t \\ 0 \end{bmatrix} u,$$

$$x_1(0) = 8, x_2(0) = 4$$

представляет собой схему, показанную на рис. 7, а.

Пусть, например, управляющее воздействие имеет вид, представленный блоком Signal Builder (рис. 7, в). Можно поэкспериментировать с вход-

ным сигналом и посмотреть на характеристика системы. Выходные сигналы системы управления приведены на рис. 7. б и 7. в.

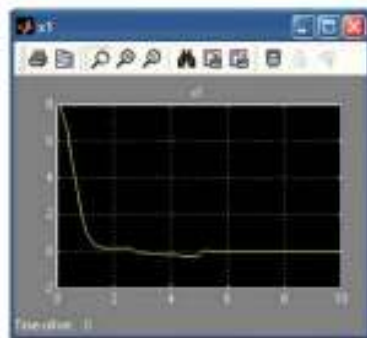
Если в качестве приемника использовать не блок Scope, а блок To Workspace, то данные будут передаваться в рабочую область MATLAB, что позволит использовать результаты (выходные процессы) в собственных функциях и приложениях. Напомним, что функция **sim** позволяет запускать на исполнение заранее созданную Simulink-модель. Например, для использования в MATLAB-приложении оптимального по быстродействию движения системы из последнего примера, необходимо выполнить следующие действия:

- 1) заменить блок Scope на To Workspace;
- 2) в поле Variable Name каждого блока To Workspace указать соответственно x1, x2;
- 3) записать модель в файл **opt1.mdl**;
- 4) использовать команду **sim('opt1')**.

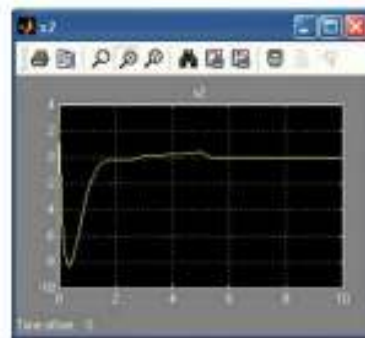
Переменные x1, x2 будут содержать результаты моделирования.

#### Литература

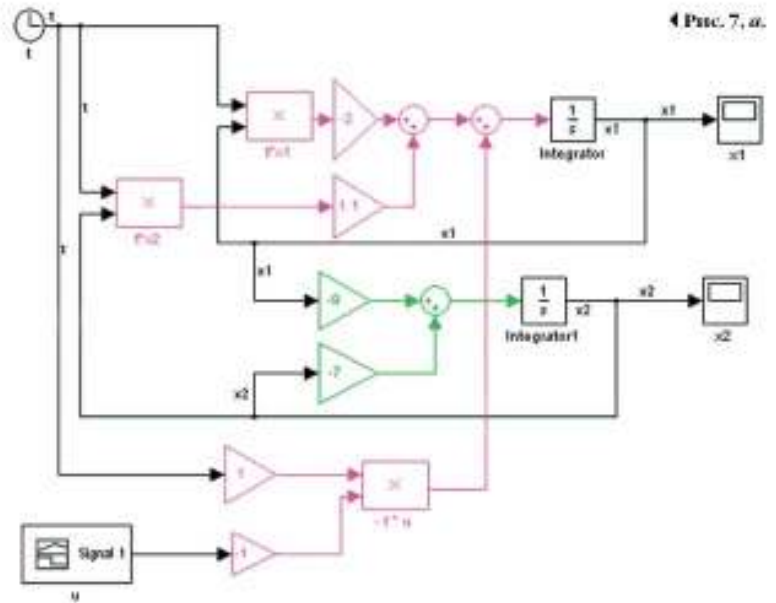
1. Гудышев А. Визуальное моделирование в среде MATLAB: учебный курс. – СПб: Питер, 2000. – 452 с.
2. Даврен Ю. Ф. MATLAB 5.x. – К.: ВЕТ, 2000. – 384 с.



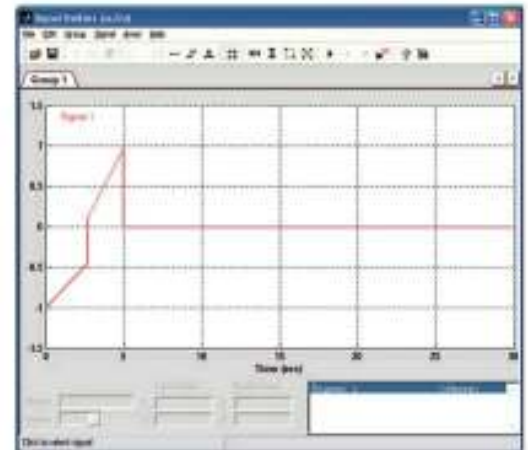
▲ Рис. 7, б.



▲ Рис. 7, а.



▲ Рис. 7, а.



▲ Рис. 7, в.



### Задание на контрольную работу

Изучить основные возможности ППП SIMULINK.

Принципы моделирования нелинейных, дифференциальных устройств, моделирование динамических систем.

#### Порядок выполнения работы

1. Ознакомиться с основными возможностями ППП SIMULINK.
2. Изучить принципы работы ППП SIMULINK.
3. Ответить на контрольные вопросы по вариантам.
4. Вариант по списку журнала старосты группы.
5. Выполнить три лабораторные работы (ответы на вопросы по варианту и три работы - это будет составлять одну контрольную работу).

№п/п	Контрольные вопросы
1	<ol style="list-style-type: none"> <li>1. Мета моделювання?</li> <li>2. Речовинні математичні?</li> <li>3. Математичне моделювання?</li> <li>4. Детерміновані моделі?</li> <li>5. Гомоморфні моделі?</li> <li>6. Схеми моделей об'єктів?</li> <li>7. Структурний підхід?</li> <li>8. Взаємодії в системі?</li> <li>9. Схема моделювання?</li> <li>10. Ізоморфні моделі?</li> </ol>
2	<ol style="list-style-type: none"> <li>1. Глибина моделювання та вимоги до моделей?</li> <li>2. Математичні моделі?</li> <li>3. Детерміновані моделі?</li> <li>4. Ідея комп'ютерного моделювання?</li> <li>5. Тактичне планування?</li> <li>6. Аспекти технології моделювання?</li> <li>7. Оцінки результатів?</li> <li>8. Точність моделей?</li> <li>9. Нестационарні моделі?</li> <li>10. Нелінійність математичних моделей?</li> </ol>
3	<ol style="list-style-type: none"> <li>1. Класифікація моделей?</li> <li>2. Ідеальні наочні моделі?</li> <li>3. Математична модель?</li> <li>4. Стахостичні моделі?</li> <li>5. Ступень ізоморфізму?</li> <li>6. Структура моделі?</li> <li>7. Функціональний підхід?</li> <li>8. Визначення моделі?</li> <li>9. Математичне моделювання?</li> <li>10. Ступень ізоморфізму?</li> </ol>
4	<ol style="list-style-type: none"> <li>1. Вимоги до моделей?</li> <li>2. Імітаційні моделі?</li> <li>3. Випадкові моделі?</li> <li>4. Організація процесу комп'ютерного моделювання?</li> <li>5. Етап експериментування?</li> <li>6. Пізнавальна модель?</li> </ol>

	<ul style="list-style-type: none"> <li>7. Модельно-змістовний компонент адекватності?</li> <li>8. Глибина моделювання та вимоги до моделей?</li> <li>9. Математичні моделі?</li> <li>10. Детерміновані моделі?</li> </ul>
<b>5</b>	<ul style="list-style-type: none"> <li>1. Речовинні натурні моделі?</li> <li>2. Ідеальні знакові моделі?</li> <li>3. Принципи побудови математичних моделей?</li> <li>4. Статичні моделі?</li> <li>5. Розробка математичних моделей?</li> <li>6. Елементи моделі?</li> <li>7. Властивості систем?</li> <li>8. Мета моделювання?</li> <li>9. Математична модель?</li> <li>10. Гомоморфні моделі?</li> </ul>
<b>6</b>	<ul style="list-style-type: none"> <li>1. Символічні моделі?</li> <li>2. Функціональні моделі?</li> <li>3. Стахостичні моделі?</li> <li>4. Визначення об'єкта?</li> <li>5. Інтерпретація?</li> <li>6. Прагматична модель?</li> <li>7. Характеристики адекватності?</li> <li>8. Вимоги до моделей?</li> <li>9. Імітаційні моделі?</li> <li>10. Випадкові моделі?</li> </ul>
<b>7</b>	<ul style="list-style-type: none"> <li>1. Речовинні фізичні моделі?</li> <li>2. Схема моделювання?</li> <li>3. Імітаційна модель?</li> <li>4. Ізоморфні моделі?</li> <li>5. Структуризація математичних моделей?</li> <li>6. Взаємозв'язок елементів в моделі?</li> <li>7. Події в системі?</li> <li>8. Класифікація моделей?</li> <li>9. Принципи побудови математичних моделей?</li> <li>10. Розробка математичних моделей?</li> </ul>
<b>8</b>	<ul style="list-style-type: none"> <li>1. Стаціонарні моделі?</li> <li>2. Лінійність математичних моделей?</li> <li>3. Комп'ютерна модель?</li> <li>4. Формалізація об'єкта?</li> <li>5. Методологія комп'ютерного моделювання?</li> <li>6. Інструментальна модель?</li> <li>7. Адекватність моделей?</li> <li>8. Символічні моделі?</li> <li>9. Функціональні моделі?</li> <li>10. Стахостичні моделі?</li> </ul>
<b>9</b>	<ul style="list-style-type: none"> <li>1. Речовинні математичні?</li> <li>2. Математичне моделювання?</li> <li>3. Детерміновані моделі?</li> <li>4. Гомоморфні моделі?</li> <li>5. Схеми моделей об'єктів?</li> <li>6. Структурний підхід?</li> <li>7. Взаємодії в системі?</li> <li>8. Визначення моделі?</li> </ul>

	<p>9. Аналітична модель?</p> <p>10. Особливості метода математичного моделювання?</p>
10	<p>1. Нестационарні моделі?</p> <p>2. Нелінійність математичних моделей?</p> <p>3. Комп'ютерне моделювання?</p> <p>4. Стратегічне планування?</p> <p>5. Тріада «модель – алгоритм – програма» ?</p> <p>6. Аналіз об'єктів моделювання?</p> <p>7. Погрішність моделей?</p> <p>8. Символічні моделі?</p> <p>9. Лінійність математичних моделей?</p> <p>10. Комп'ютерна модель?</p>
11	<p>11. Мета моделювання?</p> <p>12. Речовинні математичні?</p> <p>13. Математичне моделювання?</p> <p>14. Детерміновані моделі?</p> <p>15. Гомоморфні моделі?</p> <p>16. Схеми моделей об'єктів?</p> <p>17. Структурний підхід?</p> <p>18. Взаємодії в системі?</p> <p>19. Схема моделювання?</p> <p>20. Ізоморфні моделі?</p>
12	<p>11. Глибина моделювання та вимоги до моделей?</p> <p>12. Математичні моделі?</p> <p>13. Детерміновані моделі?</p> <p>14. Ідея комп'ютерного моделювання?</p> <p>15. Тактичне планування?</p> <p>16. Аспекти технології моделювання?</p> <p>17. Оцінки результатів?</p> <p>18. Точність моделей?</p> <p>19. Нестационарні моделі?</p> <p>20. Нелінійність математичних моделей?</p>
13	<p>11. Класифікація моделей?</p> <p>12. Ідеальні наочні моделі?</p> <p>13. Математична модель?</p> <p>14. Стахостичні моделі?</p> <p>15. Ступень ізоморфізму?</p> <p>16. Структура моделі?</p> <p>17. Функціональний підхід?</p> <p>18. Визначення моделі?</p> <p>19. Математичне моделювання?</p> <p>20. Ступень ізоморфізму?</p>
14	<p>11. Вимоги до моделей?</p> <p>12. Імітаційні моделі?</p> <p>13. Випадкові моделі?</p> <p>14. Організація процесу комп'ютерного моделювання?</p> <p>15. Етап експериментування?</p> <p>16. Пізнавальна модель?</p> <p>17. Модельно-змістовний компонент адекватності?</p> <p>18. Глибина моделювання та вимоги до моделей?</p> <p>19. Математичні моделі?</p> <p>20. Детерміновані моделі?</p>

15	<ul style="list-style-type: none"> <li>11. Речовинні натурні моделі?</li> <li>12. Ідеальні знакові моделі?</li> <li>13. Принципи побудови математичних моделей?</li> <li>14. Статичні моделі?</li> <li>15. Розробка математичних моделей?</li> <li>16. Елементи моделі?</li> <li>17. Властивості систем?</li> <li>18. Мета моделювання?</li> <li>19. Математична модель?</li> <li>20. Гомоморфні моделі?</li> </ul>
16	<ul style="list-style-type: none"> <li>11. Символічні моделі?</li> <li>12. Функціональні моделі?</li> <li>13. Стахостичні моделі?</li> <li>14. Визначення об'єкта?</li> <li>15. Інтерпретація?</li> <li>16. Прагматична модель?</li> <li>17. Характеристики адекватності?</li> <li>18. Вимоги до моделей?</li> <li>19. Імітаційні моделі?</li> <li>20. Випадкові моделі?</li> </ul>
17	<ul style="list-style-type: none"> <li>11. Речовинні фізичні моделі?</li> <li>12. Схема моделювання?</li> <li>13. Імітаційна модель?</li> <li>14. Ізоморфні моделі?</li> <li>15. Структуризація математичних моделей?</li> <li>16. Взаємозв'язок елементів в моделі?</li> <li>17. Події в системі?</li> <li>18. Класифікація моделей?</li> <li>19. Принципи побудови математичних моделей?</li> <li>20. Розробка математичних моделей?</li> </ul>
18	<ul style="list-style-type: none"> <li>11. Стаціонарні моделі?</li> <li>12. Лінійність математичних моделей?</li> <li>13. Комп'ютерна модель?</li> <li>14. Формалізація об'єкта?</li> <li>15. Методологія комп'ютерного моделювання?</li> <li>16. Інструментальна модель?</li> <li>17. Адекватність моделей?</li> <li>18. Символічні моделі?</li> <li>19. Функціональні моделі?</li> <li>20. Стахостичні моделі?</li> </ul>
19	<ul style="list-style-type: none"> <li>11. Речовинні математичні?</li> <li>12. Математичне моделювання?</li> <li>13. Детерміновані моделі?</li> <li>14. Гомоморфні моделі?</li> <li>15. Схеми моделей об'єктів?</li> <li>16. Структурний підхід?</li> <li>17. Взаємодії в системі?</li> <li>18. Визначення моделі?</li> <li>19. Аналітична модель?</li> <li>20. Особливості метода математичного моделювання?</li> </ul>
20	<ul style="list-style-type: none"> <li>11. Нестационарні моделі?</li> <li>12. Нелінійність математичних моделей?</li> </ul>

	13. Комп'ютерне моделювання? 14. Стратегічне планування? 15. Тріада «модель – алгоритм – програма» ? 16. Аналіз об'єктів моделювання? 17. Погрішність моделей? 18. Символічні моделі? 19. Лінійність математичних моделей? 20. Комп'ютерна модель?
--	---

### ЛИТЕРАТУРА:

1. Данилов А. Компьютерный практикум по курсу "Теория управления". Simulink-моделирование в среде Matlab. МГУИЭ, 2002.
2. Дьяконов В.П. Simulink. Специальный справочник. – СПб: Питер, 2002. – 528с.
3. Дьяконов В.П. MATLAB 6/6.1/6.5 + Simulink 4/5. Основы применения. Полное руководство пользователя. СОЛОН-Пресс, 2004.
4. Дьяконов В.П., Круглов В. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. Питер, 2001.
5. Поршнева С. Компьютерное моделирование физических процессов в пакете MATLAB. Горячая Линия – Телеком, 2003.
6. Чен К., Джиблин П., Ирвинг А. MATLAB в математических исследованиях. М.: Мир, 2001.
7. Ивахненко А. Г., Юрачковский Ю. П. Моделирование сложных систем по экспериментальным данным. М.: Радио и связь, 1987. 198 с.
8. Автоматизированное проектирование систем управления / Под ред. Ч. Дж. Хергета; М.: Машиностроение, 1989. 343 с.
9. Бенькович Е.С., Колесов Ю.Б., Сениченков Ю.Б. Практическое моделирование динамических систем – СПб.: БХВ-Петербург, 2002. – 464 с.:ил.
10. Бусленко И.П. Моделирование производственных процессов на цифровых вычислительных машинах. – М.: Наука, 1964. – 362 с.
11. Лазарев Ю. Моделирование процессов и систем в MATLAB. Учебный курс. – СПб.: Питер; Киев: Издательская группа ВНУ, 2005. – 512 с.:ил.
12. Дворецкий С.И., Егоров А.Ф., Дворецкий Д.С. Компьютерное моделирование и оптимизация технологических процессов и оборудования: Учеб. пособие. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2003. 224 с.