

## 2. МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКИХ ПРОЦЕССОВ СРЕДСТВАМИ ВИЗУАЛЬНОГО МОДЕЛИРОВАНИЯ SIMULINK MATLAB 6.x

### 2.1. Основы теории

Система визуального моделирования **SIMULINK MATLAB 6.x** – средство исследования динамических процессов, описываемых линейными и нелинейными дифференциальными уравнениями и последовательностями событий. Модель представляется в виде функциональной блок-схемы, именуемой S-моделью.

Для применения этой системы при построении S-моделей непрерывных систем предварительно должна быть выполнена следующая последовательность действий:

1. Уточнена цель исследования: указано, какие внешние физические переменные рассматриваются в качестве причин изменения состояния объекта в развитии изучаемого динамического процесса и какая физическая переменная является следствием – непосредственно наблюдаемым результатом. Переменные-причины называют «входами», переменная-следствие – «выходом».
2. Указать, какие промежуточные переменные целесообразно рассматривать при изучении преобразования входов в выход и от каких переменных зависят эти преобразования.
3. Провести анализ причинно-следственных связей между переменными.
4. Провести анализ конкретных зависимостей переменных-следствий от переменных-причин.
5. Построить структурную схему математической модели, в которой блоки заполняются математическими описаниями соответствующих преобразований переменных.

Рассмотрим пример.

В бак ведут две трубы. По одной в бак втекает  $q_1(t)$ , а по другой – вытекает  $q_2(t)$  литров жидкости в секунду. Это входы – причины изменения уровня воды в баке  $h(t)$ , который можно рассматривать как выход изучаемой системы.

Этот уровень зависит от количества жидкости в баке  $Q(t)$  и площади дна бака  $S$ . Установим связь между входами  $q_1(t)$  и  $q_2(t)$  и  $Q(t)$ . Очевидно, что количество жидкости в каждый момент времени  $t$  зависит от начального значения  $Q_0 = Q(0)$  и от скорости изменения потока  $Q(t) = \Delta q(t) = q_1(t) - q_2(t)$ :

$$Q(t) = Q_0 + \int_0^t \Delta q(\tau) d\tau$$

На рис.2.1 показана структурная схема модели, построенная по найденным соотношениям.

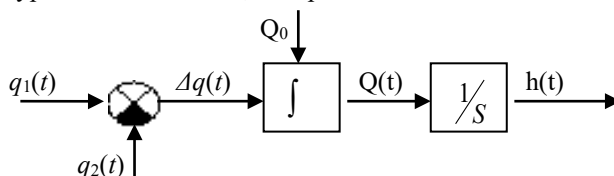


Рис.2.1.

Стрелки, представляющие переменные-причины, входят в кружок-сумматор, переменная-следствие выходит из него; Зачерненный сектор отмечает переменную, входящую в алгебраическую сумму со знаком «минус».

Пусть теперь жидкость свободно вытекает через отверстие с проходным сечением  $\sigma$  в днище бака (рис.2.2, а). В этом случае расход  $q_2 = \sigma v$  пропорционален проходному сечению  $\sigma$  отверстия, а скорость истечения жидкости  $v$  зависит от ее уровня  $h$ . Эта зависимость в данном примере описывается формулой Торичелли  $V = \sqrt{2gh}$ . Теперь расход  $q_2(t)$  не является независимой переменной, а определяется уровнем жидкости  $h(t)$  над отверстием и это показано в виде обратной связи на рис.2.2,б.

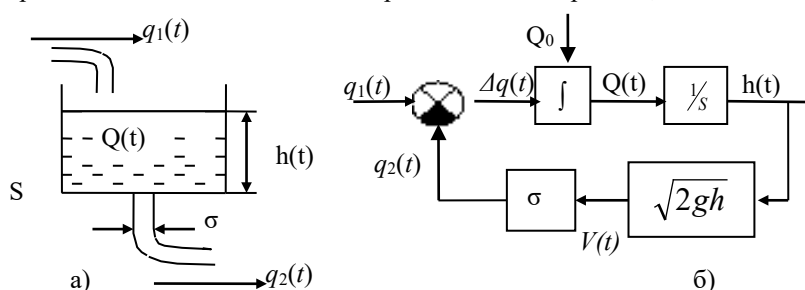


Рис.2.2.

## 2.2. Система визуального моделирования SIMULINK

### 2.2.1. Общие сведения о программе SIMULINK

Структурная схема, изображенная на рис. 2.1. представляет собой пример структурной схемы модели, подготовленной для реализации средствами инструментальной системы **SIMULINK**. Разработка S-моделей

основана на применении технологии *Drag-and-Drop* (*Перетаски и Оставь*). В составе «кирпичиков» для построения S-моделей используются модули (или блоки), хранящиеся в библиотеке **SIMULINK**. Блоки, включаемые в создаваемую модель, могут быть связаны друг с другом как по информации, так и по управлению. Тип связи зависит от типа блока и логики работы модели. Данные, которыми обмениваются блоки, могут быть скалярными величинами, векторами или матрицами произвольной размерности.

Любая S-модель может иметь иерархическую структуру, то есть состоять из моделей более низкого уровня, причем количество уровней практически не ограничено.

Наряду с другими параметрами моделирования пользователь может задавать способ изменения модельного времени (с постоянным или переменным шагом), а также условия окончания моделирования.

В ходе моделирования имеется возможность наблюдать за процессами, происходящими в исследуемой системе. Для этого существуют специальные «смотровые окна», входящие в состав библиотеки. Интересующие пользователя характеристики могут быть представлены как в числовой, так и в графической форме.

### 2.2.2. Запуск SIMULINK

Для запуска программы необходимо предварительно запустить пакет **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. 2.3).

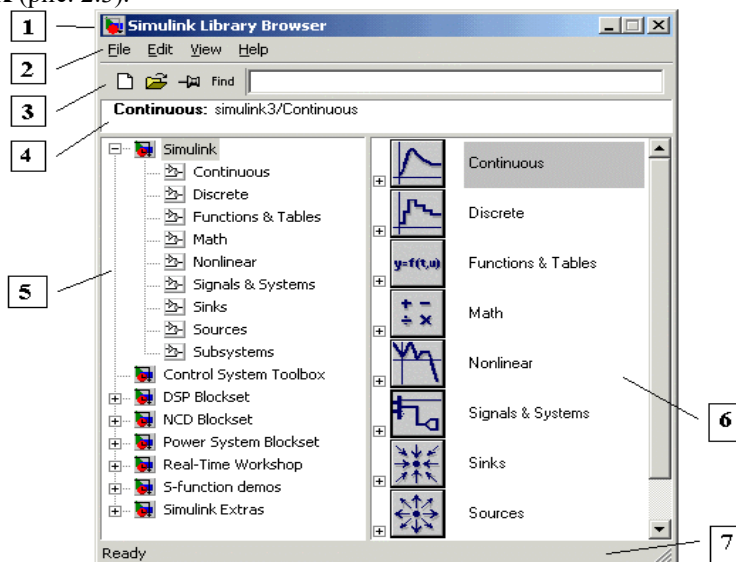


Рис.2.3.

Окно обозревателя библиотеки блоков содержит следующие элементы (рис. 2.3):

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.

На рис. 2.3 выделена основная библиотека **SIMULINK** (в левой части окна) и показаны ее разделы (в правой части окна).

### 2.2.3. Библиотека модулей

Библиотека **SIMULINK** содержит следующие основные разделы:

1. **Continuous** – непрерывные 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, и правила работы с ним являются такими же, как и при работе с деревьями каталогов Windows. Пиктограмма свернутого узла дерева содержит символ "+", а пиктограмма развернутого содержит символ "-".

Для того чтобы развернуть или свернуть узел дерева, достаточно щелкнуть на его пиктограмме ЛКМ.

При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое (Рис. 2.4).

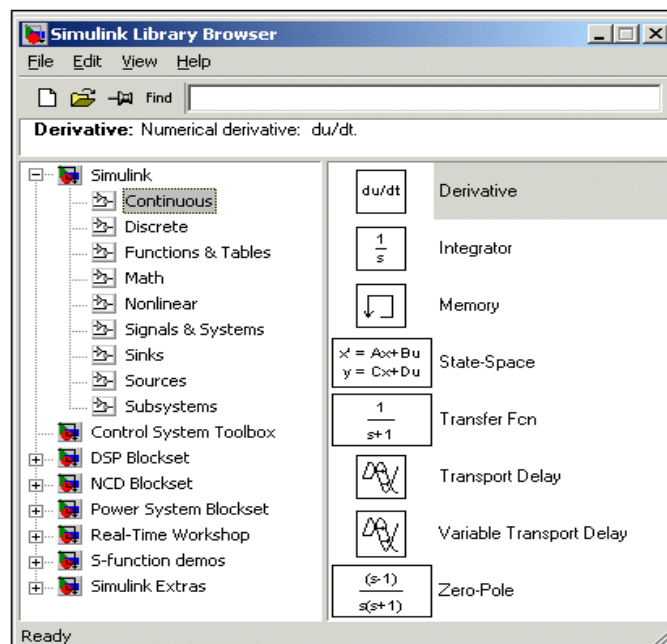


Рис 2.4.

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.2.5).

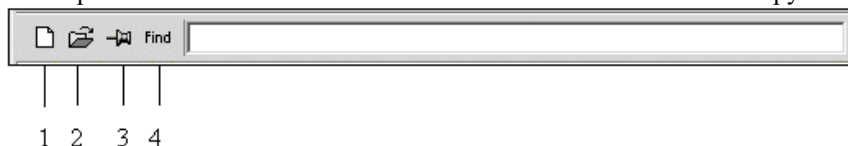


Рис.2.5.

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).

Рассмотрим основные разделы библиотеки блоков **SIMULINK**, обращая особое внимание на те из них, которые будут применяться при выполнении лабораторных работ.

### Continuous – непрерывные элементы

Раздел содержит блоки, которые можно условно разделить на две группы:

блоки, предназначенные для описания линейных непрерывных систем;

блоки общего назначения, которые могут использоваться в модели любой системы.

К первой группе относится, например, блок вычисления производной входного сигнала по времени - **Derivative**. Его назначение - численное дифференцирование входного сигнала. Для вычисления производной используется приближенная формула Эйлера:

$$\frac{du}{dt} = \frac{\Delta u}{\Delta t}$$

где  $\Delta u$  – величина изменения входного сигнала за время  $\Delta t$ ,  
 $\Delta t$  – текущее значение шага модельного времени.

Значение входного сигнала блока до начала расчета считается равным нулю. Начальное значение выходного сигнала также полагается равным нулю.

Точность вычисления производной существенно зависит от величины установленного шага расчета. Выбор меньшего шага расчета улучшает точность вычисления производной.

На рис. 2.6 показан пример использования дифференцирующего блока для вычисления производной прямоугольного сигнала. В рассматриваемом примере, для повышения наглядности, шаг расчета выбран достаточно большим.

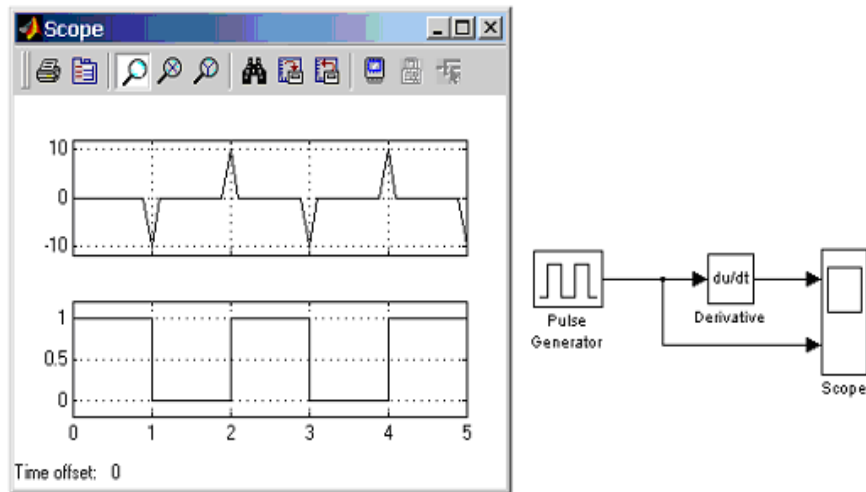


Рис.2.6.

Данный блок используется для дифференцирования аналоговых сигналов. При дифференцировании дискретного сигнала с помощью блока **Derivative** его выходной сигнал будет представлять собой последовательность импульсов соответствующих моментам времени скачкообразного изменения дискретного сигнала.

Ко второй группе относится, например, интегрирующий блок – **Integrator**. Его назначение – численное интегрирование входного сигнала.

Параметры блока:

*External reset* – внешний сброс. Тип внешнего управляющего сигнала, обеспечивающего сброс интегратора к начальному состоянию. Выбирается из списка:

*none* – нет (сброс не выполняется),

*rising* - нарастающий сигнал (передний фронт сигнала),

*falling* - спадающий сигнал (задний фронт сигнала),

*either* – нарастающий либо спадающий сигнал,

*level* – не нулевой сигнал (сброс выполняется если сигнал на управляющем входе становится не равным нулю);

В том случае, если выбран какой-либо (но не *none*), тип управляющего сигнала, то на изображении блока появляется дополнительный управляющий вход. Рядом с дополнительным входом будет показано условное обозначение управляющего сигнала.

*Initial condition source* — источник начального значения выходного сигнала интегратора, Выбирается из списка:

*internal* – внутренний

*external* – внешний. В этом случае на изображении блока появляется дополнительный вход, обозначенный  $x_0$ , на который необходимо подать сигнал задающий начальное значение выходного сигнала интегратора.

*Initial condition* — начальное условие. Установка начального значения выходного сигнала интегратора. Параметр доступен, если выбран внутренний источник начального значения выходного сигнала.

*Limit output* (флажок) — определяет, будут ли использоваться следующие параметры настройки ограничителя выходного сигнала:

*upper saturation limit* — верхний уровень ограничения выходного сигнала. Может быть задан как числом, так и символьной последовательностью **inf** – не ограничено.

*lower saturation limit* — нижний уровень ограничения выходного сигнала. Может быть задан как числом, так и символьной последовательностью **inf** – не ограничено..

*Show saturation port* — управляет отображением порта, выводящего сигнал, свидетельствующий о выходе интегратора на ограничение. Выходной сигнал данного порта может принимать следующие значения: 0 - если интегратор не находится на ограничении; +1, если выходной сигнал интегратора достиг верхнего ограничивающего предела; -1, если выходной сигнал интегратора достиг нижнего ограничивающего предела.

*Show state port* (флажок) — отобразить/скрыть порт состояния блока. Данный порт используется в том случае, если выходной сигнал интегратора требуется подать в качестве сигнала обратной связи этого же

интегратора. Например, при установке начальных условий через внешний порт или при сбросе интегратора через порт сброса. Выходной сигнал с этого порта может использоваться также для организации взаимодействия с управляемой подсистемой.

**Absolute tolerance** — абсолютная погрешность.

На рис.2.7 показан пример работы интегратора при подаче на его вход ступенчатого сигнала. Начальное условие принято равным нулю.

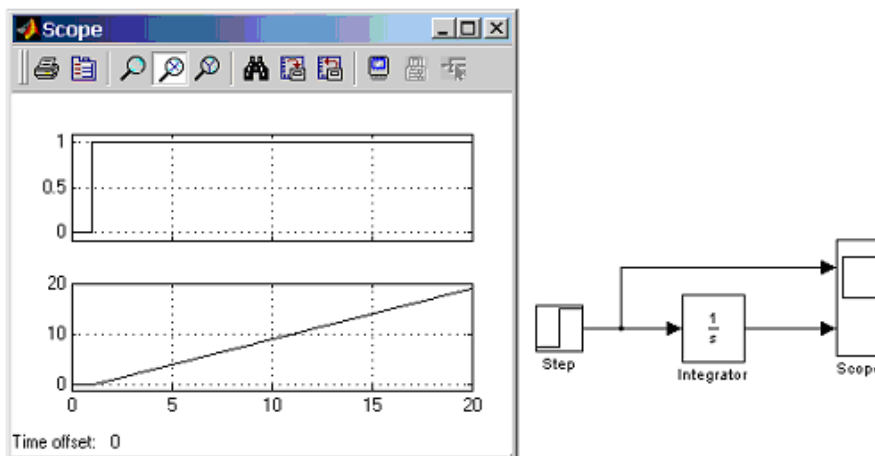


Рис. 2.7

### Sources - источники сигналов

Блоки, входящие в этот раздел, предназначены для описания рабочей нагрузки моделируемой системы, а также для формирования сигналов, обеспечивающих управление работой S – модели в целом или отдельных ее частей.

Все блоки имеют по одному входу и не имеют выходов и могут настраиваться пользователем.

Рассмотрим особенности настройки тех блоков, которые будут нужны при выполнении лабораторных работ.

**Источник постоянного сигнала Constant** задает постоянный по уровню сигнал.

Параметры блока:

*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**.

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

Рис. 2.8 иллюстрирует применение этого источника и измерение его выходного сигнала с помощью цифрового индикатора **Display**.

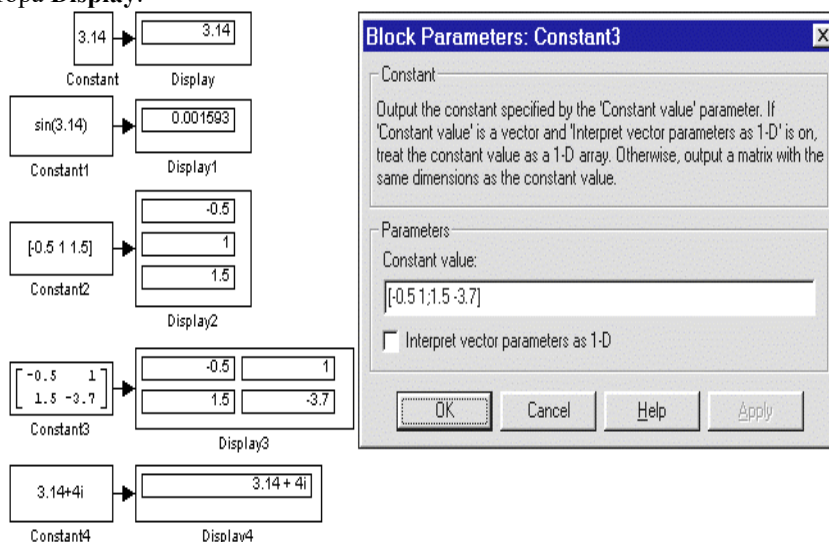


Рис. 2.8

Блок **Ramp** формирует линейный сигнал вида  $y(t) = Slope * time + Initial\ value$ .

Параметры блока:

*Slope* — скорость изменения выходного сигнала.

*Start time* — время начала формирования сигнала.

*Initial value* — начальный уровень сигнала на выходе блока.

На рис. 2.9. показано использование данного блока.

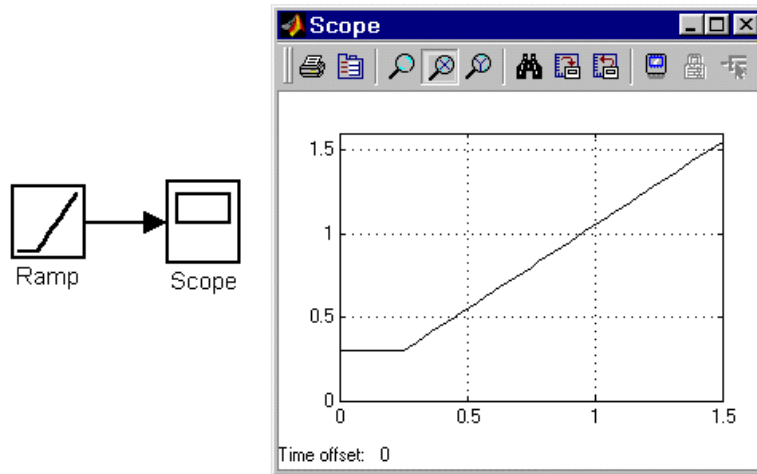


Рис. 2.9

**Генератор ступенчатого сигнала Step** формирует ступенчатый сигнал.

Параметры блока:

*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. 2.10 показано использование генератора ступенчатого сигнала.

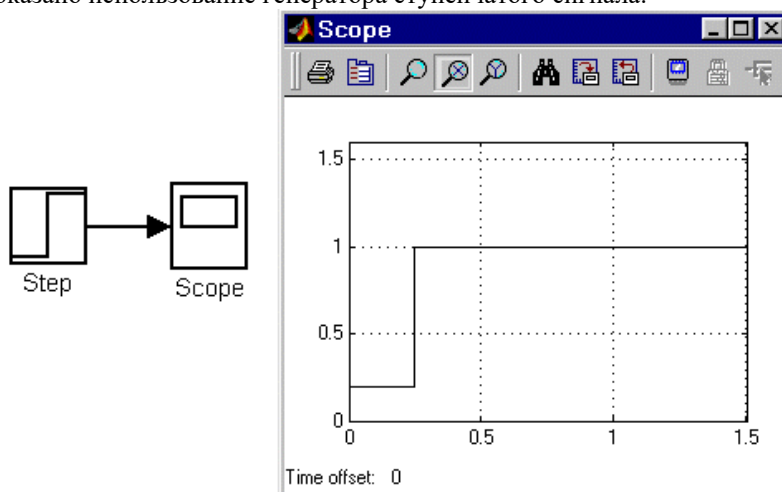


Рис. 2.10

**Источник импульсного сигнала Pulse Generator** формирует прямоугольные импульсы.

Параметры блока:

*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 % 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*.

Пример использования **Pulse Generator** показан на рис. 2.11.

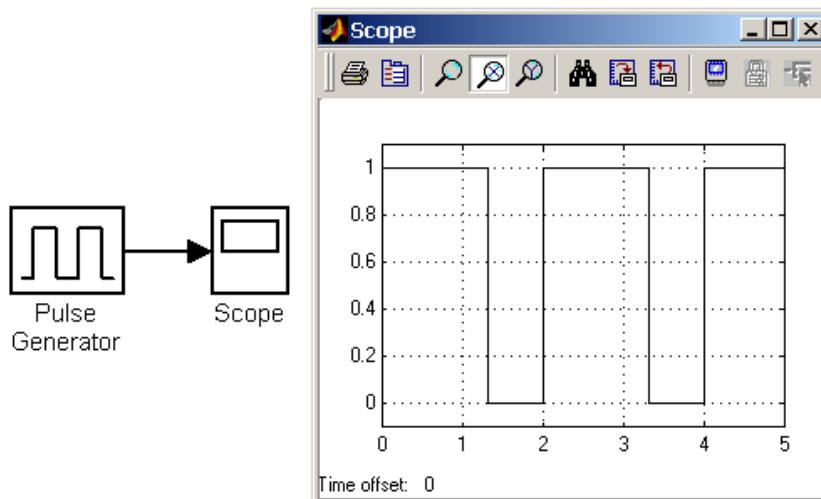


Рис. 2.11

**Блок сигнала нулевого уровня Ground** формирует сигнал нулевого уровня. Он может использоваться также и в качестве «заглушки» для тех входных портов, которые по какой-либо причине (например, на этапе отладки) оказались не подключенными к другим блокам S – модели.

**Блок входного порта Inport** создает входной порт для подсистемы или модели верхнего уровня иерархии.

Параметры блока:

*Port number* – номер порта.

*Sample time* – шаг модельного времени, определяется условиями задачи и выбирается пользователем интуитивно.

*Data type* – тип данных входного сигнала: *auto*, *double*, *single*, *int8*, *uint8*, *int16*, *uint16*, *int32*, *uint32* или *boolean*.

*Signal type* – тип входного сигнала:

*auto* – автоматическое определение типа.

*real* – действительный сигнал.

*complex* – комплексный сигнал.

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

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:

- при создании подсистемы с помощью команды *Edit/Create subsystem* входные порты создаются и нумеруются автоматически начиная с 1.
- если в подсистему добавляется новый блок **Inport**, то ему присваивается следующий по порядку номер.
- 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. 2.13 показана модель, использующая подсистему и схема этой подсистемы.

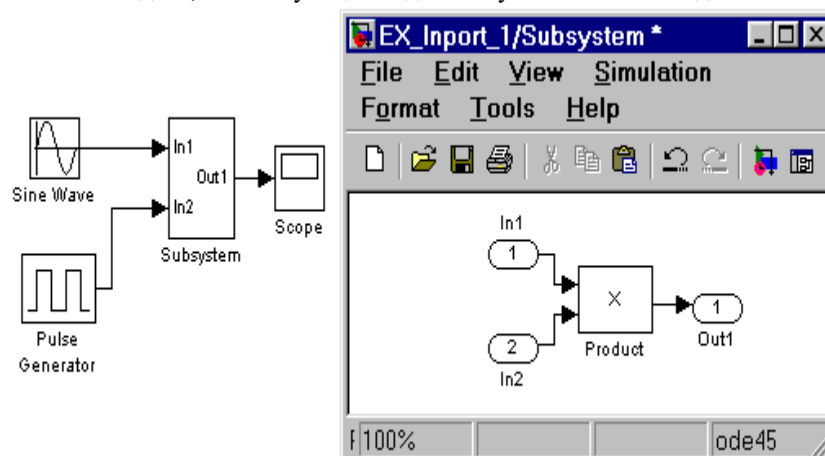




Рис. 2.13

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* (если вводится массив данных), *Structure* (данные вводятся в виде структуры, имеющей три поля: *time* – время, *signals* – сохраняемые значения сигналов если вводится структура модели *blockName* – имя модели и блока *To Workspace*) или *Structure with time* (структура с дополнительным полем "время") задается на этой же вкладке.

На рис. 2.14 показана модель, считывающая входной сигнал из рабочего пространства **MATLAB**. Подсистема **Load Data** обеспечивает ввод данных из файла в рабочую область **MATLAB**.

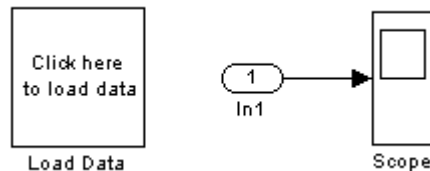


Рис. 2.14

### Sinks - приемники сигналов

Блоки, собранные в этом разделе, условно можно разделить на три группы:

1. Блоки, используемые при моделировании и в качестве «смотровых окон»:
  - блок **Scope** (индикатор);
  - блок **XY Graph** (графопостроитель) обеспечивает создание двумерных графиков в прямоугольной системе координат;
  - блок **Display** (экран) предназначен для отображения численных значений величин.
2. Блоки, обеспечивающие сохранение промежуточных и/или выходных результатов моделирования:
  - блок **To File** – запись в файл;
  - блок **To Workspace** – запись в рабочую область;
  - блок выходного порта **Output** – обеспечивает сохранение промежуточных данных.
3. Блок управления моделированием (остановка моделирования) позволяет прервать моделирование при выполнении тех или иных условий. Блок срабатывает, если на его вход поступает ненулевой сигнал.

Рассмотрим подробнее блоки, используемые при выполнении лабораторных работ.

**Осциллограф Scope** 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. 2.15.

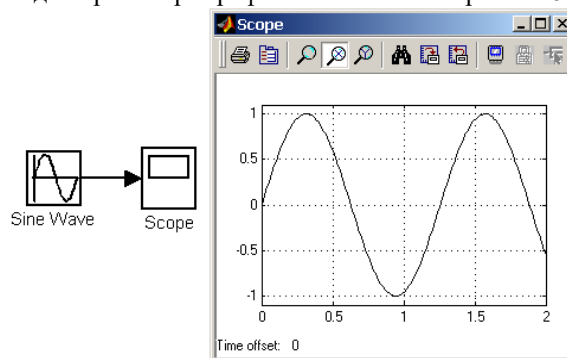


Рис. 2.15

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.2.16).

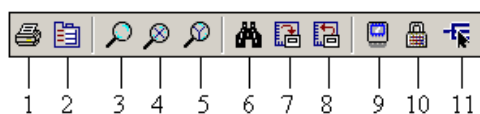


Рис. 2.16

Панель инструментов содержит 11 кнопок:

1. *Print* – печать содержимого окна осциллографа.
2. *Parameters* – доступ к окну настройки параметров.
3. *Zoom* – увеличение масштаба по обеим осям.



4. *Zoom X-axis* – увеличение масштаба по горизонтальной оси.
5. *Zoom Y-axis* – увеличение масштаба по вертикальной оси.
6. *Autoscale* – автоматическая установка масштабов по обеим осям.
7. *Save current axes settings* – сохранение текущих настроек окна.
8. *Restore saved axes settings* – установка ранее сохраненных настроек окна.
9. *Floating scope* – перевод осциллографа в “свободный” режим.
10. *Lock/Unlock axes selection* – закрепить/разорвать связь между текущей координатной системой окна и отображаемым сигналом. Инструмент доступен, если включен режим *Floating scope*.
11. *Signal selection* – выбор сигналов для отображения. Инструмент доступен, если включен режим *Floating scope*.

Изменение масштабов отображаемых графиков можно выполнять несколькими способами:

1. Нажать соответствующую кнопку (3,4,5) и щелкнуть один раз левой клавишей “мыши” в нужном месте графика. Произойдет 2,5 кратное увеличение масштаба.
2. Нажать соответствующую кнопку (3, 4 или 5) и, нажав левую клавишу “мыши”, с помощью динамической рамки или отрезка указать область графика для увеличенного изображения. Рис. 2.17 поясняет этот процесс.

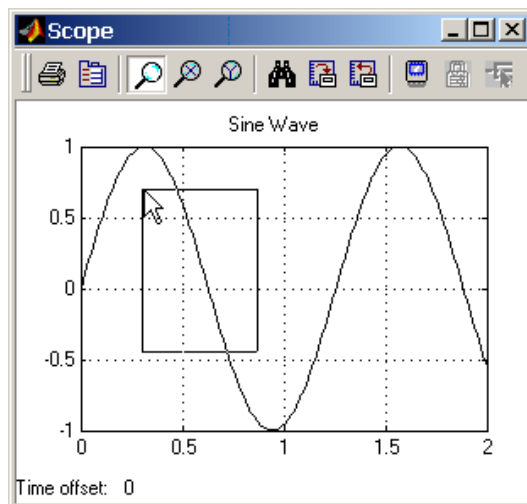


Рис. 2.17

3. 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. 2.18.

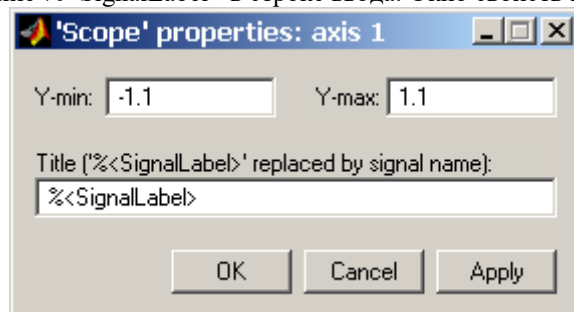


Рис 2.18.

Параметры блока **Scope** устанавливаются в окне *'Scope' parameters*, которое открывается с помощью инструмента 2 (*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. рис. 2.19.

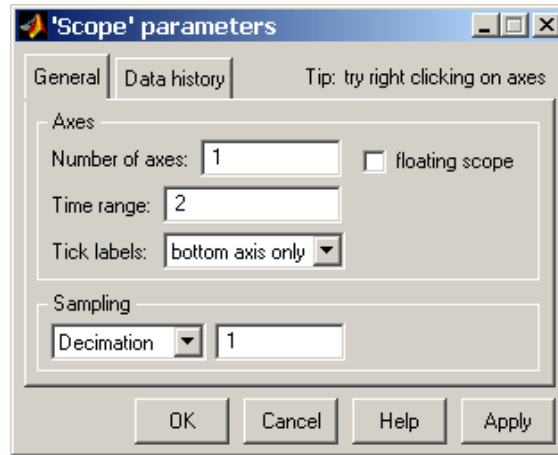


Рис. 2.19

На вкладке *General* (рис.2.19) можно задавать 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. Задаст режим вывода расчетных точек на экран. При выборе *Decimation* кратность вывода устанавливается числом, задающим шаг выводимых расчетных точек. На рис. 2.20 и 2.21 показаны графики синусоидальных сигналов рассчитанных с фиксированным шагом **0.1** с. На рис. 2.20 в окне блока **Scope** выводится каждая расчетная точка (параметр *Decimation* равен **1**). На рис. 2.21 показан вывод каждого второго значения (параметр *Decimation* равен **2**). Маркерами на графиках отмечены расчетные точки.

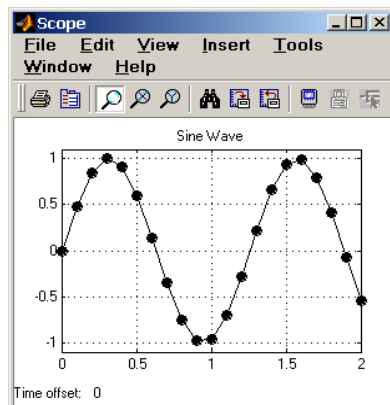


Рис.2.20

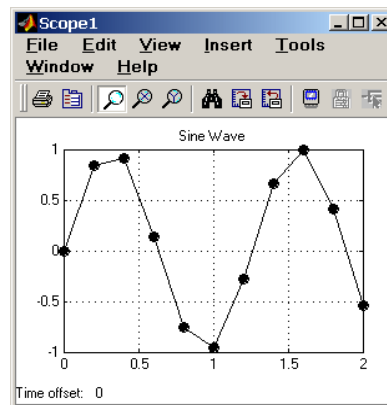


Рис.2.21

*Floating scope* – перевод осциллографа в “свободный” режим (при установленном флажке).

На вкладке *Data history* (рис. 2.22) 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) данные сохраняются как массив, в котором число строк определяется числом расчетных точек по времени, а число столбцов – размерностью вектора подаваемого на вход блока. Если на вход подается скалярный сигнал, то матрица будет содержать лишь один столбец; *Structure* – (структура) данные сохраняются в виде структуры, имеющей три поля: *time* – время, *signals* – сохраняемые значения сигналов, *blockName* – имя модели и блока *To Workspace*. Поле *time* для данного формата остается не заполненным; *Structure with time* – структура с дополнительным полем “время”. Для данного формата, в отличие от предыдущего, поле *time* заполняется значениями времени.

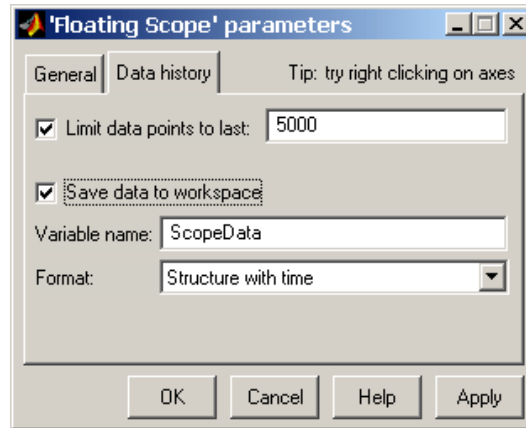


Рис. 2.22.

**Графопостроитель XY Graph** строит график одного сигнала в функции другого (график вида  $Y(X)$ ).

Параметры блока:

$x_{-min}$  – 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. 2.23, в качестве примера использования графопостроителя, показано построение фазовой траектории колебательного звена.

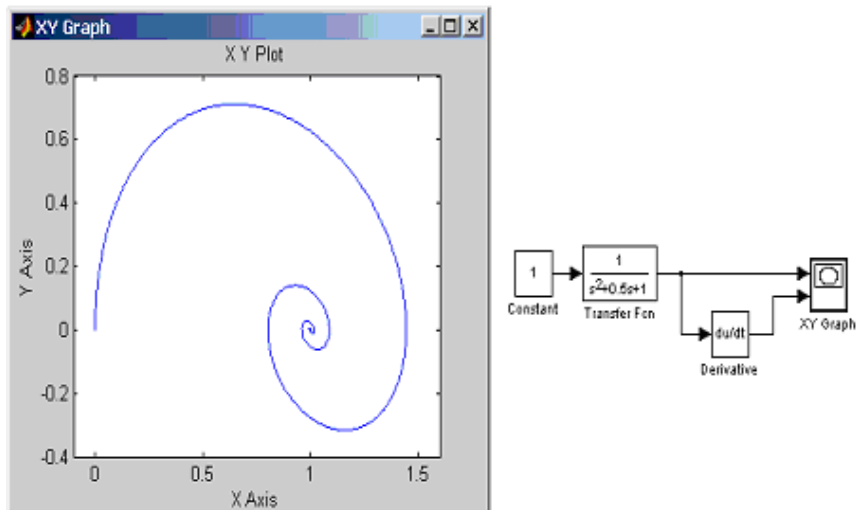


Рис. 2.23

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. 2.24.

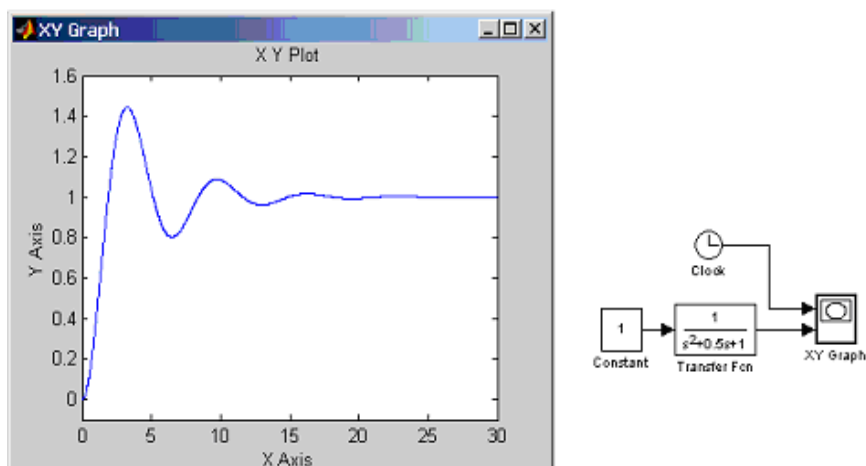


Рис. 2.24

**Блок сохранения данных в файле To File** записывает данные, поступающие на его вход, в файл.

Параметры блока:

*Filename* – имя файла для записи. По умолчанию файл имеет имя *untitled.mat*. Если не указан полный путь файла, то файл сохраняется в текущей рабочей папке.

*Variable name* – имя переменной, содержащей записываемые данные.

*Decimation* – кратность записи в файл входного сигнала. При *Decimation* = 1 записывается каждое значение входного сигнала, при *Decimation* = 2 записывается каждое второе значение, при *Decimation* = 3 – каждое третье значение и т.д.

*Sample time* – шаг модельного времени. Определяет дискретность записи данных.

Данные в файле сохраняются в виде матрицы:

$$\begin{bmatrix} t_1 & t_2 & \dots & t_{final} \\ u1_1 & u1_2 & \dots & u1_{final} \\ \dots & \dots & \dots & \dots \\ un_1 & un_2 & \dots & un_{final} \end{bmatrix}$$

Значения времени записываются в первой строке матрицы, а в остальных строках будут находиться значения сигналов, соответствующих данным моментам времени.

Файл данных (**mat**-файл), в который записываются данные, не является текстовым. Структура файла подробно описана в справочной системе **MATLAB**. Пользователям **SIMULINK** удобнее всего считывать данные из **mat**-файла с помощью блока **From File** (библиотека **Sources**).

На рис. 2.25 показан пример использования данного блока. Результаты расчета сохраняются в файле **result.mat**.

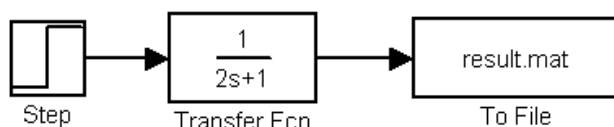


Рис. 2.25

**Блок сохранения данных в рабочей области To Workspace** записывает данные, поступающие на его вход, в рабочую область **MATLAB**.

Параметры блока:

*Variable name* – имя переменной, содержащей записываемые данные.

*Limit data points to last* – максимальное количество сохраняемых расчетных точек по времени (отсчет ведется от момента начала моделирования). В том случае, если значение параметра *Limit data points to last* задано как **inf**, то в рабочей области будут сохранены все данные.

*Decimation* – кратность записи данных в рабочую область.

*Sample time* – шаг модельного времени. Определяет дискретность записи данных.

*Save format* – формат сохранения данных. Может принимать значения: *Matrix* (матрица) – данные сохраняются как массив, в котором число строк определяется числом расчетных точек по времени, а число столбцов – размерностью вектора подаваемого на вход блока; *Structure* – структура. Данные сохраняются в виде структуры, имеющей три поля. *Structure with Time* – структура с дополнительным полем (время).

На рис. 2.26 показан пример использования блока **To Workspace**. Результаты расчета сохраняются в переменной *simout*.

Для считывания данных, сохраненных в рабочей области **MATLAB**, можно использовать блок **From Workspace** (библиотека **Sources**).

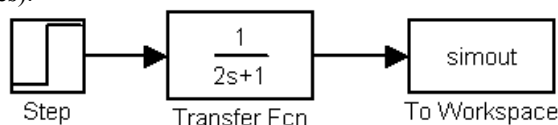


Рис. 2.26

**Блок выходного порта Outport** создает выходной порт для подсистемы или для модели верхнего уровня иерархии.

Параметры блока:

*Port number* – номер порта;

*Output when disabled* – вид сигнала на выходе подсистемы, в случае если подсистема выключена. Используется для управляемых подсистем. Может принимать значения (выбираются из списка): *held* – выходной сигнал подсистемы равен последнему рассчитанному значению; *reset* – выходной сигнал подсистемы равен значению задаваемому параметром *Initial output*.

*Initial output* – значение сигнала на выходе подсистемы до начала ее работы и в случае, если подсистема выключена. Используется для управляемых подсистем.

Блоки **Outport** подсистемы являются ее выходами. Сигнал, подаваемый в блок **Outport** внутри подсистемы, передается в модель (или подсистему) верхнего уровня. Название выходного порта будет показано на изображении подсистемы как метка порта.

При создании подсистем и добавлении блока **Outport** в подсистему **SIMULINK** использует следующие правила:

- при создании подсистемы с помощью команды *Edit/Create subsystem* выходные порты создаются и нумеруются автоматически начиная с 1.
- если в подсистему добавляется новый блок **Outport**, то ему присваивается следующий по порядку номер.
- если какой либо блок **Outport** удаляется, то остальные порты переименовываются таким образом, чтобы последовательность номеров портов была непрерывной.

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. 2.27 показана модель, использующая подсистему и схема этой подсистемы.

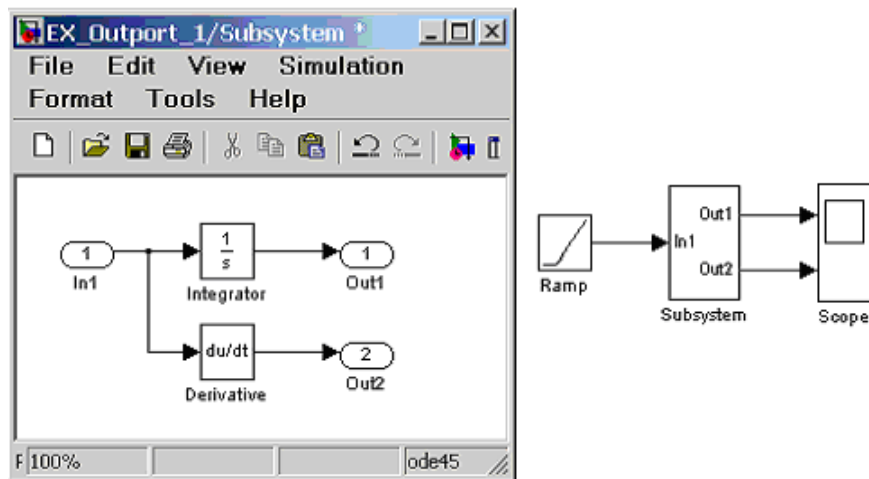


Рис. 2.27

### Math – математические блоки

Раздел Math содержит блоки, которые реализуют элементарные (алгебраические и тригонометрические) функции, а также операции математической логики.

**Блок вычисления суммы Sum** выполняет вычисление суммы текущих значений сигналов. Он может использоваться в двух режимах:

1. Сложение входных сигналов (в том числе с разными знаками);
2. Суммирование элементов вектора, поступающего на вход блока.

Параметры блока:

*Icon shape* – форма блока. Выбирается из списка: *round* – окружность; *rectangular* – прямоугольник.

*List of sign* – список знаков. значения могут задаваться одним из трех способов: 1) в виде последовательности знаков + и –, причем число знаков определяет число входов блока, а сами знаки – полярности соответствующих входных сигналов; 2) в виде целой положительной константы, большей 1, значение которой равно числу входов блока, а все входы считаются положительными; 3) в виде константы 1, что означает суммирование элементов вектора, подаваемого на вход в виде последовательности значений, заключенных в квадратные скобки (в этом случае внутри блока выводится символ  $\Sigma$ ); В списке можно использовать следующие знаки: + (плюс), - (минус) и | (разделитель знаков).

*Saturate on integer overflow* (флажок) – подавлять переполнение для целых чисел. При установленном флажке дает возможность задать режим «усечения» результата сложения, если он выходит за пределы диапазона, установленного для целочисленных значений.

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 3, 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. 2.28.

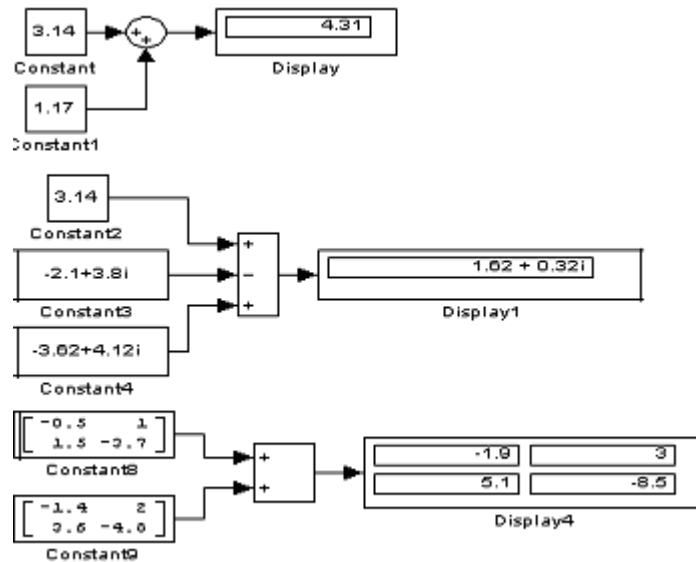


Рис.2.28

**Блоки усиления сигналов Gain и Matrix Gain** выполняют умножение входного сигнала на постоянный коэффициент.

Параметры блока:

*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.

Блоки усилителей *Gain* и *Matrix Gain* есть один и тот же блок, но с разными начальными установками параметра *Multiplication*.

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 *Element-wise  $K*u$* , 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*.

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. 2.29.

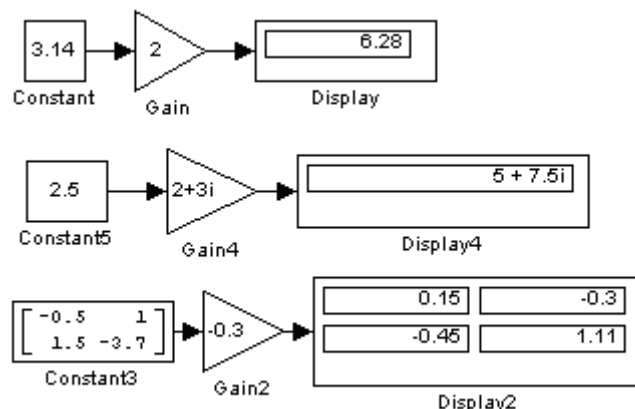


Рис.2.29

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 рис. 2.30.

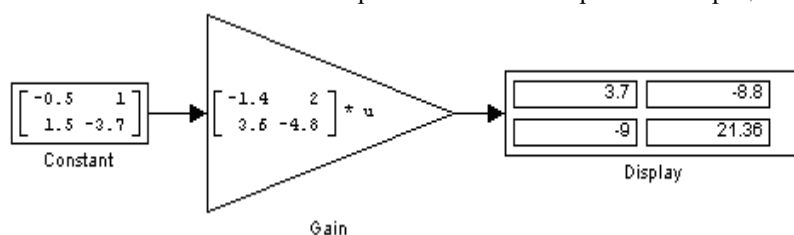


Рис.2.30

#### 2.2.4. 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 кнопки панели инструментов рис.2.5. Новое созданное окно модели показано на рис. 2.31.

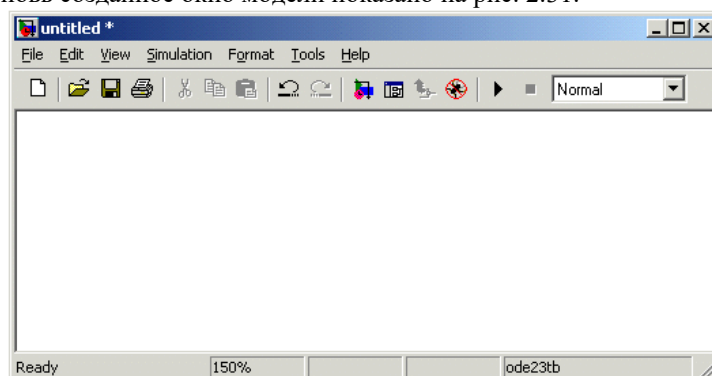


Рис 2.31

2. Расположить блоки в окне модели. Для этого необходимо открыть соответствующий раздел библиотеки и затем, указав курсором на требуемый блок и нажав ЛКМ “перетащить” блок в созданное окно. **Клавишу мыши нужно держать нажатой.** На рис 2.32 показано окно модели, содержащее блоки.

Для удаления блока следует выбрать блок, указав курсором на его изображение и нажать ЛКМ, 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, можно 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 **<OK>**.

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. 2.32.

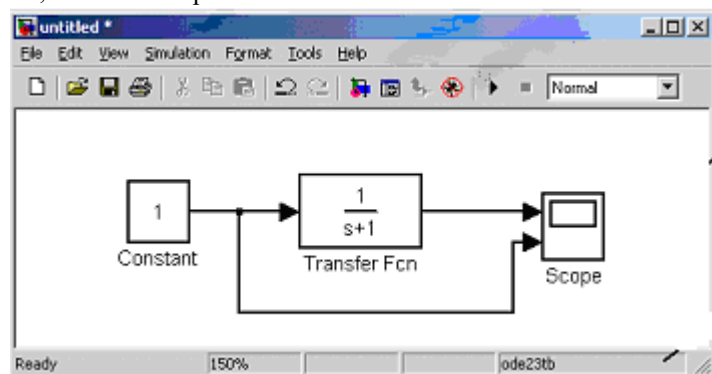




Рис 2.32

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. Имя файла не должно превышать 32 символов, должно начинаться с буквы и не может содержать символы кириллицы и спецсимволы. 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**.

Следует заметить, что в **MATLAB** имеют особое значение файлы нескольких типов:

файлы с расширением *.mat* – бинарные файлы, в которых могут храниться значения переменных;

файлы с расширением *.m* – текстовые файлы, содержащие внешние программы, определения команд и функций системы;

файлы с расширением *.tex* – откомпилированные коды;

файлы с расширением *.mdl* – S-модель функциональной блок-схемы, созданной в **SIMULINK**.

### Окно модели

Окно модели содержит следующие элементы:

**Заголовок**, с названием окна. Вновь созданному окну присваивается имя **Untitled** с соответствующим номером.

**Меню** с командами *File, Edit, View* и т.д.

**Панель инструментов**.

**Окно для построения схемы** модели.

**Строка состояния**, содержащая информацию о текущем состоянии модели.

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

**File (файл)** — работа с файлами моделей.

**Edit (редактирование)** — изменение модели и поиск блоков.

**View (вид)** — управление показом элементов интерфейса.

**Simulation (моделирование)** — задание настроек для моделирования и управление процессом расчета.

**Format (форматирование)** — изменение внешнего вида блоков и модели в целом.

**Tools (инструментальные средства)** — применение специальных средств для работы с моделью (отладчик, линейный анализ и т.п.)

**Help (справка)** — вывод окон справочной системы.

Для работы с моделью можно также использовать кнопки на панели инструментов (рис.2.33).

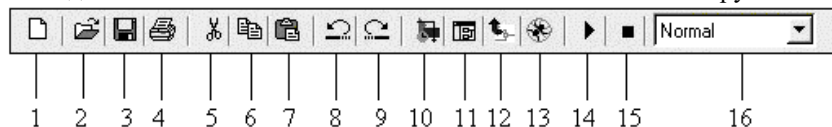



Рис 2.33

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). В строке состояния отображаются также:

- 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),
- используемый алгоритм расчета состояний модели (метод решения).

## 2.2.5. Основные приемы подготовки и редактирования модели

### Добавление текстовых надписей

Для создания надписи нужно указать курсором мыши место надписи и дважды щелкнуть ЛКМ. После этого появится прямоугольная рамка с курсором ввода. Аналогичным образом можно изменить и подписи к блокам моделей..

### Выделение объектов

Для выполнения какого-либо действия с элементом модели (блоком, соединительной линией, надписью) его необходимо сначала выделить.

Выделение объектов проще всего осуществляется мышью. Для этого необходимо установить курсор мыши на нужном объекте и щелкнуть ЛКМ. О выделении объекта будут свидетельствовать маркеры по углам объекта. Можно также выделить несколько объектов. Для этого надо установить курсор мыши вблизи группы объектов, нажать ЛКМ и, не отпуская ее, начать перемещать мышшь. Появится пунктирная рамка, размеры которой будут изменяться при перемещении мыши. Все охваченные рамкой объекты становятся выделенными. Выделить все объекты также можно, используя команду *Edit/Select All*. После выделения объекта его можно копировать или перемещать в буфер промежуточного хранения, извлекать из буфера, а также удалять, используя стандартные приемы работы в **WINDOWS**-программах.

### Копирование и перемещение объектов в буфер промежуточного хранения

Для копирования объекта в буфер его необходимо предварительно выделить, а затем выполнить команду *Edit/Copy* или воспользоваться кнопкой 6 на панели инструментов (рис.2.33).

Копирование можно выполнить и таким образом: нажать **правую** клавишу мыши, и не отпуская ее, переместить объект. При этом будет создана копия объекта, которую можно переместить в необходимое место.

Для вырезания объекта его необходимо предварительно выделить, а затем выполнить команду *Edit/Cut* или воспользоваться кнопкой 5 на панели инструментов. При выполнении данных операций следует иметь в виду, что объекты помещаются в собственный буфер **MATLAB** и недоступны из других приложений. Использование команды *Edit/Copy model to Clipboard* позволяет поместить **графическое изображение** модели в буфер **WINDOWS**, что, соответственно, делает его доступным для остальных программ.

### Вставка объектов из буфера промежуточного хранения

Для вставки объекта из буфера необходимо предварительно указать место вставки, ЛКМ в предполагаемом месте вставки, а затем выполнить команду *Edit/Paste* или воспользоваться кнопкой 7 на панели инструментов.

### Удаление объектов

Для удаления объекта его необходимо предварительно выделить, а затем выполнить команду **Edit/Clear** или воспользоваться клавишей **Delete** на клавиатуре. Следует учесть, что команда **Clear** удаляет блок без помещения его в буфер обмена. Эту операцию можно отменить командой меню **File/Undo**.

### Перемещение блоков

Любой блок модели можно переместить, выделив его, и передвинуть, держа нажатой ЛКМ. Если ко входам и выходам блока подведены соединительные линии, то они не разрываются, а лишь сокращаются или увеличиваются по длине. В соединении можно также вставить блок, имеющий один вход и один выход. Для этого его нужно расположить в требуемом месте соединительной линии.

### Использование команд Undo и Redo

В процессе освоения программы пользователь может совершать ошибочные действия (например, случайное удаление части модели, копирование и т.д.). В этом случае следует воспользоваться командой *Undo* — отмена последней операции. Команду можно вызвать с помощью кнопки 8 в панели инструментов или из меню *Edit*. Для восстановления отмененной операции служит команда *Redo* (кнопка 9).

### Форматирование объектов

В меню **Format** (также как и в контекстном меню, вызываемом нажатием правой клавиши мыши на объекте) находится набор команд форматирования блоков. Команды форматирования разделяются на несколько групп:

#### 1. Изменение отображения надписей:

*Font* — форматирование шрифта надписей и текстовых блоков;

*Text alignment* — выравнивание текста в текстовых надписях;

*Flip name* — перемещение подписи блока;

*Show/Hide name* — отображение или скрытие подписи блока.

#### 2. Изменение цветов отображения блоков:

*Foreground color* — выбор цвета линий для выделенных блоков;

*Background color* — выбор цвета фона выделенных блоков;

*Screen color* — выбор цвета фона для всего окна модели.

3. Изменение положения блока и его вида:

*Flip block* — зеркальное отображение относительно вертикальной оси симметрии;

*Rotate block* — поворот блока на 90° по часовой стрелке;

*Show drop shadow* — показ тени от блока;

*Show port labels* — показ меток портов.

4. Прочие установки:

*Library link display* — показ связей с библиотеками;

*Sample time colors* — выбор цвета блока индикации времени;

*Wide nonscalar lines* — увеличение/уменьшение ширины не скалярных линий;

*Signal dimensions* — показ размерности сигналов;

*Port data types* — показ данных о типе портов;

*Storage class* — класс памяти. Параметр, устанавливаемый при работе **Real-Time Workshop**;

*Execution order* — вывод порядкового номера блока в последовательности исполнения.

### 2.2.6. Установка параметров расчета и его выполнение

Перед выполнением расчетов необходимо предварительно задать параметры расчета. Задание параметров расчета выполняется в панели управления меню **Simulation/Parameters**. Вид панели управления приведен на рис.2.34.

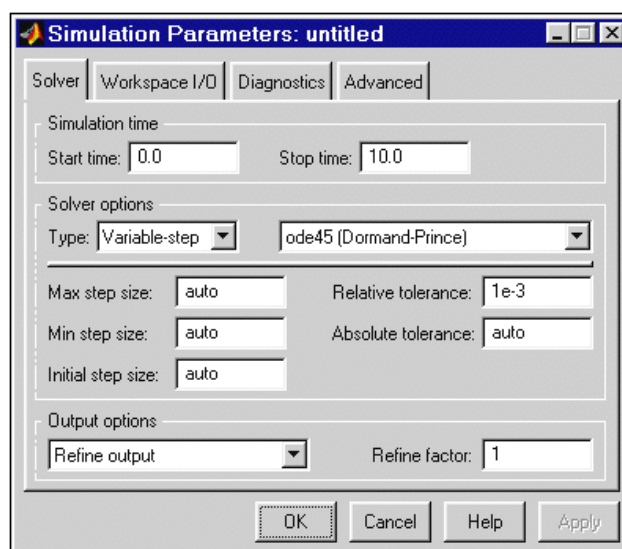


Рис 2.34. Панель управления

Окно настройки параметров расчета имеет 4 вкладки:

**Solver (расчет)** — установка параметров расчета модели;

**Workspace I/O** (ввод/вывод данных в рабочую область) — установка параметров обмена данными с рабочей областью **MATLAB**;

**Diagnostics** (диагностика) — выбор параметров диагностического режима;

**Advanced** (дополнительно) — установка дополнительных параметров.

Установка параметров расчета модели выполняется с помощью элементов управления, размещенных на вкладке **Solver**. Эти элементы разделены на три группы (рис.2.35): **Simulation time** (интервал моделирования или, иными словами, время расчета), **Solver options** (параметры расчета), **Output options** (параметры вывода).

#### Установка параметров расчета модели Solver

**Simulation time** (интервал моделирования или время расчета)

Время расчета задается указанием начального (*Start time*) и конечного (*Stop time*) значений времени расчета. Начальное время, как правило, задается равным нулю. Величина конечного времени задается пользователем исходя из условий решаемой задачи. Время моделирования является условным и понятием и, как правило, не совпадает с реальным.

**Solver options** (параметры расчета)

При выборе параметров расчета необходимо указать способ моделирования (*Type*) и метод расчета нового состояния системы. Для параметра *Type* доступны два варианта - с фиксированным (*Fixed-step*) или с переменным (*Variable-step*) шагом. Как правило, *Variable-step* используется для моделирования непрерывных систем, а *Fixed-step* - для дискретных.

Список методов расчета нового состояния системы содержит несколько вариантов. Первый вариант (*discrete*) используется для расчета дискретных систем. Остальные методы используются для расчета непрерывных систем. Эти методы различны для переменного (*Variable-step*) и для фиксированного (*Fixed-step*) шага времени, но, по сути, представляют собой процедуры решения систем дифференциальных уравнений. Подробное описание каждого из методов расчета состояний системы приведено во встроенной справочной системе **MATLAB**.

Ниже двух раскрывающихся списков **Type** находится область, содержимое которой меняется в зависимости от выбранного способа изменения модельного времени. При выборе *Fixed-step* в данной области появляется текстовое поле **Fixed-step size** (величина фиксированного шага) позволяющее указывать величину шага моделирования (рис.2.35). Величина шага моделирования по умолчанию устанавливается системой автоматически (*auto*). Требуемая величина шага может быть введена вместо значения *auto* либо в форме числа, либо в виде вычисляемого выражения (то же самое относится и ко всем параметрам устанавливаемым системой автоматически).

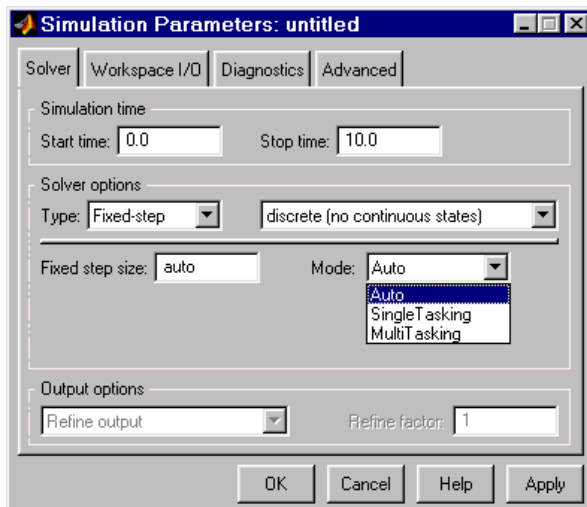


Рис 2.35

При выборе *Fixed-step* необходимо также задать режим расчета (*Mode*). Для параметра *Mode* доступны три варианта:

*MultiTasking* (многозадачный) – необходимо использовать, если в модели присутствуют параллельно работающие подсистемы, и результат работы модели зависит от временных параметров этих подсистем.

*SingleTasking* (однозадачный) - используется для тех моделей, в которых недостаточно строгая синхронизация работы отдельных составляющих не влияет на конечный результат моделирования.

*Auto* (автоматический выбор режима) - позволяет **SIMULINK** автоматически устанавливать режим *MultiTasking* для тех моделей, в которых используются блоки с различными скоростями передачи сигналов и режим *SingleTasking* для моделей, в которых содержатся блоки, оперирующие одинаковыми скоростями.

При выборе *Variable-step* в области появляются поля для установки трех параметров:

*Max step size* - максимальный шаг расчета. По умолчанию он устанавливается автоматически (**auto**) и его значение в этом случае равно  $(SfopTime - StartTime)/50$ . Довольно часто это значение оказывается слишком большим, и наблюдаемые графики представляют собой ломаные (а не плавные) линии. В этом случае величину максимального шага расчета необходимо задавать явным образом.

*Min step size* - минимальный шаг расчета.

*Initial step size* - начальное значение шага моделирования.

При моделировании непрерывных систем с использованием переменного шага необходимо указать точность вычислений: относительную (*Relative tolerance*) и абсолютную (*Absolute tolerance*). По умолчанию они равны соответственно  $10^{-3}$  и **auto**.

#### **Output options (параметры вывода)**

В нижней части вкладки **Solver** задаются настройки параметров вывода выходных сигналов моделируемой системы (*Output options*). Для данного параметра возможен выбор одного из трех вариантов:

*Refine output* (скорректированный вывод) – позволяет изменять дискретность регистрации модельного времени и тех сигналов, которые сохраняются в рабочей области **MATLAB** с помощью блока **To Workspace**. Установка величины дискретности выполняется в строке редактирования *Refine factor*, расположенной справа. По умолчанию значение *Refine factor* равно 1, это означает, что регистрация производится с шагом  $\Delta t = 1$  (то есть для каждого значения модельного времени:). Если задать *Refine factor* равным 2, это означает, что будет регистрироваться каждое второе значение сигналов, 3 - каждое третье и т. д. Параметр *Refine factor* может принимать только целые положительные значения

*Produce additional output* (дополнительный вывод) — обеспечивает дополнительную регистрацию параметров модели в заданные моменты времени; их значения вводятся в строке редактирования (в этом случае она называется *Output times*) в виде списка, заключенного в квадратные скобки. При использовании этого варианта базовый шаг регистрации ( $\Delta t$ ) равен 1. Значения времени в списке *Output times* могут быть дробными числами и иметь любую точность.

*Produce specified output only* (формировать только заданный вывод) — устанавливает вывод параметров модели только в заданные моменты времени, которые указываются в поле *Output times* (Моменты времени вывода).

#### **Установка параметров обмена с рабочей областью Workspace I/O**

Элементы, позволяющие управлять вводом и выводом в рабочую область **MATLAB** промежуточных данных и результатов моделирования, расположены на вкладке **Workspace I/O** (рис. 2.36).

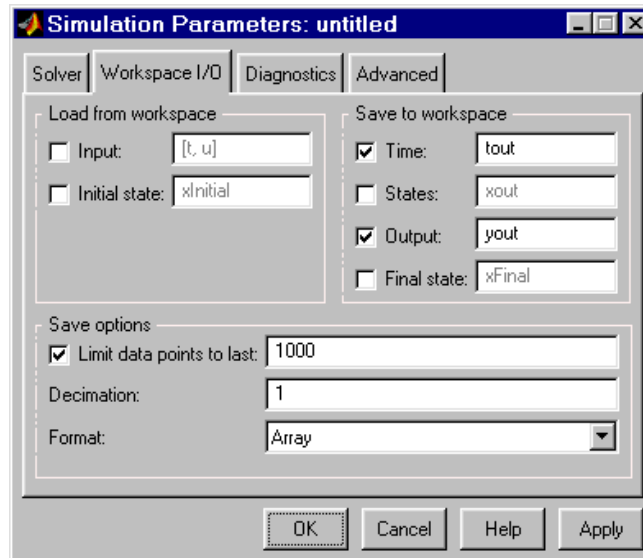


Рис 2.36

Элементы вкладки разделены на 3 поля:

*Load from workspace* (загрузить из рабочей области). Если флажок *Input* (Входные данные) установлен, то в расположенном справа текстовом поле можно ввести формат данных, которые будут считываться из рабочей области **MATLAB**. Установка флажка *Initial State* (Начальное состояние) позволяет ввести в связанном с ним текстовом поле имя переменной, содержащей параметры начального состояния модели. Данные, указанные в полях *Input* и *Initial State*, передаются в исполняемую модель посредством одного или более блоков *In* (из раздела библиотеки *Sources*).

*Save to workspace* (записать в рабочую область) – Позволяет установить режим вывода значений сигналов в рабочую область **MATLAB** и задать их имена.

*Save options* (параметры записи) – Задаёт количество строк при передаче переменных в рабочую область. Если флажок *Limit rows to last* установлен, то в поле ввода можно указать количество передаваемых строк (отсчет строк производится от момента завершения расчета). Если флажок не установлен, то передаются все данные. Параметр *Decimation* (Исключение) задает шаг записи переменных в рабочую область (аналогично параметру *Refine factor* вкладки **Solver**). Параметр *Format* (формат данных) задает формат передаваемых в рабочую область данных. Доступные форматы *Array* (Массив), *Structure* (Структура), *Structure With Time* (Структура с дополнительным полем – “время”).

#### Установка параметров диагностирования модели **Diagnostics**

Вкладка **Diagnostics** (рис. 2.37) позволяет изменять перечень диагностических сообщений, выводимых **SIMULINK** в командном окне **MATLAB**, а также устанавливать дополнительные параметры диагностики модели.

Сообщения об ошибках или проблемных ситуациях, обнаруженных **SIMULINK** в ходе моделирования и требующих вмешательства разработчика выводятся в командном окне **MATLAB**. Исходный перечень таких ситуаций и вид реакции на них приведен в списке на вкладке **Diagnostics**. Разработчик может указать вид реакции на каждое из них, используя группу переключателей в поле *Action* (они становятся доступны, если в списке выбрано одно из событий):

*None* — игнорировать,

*Warning* — выдать предупреждение и продолжить моделирование,

*Error* — выдать сообщение об ошибке и остановить сеанс моделирования.

Выбранный вид реакции отображается в списке рядом с наименованием события.

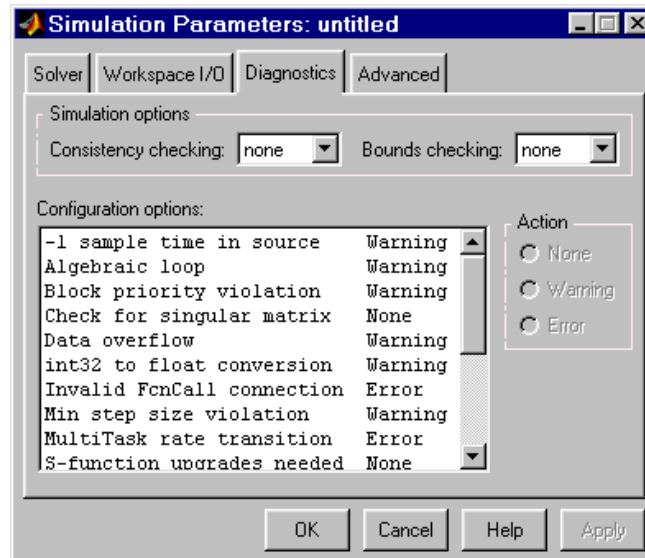


Рис 2.37

### Выполнение расчета

Запуск расчета выполняется выбором пункта меню **Simulation/Start** или кнопки 14 на панели инструментов. Процесс расчета можно завершить досрочно, выбрав пункт меню **Simulation/Stop** или кнопки 15. Расчет также можно остановить (*Simulation/Pause*) и затем продолжить (*Simulation/Continue*).

### Завершение работы

Для завершения работы необходимо сохранить модель в файле, закрыть окно модели, окно обозревателя библиотек, а также основное окно пакета **MATLAB**.

## Лабораторная работа №1. ОЗНАКОМЛЕНИЕ С ОСНОВНЫМИ БЛОКАМИ СИСТЕМЫ SIMULINK

**Цель работы** – освоить основные приемы работы с инструментарием системы SIMULINK.

**Задание:** составить структурные схемы простейших S-моделей и реализовать их средствами системы SIMULINK с целью проверки правильности их функционирования.

### Подготовка к выполнению работы

Для подготовки к лабораторной работе предварительно необходимо выполнить следующее:

1. Изучить основы теории и ответить на контрольные вопросы.
2. Составить структурные схемы для изучения работы
  - генератора импульсов;
  - генератора линейно нарастающего напряжения;
  - генератора ступенчатого сигнала;
  - дифференцирующего звена;
  - интегратора;
  - генератора пилообразного напряжения

**Указание:** для формирования пилообразного напряжения используйте интегратор со сбросом при достижении выходным сигналом заданного уровня.

### Указания к выполнению работы

1. Проведите эксперименты с перечисленными выше блоками, изменяя их параметры и выводя на индикатор такие пары сигналов, которые позволяют доказать правильность работы блоков.
2. Рассмотрите работу других блоков, входящих в библиотеку (по вашему выбору), и проведите с ними эксперименты, изменяя параметры.

### Отчет по лабораторной работе

Отчет по лабораторной работе должен содержать:

- структурные схемы с указанием конкретных значений параметров блоков, при которых проводились эксперименты;
- примеры входных и выходных сигналов;
- выводы.

## Лабораторная работа №2. ИЗУЧЕНИЕ МЕТОДИКИ ПОСТРОЕНИЯ S-МОДЕЛЕЙ

**Цель работы** – реализовать на практике основные этапы построения S-моделей.

**Задание:** формализовать содержательное описание системы, составить ее структурную схему и представить ее в виде функциональной блок-схемы средствами **SIMULINK** с целью экспериментального определения параметров оптимального управления.

### Подготовка к выполнению работы

Для подготовки к лабораторной работе предварительно необходимо выполнить следующее:



1. Изучить основы теории и ответить на контрольные вопросы.
2. По содержательному описанию системы составить структурную схему для изучения ее работы;
3. Составить структурную схему S-модели, используя блоки системы **SIMULINK**;
4. Предложить план эксперимента с S-моделью.

### Содержательное описание системы

Объект исследования – космический корабль, который должен причалить к космической станции (рис.2.38).

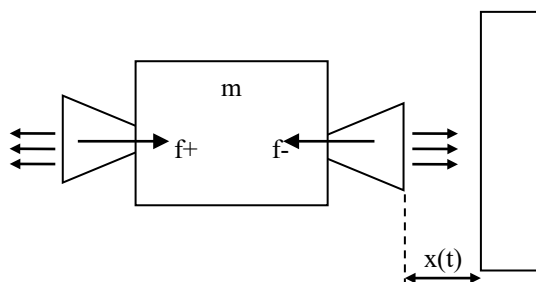


Рис.2.38

Космонавт или автомат может включать двигатели корабля: левый – чтобы разогнать его силой  $f_+$  для сближения со станцией, правый – чтобы затормозить его силой  $f_-$  при подходе к ней.

Чтобы благополучно причалить, надо детально изучить, как сила тяги  $f(t)$  преобразуется в перемещение корабля относительно станции  $x(t)$ .

Перемещение зависит от скорости корабля  $v(t)$ , а последняя – от его ускорения  $a(t)$ . Преобразование тяги в ускорение определяется параметром  $m$  – массой корабля.

*Указание:* при формализации содержательного описания использовать следующие соотношения:

$$a(t) = \frac{1}{m} f(t) \text{ – закон Ньютона,}$$

$$v(t) = v_0 + \int_0^t a(\tau) d\tau \text{ – накопление скорости,}$$

$$x(t) = x_0 + \int_0^t v(\tau) d\tau \text{ – накопление перемещения.}$$

Начальные условия:  $v(0) = v_0, x(0) = x_0$ .

### Указания к выполнению работы

1. Реализуйте составленную структурную схему S-модели, используя блоки системы **SIMULINK**.
2. Проведите эксперименты с S-моделью, изменяя параметры блоков так, чтобы в ходе экспериментов найти их оптимальное сочетание.

### Отчет по лабораторной работе

Отчет должен содержать

- структурные схемы с указанием конкретных значений параметров блоков, при которых проводились эксперименты;
- примеры входных и выходных сигналов;
- выводы.

### Лабораторная работа №3. ИССЛЕДОВАНИЕ ДИНАМИКИ СИСТЕМ С ПОМОЩЬЮ S-МОДЕЛЕЙ

**Цель работы** – освоить методику исследования динамических систем с помощью S-моделей.

**Задание:** формализовать содержательное описание системы, составить ее структурную схему и реализовать ее средствами системы **SIMULINK** с целью экспериментального определения параметров оптимального управления.

### Подготовка к выполнению работы

Для подготовки к лабораторной работе предварительно необходимо выполнить следующее:

1. Изучить основы теории и ответить на контрольные вопросы.
2. По содержательному описанию системы составить структурную схему для изучения ее работы, обратив особое внимание на обратную связь;
3. Составить структурную схему S-модели, используя блоки системы **SIMULINK**;
4. Предложить план эксперимента с S-моделью.

### Содержательное описание системы

**Вариант 1.** Объект исследования – автомобиль, который должен остановиться перед препятствием. При выключении двигателя автомобиль останавливается под действием силы сопротивления  $f_c(t)$ , которая зависит от скорости движения:  $f_c(t) = F[v(t)]$ .

Чтобы благополучно остановиться, нужно, чтобы в момент касания препятствия скорость автомобиля была бы равна нулю.



Перемещение  $x(t)$  зависит от скорости автомобиля  $v(t)$ , а последняя – от его ускорения  $a(t)$ . Преобразование силы в ускорение определяется параметром  $m$  – массой автомобиля.

Начальные условия:  $v(0) = v_0$ ,  $x(0) = x_0$

Определить момент выключения двигателя.

Указание:

1. Зависимость  $f_c(t) = F[v(t)]$  задается преподавателем,
2. Можно воспользоваться уравнениями лабораторной работы №2.

**Вариант 2.** Объект исследования – корабль, который должен остановиться перед причальной стенкой. Корабль можно затормозить с помощью реверса двигателя, т.е. включая задний ход при подходе к стенке, причем скорость заднего хода выбирается в зависимости от скорости движения и расстояния до причальной стенки.

При движении корабля возникает сила сопротивления, зависящая от скорости его движения:  $f_c(t) = F[v(t)]$ . Чтобы благополучно остановиться, нужно, чтобы в момент касания стенки скорость корабля была бы равна нулю.

Перемещение  $x(t)$  зависит от скорости корабля  $v(t)$ , а последняя – от его ускорения  $a(t)$ . Преобразование тяги в ускорение определяется параметром  $m$  – массой корабля.

Начальные условия:  $v(0) = v_0$ ,  $x(0) = x_0$

Определить момент реверса двигателя и закон изменения скорости так, чтобы минимизировать время причаливания.

Указание:

1. Зависимость  $f_c(t) = F[v(t)]$  задается преподавателем,
2. Можно воспользоваться уравнениями лабораторной работы №2.

**Вариант 3.** Объект исследования – автомобиль, который должен остановиться перед препятствием. Водитель может затормозить его силой  $f_r(t)$  при подходе к препятствию, причем он выбирает силу торможения в зависимости от скорости и расстояния до препятствия.

Чтобы благополучно остановиться, нужно, чтобы в момент касания препятствия скорость автомобиля была бы равна нулю.

Перемещение  $x(t)$  зависит от скорости автомобиля  $v(t)$ , а последняя – от его ускорения  $a(t)$ . Преобразование силы в ускорение определяется параметром  $m$  – массой автомобиля.

Начальные условия:  $v(0) = v_0$ ,  $x(0) = x_0$

Определить момент начала торможения и закон изменения силы торможения так, чтобы минимизировать время остановки.

Указание: воспользоваться уравнениями лабораторной работы №2.

**Вариант 5.** Парашютист прыгает с парашютом с высоты  $h_0$ , выбирая момент раскрытия парашюта таким образом, чтобы минимизировать время снижения, но так, чтобы скорость в момент приземления не превышала допустимой величины  $v^*$ .

Указание:

1. Причиной движения является сила гравитации  $P = mg$ , где  $g$  – ускорение свободного падения,  $m$  – масса парашютиста.

2) Сила аэродинамического сопротивления при раскрытии парашюта  $f_c = \alpha v^2$ ;

3) Начальные условия:  $v(0) = 0$ ,  $h(0) = h_0$

**Вариант 6.** Изучить процесс установления напряжения в  $RC$ -цепочке (рис.2.39). Падение напряжения на сопротивлении  $R$  равно разности между входным  $u_1$  и выходным  $u_2$  напряжениями ( $u_2 = u_c$  – напряжение на конденсаторе  $C$ ),  $u_R = u_1 - u_2$ ; напряжение  $u_R$  определяет ток  $i = u_R/R$ ; заряд конденсатора:

$$q(t) = q_0 + \int_0^t i(\tau) d\tau$$

Напряжение на конденсаторе определяется его зарядом  $u_c = q/C$ ; это напряжение является выходом  $u_2(t) = u_c(t)$  и оно вычитается из  $u_1(t)$  при определении напряжения  $u_R(t) = u_1(t) - u_2(t)$ . Ток  $i(t)$  зависит от входного напряжения  $u_1(t)$  и от напряжения  $u_2(t)$ , до которого зарядился конденсатор  $C$  током  $i(t)$ .

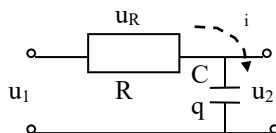


Рис.2.39

**Вариант 7.** Изучить процесс установления напряжения в  $RL$ -цепочке (рис.2.40). Напряжение на индуктивности  $u_L(t) = u_1(t) - u_2(t)$  определяет потокосцепление

$$\Psi(t) = \Psi_0 + \int_0^t u_L(\tau) d\tau$$

а потокосцепление  $\Psi(t)$  – ток  $i(t) = \Psi(t)/L$ . Ток  $i(t)$  создает падение напряжения на сопротивлении  $R$ :  $u_R = Ri$ . Это напряжение является выходным  $u_2(t) = u_R(t)$  и оно же вычитается из входного при определении  $u_L(t) = u_1(t) - u_2(t)$ .

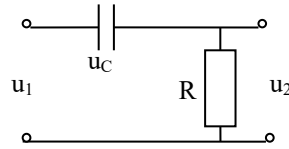


Рис.2.40

**Вариант 8.** Изучить процесс установления напряжения в  $RC$ -цепочке (рис.2.41.). Падение напряжения на сопротивлении  $R$  является выходным:  $u_R = u_2$ ; оно равно разности между входным напряжением  $u_1$  и  $u_C$  – падением напряжения на конденсаторе  $C$ :  $u_R = u_1 - u_C$ ; напряжение  $u_R$  определяет ток  $i = u_R/R$ ; заряд конденсатора

$$q(t) = q_0 + \int_0^t i(\tau) d\tau.$$

Напряжение на конденсаторе определяется его зарядом  $u_C = q/C$ ; это напряжение вычитается из  $u_1(t)$  при определении напряжения  $u_R(t) = u_1(t) - u_2(t)$ . Ток  $i(t)$  зависит от входного напряжения  $u_1(t)$  и от напряжения  $u_C(t)$ , до которого зарядился конденсатор  $C$  током  $i(t)$ .

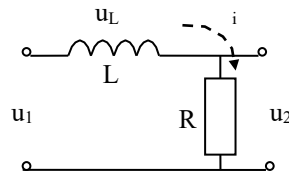


Рис.2.41

**Вариант 9.** Изучить процесс установления напряжения в  $RL$ -цепочке (рис.2.42). Напряжение на индуктивности  $u_L(t) = u_1(t) - u_R(t)$  является выходным  $u_2(t)$  и оно же определяет потокосцепление

$$\Psi(t) = \Psi_0 + \int_0^t u_L(\tau) d\tau,$$

а потокосцепление  $\Psi(t)$  – ток  $i(t) = \Psi(t)/L$ . Ток  $i(t)$  создает падение напряжения на сопротивлении  $R$ :  $u_R = Ri$ . Это напряжение вычитается из входного при определении  $u_2(t) = u_1(t) - u_R(t)$ .

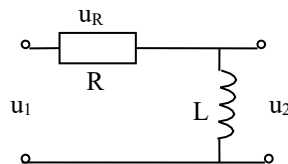


Рис.2.42

**Вариант 10.** В бак ведут две трубы. По одной в бак втекает  $q_1(t)$ , а по другой – вытекает  $q_2(t)$  литров жидкости в секунду через отверстие с проходным сечением  $\sigma(t)$  в днище бака (рис.2.43). Расход  $q_2 = \sigma v$  пропорционален проходному сечению  $\sigma$  отверстия, а скорость истечения жидкости  $v$  зависит от ее уровня  $h$ . Эта зависимость в данном примере описывается формулой Торичелли  $v = \sqrt{2gh}$ . Задача состоит в том, чтобы исследовать изменение уровня воды в баке при различных зависимостях проходного сечения выпускной трубы от времени.

*Указание:* Количество жидкости в каждый момент времени  $t$  зависит от начального значения  $Q_0 = Q(0)$  и от скорости изменения потока  $Q(t) = \Delta q(t) = q_1(t) - q_2(t)$ :

$$Q(t) = Q_0 + \int_0^t \Delta q(\tau) d\tau$$

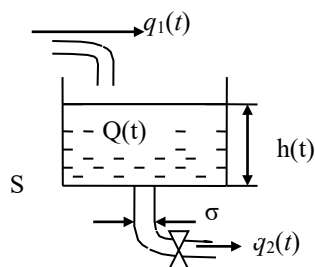


Рис.2.43

### Указания к выполнению работы

1. Реализуйте составленную структурную схему S-модели, используя блоки системы SIMULINK.
2. Проведите эксперименты с S-моделью, изменяя параметры ее блоков.

### Отчет по лабораторной работе

Отчет по лабораторной работе должен содержать:

- структурные схемы с указанием конкретных значений параметров блоков, при которых проводились эксперименты;
- примеры процессов в различных точках схемы.

### Лабораторная работа №4. ИССЛЕДОВАНИЕ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА С ПОМОЩЬЮ S-МОДЕЛИ МЕТОДОМ ПОНИЖЕНИЯ ПОРЯДКА ПРОИЗВОДНОЙ

**Цель работы:** Получение навыков исследования линейных динамических моделей с использованием пакета прикладных программ **SIMULINK** системы **MATLAB 6**. Анализ чувствительности модели с помощью варьирования параметров модели.

### Постановка задачи

Большинство объектов и процессов управления может быть описано линейными дифференциальными уравнениями с постоянными коэффициентами. На примере процесса, описываемого дифференциальным уравнением второго порядка, рассмотрим все этапы моделирования и анализа свойств объекта.

Рассмотрим объект, у которого входное воздействие - потребление пара  $x(t)$  кг/ч, а выходная величина - температура  $y(t)$  °C.

Допустим, что в результате экспериментов по исследованию динамики объекта получено следующее дифференциальное уравнение, описывающее изменение  $y(t)$  в зависимости от времени:

$$256 \frac{d^2 y(t)}{dt^2} + 118 \frac{dy(t)}{dt} + 7y(t) = 5x(t) \quad (2.1)$$

Возмущающая функция  $x(t)$  должна быть известной функцией времени.

Должны быть заданы также начальные условия.

Предположим, что в начальный момент времени температура объекта была равна

$$y|_{t=0} = 128^\circ \text{C}. \quad (2.2)$$

и происходило остывание со скоростью

$$\left. \frac{dy}{dt} \right|_{t=0} = -1,6 \left[ \frac{^\circ \text{C}}{\text{мин}} \right]. \quad (2.3)$$

Требуется определить на каком уровне следует поддерживать потребление пара, чтобы температура на выходе объекта оставалась неизменной.

На ЭВМ необходимо решить дифференциальное уравнение (2.1) при начальных условиях (2.2) и (2.3).

### Метод понижения порядка производной

Решить уравнение (2.1) с использованием **SIMULINK** можно, используя метод понижения порядка производной.

Решить дифференциальное уравнение - означает получить функцию  $y(t)$ , меняющуюся во времени. Составление функциональной блок-схемы, применяя метод понижения порядка производной, можно свести к пяти этапам.

**Этап 1.** Разрешим дифференциальное уравнение (2.1) относительно высшей производной:

$$\frac{d^2 y}{dt^2} = -0,46 \frac{dy}{dt} - 0,027 y(t) + 0,02 x(t). \quad (2.4)$$

**Этап 2.** Понизить порядок производной - это значит проинтегрировать  $\frac{d^2 y}{dt^2}$ , затем  $\frac{dy}{dt}$  и в результате получить  $y(t)$ . Сделать это можно, смоделировав схему, включающую последовательно блоки - интеграторы (рис.2.44).

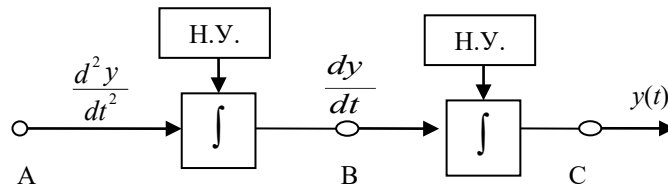


Рис. 2.44

Предположив, что в точке А  $\frac{d^2y}{dt^2}$  известно в любой момент времени. С помощью интегрирующего звена и с учетом начальных условий получим в точке В значение производной  $\frac{dy}{dt}$ . Затем, с помощью еще одного интегратора, в точке С получим значение искомой функции  $y(t)$ .

**Этап 3.** Рассмотрим теперь правую часть уравнения (2.4). Она представляет собой сумму трех функций времени, взятых с постоянными коэффициентами:

$$\frac{dy}{dt}, y(t), x(t).$$

Построение части функциональной схемы, соответствующей операции суммирования, производится следующим образом.

Будем считать, что известному значению функции  $y(t)$  соответствует какая-либо точка  $C_1$ , известному значению  $\frac{dy}{dt}$  соответствует точка  $B_1$  (рис. 2.45). Теперь, просуммировав их с коэффициентами, соответствующими правой части (2.4), получим вторую производную  $\frac{d^2y}{dt^2}$ . Таким образом, на выходе сумматора, в точке  $A_1$ , будет величина  $\frac{d^2y}{dt^2}$ , известная в любой момент времени.

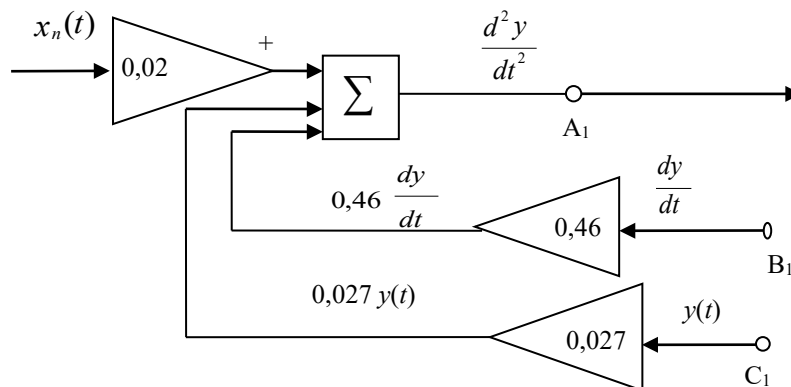


Рис. 2.45

**Этап 4.** Равенство (2.4), которое происходит из физической сущности моделируемого процесса, выполняется в любой момент времени  $t$ . Реализовать это требование легко – достаточно замкнуть схемы, показанные на рис. 2.44 и 2.45. При этом сольются точки А и  $A_1$ , В и  $B_1$ , С и  $C_1$  (рис. 2.46).

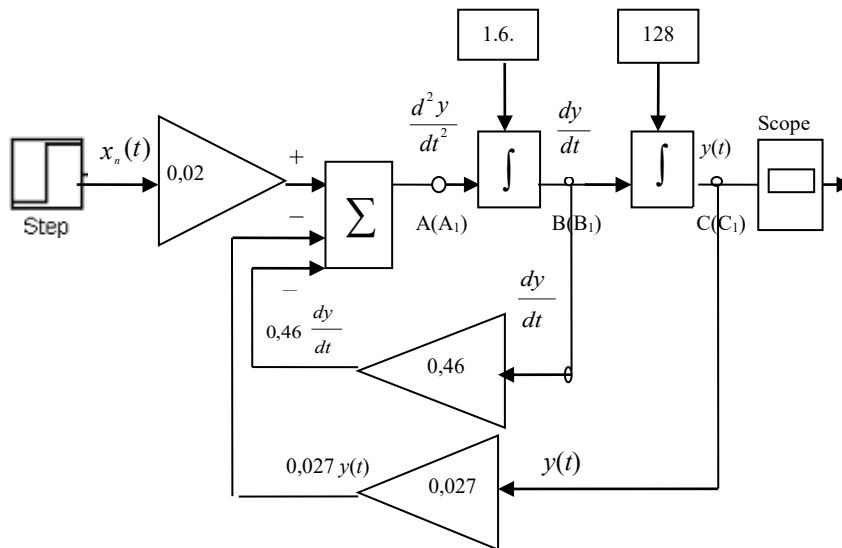


Рис. 2.46

**Этап 5.** Чтобы завершить моделирование схемы (рис.2.46) следует установить начальные условия (2.2) и (2.3), которые определяют единственность решения дифференциального уравнения и присоединить блок ступенчатого входного сигнала, который моделирует входное воздействие  $x_n(t)$ .

#### 2.4. Методика выполнения работы

1. Чтобы собрать схему, сначала необходимо «перетащить» все нужные звенья из соответствующих наборов на пространство **Untitled**, предназначенное для составления блок-схемы модели.

В соответствии со схемой на рис. 2.46 нам необходимо иметь:

- источник скачкообразного возмущения для  $x_n(t)$  (**Step**);
- сумматор (**Sum**);
- три усилителя (**Gain**);
- два интегратора (**Integrator**);
- два источника постоянного сигнала для задания начальных условий (**Constant**);
- осциллограф для наблюдения результатов моделирования (**Scope**).

2. Теперь необходимо связать между собой эти звенья, соединив их в соответствии со схемой на рис. 2.46.

Система создаст связь между элементами и обозначит ее стрелкой по направлению сигнала. Некоторые предпочитают соединять звенья по типу «вход следующего с выходом предыдущего», что иногда бывает более предпочтительным.

3. Чтобы задать начальные условия от внешнего источника, нужно выполнить следующее:

- открыть меню свойств интегратора;
- выбрать команду Источник начальных условий (Initial condition source), указать Внешний (External), после этого на иконке интегратора появляется еще один вход (нижний), на который и подают сигнал задания начального условия; в данном примере это и будет источник постоянного сигнала;
- меню свойств закрывают нажатием кнопки Применить (Apply) и затем кнопки Закрыть (Close).

4. Величину постоянного сигнала задают в свойствах источника сигнала (строка *Constant value*).

5. Величину коэффициента усиления задают в свойствах усилителя (**Gain**).

6. Величину и время возмущения задают в свойствах источника скачкообразного возмущения. Там определяют момент поступления возмущения (строка *Step time*), в нашем случае 0. Значение сигнала до и после возмущения (строки *Initial value u Final value*). Зададим значение сигнала до возмущения равным 0, а после значение сигнала после возмущения будем варьировать, чтобы получить ответ на вопрос, сформулированный в постановке задачи.

7. Запуск модели в работу, т.е. начало собственно процесса моделирования, производится командой *Start* меню **Simulation**.

8. Двойной щелчок на иконке осциллографа (**Scope**) открывает его окно, в котором можно наблюдать изменение выходных переменных во времени. Для удобства наблюдения рекомендуется разбить весь экран монитора по горизонтали на два поля: в верхнем развернуть осциллограф, а в нижнем оставить модель (или наоборот).

9. Настройка длительности процесса моделирования производится в меню **Simulation** окна построения модели. В этом меню в закладке *Решатель (Solver)* можно выставить время начала процесса моделирования (поле *Start*) и его конца (поле *Stop time*).

10. Настройка параметров осциллографа осуществляется так, как это описано ранее. Разметка оси абсцисс (времени) устанавливается автоматически в соответствии с величинами *Start time* и *Stop time* (если употребить слово *auto* в поле *Time range*).

11. Работу модели можно приостановить кнопкой **Pause**, в которую превращается кнопка запуска моделирования после ее нажатия. Рядом, справа, расположена кнопка возврата в начальное состояние.

12. Simulink автоматически преобразует блок-схему модели в систему дифференциальных уравнений. Выбор метода интегрирования этой системы можно произвести через меню **Simulation**. На панели *Solver options* можно также выставить шаг интегрирования (левое поле) и тип решателя (правое поле). По умолчанию выставлен автоподбор шага (*Variable step*) и интегрирование методом Дорманда – Принца (*ode45 Dormand – Prince*). Там же можно выставить допустимые *Абсолютную погрешность* и *Относительную погрешность* интегрирования поля *Absolute tolerance* и *Relative tolerance* соответственно. Рекомендуется оставить эти параметры без изменений.

13. Многолучевой осциллограф получается добавлением элемента *Мультиплексор (Mux)* из набора *Соединители (Connections)*. В этом случае можно наблюдать в одних и тех же координатах одновременно несколько сигналов, например  $y(t)$ ,  $\frac{dy}{dt}$ ,  $\frac{d^2y}{dt^2}$ ,  $x_n(t)$ . Двойным щелчком мыши на мультиплексоре открывается меню его свойств, где можно уставить количество входов.

14. Запись кривых – результатов моделирования (для долговременного хранения и дальнейшего использования) – осуществляется подсоединением на место осциллографа, или параллельно с ним, элемента *В файл (To file)*, расположенного в наборе *Приемники (Sinks)*.

В свойствах элемента *В файл* необходимо задать имя файла с расширением **.mat** (например, **outdata1.mat**), в который будет производиться запись данных (рис. 2.47).

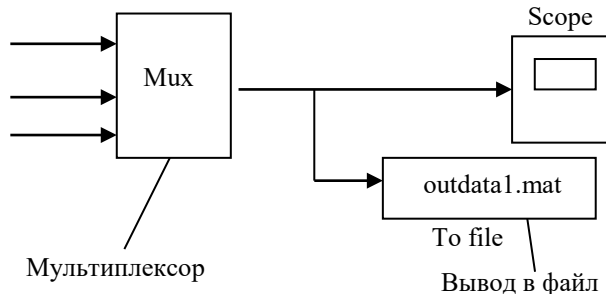


Рис. 2.47

15. Сравнение вида новой кривой (или кривых) – результата моделирования – с некоторыми «реперными кривыми», полученными ранее, легко проводить с помощью элемента *Из файла (From File)*. Для этого из набора *Источники (Sources)* надо поместить на поле блок-схемы моделирования элемент *Из файла*. Это обеспечит вывод на осциллограф «реперных кривых» вместе с новыми кривыми моделирования (рис. 2.48).

В свойствах элемента *Из файла* необходимо указать конкретное имя файла, в котором хранятся «реперные кривые». Естественно также, что файл под этим именем ранее выступал в качестве элемента *В файл*, когда в него записывались «реперные кривые».

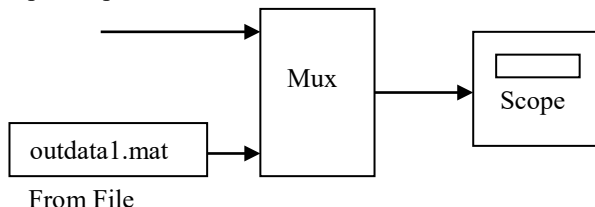


Рис. 2.48. Ввод данных из файла

16. Для сохранения блок-схемы нужно выполнить команду *Save* из меню *File*. В окне *Сохранение* будет предложено ввести имя файла. При последующих сохранениях данные в файле будут обновляться. По умолчанию файлы сохраняются в директории *.../MatLab/Bin*, но желательно создать свою поддиректорию в директории **MATLAB**. Файлу присваивается расширение **.mdl**.

17. Для вызова с жесткого диска ранее сохраненной модели нужно активизировать функцию *Open* из меню *File*.

18. Для того чтобы продолжить работу с моделью в другое время, необходимо:

сохранить *Рабочее поле (Workspace)* в файл на жестком диске (**mat**-файл);

сохранить блок-схему модели в файл на жестком диске (**mdl**-файл);

сохранить на жестком диске файл-функции (М-файл), если их создавали в окне *MatLab Editor/Debugger*.

19. Для исследования процесса и решения поставленной модели, необходимо задавать различные значения входного воздействия  $x(t)$  и каждый раз просматривать результаты в окне осциллографа. На рис.2.49 показаны результаты расчета  $y(t)$  (температура на выходе) при различных  $x(t)$  (расход пара на входе). Можно сделать вывод, что наиболее оптимальным, для поддержания температуры на выходе равной  $128^{\circ}\text{C}$ , является расход пара  $x(t) = 174$

X=0	X=250
-----	-------

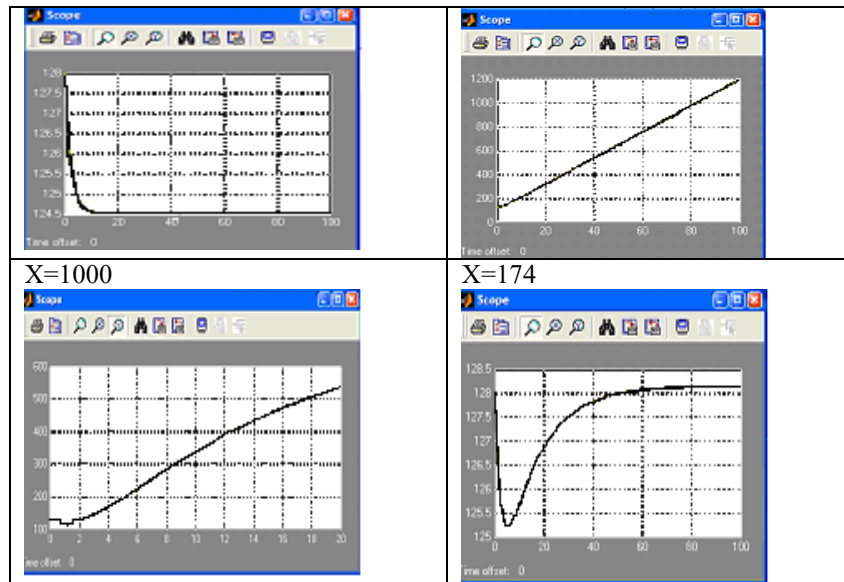


Рис.2.49

## 2.5. Методы контроля правильности набора схем и установки коэффициентов

Первым и наиболее общим показателем правильности составления структурной схемы и установки коэффициентов является изменение всех переменных в допустимых границах. Если после нажатия кнопки *Пуск* некоторые переменные монотонно возрастают и «уходят в бесконечность», то причиной этого может быть или наличие интегратора, не охваченного обратной связью (неправильно сделаны соединения), или большой коэффициент усиления в статике (неправильно выбраны коэффициенты).

Если решение сильно колебательное, то это может характеризовать отсутствие обратных связей по производным (например, при решении уравнения второго порядка).

Если переменные не выходят за предусмотренные пределы, то оценить точность моделирования позволяют следующие количественные методы.

1. Проверка правильности выставления коэффициентов по характеру переходного процесса (решения) для дифференциальных уравнений второго порядка. На вход собранной схемы подают скачкообразную возмущающую функцию, например  $x_n = 1$ , и наблюдают характер решения (переходного процесса).

Если объект второго порядка, то, в зависимости от корней характеристического уравнения, могут быть получены следующие виды переходных процессов (рис. 2.7):

- а) оба корня действительные отрицательные – переходный процесс аperiodический;
- б) оба корня комплексные сопряженные – переходный процесс колебательный;
- в) оба корня чисто мнимые (в уравнение не входит первая производная) – переходный процесс представляет собой незатухающие колебания.

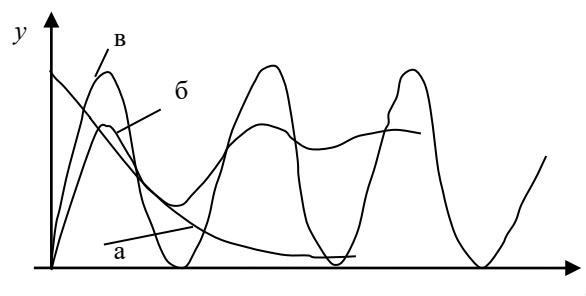


Рис. 2.7. Переходный процесс в системе

Следовательно, для такой проверки предварительно нужно вычислить корни характеристического уравнения. Соответствующие команды можно задать в *Командном Окне (MatLab Command Window)*.

2. Проверка правильности времени переходного процесса. Производится при исследовании поведения объекта на скачкообразное возмущение: переходный процесс должен закончиться за время, чуть большее  $\tau = 3T$ , где  $T$  – постоянная времени при первой производной  $\frac{dy}{dt}$ , полученная после деления всех членов дифференциального уравнения на коэффициент при  $y(t)$ . Для уравнения (2.1)

$$T = \frac{118}{7} \approx 17 \quad (2.5)$$

и весь переходный процесс закончится за время  $\tau = 3T = 3 \cdot 17 \approx 51$



3. Проверка правильности выставления коэффициентов передачи по коэффициенту усиления в статике. На вход собранной схемы подают скачкообразное возмущение, например  $x_n = \text{const} = 1$ . После окончания переходного процесса измеряют численное значение выходной величины  $y(\infty)$ . Это удобно сделать, подсоединив параллельно осциллографу элемент *Дисплей (Display)* из набора *Приемники (Sinks)*.

Пусть для модели, описываемой уравнением (2.1) получилось  $y(\infty) = 0,7$ . Следовательно, полученный коэффициент усиления в статике будет равен

$$K = \frac{y(\infty)}{x(\infty)} = \frac{0,7}{1} = 0,7.$$

Сравниваем коэффициент в статике с действительным, полученным по уравнению (2.1):

$$7y = 5x, \quad K = \frac{y}{x_n} = \frac{5}{7} \approx 0.7$$

Коэффициенты усиления в статике должны совпадать с необходимой точностью. Если нужна более высокая точность, то следует вновь выставить коэффициенты, взяв большее количество знаков после запятой.

## 2.6. Задание к лабораторной работе

1. Составить структурную блок-схему S-модели и выполнить моделирование процесса в среде **SIMULINK**, описываемого уравнением второго порядка с постоянными коэффициентами:

$$a_0 \frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_2 x = b_0 x(t).$$

Начальные условия:

$$\left. \frac{dy}{dt} \right|_{t=0} = y'(0), \quad Y|_{t=0} = y(0).$$

Ограничения:

$$x_{\max} \geq x \geq x_{\min}.$$

Исходные данные возьмите из табл. 2.1.

Рассмотрите поведение системы для  $x(t) = 0$  при следующих комбинациях начальных условий: оба начальных условия нулевые (из таблицы); только первое – нулевое; только второе – нулевое; оба начальных условия ненулевые.

2. Рассмотрите поведение системы с  $x(t)$  в виде функции Хевисайда от уровня  $x_{\min}$  до величины  $x_{\max}$  в момент времени  $t = 0$ . Пронаблюдайте поведение системы на осциллографе. Определите, при каком входном воздействии удастся поддерживать процесс на уровне начального  $y(t)$ .

3. Проверьте правильность набора структурной схемы.

4. Определите корни характеристического уравнения, предварительно получив выражения для передаточной функции и характеристического уравнения (для этого повторите соответствующий материал из курса «Основы теории управления»).

## 2.7. Отчет по лабораторной работе

В отчете должны содержаться:

1. Блок-схема моделирования звена второго порядка с коэффициентами, соответствующими вашему варианту. Осциллограммы для различных вариантов начальных условий и возмущающих функций.
2. Блоки в составе блок-схем, дающие возможность организовать длительное хранение в файле всех ранее полученных кривых, возможность просмотра кривых, полученных в результате моделирования и записанных на длительное хранение в файле.
3. Контроль правильности набора схем и установки коэффициентов.
4. Анализ влияния возмущающих воздействий на поведение выходной величины.
5. Определение времени переходного процесса.

Таблица 2.1

№ варианта	$a_0$	$a_1$	$a_2$	$b_0$	$y'(0)$	$y(0)$	$x_{\min}$	$x_{\max}$
1	$6 \frac{\text{мин}^2}{^\circ\text{C}}$	$2,1 \frac{\text{мин}}{^\circ\text{C}}$	$1,1 \frac{1}{^\circ\text{C}}$	$0,09 \frac{\text{мин}}{\text{об}}$	$8 \frac{^\circ\text{C}}{\text{мин}}$	$40^\circ\text{C}$	$400 \frac{\text{об}}{\text{мин}}$	$4400 \frac{\text{об}}{\text{мин}}$
2	$30 \frac{\text{мин}^2}{^\circ\text{C}}$	$150 \frac{\text{мин}}{^\circ\text{C}}$	$3 \frac{1}{^\circ\text{C}}$	$0,2 \frac{1}{\text{атм}}$	$38 \frac{^\circ\text{C}}{\text{мин}}$	$4,2^\circ\text{C}$	38 атм	79 атм
3	$0,5 \frac{\text{сек}^2}{^\circ\text{C}}$	$0,8 \frac{\text{сек}}{^\circ\text{C}}$	$0,1 \frac{1}{^\circ\text{C}}$	$0,08 \frac{1}{\text{см}}$	$0,1 \frac{^\circ\text{C}}{\text{сек}}$	$3,0^\circ\text{C}$	3,4 см	8,2 см

4	$120 \frac{\text{с}^2}{\%}$	$866 \frac{\text{с}^2}{\%}$	$14 \frac{1}{\%}$	$0,2 \frac{1}{\text{мм}}$	$-4,5 \frac{\%}{\text{мин}}$	10%	100 мм	6250 мм
5	$400 \frac{\text{мин}}{\%}$	$320 \frac{\text{мин}}{\%}$	$120 \frac{1}{\%}$	$93 \frac{1}{\text{атм}}$	$-2 \frac{\%}{\text{мин}}$	40 %	0,3 атм	8,3 атм
6	$62 \frac{\text{мин}}{\text{кг/ч}}$	$51 \frac{\text{мин}}{\text{кг/ч}}$	$51 \frac{1}{\text{кг/ч}}$	$29 \frac{1}{\text{°C}}$	$-1,6 \frac{\text{кг/ч}}{\text{мин}}$	$100 \frac{\text{кг}}{\text{ч}}$	20 °C	81 °C
7	$4600 \frac{\text{мин}}{\text{°C}}$	$400 \frac{\text{мин}}{\text{°C}}$	$2,2 \frac{1}{\text{°C}}$	$18,6 \frac{1}{\text{об/с}}$	$3,2 \frac{\text{°C}}{\text{мин}}$	212 °C	$12 \frac{\text{об}}{\text{с}}$	$140 \frac{\text{об}}{\text{с}}$
8	$260 \frac{\text{мин}}{\text{кг/ч}}$	$40 \frac{\text{мин}}{\text{кг/ч}}$	$1,2 \frac{1}{\text{кг/ч}}$	$8,6 \frac{1}{\text{атм}}$	$4 \frac{\text{кг/ч}}{\text{мин}}$	$156 \frac{\text{кг}}{\text{ч}}$	11 атм	42 атм
9	$320 \frac{\text{мин}}{\text{кг/ч}}$	$40 \frac{\text{мин}}{\text{кг/ч}}$	$1,2 \frac{1}{\text{кг/ч}}$	$8,6 \frac{1}{\text{атм}}$	$24 \frac{\text{кг/ч}}{\text{мин}}$	$71 \frac{\text{кг}}{\text{ч}}$	6 атм	23 атм
10	$700 \frac{\text{мин}}{\text{мм}}$	$560 \frac{\text{мин}}{\text{мм}}$	$46,0 \frac{1}{\text{мм}}$	$332 \frac{1}{\text{°C}}$	$-4 \frac{\text{мм}}{\text{мин}}$	300 мм	32 °C	63 °C
11	$8,4 \frac{\text{с}^2}{\%}$	$0,4 \frac{\text{с}}{\%}$	$1,3 \frac{1}{\%}$	$0,068 \frac{1}{\text{мм}}$	$12,4 \frac{\text{°C}}{\text{сек}}$	3,1 °C	30 мм	102 мм
12	$320 \frac{\text{сек}}{\text{°C}}$	$480 \frac{\text{сек}}{\text{°C}}$	$115 \frac{1}{\text{°C}}$	$84 \frac{1}{\%}$	$0,2 \frac{\text{°C}}{\text{сек}}$	20 °C	20 %	82 %
13	$164 \frac{\text{М}}{\text{К}}$	$81 \frac{\text{МИН}}{\text{кг/ч}}$	$13,2 \frac{1}{\text{К}}$	$19 \frac{1}{\%}$	$-10,6 \frac{\text{К}}{\text{М}}$	$180 \frac{\text{К}}{\text{ч}}$	15 °C	105 °C

### Контрольные вопросы

1. Для решения каких задач предназначен ППП **SIMULINK MATLAB 6.x**?
2. Файлы каких форматов генерирует **MATLAB 6.x** ?
3. Какие символы нельзя использовать в имени файла при его сохранении?
4. Что такое функциональная блок-схема?
5. Какие существуют библиотеки инструментов в **SIMULINK**?
6. Какие вы знаете источники сигналов?
7. Перечислите приемники сигналов.
8. Какие математические операции доступны в **SIMULINK**?
9. Можно ли увеличить размер блоков в структурной схеме?
10. Какие способы изображения результатов расчета используются в **SIMULINK**?
11. Можно ли в модели, изображенной на рис. 2.46 в качестве источника возмущения использовать пилообразный сигнал?
12. Сколько интеграторов понадобится для реализации модели, использующей дифференциальное уравнение четвертого порядка?
13. Можно ли вместо источника ступенчатого возмущения использовать математическую модель?
14. Что следует изменить в функциональной схеме на рис. 2.46, чтобы выходная величина увеличилась в два раза?
15. Назовите назначение интегрирующего и дифференцирующего блоков.
16. В каких операциях в моделях **SIMULINK** можно использовать данные из файла в виде матрицы или вектора?
17. Как сохранить результаты исследования модели, если экспериментов было несколько?
18. Что подразумевается под понятием «модель верхнего уровня»?
19. Какими инструментами в **SIMULINK** можно приостанавливать процесс вычисления?
20. Можно ли в **SIMULINK** получать результаты промежуточных расчетов?
21. Перечислите достоинства и недостатки в **SIMULINK**.

### 2.9. Литература

1. Данилов, А. И. Компьютерный практикум по курсу «Теория управления». Simulink – моделирование в среде Matlab : учеб. пособие / А. И. Данилов ; под ред. А. Э. Софиева. – М. : МГУИЭ, 2002.
2. Дьяконов, В. Simulink 4. Специальный справочник / В. Дьяконов. – СПб. : Питер, 2002.