

Министерство образования Республики Беларусь

Учреждение образования  
«Гомельский государственный технический  
университет имени П. О. Сухого»

Кафедра «Автоматизированный электропривод»

**В. С. Захаренко**

## **МОДЕЛИРОВАНИЕ В ЭЛЕКТРОПРИВОДЕ**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ  
по одноименной дисциплине  
для студентов специальности 1-53 01 05  
«Автоматизированные электроприводы»  
дневной формы обучения**

Гомель 2017

УДК 62-83-52:004.94(075.8)  
ББК 31.291я73  
3-38

*Рекомендовано научно-методическим советом  
факультета автоматизированных и информационных систем  
ГГТУ им. П. О. Сухого  
(протокол № 5 от 28.12.2015 г.)*

Рецензент: зав. каф. «Промышленная электроника» ГГТУ им. П. О. Сухого  
канд. техн. наук, доц. *Ю. В. Крышнев*

**Захаренко, В. С.**  
3-38 Моделирование в электроприводе : учеб.-метод. пособие по одноим. дисциплине для студентов специальности 1-53 01 05 «Автоматизированные электроприводы» днев. формы обучения / В. С. Захаренко. – Гомель : ГГТУ им. П. О. Сухого, 2017. – 89 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

Изложены теоретические сведения, необходимые для выполнения лабораторных работ по дисциплине «Моделирование в электроприводах». Приведены программа работы и общие методические указания. В ходе выполнения лабораторных работ изучается математический аппарат моделирования и некоторые особенности моделей элементов систем управления и, в целом, моделей систем управления электроприводами. Рассмотрены приемы работы с программным обеспечением при выполнении расчетов, проводимых при синтезе и анализе систем автоматического управления электроприводами.

Для студентов специальности 1-53 01 05 «Автоматизированные электроприводы» дневной формы обучения.

**УДК 62-83-52:004.94(075.8)  
ББК 31.291я73**

© Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», 2017

## **Введение**

По дисциплине «Моделирование в электроприводах», согласно учебного плана, выполняются лабораторные работы.

Цель работ – закрепить полученные теоретические знания и научиться применять их для практических расчетов.

Отчет по лабораторным работам должен выполняться на листах формата А4. Отчет должен содержать:

- титульный лист;
- цель работы;
- задание на лабораторную работу (структурная схема модели или система дифференциальных уравнений, числовые данные, указанные в задании);
- вывод системы уравнений (если он выполнялся);
- расчет параметров модели;
- текст программы, или документ MathCAD, или модель в Matlab Simulink;
- результаты выполненных расчетов;
- выводы.

# 1. Теоретические сведения

## 1.1. Математическое моделирование

Моделирование – процесс изучения свойств объекта путем изучения свойств другого, более удобного для исследования объекта, находящегося в некоторой степени подобия с первым объектом. Математическая модель – система уравнений, описывающих взаимосвязь характеристик и параметров объекта и отражающая физические процессы, происходящие в объекте.

Математическое моделирование производится путем численного решения дифференциальных уравнений, которое заключается в расчете значений переменных на текущем шаге  $Y_i$  по значениям переменных модели на других шагах расчета. Фактически производится численное интегрирование функций, соответствующих правым частям дифференциальных уравнений, записанных в форме Коши.

$$\frac{dy_j}{dt} = f_j(t, X, Y);$$
$$y_j(t) = \int_0^t f_j(t, X, Y) \cdot dt.$$

Математическое моделирование осуществляется в соответствии с одним из двух принципов:

1. По системе уравнений.
2. Структурное моделирование (моделирование по структурной схеме).

При моделировании по системе уравнений на основании системы составляется подпрограмма расчета производных переменных. На основании выбранного численного метода решения дифференциальных уравнений составляется подпрограмма расчета значений переменных на текущем шаге  $Y_i$ . Данная подпрограмма вызывает подпрограмму расчета производных. В программе реализуется цикл по шагам расчета, внутри которого рассчитываются значения переменных на текущем шаге путем вызова подпрограммы реализации численного метода и выводится результат расчета. Т.о. программа имеет структуру, представленную на рис. 1.1.

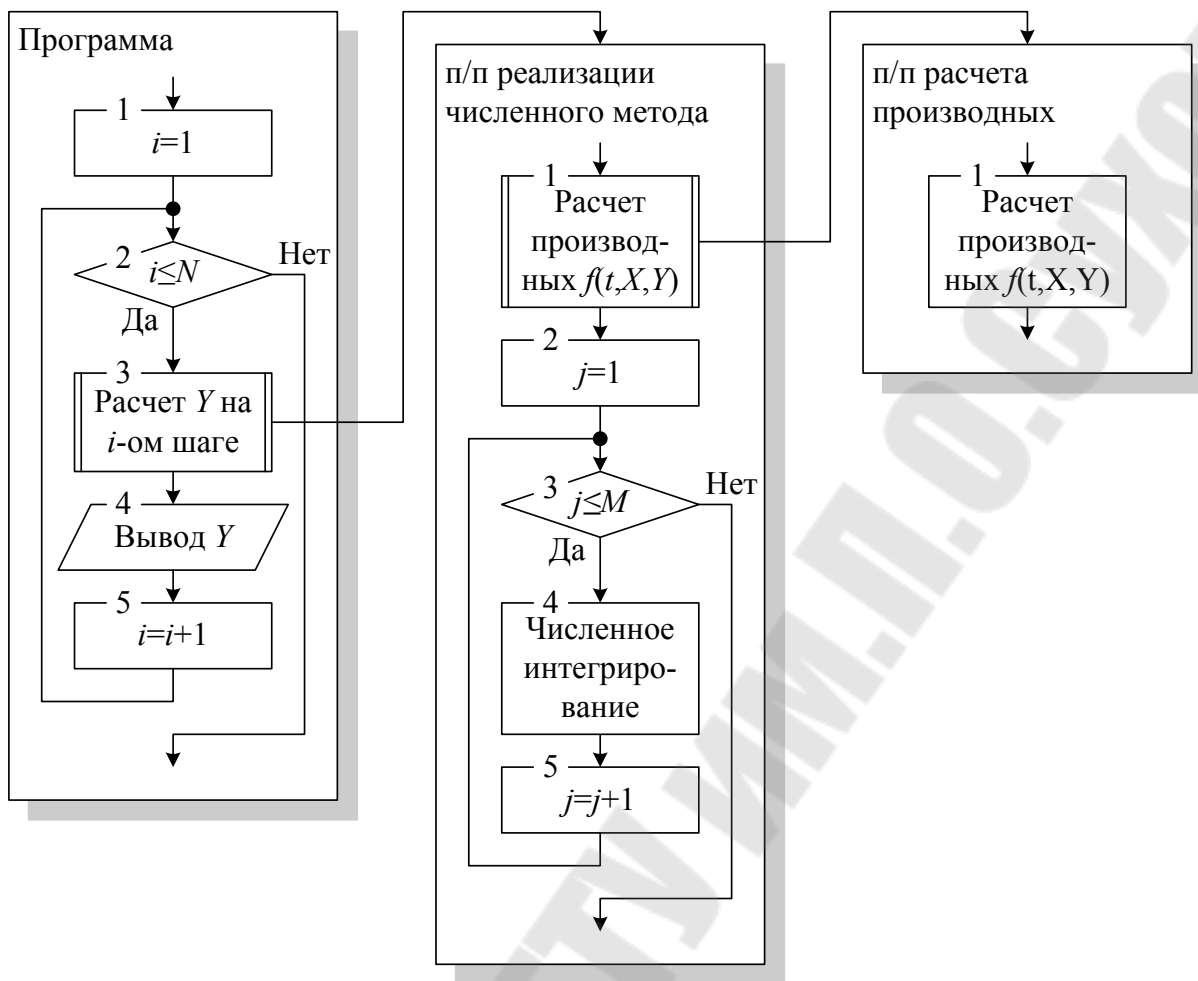


Рис. 1.1. Структура программы при моделировании по системе дифференциальных уравнений.

При структурном моделировании для всех типов звеньев, входящих в состав модели, на основании выбранного численного метода и уравнения, описывающего звено, составляются подпрограммы для определения значения выходного сигнала звена. По структурной схеме модели составляется подпрограмма расчета значений переменных на текущем шаге  $Y_i$ . Эта подпрограмма в порядке следования сигналов в модели вызывает подпрограммы звеньев. Сама программа выполняется аналогично моделированию по системе уравнений. Структура программы показана на рис. 1.2.

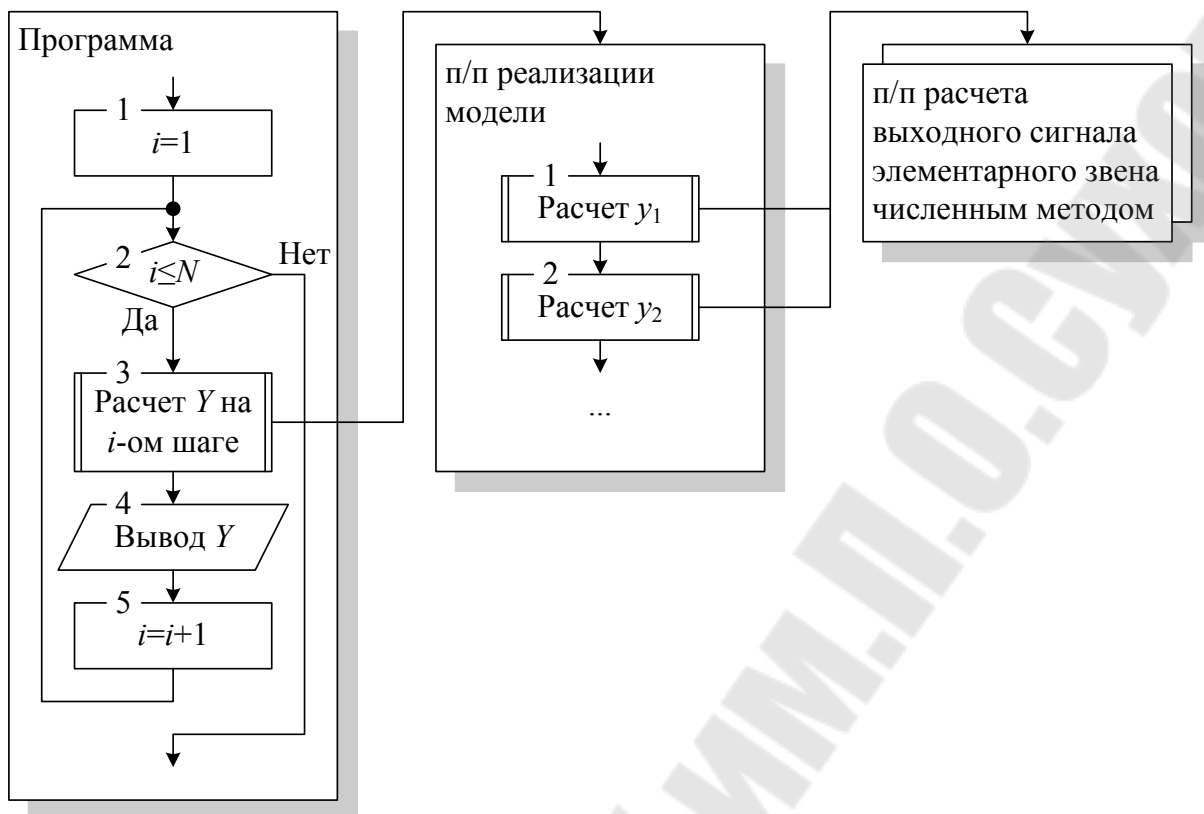


Рис. 1.2. Структура программы при структурном моделировании.

## 1.2. Методы численного решения дифференциальных уравнений

Расчетные соотношения и параметры основных численных методов приведены в табл. 1.1. В расчетных соотношениях приняты обозначения:

- $Y$  – вектор переменных модели,
- $i$  – номер текущего шага расчета, для которого рассчитываются значения переменных модели (соответствует текущему моменту времени  $t$ ),
- $\Delta t$  – шаг расчета (интервал времени между соседними шагами расчета),
- $f$  – вектор значений правых частей дифференциальных уравнений, записанных в форме Коши.
- $v$  – номер текущей итерации.

Таким образом  $Y_i$  – вектор значений переменных на текущем шаге расчета, т.е. в момент времени  $t = i \cdot \Delta t$ ,  $Y_{i-1}$  – на предыдущем шаге расчета ( $t - \Delta t$ ) и т.д. Значения  $Y_{i-0,5}$  соответствуют времени посередине между текущим и предыдущим шагами расчета.

Таблица 1.1.

## Расчетные соотношения и параметры численных методов.

Название метода, порядок точности, условие численной устойчивости	Расчетное соотношение
Эйлера. $p = 1, \Delta t < T$ .	$Y_i = Y_{i-1} + \Delta t \cdot f_{i-1}$
Неявный метод Эйлера. $p = 1$ .	$Y_i^{(v)} = Y_{i-1} + \Delta t \cdot f_i^{(v-1)}$
Усовершенствованный метод Эйлера. $p = 2, \Delta t < 2 \cdot T$ .	$Y_i^{(1)} = Y_{i-1} + \Delta t \cdot f_{i-1},$ $Y_i^{(2)} = Y_{i-1} + \frac{\Delta t}{2} \cdot (f_{i-1} + f_i^{(1)})$
Метод Рунге-Кутты. $p = 4,$ $\Delta t < 2,7853 \cdot T$ .	$Y_{i-0,5}^{(1)} = Y_{i-1} + \frac{\Delta t}{2} \cdot f_{i-1},$ $Y_{i-0,5}^{(2)} = Y_{i-1} + \frac{\Delta t}{2} \cdot f_{i-0,5}^{(1)},$ $Y_i^{(1)} = Y_{i-1} + \Delta t \cdot f_{i-0,5}^{(2)},$ $Y_i^{(2)} = Y_{i-1} + \frac{\Delta t}{6} \cdot (f_{i-1} + 2 \cdot f_{i-0,5}^{(1)} + 2 \cdot f_{i-0,5}^{(2)} + f_i^{(1)})$
Метод Адамса-Бэшфорга. $p = 2$ .	$Y_i = Y_{i-1} + \frac{\Delta t}{2} \cdot (3 \cdot f_{i-1} - f_{i-2})$
Метод Адамса-Мултона. $p = 4$ .	$Y_i^{(1)} = Y_{i-1} + \frac{\Delta t}{24} \cdot (55 \cdot f_{i-1} - 59 \cdot f_{i-2} + 37 \cdot f_{i-3} - 9 \cdot f_{i-4}),$ $Y_i^{(2)} = Y_{i-1} + \frac{\Delta t}{24} \cdot (9 \cdot f_i^{(1)} + 19 \cdot f_{i-1} - 5 \cdot f_{i-2} + f_{i-3})$
Метод Хэмминга.	$Y_i^{(1)} = Y_{i-4} + \frac{4}{3} \cdot \Delta t \cdot (2 \cdot f_{i-1} - f_{i-2} + 2 \cdot f_{i-3}),$ $Y_i^{(2)} = Y_i^{(1)} + \frac{112}{121} \cdot (Y_{i-1} - Y_{i-1}^{(1)}),$ $Y_i^{(v)} = \frac{1}{8} \cdot (9 \cdot Y_{i-1} - Y_{i-3}) + \frac{3}{8} \cdot \Delta t \cdot (f_i^{(v-1)} + 2 \cdot f_{i-1} - f_{i-2})$
Метод Башарина. $p = 2, \Delta t < 2 \cdot T$ .	$Y_i^{(v)} = Y_{i-1} + \Delta t \cdot f\left(t_i - \frac{\Delta t}{2}, X\left(t_i - \frac{\Delta t}{2}, \frac{Y_{i-1} + Y_i^{(v-1)}}{2}\right)\right)$

Некоторые методы имеют несколько итераций, номер которой записывается в верхнем индексе. Т.е. значения переменных модели на текущем шаге рассчитываются несколько раз, постепенно уточняясь.

Если в расчетном соотношении метода отсутствует выражение для первой итерации ( $Y_i^{(1)}$ ), то его следует рассчитывать по методу Эйлера.

Индексы значений правых частей уравнений показывают какие значения переменных  $Y$  и времени подставляются в выражения правых частей  $f(t, X, Y)$ . Например:  $f_{i-1} = f(t - \Delta t, X(t - \Delta t), Y_{i-1})$ ,  $f_i^{(v-1)} = f(t, X(t), Y_i^{(v-1)})$ ,  $f_{i-0,5}^{(1)} = f(t - 0,5 \cdot \Delta t, X(t - 0,5 \cdot \Delta t), Y_{i-0,5}^{(1)})$ .

### 1.3. Статические звенья

В состав любой системы автоматического управления входят статические элементы, такие как сумматоры, вычитатели, перемножители, звенья с ограничением, с зоной нечувствительности и др. В таблице 1.2 приведены соотношения для расчета выходного сигнала звеньев с нелинейными характеристиками.

### 1.4. Динамические звенья

В состав автоматизированных электроприводов входят различные динамические элементы. Их обозначения и передаточные функции приведены в таблице 1.3.

При составлении системы уравнений следует иметь в виду, что уравнение записывается для выходного сигнала звена. Поэтому для интегрирующего и аperiodического звеньев составляются дифференциальные уравнения в форме Коши, а для дифференцирующего и форсирующего – алгебраические. Например, передаточная функция аperiodического звена

$$W(p) = \frac{y(p)}{x(p)} = \frac{K}{T \cdot p + 1}$$

где:  $x(p)$  – операторное изображение входного сигнала звена,  
 $y(p)$  – операторное изображение выходного сигнала звена,  
 $K$  – коэффициент передачи звена,  
 $T$  – постоянная времени звена.  
 Выразим  $y(p)$



Таблица 1.2.

## Расчетные соотношения для статических нелинейных звеньев.

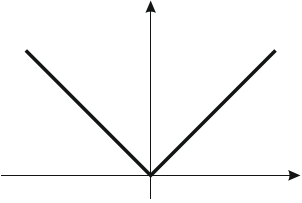
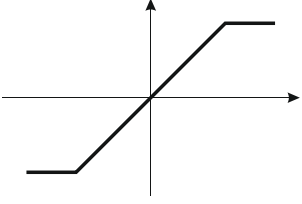
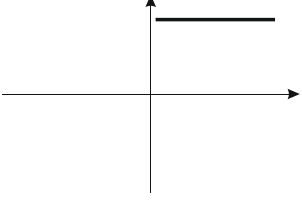
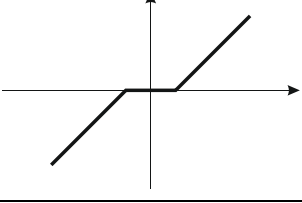
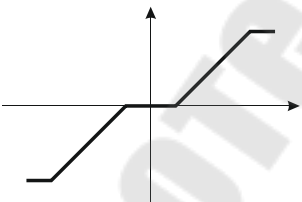
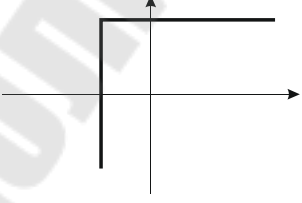
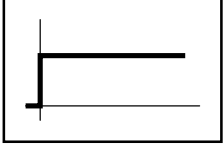
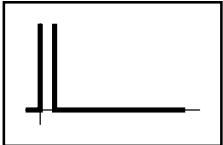
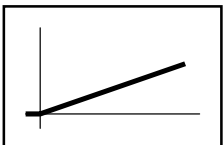
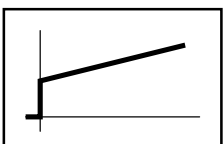
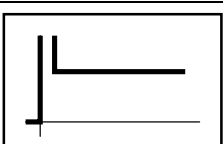
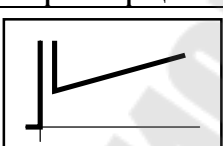
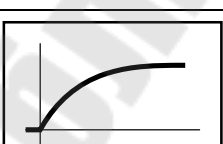
Статическая характеристика звена	Расчетное соотношение
Выделитель модуля	
	$y =  x .$
Звено с ограничением	
	$y = K \cdot x,$ $y = \begin{cases} Y_{0+}, & \text{при } y > Y_{0+}, \\ Y_{0-}, & \text{при } y < Y_{0-}. \end{cases}$
Релейный элемент (сигнатура)	
	$y = \begin{cases} Y_{0+}, & \text{при } x > 0, \\ 0, & \text{при } x = 0, \\ Y_{0-}, & \text{при } x < 0. \end{cases}$
Звено с зоной нечувствительности	
	$y = \begin{cases} K \cdot (x - X_{0+}), & \text{при } x > X_{0+}, \\ 0, & \text{при } X_{0-} \leq x \leq X_{0+}, \\ K \cdot (x + X_{0-}), & \text{при } x < X_{0-}. \end{cases}$
Звено с зоной нечувствительности и ограничением	
	$y = \begin{cases} K \cdot (x - X_{0+}), & \text{при } x > X_{0+}, \\ 0, & \text{при } X_{0-} \leq x \leq X_{0+}, \\ K \cdot (x + X_{0-}), & \text{при } x < X_{0-}, \end{cases}$ $y = \begin{cases} Y_{0+}, & \text{при } y > Y_{0+}, \\ Y_{0-}, & \text{при } y < Y_{0-}. \end{cases}$
Звено с петлей гистерезиса	
	$y = \begin{cases} Y_{0+}, & \text{при } x > X_{0+}, \\ Y_{0-}, & \text{при } x < X_{0-}. \end{cases}$

Таблица 1.3.

## Передаточные функции динамических звеньев.

Обозначение звена на функциональной схеме	Передаточная функция
Безинерционное звено, пропорциональное звено (П-звено)	
	$W(p) = K$
Дифференцирующее звено (И-звено)	
	$W(p) = T \cdot p$
Интегрирующее звено (И-звено)	
	$W(p) = \frac{1}{T \cdot p}$
Пропорционально-интегрирующее звено (ПИ-звено)	
	$W(p) = K \cdot \frac{T \cdot p + 1}{T \cdot p} = \frac{T_1 \cdot p + 1}{T_2 \cdot p}$
Форсирующее звено, пропорционально-дифференцирующее звено (ПД-звено)	
	$W(p) = K \cdot (T \cdot p + 1)$
Пропорционально-интегро-дифференцирующее звено (ПИД-звено)	
	$W(p) = K \cdot \frac{(T_1 \cdot p + 1) \cdot (T_2 \cdot p + 1)}{T_3 \cdot p} = \frac{(T_1 \cdot p + 1) \cdot (T_2 \cdot p + 1)}{T_3 \cdot p}$
Апериодическое звено	
	$W(p) = \frac{K}{T \cdot p + 1}$

$$y(p) = \frac{K}{T \cdot p + 1} \cdot x(p).$$

Умножим на  $T \cdot p + 1$  и раскроем скобки

$$\begin{aligned} y(p) \cdot (T \cdot p + 1) &= K \cdot x(p), \\ T \cdot p \cdot y(p) + y(p) &= K \cdot x(p). \end{aligned}$$

Оставим в левой части только  $p \cdot y(p)$ , чтобы при переходе от операторной формы к функциям времени получить дифференциальное уравнение в нормальной форме Коши

$$\begin{aligned} T \cdot p \cdot y(p) &= K \cdot x(p) - y(p), \\ p \cdot y(p) &= \frac{K \cdot x(p) - y(p)}{T}, \\ \frac{dy}{dt} &= \frac{K \cdot x - y}{T}. \end{aligned}$$

Для форсирующего получится алгебраическое уравнение с производной для входного сигнала звена

$$\begin{aligned} W(p) = \frac{y(p)}{x(p)} &= K \cdot (T \cdot p + 1), \\ y(p) &= K \cdot (T \cdot p + 1) \cdot x(p), \\ y(p) &= K \cdot T \cdot p \cdot x(p) + K \cdot x(p), \\ y &= K \cdot T \cdot \frac{dx}{dt} + K \cdot x. \end{aligned}$$

Если перед форсирующим звеном стоит апериодическое или интегрирующее, как показано на рис. 1.3, то в алгебраическое уравнение для форсирующего звена можно подставить выражение для производной его входного сигнала  $y$  из уравнения в форме Коши предыдущего звена.

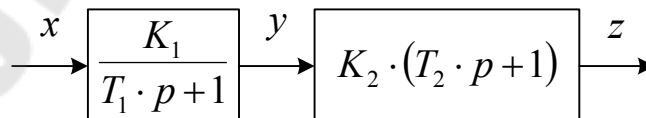


Рис. 1.3. Пример структурной схемы.

$$\begin{cases} y(p) = \frac{K_1}{T_1 \cdot p + 1} \cdot x(p), \\ z(p) = K_2 \cdot (T_2 \cdot p + 1) \cdot y(p), \end{cases}$$

$$\begin{cases} \frac{dy}{dt} = \frac{K_1 \cdot x - y}{T_1}, \\ z = K_2 \cdot T_2 \cdot \frac{dy}{dt} + y, \end{cases}$$

$$\begin{cases} \frac{dy}{dt} = \frac{K_1 \cdot x - y}{T_1}, \\ z = K_2 \cdot T_2 \cdot \frac{K_1 \cdot x - y}{T_1} + y. \end{cases}$$

Если в правой части присутствует производная переменной, для которой в системе записано алгебраическое уравнение, или производная входного сигнала, то можно прибегнуть к численному дифференцированию, заменив производные на отношения

$$\frac{dz}{dt} \approx \frac{\Delta z}{\Delta t} = \frac{z_i - z_{i-1}}{\Delta t}, \quad \frac{dx}{dt} \approx \frac{\Delta x}{\Delta t} = \frac{x(t) - x(t - \Delta t)}{\Delta t}.$$

### 1.5. Модели двигателей постоянного тока независимого и последовательного возбуждения

При постоянстве магнитного потока ( $k\Phi = k\Phi_H = const$ ) из полной модели двигателя можно исключить цепь возбуждения. При этом структурная схема модели примет вид, представленный на рис. 1.4.

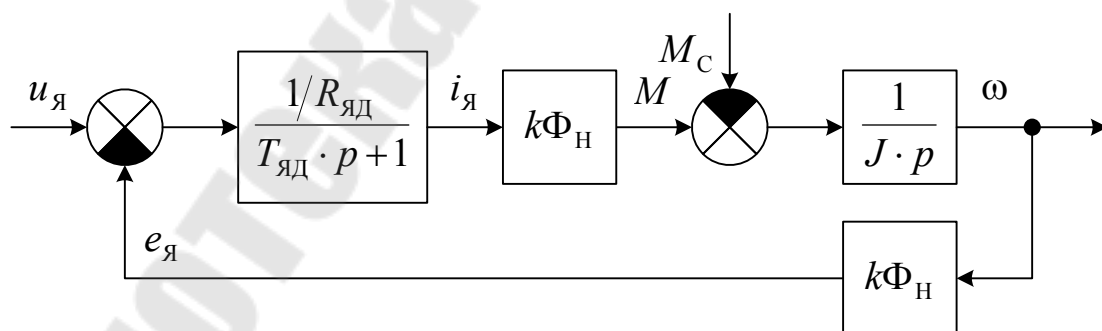


Рис. 1.4. Структурная схема модели ДПТ НВ.

На схеме обозначено:

- $u_я$  – напряжение якоря,
- $e_я$  – ЭДС якоря,
- $i_я$  – ток якоря,

- $M$  – момент двигателя,
- $M_C$  – момент сопротивления,
- $\omega$  – угловая скорость вращения,
- $R_{яд}$  – активное сопротивление якорной цепи двигателя,
- $T_{яд}$  – постоянная времени якорной цепи

$$T_{яд} = \frac{L_{яд}}{R_{яд}},$$

$L_{яд}$  – индуктивность якорной цепи двигателя.

Для двигателей последовательного возбуждения необходимо учитывать нелинейность кривой намагничивания. Структурная схема модели двигателя последовательного возбуждения при упрощенном представлении цепи возбуждения (без учета влияния вихревых токов) представлена на рис. 1.5.

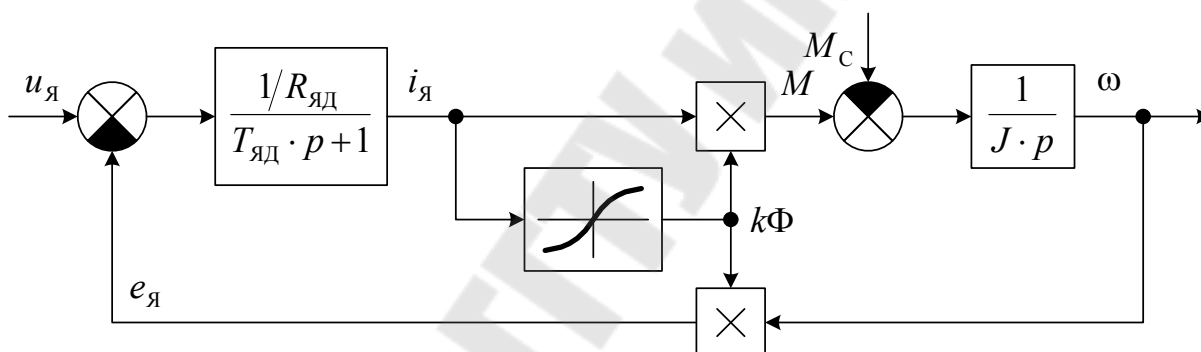


Рис. 1.5. Структурная схема модели ДПТ ПВ.

На схеме нелинейное звено от  $i_{я}$  до  $k\Phi$  моделирует кривую намагничивания. Кривую намагничивания для двигателей серии Д можно аппроксимировать функцией

$$k\Phi^0(i_{я}^0) = 0,094 \cdot i_{я}^0 + 1,074 \cdot (1 - e^{-1,785 \cdot i_{я}^0}),$$

где:  $k\Phi^0$  – относительное значение  $k\Phi$

$$k\Phi^0 = \frac{k\Phi}{k\Phi_H},$$

$i_{я}^0$  – относительное значение тока якоря

$$i_{я}^0 = \frac{i_{я}}{I_H}.$$

При определении параметров якорной цепи следует иметь ввиду, что для двигателя последовательного возбуждения в них необходимо включить параметры обмотки возбуждения.

Недостающие параметры двигателя можно рассчитать по формулам (1.1)...(1.8) [1] по каталожным данным, приведенным в [2] и [3].

### 1.6. Модель асинхронного электродвигателя с короткозамкнутым ротором

В зависимости от решаемой задачи модель асинхронного электродвигателя удобно представлять в одной из трех координатных систем:  $\alpha\beta$ ,  $xу$  или  $dq$ .

В координатной системе  $\alpha\beta$  система уравнений (разрешенная относительно потокосцеплений), описывающая асинхронный электродвигатель имеет вид

$$\left\{ \begin{array}{l} \frac{d\psi_{1\alpha}}{dt} = u_{1\alpha} - \frac{R_1 \cdot L_2}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1\alpha} + \frac{R_1 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2\alpha}, \\ \frac{d\psi_{1\beta}}{dt} = u_{1\beta} - \frac{R_1 \cdot L_2}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1\beta} + \frac{R_1 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2\beta}, \\ \frac{d\psi_{2\alpha}}{dt} = -\frac{R_2 \cdot L_1}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2\alpha} + \frac{R_2 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1\alpha} - \omega_{\text{ЭЛ}} \cdot \psi_{2\beta}, \\ \frac{d\psi_{2\beta}}{dt} = -\frac{R_2 \cdot L_1}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2\beta} + \frac{R_2 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1\beta} + \omega_{\text{ЭЛ}} \cdot \psi_{2\alpha}, \\ M = \frac{p_{\text{Д}} \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot (\psi_{1\beta} \cdot \psi_{2\alpha} - \psi_{1\alpha} \cdot \psi_{2\beta}), \\ \frac{d\omega_{\text{ЭЛ}}}{dt} = \frac{p_{\text{Д}}}{J} \cdot (M - M_{\text{С}}), \end{array} \right.$$

где  $\psi_{1\alpha} \dots \psi_{2\beta}$  – потокосцепления обмоток,  
 $u_{1\alpha}, u_{1\beta}$  – напряжения на обмотках статора,  
 $M, M_{\text{С}}$  – момент двигателя и момент сопротивления,  
 $\omega_{\text{ЭЛ}}$  – электрическая угловая скорость вращения ротора

$$\omega_{\text{ЭЛ}} = \frac{\omega}{p_{\text{Д}}},$$

$R_1, L_1$  – активное сопротивление и индуктивность фазы обмотки статора,  
 $R_2, L_2$  – активное сопротивление и индуктивность фазы обмотки ротора,  
 $L_{12}$  – взаимоиндуктивность между обмотками статора и ротора,  
 $p_d$  – число пар полюсов двигателя,  
 $J$  – момент инерции ротора двигателя.  
 При этом напряжение на обмотках статора

$$\begin{cases} u_{1\alpha} = U_1 \cdot \sqrt{2} \cdot k_C \cdot \cos(\omega_{0ЭЛ} \cdot t), \\ u_{1\beta} = U_1 \cdot \sqrt{2} \cdot k_C \cdot \sin(\omega_{0ЭЛ} \cdot t), \end{cases}$$

где  $U_1$  – действующее значение напряжения,  
 $k_C$  – коэффициент согласования при переходе от трех- к двухфазной модели

$$k_C = \sqrt{\frac{3}{2}},$$

$\omega_{0ЭЛ}$  – угловая скорость вращения векторов напряжения

$$\omega_{0ЭЛ} = 2 \cdot \pi \cdot f,$$

$f$  – частота напряжения.

В осях  $xu$  модель асинхронного двигателя выглядит следующим образом

$$\left\{ \begin{aligned}
 \frac{d\psi_{1x}}{dt} &= u_{1x} - \frac{R_1 \cdot L_2}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1x} + \frac{R_1 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2x} + \omega_{0ЭЭ} \cdot \psi_{1y}, \\
 \frac{d\psi_{1y}}{dt} &= u_{1y} - \frac{R_1 \cdot L_2}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1y} + \frac{R_1 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2y} - \omega_{0ЭЭ} \cdot \psi_{1x}, \\
 \frac{d\psi_{2x}}{dt} &= -\frac{R_2 \cdot L_1}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2x} + \frac{R_2 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1x} + (\omega_{0ЭЭ} - \omega_{ЭЛ}) \cdot \psi_{2y}, \\
 \frac{d\psi_{2y}}{dt} &= -\frac{R_2 \cdot L_1}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2y} + \frac{R_2 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1y} - (\omega_{0ЭЭ} - \omega_{ЭЛ}) \cdot \psi_{2x}, \\
 M &= \frac{p_D \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot (\psi_{1y} \cdot \psi_{2x} - \psi_{1x} \cdot \psi_{2y}), \\
 \frac{d\omega_{ЭЛ}}{dt} &= \frac{p_D}{J} \cdot (M - M_C).
 \end{aligned} \right.$$

Напряжения статора

$$\begin{cases} u_{1x} = U_1 \cdot \sqrt{2} \cdot k_C, \\ u_{1y} = 0. \end{cases}$$

При записи модели в осях  $dq$  необходимо добавить уравнение для электрического угла поворота ротора  $\varphi_{ЭЛ}$



$$\left\{ \begin{array}{l}
 \frac{d\psi_{1d}}{dt} = u_{1d} - \frac{R_1 \cdot L_2}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1d} + \frac{R_1 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2d} + \omega_{\text{ЭЛ}} \cdot \psi_{1q}, \\
 \frac{d\psi_{1q}}{dt} = u_{1q} - \frac{R_1 \cdot L_2}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1q} + \frac{R_1 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2q} - \omega_{\text{ЭЛ}} \cdot \psi_{1d}, \\
 \frac{d\psi_{2d}}{dt} = -\frac{R_2 \cdot L_1}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2d} + \frac{R_2 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1d}, \\
 \frac{d\psi_{2q}}{dt} = -\frac{R_2 \cdot L_1}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{2q} + \frac{R_2 \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot \psi_{1q}, \\
 M = \frac{p_{\text{Д}} \cdot L_{12}}{L_1 \cdot L_2 - L_{12}^2} \cdot (\psi_{1q} \cdot \psi_{2d} - \psi_{1d} \cdot \psi_{2q}), \\
 \frac{d\omega_{\text{ЭЛ}}}{dt} = \frac{p_{\text{Д}}}{J} \cdot (M - M_{\text{С}}), \\
 \frac{d\varphi_{\text{ЭЛ}}}{dt} = \omega_{\text{ЭЛ}}.
 \end{array} \right.$$

Напряжения на обмотках статора при этом равны (при постоянстве  $\omega_{0\text{ЭЭ}}$ )

$$\begin{cases}
 u_{1d} = U_1 \cdot \sqrt{2} \cdot k_{\text{С}} \cdot \cos(\omega_{0\text{ЭЭ}} \cdot t - \varphi_{\text{ЭЛ}}), \\
 u_{1q} = U_1 \cdot \sqrt{2} \cdot k_{\text{С}} \cdot \sin(\omega_{0\text{ЭЭ}} \cdot t - \varphi_{\text{ЭЛ}}).
 \end{cases}$$

Параметры модели асинхронного электродвигателя с короткозамкнутым ротором можно определить по формулам (2.1)...(2.5) [1], по каталожным данным, приведенным в [4].

### **1.7. Модели систем управления скоростью двигателя постоянного тока независимого возбуждения**

На рис. 1.6 и 1.7 приведены структурные схемы одноконтурной и двухконтурной систем автоматического управления скоростью двигателя постоянного тока независимого возбуждения.

На схемах обозначено:

- $u_{\text{ЗС}}, u_{\text{ЗТ}}$  — напряжения задания скорости и тока,
- $u_{\text{ОС}}, u_{\text{ОТ}}$  — напряжения обратных связей по скорости и току,
- $u_{\text{У}}$  — напряжение управления тиристорным преобразователем,

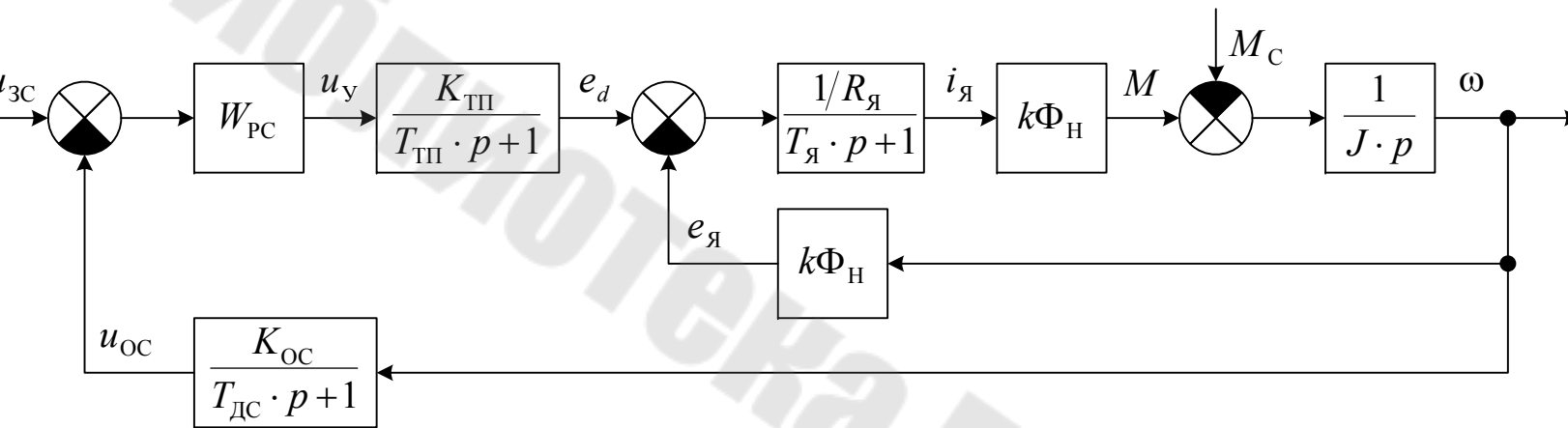


Рис. 1.6. Структурная схема одноконтурной системы автоматического управления скоростью двигателя постоянного тока независимого возбуждения.

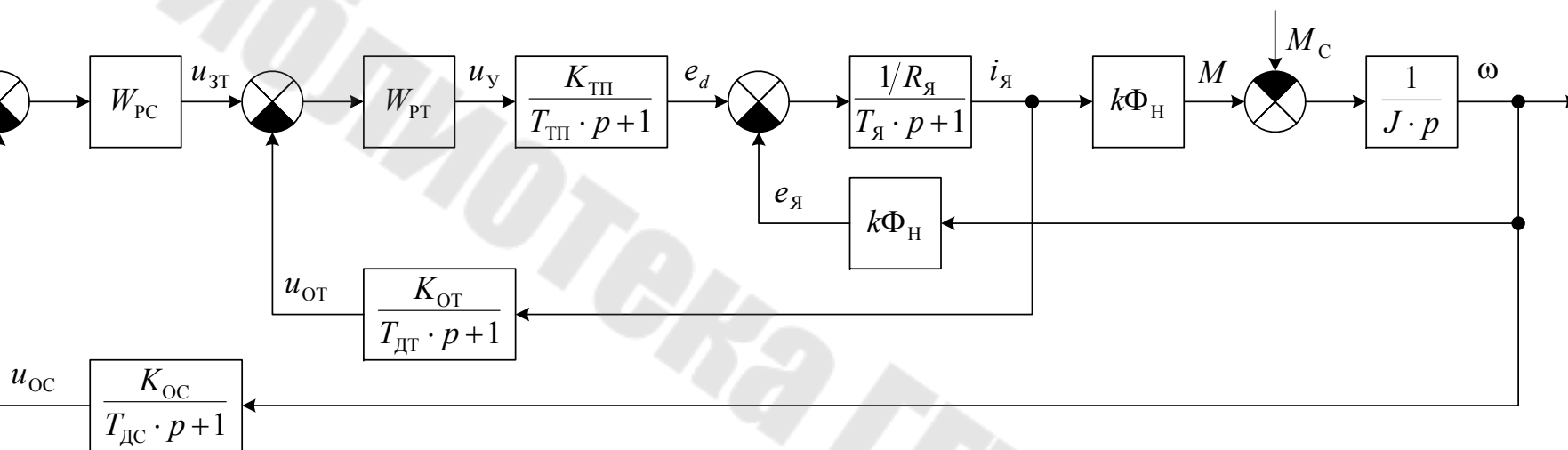


Рис. 1.7. Структурная схема двухконтурной системы автоматического управления скоростью двигателя постоянного тока независимого возбуждения

$e_d$  – ЭДС тиристорного преобразователя,  
 $W_{PC}, W_{PT}$  – передаточные функции регуляторов скорости и тока,  
 $K_{OC}, K_{OT}$  – коэффициенты передачи обратных связей по скорости и току,  
 $T_{ДС}, T_{ДТ}$  – постоянные времени датчиков скорости и тока,  
 $K_{ТП}$  – коэффициент передачи тиристорного преобразователя

$$K_{ТП} = \frac{E_{d0}}{U_{ОПМАХ}},$$

$E_{d0}$  – максимальное значение выпрямленной ЭДС преобразователя, определяется по формуле (1.17) [1],  
 $U_{ОПМАХ}$  – амплитуда опорного напряжения системы импульсно-фазового управления преобразователя,  
 $T_{ТП}$  – постоянная времени преобразователя, определяется по формуле (1.16) [1],  
 $R_{я}, T_{я}$  – активное сопротивление и постоянная времени якорной цепи с учетом параметров силовой цепи преобразователя, рассчитываются по формулам (1.9)...(1.15) [1].

При расчетах параметров силовой части системы управления потребуются каталожные данные трансформаторов, которые можно найти в [5].

### 1.8. Структурные преобразования и операции с передаточными функциями в MathCAD

Для структурных преобразований и работы с передаточными функциями необходимо пользоваться операцией **Factor (Simplify** для версий MathCAD ранее 14-ой) символьных вычислений. Порядок работы рассмотрим на примере простейшей структурной схемы, представленной на рис. 1.8.

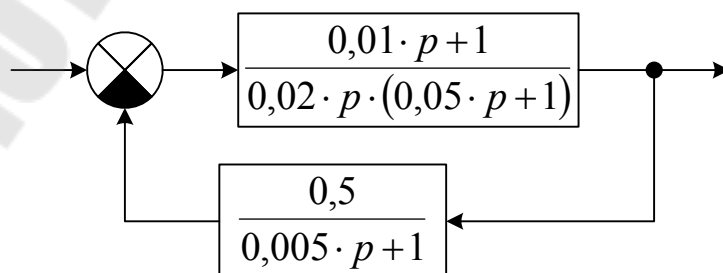


Рис. 1.8. Структурная схема для преобразования.

Передаточные функции прямого канала

$$W_{\Pi}(p) = \frac{0,01 \cdot p + 1}{0,02 \cdot p \cdot (0,05 \cdot p + 1)}$$

и обратной связи

$$W_{\text{OC}}(p) = \frac{0,5}{0,005 \cdot p + 1}$$

Для получения передаточной функции данной структуры зададим в MathCAD'е функции аргумента  $p$  для передаточных функций прямого канала и обратной связи. Затем воспользуемся правилом структурных преобразований для получения общей передаточной функции, и, используя операцию **Factor** получим окончательный результат (см. рис. 1.9).

$$W_{\Pi}(p) := \frac{0.01 \cdot p + 1}{0.02 \cdot p \cdot (0.05 \cdot p + 1)} \quad W_{\text{OC}}(p) := \frac{0.5}{0.005 \cdot p + 1}$$

$$W(p) := \frac{W_{\Pi}(p)}{W_{\Pi}(p) \cdot W_{\text{OC}}(p) + 1}$$

$$W(p) \text{ factor} \rightarrow \frac{10 \cdot (p + 200.0) \cdot (p + 100.0)}{p^3 + 220.0 \cdot p^2 + 5000.0 \cdot p + 100000.0}$$

Рис. 1.9. Фрагмент документа MathCAD'а с передаточными функциями.

В знаменателе полученной результирующей передаточной функции присутствует полином 3-ей степени. Если требуется разложить передаточную функцию на элементарные звенья, то следует воспользоваться функцией MathCAD'а **polyroots**. В качестве аргумента в данную функцию необходимо передавать матрицу-столбец коэффициентов полинома. Элементы матрицы должны содержать коэффициенты при степенях аргумента, соответствующих номерам строк (см. рис. 1.10).

$$p := \text{polyroots} \left( \begin{pmatrix} 100000. \\ 5000. \\ 220. \\ 1 \end{pmatrix} \right) \quad p = \begin{pmatrix} -197.218 \\ -11.391 - 19.424i \\ -11.391 + 19.424i \end{pmatrix}$$

Рис. 1.10. Фрагмент документа MathCAD'a с определением корней.

Для вещественного корня  $p$  постоянная времени соответствующего звена равна

$$T = -\frac{1}{p}.$$

Комплексно-сопряженные корни соответствуют звену 2-го порядка. Установим связь между корнями и параметрами звена. Обозначим:  $\alpha$  – вещественная часть,  $\beta$  – комплексная. Тогда корни  $p_1 = \alpha + j\beta$ ,  $p_2 = \alpha - j\beta$ .

$$\begin{aligned} (p - p_1) \cdot (p - p_2) &= (p - \alpha - j\beta) \cdot (p - \alpha + j\beta) = p^2 - 2 \cdot \alpha \cdot p + \alpha^2 + \beta^2 = \\ &= (\alpha^2 + \beta^2) \cdot \left( \frac{1}{\alpha^2 + \beta^2} \cdot p^2 - \frac{2 \cdot \alpha}{\alpha^2 + \beta^2} \cdot p + 1 \right). \end{aligned}$$

Сравним теперь полученное выражение с характеристическим уравнением звена второго порядка  $T^2 \cdot p^2 + 2 \cdot T \cdot \xi \cdot p + 1$  и получим

$$\begin{aligned} T &= \sqrt{\frac{1}{\alpha^2 + \beta^2}}, \\ \xi &= \frac{-2 \cdot \alpha}{\alpha^2 + \beta^2} \cdot \frac{1}{2 \cdot T} = \frac{-\alpha \cdot \sqrt{\alpha^2 + \beta^2}}{\alpha^2 + \beta^2} = -\frac{\alpha}{|p_1|}. \end{aligned}$$

Итак, определим параметры элементарных звеньев, вынеся за скобки коэффициенты при 0-ой степени  $p$  (см. рис. 1.11).

$$\begin{aligned}
 K &:= \frac{10 \cdot 200 \cdot 100}{100000} & K &= 2 \\
 T_1 &:= \frac{1}{200} & T_1 &= 5 \times 10^{-3} \\
 T_2 &:= \frac{1}{100} & T_2 &= 0.01 \\
 T_3 &:= -\frac{1}{p_0} & T_3 &= 5.071 \times 10^{-3} \\
 T_4 &:= \frac{1}{|p_1|} & T_4 &= 0.044 & \xi_4 &:= -\frac{\operatorname{Re}(p_1)}{|p_1|} & \xi_4 &= 0.506
 \end{aligned}$$

Рис. 1.11. Фрагмент документа MathCAD'a с расчетом параметров результирующей передаточной функции.

Т.о. получили передаточную функцию

$$\begin{aligned}
 W(p) &= K \cdot \frac{(T_1 \cdot p + 1) \cdot (T_2 \cdot p + 1)}{(T_3 \cdot p + 1) \cdot (T_4^2 \cdot p^2 + 2 \cdot T_4 \cdot \xi \cdot p + 1)} = \\
 &= 2 \cdot \frac{(5 \cdot 10^{-3} \cdot p + 1) \cdot (0,01 \cdot p + 1)}{(5,071 \cdot 10^{-3} \cdot p + 1) \cdot (0,044^2 \cdot p^2 + 2 \cdot 0,044 \cdot 0,506 \cdot p + 1)}.
 \end{aligned}$$

### 1.9. Построение логарифмических частотных характеристик в MathCAD

Для построения логарифмических характеристик используется оператор модуля (для амплитудной характеристики) и функция **arg** (для фазовой). Для полученной в предыдущем пункте передаточной функции

$$W(p) = 2 \cdot \frac{(5 \cdot 10^{-3} \cdot p + 1) \cdot (0,01 \cdot p + 1)}{(5,071 \cdot 10^{-3} \cdot p + 1) \cdot (0,044^2 \cdot p^2 + 2 \cdot 0,044 \cdot 0,506 \cdot p + 1)}$$

определим в MathCAD'е функции для логарифмических амплитудной и фазовой характеристик (см. рис. 1.12. Следует обратить внимание, что в аргументе передаточной функции  $i$  – мнимая единица, набираемая в MathCAD'е как **1i**).

$$A(\omega) := 20 \cdot \log(|W(i \cdot \omega)|)$$

$$\phi(\omega) := \arg(W(i \cdot \omega))$$

*Рис. 1.12. Фрагмент документа MathCAD'a с функциями для логарифмических характеристик.*

Теперь, задав дискретную переменную  $i$  и границы частоты в декадах (начальная граница – переменная  $n$ , конечная –  $k$ ) определим значения частоты  $\omega$  и построим характеристики (см. рис. 1.13).

Однако, возможен случай, когда фазовую характеристику необходимо задавать как сумму фазовых характеристик элементарных звеньев. В качестве примера уберем из рассмотренной передаточной функции форсирующие звенья.

$$W(p) = 2 \cdot \frac{1}{(5,071 \cdot 10^{-3} \cdot p + 1) \cdot (0,044^2 \cdot p^2 + 2 \cdot 0,044 \cdot 0,506 \cdot p + 1)}$$

В этом случае фазовая характеристика получится с разрывом в момент ее перехода через  $-180$  градусов, как показано на рис. 1.14 (сплошная линия). Это происходит, из-за того, что функция **arg** возвращает результат в пределах от  $-180$  до  $+180$  градусов. В подобных случаях необходимо задавать фазовую характеристику в виде суммы характеристик элементарных звеньев. Для рассматриваемого примера задана функция  $\phi_1$  на рис. 1.14.



$i := 0..100$      $n := 0$      $k := 4$      $\omega_i := 10^{\frac{i}{100} \cdot (k-n) + n}$

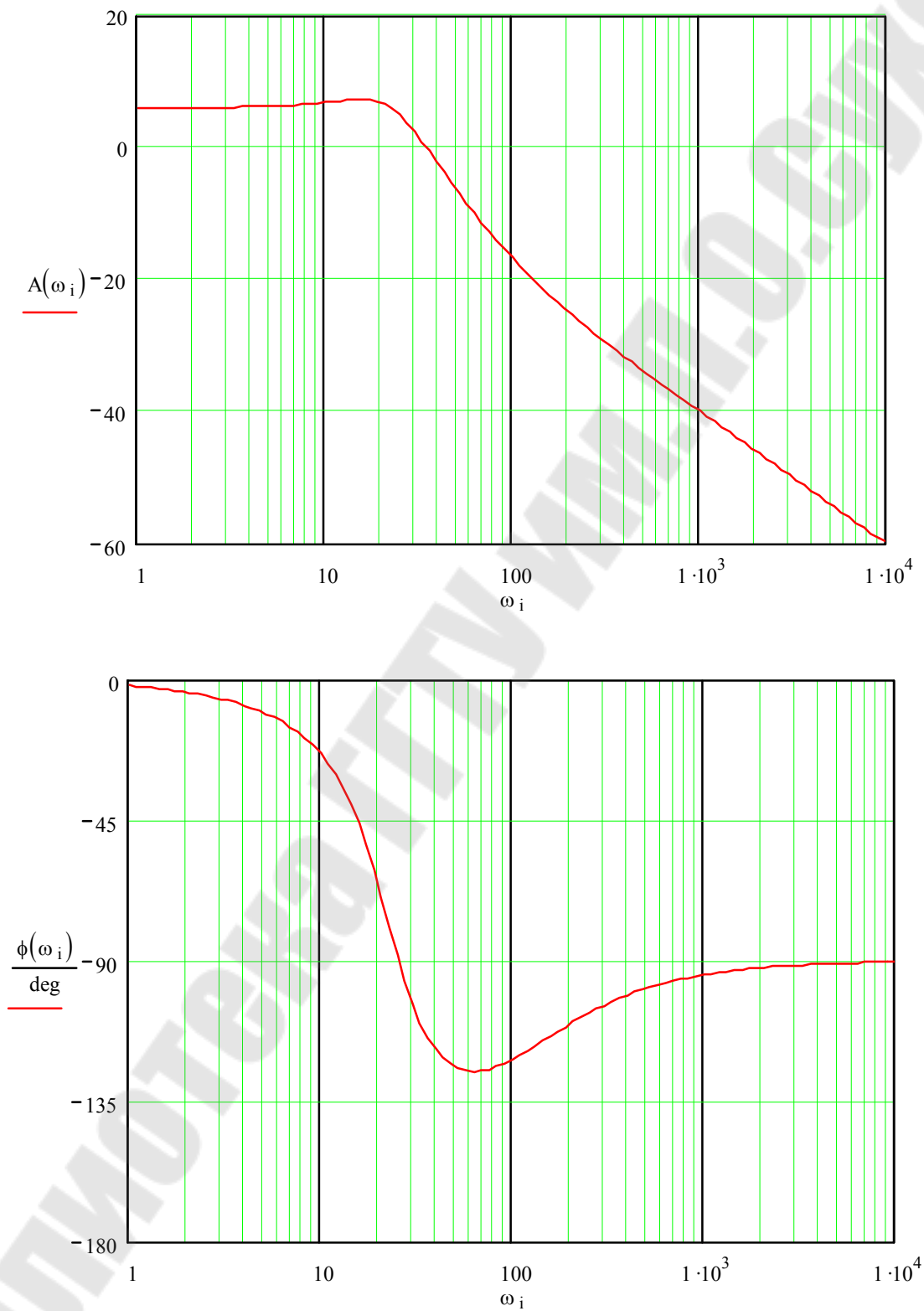


Рис. 1.13. Фрагмент документа MathCAD'a с характеристиками.

$$W(p) := 2 \cdot \frac{1}{(5.071 \cdot 10^{-3} \cdot p + 1) \cdot (0.044^2 \cdot p^2 + 2 \cdot 0.044 \cdot 0.506 p + 1)}$$

$$\phi(\omega) := \arg(W(i \cdot \omega))$$

$$\phi_1(\omega) := \arg\left(\frac{1}{T_3 \cdot i \cdot \omega + 1}\right) + \arg\left[\frac{1}{(T_4 \cdot i \cdot \omega)^2 + 2 \cdot T_4 \cdot \xi \cdot i \cdot \omega + 1}\right]$$

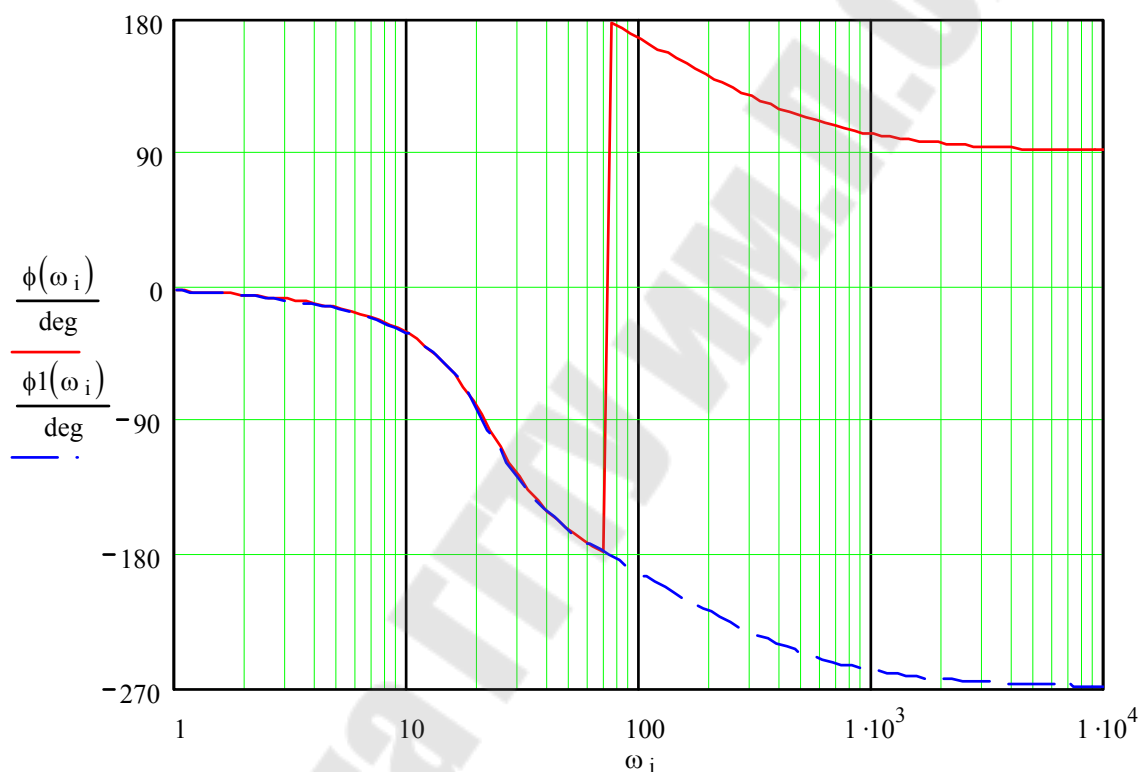


Рис. 1.14. Фрагмент документа MathCAD'a с фазовыми характеристиками.

### 1.10. Расчет переходных процессов

Для расчета переходных процессов можно воспользоваться в MathCAD'e командой символьческих вычислений для обратного преобразования Лапласа – **Inverse Laplace Transform**. Пример для ПИД-звена с замедлением показан на рис. 1.15. В данном примере модификатор символьных вычислений **float** используется для ограничения числа значащих цифр до четырех.

Выражение перед преобразованием MathCAD раскладывает на элементарные дроби со знаменателем только первого порядка, поэтому при наличии апериодического звена второго порядка в преобразованном выражении будет присутствовать мнимая единица.

$$T_1 := 0.01 \quad T_2 := 0.2 \quad T_3 := 0.005$$

$$W(p) := \frac{(T_1 \cdot p + 1) \cdot (T_2 \cdot p + 1)}{T_1 \cdot p \cdot (T_3 \cdot p + 1)}$$

$$\frac{W(p)}{p} \left| \begin{array}{l} \text{invlaplace, } p \\ \text{float, 4} \end{array} \right. \rightarrow 100 \cdot t + 20.50 + 19.50 \exp(-200 \cdot t)$$

$$f(t) := 100 \cdot t + 20.50 + 19.50 \exp(-200 \cdot t) \quad t := 0, 0.001.. 0.1$$

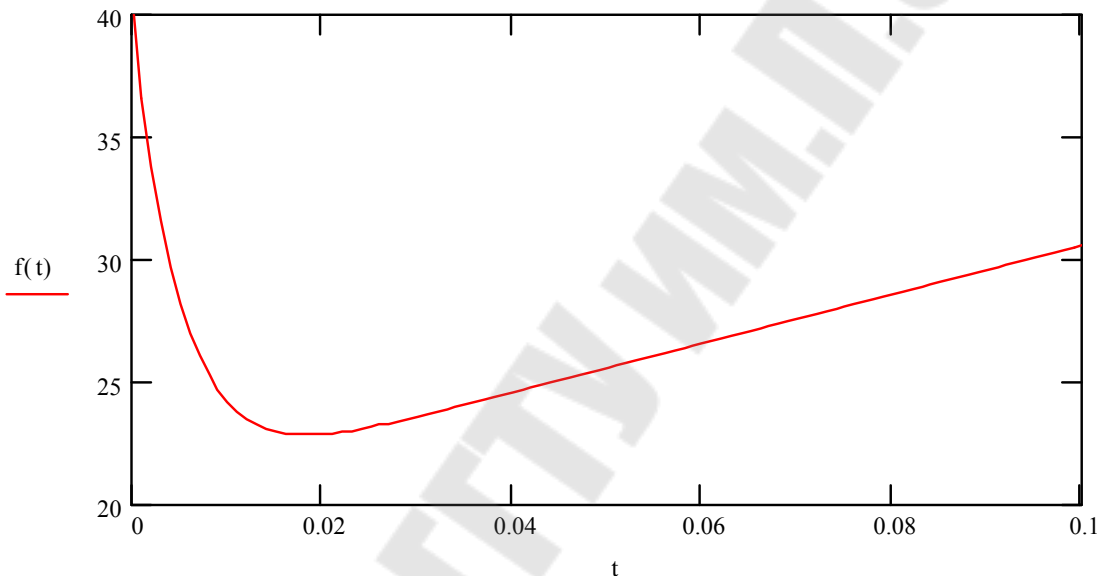


Рис. 1.15. Определение переходной характеристики ПИД-звена с замедлением.

В MathCAD'е рассчитывать переходные процессы можно также численно решая системы дифференциальных уравнений с помощью встроенных функций и с помощью средств программирования.

Рассмотрим встроенную функцию **rkfixed**, предназначенную для решения систем однородных дифференциальных уравнений, записанных в форме Коши. Она записывается следующим образом

$$\text{rkfixed}(y, x1, x2, \text{npoints}, D),$$

- где:  $y$  – матрица-столбец начальных условий;  
 $x1$  – начальное значение аргумента (времени);  
 $x2$  – конечное значение аргумента (времени);  
 $\text{npoints}$  – количество шагов расчета;  
 $D$  – матрица-столбец правых частей уравнений.

Рассмотрим на примере простейшей системы, состоящей из 2 апериодических звеньев, представленной на рис. 1.16.

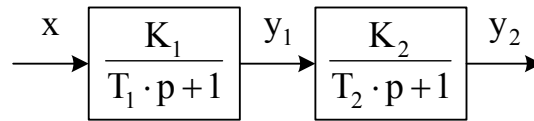


Рис. 1.16. Структурная схема.

Запишем систему уравнений

$$\begin{cases} y_1 = \frac{K_1}{T_1 \cdot p + 1} \cdot x, \\ y_2 = \frac{K_2}{T_2 \cdot p + 1} \cdot y_1. \end{cases}$$

Преобразуем ее

$$\begin{cases} p \cdot y_1 = \frac{K_1 \cdot x - y_1}{T_1}, \\ p \cdot y_2 = \frac{K_2 \cdot y_1 - y_2}{T_2}. \end{cases}$$

И, перейдя к функциям времени, получим систему дифференциальных уравнений в нормальной форме Коши

$$\begin{cases} \frac{dy_1}{dt} = \frac{K_1 \cdot x - y_1}{T_1}, \\ \frac{dy_2}{dt} = \frac{K_2 \cdot y_1 - y_2}{T_2}. \end{cases}$$

Поскольку в системе 2 уравнения, то функция-матрица для правых частей должна содержать 2 строки. Аргументами этой функции являются 2 переменные: время и матрица-столбец значений переменных. Поскольку по умолчанию в MathCAD'е нумерация элементов начинается с 0, то  $y_1$  будет  $y_0$ , а  $y_2 - y_1$ .

На рис. 1.17 представлен документ MathCAD'а. Для задания скачка на входе системы используется функция **if**. Первый аргумент функции – условие, второй – значение, которое должна принять функция при выполнении условия, третий – значение, если условие не выполняется.

$$K_1 := 2 \quad K_2 := 3 \quad T_1 := 0.001 \quad T_2 := 0.002$$

$$x(t) := \text{if}(t \geq 0, 1, 0)$$

$$D(t, y) := \begin{pmatrix} \frac{K_1 \cdot x(t) - y_0}{T_1} \\ \frac{K_2 \cdot y_0 - y_1}{T_2} \end{pmatrix} \quad y := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$a := \text{rkfixed}(y, 0, 0.01, 1000, D)$$

$$t := a \langle 0 \rangle \quad y_1 := a \langle 1 \rangle \quad y_2 := a \langle 2 \rangle$$

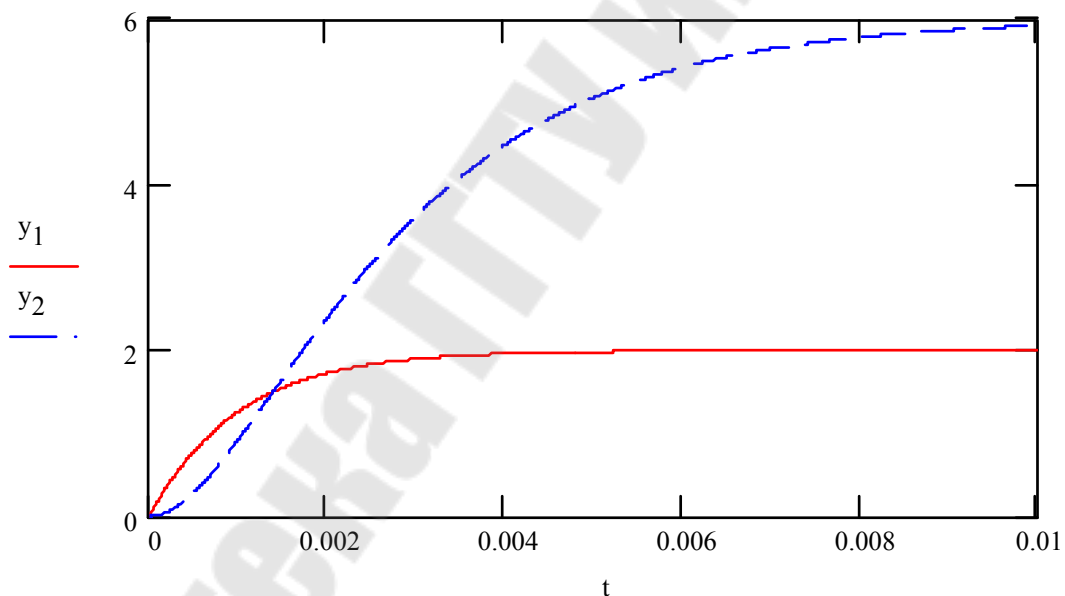


Рис. 1.17. Пример использования функции **rkfixed**.

В MathCAD'е программные блоки выделяются жирной вертикальной линией и состояются из операторов, находящихся на панели набора программирования. Программный блок может быть как самостоятельным выражением, без аргументов, так и телом определения функции.

Набор программных элементов для создания программных блоков весьма ограничен и содержит следующие элементы:

1. **Add Line** – создает и при необходимости расширяет жирную вертикальную линию, справа от которой в шаблонах задается запись программного блока.
2.  $\leftarrow$  – символ локального присваивания (в теле модуля).
3. **if** – оператор условного выражения.
4. **for** – оператор задания цикла с фиксированным числом повторений.
5. **while** – оператор задания цикла типа «пока» (цикл выполняется, пока выполняется некоторое условие).
6. **otherwise** – оператор иного выбора (обычно применяется с **if**).
7. **break** – оператор прерывания.
8. **continue** – оператор продолжения.
9. **return** – оператор возврата.
10. **on error** – оператор обработки ошибок.

Оператор **Add Line** выполняет функции добавления строки в программный блок. При этом удлиняется вертикальная черта блока. Благодаря этому оператору, в принципе, можно создавать сколь угодно большие программы.

Оператор  $\leftarrow$  выполняет функции внутреннего локального присваивания. Например, выражение  $x \leftarrow 123$  присваивает переменной  $x$  значение 123. Локальный характер присваивания означает, что такое значение  $x$  сохраняет только в теле программы. За пределами тела программы значение переменной  $x$  может быть не определенным, либо равно значению, которое задается операторами присваивания вне программного блока.

Оператор **if** является оператором для создания условных выражений. Он задается в виде:

Выражение **if** Условие

Если Условие выполняется, то возвращается значение Выражения, Совместно с этим оператором часто используются операторы прерывания **break** и оператор иного выбора **otherwise**.

Оператор **for** служит для организации циклов с заданным числом повторений. Он записывается в виде:

**for** Переменная  $\in$  min .. max

Эта запись означает, что тело цикла будет выполнено для каждого значения Переменной в интервале от min до max. Переменную-счетчик можно использовать в выражениях программы.

$$\left| \begin{array}{l} \text{for } i \in 0..2 \\ \quad a_i \leftarrow i \\ \text{return } a \end{array} \right. = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \blacksquare$$

Оператор **while** служит для организации циклов, действующих до тех пор, пока выполняется некоторое условие. Этот оператор записывается в виде:

#### **while** Условие

Выполняемое выражение записывается на место шаблона под оператором.

Оператор **otherwise** (“иначе”) используется совместно с оператором **if**. Его использование поясняет следующая программная конструкция:

$$\text{sign}(x) := \left| \begin{array}{l} 1 \text{ if } x > 0 \\ -1 \text{ otherwise} \end{array} \right.$$

$$\text{sign}(-3) = -1 \qquad \text{sign}(2) = 1$$

Оператор **break** вызывает прерывание работы блока всякий раз, как он встречается. Чаще всего он используется совместно с оператором условного выражения **if** и операторами циклов **while** и **for**, обеспечивая переход в конец тела цикла.

Оператор продолжения используется для перехода к следующей итерации цикла.

Особый оператор-функция **return** прерывает выполнение программы и возвращает значение своего операнда, стоящего следом за ним.

Приведем пример программного блока для реализации метода Эйлера и его использование для системы, представленной на рис. 1.16. Соответствующий документ MathCAD’a представлен на рис. 1.18.

$K_1 := 2 \quad K_2 := 3 \quad T_1 := 0.001 \quad T_2 := 0.002$

$x(t) := \text{if}(t \geq 0, 1, 0)$

$$D(t, y) := \begin{pmatrix} \frac{K_1 \cdot x(t) - y_0}{T_1} \\ \frac{K_2 \cdot y_0 - y_1}{T_2} \end{pmatrix}$$

$t_n := 0 \quad t_k := 0.01 \quad \Delta t := 0.0005 \quad N := \frac{t_k - t_n}{\Delta t}$

Нулевые НУ

```

a :=
| y0,0 ← 0
| y0,1 ← 0
| t0 ← 0
| for i ∈ 1.. N
|   | ti ← i·Δt
|   | x ← x(ti)
|   | for j ∈ 0.. 1
|   |   | zj ← yi-1,j
|   |   | f ← D(ti, z)
|   |   | for j ∈ 0.. 1
|   |     | yi,j ← yi-1,j + fj·Δt
|   | b<0> ← t
|   | for j ∈ 0.. 1
|   |   | b<j+1> ← y<j>
| return b

```

Цикл по шагам расчета

Текущее время

Значение входного воздействия

Значения правых частей

Реализация метода Эйлера

Формирование матрицы-результата

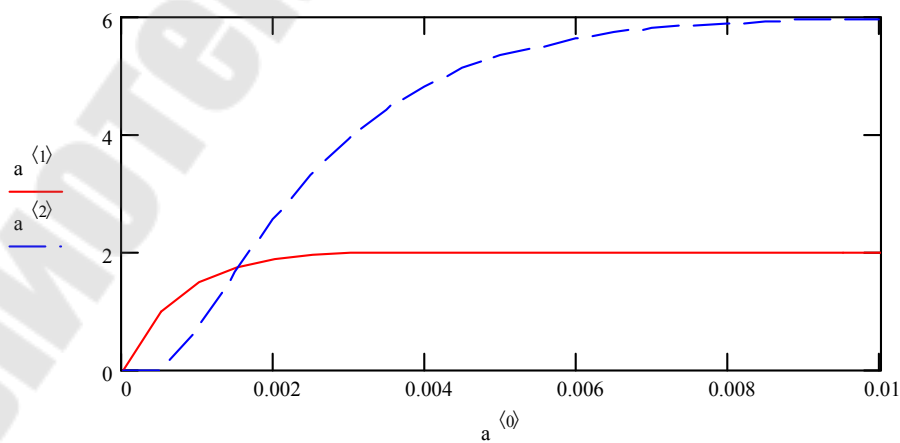



Рис. 1.18. Расчет переходных процессов в блоке программирования.



## 1.11. Основы работы с Simulink

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

Для запуска Simulink на панели инструментов MATLAB предназначена кнопка . При этом на экране появляется обозреватель библиотеки блоков, показанный на рис. 1.19.

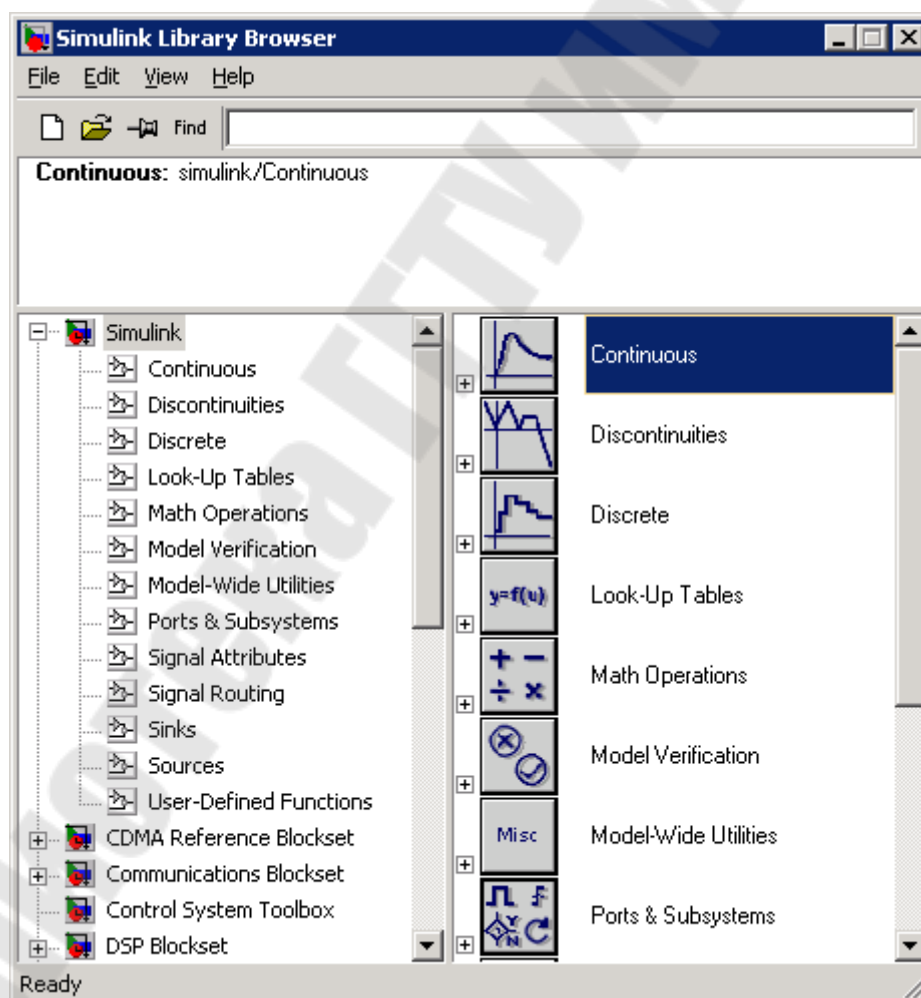


Рис. 1.19. Окно обозревателя блоков библиотеки Matlab Simulink.

Окно обозревателя библиотеки блоков содержит следующие элементы (кроме стандартных для любой программы: заголовка, меню, панели инструментов и строки состояния):

1. Окно комментария для вывода поясняющего сообщения о выбранном блоке в верхней части окна.
2. Список разделов библиотеки, реализованный в виде дерева в левой половине окна.
3. Окно содержимого раздела библиотеки (список вложенных разделов библиотеки или блоков) в правой половине окна.

Библиотека Simulink содержит следующие основные разделы:

4. Continuous – линейные блоки.
5. Discontinuities – нелинейные блоки.
6. Discrete – дискретные блоки.
7. Look-Up Tables – блоки с регулировочными характеристиками, заданными таблицей.
8. Math Operations – блоки математических операций.
9. Sinks - регистрирующие устройства.
10. Sources — источники сигналов и воздействий.

Список разделов библиотеки Simulink представлен в виде дерева, и правила работы с ним являются общими для списков такого вида. Пиктограмма свернутого узла дерева содержит символ «+», а пиктограмма развернутого содержит символ «-». Для того чтобы развернуть или свернуть узел дерева, достаточно щелкнуть на его пиктограмме левой клавишей мыши. При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое, как показано на рис. 1.20.

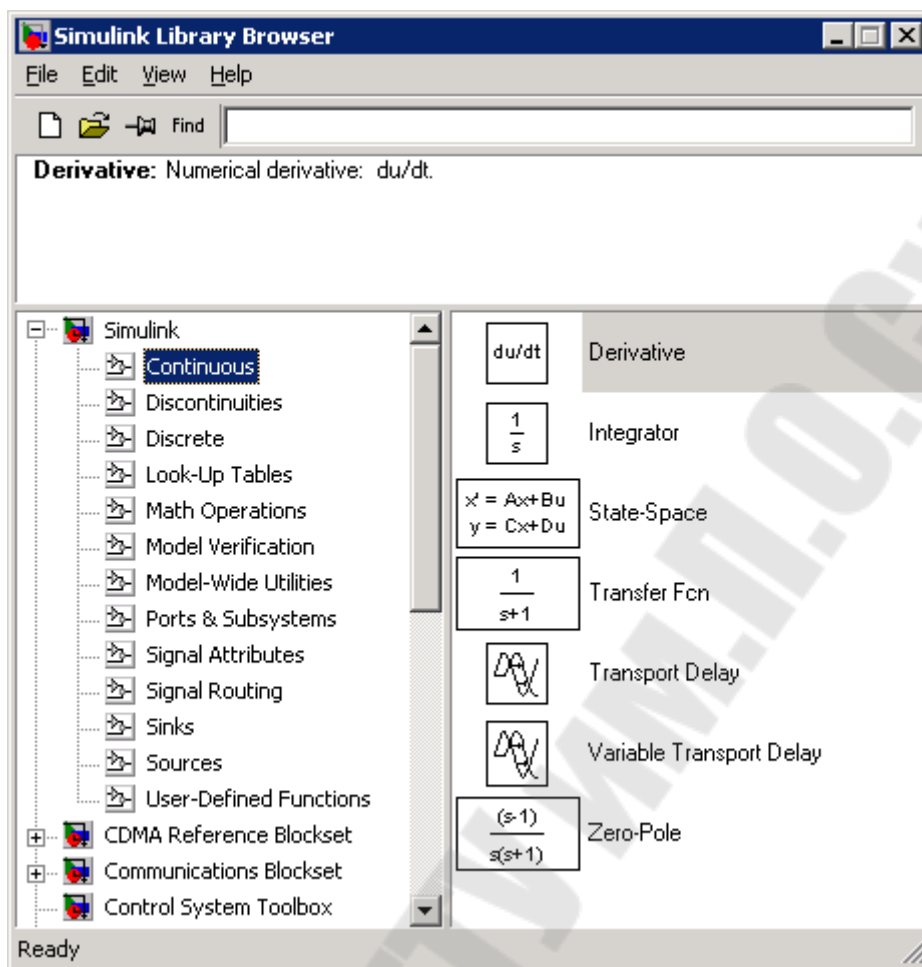



Рис. 1.20. Блоки библиотеки линейных элементов.

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

1. **File** (Файл) — работа с файлами библиотек и моделей.
2. **Edit** (Редактирование) — добавление блоков и их поиск (по названию).
3. **View** (Вид) — управление показом элементов интерфейса.
4. **Help** (Справка) — вывод окна справки по обозревателю библиотек.

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

1. Создать новый файл модели с помощью команды **File/New/Model**, или используя кнопку  на панели инструментов (здесь и далее, с помощью символа «/», указаны пункты меню программы, которые необходимо последовательно выбрать для выполнения указанного действия). Вновь созданное окно модели показано на рис 1.21.

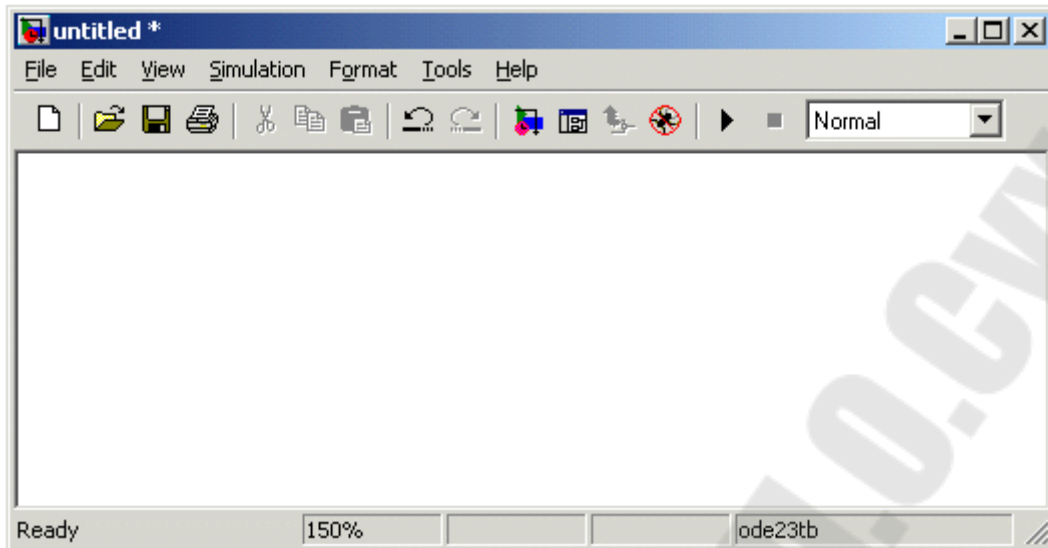


Рис. 1.21. Окно редактора модели.

2. Расположить блоки в окне модели. Для этого необходимо открыть соответствующий раздел библиотеки (Например, Sources - Источники). Далее, указав курсором на требуемый блок и нажав на левую клавишу «мыши» - «перетащить» блок в созданное окно. Клавишу мыши нужно держать нажатой. На рис. 1.22 показано окно модели, содержащее блоки.

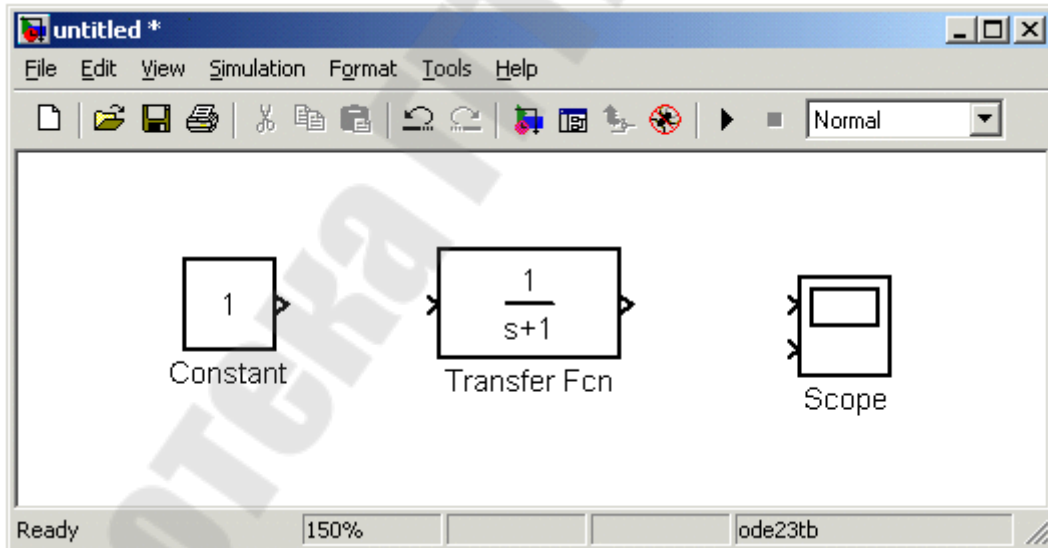


Рис. 1.22. Окно модели с блоками.

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

Для изменения размеров блока требуется выбрать блок, установить курсор в один из углов блока и, нажав левую клавишу

«мыши», изменить размер блока (курсор при этом превратится в двухстороннюю стрелку).

3. Далее, если это требуется, нужно изменить параметры блока, установленные программой «по умолчанию». Для этого необходимо дважды щелкнуть левой клавишей «мыши», указав курсором на изображение блока. Откроется окно редактирования параметров данного блока. При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая. После внесения изменений нужно закрыть окно кнопкой ОК. На рис. 1.23 в качестве примера показаны блок, моделирующий передаточную функцию и окно редактирования параметров данного блока.

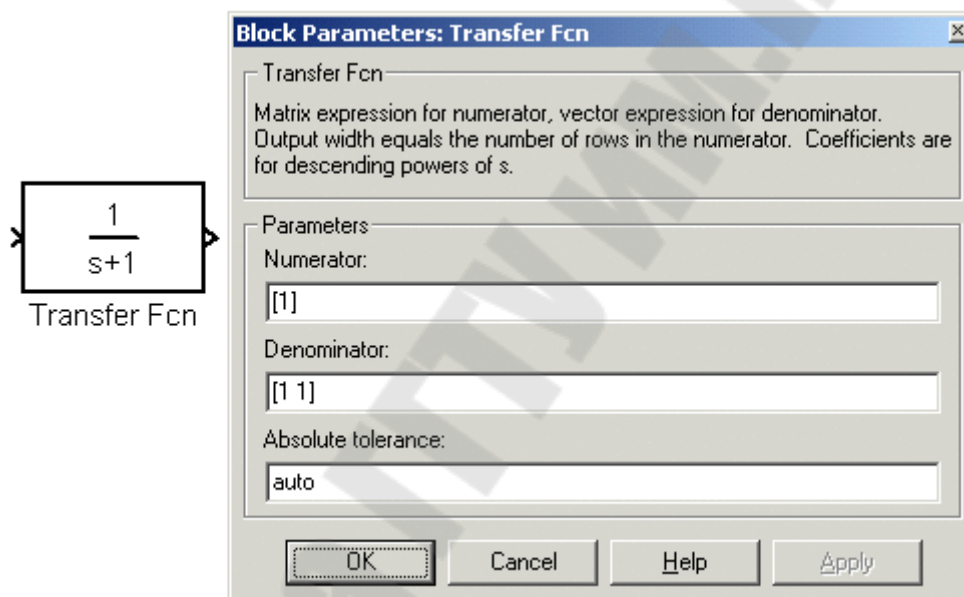


Рис. 1.23. Блок «Передаточная функция» и окно его параметров.

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, нажать и, не отпуская левую клавишу «мыши», провести линию к входу другого блока. После чего отпустить клавишу. В случае правильного соединения изображение курсора меняется. 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. 1.24.

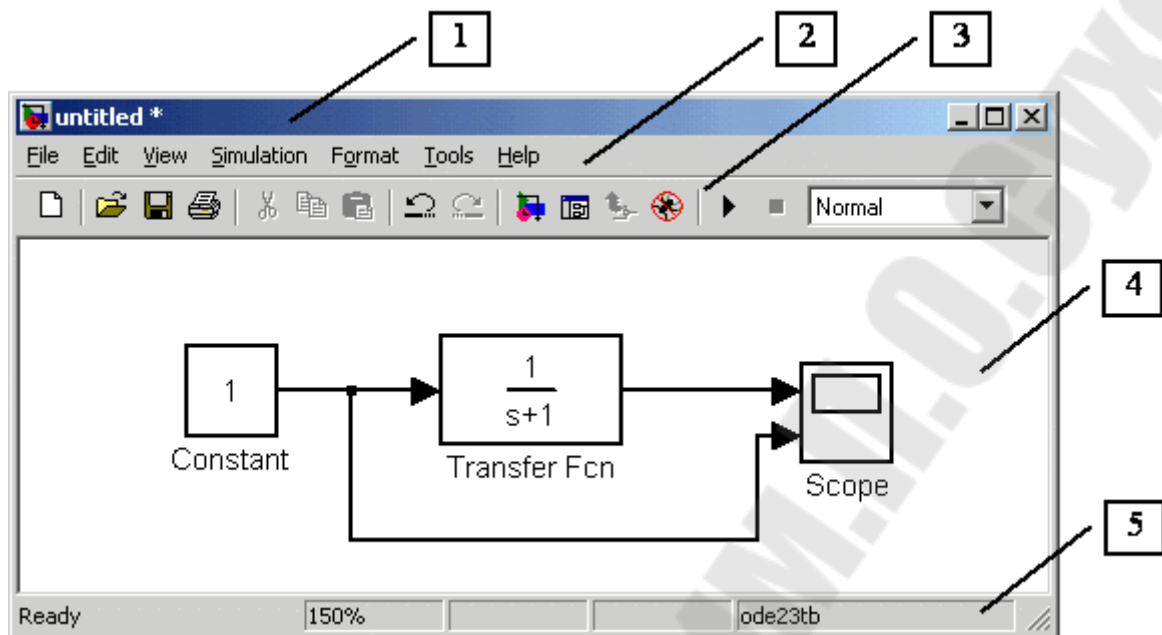


Рис. 1.24. Окно модели с составленной моделью.

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. Следует иметь в виду, что имя файла не должно превышать 32 символов, должно начинаться с буквы и не может содержать символы кириллицы и спецсимволы. Это же требование относится и к пути файла (к тем папкам, в которых сохраняется файл). При последующем редактировании схемы можно пользоваться пунктом меню **File/Save**. При повторных запусках программы Simulink загрузка схемы осуществляется с помощью меню **File/Open...** в окне обозревателя библиотеки или из основного окна Matlab.

Окно модели содержит следующие элементы (рис. 1.25):

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

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

1. **File** (Файл) — Работа с файлами моделей.
2. **Edit** (Редактирование) — Изменение модели и поиск блоков.
3. **View** (Вид) — Управление показом элементов интерфейса.
4. **Simulation** (Моделирование) — Задание настроек для моделирования и управление процессом расчета.
5. **Format** (Форматирование) — Изменение внешнего вида блоков и модели в целом.
6. **Tools** (Инструментальные средства) — Применение специальных средств для работы с моделью (отладчик, линейный анализ и т.п.).
7. **Help** (Справка) — Вывод окон справочной системы.

Для работы с моделью можно также использовать кнопки на панели инструментов, показанной на рис. 1.25.



Рис. 1.25. Панель инструментов редактора моделей.

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 обмена.
6. Copy — скопировать выделенную часть модели в буфер обмена.
7. Paste — вставить в окно модели содержимое буфера обмена.
8. Undo — отменить предыдущую операцию редактирования.
9. Redo — восстановить результат отмененной операции редактирования.
10. 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).
11. Stop — закончить моделирование. Кнопка становится доступной после начала моделирования, а также после выполнения команды Pause.
12. Library Browser — открыть окно обозревателя библиотек.
13. Toggle Model Browser — открыть окно обозревателя модели.

14. Go to parent system — переход из подсистемы в систему высшего уровня иерархии («родительскую систему»). Команда доступна только, если открыта подсистема.

15. Debug — запуск отладчика модели.

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).

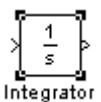
## 1.12. Обзор наиболее часто используемых блоков MATLAB Simulink

### 1.12.1. Библиотека линейных элементов – Continuous



Блок **Derivative** – производная.

Данный блок выполняет численное дифференцирование входного сигнала. Параметры блока отсутствуют.



Блок **Integrator** – интегратор.

Блока выполняет интегрирование входного сигнала. Диалоговое окно параметров блока приведено на рис. 1.26.



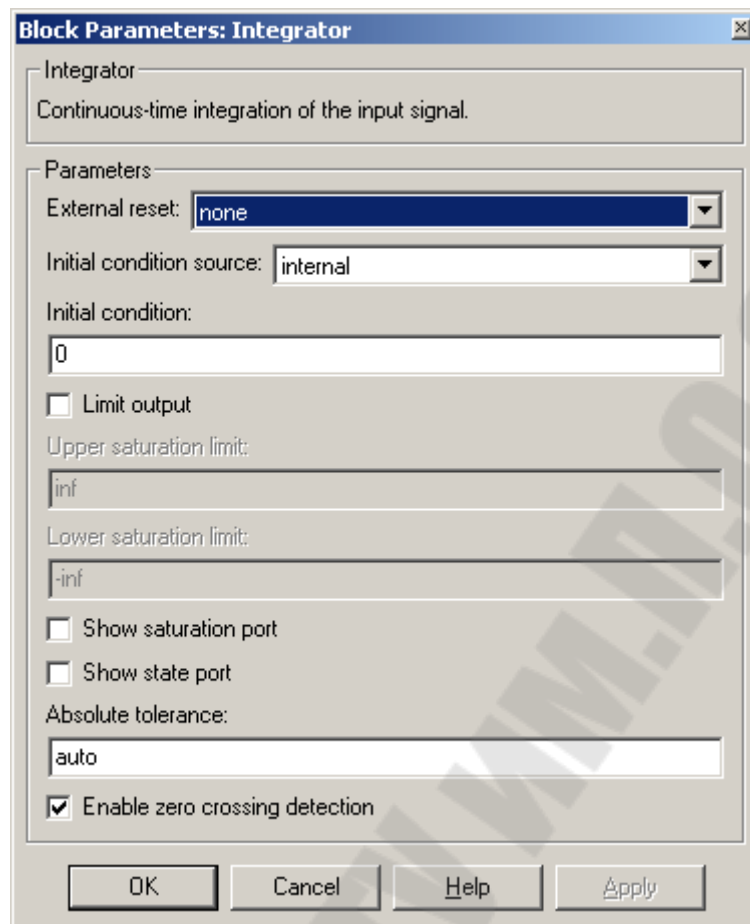
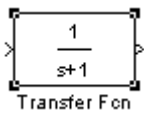


Рис. 1.26. Окно параметров интегратора.

Параметры интегратора:

- External reset – внешний сброс (тип дополнительного входного сигнал, служащего для сброса интегратора в исходное состояние);
- Initial condition source – источник начального значения выходного сигнала блока: внутренний – internal или внешний – external (при выборе внешнего у блока появляется дополнительный вход  $x_0$ );
- Initial condition – начальное значение выходной переменной при внутреннем источнике;
- Limit output – ограничение выходного сигнала;
- Upper saturation limit – верхний уровень ограничения (число или текстовое значение «inf» для бесконечности);
- Lower saturation limit – нижний уровень ограничения (число или текстовое значение «-inf» для минус бесконечности);
- Show saturation port – отображение выходного порта, показывающего находится ли интегратор в ограничении;
- Show state port – отображение выходного порта, показывающего состояние интегратора;
- Absolute tolerance – абсолютная погрешность;

- Enable zero crossing detection – определение момента перехода через ноль для методов решения с переменным шагом.



Блок **Transfer Fcn** – передаточная функция.

Блок передаточной функции предназначен для задания передаточной функции в виде отношения полиномов

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

Причем, на порядок полиномов накладывается ограничение  $m \geq n$ .  
Диалоговое окно параметров блока представлено на рис. 1.27.

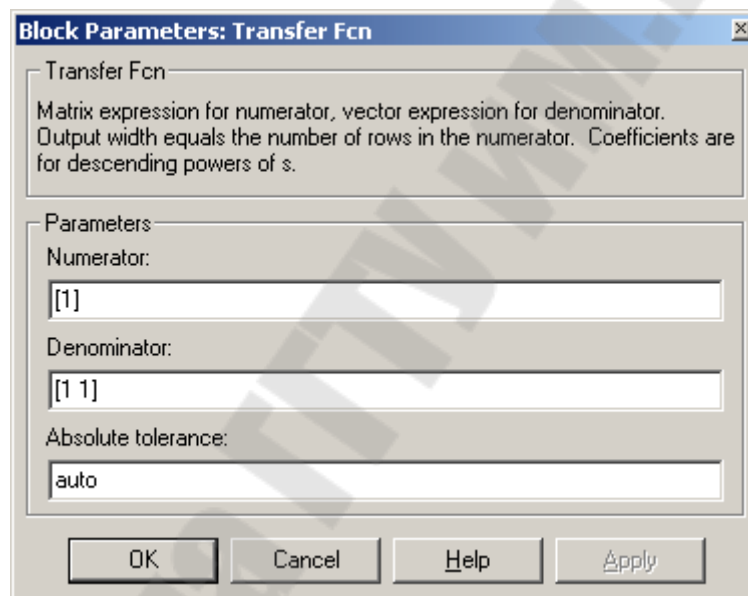


Рис. 1.27. Окно параметров блока передаточной функции.

Параметры блока:

- Numerator – коэффициенты полинома – числителя передаточной функции в порядке убывания степени оператора Лапласа;
- Denominator – коэффициенты полинома – знаменателя передаточной функции в порядке убывания степени оператора Лапласа;
- Absolute tolerance – абсолютная погрешность.

Например, для моделирования передаточной функции

$$\frac{0,01 \cdot p + 1}{0,01 \cdot p^2 + 0,02 \cdot p + 1}$$

в окне параметров блока необходимо ввести числа, как показано на рис. 1.28.

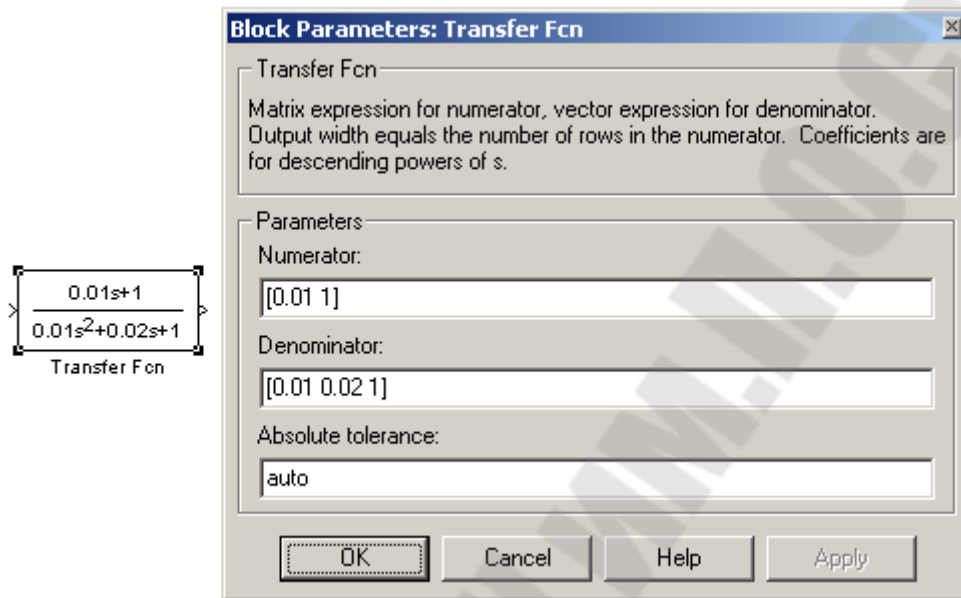


Рис. 1.28. Пример использования блока передаточной функции.

### 1.12.2. Библиотека математических операций – Math Operations



Блок **Gain** – усилитель.

Блок выполняет умножение входного сигнала на постоянный коэффициент. Диалоговое окно параметров блока представлено на рис. 1.29.

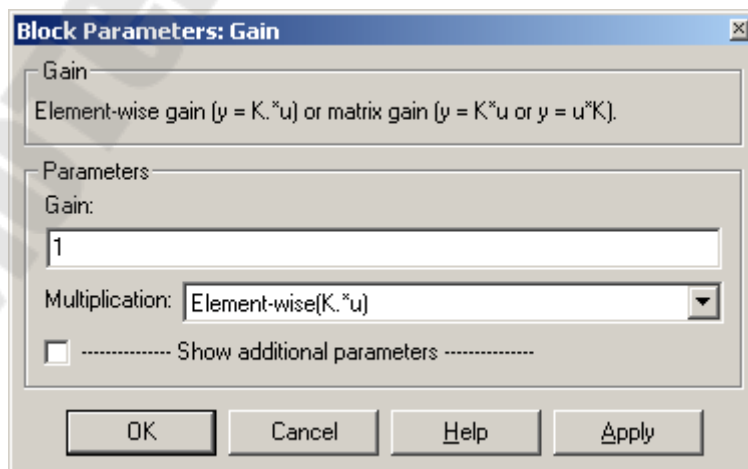


Рис. 1.29. Окно параметров блока усилителя.

Параметры блока:

- Gain – коэффициент передачи усилителя (может быть числом, вектором или матрицей);
- Multiplication – способ выполнения операции умножения – поэлементный (Element-wise) или матричный (Matrix);
- Show additional parameters – показать дополнительные параметры (тип данных коэффициента, параметры обработки цифровых сигналов).



Блок **Sum** – сумматор.

Предназначен для выполнения операций сложения и вычитания. Окно параметров блока представлено на рис. 1.30.

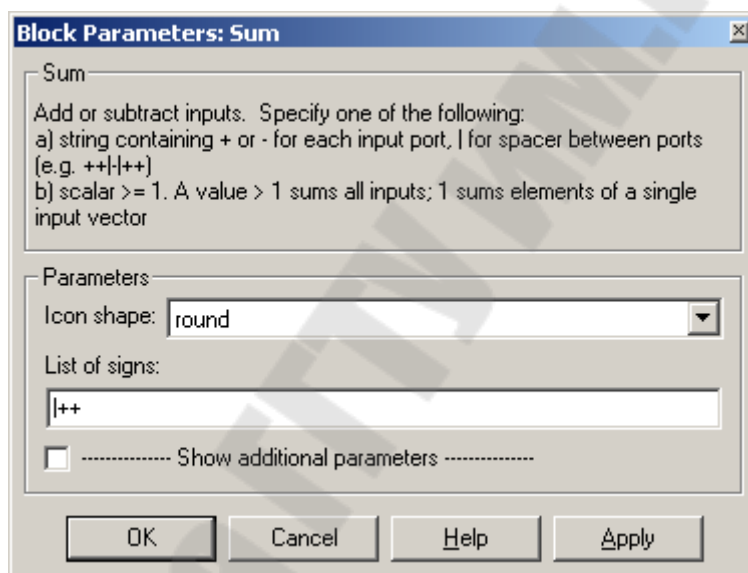
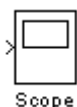


Рис. 1.30. Окно параметров сумматора.

Параметры блока:

- Icon shape – отображение сумматора кругом (round) или прямоугольником (rectangular);
- List of signs – список знаков, соответствующих входам блока (количество входов определяется количеством заданных символов “+” и “-”, вертикальная черта служит для задания дополнительного расстояния между входами) или число входов (в этом случае все входы – суммирующие);
- Show additional parameters – дополнительные параметры (типы данных и параметры обработки цифровых сигналов).

### 1.12.3. Библиотека приемников сигналов – Sinks



Блок **Scope** – осциллограф.

Блок предназначен для построения диаграмм исследуемых сигналов и наблюдать за ними в процессе моделирования. При двойном щелчке мышкой на блоке появляется окно, показанное на рис. 1.31.

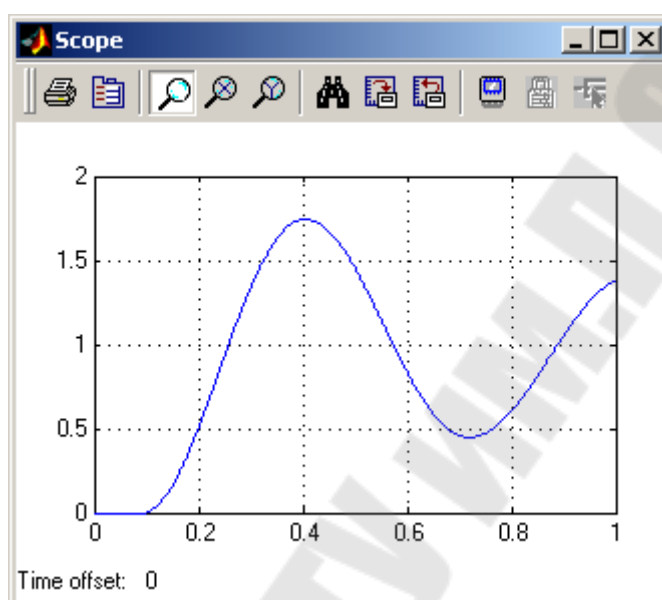


Рис. 1.31. Окно осциллографа.

Панель инструментов окна осциллографа показана на рис. 1.32.



Рис. 1.32. Панель инструментов осциллографа.

Назначение кнопок панели инструментов следующее:

1. Print – печать осциллограмм.
2. Parameters – открыть окно изменения параметров осциллографа.
3. Zoom – увеличение масштаба.
4. Zoom x-axis – увеличение масштаба по оси x.
5. Zoom y-axis – увеличение масштаба по оси y.
6. Autoscale – автоматическая установка масштаба.
7. Save current axes settings – сохранение текущих параметров окна.
8. Restore saved axes settings – установка ранее сохраненных параметров окна.
9. Floating scope – перевод осциллографа в “свободный” режим.

10. Lock/Unlock axes selection – закрепить/разорвать связь координатной системы и отображаемого сигнала (доступно в “свободном” режиме).
11. Signal selection – выбор сигналов для отображения (доступно в “свободном” режиме).

При нажатии кнопки параметров появляется окно, показанное на рис. 1.33.

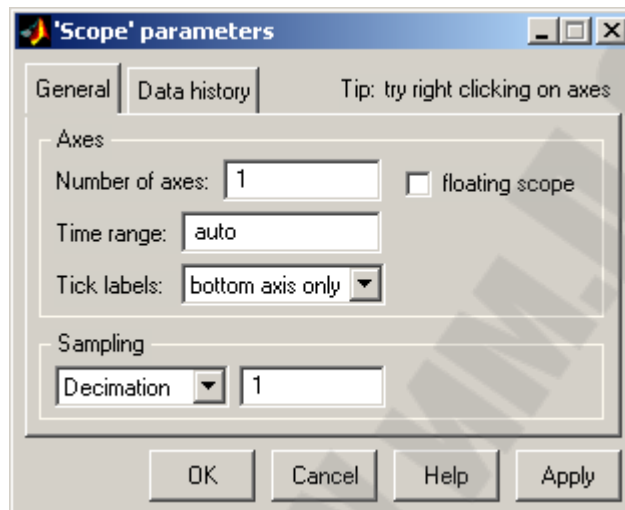


Рис. 1.33. Окно параметров осциллографа.

Параметры осциллографа:

- Number of axes – количество входов блока и диаграмм;
- Time range – временной интервал для отображения диаграмм;
- Tick labels – отображение меток осей (all – для всех осей, none – без меток, bottom axis only – метки для оси времени отображаются только для нижней диаграммы);
- floating scope – включить/выключить “свободный” режим;
- Sampling – параметры отбора точек для построения диаграмм.



Блок **XY Graph** – построитель графика.

Блок предназначен для построения зависимостей вида  $y(x)$ . Имеет 2 входа. Верхний вход – аргумент (откладываемый на оси  $x$ ), нижний – функция (откладываемый на оси  $y$ ). Окно параметров блока приведено на рис. 1.34.

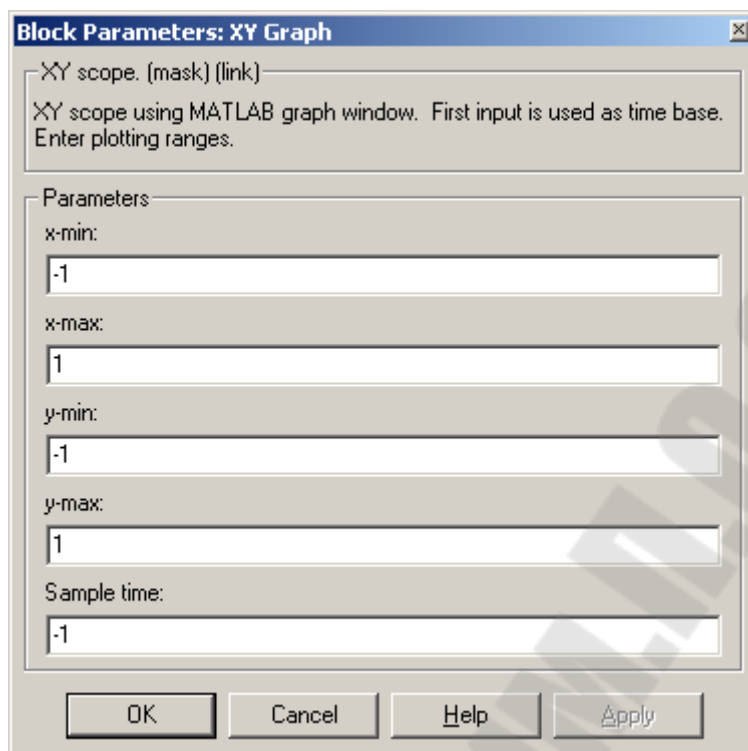


Рис. 1.34. Окно параметров блока построения графика.

Параметры блока:

- x-min, x-max – минимальное и максимальное значение аргумента;
- y-min, y-max – минимальное и максимальное значение функции;
- sample time – шаг дискретизации.

На рис. 1.35. приведен пример построения динамической характеристики двухконтурной системы автоматического управления скоростью электродвигателя постоянного тока с учетом ограничения регуляторов. По оси x отложен ток якоря, по y – скорость.

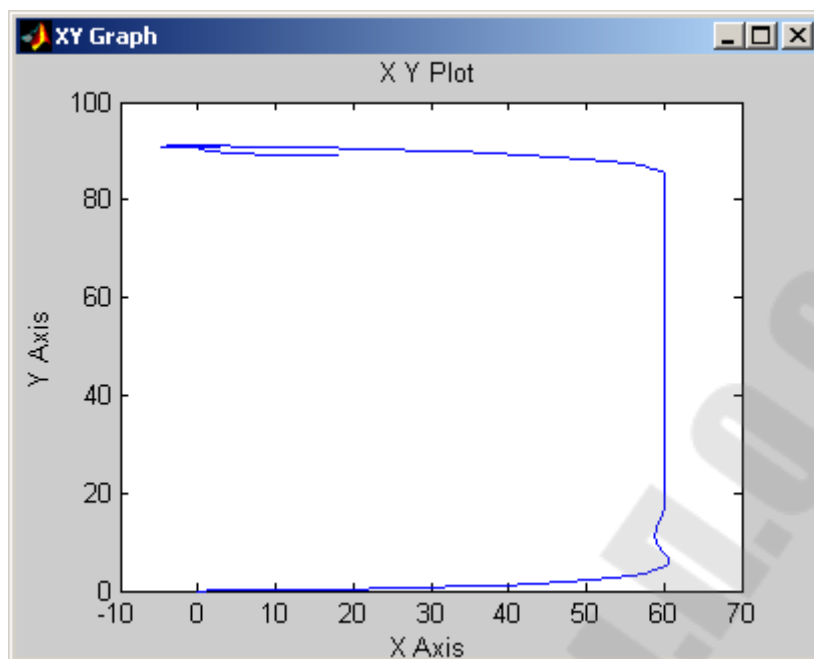


Рис. 1.35. Динамическая характеристика электропривода постоянного тока.

#### 1.12.4. Библиотека источников сигналов – Sources



Блок **Step** – скачек.

Блок предназначен для подачи на вход моделируемой системы скачка. Вид диалогового окна параметров блока представлен на рис. 1.36.



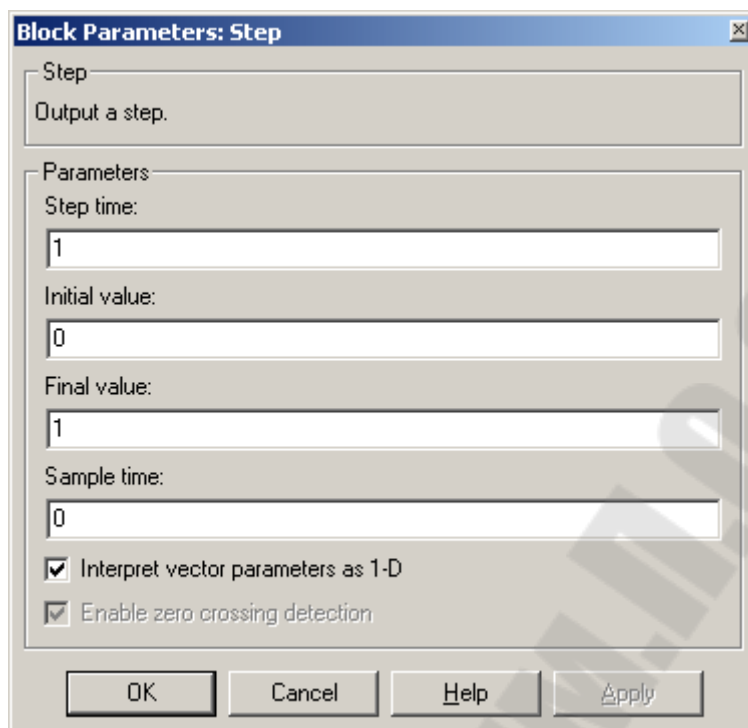


Рис. 1.36. Параметры блока скачкообразного воздействия.

Параметры блока:

- Step time – момент времени, когда происходит скачкообразное изменение сигнала;
- Initial value – значение сигнала до скачка;
- Final value – значение сигнала после скачка;
- Sample value – шаг дискретизации.

### 1.12.5. Блок решения системы дифференциальных уравнений

Блок решения системы дифференциальных уравнений не входит в состав библиотек. После ввода в командном окне Matlab'a команды **dee** появляется окно, показанное на рис. 1.37. Из указанного окна перетаскивается в окно редактирования модели блок DEE. При двойном щелчке мышкой на блоке в окне редактирования модели появляется диалоговое окно, показанное на рис. 1.38.

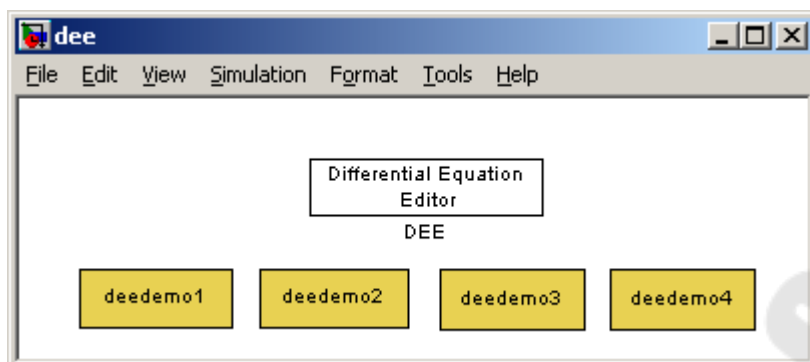


Рис. 1.37. Окно с блоком решения систем дифференциальных уравнений.

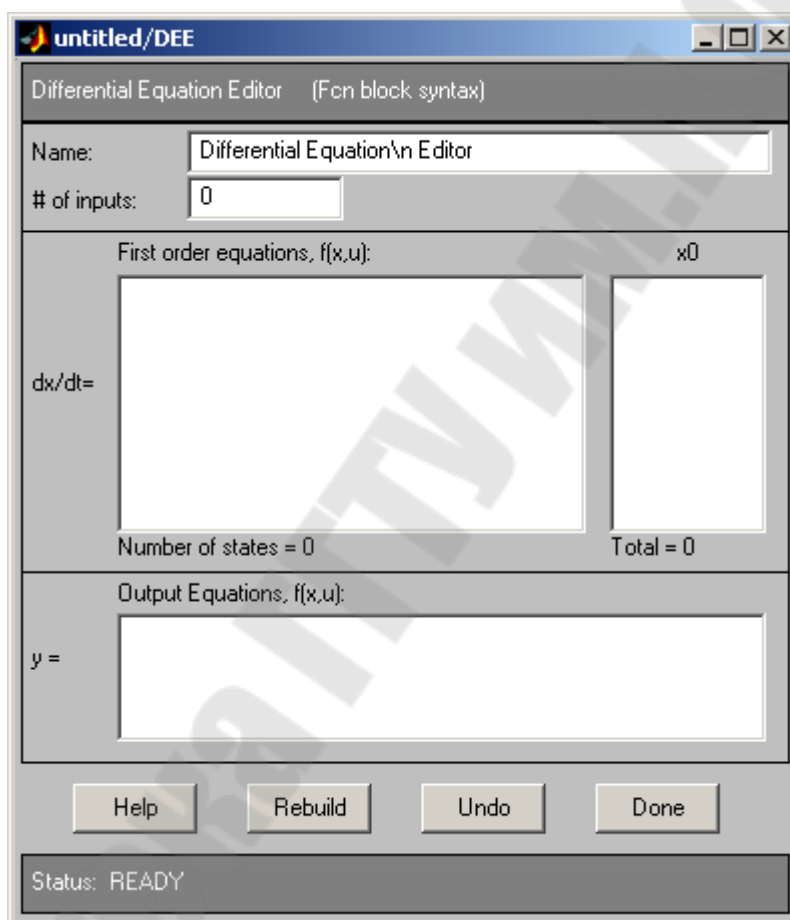


Рис. 1.38. Окно редактирования системы дифференциальных уравнений.

Параметры блока:

- Name – имя блока, которое будет отображаться в окне редактирования модели;
- # of inputs – количество входных переменных блока;
- First order equations,  $f(x,u)$  – выражения для правых частей уравнений в текстовой записи;
- $x_0$  – начальные значения переменных;

– Output Equations,  $f(x,u)$  – выражения, определяющие выходные сигналы блока.

Приведем пример для схемы, представленной на рис. 1.16. В блоке дифференциальных уравнений входные сигналы обозначаются  $u(1)$ ,  $u(2)$  и т.д. Переменные, относительно которых записаны уравнения –  $x(1)$ ,  $x(2)$  и т.д. Тогда система уравнений примет вид

$$\begin{cases} \frac{dx(1)}{dt} = \frac{2 \cdot u(1) - x(1)}{0,001}, \\ \frac{dx(2)}{dt} = \frac{3 \cdot x(1) - x(2)}{0,002}. \end{cases}$$

Для данной системы входной сигнал – один, обозначенный на рис. 1.16 через  $x$ . Внутренние переменные системы – две, обозначенные на рис. 1.16 через  $y_1$  и  $y_2$ . Выходной сигнал будет  $y_2$ . При нулевых начальных условиях редактор системы дифференциальных уравнений для рассматриваемого случая должен быть заполнен как показано на рис. 1.39.

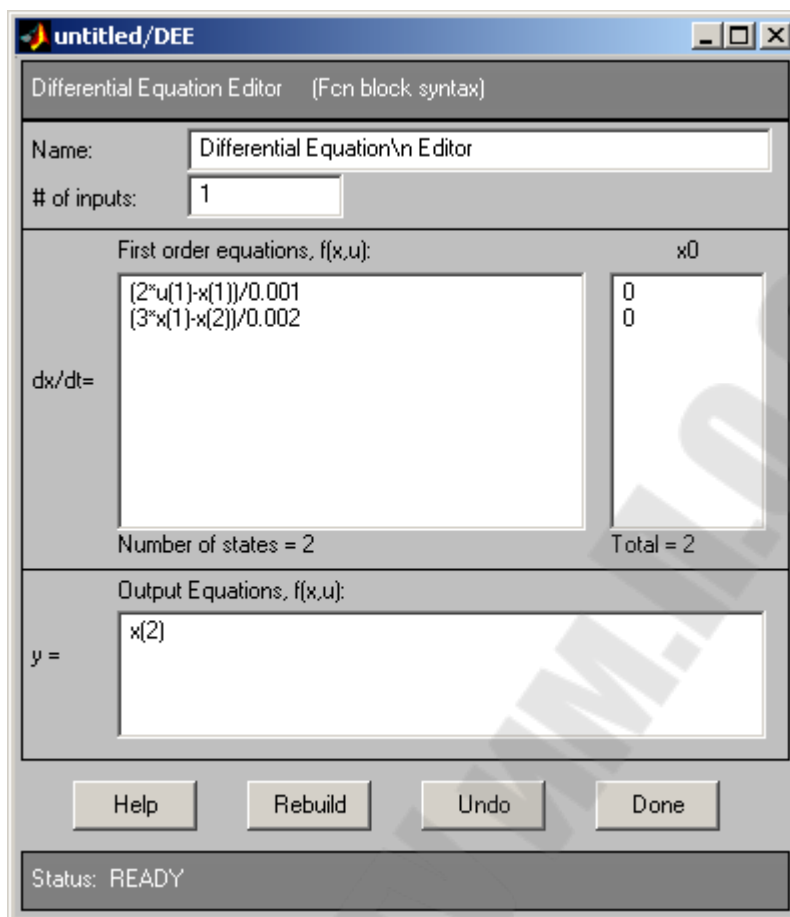


Рис. 1.39. Окно редактора системы дифференциальных уравнений, заполненный для системы, показанной на рис. 1.16.

### 1.13. Расчет переходных процессов

Когда модель набрана и заданы параметры всех блоков (или готовая модель загружена из файла), можно приступить к расчету переходных процессов.

Вначале необходимо задать параметры расчета. Для этого выбираем пункт меню **Simulation/Simulation Parameters...** При этом появляется окно параметров расчета, показанное на рис. 1.40.

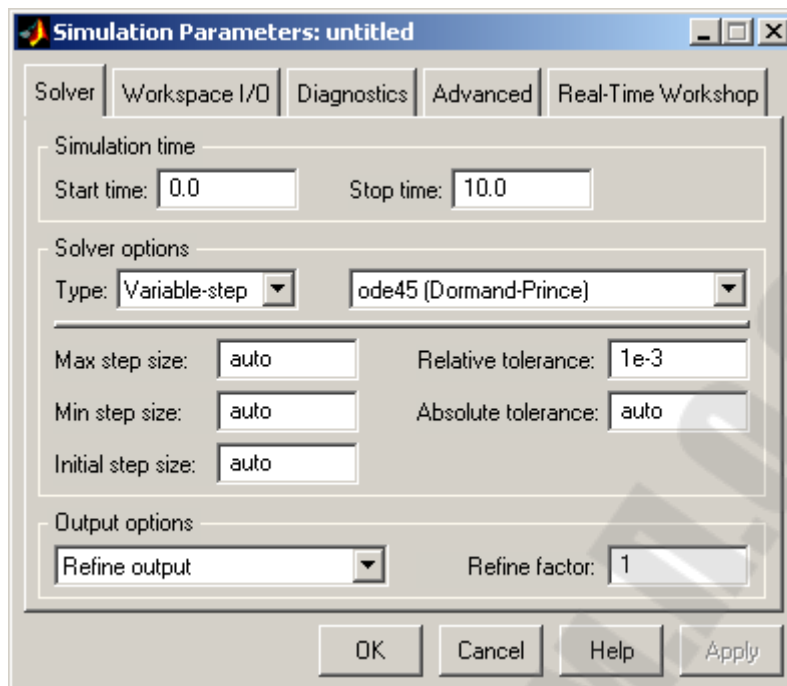


Рис. 1.40. Окно параметров расчета.

В указанном окне в группе элементов Simulation time задаются:

- Start time – начальное время расчета;
- Stop time – конечное время расчета;

В группе элементов Solver options – параметры метода численного решения:

- Type – тип метода: с переменным (Variable-step) или фиксированным (Fixed-step) шагом расчета;
- выпадающий список, правее списка Type – позволяет выбрать непосредственно численный метод;
- Max step size (для методов с переменным шагом) – максимальное значение шага расчета;
- Min step size (для методов с переменным шагом) – минимальное значение шага расчета;
- Initial step size (для методов с переменным шагом) – начальное значение шага расчета;
- Relative tolerance (для методов с переменным шагом) – относительная точность расчета;
- Absolute tolerance (для методов с переменным шагом) – абсолютная точность расчета;
- Fixed step size (для методов с постоянным шагом) – значение шага расчета (“auto” – для автоматического определения);
- Mode (для методов с постоянным шагом) – режим расчета: многозадачный (MultiTasking), однозадачный (SingleTasking) и

автоматически определяемый (Auto). Режим многозадачности даст увеличение производительности и снизит затраты времени, когда в модели есть подсистемы, работающие параллельно.

После задания параметров запуск процесса расчета производится соответствующей кнопкой на панели инструментов или выбором пункта меню **Simulation/Start**.

### **1.14. Инструменты анализа линейных систем в MATLAB Simulink**

Средства анализа линейных систем в Matlab Simulink позволяют строить переходные, логарифмические и ряд других характеристик. В Matlab'е версии 6.5 при выборе пункта меню **Tools/Linear analysis...** открывается 2 окна:

1. Окно входов и выходов, представленное на рис. 1.41.
2. Окно просмотра результатов анализа, представленное на рис. 1.42.

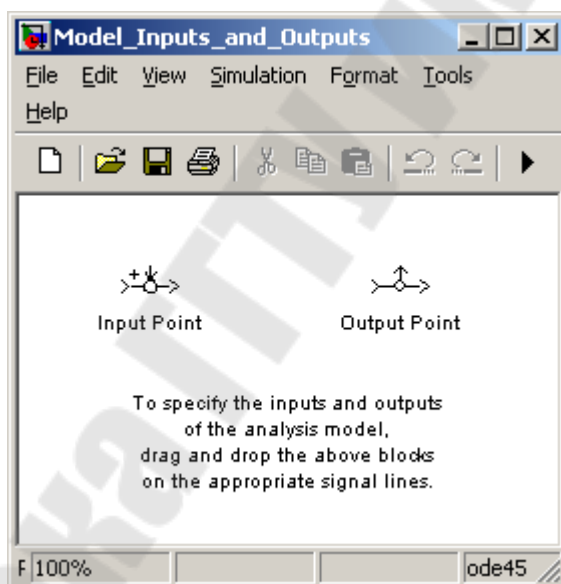


Рис. 1.41. Окно входов и выходов.

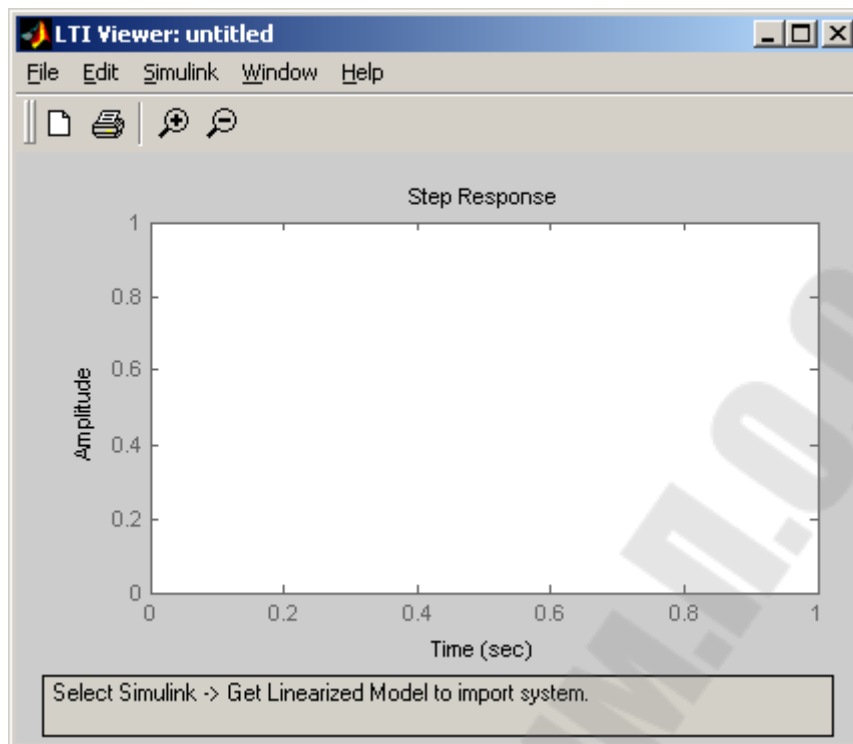


Рис. 1.42. Окно просмотра результатов анализа.

Порядок работы следующий:

1. В окне редактирования модели перетаскиваются блоки входа (Input Point) и выхода (Output Point) и соединяются с соответствующими выводами анализируемой модели. Пример показан на рис. 1.43.

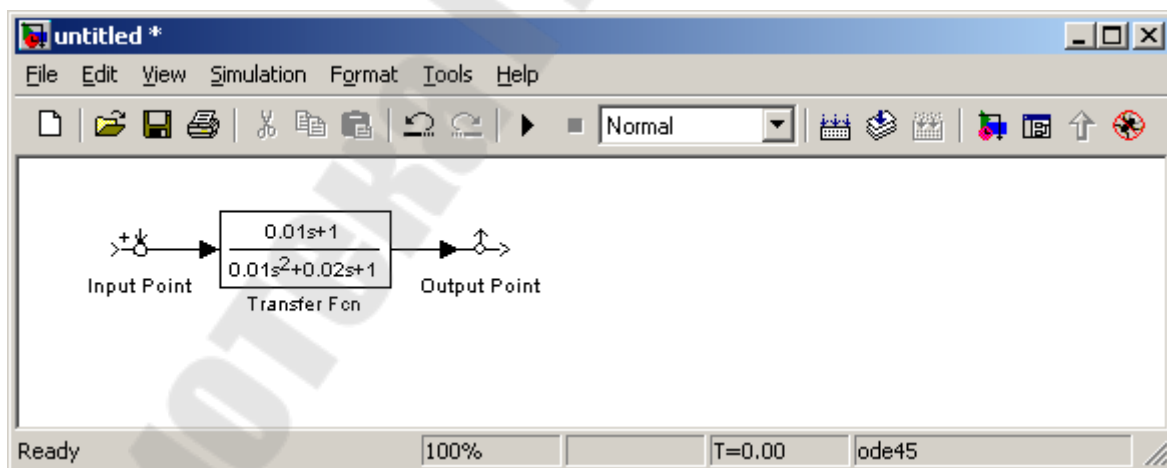


Рис. 1.43. Пример подготовки модели к анализу.

2. В окне результатов анализа выбирается пункт меню **Simulink/Get Linearized Model**. После этого появляются результаты анализа. По умолчанию строится только переходная характеристика. При необходимости построения других характеристик необходимо

выбрав пункт меню **Edit/Plot Configurations...** изменить параметры окна. В левой части появившегося окна (представленного на рис. 1.44) выбирается количество и расположение графиков, в правой части – вид характеристик для построения. Доступны следующие виды:

- Step – переходная характеристика (реакция системы на скачек);
- Impulse – весовая характеристика (реакция системы на импульс);
- Bode – логарифмические амплитудная и фазовая частотные характеристики;
- Bode Magnitude – логарифмическая амплитудная частотная характеристика;
- Nyquist – диаграмма Найквиста;
- Nichols – годограф Николса;
- Pole/Zero – нули и полюса системы.

На рис. 1.45 представлен пример окна результатов анализа с четырьмя характеристиками.

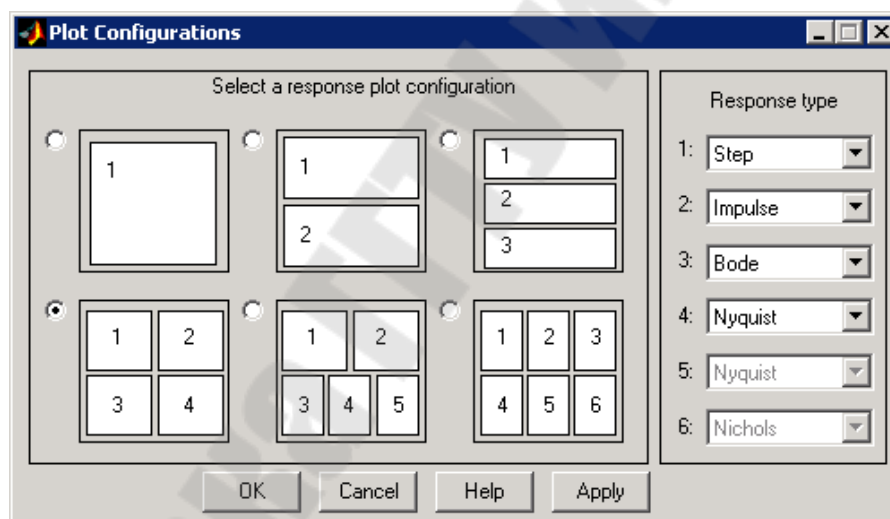


Рис. 1.44. Окно параметров LTI.



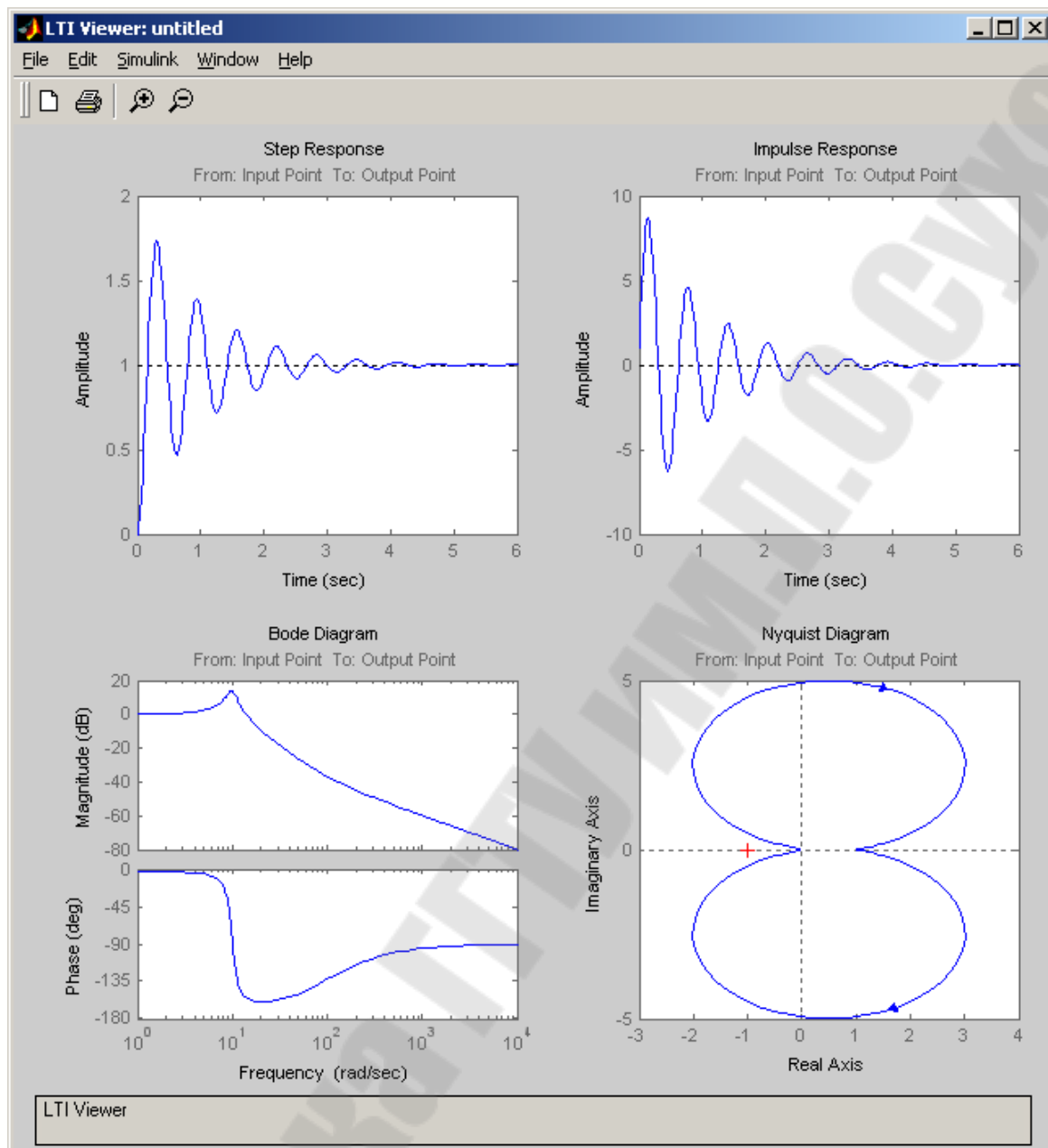


Рис. 1.45. Окно просмотра результатов анализа с четырьмя характеристиками.

В Matlab'e версии 7.0 порядок работы несколько иной. Вначале в окне редактирования модели задаются входная и выходная переменные. Для этого в контекстном меню необходимой переменной выбирается пункт **Linearization Points/Input Point** для входной и **Linearization Points/Output Point** для выходной переменной, как показано на рис. 1.46.

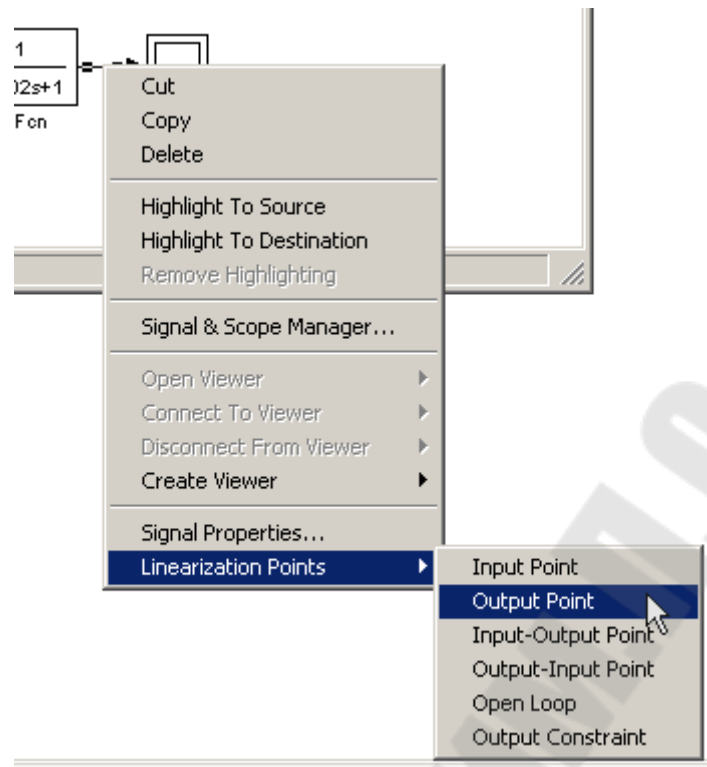


Рис. 1.46. Подменю Linearization Points контекстного меню переменной.

Возле переменных появятся соответствующие значки, как показано на рис. 1.47.

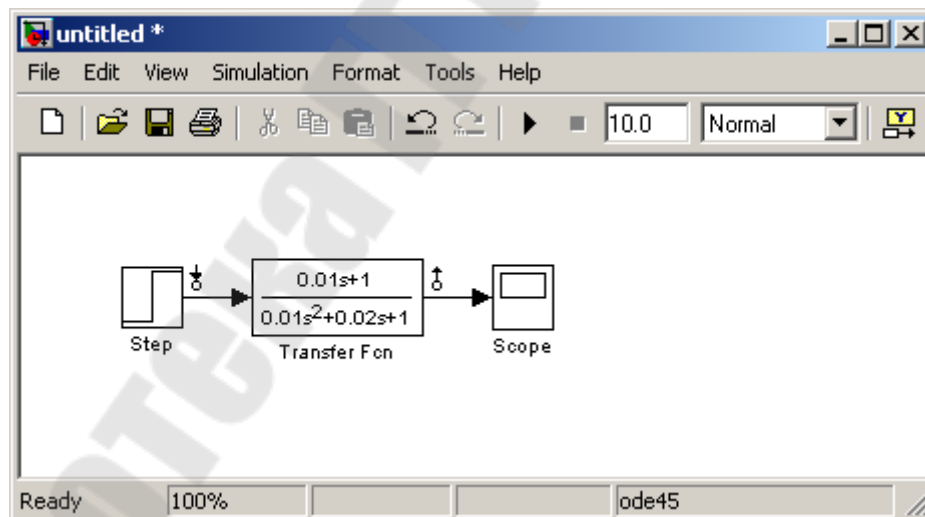


Рис. 1.47. Пример подготовки модели к анализу.

Затем в меню **Tools** окна редактирования модели выбирается пункт **Control Design/Linear Analysis...** В появившемся окне диспетчера инструментов анализа нажимается кнопка **Linearize Model**. При этом Matlab производит линейризацию модели и отображает окно просмотра результатов анализа, аналогичное

показанному на рис. 1.45. Работа с которым аналогична рассмотренному ранее.

### 1.15. Синтез регуляторов

В Simulink имеется несколько инструментов синтеза регуляторов. Рассмотрим инструмент Time-domain response optimization. Порядок работы следующий:

1. Сформировать модель объекта управления и системы управления.
2. В рабочей области Matlab создать переменные для параметров регулятора и задать для них начальные значения. Для этого можно выполнить команды присвоения в окне **Command Window**, как показано на рис. 1.48. Параметры блоков, входящих в состав регулятора, задать используя эти переменные. Следует иметь в виду, что блок производной использовать нельзя. Для ПИ-регулятора передаточная функция раскладывается на сумму двух составляющих

$$W(p) = K \cdot \frac{T \cdot p + 1}{T \cdot p} = K + \frac{K}{T \cdot p}$$

и собирается в Matlab как параллельное соединение. Для ПИД-регулятора вместо производной в дифференцирующей составляющей следует использовать дифференцирующее звено с замедлением с малой постоянной времени  $\tau$  в знаменателе

$$W(p) = K \cdot \frac{(T_1 \cdot p + 1) \cdot (T_2 \cdot p + 1)}{T_1 \cdot p} \approx K \cdot \frac{T_2 \cdot p}{\tau \cdot p + 1} + K \cdot \frac{T_1 + T_2}{T_1} + \frac{K}{T_1 \cdot p}.$$

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> Krt=0.01

Krt =

    0.0100

>> Trt=0.01

Trt =

    0.0100

fx >>
  
```

Рис. 1.48. Присвоение начальных значений параметрам регулятора в окне команд.

3. Задать входные сигналы системы управления для получения на выходе системы ожидаемой реакции.
4. Задать интервал расчета и, при необходимости, изменить метод расчета и шаг.
5. Добавить в модель блок **Signal Constraint** библиотеки **Simulink Response Optimization**. Для удобства работы перед блоком желательно задать коэффициент передачи для перевода значения сигнала в относительные единицы. Пример подготовленной модели показан на рис. 1.49.

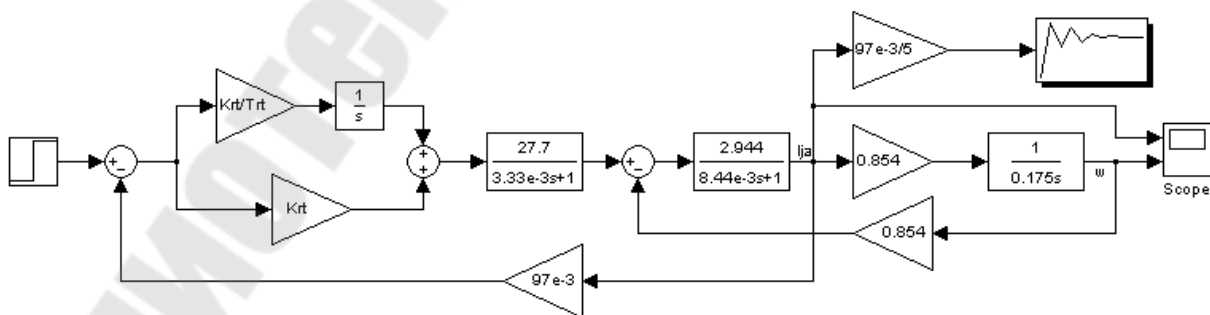


Рис. 1.49. Пример модели контура тока якоря, подготовленной к оптимизации.

6. Двойным щелчком мышкой по блоку **Signal Constraint** открыть окно диаграммы ограничения сигнала.

7. Для указания параметров регулятора в меню **Optimization** выбрать пункт **Tuned parameters...** В появившемся окне, показанном на рис. 1.50, для добавления переменных регулятора из рабочей области нажать кнопку **Add**. В окне выбора, показанном на рис. 1.51, выбрать необходимые переменные рабочей области и нажать **Ok**. Окно параметров после этого примет вид, показанный на рис. 1.52. Тут можно задать допустимые пределы для каждого параметра. После завершения работы со списком параметров нажать **Ok**.

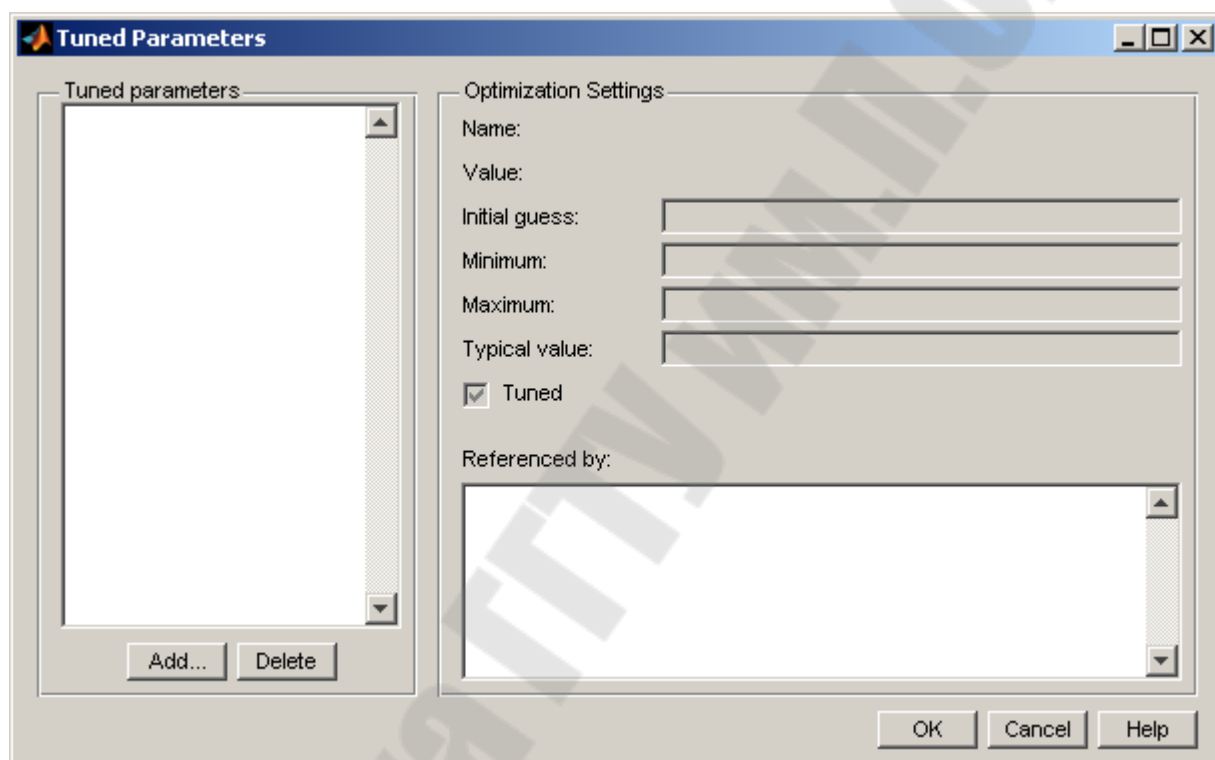


Рис. 1.50. Окно списка параметров для настройки.

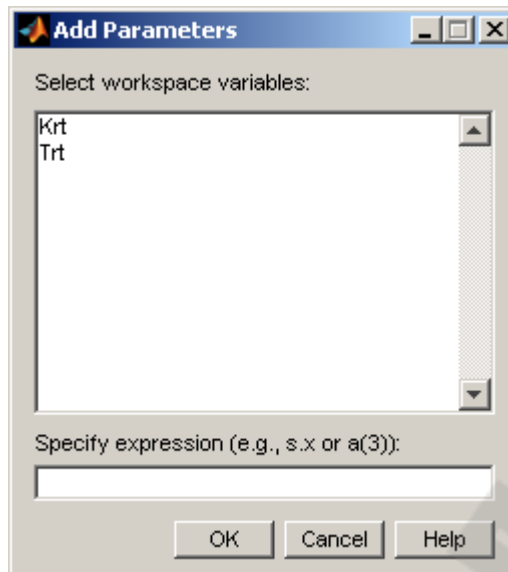


Рис. 1.51. Окно выбора переменных.

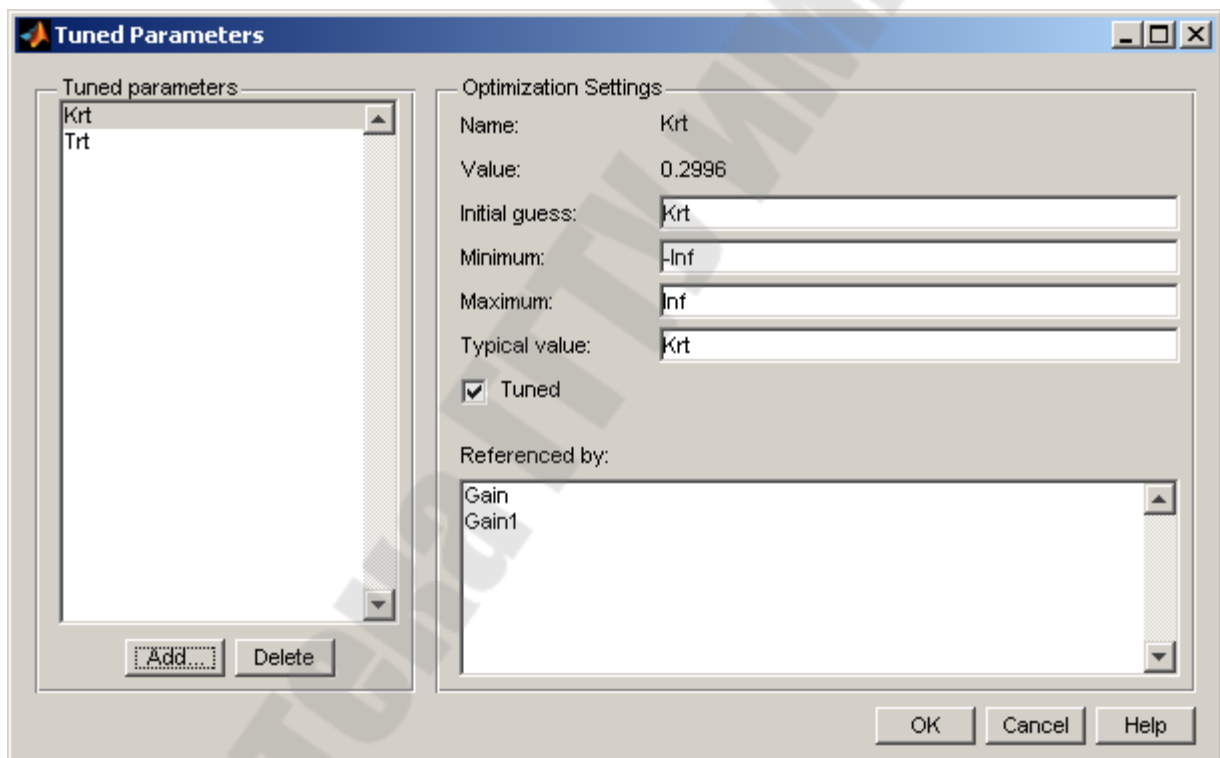


Рис. 1.52. Окно списка параметров для настройки.

8. В окне редактирования ограничений сигнала отредактировать их. Перетаскивать можно сегменты и точки диаграммы. Для точного указания значений необходимо воспользоваться соответствующим инструментом. Для этого в контекстном меню сегмента, показанном на рис. 1.53, выбрать пункт меню **Edit...** Вид окна редактирования показан на рис. 1.54. Выпадающий список **Design Requirement** позволяет переключаться между нижней **Lower time**

**response bound** и верхней **Upper time response bound** границами области. Колонки **Time** и **Amplitude** указывают время и значение сигнала в начале **Start** и конце **End** сегмента диаграммы. Колонки **Slope** и **Weight** содержат значение скорости изменения и значимости сегмента. Значимость может принимать значения от 0 до 1. При значении 0 сегмент не учитывается. При 1 – обязателен.

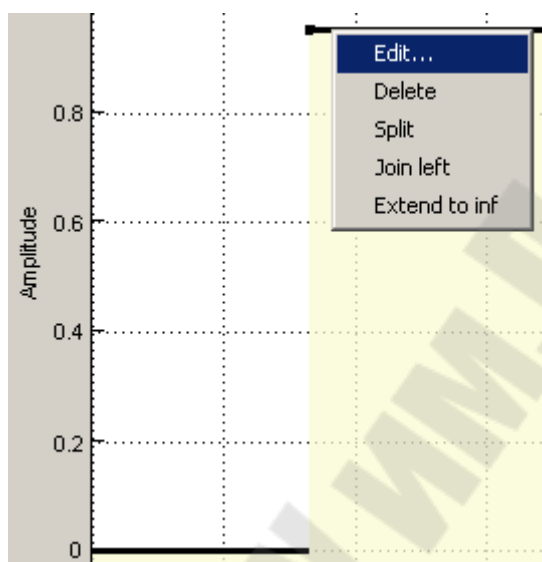


Рис. 1.53. Контекстное меню сегмента области ограничения сигнала.

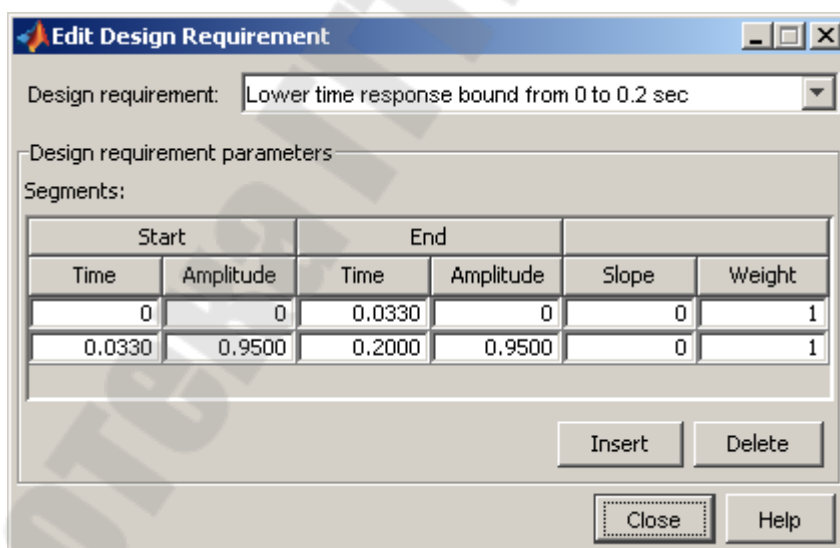
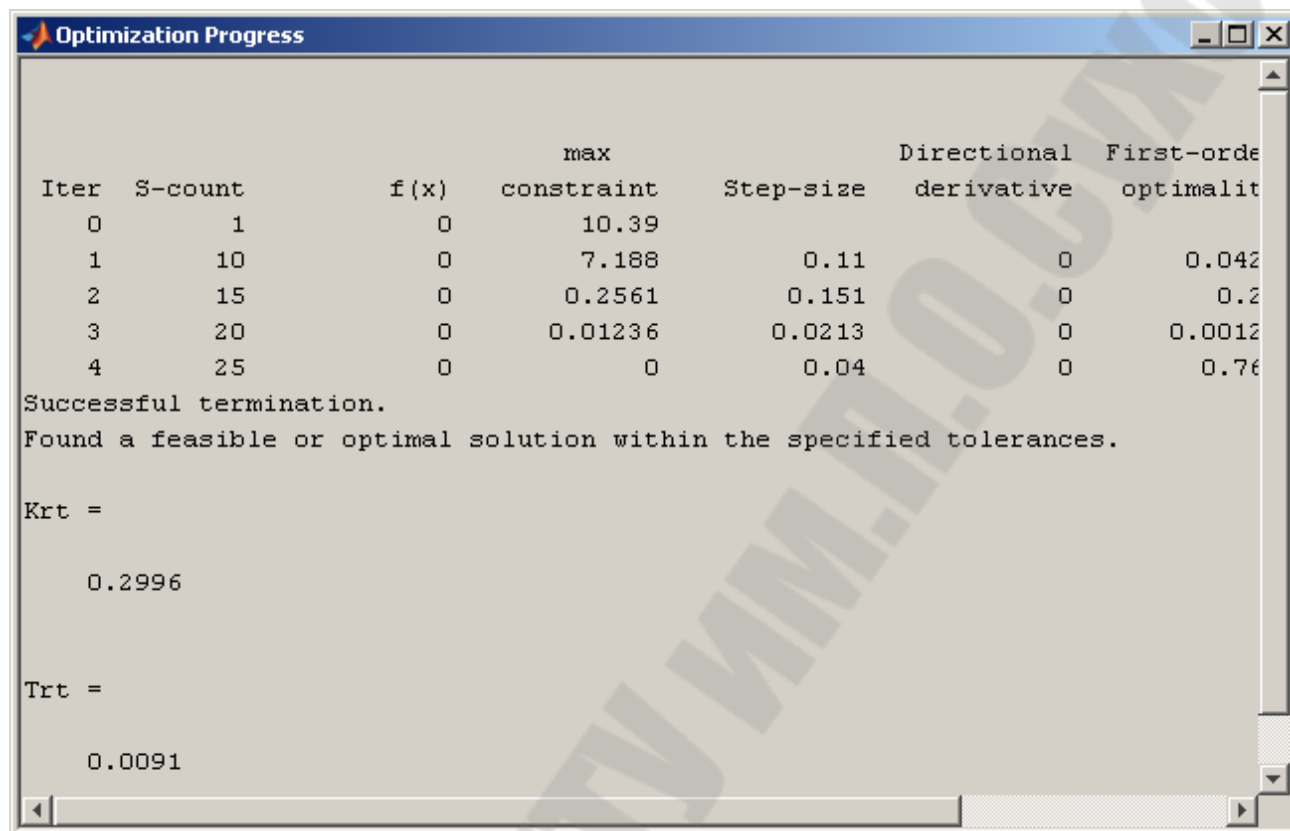


Рис. 1.54. Окно редактирования области ограничения сигнала.

- После задания параметров и редактирования области ограничения сигнала для старта процесса оптимизации необходимо выбрать в меню **Optimization** пункт **Start** или нажать соответствующую кнопку на панели инструментов. В процессе оптимизации отображается ход выполнения и полученные диаграммы сигнала.

Окно хода выполнения и результаты оптимизации показаны на рис. 1.55 и 1.56.



The screenshot shows a window titled "Optimization Progress" with a table of iteration data and a text message indicating successful termination. The table has columns for Iter, S-count, f(x), max constraint, Step-size, Directional derivative, and First-order optimality. The data shows a decrease in the max constraint from 10.39 to 0 over 4 iterations. Below the table, the text reads: "Successful termination. Found a feasible or optimal solution within the specified tolerances. Krt = 0.2996 Trt = 0.0091".

Iter	S-count	f(x)	max constraint	Step-size	Directional derivative	First-order optimality
0	1	0	10.39			
1	10	0	7.188	0.11	0	0.042
2	15	0	0.2561	0.151	0	0.2
3	20	0	0.01236	0.0213	0	0.0012
4	25	0	0	0.04	0	0.76

Successful termination.  
Found a feasible or optimal solution within the specified tolerances.

Krt =  
0.2996

Trt =  
0.0091

Рис. 1.55. Окно хода выполнения и результатов оптимизации.



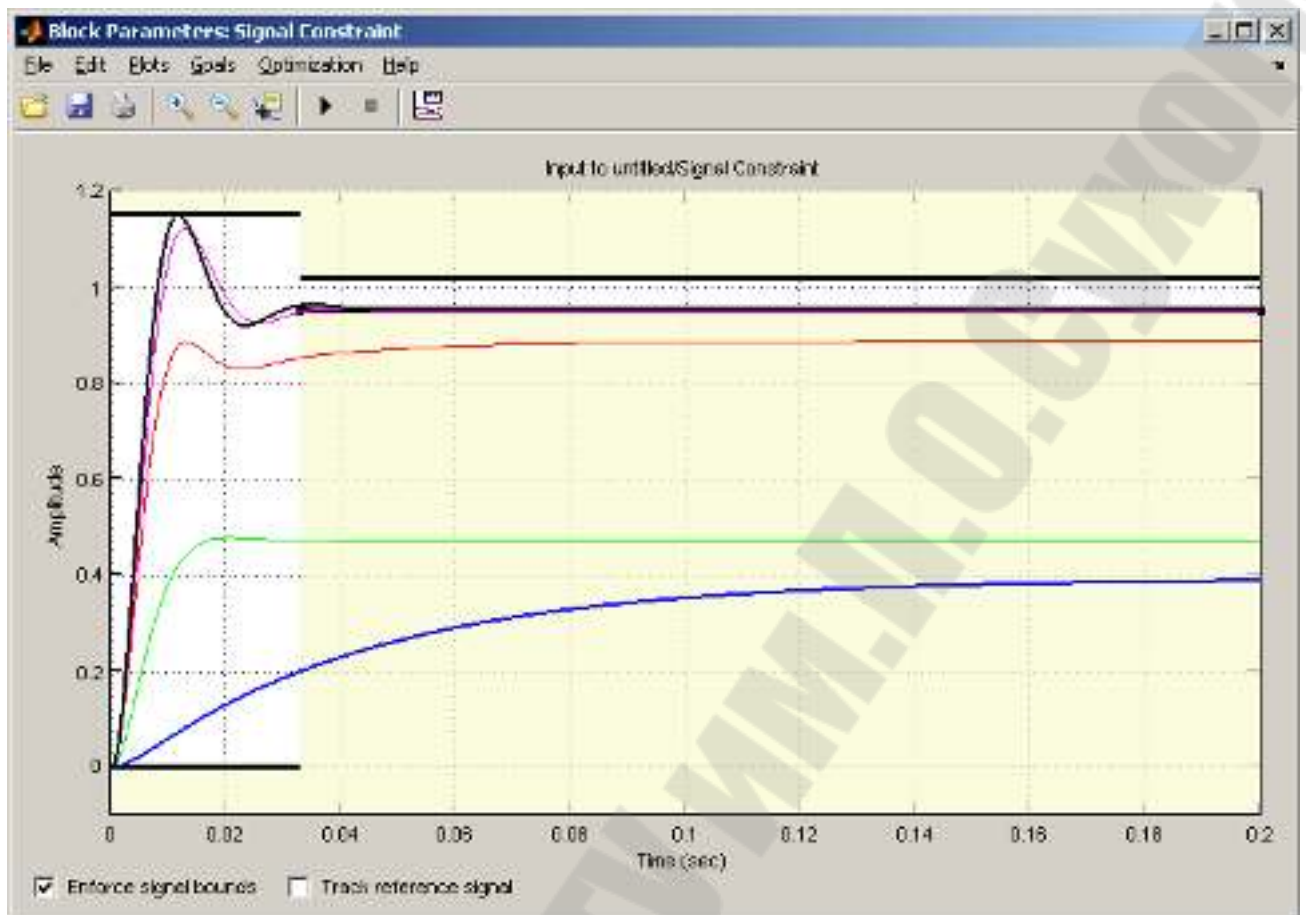


Рис. 1.56. Окно области ограничения с полученными диаграммами сигнала.

## **2. Лабораторные работы**

В данном разделе приведены задания на лабораторные работы и методические указания по их выполнению. Исходные данные для определения параметров моделей здесь не приводятся. Исходные данные выдаются преподавателем.

### **2.1. Лабораторная работа №1 Моделирование типовых статических и динамических элементов систем автоматического управления**

Цель работы:

- ознакомиться с алгоритмами моделирования статических элементов систем автоматического управления;
- ознакомиться с приемами моделирования динамических элементов;
- получить практические навыки составления программ моделирования.

Задание для моделирования по системе уравнений:

11. По заданной структурной схеме модели составить систему уравнений.
12. На основе заданного метода численного решения дифференциальных уравнений и составленной системы уравнений составить программу для моделирования.
13. Используя заданные параметры входных воздействий и звеньев модели с помощью составленной программы рассчитать переходной процесс.
14. Построить диаграммы переходных процессов.

Задание для моделирования по структурной схеме:

1. На основе заданного метода численного решения дифференциальных уравнений записать расчетные соотношения для определения значения выходного сигнала звена на текущем шаге для всех типов звеньев, входящих в состав заданной модели.
2. По структурной схеме модели составить программу для моделирования.
3. Используя заданные параметры входных воздействий и звеньев модели с помощью составленной программы рассчитать переходной процесс.
4. Построить диаграммы переходных процессов.

Контрольные вопросы:

1. Опишите структуру программы моделирования.
2. Составьте расчетное соотношение для звена с наклонной петлей гистерезиса.
3. Постройте качественно вид выходного сигнала системы при изменении вида одного или нескольких входных воздействий.
4. Как изменится вид выходного сигнала при изменении какого-либо параметра динамического звена?

## **2.2. Лабораторная работа №2**

### **Моделирование электродвигателя постоянного тока независимого и последовательного возбуждения**

Цель работы:

- закрепить полученные практические навыки составления программ моделирования;
- исследовать динамические свойства двигателей постоянного тока.

Задание для моделирования по системе уравнений:

1. Для модели двигателя постоянного тока заданного типа возбуждения составить систему уравнений (при этом учесть реактивный характер момента сопротивления  $M_C = \text{sign}(\omega)$ ).
2. На основе заданного метода численного решения дифференциальных уравнений и составленной системы уравнений составить программу для моделирования.
3. С помощью составленной программы рассчитать переходные процессы, указанные ниже.
4. Построить диаграммы переходных процессов и динамическую электромеханическую характеристику.

Задание для моделирования по структурной схеме:

1. На основе заданного метода численного решения дифференциальных уравнений записать расчетные соотношения для определения значения выходного сигнала звена на текущем шаге для всех типов звеньев, входящих в состав модели двигателя постоянного тока заданного типа возбуждения.
2. По структурной схеме модели составить программу для моделирования (при этом учесть реактивный характер момента сопротивления  $M_C = \text{sign}(\omega)$ ).

3. С помощью составленной программы рассчитать переходные процессы, указанные ниже.
4. Построить диаграммы переходных процессов и динамическую электромеханическую характеристику.

Переходные процессы для двигателя независимого возбуждения:

1. Пуск под нагрузкой ( $M_C = 0,5 \cdot M_H$ ).
2. Наброс нагрузки (до  $M_H$ ).
3. Реверс.
4. Торможение.

Переходные процессы для двигателя последовательного возбуждения:

1. Пуск под нагрузкой ( $M_C = 0,5 \cdot M_H$ ).
2. Наброс нагрузки (до  $M_H$ ).
3. Торможение.

Контрольные вопросы:

1. Приведите структурную схему двигателя постоянного тока при регулировании по цепям якоря и возбуждения.
2. Составьте систему дифференциальных уравнений для моделирования двигателя постоянного тока при регулировании по цепям якоря и возбуждения.
3. Приведите критерий выбора величины шага расчета для метода Башарина.

### **2.3. Лабораторная работа №3 Моделирование асинхронного электродвигателя**

Цель работы:

- получить практические навыки моделирования асинхронных электродвигателей;
- ознакомиться с особенностями динамики асинхронных электродвигателей.

Задание:

1. Для модели асинхронного двигателя с короткозамкнутым ротором, записанной в заданной системе координат в MathCAD (используя функцию `rkfixed`) рассчитать переходные процессы при пуске ( $M_C = 0,1 \cdot M_H$ ) и набросе нагрузки (до  $M_H$ ).
2. Построить диаграммы переходных процессов, статическую и динамическую механические характеристики, как показано на рис. 2.1.

Контрольные вопросы:

1. Является ли модель асинхронного двигателя жесткой?
2. Является ли модель асинхронного двигателя жестко-колебательной?

#### **2.4. Лабораторная работа №4** **Моделирование системы автоматического** **управления скоростью двигателя** **постоянного тока независимого возбуждения**

Цель работы:

- освоить приемы работы с программой SMED;
- изучить влияние ограничения регуляторов на динамические процессы в системах автоматического регулирования.

Задание:

1. Для двигателя независимого возбуждения заданного в лабораторной работе №2 типоразмера и заданного трансформатора рассчитать параметры силовой части системы автоматического управления.
2. Собрать в Matlab Simulink модель заданной системы автоматического управления, используя заданные параметры регуляторов и рассчитанные параметры силовой части системы. При этом учесть ограничение регуляторов на уровне  $\pm 10$  В и реактивный характер момента сопротивления.
3. Рассчитать переходные процессы, указанные ниже (номинальное значение сигнала задания  $U_{зсн} = 5$  В).
4. Построить диаграммы переходных процессов (скорость вращения двигателя, ток или момент) и динамическую механическую или электромеханическую характеристику.

Переходные процессы:

1. Пуск под нагрузкой ( $M_C = 0,5 \cdot M_H$ ) при  $u_{зс} = 0,95 \cdot U_{зсн}$ .
2. Наброс сигнала задания до  $u_{зс} = U_{зсн}$ .
3. Наброс нагрузки (до  $M_H$ ).
4. Реверс.
5. Торможение.

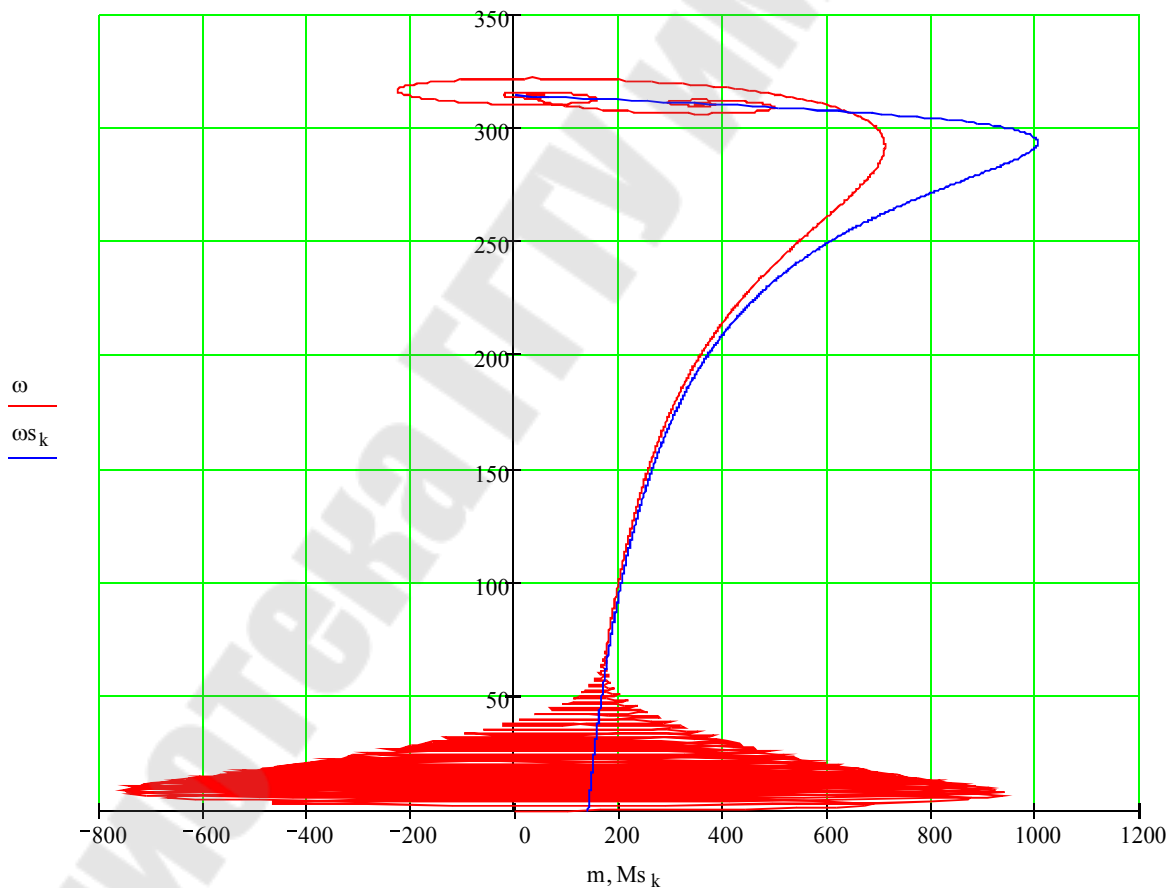
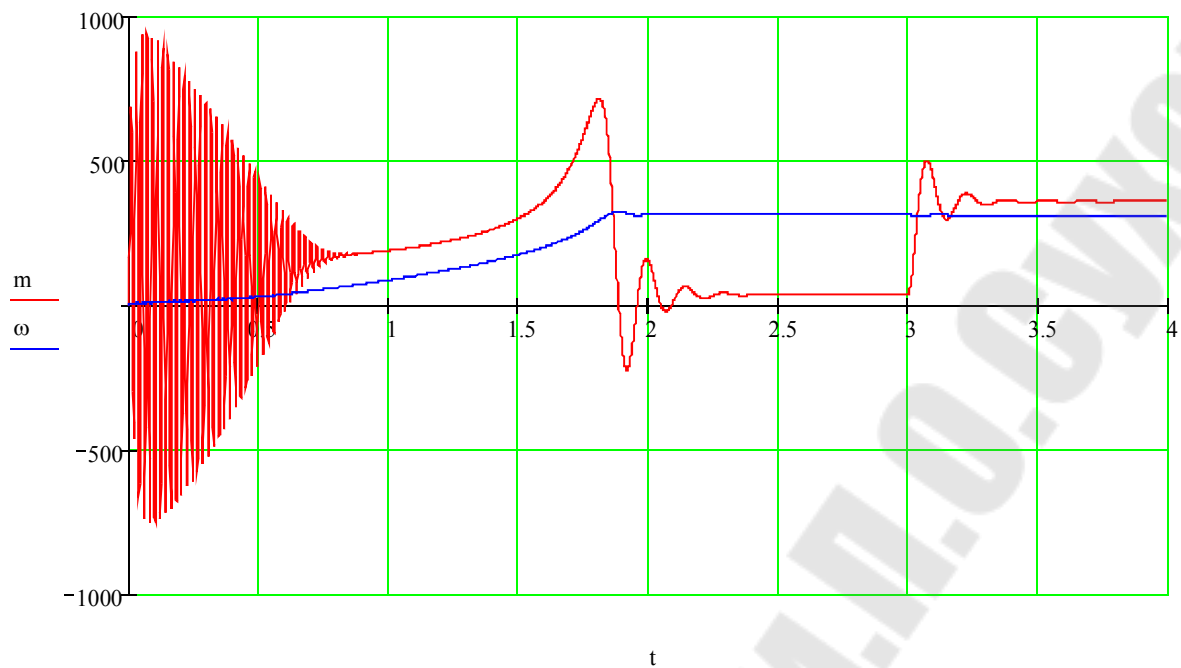


Рис. 2.1. Пример построения диаграмм переходных процессов и механических характеристик.

Контрольные вопросы:

1. Составьте систему уравнений для модели заданной системы автоматического управления.

2. Как изменятся диаграммы переходных процессов, если не учитывать ограничение регуляторов?
3. Как изменится динамическая характеристика, если не учитывать ограничение регуляторов?

### **2.5. Лабораторная работа № 5. Структурные преобразования и построение частотных характеристик в MathCAD**

Цель работы:

- освоить приемы работы в MathCAD'е для проведения преобразований структурных схем,
- освоить методы построений логарифмических характеристик в MathCAD'е.

Задание:

1. В MathCAD'е произвести преобразования структурной схемы, заданной преподавателем, для получения передаточной функции всей системы.
2. Построить логарифмические амплитудную и фазовую частотные характеристики.

Контрольные вопросы:

1. Как задавать фазовую характеристику, если в ней получился разрыв?
2. Как в MathCAD'е задается мнимая единица?

### **2.6. Лабораторная работа № 6. Использование символьных вычислений и программирования в MathCAD**

Цель работы:

- освоить метод расчета переходных процессов в MathCAD'е с использованием символьных вычислений,
- освоить метод расчета переходных процессов в MathCAD'е с использованием блока программирования.

Задание:

1. В MathCAD'е произвести расчет переходных процессов в структурной схеме, заданной преподавателем, используя обратное преобразование Лапласа.
2. Произвести расчет переходных процессов, используя блок программирования (методом Эйлера). По результатам расчета

построить диаграммы изменения выходного сигнала системы от времени.

### **2.7. Лабораторная работа № 7. Расчет переходных процессов в Matlab Simulink**

Цель работы:

- освоить работу с программой Matlab Simulink, ее интерфейс,
- освоить приемы работы в Matlab Simulink при расчете переходных процессов.

Задание:

1. В Matlab Simulink набрать структурную схему, заданную преподавателем.
2. Рассчитать переходной процесс в системе при единичном скачке на входе и построить диаграмму изменения выходного сигнала во времени.

### **2.8. Лабораторная работа № 8. Использование инструментов анализа линейных систем в Matlab Simulink**

Цель работы:

- ознакомиться со средствами анализа линейных систем Matlab Simulink.

Задание:

1. в Matlab Simulink набрать структурную схему, заданную преподавателем.
2. Используя инструмент анализа линейных систем, построить: реакцию системы на импульс, логарифмические амплитудную и фазовую частотные характеристики, годограф, поле распределения корней.

### **2.9. Лабораторная работа № 9. Использование средств решения систем дифференциальных уравнений Matlab Simulink**

Цель работы:

- ознакомиться со средствами численного решения систем дифференциальных уравнений Matlab Simulink.

Задание:

1. По заданной преподавателем структурной схеме системы составить систему дифференциальных уравнений.



2. В Matlab Simulink набрать схему, состоящую из источника сигнала, осциллографа (**Scope**) и блока решения системы дифференциальных уравнений.
3. В блок решения дифференциальных уравнений ввести составленную систему.
4. Рассчитать переходной процесс в системе при единичном скачке на входе и построить диаграмму изменения во времени выходного сигнала системы.

### **2.10. Лабораторная работа № 10. Моделирование системы автоматического управления скоростью двигателя постоянного тока независимого возбуждения в Matlab Simulink**

Цель работы:

- освоить приемы моделирования систем управления электроприводами с учетом их нелинейности.

Задание:

1. Для заданных преподавателем двигателя и трансформатора, рассчитать их параметры.
2. В Matlab Simulink набрать схему двухконтурной однозонной САУ скоростью ДПТ НВ с учетом ограничения регуляторов на уровне  $\pm 10$  В. Передаточные функции регуляторов и коэффициенты обратных связей задаются преподавателем.
3. Рассчитать переходные процессы пуска на холостой ход и наброса нагрузки до половины номинальной.
4. По результатам расчета построить диаграммы тока и скорости, динамическую электромеханическую характеристику.

### **2.11. Лабораторная работа № 11. Синтез регуляторов в Matlab Simulink**

Цель работы:

- освоить синтез систем с помощью инструмента Time-domain response optimization.

Задание:

1. Для модели САУ, полученной при выполнении лабораторной работы № 10 произвести синтез регуляторов тока и скорости в следующем порядке:
  - Разорвать контур скорости и произвести синтез ПИ-регулятора тока при следующих ограничениях: выход на установившееся

значение (+2 % и -5 %) за 29 мс, перерегулирование 5 %. В случае невозможности получения решения, ослабить требования, но перерегулирование не более 8 %.

- Замкнуть контур скорости и произвести синтез П-регулятора скорости при следующих ограничениях: выход на установившееся значение ( $\pm 2$  %) в течении времени в 2 раза большего полученному времени переходного процесса контура тока, перерегулирование 5 %. В случае невозможности получения решения, ослабить требования, но перерегулирование не выше 8 %.
- 2. Рассчитать переходные процессы «в малом» по управлению в исходной системе (по лабораторной работе № 6) и полученной. Сравнить динамические показатели.

### 3. Пример выполнения лабораторной работы №1

В данном разделе подробно рассмотрен порядок выполнения работы, а не пример оформления отчета.

#### 3.1. Моделирование по системе уравнений

Дана структурная схемы, представленная на рис. 3.1. Обозначим переменные модели:

- $x$  – входные воздействия,
- $y$  – сигналы, для которых будут составлены дифференциальные уравнения в форме Коши (выходные звенья динамических звеньев, имеющих оператор Лапласа в знаменателе передаточной функции),
- $z$  – сигналы, для которых будут записаны алгебраические уравнения (выходные сигналы статических звеньев, а также динамических, не имеющих оператор Лапласа в знаменателе передаточной функции).

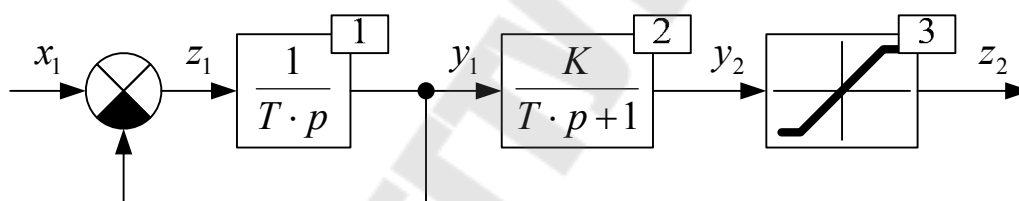


Рис. 3.1. Структурная схема.

Составим систему уравнений.

Уравнение для вычитателя

$$z_1 = x_1 - y_1.$$

Передаточная функция интегрирующего звена

$$W(p) = \frac{1}{T_1 \cdot p} = \frac{y_1(p)}{z_1(p)}.$$

Выразим операторное изображение выходного сигнала звена

$$y_1(p) = \frac{1}{T_1 \cdot p} \cdot z_1(p).$$

Умножив на оператор Лапласа, получим выражение для производной выходного сигнала звена в операторной форме

$$p \cdot y_1(p) = \frac{1}{T_1} \cdot z_1(p).$$

Перейдя к функциям времени получим дифференциальное уравнение в форме Коши

$$\frac{dy_1}{dt} = \frac{1}{T_1} \cdot z_1.$$

Передаточная функция апериодического звена

$$W(p) = \frac{1}{T_2 \cdot p + 1} = \frac{y_2(p)}{y_1(p)}.$$

Выразим операторное изображение выходного сигнала звена

$$y_2(p) = \frac{K_2}{T_2 \cdot p + 1} \cdot y_1(p).$$

Умножим на знаменатель передаточной функции и раскроем скобки в левой части уравнения

$$\begin{aligned} y_2(p) \cdot (T_2 \cdot p + 1) &= K_2 \cdot y_1(p), \\ T_2 \cdot p \cdot y_2(p) + y_2(p) &= K_2 \cdot y_1(p). \end{aligned}$$

Перенеся второе слагаемое левой части в правую и разделив на постоянную времени  $T_2$ , получим выражение для производной выходного сигнала звена в операторной форме

$$\begin{aligned} T_2 \cdot p \cdot y_2(p) &= K_2 \cdot y_1(p) - y_2(p) \\ p \cdot y_2(p) &= \frac{1}{T_2} \cdot [K_2 \cdot y_1(p) - y_2(p)]. \end{aligned}$$

Перейдя к функциям времени получим дифференциальное уравнение в форме Коши

$$\frac{dy_2}{dt} = \frac{1}{T_2} \cdot (K_2 \cdot y_1 - y_2).$$

Для выходного сигнала звена с ограничением запишем алгебраическое выражение по таблице 1.2, учетом выбранных обозначений сигналов и параметров звеньев.

$$z_2 = K_3 \cdot y_2,$$

$$z_2 = \begin{cases} Y_{03}, & \text{при } z_2 > Y_{03}, \\ -Y_{03}, & \text{при } z_2 < -Y_{03}. \end{cases}$$

Т.о. получим систему уравнений

$$\begin{cases} z_1 = x_1 - y_1, \\ \frac{dy_1}{dt} = \frac{1}{T_1} \cdot z_1, \\ \frac{dy_2}{dt} = \frac{1}{T_2} \cdot (K_2 \cdot y_1 - y_2), \\ z_2 = K_3 \cdot y_2, \\ z_2 = \begin{cases} Y_{03}, & \text{при } z_2 > Y_{03}, \\ -Y_{03}, & \text{при } z_2 < -Y_{03}. \end{cases} \end{cases}$$

Составим подпрограмму расчета правых частей дифференциальных уравнений и алгебраических уравнений по составленной системе. При этом выделим в отдельную подпрограмму расчет значений входных воздействий. В рассматриваемом примере входное воздействие одно. Пусть вид входного воздействия задан скачок

$$x_1 = \begin{cases} 0, & \text{при } t < 0, \\ x_{m1}, & \text{при } t \geq 0. \end{cases}$$

В строках 3, 4 приведенного ниже фрагмента программы производится объявление типа переменной – одномерного массива. Без объявления типа передавать в подпрограммы переменные – массивы нельзя. В строках 6...13 объявлены глобальные переменные программы. В строке 8 объявлены массивы для хранения параметров звеньев и входных воздействий (назначение остальных переменные рассмотрим позже):

- 1K – коэффициенты передачи звеньев,
- 1T – постоянные времени звеньев,
- 1y0 – уровни ограничения звеньев,
- xm – амплитуда входных воздействий.

```

01 procedure TForm1.Button1Click(Sender: TObject);
02
03 type
04   DArray = array [1..10] of double;
05
06 var
07   Y, Yp, Z,
08   lK, lT, ly0,
09   xm: Darray;
10   tk, dt,
11   t: double;
12   i,
13   N: integer;
14
15 procedure cX(t: double; var x: DArray);
16 begin
17   if t >= 0 then x[1] := xm[1];
18   else x[1] := 0;
19 end;
20
21 procedure cF(Y: DArray; t: double; var Z, F: DArray);
22 var
23   x: Darray;
24 begin
25   cX(t, x);
26   Z[1] := x[1] - Y[1];
27   F[1] := Z[1]/lT[1];
28   F[2] := (lK[2]*Y[1] - Y[2])/lT[2];
29   Z[2] := lK[3]*Y[2];
30   if Z[2] > ly0[3] then Z[2] := ly0[3];
31   else if Z[2] < -ly0[3] then Z[2] := -ly0[3];
32 end;
33

```

$$x_1 = \begin{cases} 0, & ??? \ t < 0, \\ x_{m1}, & ??? \ t \geq 0. \end{cases}$$

$$\begin{cases} z_1 = x_1 - y_1, \\ \frac{dy_1}{dt} = \frac{1}{T_1} \cdot z_1, \\ \frac{dy_2}{dt} = \frac{1}{T_2} \cdot (K_2 \cdot y_1 - y_2), \\ z_2 = K_3 \cdot y_2, \\ z_2 = \begin{cases} Y_{03}, & ??? \ z_2 > Y_{03}, \\ -Y_{03}, & ??? \ z_2 < -Y_{03}. \end{cases} \end{cases}$$

В строках 15...19 находится подпрограмма расчета значений входных воздействий. Параметры подпрограммы:  $t$  – значение текущего времени;  $x$  – рассчитываемые значения входных воздействий.

В строках 21...32 находится подпрограмма, написанная по составленной системе уравнений. Параметры подпрограммы:  $Y, Z$  – массивы значений переменных;  $F$  – массив рассчитываемых значений правых частей дифференциальных уравнений ( $dy/dt$ );  $t$  – значение текущего времени. Внутренняя переменная  $x$  – значения входных воздействий.

На основе метода Эйлера составим подпрограмму расчета значений переменных модели на текущем шаге, в которой будет вызываться подпрограмма  $cF$  для расчета значений переменных, для которых записаны алгебраические уравнения ( $z$ ) и правых частей дифференциальных уравнений ( $f = dy/dt$ ). По таблице 1.1 расчетное соотношение для данного метода

$$Y_i = Y_{i-1} + \Delta t \cdot f_{i-1}.$$

Параметры данной подпрограммы:  $Y_p$  – массив значений переменных на предыдущем шаге;  $Y, Z$  – массивы рассчитываемых значений переменных на текущем шаге;  $t, dt$  – значение текущего времени и шага расчета. Внутренние переменные подпрограммы:  $F$  – массив значений правых частей дифференциальных уравнений ( $dy/dt$ );  $j$  – счетчик (номер дифференциального уравнения в системе, или номер переменной  $y_j$ ).

```

34 procedure cY(Yp: DArray; t, dt: double;
35   var Y, Z: DArray);
36 var
37   F: DArray;
38   j: integer;
39 begin
40   cF(Yp, t-dt, Z, F);
41   for j := 1 to 2 do
42     Y[j] := Yp[j] + dt*F[j];
43 end;
44
```

$$f_{i-1} = f(t - \Delta t, X(t - \Delta t), Y_{i-1})$$

$$Y_i = Y_{i-1} + \Delta t \cdot f_{i-1}$$

Составим подпрограмму для ввода исходных данных для расчета: параметров звеньев и входных воздействий, интервала и шага расчета.

```
45 procedure pIn;  
46 begin  
47   Xm[1] := StrToFloat(Xm1edit.Text);  
48   lT[1] := StrToFloat(T1edit.Text);  
49   lK[2] := StrToFloat(K2edit.Text);  
50   lT[2] := StrToFloat(T2edit.Text);  
51   lK[3] := StrToFloat(K3edit.Text);  
52   ly0[3] := StrToFloat(Y03edit.Text);  
53   tk := StrToFloat(Tkedit.Text);  
54   dt := StrToFloat(dtedit.Text);  
55 end;  
56
```

Теперь составим основную часть программы. В строке 58 вызывается подпрограмма ввода исходных данных. В строках 59...61 производится обнуление (задание нулевых начальных условий) переменных:  $Y$  – значений переменных  $y$  на текущем шаге ( $Y_i$ );  $Y_p$  – значений переменных  $y$  на предыдущем шаге ( $Y_{i-1}$ );  $Z$  – значений переменных  $z$ . В строках 62...64 очищаются диаграммы на второй форме. В строке 65 определяется количество шагов расчета. В строке 66 задается нулевое время, и в строках 67...69 выводятся значения для момента времени  $t=0$  (добавляются точки на диаграммы). В строках 70, 71, 78 организуется цикл по шагам расчета  $i$  от 1 до  $N$ . В цикле:

1. В строке 72 определяется текущее время  $t$ , соответствующее текущему шагу расчета.
2. В строке 73 осуществляется расчет значений переменных модели на текущем шаге расчета. Для этого вызывается подпрограмма  $sY$ .
3. В строках 74...76 рассчитанные значения выводятся (добавляются точки на диаграммы).
4. В строке 77 осуществляется перенос значений с текущего шага на предыдущий.

В строке 79 разрешается работа кнопки, отображающей вторую форму.

```
57 begin  
58   pIn;  
59   FillChar(Y, SizeOf(Y), 0);
```



```

60 FillChar(Yp, SizeOf(Yp), 0);
61 FillChar(Z, SizeOf(Z), 0);
62 Form2.Chart1LineSeries1.Clear;
63 Form2.Chart1LineSeries2.Clear;
64 Form2.Chart1LineSeries3.Clear;
65 N := round(tk/dt);
66 t := 0;
67 Form2.Chart1LineSeries1.AddXY(t, Y[1]);
68 Form2.Chart1LineSeries2.AddXY(t, Y[2]);
69 Form2.Chart1LineSeries3.AddXY(t, Z[2]);
70 for i := 1 to N do
71     begin
72         t := i*dt;
73         cY(Yp, t, dt, Y, Z);
74         Form2.Chart1LineSeries1.AddXY(t, Y[1]);
75         Form2.Chart1LineSeries2.AddXY(t, Y[2]);
76         Form2.Chart1LineSeries3.AddXY(t, Z[2]);
77         Yp := Y;
78     end;
79 Button2.Enabled := true;
96 end.

```

На рис. 3.2 приведены диаграммы переходных процессов.

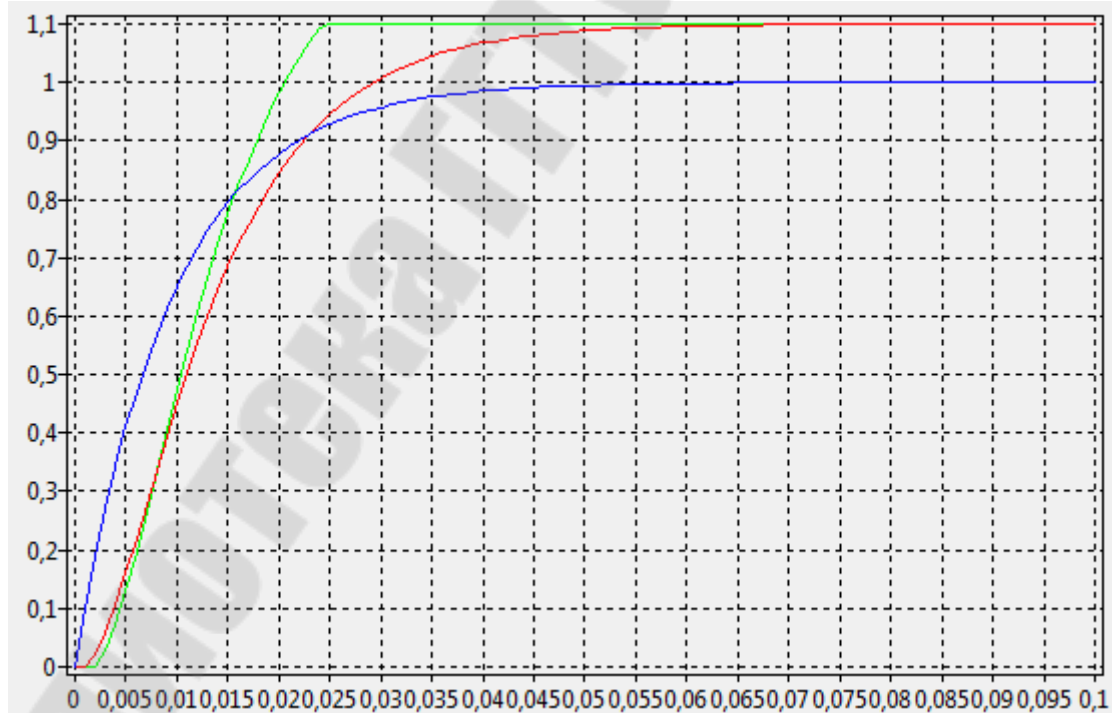


Рис. 3.2. Диаграммы переходных процессов.

### 3.2. Структурное моделирование

При структурном моделировании нет необходимости делить переменные модели на те, для которых записаны дифференциальные уравнения и те, для которых алгебраические. Поэтому обозначим переменные, как показано на рис. 3.3.

Начальная часть программы с объявлениями переменных будет отличаться от предыдущего примера отсутствием массива Z, поскольку сейчас все переменные модели обозначены y.

Подпрограмма cX будет полностью такой же, как и в предыдущем примере.

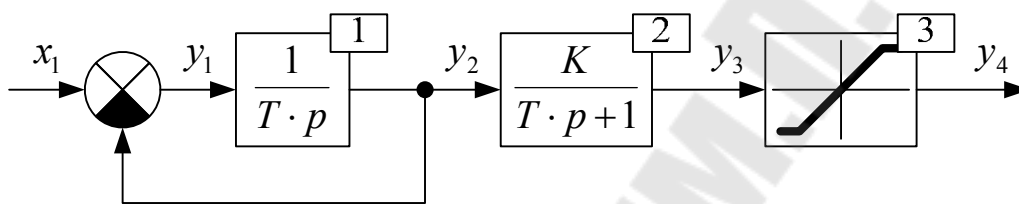
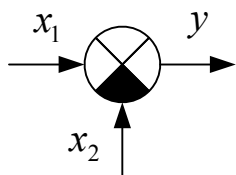


Рис. 3.3. Структурная схема.

```
001 procedure TForm1.Button1Click(Sender: TObject);
002
003 type
004   DArray = array [1..10] of double;
005
006 var
007   Y, Yp,
008   lK, lT, ly0,
009   xm: DArray;
010   tk, dt,
011   t: double;
012   i,
013   N: integer;
014
015 procedure cX(t: double; var x: DArray);
016 begin
017   if t >= 0 then x[1] := xm[1]
018   else x[1] := 0;
019 end;
020
```

Составим на основе метода Эйлера подпрограммы-функции расчета значения выходного сигнала звена на текущем шаге для всех типов звеньев, входящих в состав модели:

1. Вычитатель.



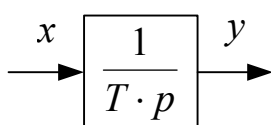
Вычитатель описывается выражением

$$y = x_1 - x_2.$$

```
021 function Sub(X1, X2: double): double;
022 begin
023     Sub := X1 - X2;
024 end;
025
```

$$y = x_1 - x_2$$

2. Интегрирующее звено.



Выходной сигнал звена в операторной форме

$$y(p) = \frac{1}{T \cdot p} \cdot x(p).$$

Составим дифференциальное уравнение для выходного сигнала, аналогично рассматриваемым ранее примерам

$$\frac{dy}{dt} = \frac{1}{T} \cdot x.$$

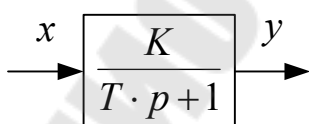
Применив метод Эйлера к данному уравнению получим

$$y_i = y_{i-1} + \Delta t \cdot f_{i-1} = y_{i-1} + \Delta t \cdot \frac{1}{T} \cdot x.$$

```
026 function Int(X, Yp, T, dt: double): double;
027 begin
028     Int := Yp + dt*X/T;
029 end;
030
```

$$y_i = y_{i-1} + \Delta t \cdot \frac{1}{T} \cdot x$$

3. Аperiodическое звено.



Выходной сигнал звена в операторной форме

$$y(p) = \frac{K}{T \cdot p + 1} \cdot x(p).$$

Составим дифференциальное уравнение для выходного сигнала

$$\frac{dy}{dt} = \frac{1}{T} \cdot (K \cdot x - y).$$

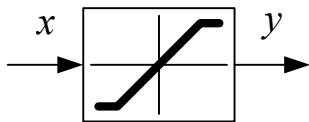
Применив метод Эйлера к данному уравнению получим

$$y_i = y_{i-1} + \Delta t \cdot f_{i-1} = y_{i-1} + \Delta t \cdot \frac{1}{T} \cdot (K \cdot x - y_{i-1}).$$

```
031 function Ap(X, Yp, K, T, dt: double): double;
032 begin
033   Ap := Yp + dt*(K*X-Yp)/T;
034 end;
035
```

$$y_i = y_{i-1} + \Delta t \cdot \frac{1}{T} \cdot (K \cdot x - y_{i-1})$$

#### 4. Звено с ограничением.



Звено с ограничением описывается выражением

$$y = K \cdot x,$$

$$y = \begin{cases} Y_0, & \text{при } y > Y_0, \\ -Y_0, & \text{при } y < -Y_0. \end{cases}$$

```
036 function Lim(X, K, y0: double): double;
037 var
038   y: double;
039 begin
040   y := K*X;
041   if y > y0 then y := y0;
042   else if y < -y0 then y := -y0;
043   Lim := y;
044 end;
045
```

$$y = K \cdot x,$$

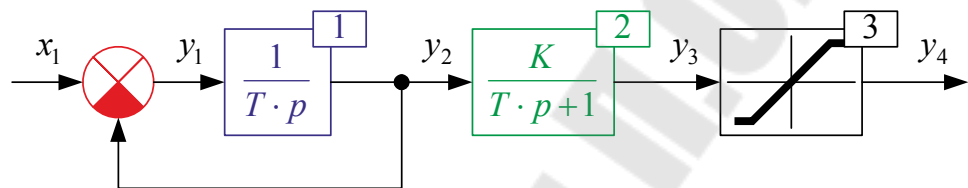
$$y = \begin{cases} Y_0, & \text{при } y > Y_0, \\ -Y_0, & \text{при } y < -Y_0. \end{cases}$$

Теперь, по структурной схеме составим подпрограмму расчета значений всех переменных модели на текущем шаге.

```

046 procedure cY(Yp: DArray; t, dt: double;
047   var Y: DArray);
048 var
049   x: DArray;
050 begin
051   cX(t, x);
052   Y[1] := Sub(x[1], Y[2]);
053   Y[2] := Int(Y[1], Yp[2], lT[1], dt);
054   Y[3] := Ap(Y[2], Yp[3], lK[2], lT[2], dt);
055   Y[4] := Lim(Y[3], lK[3], ly0[3]);
056 end;
057

```



Подпрограмма ввода исходных данных будет точно такой же, как и в прошлом примере.

В самой программе будут небольшие отличия: будет отсутствовать строка с заполнением нулями массива Z, поскольку в данном примере он отсутствует. Изменится вывод результатов расчета в соответствии с принятыми обозначениями переменных модели.

```

058 procedure pIn;
059 begin
    ...
068 end;
069
070 begin
071   pIn;
072   FillChar(Y, SizeOf(Y), 0);
073   FillChar(Yp, SizeOf(Y), 0);
074   Form2.Chart1LineSeries1.Clear;
075   Form2.Chart1LineSeries2.Clear;
076   Form2.Chart1LineSeries3.Clear;
077   N := round(tk/dt);
078   t := 0;
079   Form2.Chart1LineSeries1.AddXY(t, Y[2]);
080   Form2.Chart1LineSeries2.AddXY(t, Y[3]);
081   Form2.Chart1LineSeries3.AddXY(t, Y[4]);
082   for i := 1 to N do
083     begin
084       t := i*dt;

```

```

085     cY(Yp, t, dt, Y);
086     Form2.Chart1LineSeries1.AddXY(t, Y[2]);
087     Form2.Chart1LineSeries2.AddXY(t, Y[3]);
088     Form2.Chart1LineSeries3.AddXY(t, Y[4]);
089     Yp := Y;
090     end;
091     Button2.Enabled := true;
092 end.

```

На рис. 3.4 приведены диаграммы переходных процессов для данного примера.

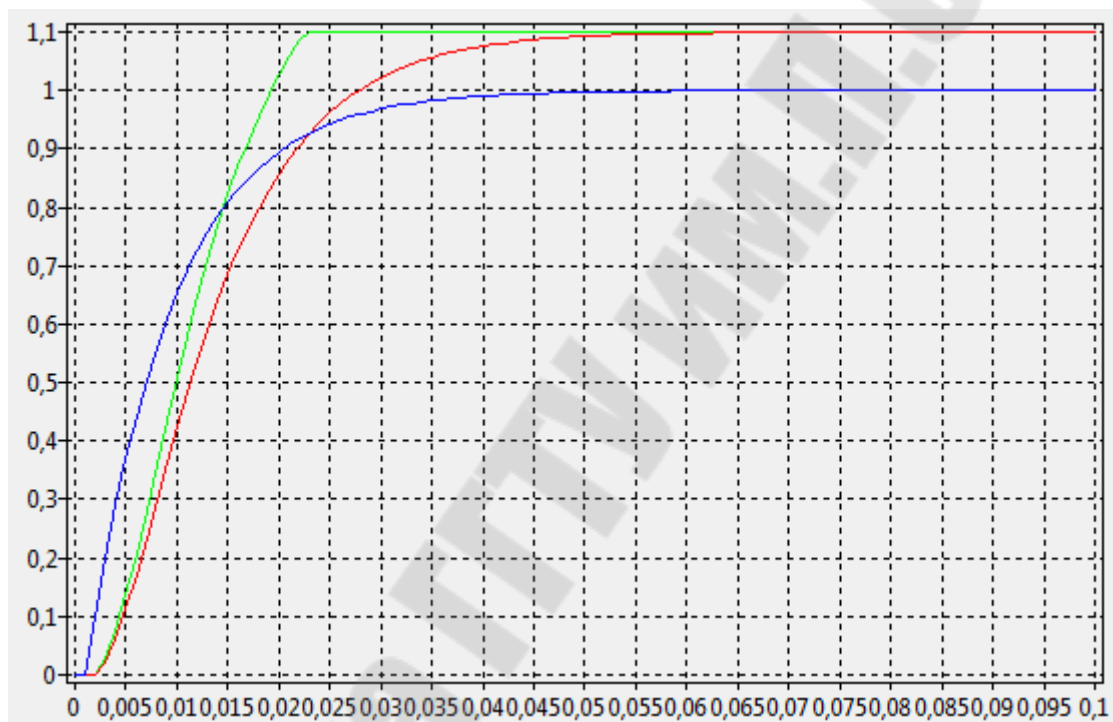


Рис. 3.4. Диаграммы переходных процессов.

## **Литература**

1. Захаренко В.С. Системы автоматического управления скоростью электроприводов постоянного и переменного тока: Практическое руководство к практическим занятиям по дисциплине “Системы управления электроприводами” для студентов специальности 1-53 01 05. – Гомель: ГГТУ им. П.О. Сухого, 2006. – 41 с.
2. Справочник по электрическим машинам: В 2 т. / Под общ. ред. И.П. Копылова и Б.К. Клокова. Т. 1. – М.: Энергоатомиздат, 1988. – 456 с.
3. Яуре А.Г., Певзнер Е.М. Крановый электропривод: Справочник. – М.: Энергоатомиздат, 1988. – 344 с.
4. Асинхронные двигатели серии 4А: Справочник / А.Э. Кравчик, М.М. Шлаф, В.И. Афонин, Е.А. Соболенская. – М.: Энергоатомиздат, 1982. – 504 с.
5. Комплектные тиристорные электроприводы: Справочник / И.Х. Евзеров, А.С. Горобец, Б.И. Мошкович и др.; Под ред. канд. техн. наук В.М. Перельмутера. – М.: Энергоатомиздат, 1988. – 319 с.
6. Макаров, Е.Г. Mathcad: учебный курс / Е.Г. Макаров. – Санкт-Петербург [и др.]: Питер, 2009 – 381 с.
7. Максфилд, Б. Mathcad в инженерных расчетах / Brent Максфилд; пер. с англ. Н.Ю. Устьян. – Санкт-Петербург: Корона-Век: Киев: МК-Пресс, 2010 – 365 с.
8. Дьяконов В.П. SIMULINK 4: спец. справочник. – Санкт-Петербург: Питер, 2002 – 528 с.
9. Потемкин. В. Вычисления в среде MATLAB. – М.: Диалог-МИФИ, 2004.

## Содержание

Введение	3
1. Теоретические сведения	4
1.1. Математическое моделирование	4
1.2. Методы численного решения дифференциальных уравнений	6
1.3. Статические звенья	8
1.4. Динамические звенья	8
1.5. Модели двигателей постоянного тока независимого и последовательного возбуждения	12
1.6. Модель асинхронного электродвигателя с короткозамкнутым ротором	14
1.7. Модели систем управления скоростью двигателя постоянного тока независимого возбуждения	17
1.8. Структурные преобразования и операции с передаточными функциями в MathCAD	20
1.9. Построение логарифмических частотных характеристик в MathCAD	23
1.10. Расчет переходных процессов	26
1.11. Основы работы с Simulink	33
1.12. Обзор наиболее часто используемых блоков MATLAB Simulink	40
1.12.1. Библиотека линейных элементов – Continuous	40
1.12.2. Библиотека математических операций – Math Operations	43
1.12.3. Библиотека приемников сигналов – Sinks	45
1.12.4. Библиотека источников сигналов – Sources	48
1.12.5. Блок решения системы дифференциальных уравнений	49
1.13. Расчет переходных процессов	52
1.14. Инструменты анализа линейных систем в MATLAB Simulink	54
1.15. Синтез регуляторов	59
2. Лабораторные работы	66
2.1. Лабораторная работа №1 Моделирование типовых статических и динамических элементов систем автоматического управления	66
2.2. Лабораторная работа №2 Моделирование электродвигателя постоянного тока независимого и последовательного возбуждения	67



2.3. Лабораторная работа №3 Моделирование асинхронного электродвигателя _____	68
2.4. Лабораторная работа №4 Моделирование системы автоматического управления скоростью двигателя постоянного тока независимого возбуждения _____	69
2.5. Лабораторная работа № 5. Структурные преобразования и построение частотных характеристик в MathCAD _____	71
2.6. Лабораторная работа № 6. Использование символьных вычислений и программирования в MathCAD _____	71
2.7. Лабораторная работа № 7. Расчет переходных процессов в Matlab Simulink _____	72
2.8. Лабораторная работа № 8. Использование инструментов анализа линейных систем в Matlab Simulink _____	72
2.9. Лабораторная работа № 9. Использование средств решения систем дифференциальных уравнений Matlab Simulink _____	72
2.10. Лабораторная работа № 10. Моделирование системы автоматического управления скоростью двигателя постоянного тока независимого возбуждения в Matlab Simulink _____	73
2.11. Лабораторная работа № 11. Синтез регуляторов в Matlab Simulink _____	73
3. Пример выполнения лабораторной работы №1 _____	75
3.1. Моделирование по системе уравнений _____	75
3.2. Структурное моделирование _____	82
Литература _____	87

**Захаренко Владимир Сергеевич**

## **МОДЕЛИРОВАНИЕ В ЭЛЕКТРОПРИВОДЕ**

**Учебно-методическое пособие  
по одноименной дисциплине  
для студентов специальности 1-53 01 05  
«Автоматизированные электроприводы»  
дневной формы обучения**

Подписано к размещению в электронную библиотеку  
ГГТУ им. П. О. Сухого в качестве электронного  
учебно-методического документа 14.09.17.

Рег. № 59Е.  
<http://www.gstu.by>