

## *Моделирование и анализ систем автоматического управления в программе VisSim.*

**Цель работы** – изучить методы исследования математических моделей. Освоить различные методы решения линейных неоднородных дифференциальных уравнений. Научиться преобразовывать дифференциальное уравнение в передаточную функцию, а также строить амплитудно–фазовую (АФХ), амплитудно–частотную (АЧХ), фазово–частотную (ФЧХ) характеристики.

### **Основные понятия**

#### *1. Основные сведения по интерфейсу пакета VisSim*

Программа имеет стандартный для Windows – программ интерфейс с ориентацией на работу с мышью. Поэтому для более эффективной работы с ней необходимо обладать начальными навыками работы в системе Windows.

Окно программы в полоске меню содержит разбитый на логические группы список всех действий, которые может выполнить программа.

Присутствуют следующие группы:

**File** – действия над файлами (основные: задать новый файл для работы – **New**, открыть с диска существующий файл с проектом – **Open**, сохранить проект в файле на диске – **Save**, распечатать изображение рабочей области – **Print...** и т.п.);

**Edit** – действия над объектами, расположенными в рабочей области (основные: вернуться на шаг назад – **Undo**, скопировать выделенные на рабочей области объекты в карман – **Copy**, вставить находящиеся в кармане объекты на то место рабочей области, где расположен указатель мыши – **Paste** и т.п.);

**Simulate** – действия, задающие режим моделирования (расчета характеристик объекта заданного в рабочей области, основные: начать моделирование – **Go**, остановить моделирование – **Stop**, сделать один шаг в моделировании – **Single step**, продолжить остановленное моделирование – **Continue**, основные параметры процесса моделирования – **Simulation properties** и т.п.);

**Blocks** – объединенные в логические группы блоки (элементы) из которых составляется моделируемое устройство (для САУ наиболее важна группа – **Linear System**. в которой расположен блок с названием **transferFunction** – передаточная функция);

**Analyze** – виды анализа. которые можно проводить для заданной модели устройства (для САУ наиболее важны: **Nyquist Response** – определение амплитудно–фазовой частотной характеристики модели устройства и **Frequency Response** – построение амплитудно–частотной и фазо–частотной характеристик);

**Tools** – средства взаимодействия программы **VisSim** с другими программами (например, действие **Code Gen** – генерирует текст программы на языке C, описывающей заданную модель);

**View** – различные средства позволяющие изменять настройки пакета под конкретного пользователя (увеличить масштаб – **Zoom In**, уменьшить масштаб – **Zoom Out**, определить набор выводимых в панели инструментов – **Tool Bar...** и т.п.);

**Help** – набор файлов помощи по работе в программе **VisSim** (на английском языке).

После стандартной инсталляции программа содержит на панели инструментов следующие группы кнопок:

#### Основная



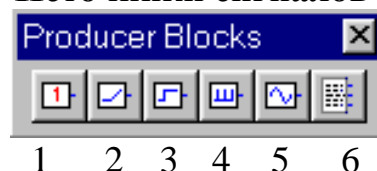
1. Подготовить пакет к работе с новым проектом;
2. Открыть с диска существующий файл с проектом;
3. Сохранить проект в файле на диске;
4. Распечатать изображение рабочей области;
5. Вырезать выделенные на рабочей области объекты и поместить в карман;
6. Скопировать выделенные на рабочей области объекты в карман;
7. Вставить находящиеся в кармане объекты на то место рабочей области, где расположен указатель мыши;
8. Добавит на рабочую область соединитель блоков;
9. Убрать с рабочей области соединитель блоков;
10. Увеличить масштаб объектов на рабочей области;
11. Уменьшить масштаб объектов на рабочей области.

#### Управление моделированием



1. Начать моделирование;
2. Остановить моделирование;
3. Сделать один шаг в моделировании;
4. Продолжить моделирование, после остановки.

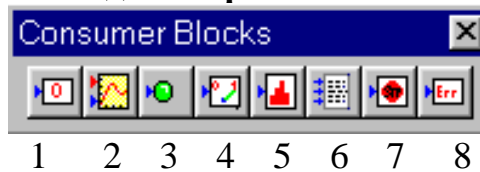
#### Источники сигналов



1. Постоянный сигнал;
2. Линейно нарастающий сигнал;
3. Ступенчатый сигнал;
4. Генератор импульсов;
5. Генератор гармонического сигнала;

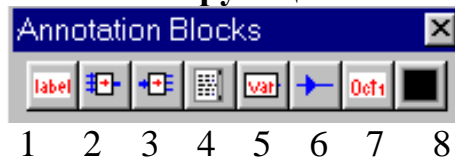
6. Ввод значений сигнала из файла, подготовленного на диске.

### Индикаторы сигналов



1. Дисплей (индицирует величину сигнала на входе в цифровом виде);
2. График (представляет сигнал в виде функции, зависящей от времени);
3. Индикатор (изменяет цвет при выходе уровня сигнала за определенные пределы);
4. Стрелочный прибор (представляет изменение величины сигнала в аналоговом виде);
5. Гистограмма (строит гистограмму входного сигнала);
6. Вывод значений сигнала в файл на диске;
7. Остановка моделирования по условию (если значение сигнала на входе блока больше 2, то моделирование останавливается полностью, если значение сигнала больше 1, то моделирование останавливается и сразу запускается следующее моделирование, во всех остальных случаях моделирование происходит без изменения);
8. Блок индикации ошибки моделирования (когда величина сигнала на входе блока становится отличной от нуля, он становится красным, а моделирование останавливается).

### Комментирующие блоки



1. Блок, позволяющий помещать какую-либо метку в рабочей области;
2. Преобразование нескольких сигналов в один для передачи по одной шине (связи);
3. Преобразование сигнала, передаваемого по шине, в отдельные из которых он был сформирован предыдущим блоком;
4. Блок, позволяющий добавлять на рабочую область текстовые комментарии;
5. Блок, позволяющий дать имя сигналу и завести этот сигнал на другой блок по имени (без использования соединения блоков);
6. Блок для позиционирования соединения на рабочей области;
7. Блок, позволяющий вставить в рабочую область дату;
8. Блок, позволяющий добавлять на рабочую область интерактивные элементы.

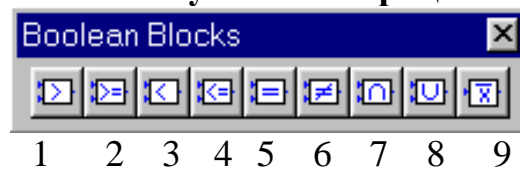
### Блоки арифметических операций



1 2 3 4 5 6 7 8 9 10 11

1. Модуль значения сигнала;
2. Блок преобразует тип данных входного сигнала в один из следующих типов: символ, беззнаковый символ, короткое целое, короткое беззнаковое целое, целое, длинное целое, беззнаковое длинное целое, с плавающей точкой, с плавающей точкой двойной длины (типы данных определены как в C);
3. Деление значений одного входного сигнала на другой;
4. Умножение значения сигнала на коэффициент;
5. Обратная величина от входного сигнала ( $1/x$ );
6. Умножение входных сигналов;
7. Инвертирование значений сигнала;
8. Возведение значения сигнала в степень;
9. Знак значения (1 если значение сигнала больше 0 и  $-1$  если  $-$  меньше 0);
10. Сумматор;
11. Преобразование одной физической величины в другую (например, градусов Кельвина в градусы Цельсия и т.п.).

### Блоки булевых операций



1. Больше;
2. Больше либо равно;
3. Меньше;
4. Меньше либо равно;
5. Равно;
6. Не равно;
7. Логическое И;
8. Логическое ИЛИ;
9. Логическое НЕ.

Рабочая область программы разделена на два поля. Слева расположено вертикальное поле в котором отмечено название проекта над которым ведется работа. В случае создания иерархического проекта, когда структура моделируемого устройства содержит несколько вложенных моделей разных уровней, в левом поле отображается вся структура проекта. Большее правое поле предназначено для работы с моделью устройства.

### 2. Операции при работе с моделью

Наиболее необходимые операции при работе с моделью это:

- задание нового блока;
- редактирование параметров блока;
- соединение блоков модели связями;
- удаление связи из модели;
- удаление блока из модели;

– перемещение блока на новое место.

**Задание нового блока** проводится либо выбором его из меню (раздел **Blocks**), либо выбором одного из блоков с панели инструментов. После подтверждения (щелчок мышью) **выбора блока из меню** на рабочем поле в месте где установлен указатель появляется прямоугольник обозначенный штриховой линией, который перемещается совместно с указателем. Для **выбора блока с панели инструментов** необходимо щелкнуть по нему и затем перевести указатель на рабочую область (**внимание!** блоки с панели инструментов на рабочую область не перетаскиваются!). После щелчка мышью в указанном месте рабочей области появляется обозначение выбранного блока.

**Редактирование параметров блока** можно проводить после того как он помещен на рабочую область. При наведении на блок указатель со стрелки изменяется на перекрестье, щелкнув в этот момент правой кнопкой мыши Вы вызовете окно параметров блока, например, на рис. 3.1 приведено окно параметров блока **pulseTrain** (Генератор импульсов).

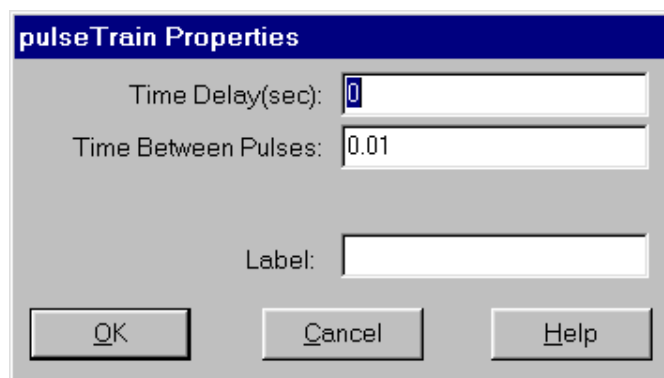


Рис. 3. 1 Окно параметров блока **pulseTrain** (Генератор импульсов)

После присвоения параметрам требуемых значений они могут быть либо приняты для моделирования (кнопка **OK**), либо по блоку получена справочная информация (кнопка **Help**, справка на английском языке), либо возможен отказ от внесения изменений (кнопка **Cancel**).

**Соединение блоков модели связями** может быть проведено после того, как оба соединяемых блока расположены на рабочей области. При наведении указателя на выход (или вход) блока он принимает вид вертикальной стрелки – ↑, удерживая в этот момент нажатой левую кнопку мыши необходимо показать с каким входом (или выходом) другого блока вы хотите соединить данную связь и затем отпустить нажатую кнопку (используется стандартная Windows – операция **Drag and Drop**). При этом допустимы только соединения «выход–вход», при попытке провести соединение выхода с выходом или входа со входом программа выдаст сообщение о невозможности такого соединения.

**Удаление связи из модели** проводится выбором ее конца (вход блока на который пришла удаляемая связь) и перетаскиванием ее (с удерживаемой нажатой левой кнопкой мыши) на пустое место на рабочей области (где кнопка мыши должна быть отпущена).

Для **удаления блока из модели** его необходимо сначала выделить (отметить). Для этого помещают указатель чуть выше и левее блока и, удерживая нажатой левую кнопку мыши, перемещают его ниже и правее блока. Во время этих действий блок оказывается охваченным штриховым прямоугольником. После отпущения кнопки цвет блока инвертируется, что говорит о том, что он выделен (отмечен). После нажатия на клавиатуре клавиши Delete, блок будет удален из модели. Для снятия выделения блока достаточно щелкнуть мышью на пустом месте рабочей области.

**Перемещение блока на новое место** проводится его простым перетаскиванием (операция **Drag and Drop**). При этом все существующие связи блока сохраняются.

### 3. Основные блоки для моделирования САУ

Практически любая сложная система может быть смоделирована в виде набора достаточно ограниченного числа простых, типовых блоков и устройств.

В программе **VisSim** САУ наиболее просто можно смоделировать используя следующие блоки:

- передаточная функция;
- источник входного сигнала;
- сумматор;
- инвертор значения сигнала;
- индикатор выходного сигнала.

#### **Передаточная функция.**

**Передаточной функцией** называется отношение преобразования Лапласа выходной величины к преобразованию Лапласа входной величины:

$$W(p) = \frac{Y(p)}{X(p)}.$$

**Частотной передаточной функцией** называется функция  $W(i\omega)$ , полученная из передаточной функции при подстановке в неё аргумента  $i\omega$ ,

$$W(i\omega) = \text{Re} + i\text{Im}.$$

Основным элементом для моделирования САУ в программе **VisSim** является блок **transferFunction** (**передаточная функция**), рис. 3.2, расположенный в разделе меню **Blocks**, группе **Linear System**.

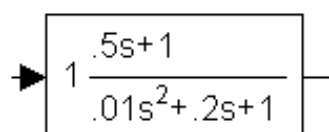


Рис. 3.2. Блок передаточная функция

Как видно из примера на рис. 3.2 динамическое звено моделируется передаточной функцией стандартного полиномиального вида:

$$W(s) = K \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_0}$$

где  $s = d/dt$  – оператор дифференцирования.

Окно описания параметров блока **transferFunction** (передаточная функция) приведено на рис. 3.3.

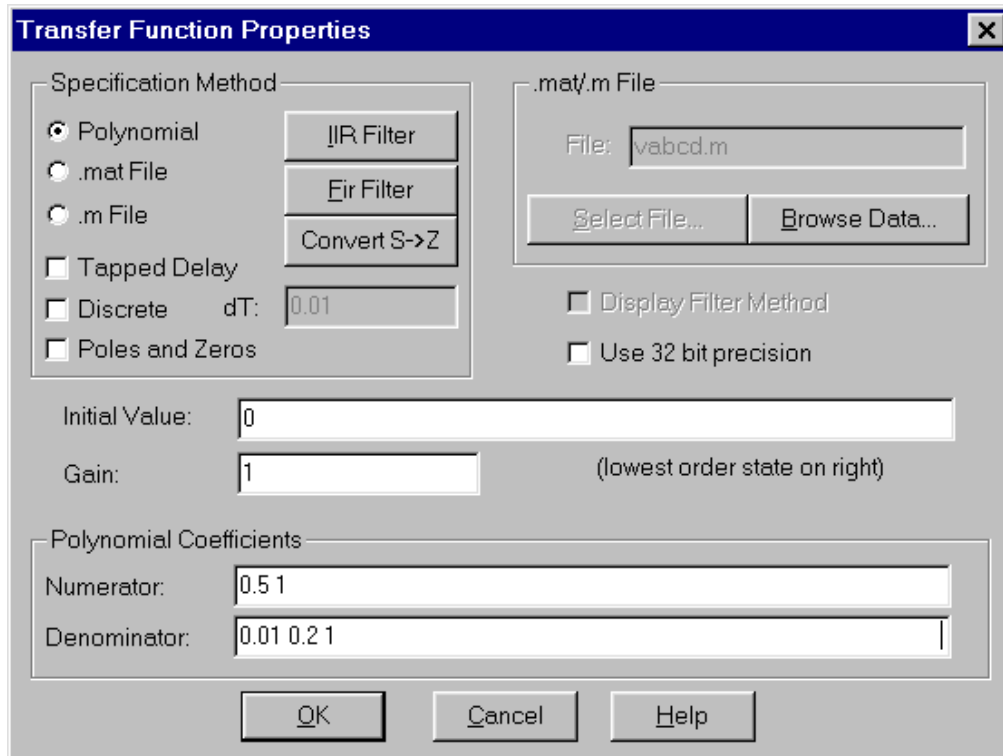


Рис. 3.3. Окно описания параметров блока **transferFunction** (передаточная функция)

Значение коэффициента усиления  $K$  передаточной функции задается в строке **Gain**. Значения коэффициентов числителя  $b_i$  перечисляются в строке **Numerator** через пробел начиная со старшей степени. Значения коэффициентов знаменателя  $a_i$  перечисляются в строке **Denominator** аналогично. Так на рис. 3.3 задается передаточная функция динамического звена изображенного на рис. 3.2. Остальные параметры в данном блоке при простом моделировании линейных непрерывных звеньев САУ задавать не требуется.

Данный блок позволяет также моделировать дискретные передаточные функции. Для перехода к ним необходимо в окне описания параметров блока **transferFunction** (передаточная функция) установить опцию **Discrete** (дискретность) и задать параметр **dT** (период дискретности). При выбранной опции **Discrete** (дискретность) в строке **Numerator** задаются значения коэффициентов числителя  $b_i$ , начиная со старшей степени оператора  $z$ , а в строке **Denominator** аналогично перечисляются значения коэффициентов знаменателя  $a_i$ .

### Источник входного сигнала.

Одним из основных видов анализа САУ является анализ их переходных процессов. Для проведения этого вида анализа в программе **VisSim** требуется на вход исследуемой системы подключить соответствующий источник входного сигнала. Так как в теории управления, в основном, проводится определение типовых переходных процессов – переходной функции или импульсной функции, то в качестве возмущающих воздействий используются единичная ступенчатая функция  $x(t) = 1(t)$  ( $x = 0$  при  $t \leq 0$  и  $x = 1$  при  $t > 0$ ) и единичная импульсная функция (дельта-функция)  $x(t) = \delta(t)$  ( $x = 0$  при  $t \neq 0$  и  $x = \infty$  при  $t = 0$ ).

Для задания сигнала типа ступенчатой функции используется блок **Step** (ступенчатый сигнал), расположенный на панели инструментов в группе **Producer blocks** (источники сигналов), рис. 3.4.

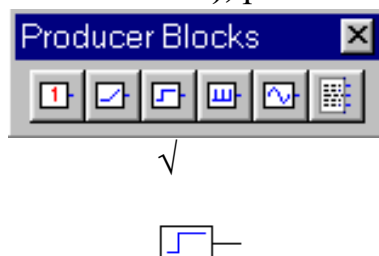


Рис. 3.4. Блок **Step** (ступенчатый сигнал), обозначение на панели инструментов и на рабочей области

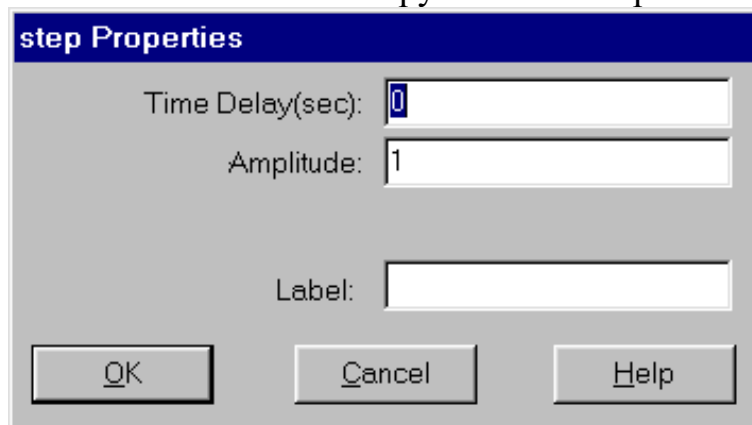


Рис. 3.5. Окно описания параметров блока **Step** (ступенчатый сигнал)

Окно описания параметров блока **Step** (ступенчатый сигнал), приведенное на рис. 3.5, содержит всего два задаваемых параметра: **Time Delay** – время переключения сигнала с нулевого уровня (в секундах) на значение – **Amplitude**.

Для задания сигнала типа импульсной функции, специального источника в **VisSim** нет, но его можно задать достаточно простым способом. Для этого используется источник сигнала **pulse Train** (генератор импульсов),



расположенный на панели инструментов в группе **Producer blocks** (источники сигналов), рис. 3.6.

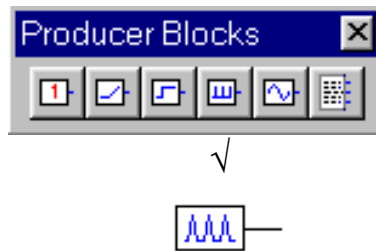


Рис. 3.6. Блок **pulse Train** (генератор импульсов), обозначение на панели инструментов и на рабочей области

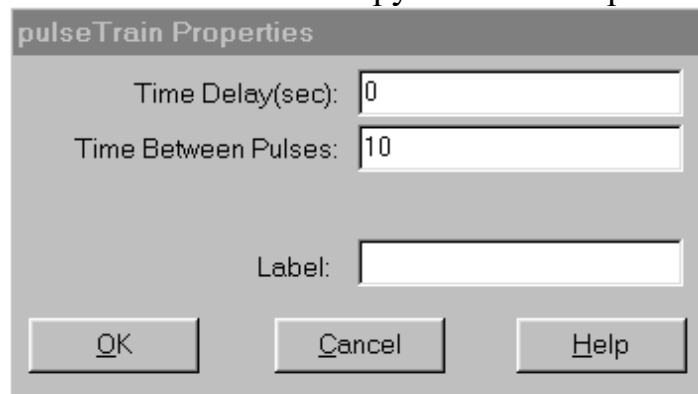


Рис. 3.7. Окно описания параметров блока **pulse Train** (генератор импульсов)

Окно описания параметров блока **Step** (ступенчатый сигнал), приведенное на рис. 3.7, содержит два задаваемых параметра: **Time Delay** – время начала генерации последовательности импульсов с единичной амплитудой; **Time Between Pulses** – время между импульсами. Установив время между импульсами достаточно большим (в несколько раз больше длительности переходного процесса в САУ) на выходе САУ фактически получим реакцию на единичный импульс.

**Сумматор и инвертор значения сигнала.** Данные блоки необходимы для организации в САУ отрицательной обратной связи.

Для суммирования сигналов в **VisSim** блок **summingJunction** (сумматор), расположенный на панели инструментов в группе **Arithmetic Blocks** (блоки арифметических операций), рис. 3.8. Специальных устанавливаемых параметров данный блок не имеет.

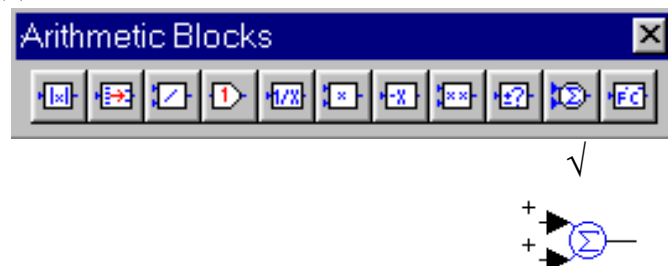


Рис. 3.8. Блок **pulse Train** (генератор импульсов),

обозначение на панели инструментов и на рабочей области  
Для того чтобы вычесть один сигнал из другого его необходимо инвертировать перед подачей на сумматор. Делается это блоком **negative** (**-X**) (**инвертор**), расположенным на панели инструментов в группе **Arithmetic Blocks** (**блоки арифметических операций**), рис. 3.9. Специальных устанавливаемых параметров данный блок также не имеет.

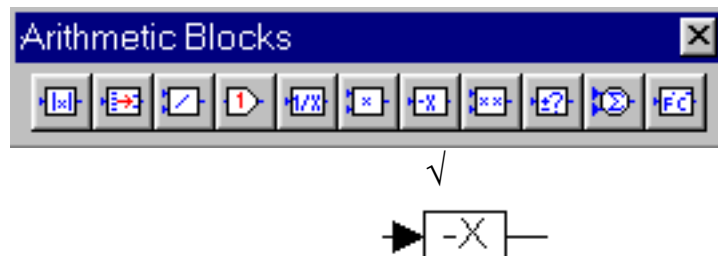


Рис. 3.9. Блок **negative** (**-X**) (**инвертор**),  
обозначение на панели инструментов и на рабочей области

**Индикатор выходного сигнала.** Для индикации выходного сигнала удобнее всего использовать блок **Plot** (**график**), расположенный на панели инструментов в группе **Consumer Blocks** (**индикаторы сигналов**), рис. 3.10.

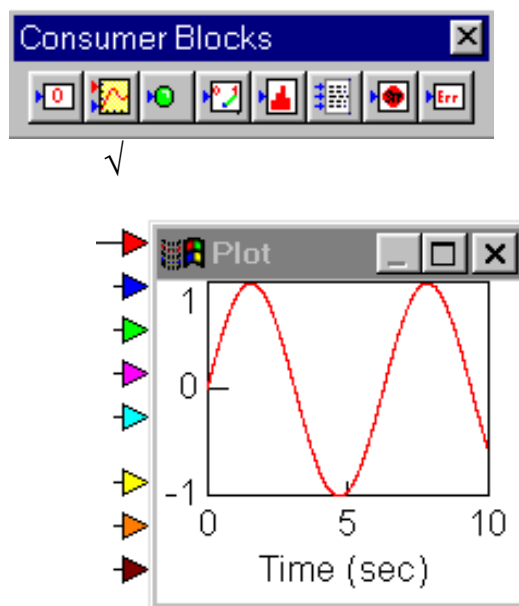


Рис. 3.10. Блок **Plot** (**график**),  
обозначение на панели инструментов и на рабочей области

На блок **Plot** (**график**) можно подать до 8 различных сигналов. В окне описания параметров блока **Plot** (**график**), рис. 3.11, имеет пять разделов:  
– **основные опции (Options)**;  
– **метки (Labels)**;

- оси (Axes);
- опции отображения блока (Appearance);
- опции отображения графиков (Traces).

Наиболее часто используются следующие:

в разделе **основные опции (Options)**:

**Log X** – установить по шкале X логарифмический масштаб;

**Log Y** – установить по шкале Y логарифмический масштаб;

**Decibel Y** – по шкале Y значения отображать в децибелах;

**Grid Lines** – нанести на поле графика сетку;

в разделе **оси (Axes)**:

**Y Upper Bound** – верхняя граница шкалы Y;

**Y Lower Bound** – нижняя граница шкалы Y;

**X Upper Bound** – верхняя граница шкалы X;

**X Lower Bound** – нижняя граница шкалы X;

**Time Scaling** – масштаб времени по шкале X.

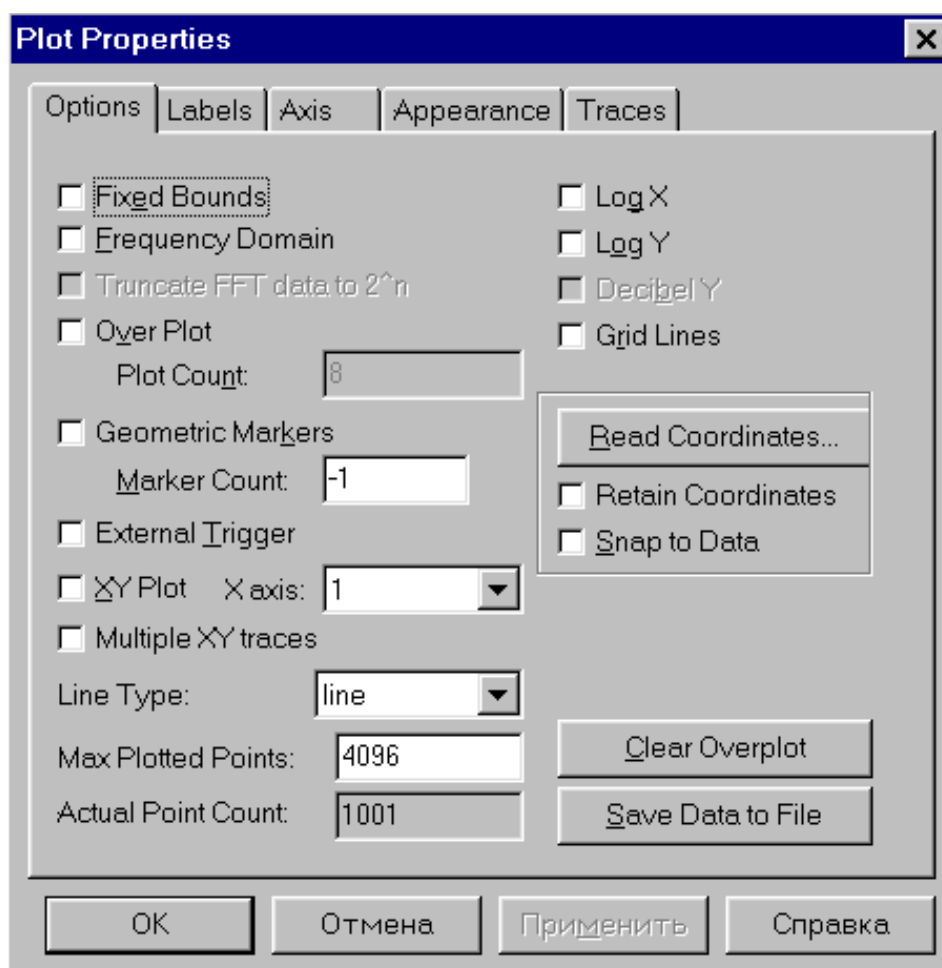


Рис. 3.11. Окно описания параметров блока **Plot (график)**

#### 4. Моделирование САУ.

Смоделировать работу САУ в программе **VisSim** возможно двумя способами:

- **в виде одного блока**, имеющего передаточную функцию системы;
- **в виде структурной схемы** исследуемой системы.

В первом случае САУ задается **в виде одного блока** (имеющего передаточную функцию разомкнутой или замкнутой системы), но такой способ имеет тот недостаток, что передаточные функции у сложных систем представляют, соответственно, достаточно сложные и громоздкие выражения с высокими порядками полиномов в числителе и знаменателе, а численное решение таких полиномов при моделировании влияет на его точность. Также возникают проблемы со сходимостью решений. Кроме того, при данном способе моделирования невозможно просмотреть сигналы сразу в нескольких точках структурной схемы, что часто бывает необходимо для анализа.

Данных недостатков лишен второй способ моделирования – **в виде воспроизведения структурной схемы** из набора отдельных блоков, часто типовых. Получившуюся схему замыкают в случае необходимости местными и/или общими обратными связями. Если система управления задана в виде структурной схемы, то сделать это сравнительно просто. Если она задана передаточной функцией (для разомкнутой или замкнутой системы), то необходимо преобразовать выражение к удобному виду и разбить на ряд более простых, каждое из которых в дальнейшем представляется одним блоком. В этом случае модели отдельных блоков проще, возможно снятие при моделировании параллельно нескольких сигналов в различных точках структурной схемы. Сложную структурную модель проще отладить по блоку.

В любом случае в модели обязательно должны присутствовать:

- источник сигнала,
- модель исследуемой системы,
- устройство индикации результатов моделирования.

Пример модели простой САУ приведен на рисунке 3.12.

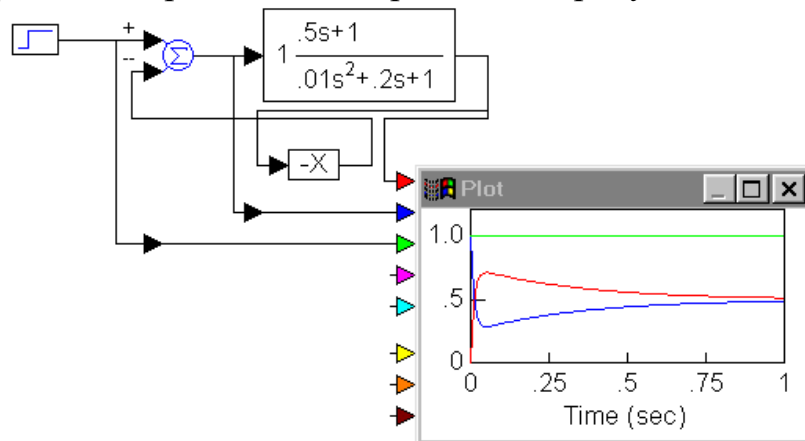



Рис. 3.12. Модель САУ для вычисления переходной функции

## 5. Построение переходных процессов

После того, как модель САУ набрана в рабочей области **VisSim**, на ее вход подключен требуемый источник сигнала, а выход системы подключен к блоку индикации, она полностью готова к моделированию. Начало моделирования задается щелчком по кнопке  (команда **Simulate/Go** или кнопка **F5**). Если модель задана корректно, то программа **VisSim** проведет расчет и результаты автоматически отобразятся на блоке индикации. В случае наличия ошибок в модели программа выдаст соответствующее сообщение на экран.

## 6. Построение частотных характеристик

Один из основных видов анализа САУ – частотный.

Зависимость фазового сдвига  $\varphi$  между выходными и входными колебаниями от частоты  $\omega$  называется **фазовой частотной характеристикой** (ФЧХ) и обозначается  $\varphi(\omega)$ ,

$$\varphi(\omega) = \arctg(\text{Im}/\text{Re}).$$

Зависимость отношения амплитуды выходных колебаний к амплитуде входных колебаний от частоты колебаний  $\omega$  называется **амплитудной частотной характеристикой** (АЧХ) и обозначается  $A(\omega)$ ,

$$A(\omega) = \sqrt{\text{Re}^2 + \text{Im}^2}.$$

**Амплитудно-фазовая характеристика** (АФЧХ) называется также **диаграммой Найквиста**, строится на комплексной плоскости и представляет собой годограф частотной передаточной функции при изменении частоты от нуля до бесконечности. То есть АФЧХ – это траектория, описываемая на комплексной плоскости концом радиуса-вектора, модуль и аргумент которого соответственно равны  $A(\omega)$  и  $\varphi(\omega)$ , при изменении частоты от нуля до бесконечности.

Программа **VisSim** позволяет строить как амплитудно-частотную и фазо-частотную характеристики (**Frequency Response**), так и амплитудно-фазовую частотную характеристику (**Nyquist Response**). Для построения характеристики звена, группы звеньев или всей системы в целом их необходимо выделить (см. ранее раздел **Удаление блока из модели**). Затем в разделе меню **Analyze (Анализ)** выбирается соответствующий вид представления частотных характеристик. Частотные характеристики, построенные для выделенного объекта, будут помещены программой в новом окне непосредственно на рабочей области, рис. 3.13.

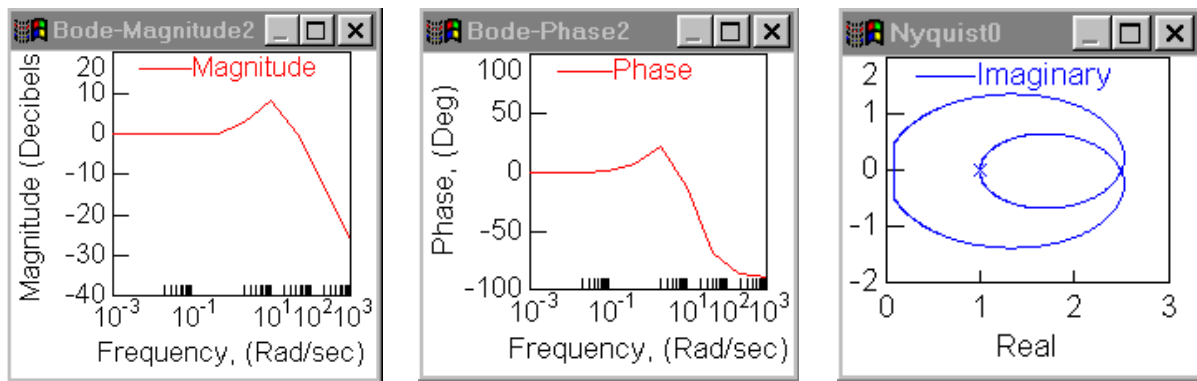


Рис. 3.13. Примеры построения АЧХ, ФЧХ и АФЧХ программой **VisSim**

Структура окон, в которые помещаются графики АЧХ, ФЧХ и АФЧХ аналогична окну блока **Plot (график)** и они имеют одинаковые параметры (см. ранее раздел **Индикатор выходного сигнала**) изменяя которые можно задать наиболее удачное визуальное представление характеристик. Для АЧХ, ФЧХ (**Response**) можно задать диапазон частот и шаг, с которым они будут рассчитаны. Это можно сделать в опции **Frequency Range (Диапазон частот)**, рис. 3.14, расположенной в том же разделе.

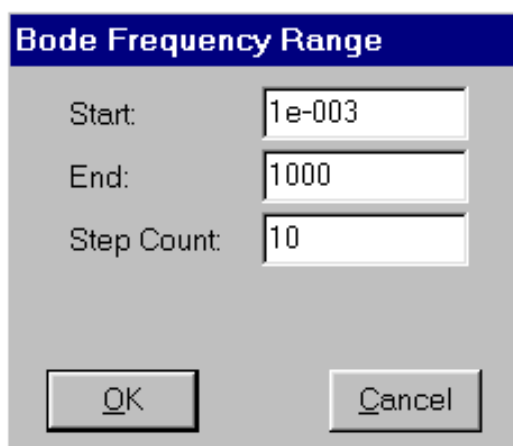


Рис. 3.14. Окно **Frequency Range (Диапазон частот)** для построения АЧХ и ФЧХ

В данном окне:

**Start** – начальное значение частоты;

**End** – конечное значение частоты;

**Step Count** – шаг приращения частоты.

Рассмотренное выше относится к основам моделированию линейных, стационарных САУ. При необходимости исследовать процессы в системах с переменными коэффициентами или в системах с временным запаздыванием к линейным блокам добавляются соответственно блоки переменных коэффициентов и блоки временного запаздывания. Добавление нелинейных блоков позволяет исследовать процессы в нелинейных системах. Наличие в

программе **VisSim** дискретных передаточных функций позволяет включать в структурную модель системы блоки аналого-цифрового и цифро-аналогового преобразования, устройства обработки цифровой информации. Все это существенно повышает эффективность моделей, так как позволяет сравнительно просто, быстро и достаточно точно исследовать процессы в особых линейных, сложных нелинейных или цифровых системах, что является в большинстве случаев недоступным для аналитических методов расчета.

#### 6. Решение неоднородных дифференциальных уравнений

Система управления, как любая динамическая система, описывается неоднородными дифференциальными уравнениями. Для линейных систем управления эти уравнения имеют постоянные коэффициенты

$$y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_1y' + a_0y = b_m f^{(m)} + b_{m-1}f^{(m-1)} + \dots + b_1f' + b_0f + \\ + g_k u^{(k)} + g_{k-1}u^{(k-1)} + \dots + g_1u' + g_0u,$$

где  $y(t)$ ,  $y'(t)$ , ...,  $y^{(n)}(t)$  – неизвестная функция времени и ее производные;  $f(t)$  и  $u(t)$  – известные функции.

В этой связи промоделируем решение обыкновенного неоднородного дифференциального уравнения

$$\ddot{y} + 3\dot{y} + y + 3y = 5t + 1,$$

где  $y$  – некоторая искомая функция времени на отрезке  $[0; T]$  при нулевых начальных условиях. Для достижения поставленной цели необходимо проделать следующие процедуры:

1) разрешить заданное уравнение относительно наивысшей из производных

$$\ddot{y} = -3\dot{y} - y - 3y + 5t + 1;$$

2) составить схему модели (рис.3.15), реализующую это уравнение, для чего выбрать следующие блоки:

- три интегратора (INTEGRATOR), которые разместим последовательно друг за другом;
- четыре блока с постоянными коэффициентами передачи (GAIN), которые равны соответствующим коэффициентам уравнения;
- четыре блока-сумматора (SUMMINGJUNCTION);
- один блок задания постоянной величины (CONST);
- блок источника сигнала (RAMP);
- блок вывода графиков (PLOT) для наблюдения процессов.

Входу крайнего левого интегратора соответствует переменная  $\ddot{y}$ , входу среднего интегратора –  $\dot{y}$ , а входу крайнего правого интегратора –  $y$ . Выход этого интегратора будет соответствовать переменной  $y$ ;

3) соединить входы и выходы соответствующих блоков, для чего необходимо подвести курсор к выходу (входу) блока так, чтобы курсор принял вид черной вертикальной стрелки, нажать ЛК мыши и, удерживая ее, соединить выход (вход) одного блока с входом (выходом) другого.

С целью более упорядоченного расположения связей иной блок следует развернуть на  $180^\circ$ , для чего необходимо выделить этот блок (либо с помощью клавиши Shift и щелчка ЛК мыши, либо обвести элемент прямоугольником, удерживая ЛК мыши) и затем нажать Alt+«стрелка влево», или воспользоваться пунктом Rotate меню Edit. После выполнения всех соединений блоков получим следующее изображение на экране;

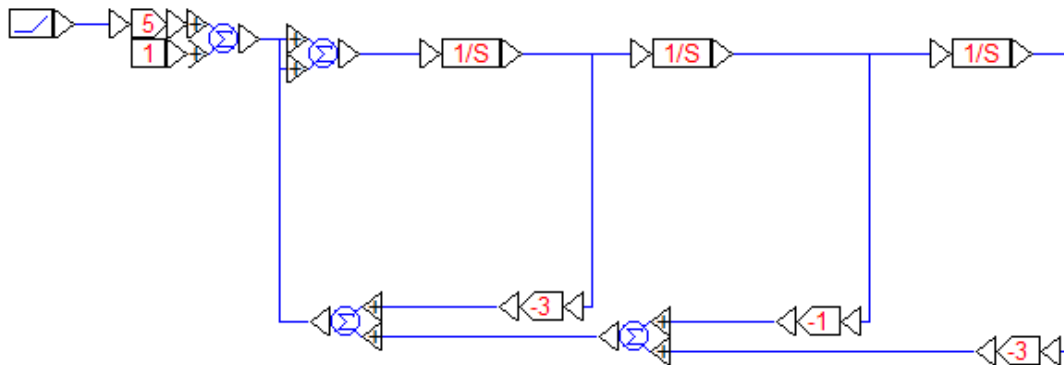


Рис. 3.15. Графическое представление уравнения

4) указать метод интегрирования и шаг разбиения по сетке времени, войдя в меню Simulate/Simulation Properties. Например: диапазон моделирования (время начала и окончания, шаг), алгоритм интегрирования (метод Эйлера, Рунге-Кутты и т.д.);

5) инициировать процесс моделирования, войдя в меню Simulate/Go;

6) зафиксировать графический результат моделирования и проанализировать его с помощью блока Plot, входы которого могут быть соединены с любой точкой анализа.

#### 7. Моделирование систем и объектов управления, представленных передаточными функциями

В предыдущих пунктах рассмотрен один из способов решения дифференциальных уравнений. Второй способ моделирования заключается в решении уравнения, представленного передаточной функцией, которая связывает изображение входной и выходной величин при нулевых начальных условиях. В данном случае исходное дифференциальное уравнение может иметь только две функции – искомую и известную

$$y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_1y' + a_0y = b_m f^{(m)} + b_{m-1}f^{(m-1)} + \dots + b_1f' + b_0f.$$

Формально получить передаточную функцию можно путем замены оператора дифференцирования  $d/dt$  на комплексную переменную  $s$  в степени, равной порядку производной (теоретическое обоснование такой операции вытекает из преобразований Лапласа). В соответствии с этим перепишем уравнение и определим передаточную функцию

$$s^n Y(s) + a_{n-1}s^{n-1}Y(s) + \dots + a_1sY(s) + a_0Y(s) = b_m s^m F(s) + b_{m-1}s^{m-1}F(s) + \dots + b_1sF(s) + b_0F(s).$$

Решим это уравнение относительно изображения  $Y(s)$  и получим



$$Y(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} F(s),$$

где

$$W(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} -$$

передаточная функция.

Изменим уравнение, с которым работали в пункте 6, исключив вторую известную функцию (единичный скачок)  $\ddot{y} + 3\dot{y} + y + 3y = 5t$ . С использованием аппарата передаточных функций это уравнение можно привести к виду:

$$X(s) = \frac{5}{s^3 + 3s^2 + s + 3} \cdot \frac{1}{s^2},$$

где  $1/s^2$  – изображение временной функции  $t$ .

Уравнению  $\ddot{y} + 3\dot{y} + y + 3y = 1$  соответствует уравнение

$$X(s) = \frac{1}{s^3 + 3s^2 + s + 3} \cdot \frac{1}{s}.$$

Таким образом, полное решение анализируемого уравнения

$$X(s) = \frac{5}{s^3 + 3s^2 + s + 3} \cdot \frac{1}{s^2} + \frac{1}{s^3 + 3s^2 + s + 3} \cdot \frac{1}{s}.$$

В процессе моделирования нет необходимости задавать изображения известных функций. Они в данном случае задаются двумя блоками: RAMP и STEP как функции времени. Для моделирования необходимо проделать следующие процедуры:

1). Вставить два блока TRANSFERFUNCTION. Для ввода передаточной функции нажать ПК мыши на только что вставленном блоке, появится диалоговое окно. В поле GAIN первого блока необходимо ввести число 5, в поле Numertor необходимо ввести значение числителя – 1, в поле Denominator – значение знаменателя в виде разделенных пробелами чисел при  $s$  от старшей степени к младшей: 1 3 1 3.

2). Вставить блок SUMMINGJUNCTION. Соединить его входы с выходами интеграторов (блоки, описывающие передаточные функции).

3). Вставить блок RAMP. Его выход соединить со входом интегратора, который имеет коэффициент передачи, равный пяти.

4). Вставить блок STEP. Его выход соединить со входом другого интегратора, у которого коэффициент передачи равен 1.

5). Вставить блок PLOT. На его вход подать сигнал с выхода сумматора.