



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

---

Кафедра «Электротехники, информатики и компьютерных технологий»

В.И. БУДИН, Ф.В. ДРЕМОВ

# ТЕОРИЯ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ В СРЕДЕ МАТЛАВ

*Учебное пособие*

Самара  
Самарский государственный технический университет  
2014

Печатается по решению редакционно-издательского совета СамГТУ

УДК 681.51

**Будин В. И.**

**Теория автоматического управления в среде MATLAB:** учеб. пособие / В. И. Будин, Ф. В. Дремов. – Самара: Самар. гос. техн. ун-т, 2014. – 127 с.: ил. 36, табл. 7. Библиогр.: 9 назв.

Излагаются функциональные возможности системы MATLAB с ее основными приложениями Control System Toolbox и Simulink, которые ориентированы на решение задач теории автоматического управления. Материал сопровождается значительным количеством примеров. Приводятся контрольные вопросы и упражнения для самостоятельного решения.

Для студентов технических вузов всех форм обучения, изучающих дисциплину «Теория автоматического управления» по программам направлений бакалавриата «Автоматизация технологических процессов и производств» и «Электроэнергетика и электротехника (профиль - Электропривод и автоматика промышленных установок)». Может быть также полезно при выполнении выпускных квалификационных работ по данным направлениям.

УДК 681.51

Рецензенты: зам. начальника кафедры АРЭО ВУНЦ ВВС "ВВА"  
к.т.н. *Алексеев Э.О.*;

доцент кафедры «Электромеханика и промышленная автоматика» Сф СамГТУ к.т.н. *Горшков Р.Г.*

© В. И. Будин, Ф. В. Дремов, 2014

© Самарский государственный  
технический университет, 2014

## ПРЕДИСЛОВИЕ

Предлагаемое учебное пособие написано в соответствии с требованиями Федерального государственного образовательного стандарта для высшего профессионального образования к уровню подготовки бакалавров направлений «Автоматизация технологических процессов и производств» и «Электроэнергетика и электротехника» по дисциплине «Теория автоматического управления» с целью самостоятельного изучения пакета научно-инженерских расчетов MATLAB и его практического применения для решения задач анализа и синтеза автоматических систем, в том числе, освоение способов построения моделей динамических объектов и систем непрерывного и дискретного управления в среде MATLAB, формирование устойчивых навыков использования функций и команд приложений пакета MATLAB для исследования и проектирования систем автоматического управления во временной и частотной областях, освоение методов машинного моделирования нелинейных систем и их параметрическая оптимизация с использованием *Simulink*.

Данное пособие является дополнением к учебникам по теории автоматического управления и имеет практическую направленность. Особенно оно полезно студентам при выполнении дипломного проекта и курсовых работ, связанных с проектированием автоматических систем.

В пособии основной упор делается на получение и закрепление устойчивых навыков применения MATLAB: в каждом разделе приводится большое количество примеров, иллюстрирующих использование функций и команд MATLAB; в практической части каждой темы предлагается самостоятельное выполнение значительного количества заданий; каждая глава заканчивается объемным дополнительным упражнением и контрольными вопросами.

Начинать изучение материала пособия целесообразно с первой главы, т.к. он выстроен в соответствии с принципами последовательного перехода от известной информации к новой, от простого материала к более сложному, что позволяет наиболее быстро получить практические навыки даже начинающему пользователю MATLAB, не имеющего опыта работы в этой среде. При этом рекомендуется использовать все версии MATLAB выше 7.1, т.к. интерфейс среды более ранних версий пакета несколько отличается, что может вызвать некоторые затруднения при выполнении заданий и упражнений.

## ВВЕДЕНИЕ

Развитие теории автоматического управления (ТАУ) в последнее десятилетие показывает, что наиболее эффективные методы анализа и синтеза автоматических систем ориентированы на использование ЭВМ. В инженерной практике среди математических компьютерных систем наибольшее распространение получил пакет MATLAB, который имеет значительное количество специальных приложений (*Toolbox*) для решения задач исследования и проектирования систем автоматического управления (САУ). Среди них можно особо выделить *Control System Toolbox* и *Simulink*. Эти приложения являются основным инструментом специалистов в области машинного анализа и синтеза автоматических систем. В большинстве зарубежных изданий по ТАУ среда MATLAB используется как средство решения сложных с вычислительной точки зрения задач, встречающихся при проектировании САУ, в том числе создание математических моделей объектов и систем в целом, их преобразование, расчет динамических процессов и определение точности в установившихся режимах работы систем регулирования.

Следует отметить, что современная теория автоматического управления, в основе которой лежат математические модели управления в виде матричных дифференциальных уравнений практически не применимы без использования системы MATLAB из-за сложности вычислительных процедур.

MATLAB по существу создавался как система научно-инженерных расчетов, ядром которой являются операции над матрицами. Это предопределило аббревиатуру пакета MATrix LABoratory (матричная лаборатория). Поэтому современное направление ТАУ в виде метода пространства состояний в значительной мере опирается на вычислительно-алгоритмическое обеспечение системы MATLAB и указывает в свою очередь пути совершенствования и разработки ее новых приложений.

В области автоматизированного проектирования систем автоматического управления в настоящее время наибольшее внимание уделяется созданию в вычислительных средах динамических моделей. Поэтому в предлагаемом пособии в первую очередь рассматривается тема формирования в среде MATLAB типовых математических моделей объектов и систем с помощью приложения *Control System Toolbox*. Он позволяет создавать модели трех классов: два типа на основе передаточных функций (ПФ) и один на основе стандартной формы уравнений состояния. Все эти объекты однозначно взаимосвязаны, имеют определенные приоритеты и могут использоваться для преобразования структурных схем САУ и анализа динамических и статических свойств разрабатываемых систем управления. В этом разделе рассматриваются также вопросы дискретизации непрерывных моделей, создания моделей с запаздыванием и дискретных систем с амплитудно-импульсной модуляцией.

Второй раздел посвящен изучению и приобретению навыков исследования САУ в среде MATLAB. В основном это решение задач по оценке динамических параметров систем регулирования, расчету переходных и весовых характеристик, получению частотных характеристик разомкнутых и замкнутых САУ, в том числе в логарифмическом масштабе. Ранее это считалось сложной трудоемкой рутинной работой. Использование среды MATLAB исключает эту проблему. Приводимые функции и команды приложения *Control System Toolbox* позволяют значительно упростить решение задач анализа линейных САУ.

В третьем разделе рассматриваются вопросы использования *Simulink* для исследования нелинейных САУ. Это самое объемное приложение MATLAB, которое, по существу, с добавлением ядра MATLAB приобретает свойства самостоятельного пакета, предназначенного для моделирования динамических систем. В его основе лежат принципы визуального программирования, благодаря чему модель САУ набирается в виде схемы, состоящей из отдельных

типовых блоков. Такая структура позволяет исследовать динамические свойства автоматических систем сразу по нескольким «физическим» координатам с учетом их ограничений и влияния нелинейностей статистических характеристик отдельных звеньев. В основе вычислительных процессов моделирования в *Simulink* лежат численные методы интегрирования дифференциальных уравнений в виде разностных схем Рунге-Кутты. В качестве метода интегрирования для неопытного пользователя рекомендуется использовать схему, предлагаемую по умолчанию.

При проектировании САУ полученный регулятор, как правило, требует параметрической настройки по заданному критерию качества. Для решения этой задачи в пакет MATLAB встроено приложение *Simulink Response Optimization*, которому посвящен четвертый раздел пособия. Этот программный продукт интегрирован с *Simulink* и по существу построен на использовании его модельной схемы. Настройка осуществляется путем одновременной минимизации перерегулирования и длительности переходного процесса по вычисляемым при поиске оптимума переходным характеристикам путем их вписывания в специальную зону, формируемую по заданным показателям качества. Поиск оптимальных настроек регулятора может осуществляться сразу по нескольким параметрам одновременно. При этом учитывается неустойчивость параметров объекта управления и системы в целом.

В последнем (пятом) разделе предлагается пример проектирования следящей системы с применением рассмотренных функциональных возможностей приложений пакета MATLAB. При этом синтез управляющего устройства осуществляется с применением методики построения многоконтурных систем с подчиненным регулированием координат. На всех этапах проектирования этой САУ используется среда MATLAB.

# 1. ФОРМИРОВАНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ С ПОМОЩЬЮ ПАКЕТА CONTROL SYSTEM TOOLBOX

Пакет *Control System Toolbox* предназначен для расчета и исследования линейных систем с постоянными параметрами.

При реализации этого пакета использованы принципы объектно-ориентированного программирования. Для линейных стационарных систем введен класс так называемых *lti*-объектов (*linear time invariant object* – линейные инвариантные во времени объекты). Этот класс включает в себя три подкласса: подкласс *ss* – для описания объектов в пространстве состояний и подклассы *tf* и *zpk* – для представления объектов в виде передаточных функций (ПФ).

Модель системы, заданная в одном из этих подклассов, может быть преобразована в любой другой подкласс. Дополнительные параметры позволяют указать, является ли система непрерывной или дискретной, имеется ли запаздывание на входах системы, задать наименования входов, выходов, переменных состояния, а так же другую вспомогательную информацию о системе.

## 1.1. Формирование *tf*-объектов

Математическая модель непрерывной системы в форме *tf*-объекта (*transfer function* - передаточная функция) имеет следующий вид:

$$W(s) = \frac{b_1 s^m + b_2 s^{m-1} + \dots + b_m s + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + \dots + a_n s + a_{n+1}}, \quad (1.1)$$

где  $b_i$ ,  $a_i$  – коэффициенты операторных полиномов соответственно числителя и знаменателя ПФ;  $s$  – комплексная переменная;  $m$ ,  $n$  – соответственно порядки числителя и знаменателя ПФ.

Для создания (ввода) *tf*-модели используется функция с одноименным идентификатором:

$$w = tf(num, den),$$

где  $num = [b_1, b_2, \dots, b_{m+1}]$ ,  $den = [a_1, a_2, \dots, a_{n+1}]$  – соответственно массивы коэффициентов числителя и знаменателя ПФ вида (1.1);  $w$  – имя созданного объекта (передаточной функции) подкласса  $tf$ .

По умолчанию сформированные ПФ выводятся на экран в функции переменной  $s$ .

*Пример 1.1.* Создание объекта  $w_1$  подкласса  $tf$  по передаточной функции вида

$$W_1(s) = \frac{199,54s + 1585}{1,85 \cdot 10^{-6} s^4 + 3,0854 \cdot 10^{-3} s^3 + 1,2874 s^2 + s}.$$

Сравнивая с формой (1.1), получим

$$b_1=199,54, b_2=1585;$$

$$a_1=1,85 \cdot 10^{-6}, a_2=3,0854 \cdot 10^{-3}, a_3=1,2874, a_4=1, a_5=0.$$

В строке ввода команд набирается команда

$$\gg w1=tf([199.54, 1585],[1.85e-6, 3.0854e-3, 1.2874, 1, 0])$$

Результат отобразится в следующей форме:

Transfer function:

$$\frac{199.5 s + 1585}{1.85e-006 s^4 + 0.003085 s^3 + 1.287 s^2 + s}$$

*Задание 1.1.* Сформируйте модель  $h1$   $tf$ -подкласса по ПФ вида

$$H_1(s) = \frac{17s + 95}{0.000216s^3 + 0.045s^2 + 0.765s + 1}.$$

Для создания модели статического звена можно использовать команду  $tf(b_1, a_1)$  или  $tf(k)$ , где  $k$  – коэффициент передачи.

*Пример 1.2.* Формирование модели электронного усилителя с коэффициентом передачи  $W(s)=150$ .

```
>> w2=tf(150)
```

Transfer function:

150

*Задание 1.2.* Создайте *tf*-модель *h2* для безынерционного объекта с ПФ  $H_2(s)=11,45$ .

## 1.2. Формирование *zpk*-объектов

Математическая модель непрерывной системы в форме *zpk*-объекта (*zero-pole-gain* – нули-полюсы-коэффициент) представляет передаточную функцию вида:

$$W(s) = k \frac{(s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_n)}, \quad (1.2)$$

где  $z_i$  и  $p_i$  – соответственно нули и полюсы ПФ;  $k$  – обобщенный коэффициент передачи, связанный с коэффициентом передачи  $k$  соотношением:

$$K = k \frac{(-1)^m \prod_{i=1}^m z_i}{(-1)^n \prod_{i=1}^n p_i}.$$

Для создания *zpk*-объекта используется функция с одноименным именем:

$$W = \text{zpk}(z, p, k),$$

где  $k$  – обобщенный коэффициент передачи;  $z=[z_1, z_2, \dots, z_m]$ ,  $p=[p_1, p_2, \dots, p_n]$  – соответственно векторы нулей и полюсов ПФ вида

(1.2). При этом, если нули или полюсы отсутствуют, то в функцию вместо них вводятся пустые массивы [ ].

*Пример 1.3.* Формирование *zpk*-объекта на основе передаточной функции

$$W_3(s) = \frac{102702,7027}{s(s + 845)(s + 822)(s + 0.7782)}.$$

>> w3=zpk([], [0, -845, -822, -0.7782], 102702.7027)

Zero/pole/gain:

$$\frac{102702.7027}{s (s+845) (s+822) (s+0.7782)}$$

*Задание 1.3.* Создайте *zpk*-объект *h3* на основе передаточной функции:

$$H_3(s) = \frac{78703,7037(s + 5,588)}{(s + 189,8)(s + 17,11)(s + 1,426)}.$$

Для формирования *zpk*-модели безинерционного звена можно использовать функцию *zpk* в двух формах: стандартный вид *zpk*([], [], k) и сокращенный вариант *zpk*(k), где k - коэффициент передачи.

*Пример 1.4.* Создание *zpk*-модели статического звена с ПФ  $W_4(s)=75,3$ .

>> w4=zpk(75.3)

Zero/pole/gain:

75.3

*Задание 1.4.* Сформируйте *lti*-объект *h4* подкласса *zpk* по ПФ  $H_4(s)=0,375$ .

### 1.3. Формирование ss-объектов

Математическая модель непрерывной системы в форме *ss*-объекта (*state space* – пространство состояния) представляется в виде стандартной системы уравнений состояния:

$$\dot{X} = AX + BU, \quad (1.3a)$$

$$Y = CX + DU, \quad (1.3б)$$

где  $A, B, C, D$  – соответственно матрицы состояния, входа, выхода и обхода;  $X, U, Y$  – соответственно векторы состояния, входа и выхода.

Для формирования *ss*-объектов предназначена функция:

$$sys = ss(a, b, c, d),$$

где  $a, b, c, d$  – массивы коэффициентов одноименных матриц ( $A, B, C, D$ ); *sys* – имя объекта.

Для моделей с нулевой матрицей  $D$  (наиболее часто встречающийся вариант) можно использовать присваивание  $d=0$  как краткую форму записи нулевой матрицы соответствующих размеров.

*Пример 1.5.* Формирование *ss*-модели с именем *sys5* для динамического объекта, заданного в пространстве состояний четверкой матриц:

$$A = \begin{vmatrix} -1668 & -679,6 \\ 1024 & 0 \end{vmatrix}; \quad B = \begin{vmatrix} 8192 \\ 0 \end{vmatrix};$$

$$C = [131,70 \ 102,1]; \quad D = 0.$$

```
>> sys5=ss([-1668, -679.6; 1024, 0],[8192; 0],[131.70, 102.1], 0)
```

$$\begin{array}{r}
 a = \quad x1 \quad \quad x2 \\
 x1 \quad -1668 \quad -679.6 \\
 x2 \quad 1024 \quad \quad 0
 \end{array}$$

$$\begin{array}{r}
 b = \quad u1 \\
 x1 \quad 8192 \\
 x2 \quad 0
 \end{array}$$

$$\begin{array}{r}
 c = \quad x1 \quad \quad x2 \\
 y1 \quad 131.7 \quad 102.1
 \end{array}$$

$$\begin{array}{r}
 d = \quad u1 \\
 y1 \quad 0
 \end{array}$$

Continuous-time model.

*Задание 1.5. Создайте ss-модель h5 по математической модели, заданной матрицами:*

$$A = \begin{vmatrix} -208,3 & -13,83 & -0,5651 \\ 256 & 0 & 0 \\ 0 & 32 & 0 \end{vmatrix}; \quad B = \begin{vmatrix} 16 \\ 0 \\ 0 \end{vmatrix}; \quad C = |0 \quad 19,21 \quad 3,356|; \quad D = 0.$$

Модель ss-подкласса удобно использовать для задания матрицы коэффициентов передачи многомерного безынерционного объекта с помощью функции  $ss(D)$ .

*Пример 1.6.* Построение ss-модели для двухмерного безынерционного объекта, заданного матрицей

$$D = \begin{vmatrix} 3,6 & 0,5 \\ 1,75 & 14,3 \end{vmatrix}.$$

>> sys6=ss([3.6 0.5; 1.75 14.3])

d =	u1	u2
y1	3.6	0.5
y2	1.75	14.3

Static gain.

*Задание 1.6. Сформируйте статический объект в виде ss-модели hб с тремя входами и двумя выходами с матрицей коэффициентов передачи вида*

$$M = \begin{vmatrix} 0,37 & 25,4 & 4,73 \\ 1,02 & 13,7 & 0,27 \end{vmatrix}.$$

#### 1.4. Формирование дискретных lti-объектов

Для создания дискретной модели с заданным периодом квантования к входным аргументам функций *tf*, *zpk* и *ss* добавляется период дискретности  $T_s$ , измеряемый в секундах:

$$\begin{aligned} \text{sys} &= \text{tf}(\text{num}, \text{den}, T_s), \\ \text{sys} &= \text{zpk}(z, p, k, T_s), \\ \text{sys} &= \text{ss}(a, b, c, d, T_s). \end{aligned}$$

*Пример 1.7. Формирование дискретной модели w7 по передаточной функции*

$$W_7(z) = \frac{z + 0,7174}{2,7174z^2 - 3,7174z + 1} \quad \text{при } T_s = 0,1.$$

>> w7=tf([1 0.7174], [2.7174 -3.7174 1], 0.1)

Transfer function:

$$\frac{z + 0.7174}{2.717 z^2 - 3.717 z + 1}$$

Sampling time: 0.1

*Задание 1.7. Создайте дискретную модель h7 подкласса tf по передаточной функции  $H_7(z)$  при  $T_s=0.2$  с:*

$$H_7(z) = \frac{0,02z^2 + 0,5z + 0,025}{z^3 - 2,6z^2 + 1,9z - 0,4}.$$

Оператор  $sys=ss(a,b,c,d, T_s)$  формирует модель дискретной системы в пространстве состояний следующего вида:

$$\begin{aligned} x[(n+1) T_s] &= Ax[n T_s] + Bu[n T_s]; \\ y[nT_s] &= Cx[nT_s] + Du[n T_s], \end{aligned}$$

где  $n=0,1,2, \dots$  – целочисленный аргумент (дискретное время),  $T_s$  - период квантования, который для упрощения записи уравнений состояния часто опускают.

*Пример 1.8. Создание дискретной ss-модели с периодом квантования  $T_s = 0,1$  с, заданной четверкой матриц  $A, B, C, D$ :*

$$A = \begin{vmatrix} 1 & 0,6552 \\ 0 & 0,368 \end{vmatrix}; \quad B = \begin{vmatrix} 0 \\ 1,213 \end{vmatrix};$$

$$C = [0,7949 \quad 0,3033]; \quad D = 0$$

```
>> sys8=ss([1 0.6552; 0 0.368],[0; 1.213], [0.7949 0.3033], 0, 0.1)
```

```
a =   x1       x2
     x1   1     0.6552
     x2   0     0.368
```

```
b =   u1
     x1   0
     x2  1.213
```

$$c = \begin{matrix} x1 & x2 \\ y1 & 0.7949 & 0.3033 \end{matrix}$$

$$d = \begin{matrix} u1 \\ y1 & 0 \end{matrix}$$

Sampling time: 0.1  
Discrete-time model.

*Задание 1.8. Сформировать дискретную модель  $h8$  в пространстве состояний, заданную матрицами  $A$ ,  $B$ ,  $C$ ,  $D$  и периодом дискретности  $T_s=0,2$  с:*

$$A = \begin{vmatrix} 2,6 & -0,2375 & 0,1 \\ 8 & 0 & 0 \\ 0 & 0,5 & 0 \end{vmatrix}; \quad B = \begin{vmatrix} 0,25 \\ 0 \\ 0 \end{vmatrix};$$

$$C = [0,08 \ 0,25 \ 0,025]; \quad D=0.$$

В *Control System Toolbox* принято, что период дискретности  $T_s=0$  соответствует непрерывной системе, а значение  $T_s= -1$  - случаю, когда период квантования для дискретной системы не специфицирован.

*Пример 1.9.* Формирование дискретной *zpk*-модели без приписки значения периоду дискретности для объекта с дискретной ПФ:

$$W_9(z) = \frac{0,368(z + 0,7174)}{(z - 0,368)(z - 1)}.$$

>> w9=zpk(-0.7174, [0.368 1], 0.368, -1)

Zero/pole/gain:  $\frac{0.368 (z+0.7174)}{(z-0.368) (z-1)}$

Sampling time: unspecified

*Задание 1.9.* Сформируйте дискретную модель  $h_9$  в  $zpk$ -форме по передаточной функции  $H_9(z)$  для неспецифицированного периода дискретности  $T_s$ :

$$H_9(z) = \frac{0,02(z + 24,95)(z + 0,0501)}{(z - 1,528)(z - 0,6963)(z - 0,3761)} \cdot$$

В ряде задач дискретные передаточные функции принято записывать как рациональные выражения от аргумента  $z^{-1}$  ( $dsp$ -форма), например:

$$H(z^{-1}) = \frac{(1 + 0,5 z^{-1})}{(1 + 2 z^{-1} + 3 z^{-2})} \cdot$$

Для формирования дискретной ПФ в  $dsp$ -форме используется функция *filt*, которая по существу создает объекты подкласса *tf*. Возможны два способа обращения к функции *filt*:

$$\begin{aligned} h &= \text{filt}(\text{num}, \text{den}); \\ h &= \text{filt}(\text{num}, \text{den}, T_s). \end{aligned}$$

Первый способ используется для формирования дискретной ПФ, которой не приписывается период квантования; второй – при заданном периоде квантования  $T_s$ .

*Пример 1.10.* Создание модели  $w10$  подкласса *tf* с неспецифицированным периодом дискретности для объекта с передаточной функцией

$$W_{10}(z^{-1}) = \frac{0,368 + 0,264 z^{-1}}{1 - 1,368 z^{-1} + 0,368 z^{-2}} \cdot$$

```
>> w10=filt([0.368 0.264], [1 -1.368 0.368])
```

Transfer function: 
$$\frac{0.368 + 0.264 z^{-1}}{1 - 1.368 z^{-1} + 0.368 z^{-2}}$$

Sampling time: unspecified

!! Задание 1.10. Сформируйте дискретную модель  $h10$  в *dsp*-форме для объекта с передаточной функцией:

$$H_{10}(z^{-1}) = \frac{0,02 + 0,5z^{-1} + 0,025z^{-2}}{1 - 2,6z^{-1} + 1,9z^{-2} - 0,4z^{-3}}, \quad \text{при } T_s = 0,5\text{с.}$$

### 1.5. Построение *lti*-моделей с запаздыванием

Для непрерывных *lti*-объектов учет запаздывания входного сигнала осуществляется установлением в функциях *tf*, *zpk* и *ss* значений свойства *lti*-объекта *InputDelay* в виде дополнительной записи: (... , 'InputDelay',  $\tau$ ), где  $\tau$  – запаздывание по входу.

Для введения запаздывания выходного сигнала используется свойство *OutputDelay*. Если применить оба вида запаздывания, то в модели они будут представлены суммарным значением запаздывания.

*Пример 1.11.* Создание *tf*-модели для одномерной системы с ПФ

$$W_{11}(s) = \frac{10e^{-0,25s}}{s(0,5s + 1)}$$

>> w11=tf(10, [0.5, 1, 0], 'inputdelay', 0.25)

Transfer function:

$$\exp(-0.25*s) * \frac{10}{0.5 s^2 + s}$$

*Задание 1.11. Сформируйте zpk-модель h11 для системы с ПФ*

$$H_{11}(s) = \frac{4,4(0,5s + 1)e^{-0,15s}}{0,2s + 1} .$$

### **1.6. Построение дискретных моделей непрерывных систем**

Для преобразования непрерывных моделей в дискретные в состав *Control System Toolbox* включена функция *c2d*, основная форма обращения к которой имеет следующий вид:

$$\text{sysd}=\text{c2d}(\text{sysc}, T_s),$$

где *sysc*, *sysd* – соответственно идентификаторы исходного непрерывного и создаваемого дискретного объектов. В этой форме по умолчанию используется схема выборки с фиксатором нулевого порядка, который реализует основной способ экстраполяции квантованного сигнала в дискретно-непрерывных системах управления.

*Пример 1.12. Построение эквивалентной дискретной zpk-модели w12d для непрерывного объекта с ПФ*

$$W_{12}(s) = \frac{0,25}{(0,5s + 1)(0,25s + 1)} = \frac{2}{(s + 2)(s + 4)}$$

при  $T_s=0.15$ с и использовании экстраполятора нулевого порядка.

```
>> w12=zpk([], [-2 -4], 2); % создание zpk-модели  
>> w12d=c2d(w12, 0.15); % переход к дискретной zpk-модели  
Zero/pole/gain:
```

$$\frac{0.016794 (z+0.7408)}{(z-0.7408) (z-0.5488)}$$

Sampling time: 0.15

*Задание 1.12. Сформируйте дискретную zpk-модель h12d для непрерывного объекта с ПФ*

$$H_{12}(s) = \frac{2}{s(0,1s + 1)(0,05s + 1)}$$

*при наличии на его входе ключа с фиксатором нулевого порядка и  $T_s=0,1c$ .*

В задачах переоборудования непрерывных регуляторов в дискретные часто используется аппроксимация Тастина, которая связана с подстановкой в непрерывную передаточную функцию  $W(s)$  выражения  $s = \frac{2}{T_s} \cdot \frac{z-1}{z+1}$ . Соответствующая этому способу функция

*c2d* имеет следующую форму:

$$\text{sysd}=\text{c2d}(\text{sysc}, T_s, \text{'tustin'}),$$

где 'tustin' – идентификатор метода.

*Пример 1.13. Построение эквивалентной дискретной zpk-модели с использованием аппроксимации Тастина для непрерывного объекта с ПФ  $W_{12}(s)$  и  $T_s=0,15c$ .*

```
>> w13=c2d(w12, 0.15, 'tustin')
```

```
Zero/pole/gain:
```

$$\frac{0.0075251 (z+1)^2}{(z-0.7391) (z-0.5385)}$$

```
Sampling time: 0.15
```

*Задание 1.13. Сформируйте дискретную zpk-модель h13 для объекта с ПФ  $H_{12}(s)$  с использованием аппроксимации Тастина и  $T_s=0,1c$ .*

Если требуется согласовать характеристики непрерывной и дискретной моделей на некоторой частоте  $\omega_c$ , то применяют билинейную аппроксимацию Тастина с коррекцией

$$s = \frac{\omega_c}{\operatorname{tg} \frac{\omega_c T_s}{2}} \cdot \frac{z-1}{z+1},$$

которая реализована в следующей форме функции *c2d*:

$$\text{sysd}=\text{c2d}(\text{sysc}, T_s, \text{'prewarp'}, \omega_c),$$

где *'prewarp'* – идентификатор метода аппроксимации Тастина с коррекцией на частоте  $\omega_c$ .

*Пример 1.14.* Построение дискретной *zpk*-модели на основе ПФ  $W_{12}(s)$  с применением аппроксимации Тастина с коррекцией на частоте  $\omega_c=3\text{c}^{-1}$  при  $T_s=0,15\text{c}$ :

```
>> w14=c2d(w12, 0.15, 'prewarp', 3)
```

```
Zero/pole/gain:
```

$$\frac{0.0077383 (z+1)^2}{(z-0.7352) (z-0.5324)}$$

```
Sampling time: 0.15
```

*Задание 1.14.* Сформируйте дискретную *tf*-модель *h14* на основе  $H_{12}(s)$  с использованием аппроксимации Тастина с коррекцией на частоте  $\omega_c=5\text{c}^{-1}$  при  $T_s=0,1\text{c}$ .

Функция *c2d* позволяет строить дискретные модели для непрерывных систем с запаздыванием при условии использования в качестве экстраполятора фиксатора нулевого и первого порядков.

*Пример 1.15.* Созданный ранее объект *w11* (с запаздыванием  $\tau = 0,25$  с) можно преобразовать в дискретную *tf*-модель с  $T_s=0,15$ с следующей командой:

```
>> w15=c2d(w11, 0.15)
Transfer function:
      0.02419 z^2 + 0.2853 z + 0.07928
z^(-1) * -----
      z^3 - 1.741 z^2 + 0.7408 z
Sampling time: 0.15
```

*Задание 1.15.* Постройте эквивалентную дискретную модель *h15* на основе ранее сформированной непрерывной модели с запаздыванием *h11* при  $T_s = 0,1$ с.

## 1.7. Построение непрерывных моделей дискретных систем

Переход от дискретной модели к непрерывной осуществляется с помощью функции *d2c*:

$$\text{sysc}=\text{d2c}(\text{sysd}, \text{'method'}),$$

где *sysd*, *sysc* – соответственно дискретная и непрерывная модели, *method* – идентификатор метода экстраполяции, использованного при построении дискретного объекта: *zoh*, *tustin*, *prewarp* и др., кроме *'foh'*.

Первый вид экстраполяции применяется наиболее часто, поэтому для этого случая введена упрощенная форма функции *d2c*:

$$\text{sysc}=\text{d2c}(\text{sysd}).$$

*Пример 1.16.* Переход к непрерывной *lti*-модели от дискретной модели *w13*, созданной на основе аппроксимации Тастина:

```
>> w16=d2c(w13, 'tustin')
```

Zero/pole/gain:

$$\frac{2}{(s+2)(s+4)}$$

*Задание 1.16. Сформируйте непрерывную модель на основе ранее созданной дискретной модели h14. Сравните полученные результаты с исходной непрерывной моделью h12.*

## 1.8. Переопределение дискретной системы

Дискретная модель однозначно связана с периодом квантования  $T_s$ . Поэтому изменение частоты дискретизации в системе требует переопределения дискретной модели. Для этого в *Control System Toolbox* введена функция *d2d*, которая применяется для преобразования дискретных моделей, полученных с использованием экстраполяции нулевого порядка:

$$\text{sys}=\text{d2d}(\text{sys}, T_s),$$

где *sys* – дискретная модель;  $T_s$  – новый период квантования.

В основе алгоритма функции *d2d* лежит следующая процедура преобразований:

$$\text{sys2}=\text{d2d}(\text{d2c}(\text{sys1}), T_s),$$

где *sys1*, *sys2* – соответственно дискретные модели с исходным и новым периодом квантования  $T_s$ , т.е. сначала осуществляется переход от дискретной и непрерывной модели, а затем снова к дискретной, но уже с другим периодом квантования  $T_s$ .

*Пример 1.17. Переопределение дискретной модели w12d для периода квантования  $T_s=0,25\text{c}$ .*

```
>> w17=d2d(w12d, 0.25)
```

Zero/pole/gain:

$$\frac{0.038705 (z+0.6065)}{(z-0.6065) (z-0.3679)}$$

Sampling time: 0.25

*Задание 1.17. Преобразуйте дискретную, zpk-модель h12d в zpk - модель h17 с периодом квантования  $T_s=0,2c$ .*

Для учета запаздывания на входе дискретного объекта в его модель можно ввести задержку, кратную периоду квантования –  $\tau = NT_s$ . Для этого используется функция *d2d* в следующей форме:

$$\text{sys}=\text{d2d}(\text{sys}, [ ], N).$$

*Пример 1.18. Введение задержки  $\tau = 2T_s$  на входе объекта w12d.*

```
>> w18=d2d(w12d, [], 2)
```

Zero/pole/gain:

$$\frac{0.016794 (z+0.7408)}{z^2 (z-0.7408) (z-0.5488)}$$

Sampling time: 0.15

*!! Задание 1.18. Сформируйте задержку  $\tau= 3T_s$  на входе ранее созданного объекта h12d.*

## 1.9. Взаимное преобразование lti-объектов

Для преобразования lti-моделей из одной формы в другую применяются функции *tf*, *zpk* и *ss*, которые преобразуют произвольные модели в одноименный с функцией подкласс. Синтаксис этих функций по форме одинаковый:

<code>tsys=tf(sys)</code>	% преобразование в tf-модель;
<code>zsys=zpk(sys)</code>	% преобразование в zpk-модель;
<code>ssys=ss(sys)</code>	% преобразование в ss-модель.

При работе с *lti*-моделями следует принимать во внимание, что подклассы *lti*-моделей не эквивалентны при моделировании по точности вычислений. Так, например, точность вычислений при применении *tf*-моделей передаточных функций высокого порядка может оказаться низкой. Рекомендуется работать с масштабированными моделями в пространстве состояний, а ПФ использовать только для иллюстрации или интерпретации результатов.

*Пример 1.19.* Проверка эквивалентности преобразования *zpk*-модели  $w3$ , созданной ранее на основе  $W_3(s)$ :

```
>> w19=zpk(ss(tf(w3)))
```

Zero/pole/gain:

$$\frac{102702.7027}{s(s+845)(s+822)(s+0.7782)}$$

Результат показывает, что  $w19$  совпадает с  $w3$ .

*Задание 1.19.* Проверить эквивалентность результата функционального преобразования  $h19 = zpk(tf(ss(h12d)))$  и дискретной *zpk*-модели  $h12d$ .

## 1.10. Типовые соединения *lti*-моделей

Системы автоматического управления, как известно, часто задаются в виде структурных схем, которые представляют собой графическое изображение математической модели САУ в виде соединений элементарных динамических звеньев с ПФ низкого порядка. Если в системе отсутствуют перекрестные связи, то за счет

выделения и преобразования так называемых типовых соединений звеньев всегда можно получить единую математическую *lti*-модель САУ. В теории управления к типовым относят три элементарных соединения динамических звеньев: параллельное, последовательное и встречно-параллельное.

Для определения результирующей математической модели таких соединений в *Control System Toolbox* предусмотрены специальные функции и операции. В качестве *lti*-объектов могут использоваться все типы моделей. Однако, модели, входящие в типовые соединения, должны быть либо непрерывными, либо дискретными с одинаковым периодом квантования  $T_s$ . При этом тип результирующей модели определяется правилами предпочтения, согласно которым наивысший приоритет имеет *ss*-подкласс, затем следует подкласс *zpk* и далее объекты *tf*-подкласса. Поэтому результатом операции преобразования двух *lti*-объектов будет:

- *ss*-объект, если хотя бы один из операндов относится к *ss*-подклассу;
- *zpk*-объект, если операнды не относятся к *ss*-подклассу и хотя бы один из операндов является *zpk*-моделью;
- *tf*-объект, если оба операнда принадлежат подклассу *tf*.

Согласно этим правилам предпочтения в пакете *Control System Toolbox* перед выполнением операций все операнды более низкого приоритета переводятся в *lti*-модель операнда с наивысшим приоритетом.

Если необходимо получить результирующую модель определенного подкласса, можно предложить два пути: после выполнения операции перевести результат в требуемый подкласс, либо преобразовать в этот подкласс все операнды до выполнения операции.

Структура параллельного соединения двух моделей представлена на рис.1.1.

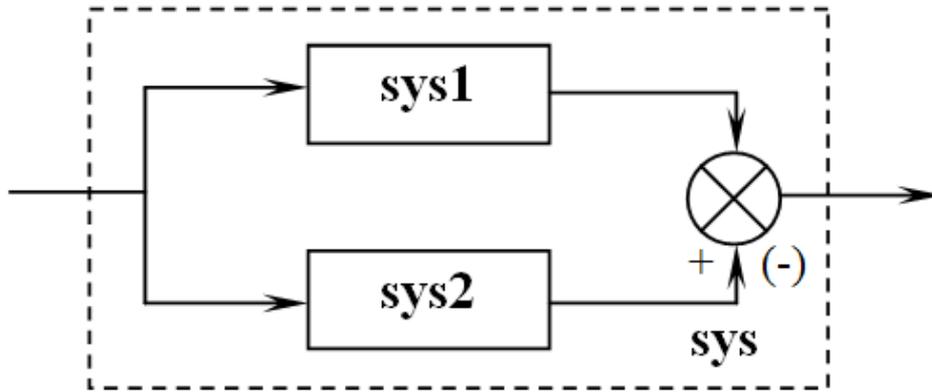


Рис. 1.1. Схема параллельного соединения

Для получения результирующей *lti*-модели *sys* при суммировании выходов можно использовать операцию сложения или эквивалентные ей функции *parallel* и *plus*, а при вычитании – *minus*:

```

sys = sys1 + sys2
sys = plus(sys1, sys2)
sys = parallel(sys1, sys2)
sys = sys1 - sys2
sys = minus(sys1, sys2)

```

*Пример 1.20.* Определение результирующей модели для параллельного соединения моделей с передаточными функциями

$$V_1(s) = \frac{10}{s+1}, \quad V_2(s) = \frac{0,1}{s(0,5s+1)}.$$

```

>> v1=tf(10, [1, 1]);
>> v2=zpk([], [0, -2], 0.2);
>> w20=v1+v2
Zero/pole/gain:

```

$$\frac{10 (s+2.01) (s+0.00995)}{s (s+2) (s+1)}$$

Задание 1.20. Сформируйте с помощью функции *parallel* результирующую *tf*-модель *h20* для параллельного соединения звеньев с ПФ:

$$G_1(s) = \frac{0,5}{s}, \quad G_2(s) = \frac{0,4s + 1}{1,2s + 1}.$$

Схема последовательного соединения двух *lti*-моделей приведена на рис.1.2.

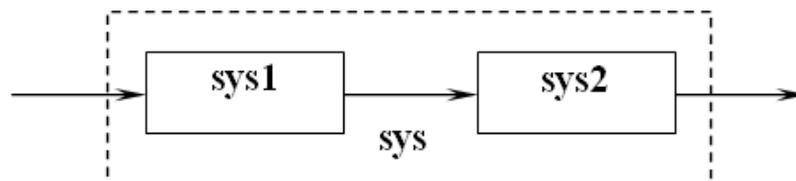


Рис.1.2. Схема последовательного соединения

Для определения результирующей модели *sys* можно использовать операцию умножения моделей *sys1* и *sys2* или две специальные функции *series* и *mtimes*:

```
sys = sys1 * sys2
sys = series (sys1, sys2)
sys = mtimes (sys1, sys2)
```

Пример 1.21. Получение результирующей модели для последовательного соединения объектов с передаточными функциями  $V_1(s)$  и  $V_2(s)$ .

```
>> w21=series(v1,v2)
```

Zero/pole/gain:

$$\frac{2}{s(s+1)(s+2)}$$

*Задание 1.21.* Сформируйте с помощью функции *mtimes* результирующую *zpk*-модель *h21* для последовательного соединения звеньев с ПФ  $G_1(s)$  и  $G_2(s)$ .

Встречно-параллельное соединение моделей представлено на рис.1.3. В общем случае обратная связь может быть как отрицательной, так и положительной.

Для получения результирующей модели этого соединения используется функция *feedback*, которая для отрицательной обратной связи может применяться в двух вариантах:

$$\begin{aligned} \text{sys} &= \text{feedback}(\text{sys1}, \text{sys2}) \\ \text{sys} &= \text{feedback}(\text{sys1}, \text{sys2}, -1), \end{aligned}$$

а при положительной обратной связи вызывается в следующем виде:

$$\text{sys} = \text{feedback}(\text{sys1}, \text{sys2}, 1),$$

где *sys1* – охватываемая модель,  
*sys2* – модель цепи обратной связи.

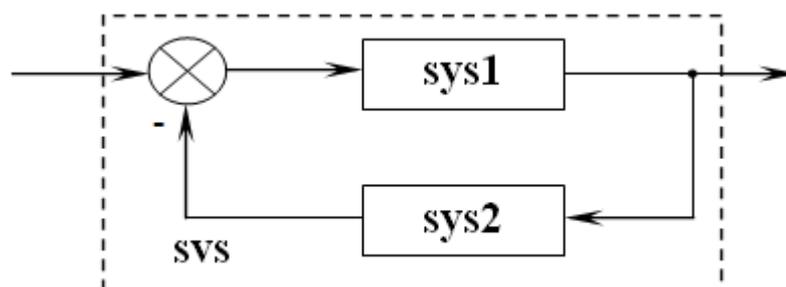


Рис. 1.3. Схема встречно-параллельного соединения

*Пример 1.22.* Определение результирующей модели для встречно-параллельного соединения звеньев  $V_1(s)$  и  $V_2(s)$ .

```
>> w22=feedback(v1,v2)
```

Zero/pole/gain:

$$\frac{10 s (s+2)}{(s+2.521) (s^2 + 0.4786s + 0.7932)}$$

Следует отметить, что в знаменателе полученной *zpk*-модели имеется оператор второго порядка. Это форма используется в MatLab при появлении в модели комплексно-сопряженных корней.

*Задание 1.22. Сформируйте результирующую ss-модель h22 для схемы с отрицательной обратной связью, в которой sys1=G<sub>1</sub>(s), sys2=G<sub>2</sub>(s).*

### 1.11. Установка и извлечение значений свойств lti-объектов

Кроме данных о параметрах модели *lti*-объекты могут содержать дополнительную информацию: название модели, ее входной и выходной координат и другие характеристики.

Вся информация о модели формируется в полях *lti*-объекта в виде так называемых свойств. Каждое свойство задается парой аргументов:

- *PropertyName* – полное или сокращенное имя свойства, достаточное для его идентификации;
- *PropertyValue* – значение свойства.

Свойства разделяются на две основных группы: родовые, которые являются общими для всех *lti*-объектов (табл.1.1), и специальные, которые присущи определенному подклассу *lti*-модели (табл. 1.2÷1.4).

Таблица 1.1

**Родовые свойства**

PropertyName	PropertyValue
InputName	Название входов
OutputName	Название выходов
Notes	Название модели
Ts	Значение периода дискретности (для дискретных систем), с
InputDelay	Вектор запаздываний на входах (для непрерывных систем)
Userdata	Произвольные дополнительные данные

Таблица 1.2

**Специальные свойства tf-моделей**

PropertyName	PropertyValue
num	Числитель ПФ
den	Знаменатель ПФ
variable	Набор альтернативных символов для обозначения переменной ПФ: 's' или 'p', 'z' или 'q'

Таблица 1.3

**Специальные свойства zpk-моделей**

PropertyName	PropertyValue
z	Нули ПФ
p	Полюсы ПФ
k	Обобщенный коэффициент передачи
variable	Набор альтернативных символов, для обозначения переменной ПФ: 's' или 'p', 'z' или 'q'

## Специальные свойства ss-объектов

PropertyName	PropertyValue
a	Матрица состояний
b	Матрица входа
c	Матрица выхода
d	Матрица обхода
StateName	Названия переменных состояния

Для извлечения данных из созданных lti-моделей используются функции *tfdata*, *zpkdata*, *ssdata* в следующей форме:

$$[\text{num}, \text{den}, T_s] = \text{tfdata}(\text{sys}, 'v')$$

$$[z, p, k, T_s] = \text{zpkdata}(\text{sys}, 'v')$$

$$[a, b, c, d, T_s] = \text{ssdata}(\text{sys}),$$

где *sys* – модель любого подкласса lti.

При получении данных непрерывных моделей параметр  $T_s$  опускают. Параметр 'v' тоже можно опустить, но он улучшает форму представления *num*, *den*, *z* и *p* (вывод в виде векторов-строк). Указанные функции применимы к моделям *sys* любого подкласса. При этом автоматически осуществляется предварительное преобразование объекта *sys* в lti-модель, определяемую типом функции.

*Пример 1.23.* Вывод параметров *tf*-объекта на примере модели *w22*:

```
>> [num, den]=tfdata(w22, 'v')
```

```
num =    0    0   10
```

```
den =   0.0432   0.3600   1.0000
```

*Задание 1.23. Проверьте параметры lti-модели для всех трех подклассов модели h22.*

Более полную характеристику lti-моделей дает функция *get*:

```
Value=get(sys, 'PropertyName')    % возвращает текущее
    значение свойства
    PropertyName
Struct= get(sys)                  % формирование
    информации о структуре
    lti-объекта в виде
    массива записей
get(sys)                          % ВЫВОД СВОЙСТВ
    lti-объекта и их текущих
    значений.
```

На практике чаще применяется последний вариант вызова функции *get*.

*Пример 1.24. Использование функции get для вывода свойств модели w22:*

```
>> get(w22)
    z: {1x1 cell}
    p: {1x1 cell}
    k: 10
Variable: 's'
DisplayFormat: 'roots'
    Ts: 0
    ioDelay: 0
    InputDelay: 0
    OutputDelay: 0
    InputName: {}
    OutputName: {}
    InputGroup: {0x2 cell}
    OutputGroup: {0x2 cell}
```

Notes: {}

UserData: []

*Задание 1.24. Просмотрите параметры модели h22 с помощью функции get.*

Для установки значений *lti*-объекта используются три способа:

- с помощью функций *tf*, *zpk*, *ss* (при создании одноименных *lti*-моделей);
- с помощью функции *set*;
- непосредственным присваиванием значений свойствам *lti*-модели.

Первый способ по существу расширяет список аргументов функций *tf*, *zpk*, *ss* парами '*Property Name*', '*Property Value*'.

*Пример 1.25.* Создание *tf*-модели по передаточной функции электродвигателя постоянного тока независимого возбуждения с передаточной функцией:

$$W_{25}(s) = \frac{N(p)}{U(p)} = \frac{2}{0,0432p^2 + 0,36p + 1},$$

где *U* – напряжение питания (вход), *N* – скорость вращения электродвигателя (выход).

```
>> w25 =tf(2, [0.0432, 0.36, 1], 'inputname', 'U', 'outputname', 'N',  
'notes', 'Электродвигатель','variable', 'p')
```

Transfer function from input "U" to output "N":

$$\frac{2}{0.0432 p^2 + 0.36 p + 1}$$

Проверка свойств модели *w25* с помощью функции *get*:

```

>> get(w25)
  num: {[0 0 2]}
  den: {[0.0432 0.36 1]}
  Variable: 'p'
  Ts: 0
  ioDelay: 0
  InputDelay: 0
  OutputDelay: 0
  InputName: {'U'}
  OutputName: {'N'}
  InputGroup: {0x2 cell}
  OutputGroup: {0x2 cell}
  Notes: {'Электродвигатель'}
  UserData: []

```

*Задание 1.25. Сформируйте ss-модель h25 следящего электропривода sys25 по его передаточной функции*

$$H_{25}(s) = \frac{0,271s^2 + 67,392s + 800}{1,44 \cdot 10^{-7} s^5 + 6,61 \cdot 10^{-5} s^4 + 0,014s^3 + 1,285s^2 + 68,392s + 800}$$

*с введением в модель следующих названий: ВЕТА - входная координата, ALPHA – выходная координата, СЛЕДЯЩИЙ ЭП - название модели. Затем проверьте установку свойств с помощью функции `struct = get (h25)`.*

Второй способ установки значений *lti*-объекта заключается в изменении его отдельных свойств с помощью команды `set`:

```
set(sys, 'Property1', Value1, 'Property2', Value2,...),
```

где `sys` – идентификатор модели.

*Пример 1.26.* Изменение названия входа и выхода модели `w25` соответственно на НАПРЯЖЕНИЕ и СКОРОСТЬ, а переменной `p` на `s`:

```
>>set(w25, 'inputname', 'НАПРЯЖЕНИЕ', 'outputname',  
'СКОРОСТЬ', 'variable', 's')
```

Проверка с помощью функции `get` подтверждает правильность сделанных изменений:

```
>> get(w25)  
    num: {[0 0 2]}  
    den: {[0.0432 0.36 1]}  
Variable: 's'  
    Ts: 0  
    ioDelay: 0  
    InputDelay: 0  
    OutputDelay: 0  
    InputName: {'НАПРЯЖЕНИЕ'}  
    OutputName: {'СКОРОСТЬ'}  
    InputGroup: {0x2 cell}  
    OutputGroup: {0x2 cell}  
    Notes: {'Электродвигатель'}  
    UserData: []
```

*Задание 1.26.* Измените название модели `h25` на СЕРВОПРИВОД и введите запаздывание по входу  $\tau=0,18$  с. Извлеките информацию о новых свойствах модели.

Третий способ установки значений свойств `lti`-объекта связан с использованием операции присваивания полям модели требуемого значения:

`sys.PropertyName = PropertyValue,`

где *sys.PropertyName* – имя свойства модели *sys*;

*PropertyValue* – значение свойства.

*Пример 1.27.* Изменение названия модели *w25* на ЭД и свойства *num* на значение 10:

```
>> w25.notes='ЭД'; w25.num=10
```

Transfer function from input "НАПРЯЖЕНИЕ" to output  
"СКОРОСТЬ":

$$\frac{10}{0.0432 s^2 + 0.36 s + 1}$$

*Задание 1.27.* Используя метод присваивания, установите в модели *h25* значение запаздывания  $\tau=0,2$  с и измените название ее входной координаты на ТЕТА. Проверьте сделанные изменения путем извлечения информации о свойствах модели.

Следует иметь ввиду, что переустановка значения периода квантования  $T_s$  в дискретной модели не изменяет ее параметры. Поэтому такая задача решается только с помощью функции *d2d*.

*Пример 1.28.* Установка в дискретной модели *w17* нового значения периода дискретности  $T_s = 0,2$  с помощью функции *set*:

```
>> w17.Ts = 0.2
```

Zero/pole/gain:

$$\frac{0.038705 (z+0.6065)}{(z-0.6065) (z-0.3679)}$$

Sampling time: 0.2

Как видно из полученного результата коэффициенты модели не изменились, хотя период  $T_s$  установился. Для получения правильного результата необходимо применить функцию  $d2d$ :

```
>> w17. Ts = 0.25;           % восстановление  $T_s$ 
```

```
>> w17d = d2d (w17, 0.2)
```

Zero/pole/gain:

$$\frac{0.027172 (z+0.6703)}{(z-0.6703) (z-0.4493)}$$

Sampling time: 0.2

Полученные результаты говорят, что изменился и период квантования  $T_s$  и соответствующие ему коэффициенты модели.

### Дополнительное упражнение

1. Построить *lti*-модели всех звеньев САУ, структурная схема которой приведена на рис.1.4, где  $k_u=2,2$ ;  $k_n=0,8$ ;  $c_0=0,9$ ;  $k_0=10$ ,  $T_{01}=0,1\text{с}$ ,  $T_{02}=1\text{с}$ .

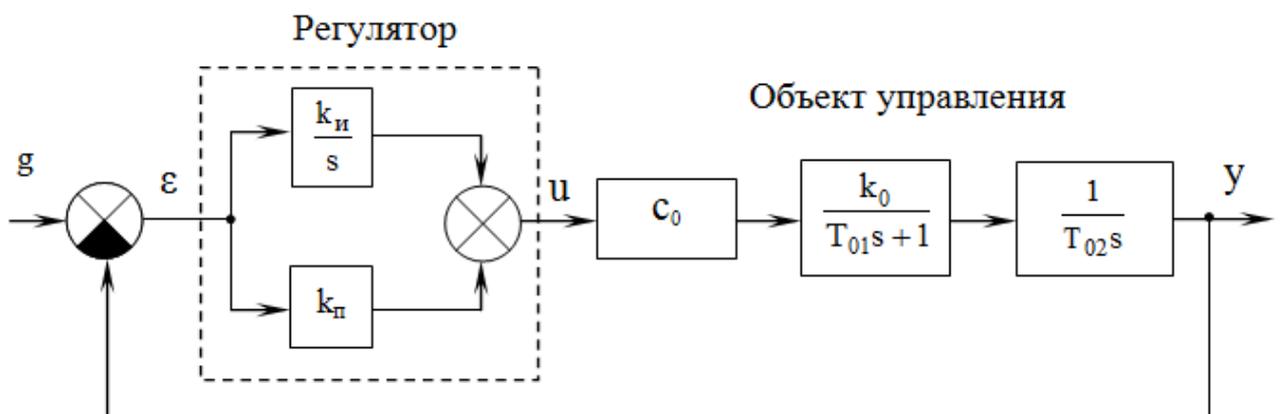


Рис. 1.4. Структурная схема САУ

2. Получить *tf*-, *zpk*- и *ss*-модели разомкнутой и замкнутой САУ.

3. Преобразовать непрерывную САУ в дискретную систему (рис. 1.5) с  $T_s=0,1$ с, переоборудовав регулятор с помощью аппроксимации Тастина в дискретное устройство управления с передаточной функцией  $D(z)$ .

4. Сформировать дискретную модель непрерывной части системы  $W(z)$  (рис.1.5), где  $H_0(s)$  – фиксатор нулевого порядка,  $W(s)$  – ПФ непрерывной части системы.

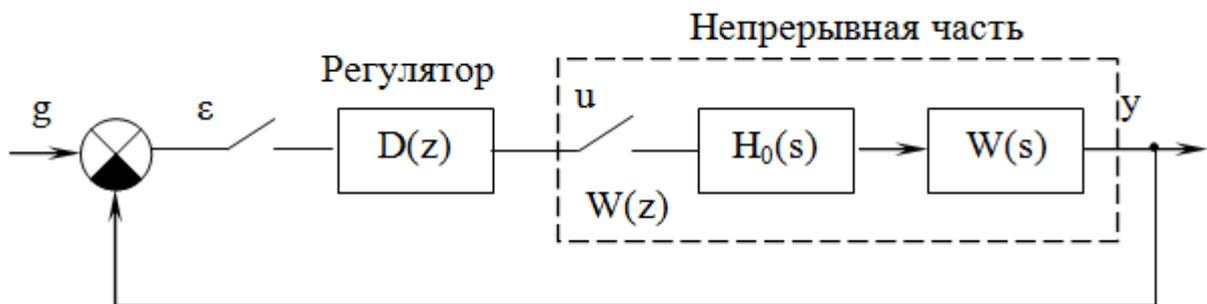


Рис. 1.5. Дискретная система управления

5. Изменить период квантования на  $T_s=0,05$ с и переопределить дискретные передаточные функции регулятора  $D(z)$  и непрерывной части системы  $W(z)$ .

6. Получить *lti*-модель замкнутой дискретной системы.

7. Ввести в модель дискретной САУ наименования координат входа и выхода и название системы "СУЛ-3".

8. Проверьте все параметры и значения свойств полученной дискретной модели замкнутой САУ.

### Контрольные вопросы

1. Какие подклассы *lti*-моделей используются в *Control System Toolbox*?
2. С помощью каких функций формируются *lti*-объекты?
3. Каким образом создать *lti*-модель статического звена?
4. Поясните особенности создания дискретных объектов.

5. Что такое *dsp*-форма дискретной *tf*-модели и как ее сформировать?
6. Как создать *lti*-модель с запаздыванием?
7. Какие функции используются для преобразования непрерывных моделей в эквивалентные дискретные *lti*-модели?
8. В чем отличие обычного преобразования Тастина от преобразования Тастина с коррекцией?
9. Как перейти от дискретной модели динамической системы к эквивалентной непрерывной *lti*-модели?
10. Поясните идею алгоритма, реализуемого в функции *d2d*?
11. Как осуществить переход от модели одного *lti*-подкласса к другому подклассу?
12. Какие правила предпочтения определяют получение результирующей модели?
13. Какие функции используются для определения результирующей модели при параллельном соединении двух *lti*-моделей?
14. Какие функции используются для определения результирующей модели схемы из двух последовательно соединенных *lti*-объектов?
15. Поясните особенности использования функции *feedback* для встречно-параллельного соединения *lti*-моделей.
16. Какие способы установки новых свойств *lti*-объекта Вы знаете?
17. Какие функции используются для извлечения данных из созданных *lti*-объектов?
18. Поясните особенности использования функции *get*.

## 2. АНАЛИЗ ЛИНЕЙНЫХ СТАЦИОНАРНЫХ СИСТЕМ С ПОМОЩЬЮ ПАКЕТА CONTROL SYSTEM TOOLBOX

Для исследования стационарных линейных систем в пакете *Control System Toolbox* включены специальные функции, которые можно условно разделить на четыре группы:

- функции, предназначенные для анализа динамических параметров *lti*-моделей, таких как полюсы и нули, коэффициенты демпфирования, частоты собственных колебаний и другие;
- функции, предназначенные для расчета и построения временных характеристик, в частности, переходной, весовой и других переходных процессов;
- функции, предназначенные для расчета и построения различного вида частотных характеристик;
- функции, предназначенные для исследования свойств и оценки параметров динамических моделей в пространстве состояний.

Для изучения всех этих функций далее будет использоваться линейная САУ, имеющая передаточные функции разомкнутой  $W(s)$  и замкнутой  $\Phi(s)$  систем следующего вида:

$$W(s) = \frac{684,3s^2 + 932,3s + 316,8}{0,002731s^7 + 0,08829s^6 + 2,467s^5 + 47,05s^4 + 277,2s^3 + 133,8s^2 + 20,65s + 1}$$

$$\Phi(s) = \frac{21,53s^2 + 29,34s + 9,969}{8,593 \cdot 10^{-6} s^7 + 2,778 \cdot 10^{-4} s^6 + 7,763 \cdot 10^{-3} s^5 + 0,148s^4 + 0,8722s^3 + 2,574s^2 + 2,998s + 1}$$

Соответствующие им *lti*-модели  $w$  и  $f$  можно сформировать с помощью функции *tf*:

```
>> w=tf([684.3 932.3 316.8], [0.002731 0.08829 2.467 47.05 277.2 133.8 20.65 1])
```

Transfer function:

$$\frac{684.3 s^2 + 932.3 s + 316.8}{0.002731 s^7 + 0.08829 s^6 + 2.467 s^5 + 47.05 s^4 + 277.2 s^3 + 133.8 s^2 + 20.65 s + 1}$$

>> f=tf([21.53 29.34 9.969], [8.593e-6 2.778e-4 7.763e-3 0.148 0.8722 2.574 2.998 1])

Transfer function:

$$\frac{21.53 s^2 + 29.34 s + 9.969}{8.593e-06s^7+0.0002778s^6+0.007763 s^5+0.148s^4+0.8722s^3+2.574 s^2+2.998s+1}$$

Аналогично для выполнения текущих заданий предлагается использовать дискретную САУ с передаточной функцией разомкнутой системы  $D(z)$  следующего вида:

$$D(z) = \frac{0,018637(z + 5,907)(z - 0,9427)(z - 0,9611)(z + 0,05765)(z^2 + 1,084z + 0,6728)}{(z - 0,9825)(z - 0,9911)(z - 1)(z - 0,3887)(z - 0,2808)(z^2 + 1,347z + 0,8581)}$$

**!!** Сформируйте сначала *tf*-модель  $d$  по ПФ разомкнутой дискретной системы  $D(z)$  с периодом квантования  $T_s=0,05$  с, а затем – *tf*-модель замкнутой дискретной системы  $T$  с единичной отрицательной обратной связью.

## 2.1. Динамические параметры *lti*-моделей

Динамические свойства систем управления определяются величинами и соотношением полюсов (собственных значений) и нулей их *lti*-моделей. Поэтому для расчета нулей и полюсов были разработаны соответствующие функции.

Для определения полюсов используются функции *pole* и *eig*, которые можно применять к *lti*-моделям любого подкласса:

$$p=pole(sys)$$
$$p=eig(sys),$$

где  $p$  – массив полюсов в виде вектор-столбца.

При этом следует иметь в виду, что обе функции дают надежный результат только для некрратных полюсов.

*Пример 2.1.* Вычисление полюсов непрерывной системы с ПФ  $\Phi(s)$ :

```
>> p1=pole(f)
p1 = -2.7524 +24.1162i
      -2.7524 -24.1162i
      -19.5359
      -2.6558 + 2.4233i
      -2.6558 - 2.4233i
      -1.4289
      -0.5474
```

*Задание 2.1.* Определите с помощью функции *eig* полюсы дискретной системы, заданной передаточной функцией  $T(z)$ .

Для определения нулей используется одна функция *tzero* в двух вариантах обращения:

```
z=tzero(sys)
[z, gain]=tzero(sys)
```

где  $z$  – вектор возвращаемых нулей *lti*-модели *sys*; *gain* – коэффициент передачи в *zpk*-модели.

*Пример 2.2.* Вычисление нулей динамической системы с ПФ  $\Phi(s)$ :

```
>> z2=tzero(f)
z2 = -0.7166
      -0.6461
```

*Задание 2.2.* Определите нули дискретной системы с ПФ  $T(z)$ .

Вместе с этими основными функциями часто используются еще две дополнительные функции для сортировки нулей и полюсов: *esort* – для непрерывных систем, *dsort* – для дискретных систем.

Функция *esort* используется для сортировки полюсов и нулей в порядке убывания их действительных частей, а функция *dsort* в порядке убывания их модулей.

Обе функции имеют две формы обращения:

$$q = \text{esort}(p), q = \text{dsort}(p) \\ [q, \text{ndx}] = \text{esort}(p), [q, \text{ndx}] = \text{dsort}(p),$$

где  $p$  – нули или полюсы *lti*-модели;  $q$  – вектор возвращаемых нулей (полюсов);  $\text{ndx}$  – вектор индексов нулей (полюсов), т.е. их порядковые номера до сортировки.

*Пример 2.3.* Сортировка полюсов и нулей непрерывной системы с ПФ  $\Phi(s)$ .

```
>> p3=esort(p1)
p3 = -0.5474
      -1.4289
      -2.6558 + 2.4233i
      -2.6558 - 2.4233i
      -2.7524 +24.1162i
      -2.7524 -24.1162i
      -19.5359

>> z3=esort(z2)
z3 = -0.6461
      -0.7166
```

*Задание 2.3.* Произведите сортировку нулей и полюсов замкнутой дискретной системы с ПФ  $T(z)$ .

Для графического отображения расположения полюсов и нулей на комплексной плоскости используется функция *pzmap*, которая имеет следующий синтаксис:

`pzmap(sys),`

где *sys* – непрерывная или дискретная *lti*-модель.

Комплексная плоскость строится по команде автоматически. Полюсы на ней изображаются маркером *x*, а нули – маркером *o*.

*Пример 2.4.* Получение схемы расположения нулей и полюсов непрерывной системы с ПФ  $\Phi(s)$  на комплексной плоскости (рис.2.1) с помощью команды *pzmap*.

```
>> pzmap(f)
```

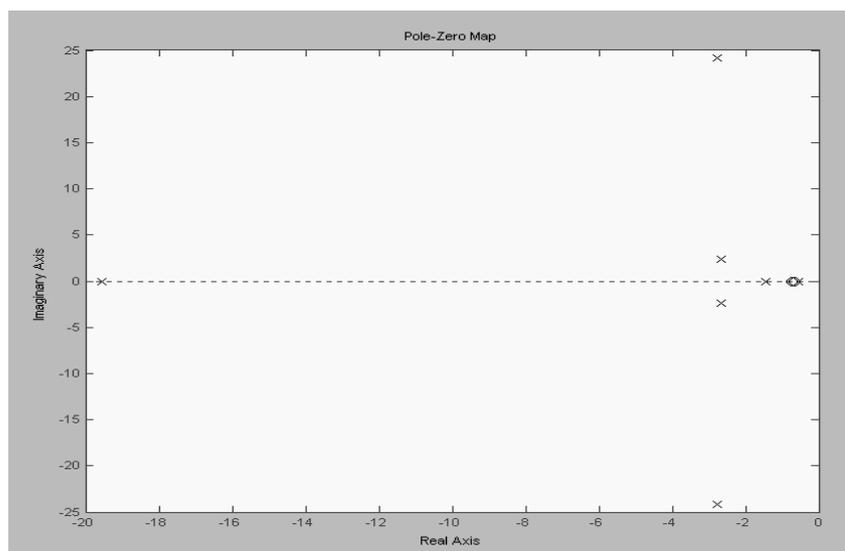


Рис. 2.1. Схема расположения нулей и полюсов

*Задание 2.4.* Постройте на *z*-плоскости план расположения нулей и полюсов дискретной системы с ПФ  $T(z)$ .

Имеется еще один вариант использования функции *pzmap*:

`[p,z]= pzmap(sys),`

при котором возвращаются значения полюсов и нулей *lti*-модели *sys* соответственно в виде векторов *p* и *z*, но без построения их схемы расположения на комплексной плоскости. Эта форма по существу объединяет действие функций *pole* и *tzero*.

*Пример 2.5.* Определение нулей и полюсов непрерывной системы с ПФ  $\Phi(s)$  с помощью функции *pzmap*:

```
>> [p5,z5]= pzmap(f)
p5 = -2.7524 +24.1162i
      -2.7524 -24.1162i
      -19.5359
      -2.6558 + 2.4233i
      -2.6558 - 2.4233i
      -1.4289
      -0.5474
z5 = -0.7166
      -0.6461
```

*Задание 2.5.* Вычислите полюсы и нули дискретной системы с ПФ  $T(z)$  с помощью функции *pzmap* и сравните их значения с ранее полученными результатами.

Если *lti*-модель *sys* содержит комплексно сопряженные полюсы, то с помощью функции *damp* можно рассчитать собственные частоты  $\omega_0$  и коэффициент демпфирования  $\xi$  соответствующего им колебательного оператора  $s^2 + 2\xi\omega_0s + \omega_0^2$ .

Функция имеет три формы обращения:

```
[W,Q]= damp(sys)
[W,Q,P]= damp(sys)
damp(sys)
```

В первом случае возвращаются значения собственных частот  $\omega_0$  и коэффициентов демпфирования  $\xi$  соответственно в виде векторов  $W$ ,  $Q$ , во втором – дополнительно полюсы  $lti$ -модели в виде вектора  $P$ , в третьем – полюсы в порядке возрастания их модуля, собственно модули и соответствующие им собственные частоты и коэффициенты демпфирования. При этом для операторов первого порядка коэффициент демпфирования принимается равным единице, а собственная частота – абсолютному значению полюса или нуля.

Для дискретной системы программа предварительно рассчитывает эквивалентные полюсы непрерывной  $lti$ -модели из соотношения  $z_i = \exp(p_i T_s)$ , где  $T_s$  – период квантования, а затем эквивалентные собственные частоты и коэффициенты демпфирования.

*Пример 2.6.* Расчет собственных частот и коэффициентов демпфирования непрерывной системы с ПФ  $\Phi(s)$ :

```
>> [W6,Q6]= damp(f)
```

W6 =	0.5474	Q6 =	1.0000
	1.4289		1.0000
	3.5952		0.7387
	3.5952		0.7387
	19.5359		1.0000
	24.2728		0.1134
	24.2728		0.1134

*Задание 2.6.* Вычислите эквивалентные значения полюсов, собственных частот и коэффициентов демпфирования дискретной системы с ПФ  $T(z)$ .

Кроме рассмотренных выше функций расчета динамических параметров имеется еще одна полезная функция  $dsgain$ , которая используется для вычисления коэффициента передачи  $lti$ -модели  $sys$ :

$$k = \text{dcgain}(\text{sys}).$$

*Пример 2.7.* Определение коэффициента передачи замкнутой САУ, заданной ПФ  $\Phi(s)$ :

```
>> k7 = dcgain(f)
k7 = 9.9690
```

*Задание 2.7.* Вычислите коэффициент передачи дискретной системы с ПФ  $T(z)$ .

## 2.2. Частотные характеристики

В теории и практике систем автоматического управления наибольшее применение находят амплитудно-фазовые частотные характеристики (АФХ или их иногда называют годографами Найквиста), логарифмические частотные характеристики (диаграммы Боде) и амплитудные частотные характеристики. Для их расчета и построения в *Control System Toolbox* включены соответствующие функции.

Построение частотного годографа Найквиста осуществляется с помощью группы функций *nyquist*:

```
nyquist(sys),
nyquist(sys,dw),
```

где  $\text{sys}$  – непрерывная или дискретная *lti*-модель любого подкласса;  
 $\text{dw} = \{W_{\min}, W_{\max}\}$  – диапазон частот;  
 $W_{\min}, W_{\max}$  – соответственно минимальная и максимальная частоты.

Частотный диапазон можно также задать с помощью массива конкретных частот  $\text{dw} = [w_1, w_2, \dots, w_n]$ , где  $n$  – число частот.

Для создания логарифмически распределенного вектора частот используется команда *logspace*. Эта функция возвращает вектор-строку логарифмически возрастающих частот:

```
logspace(log10(Wmin), log10(Wmax))  
logspace(log10(Wmin), log10(Wmax), N),
```

где  $N$  – число точек в диапазоне частот (по умолчанию  $N=50$ ).

Основной функцией является *nyquist(sys)*, в которой частотный диапазон определяется автоматически по минимальному и максимальному значениям и массива нулей и полюсов *lti*-модели. Если последняя содержит астатизм, то выдается предупреждающее сообщение, что на нулевой частоте характеристика равна бесконечности (неопределенное значение). Поэтому в этом случае целесообразно использовать функцию вида *nyquist(sys,dw)*. В практических задачах функция *nyquist* наиболее часто применяется для построения годографа разомкнутой САУ с целью исследования устойчивости замкнутой системы с помощью критерия Найквиста.

Для дискретных систем из-за периодичности частотной характеристики АФХ рассчитывается в диапазоне от нуля до частоты Найквиста ( $\omega_s = \pm\pi/T_s$ ). Если период  $T_s$  неспецифицирован, то по умолчанию принимается  $T_s = 1$ .

*Пример 2.8.* Построение АФХ разомкнутой системы с ПФ  $W(s)$  (рис.2.2).

```
>> nyquist(w)
```

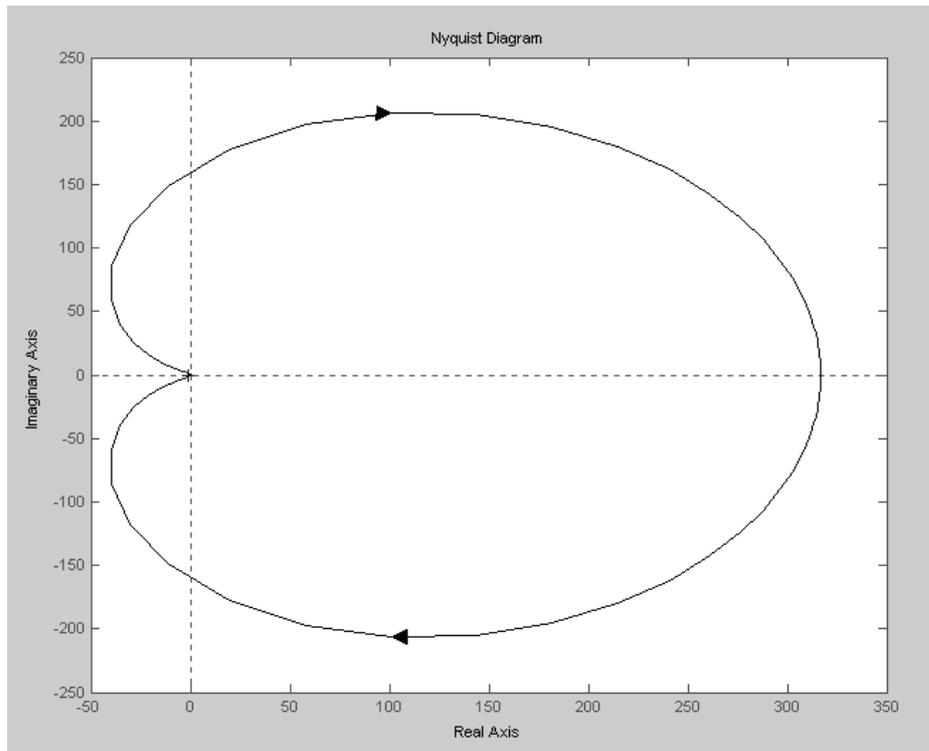


Рис. 2.2. Амплитудно-фазовая характеристика разомкнутой системы

*Задание 2.8. Постройте годограф Найквиста дискретной САУ по ПФ разомкнутой системы  $D(z)$ .*

Если требуется построить годографы Найквиста для нескольких систем в одних координатах, то можно использовать функцию *nyquist* в более общих формах:

```
nyquist(sys1, sys2,... ,sysN)
nyquist(sys1, sys2,... ,sysN, dw)
nyquist(sys1, 'plotstyle1',... ,sysN, 'plotstyleN'),
```

где *'plotstyleK'* – аргумент, определяющий стиль линии, т.е. цвет и тип годографа *K*-ой *lti*-модели.

*Примечание.* Задание стиля осуществляется в соответствии с правилами оформления графиков в MATLAB – в виде набора трех символьных маркеров, заключенных в апострофы. Один из них определяет тип линии (табл. 2.1), другой – цвет (табл. 2.2). При этом

можно указывать не все маркеры. В этом случае действуют маркеры, установленные по умолчанию. Порядок маркеров в строке стиля не регламентирован, т.е. может быть произвольным.

Таблица 2.1

**Тип линии**

Маркер	Тип линии
—	непрерывная
- -	штриховая
:	пунктирная
-. .	штрихпунктирная

Таблица 2.2

**Цвет линии**

Маркер	Цвет линии
c	голубой ( <b>cyan</b> )
m	фиолетовый ( <b>magenta</b> )
y	желтый ( <b>yellow</b> )
r	красный ( <b>red</b> )
g	зеленый ( <b>green</b> )
b	синий ( <b>blue</b> )
w	белый ( <b>white</b> )
k	черный ( <b>black</b> )

*Пример 2.9.* Построение АФХ разомкнутой системы соответственно с ПФ  $W(s)$  и  $0.5W(s)$  в одних координатах комплексной плоскости с пометкой годографа  $W(s)$  маркером 'r-', а годографа  $0.5W(s)$  - маркером ':b' (рис.2.3).

```
>> nyquist(w,'r-',0.5*w,':b')
```

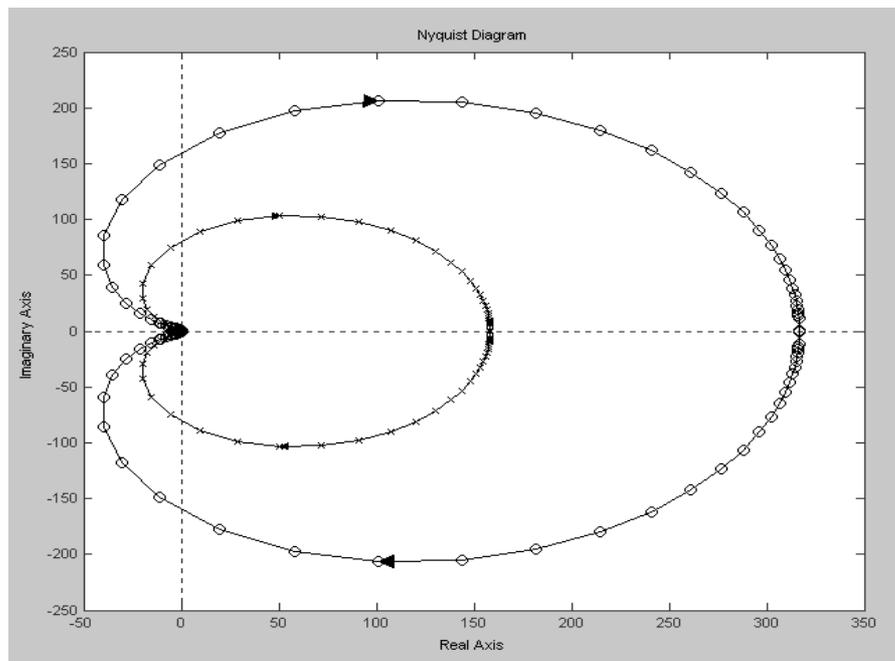


Рис. 2.3. Маркированные годографы Найквиста

*Задание 2.9. Постройте годографы Найквиста для разомкнутой дискретной системы с ПФ  $D(z)$  и  $0.7D(z)$ , пометив годограф  $D(z)$  маркером ‘-g’, а годограф  $0.7D(z)$  – маркером ‘m-’.*

Имеются две формы вызова функции *nyquist*, которые применяются для расчета годографа Найквиста без его построения:

$$[re,im,w]= nyquist(sys)$$

$$[re,im,w]= nyquist(sys,dw)$$

где *re,im* – соответственно векторы значений вещественной и мнимой частей частотной передаточной функции; *w* – вектор вычисленных частот (при необходимости его можно опустить).

*Пример 2.10.* Расчет значений действительной и мнимой частотных характеристик разомкнутой системы с ПФ  $W(s)$  в частотном диапазоне 1 - 30 рад/с.

```
>> [Re,Im]= nyquist(w,logspace(log10(1),log10(30),5))
```

Re(:, :, 1) = -2.8548	Im(:, :, 1) = -2.4108
Re(:, :, 2) = -0.8272	Im(:, :, 2) = -0.8645
Re(:, :, 3) = -0.3703	Im(:, :, 3) = -0.2049
Re(:, :, 4) = -0.1267	Im(:, :, 4) = 0.0310
Re(:, :, 5) = 0.0214	Im(:, :, 5) = -0.0070

*Задание 2.10.* Вычислите значения параметров АФХ для разомкнутой дискретной системы с ПФ  $D(z)$  в диапазоне частот от 0 до  $\pi/T_s$ .

Построение логарифмических частотных характеристик (ЛЧХ) осуществляется с помощью функции *bode*, которая как и функция *nyquist* имеет несколько форм обращения:

```

bode(sys)
bode(sys,dw)
bode (sys1, sys2,... ,sysN)
bode (sys1, sys2,... ,sysN, dw)
bode (sys1, 'plotstyle1',... ,sysN, 'plotstyleN')
[mag, phase, w]= bode(sys)

```

где *sys* – непрерывная или дискретная *lti*-модель; *mag*, *phase*, *w* – соответственно векторы амплитуд, фаз (в градусах) и частот (рад/с).

Все команды, кроме последней, осуществляют построение диаграмм Боде, представляющих совокупность двух частотных характеристик в логарифмическом масштабе: логарифмической амплитудно-частотной характеристики (ЛАЧХ) и логарифмической фазо-частотной характеристики (ЛФЧХ).

*Пример 2.11.* Построение логарифмических частотных характеристик для системы, заданной ПФ  $W(s)$  (рис.2.4), где команда *grid on* использована для нанесения сетки на график.

>> bode(w); grid on % Здесь команда *grid on* использована для % нанесения сетки на график

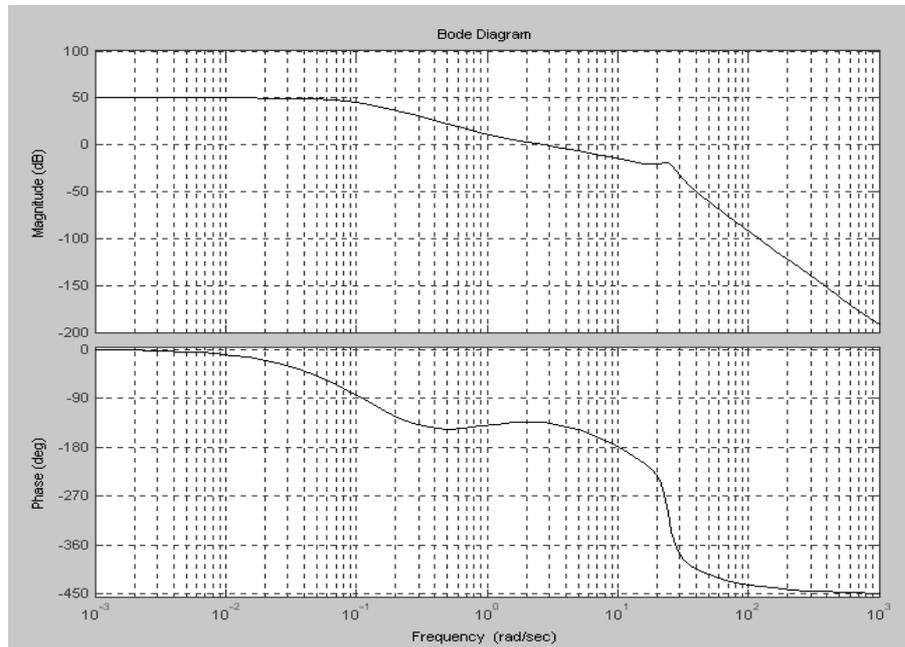


Рис. 2.4. Логарифмические частотные характеристики системы

*Задание 2.11. Постройте диаграмму Бode для разомкнутой дискретной системы с ПФ  $D(z)$ .*

Использование других вариантов функции *bode* аналогично функции *nyquist*.

*Пример 2.12. Расчет параметров ЛЧХ для разомкнутой системы с ПФ  $W(s)$ .*

```
>> [Mag12,Phase12,W]=bode(w,[0.5,1,4,8,17])
```

```
Mag12(:,:,1) = 12.7870
```

```
Mag12(:,:,2) = 3.7365
```

```
Mag12(:,:,3) = 0.6285
```

```
Mag12(:,:,4) = 0.2531
```

```
Mag12(:,:,5) = 0.0974
```

$$\text{Phase12}(:, :, 1) = -146.7687$$

$$\text{Phase12}(:, :, 2) = -139.8197$$

$$\text{Phase12}(:, :, 3) = -141.4908$$

$$\text{Phase12}(:, :, 4) = -167.2469$$

$$\text{Phase12}(:, :, 5) = -214.7002$$

$$W = 0.5000$$

$$1.0000$$

$$4.0000$$

$$8.0000$$

$$17.0000$$

*Задание 2.12. Постройте таблицу значений амплитуды и фазы от частоты для разомкнутой дискретной системы с ПФ  $D(z)$ .*

Очень часто требуется определять запасы устойчивости системы по амплитуде (модулю) и фазе. Из теории автоматического управления известно, что запас устойчивости по модулю равен значению амплитудной частотной характеристики (АЧХ) на частоте  $W_c$ , при которой фазовая частотная характеристика (ФЧХ) имеет значение  $-180^0$ , а запас по фазе – значение разности между ФЧХ и  $-180^0$  на частоте среза  $W_{cp}$ . Для определения этих параметров используется функция *margin*:

`margin(sys)`

`[Gm, Pm, Wc, Wcp]= margin(sys)`

Здесь *sys* – непрерывная или дискретная *lti*-модель;

*Gm* – запас по модулю на частоте  $W_c$ ;

*Pm* – запас по фазе на частоте среза  $W_{cp}$ .

Команда  $margin(sys)$  строит диаграмму Бode с указанием запасов устойчивости, а функция  $[Gm, Pm, Wc, Wcp]=margin(sys)$  рассчитывает значения  $Gm$ ,  $Pm$ ,  $Wc$  и  $Wcp$  без построения логарифмических частотных характеристик.

*Пример 2.13.* Определение запасов устойчивости непрерывной системы с ПФ разомкнутой системы  $W(s)$  (рис. 2.5).

```
>> margin(w)
```

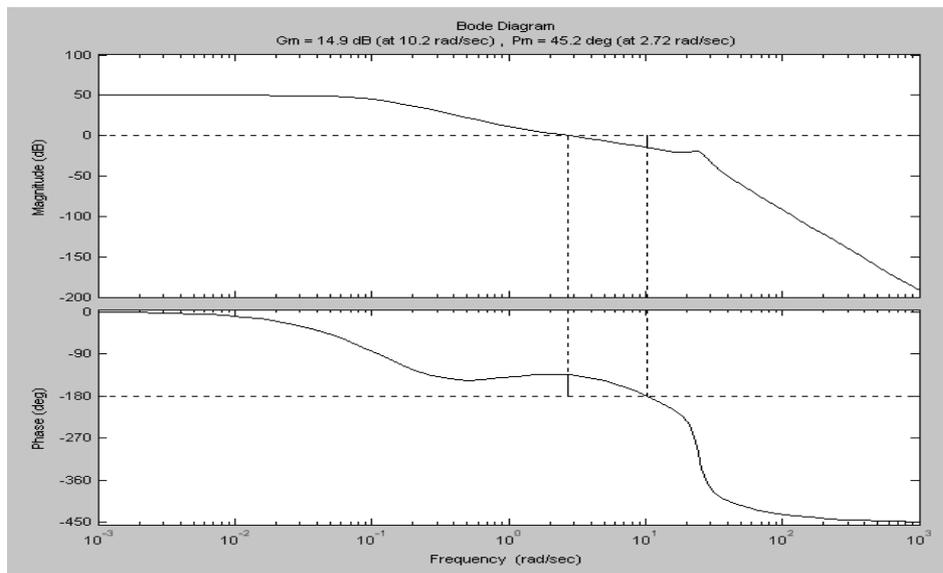


Рис. 2.5. Диаграмма Бode, построенная с помощью функций  $margin$

*Задание 2.13.* Рассчитайте значения запасов устойчивости по модулю и фазе дискретной системы с ПФ разомкнутой системы  $D(z)$  без построения ЛЧХ.

При проектировании САУ часто используется показатель колебательности, который определяется в виде отношения максимального значения модуля частотной характеристики замкнутой системы к его величине на нулевой частоте (коэффициенту передачи системы).

Максимальное значение модуля частотной характеристики связано с нормой типа  $\|\cdot\|_{\infty}$

- для непрерывных систем соотношением

$$\|W(s)\|_{\infty} = \max_{\omega} |W(j\omega)|,$$

- для дискретных систем - соотношением

$$\|H(z)\|_{\infty} = \max_{\omega \in (0, \pi/T_s)} |H(e^{j\omega T_s})|.$$

В связи с этим была составлена и включена в пакет *Control System Toolbox* функция *norm* в следующих формах:

$$\begin{aligned} \text{normsys} &= \text{norm}(\text{sys}, \text{inf}) \\ \text{normsys} &= \text{norm}(\text{sys}, \text{inf}, \text{tol}) \\ [\text{ninf}, \text{fpeak}] &= \text{norm}(\text{sys}, \text{inf}), \end{aligned}$$

где *normsys* – максимальное значение АЧХ lti-модели *sys*;

*inf* – идентификатор нормы  $\|\cdot\|_{\infty}$ ;

*tol* – точность расчета нормы (по умолчанию *tol*=1e-2);

*ninf* - максимальное значение модуля частной характеристики на частоте *fpeak*.

При этом необходимо учитывать, что если объект *sys* имеет астатизм, то максимальное значение модуля АЧХ равно бесконечности.

*Пример 2.14.* Расчет максимального значения АЧХ динамической системы с ПФ замкнутой системы  $\Phi(s)$ .

```
>> [Am, Fm]=norm(f, inf)
Am =
13.7209
Fm =
1.7976
```

*Задание 2.14. Определите значение показателя колебательности дискретной САУ с ПФ замкнутой системы  $T(z)$ , учитывая, что ее коэффициент передачи равен единице.*

### 2.3. Временные характеристики

Для исследования динамических свойств систем автоматического управления часто используются типовые переходные режимы работы (временные характеристики). Наибольшее применение находят следующие временные характеристики:

- переходная функция (характеристика) – реакция системы на единичный скачкообразный сигнал при нулевых начальных условиях;
- весовая функция (импульсная переходная характеристика) – реакция непрерывной системы на воздействие в виде  $\delta$ -импульса Дирака, а для дискретной системы – на единичный импульс.
- реакция системы на ненулевые начальные условия.

По временным характеристикам можно определить ряд качественных показателей, характеризующих динамические свойства системы управления, в частности, перерегулирование, время переходного процесса и др.

В пакете *Control System Toolbox* имеется ряд функций для расчета и построения временных характеристик линейных систем. Основными из них являются *step*, *impulse*, *initial*.

Функция *step* вычисляет и отображает на экране переходную характеристику *lti*-модели. Она имеет несколько форм:

`step (sys)`

`step (sys,t)`

`step (sys1, sys2,... ,sysN)`

`step (sys1, sys2,... ,sysN, t)`

`step (sys1, 'plotstyle1',... ,sysN, 'plotstyleN')`

`step (sys1, 'plotstyle1',... ,sysN, 'plotstyleN', t).`

Здесь  $t$  – аргумент, задающий время моделирования в виде длительности переходного режима  $t=T_{final}$  (в секундах) или в виде вектора  $t=0:dt:T_{final}$ . Для дискретных моделей  $dt=T_s$ , а для непрерывных выбирается достаточно малым для учета быстрых изменений переходного процесса. Продолжительность моделирования при отсутствии аргумента  $t$  определяется автоматически. Остальные параметры функции *step* такие же, как в функциях *nyquist* и *bode*.

*Пример 2.15.* Построение переходной характеристики для непрерывной САУ с ПФ  $\Phi(s)$  (рис.2.6).

```
>> step(f); grid on
```

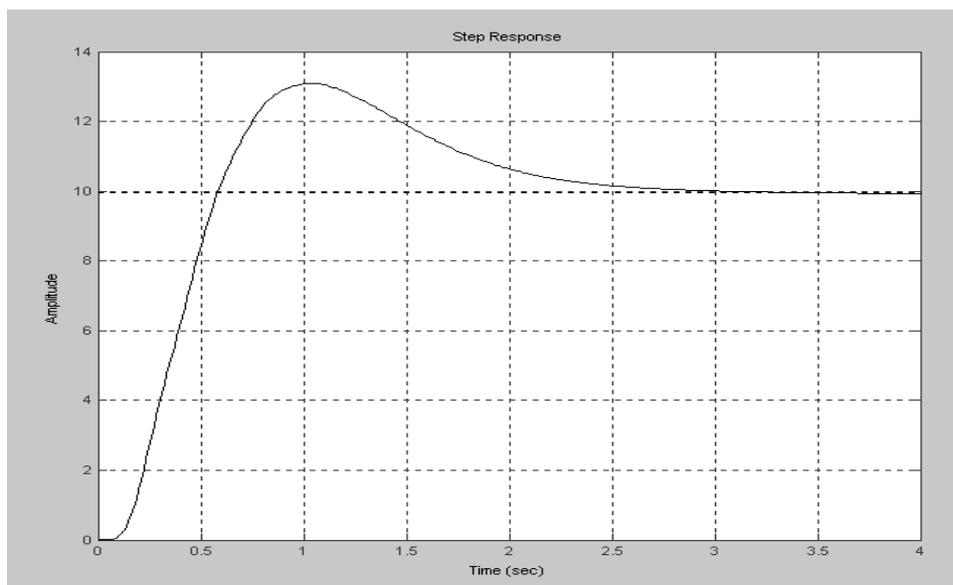


Рис. 2.6. Переходная характеристика

*Задание 2.15.* Постройте переходную функцию для дискретной САУ, заданной передаточной функцией замкнутой системы  $T(z)$ .

Для расчета переходной характеристики без ее построения используются две дополнительные формы функции *step*:

$$[y, t, x] = \text{step}(\text{sys})$$

$$[y, t, x] = \text{step}(\text{sys}, t)$$

В этом случае функция *step* возвращает векторы значений выхода  $y$ , времени  $t$  и переменных состояния  $x$  (количество столбцов вектора  $x$  равно размерности *lti*-модели). Если модель представлена в *tf* или *zpk* формах, то параметр  $x$  опускается.

*Пример 2.16.* Расчет переходной характеристики с шагом 0.2 с без ее построения для непрерывной САУ с ПФ замкнутой системы  $\Phi(s)$ .

```
>> [y,t]=step(f,[0:0.2:4])
```

y =	0	t =	0
	1.4753		0.2000
	6.1814		0.4000
	10.2384		0.6000
	12.4296		0.8000
	13.0739		1.0000
	12.8339		1.2000
	12.2280		1.4000
	11.5721		1.6000
	11.0248		1.8000
	10.6314		2.0000
	10.3729		2.2000
	10.2107		2.4000
	10.1104		2.6000
	10.0479		2.8000
	10.0075		3.0000
	9.9800		3.2000
	9.9607		3.4000
	9.9471		3.6000
	9.9379		3.8000
	9.9321		4.0000

*Задание 2.16.* Произведите вычисление переходной функции дискретной системы с ПФ замкнутой САУ  $T(z)$  без ее построения.

Для расчета и построения импульсной переходной (весовой) характеристики используется функция *impulse* в таких же формах, как и функция *step*:

```
impulse (sys)
impulse (sys,t)
impulse (sys1, sys2,... ,sysN)
impulse (sys1, sys2,... ,sysN, t)
impulse (sys1, 'plotstyle1',... ,sysN, 'plotstyleN')
impulse (sys1, 'plotstyle1',... ,sysN, 'plotstyleN', t)
[y, t, x]= impulse (sys)
[y, t, x]= impulse (sys, t)
```

*Пример 2.17.* Построение весовой характеристики для линейной непрерывной САУ с ПФ замкнутой системы  $\Phi(s)$  (рис. 2.7).

```
>>impulse(f); grid on
```

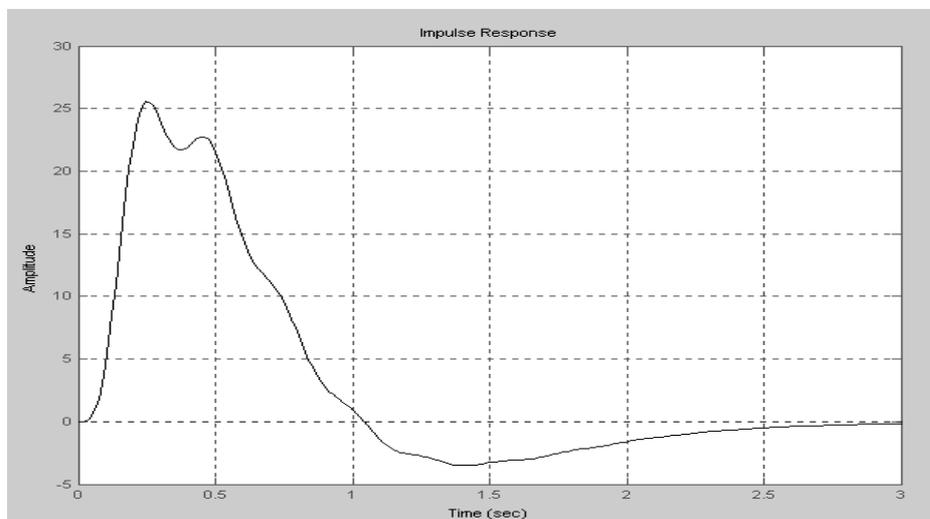


Рис. 2.7. Весовая характеристика

*Задание 2.17.* Постройте импульсную переходную характеристику для дискретной системы с ПФ замкнутой САУ  $T(z)$ .

Функция *initial* рассчитывает и строит график реакции *ss*-модели *sys* на ненулевые начальные условия. Она имеет несколько форм, эквивалентных функциям *step* и *impulse*:

```
initial (sys,x0)
initial (sys,x0,t)
initial (sys1, sys2,... ,sysN,x0)
initial (sys1, sys2,... ,sysN,x0, t)
[y, t, x]= initial (sys,x0)
[y, t, x]= initial (sys,x0, t)
```

Здесь  $x_0$  – вектор начальных условий *lti*-модели в пространстве состояний. Смысл остальных аргументов и возвращаемых величин такой же, как у функций *step* и *impulse*.

*Пример 2.18.* Построение реакции непрерывной системы, заданной математической моделью в пространстве состояний на единичное начальное условие по последней переменной состояния, т.е.  $x(0)=[0\ 0\ 0\ 0\ 0\ 0\ 1]$  (рис. 2.8).

```
>> sys 18=ss(f); x0=[0 0 0 0 0 0 1];initial(sys 18,x0); grid on
```

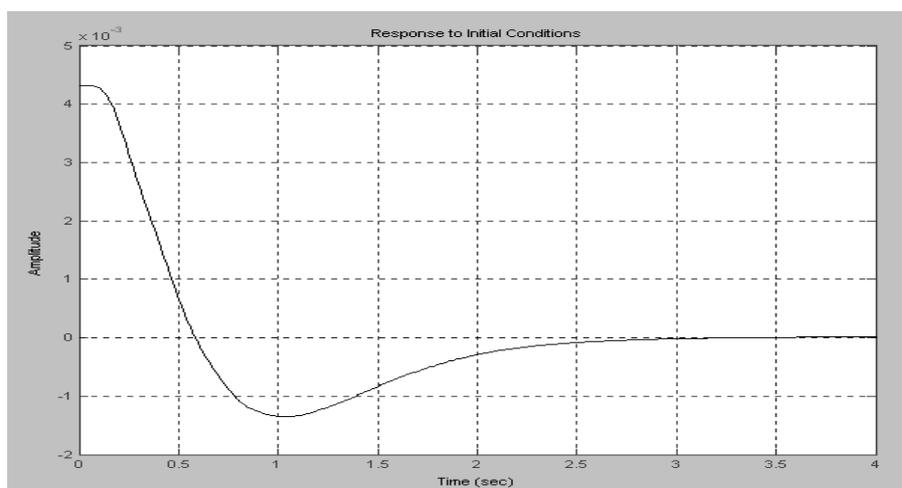


Рис. 2.8. Реакция по переменной состояния

*Задание 2.18. Постройте график реакции дискретной системы с исходной ПФ  $T(z)$  и представленной ss-моделью на единичное начальное условие по второй переменной состояния  $x(0)=[0\ 1\ 0\ 0\ 0\ 0]$ .*

К блоку временных функций пакета *Control System Toolbox* относится еще две функции: *lsim* – вычисление переходного процесса при произвольном входном воздействии, который формируется с помощью функции *gensig* – генерация периодического сигнала заданного типа (синусоида, импульсная последовательность и прямоугольный периодический сигнал). Однако на практике они используются редко, так как для моделирования систем при сложных воздействиях более удобен пакет *Simulink*.

### *Дополнительное упражнение*

1. Открыть *mat*-файл из созданного в результате выполнения дополнительного задания в разделе 1.

2. Используя *lti*-модели, полученные для разомкнутой и замкнутой непрерывной САУ, выполнить следующие виды работ.

2.1. Определить полюсы и нули замкнутой системы.

2.2. Осуществить сортировку полюсов и нулей.

2.3. Построить схему расположения нулей и полюсов на комплексной плоскости.

2.4. Произвести расчет собственных частот и коэффициентов демпфирования замкнутой системы.

2.5. Построить годограф Найквиста разомкнутой системы и оценить по нему устойчивость замкнутой системы.

2.6. Построить ЛЧХ разомкнутой системы и оценить запас устойчивости по фазе.

2.7. Определить запас устойчивости по модулю.

2.8. Рассчитать показатель колебательности системы.

2.9. Построить переходную характеристику системы.

2.10. Построить весовую характеристику системы.

3. Повторить пункты 2.1 ÷ 2.10 для дискретного варианта САУ.

## Контрольные вопросы

1. Назовите основные функции, которые используются для анализа динамических параметров *lti*-моделей?
2. Поясните действие функций *esort* и *dsort*?
3. С помощью каких функций можно определить полюсы и нули динамической системы?
4. Какие основные функции используются для расчета и построения временных характеристик?
5. Каким образом рассчитываются временные характеристики без их построения?
6. Как задается время расчета переходного процесса?
7. Какие функции служат для построения частотных характеристик динамических систем?
8. Как реализовать логарифмическое изменение частоты?
9. Дайте сравнительную оценку функциям *nyquist* и *bode*.
10. С помощью какой функции строится схема расположения нулей и полюсов на комплексной плоскости?
11. Поясните особенности форм обращения к функции *damp*.
12. Как определить коэффициент передачи системы?
13. С помощью какой функции можно оценить запас устойчивости САУ?
14. Как рассчитать показатель колебательности системы?
15. Поясните особенности использования функции *initial* и задания в ней начальных условий.

### 3. МОДЕЛИРОВАНИЕ НЕПРЕРЫВНЫХ СИСТЕМ В СРЕДЕ MATLAB С ПОМОЩЬЮ ПАКЕТА SIMULINK

Пакет прикладных программ SIMULINK (от англ. – симуляция, имитация) является расширением системы научно-технических вычислений MATLAB и предназначен для моделирования линейных и нелинейных динамических систем. В состав системы MATLAB 7.1 входит приложение SIMULINK 6.3. Это новая существенно доработанная версия программного продукта.

В SIMULINK предусмотрена возможность набора моделей динамических систем в виде блок-схем, что хорошо согласуется с аналогичным структурным представлением систем автоматического управления. Поэтому SIMULINK очень удобно использовать для моделирования различного вида САУ.

Создание блок-схем в SIMULINK основано на технологии визуально-ориентированного программирования. В качестве компонент для построения модели динамической системы используются так называемые модули (блоки) из достаточно представительной библиотеки SIMULINK. Удобный графический интерфейс позволяет осуществлять сборку схемы моделирования в интерактивном (диалоговом) режиме путем перетаскивания с помощью мыши требуемых компонент и вычерчиванием соответствующих между ними линий связи. В результате получается структура исследуемой системы, которую обычно называют *Simulink*-моделью (*S* - моделью). Последняя сохраняется на диске в виде файла с расширением *.mdl*.

Для сложных систем *S*-модель, как правило, строится по иерархической структуре, компонентами которой являются *S*-модели более низкого уровня. Наиболее низкий уровень создается из модулей библиотеки SIMULINK.

В процессе ввода выбранных блоков, организации соединений и задания параметров компонентов система SIMULINK автоматически программирует *S*-модель в виде конечно-разностных уравнений и затем

при пуске процесса моделирования решает их с помощью средств MATLAB.

Следует отметить, что SIMULINK интегрирован также с рядом других пакетов расширения MATLAB, что обеспечивает достаточно широкие возможности его применения.

### 3.1. Запуск SIMULINK

Запуск приложения SIMULINK осуществляется в среде MATLAB двумя основными способами:

- нажатием кнопки **Simulink** на панели инструментов основного окна MATLAB;
- выполнением команды **Simulink** в командной строке MATLAB.

В результате открывается окно обозревателя (браузера) библиотеки **Simulink Library Browser** (рис. 3.1).

*!! Запустите SIMULINK одним из выше приведенных способов.*

Для закрытия приложения SIMULINK достаточно через меню окна браузера подать команду **File/Close** или щелкнуть мышкой на кнопке управления окном обозревателя  (**Close**).

*!! Закройте приложение SIMULINK.*

Имеются еще две дополнительные возможности активизации системы SIMULINK. Первая из них связана с запуском созданного ранее MDL-файла, в результате чего появляется одноименное окно S-модели. Далее для вызова обозревателя библиотеки SIMULINK на панели инструментов этого окна нажимается кнопка **Library Browser** или в меню окна S-модели выбирается опция **View/Library Browser**.

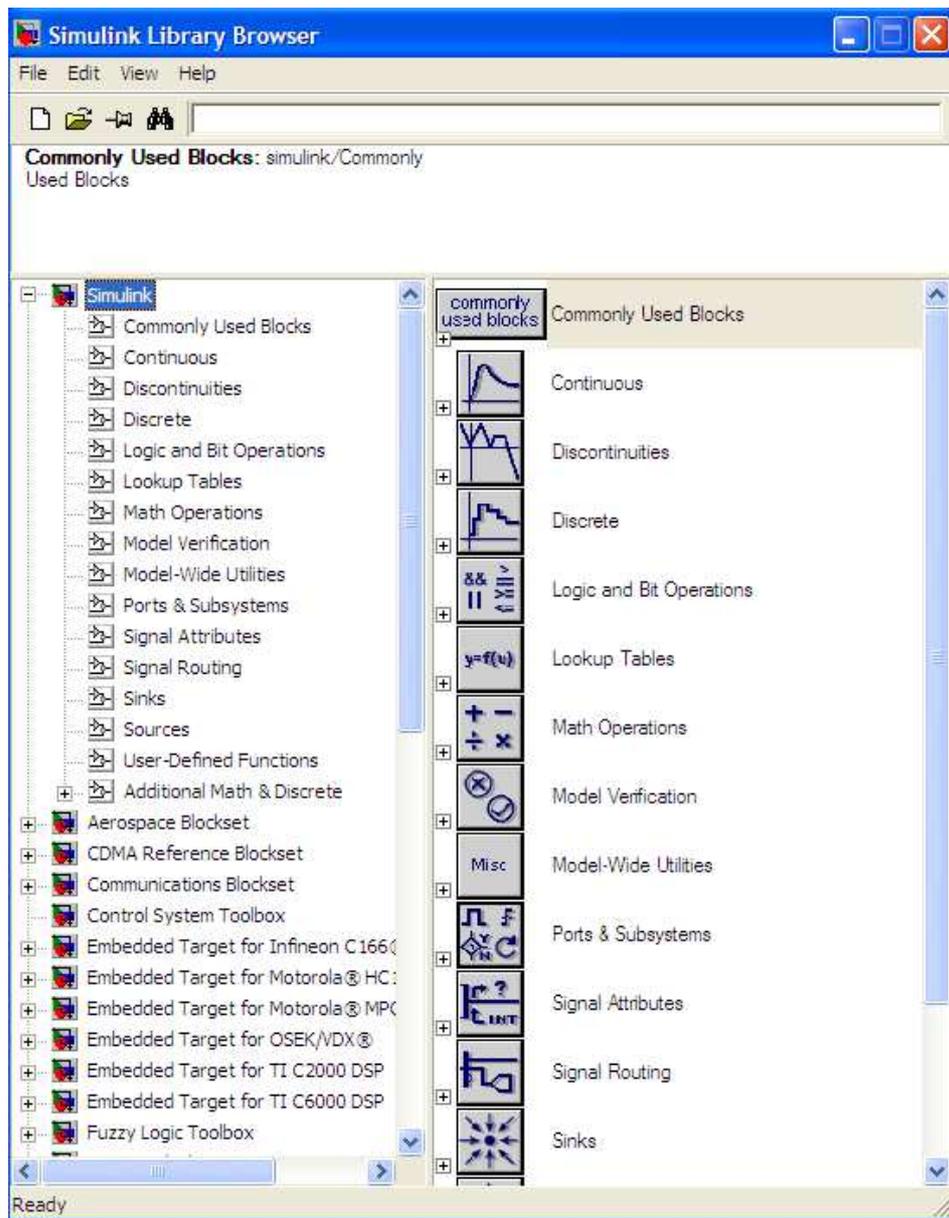


Рис. 3.1. Окно браузера библиотеки SIMULINK 6

*!! Откройте любой MDL- файл (он имеет характерную иконку как у Simulink), находящийся в каталоге MALAB71/toolbox/sloptim/sloptdemos. Затем активизируйте обозреватель библиотеки с использованием средств окна S-модели.*

Второй дополнительный путь вызова приложения SIMULINK связан с выбором через меню кнопки **Start** окна MATLAB команды **Simulink/Library Browser**.

*!! Используя все выше рассмотренные способы, поупражняйтесь еще раз в запуске и закрытии системы SIMULINK*

### **3.2. Интерфейс обозревателя библиотеки SIMULINK**

Браузер библиотеки SIMULINK служит для просмотра ее содержимого (поиска требуемого блока). Окно обозревателя имеет характерные для окон Windows элементы управления (рис. 3.1). Поэтому их назначение для пользователей, знакомых с данной операционной системой, не требует особых пояснений. Браузер во многом напоминает программу Проводник в Windows.

Его основой является иерархическое дерево разделов библиотеки, которое находится в левой части окна обозревателя. Содержимое выделенного раздела отображается в правой части окна браузера. Приемы поиска разделов библиотеки и их содержимого с помощью мыши такие же, как и в выше указанном Проводнике (раскрытие, свертывание, выделение и т.д.).

*!! Проверьте приемы навигации, используемые в Проводнике применительно к браузеру SIMULINK.*

Основными разделами библиотеки, используемыми при построении блок-схем САУ, являются:

1. *Continuous* – набор линейных непрерывных блоков: Integrator (интегратор), Derivative (дифференцирование), State-Space (пространство состояний), Transfer Fcn (передаточная функция), Zero-Pole (передаточная функция вида «нули-полюса»), Transport Delay (транспортное запаздывание) и др.

2. *Discontinuities* – набор нелинейных блоков: Saturation (ограничитель), Dead Zone (зона нечувствительности), Relay (реле), Backlash (люфт), Coulomb & Viscous Friction (сухое и вязкое трение) и др.

3. *Discrete* – набор дискретных модулей: Discrete-Time Integrator (дискретный интегратор), Discrete Filter (дискретный фильтр), Discrete Transfer Fcn (дискретная передаточная функция), Discrete

Zero-Pole (дискретная передаточная функция вида «нули-полюса»), Discrete State-Space (дискретная модель в пространстве состояний), Unit Delay (задержка на один такт), Zero-Order Hold (экстраполятор нулевого порядка), First-Order Hold (экстраполятор первого порядка) и др.

4. *Math Operations* – набор блоков математических операций: Sum (сумматор), Product (блок умножения), Gain (усилитель), Slider Gain (ползунковый усилитель), Math Function (математическая функция) и др.

5. *Sources* – набор источников сигналов: Constant (постоянный сигнал), Sine Wave (синусоидальный сигнал), Ramp (линейно изменяющийся сигнал), Step (ступенчатый сигнал), Random Number (случайный сигнал с нормальным распределением), Band-Limited White Noise (генератор «белого шума»), Clock (источник времени) и др.

6. *Sinks* – набор блоков приема и отображения сигналов: Scope (осциллограф), XY Graph (графопостроитель), Display (цифровой индикатор), Terminator (заглушка, конечной приемник) и др.

*!! Ознакомьтесь с содержанием всех вышеперечисленных разделов библиотеки SIMULINK.*

Содержимое основного каталога библиотеки SIMULINK и ее подразделов можно просмотреть в отдельном окне. Для этого необходимо выделить нужный раздел правой кнопкой мыши и на появившееся контекстное предложение «**Open the 'Название раздела' library**» дать подтверждение (мышью или клавишей **Enter**). Эти окна содержат более полную и структурированную информацию о содержимом раздела и занимают на экране значительно меньше места, чем окно браузера.

*!! Поупражняйтесь в вызове окон основных разделов библиотеки SIMULINK с помощью контекстного меню и ознакомьтесь со структурой представления в них модулей библиотеки.*

### 3.3. Создание S-модели

Для построения модели необходимо сначала создать новый файл модели с помощью команды **File/New/Model** или с использованием кнопки (**Create a new model**) на панели инструментов. В результате открывается безымянное окно **untitled** (рис. 3.2). В рабочем поле этого окна осуществляется сборка S-модели.

*!! Создайте новый файл модели любым из двух указанных способов.*

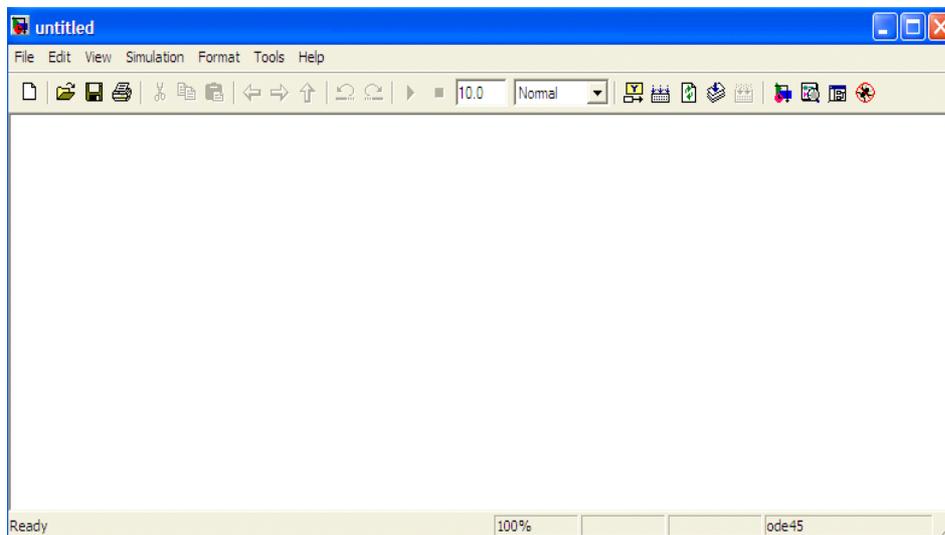


Рис. 3.2. Пустое окно файла S-модели

Методика и приемы построения S-модели поясняются ниже на основе составления модели для системы управления электроприводом постоянного тока (рис. 3.3).

Здесь обозначено:  $PC$  – регулятор скорости,  $PT$  – регулятор тока,  $ЭД$  – электродвигатель,  $УМ$  – усилитель мощности,  $U_{zc}$  – задающий сигнал;  $U_{ec}$  – ошибка по скорости;  $U_{zm}$  – задающее напряжение для контура тока;  $U_{em}$  – ошибка по току;  $E$  – ЭДС двигателя;  $U_d$  – напряжение якорной цепи;  $\Omega$  – скорость вращения;  $I_a$  – ток якоря;  $M$  – момент, развиваемый электродвигателем;  $M_n$  – момент нагрузки.

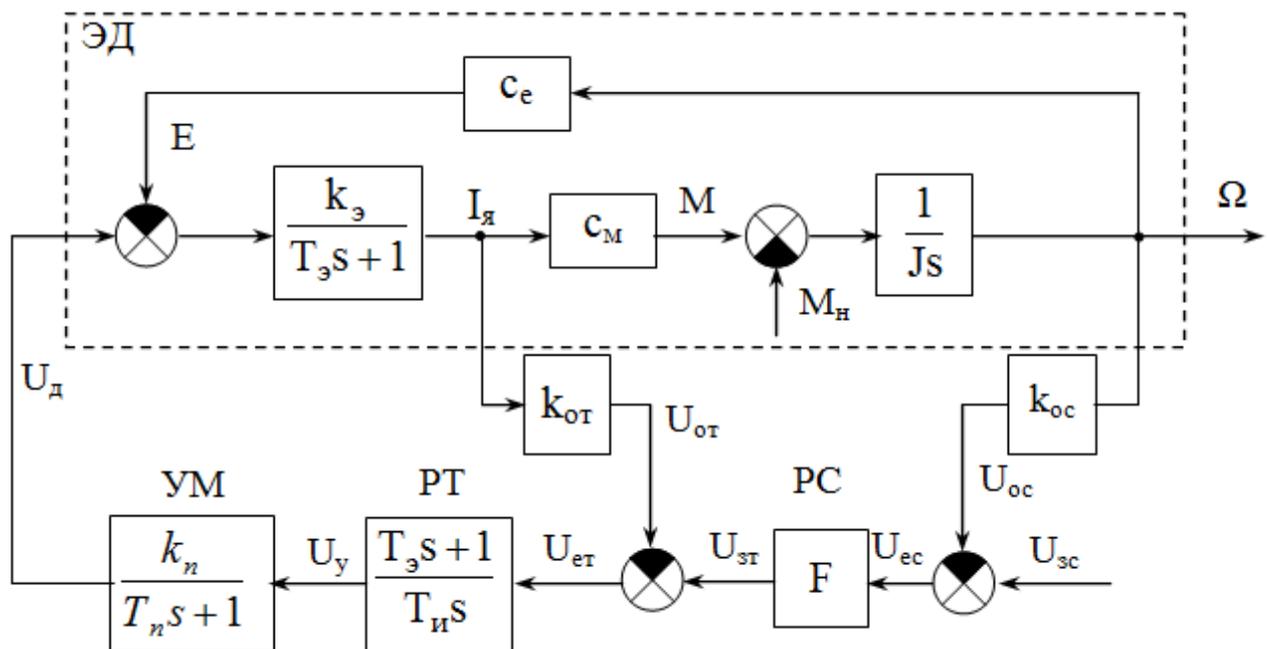


Рис. 3.3. Система стабилизации скорости электропривода

Нелинейное звено  $F$  (усилитель с ограничением выходного сигнала) имеет характеристику типа «насыщение»:

$$F = \begin{cases} k_{pc} U_{ec} & \text{при } |U_{zm}| < c, \\ c \cdot \text{sign} U_{ec} & \text{при } |U_{zm}| \geq c. \end{cases}$$

Параметры передаточных функций элементов этой САУ равны:

$$k_{pc} = 190; k_{п} = 23; k_{э} = 5; k_{от} = 0,45; k_{ос} = 0,025; c = 10; c_m = c_e = 0,42; J = 0,0885; T_n = 0,005; T_u = 0,52; T_э = 0,02.$$

Анализ структурной схемы системы управления электроприводом (рис. 3.3) показывает, что для построения ее модели необходимо использовать следующие модули библиотеки SIMULINK: линейные блоки с передаточной функцией в  $tf$ -форме, модули безынерционных звеньев (усилители, делители), сумматоры и нелинейный блок типа «насыщение». Для исследования динамических свойств систем автоматического управления обычно применяются переходные характеристики, поэтому в качестве

источников входного и возмущающего воздействий целесообразно использовать блоки *Step* (ступенчатый сигнал).

Таким образом, на первом этапе построения *S*-модели рассматриваемой САУ в окно сборки схемы требуется перенести следующие модули: блок *Transfer Fcn* из раздела библиотеки *Continuous*, блоки *Sum* и *Gain* – из раздела *Math Operations*, блок *Saturation* – из раздела *Discontinuities*, блок *Step* – из раздела *Sources*. При этом можно также воспользоваться разделом *Commonly Used Blocks*, в котором находятся наиболее общие блоки, в том числе *Sum*, *Gain*, *Saturation* и др.

Операции переноса блоков осуществляются:

- с помощью левой или правой кнопки мыши (основной метод);
- с использованием команды **Add to the current model** меню **Edit** браузера SIMULINK или контекстного меню выбранного блока;
- копированием из окна подраздела библиотеки через буфер обмена путем подачи команды **Copy** меню **Edit** или контекстного меню с последующей вставкой в окно сборки командой **Paste**.

*!! Перенесите указанные блоки по одному экземпляру в рабочее окно ранее созданного файла модели (рис. 3.2).*

Для размножения однотипных блоков можно использовать операцию копирования в окне *S*-модели. Для этого копируемый объект предварительно выделяется щелчком левой кнопки мыши. При успешном выполнении операции вокруг объекта появляется рамка с прямоугольными маркерами. Затем осуществляется копирование – либо правой кнопкой мыши, либо левой с нажатой клавишей *Ctrl*, либо через буфер обмена командами **Copy** и **Paste**. Всем копиям передаются параметры настройки оригинала и его имя с добавлением порядкового номера очередной копии.

Удаление ненужной копии или оригинала осуществляется либо командами **Edit/Delete** и **Edit/Cut** (кнопка **Cut** на панели инструментов) или соответствующими командами контекстного меню, либо нажатием клавиш **Delete** и **Backspace**.

*!! Сделайте нужное количество копий блоков, исходя из числа однотипных звеньев структурной схемы системы стабилизации.*

Далее блоки желательно расположить в окне S-модели в соответствии со структурой исходной САУ электроприводом. Это облегчит в дальнейшем процесс отладки модели и интерпретацию результатов моделирования. Перемещение блоков осуществляется левой кнопкой мыши.

*!! Расположите блоки в соответствии со структурной схемой системы стабилизации.*

Ряд блоков требуется переориентировать, т.е. повернуть их на 90° или 180°. Эти операции производятся путем использования соответственно команд **Format/Rotate block** – поворот блока на 90° по часовой стрелке, **Format/Flip block** – зеркальное отображение относительно оси симметрии (поворот на 180°). Эти опции можно также вызвать с помощью контекстного меню (щелчком на выбранном блоке правой кнопкой мыши).

*!! Используя команды **Rotate block** и **Flip block**, поверните модульные блоки в соответствии с направлением передачи информации в исследуемой системе. Если все операции сделаны правильно, то получится расположение, приведенное на рис. 3.4.*

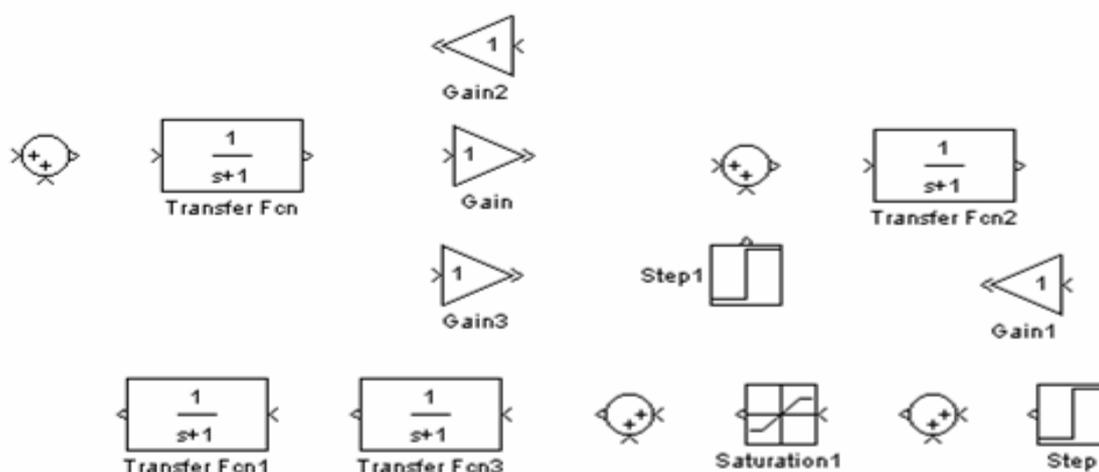


Рис. 3.4. Схема расположения блоков модели

На этом же этапе желательно установить требуемое количество и положение входов блоков *Sum*, а также знаки операций, выполняемых

на данных входах. Для этого после переориентации сумматора двойным щелчком левой кнопки мыши вызывается окно настройки данного блока **Function Block Parameters: Sum** (рис. 3.5).

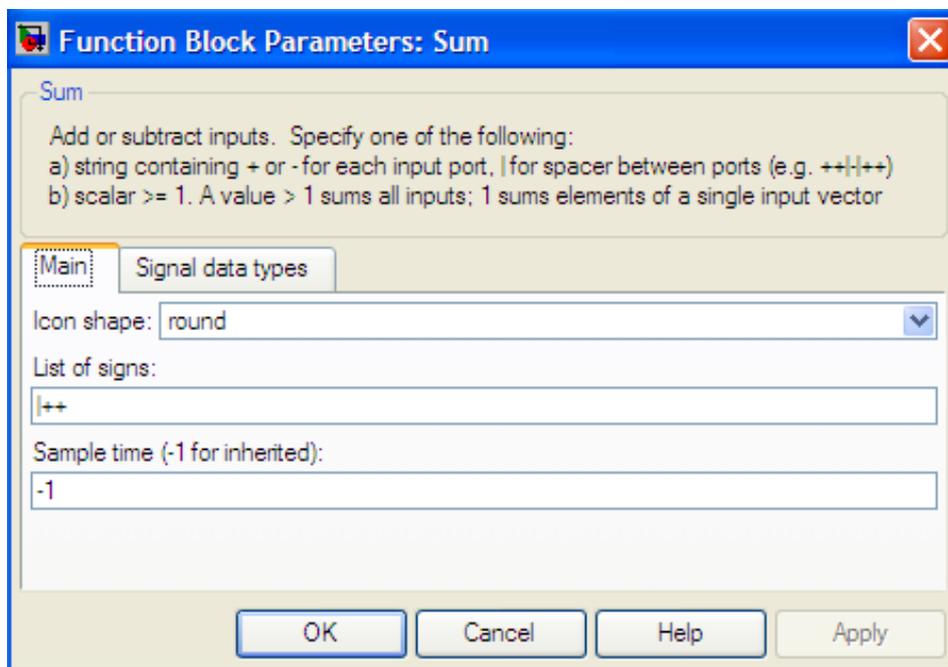


Рис. 3.5. Окно настройки блока Sum

В поле *Icon shape* выбирается вид блока: **round** – круг, **rectangular** – прямоугольник. Последний вариант используется обычно при четырех и более входах. В поле *List of signs* этого окна устанавливается последовательность входов и знаков алгебраического суммирования с помощью символов + (плюс), – (минус) и | (разделитель). После нажатия кнопки **OK** метки входов сумматора помечаются соответствующими знаками + и –.

Например, комбинации +|– и |–+ приведут соответственно к следующему расположению входов сумматоров:  ,  .

**!!** Настройте входы всех сумматоров, находящихся в окне сборки модели, в соответствии со структурной схемой системы стабилизации.

На следующем этапе построения S-модели создаются соединительные линии, связывающие входы и выходы блоков. С этой целью курсор мыши устанавливается на выходе (входе) блока; в

результате курсор превращается в крестик из тонких линий. Затем нажимается левая кнопка мыши и курсор перемещается ко входу (выходу) другого блока. Как только указатель приобретет вид крестика из сдвоенных линий клавишу мыши следует отпустить. В результате получится соединительная линия, заканчивающаяся жирной стрелкой. Если соединение получилось не совсем удачным или ошибочным, то его можно удалить как любой объект описанными выше способами или применить команду **Undo**.

Часто блоки требуется соединять ломанными линиями, состоящими из нескольких сегментов (угол излома составляет  $\pm 90^\circ$ ). В этом случае при создании соединительной линии кнопка мыши отпускается в точке предполагаемого излома, затем нажимается вновь и создается новый сегмент в нужном направлении и т.д. Ответвления от линий связи можно создавать либо правой кнопкой мыши, либо левой, но при нажатой клавише Ctrl.

*!! Установите все соединения между блоками в S-модели согласно связям соответствующих звеньев в структурной схеме электропривода.*

Структуру S-модели можно отредактировать, в частности, изменить положение блоков и сегментов соединительных линий, осуществить требуемую вставку и удаление блоков, сделать добавление нужных текстовых надписей и удаление ненужных обозначений, изменить размеры блоков и т.п. Такое редактирование делает S-модель более информационной.

Перемещение блока внутри модели осуществляется левой кнопкой мыши. При этом система SIMULINK перерисовывает линии, связывающие этот блок с другими блоками. Если требуется переместить сразу несколько блоков вместе с соединительными линиями и сохранением относительных расстояний между блоками, то их целесообразно сначала выделить с помощью рамки. Затем один из блоков зацепляют мышью и перемещают всю выделенную группу компонентов модели в нужное место окна сборки.

Изменение местоположения сегмента соединительной линии также производится левой кнопкой мыши; при этом идентификацией зацепления сегмента мышью служит курсор в виде креста со стрелками. При активизации точки излома линии связи указатель мыши трансформируется в кружок. Перемещение излома автоматически приводит обычно к появлению дополнительных точек излома, поэтому эта операция используется редко.

Для изменения размеров блока необходимо его предварительно выделить и затем навести курсор мыши на одну из угловых меток. Указатель при этом принимает вид двунаправленной стрелки, что говорит о его зацеплении с меткой. Затем блок можно растягивать или сжимать, увеличивая или уменьшая его размеры.

*!! Потренируйтесь в выполнении выше приведенных операций перемещения и изменения размеров объектов S-модели.*

В готовую модель иногда требуется вставить новый блок. В рассматриваемом примере такую процедуру нужно будет сделать для усилителя, ибо модуль *Saturation* в линейной части характеристики имеет единичный коэффициент передачи, а коэффициент  $k_{pc}$  не равен единице. Поэтому на входе ограничителя требуется установить дополнительно блок *Gain* с соответствующим коэффициентом усиления.

Для вставки нового блока его следует переместить мышью на соединительную линию. В результате операция вставки будет выполнена автоматически. Однако такая процедура возможно только для блоков, имеющих один вход и один выход.

Для исключения блока из структуры модели его следует отсоединить от линии связи. Для этого необходимо нажать клавишу **Shift** и перетащить блок мышью в другое место. На месте удаленного блока появится разрыв линии связи. Он легко ликвидируется созданием дополнительного соединения между точками разрыва.

*!! Вставьте блок Gain указанным выше способом на входе ограничителя в схеме моделирования электропривода постоянного тока.*

Для наглядности и удобства использования S-модели ее основные блоки и соединительные линии целесообразно снабдить соответственно именами и метками. Чтобы изменить имя блока следует щелкнуть левой клавишей мыши на его имени, которое дается первоначально системой SIMULINK по умолчанию. В результате должен появиться курсор ввода, после чего, используя обычные приемы редактирования, можно внести необходимые изменения.

Следует отметить, что предлагаемая версия программы SIMULINK 6 не адаптирована к использованию кириллицы, поэтому применение русскоязычного текста может привести к непредвиденным последствиям. В связи с этим без веских оснований не следует использовать символы кириллицы.

Изменить местоположение имени выделенного блока (на противоположную сторону) можно путем перетаскивания с помощью мыши или с использованием команды **Format/Flip Name**.

Иногда имя блока не требуется отображать на экране (а удалить его система SIMULINK не позволяет). В этом случае используется операция сокрытия имени. Для этого сначала блок выделяется, а затем используется команда **Format/Hide Name**. При необходимости имя блока можно восстановить с помощью команды **Format/Show Name**.

Соединительные линии в S-модели отождествляют координаты структурной схемы САУ. поэтому их целесообразно снабдить одноименными с сигналами метками. Чтобы создать метку следует дважды щелкнуть на сегменте линии и затем ввести нужное имя. При этом двойной щелчок необходимо выполнять точно на линии, в противном случае будет создан комментарий.

Метку сигналов S-модели можно перемещать на новое место путем ее перетаскивания мышью. Копирование метки осуществляется правой кнопкой мыши или левой с нажатой клавишей *Ctrl*. Чтобы отредактировать метку, необходимо щелкнуть на ней левой кнопкой мыши и затем внести соответствующие изменения в ее названии. Для удаления метки ее следует активизировать, затем стереть надпись с

помощью клавиш **Backspace** или *Delete*. В итоге после снятия выделения будут удалены все метки этой линии.

Часто в S-модели используются комментарии. Они дают возможность сопровождать блок-схему текстовой информацией о модели (название модели и т.п.) и отдельных ее составляющих. Для создания комментария необходимо щелкнуть дважды в выбранном свободном месте окна S-модели и в появившейся прямоугольной рамке ввести необходимый текст. Перемещение, копирование, редактирование, удаление комментария осуществляется вышеописанными способами для меток и текстовых надписей.

Для изменения шрифта в текстовых надписях следует выполнить команду **Format/Font...** и затем выбрать в открывшемся диалоговом окне нужный шрифт. Для имен блоков, текстовых обозначений и надписей рекомендуется использовать шрифт одинакового начертания и размера.

*!! Приведите в соответствие S-модель со структурной схемой системы стабилизации (рис. 3.3), введя соответствующие надписи блоков, метки сигналов, название модели и удалив лишние обозначения. При правильном выполнении заданий должна получиться структура S-модели, приведенная на рис. 3.6. Если имеются отличия, то внесите коррективы.*

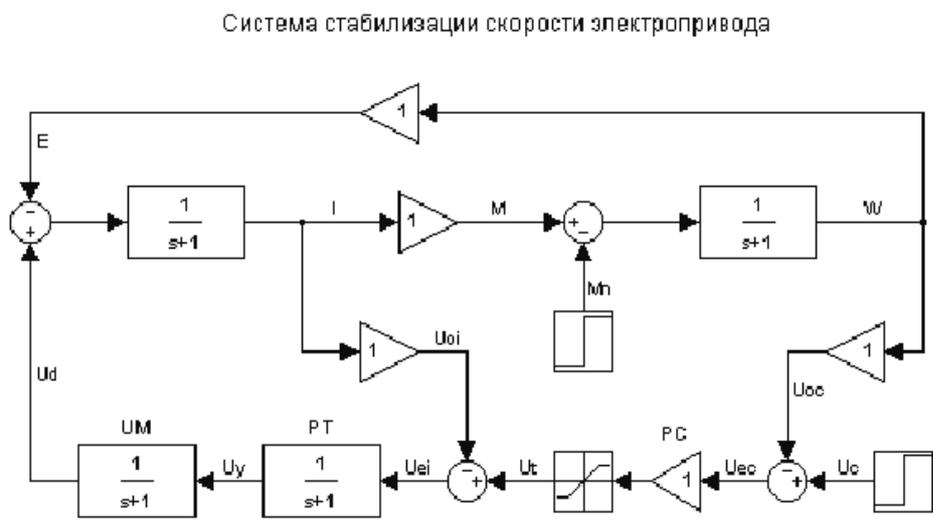


Рис. 3.6. Структура S-модели электропривода постоянного тока.

Модель можно сделать более выразительной:

- введением теней для ее компонентов;
- установкой цвета переднего плана и цвета фона блоков и комментариев;
- изменением цвета рабочего поля модели.

По умолчанию в системе SIMULINK установлен черно-белый вариант палитры цветов для формирования изображения модели без теней.

Для добавления тени к изображению любого компонента модели его требуется выделить, и затем подать команду **Format/Show drop shadow**. Для исключения тени применяется альтернативная команда **Format/Hide drop shadow**.

Цвет переднего плана представляет собой цвет графического изображения блока с исходящими из него линиями связи, его надписей и тени. Для изменения этого цвета требуется подать команду **Format/Foreground color** и в предложенном меню выбрать необходимый цвет.

Цвет фона задает цвет заливки графического изображения блока или поля комментария. Для задания цвета фона нужно подать команду **Format/Background color** и в предлагаемом меню выбрать желаемый цвет. Аналогично осуществляется изменение цвета поля сборки S-модели с применением команды **Format/Screen color**.

*!! Введите тени у всех компонентов модели, реализующих электродвигатель ЭД. Установите по своему усмотрению цвет фона блоков модели, цвет поля у названия модели и цвет фона модели в целом.*

### 3.4. Установка параметров блоков S-модели

Функции, которые выполняет блок, определяются его параметрами. Установка числовых значений последних осуществляется в диалоговом окне **Function Block Parameters**, которое вызывается двойным щелчком мыши на изображении блока,

а также выбором в меню **Edit** или контекстном меню опции '**Название блока**' **parameters....** Каждый типовой блок имеет свою собственную структуру диалогового окна с соответствующими полями параметров настроек. Поэтому в практических задачах приходится прибегать к справочнику по компонентам библиотеки SIMULINK.

Ранее была рассмотрена методика установки параметров блока Sum. Ниже приводятся способы настроек остальных типовых блоков, используемых при построении S-модели электропривода.

1. Блок **Transfer Fcn** (Передаточная функция).

В диалоговом окне этого блока **Block Parameters: Transfer Fcn** задаются массивы коэффициентов полинома числителя (поле *Numerator coefficient*) и полинома знаменателя (поле *Denominator coefficient*).

*!! Установите численные значения параметров передаточных функций всех четырех блоков типа Transfer Fcn. Проверьте правильность ввода параметров по отображению передаточной функции внутри блоков.*

2. Блок **Gain** (Усилитель).

Установка коэффициента усиления производится в диалоговом окне **Function Block Parameters: Gain** в поле Gain.

*!! Задайте численные значения коэффициентов усиления всех блоков Gain S-модели, исходя из коэффициентов передачи соответствующих безынерционных звеньев системы стабилизации. Проверьте правильность ввода по отображению численных значений коэффициентов внутри блоков Gain. Если значение коэффициента не помещается в пиктограмме блока, то увеличьте ее размеры.*

3. Блок **Saturation** (Ограничитель).

Основными параметрами блока являются значения уровней ограничения, которые вводятся в диалоговом окне **Function Block Parameters: Saturation** в поля: *Upper limit* – верхний уровень, *Lower limit* – нижний уровень.

*!! Установите уровни ограничения блока Saturation в S-модели, исходя из заданных параметров нелинейного звена F системы управления электроприводом.*

4. Блок **Step** (Генератор ступенчатого сигнала).

Этот блок формирует ступенчатый сигнал в определенный момент времени. В диалоговом окне **Source Block Parameters: Step** требуется задать:

- в поле **Step time** – время наступления перепада (скачка сигнала);
- в поле **Initial value** – начальное значение сигнала (до скачка);
- в поле **Final value** – конечное значение сигнала (после скачка).

*!! Установите входное воздействие  $U_{zc} = 5 \cdot 1(t)$ , а возмущение*

$$M_n = \begin{cases} 0,5 & \text{при } t < 3, \\ 3,7 & \text{при } t \geq 3. \end{cases}$$

Следует отметить, что установку параметров блоков S-модели можно осуществлять на более раннем этапе – до создания соединительных линий, как это было сделано для сумматора. При таком подходе размеры блоков корректируются до проведения соединительных линий и в дальнейшем их изменять, как правило, не требуется.

### **3.5. Задание параметров моделирования и его выполнение**

Перед выполнением моделирования необходимо задать общие параметры этого процесса. Они устанавливаются в диалоговом окне **Configuration Parameters: Название модели**, которое открывается командой **Simulation/Configuration parameters....** Задание параметров моделирования производится в двух основных полях диалогового окна:

- **Start time** – начальное время (обычно 0);
- **Stop time** – конечное время.

Остальные настройки целесообразно не изменять, т.е. использовать опции, установленные по умолчанию, так как они определяют наиболее оптимальные по точности и быстродействию численные методы интегрирования системы дифференциальных уравнений, описывающих *S*-модель.

**!! Установите интервал моделирования 4 сек.**

При моделировании, как правило, исследуется влияние параметров системы на характер изменения временных характеристик. В связи с этим к требуемым координатам модели с целью регистрации переходных процессов обычно подключаются индикаторы *Scope* и *Display* (из раздела *Sinks* библиотеки SIMULINK).

При этом к одному блоку *Scope* можно подключить несколько сигналов. Для этого следует дважды щелкнуть на его изображении и в открывшемся окне *Scope* нажать кнопку **Parameters**. В появившемся окне настроек параметров '**Scope**' parameters в поле **Number of axis** ввести число входов осциллографа. В итоге на изображении блока *Scope* появятся дополнительные входные порты. Кроме того, в поле **Tick labels** этого окна осуществляется ввод/скрытие осей и меток осей выбором соответствующих опций:

- **all** – подписи для всех осей;
- **none** – отсутствие всех осей и подписей к ним;
- **bottom axis only** – подписи горизонтальной оси только для нижнего графика.

**!! Подключите блоки *Scope*(виртуальный осциллограф) к координатам  $\Omega$ ,  $M$ ,  $U_{zm}$ ,  $U_d$ , а блоки *Display* – к координатам  $U_{ec}$  и  $U_{em}$  (для оценки точности). Настройте блоки *Scope* так, чтобы на графиках были отображены регистрируемые координаты**

Запуск процесса моделирования осуществляется двумя способами:

- кнопкой **Start simulation** на панели инструментов;
- командой **Simulation/Start**.

Процесс моделирования можно остановить командой (или кнопкой) **Simulation/Pause** и затем продолжить командой (или кнопкой) **Simulation/Continue**; можно также завершить процесс досрочно, выбрав опцию или кнопку **Simulation/Stop**.

Чтобы просмотреть результаты моделирования необходимо дважды щелкнуть на блоке **Score**. Если кривая переходного процесса выходит за пределы координатной сетки, следует щелкнуть на пиктограмме **Autoscale** панели инструментов блока **Score** или выбрать эту команду в контекстном меню.

*!! Запустите процесс моделирования и проанализируйте переходный процесс в окнах **Score** по указанным координатам.*

*!! Исключите из схемы блок **Saturation**; затем осуществите моделирование линейной **S**-модели. Сравните полученные результаты с переходными характеристиками нелинейной **S**-модели. Оцените влияние ограничения координаты  $U_{zm}$  на динамику системы.*

Для встраивания полученной **S**-модели в некоторый документ используется команда **Edit/Copy Model To Clipboard** (для копирования изображения модели в буфер обмена), а затем осуществляется ее вставка в искомый документ с помощью команды **Paste**.

Графические зависимости из окна **Score** переносятся в буфер обмена нажатием клавиши **Print Screen**. Однако, ввиду того, что фон графика черный, перед вставкой в документ их целесообразно отредактировать в программе *Paint* или другом графическом редакторе – обратить цвет и вырезать график из окна **Score**.

*!! Создайте документ в текстовом редакторе **Word** и вставьте в него построенную **S**-модель электропривода и полученные графики.*

## Дополнительное упражнение

1. Построить  $S$ -модель электропривода постоянного тока с двухзонным регулированием скорости вращения двигателя, структурная схема которого приведена на рис. 3.7.

В этой схеме по сравнению с рис. 3.3 введены следующие обозначения:  $PЭ$  – регулятор ЭДС,  $ТП$  – тиристорный преобразователь,  $ОВ$  – обмотка возбуждения,  $РТВ$  – регулятор тока возбуждения,  $БО$  – блок ограничения,  $ТГ$  – тахогенератор,  $ДН$  – делитель напряжения,  $БВМ$  – блок выделения модуля,  $БП$  – блок перемножения,  $ДЭ$  – датчик ЭДС двигателя,  $I_g$  – ток обмотки возбуждения,  $U_g$  – напряжение обмотки возбуждения,  $U_{zme}$  – задающий сигнал контура возбуждения,  $U_{зэ}$  – напряжение задания ЭДС,  $U_{eэ}$  – ошибка контура ЭДС,  $\Delta U$  – возмущение сети, приведенное к выходу УМ.

Для построения  $S$ -модели этой схемы в отличие от выше рассмотренной структуры САУ потребуются два новых блока: *Product* – блок умножения и *ABS* – блок выделения модуля. Оба они находятся в каталоге Math Operations. Для реализации блоков БП на основе блока Product достаточно ввести в поле *Number of inputs* диалогового окна *Function Block Parameters*: Product число входов блока умножения, а для реализации блока БМВ – никаких дополнительных установок в диалоговом окне блока *ABS* проводить не требуется.

2. Отредактировать модель и дополнить ее виртуальными индикаторами для отражения процесса изменения основных координат системы регулирования:  $\Omega$ ,  $E$ ,  $U_d$ ,  $U_{ec}$ ,  $U_{eэ}$ ,  $U_{зт}$ ,  $U_{зтв}$ ,  $I_a$ ,  $I_g$ ,  $M$ ,  $U_{em}$ ,  $U_{emв}$ .

3. Установить численные значения параметров всех блоков модели:

$$\begin{array}{lll}
k_{pc} = 7,88; & k_B = 0,5; & T_\phi = 0,02; \\
k_\Pi = 97; & k_f = 0,637; & T_\omega = 0,0095; \\
k_\omega = 1,95; & k_{дТВ} = 2,5; & T_{и} = 0,16; \\
k_{дТ} = 0,085; & k_{дЭ} = 0,05; & T_\Pi = 0,005; \\
k_{дН} = 0,2; & T_{иЭ} = 0,08; & J = 0,3. \\
k_{ТГ} = 0,318; & T_B = 0,1; & \\
k_{ТП} = 81; & T_{иВ} = 0,085; & 
\end{array}$$

Уровни ограничения в звене БО принять равным  $\pm 10$  В.

5. Задать параметры моделирования согласно следующей программе изменения входных координат системы:

$$\begin{aligned}
U_{зэ} &= 10 \cdot 1(t), \quad U_{зс} = 10 \cdot 1(t-2), \quad \Delta U = 20 \cdot 1(t-7), \\
M_n &= \begin{cases} 5 & \text{при } t < 5, \\ 19 & \text{при } t \geq 5. \end{cases}
\end{aligned}$$

Интервал моделирования взять равным 10 с.

5. Запустить процесс моделирования и посмотреть полученные результаты.

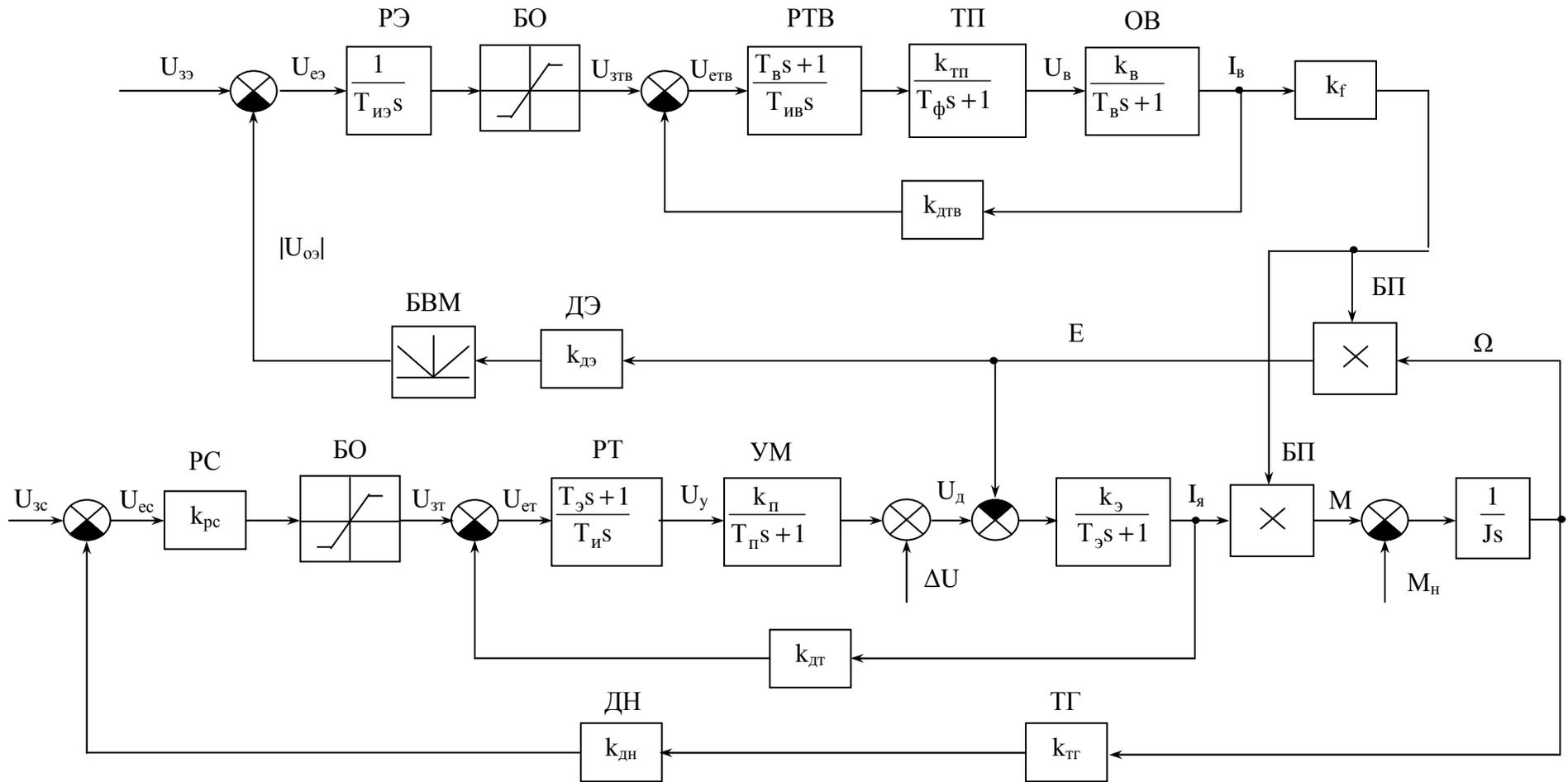


Рис. 3.7. Схема электропривода постоянного тока с двухзонным регулированием

## *Контрольные вопросы*

1. Дайте общую характеристику системе SIMULINK.
2. Как построена библиотека SIMULINK и как осуществляется доступ к ее разделам?
3. Каким образом переносятся блоки из библиотеки SIMULINK в поле сборки S-модели?
4. Поясните процедуру переориентации блоков S-модели.
5. Как создаются соединительные линии и отводы?
6. Как переместить соединительную линию, блок, узел?
7. Каким образом осуществляются операции вставки и исключения блоков при создании S-модели?
8. Как сделать метку соединительной линии и как ее скопировать или переместить в другое место?
9. Какими командами создается и убирается тень у компонента S-модели?
10. Дайте краткую характеристику возможностям пакета SIMULINK по изменению цвета изображения компонента модели и его фона.
11. Расскажите, как осуществляется установка параметров моделирования?
12. Какие блоки отображения и индикации сигналов вы знаете?
13. Перечислите основные нелинейные блоки, использованные в лабораторной работе при построении S-моделей.
14. Как вставить S-модель в документ?
15. Поясните особенности процесса переноса графиков из окна Scope в текстовый документ.

## 4. ПАРАМЕТРИЧЕСКАЯ НАСТРОЙКА ДИНАМИЧЕСКИХ СИСТЕМ С ПОМОЩЬЮ ПАКЕТА SIMULINK RESPONSE OPTIMIZATION

Пакет *Simulink Response Optimization (SRO)* предназначен для параметрической оптимизации линейных и нелинейных систем. Он интегрирован с приложением SIMULINK и по существу является его дополнением. Для решения задач оптимизации САУ в пакете *SRO* используются алгоритмы нелинейного программирования.

В состав MATLAB 7.1 входит версия 2.3 пакета *SRO*. От предыдущих версий она отличается усовершенствованием алгоритмов оптимизации. Однако принципы работы с программой *SRO* остались без изменений, потому в дальнейшем на номер версии ссылок не будет.

В *SRO* для оценки качества САУ используются всего три блока:

- CRMS (Continuous root mean square);
- DRMS (Discrete root mean square);
- Signal Constraint.

*!! Запустите SIMULINK, найдите в библиотеке пакет Simulink Response Optimization и просмотрите его содержимое.*

Блок *CRMS* реализует математическую зависимость:

$$x(t) = \sqrt{\frac{1}{t} \int_0^t e^2(\tau) d\tau}, \text{ при } t > 0,$$

где  $e(t)$ ,  $x(t)$  - соответственно вход и выход блока.

Блок *DRMS* реализует дискретный вариант указанной выше зависимости:

$$x(n) = \sqrt{\frac{1}{n} \sum_{k=0}^n e^2(k)}, \text{ при } n > 0.$$

Обычно оба критерия настройки используют для оценки САУ по ее среднеквадратичной ошибке.

Блок *Signal Constraint* является основным блоком. Он имеет свое рабочее окно и позволяет в интерактивном режиме выполнять следующие операции:

- устанавливать необходимые ограничения во временной области на переходный процесс;
- устанавливать настраиваемые параметры;
- указывать неопределенные параметры;
- осуществлять параметрическую оптимизацию системы с заданными ограничениями.
- 

#### 4.1. Подготовка САУ к параметрической оптимизации

Методику подготовки САУ к параметрической оптимизации с помощью пакета *SRO* рассмотрим на основе примера следящей системы, структура которой приведена на рис. 4.1.

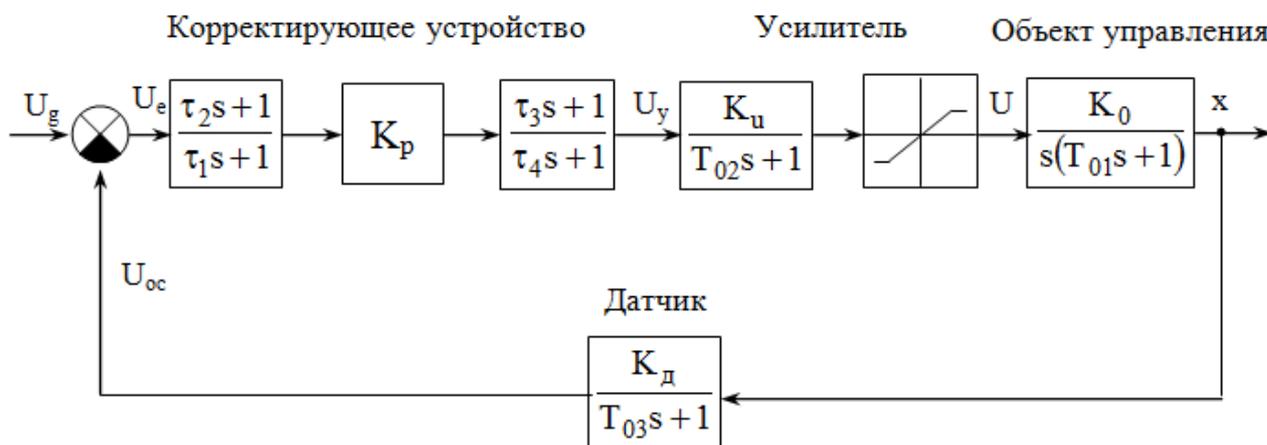


Рис. 4.1. Структурная схема следящей системы

Здесь  $K_p = 10$ ,  $K_u = 40$ ,  $K_0 = 2$ ,  $K_d = 0,25$ ,  $\tau_1 = 0,8$  с,  $\tau_2 = 0,5$  с,  $\tau_3 = 0,12$  с,  $\tau_4 = 0,00625$  с,  $T_{01} = 0.1$  с,  $T_{02} = 0,02$  с,  $T_{03} = 0,01$  с; предельные значения сигнала  $U$  равны  $\pm 50$ В. При этом значение  $K_0$  нестабильно и может изменяться в пределах  $\pm 10\%$  относительно его номинального значения. Необходимо подобрать параметры корректирующего

устройства таким образом, чтобы перерегулирование не превышало 20%, а время переходного процесса 0,8 с.

На первом этапе требуется собрать в среде SIMULINK схему моделирования динамической системы. При этом в качестве параметров исследуемых или настраиваемых звеньев целесообразно ввести их идентификаторы.

*!! Постройте S-модель САУ, приведенную на рис. 4.1. Введите численные значения параметров  $K_w$ ,  $K_d$ ,  $T_{01}$ ,  $T_{02}$ ,  $T_{03}$ , ограничения по выходному напряжению усилителя  $U$  и значение входного воздействия  $U_g = 2,5 \cdot 1(t)$ . Остальные параметры задайте в символьной форме, желательно в виде одноименных идентификаторов.*

Далее через командную строку MATLAB требуется ввести численные значения параметров звеньев системы, представленных в S-модели в виде коэффициентов передач и постоянных времени.

*!! Введите через командную строку MATLAB параметры S-модели, заданной в виде идентификаторов. Проверьте работоспособность созданной S-модели по виртуальному осциллографу, установив время моделирования 2 с.*

Такая S-модель для анализа более удобна, так как в ней можно легко изменять численные значения коэффициентов и постоянных времени звеньев САУ. Более того, эти изменения ( $K_p$ ,  $\tau_1$ ,  $\tau_2$ , ...) делаются автоматически с помощью блока *Signal Constraint*, который в результате целенаправленного поиска определяет оптимальные параметры настраиваемых блоков.

Блок *Signal Constraint* подключается обычно к выходу S-модели, то есть контролируемым сигналом, как правило, является переходная характеристика системы.

*!! Подключите блок Signal Constraint к выходу S-модели следящей системы. При правильном выполнении всех предыдущих заданий получится модель, приведенная на рис. 4.2.*

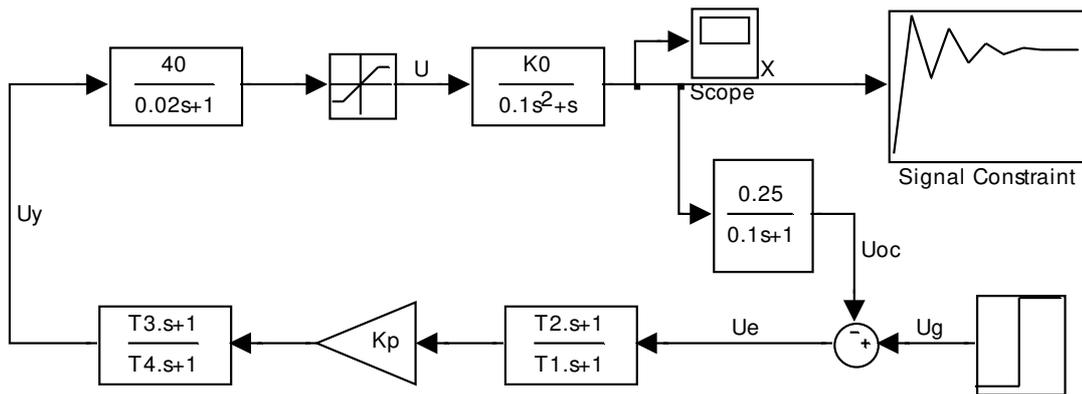


Рис. 4.2. S-модель следящей системы

Очередной этап связан с настройкой параметров блока *Signal Constraint*. Эту процедуру рекомендуется выполнять в следующем порядке.

Двойным щелчком мыши на пиктограмме *Signal Constraint* открыть основное окно *Signal Constraint – Block Parameters: Signal Constraint* (рис. 4.3).

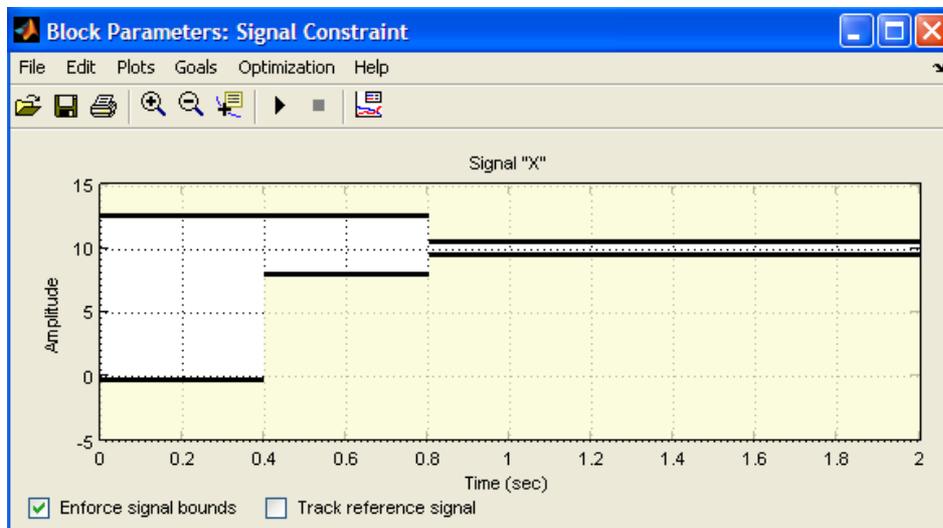


Рис. 4.3. Основное окно *Signal Constraint*

В графической части этого окна по умолчанию предлагаются границы области, в которую должна вписываться переходная характеристика системы, получаемая в результате параметрической оптимизации. Для изменения границ этой области выбирается опция

**Goals/Desired Response....** Это приводит к появлению диалогового окна **Desired Response** (рис.4.4).

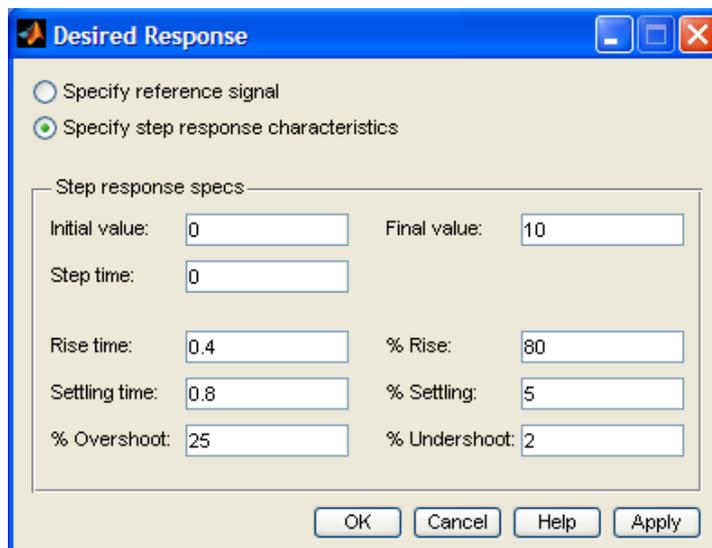


Рис. 4.4. Диалоговое окно *Desired Response*

Ввиду того, что оптимизация системы осуществляется по параметрам переходной характеристики, следует установить в этом окне переключатель в положение **Specify step response characteristics** и ввести в предлагаемые поля требуемые значения характеристик переходного процесса:

- в поле ***Initial value*** – исходное (начальное) значение переходной характеристики;
- в поле ***Final value*** – установившееся (конечное) значение переходной характеристики;
- в поле ***Step time*** – время начала наблюдения за переходным процессом;
- в поле ***Rise time*** – время нарастания сигнала;
- в поле ***%Rise*** – уровень для определения времени нарастания сигнала;
- в поле ***Setting time*** – время переходного процесса;
- в поле ***% Setting*** – зона установившегося значения в %;
- в поле ***% Overshoot*** – перерегулирование в %;

- в поле *% Undershoot* – недорегулирование (обычно 1%).

Для подтверждения ввода установленных параметров необходимо нажать последовательно кнопки **Apply** и **ОК**.

*!! Установите в первом приближении желаемые ограничения на зону переходной характеристики, исходя из результатов наблюдения по Score.*

Следующий шаг связан с заданием настраиваемых параметров. Для этого подается команда **Optimization/Tuned Parameters...**, которая приводит к появлению диалогового окна **Tuned Parameters** (рис. 4.5).

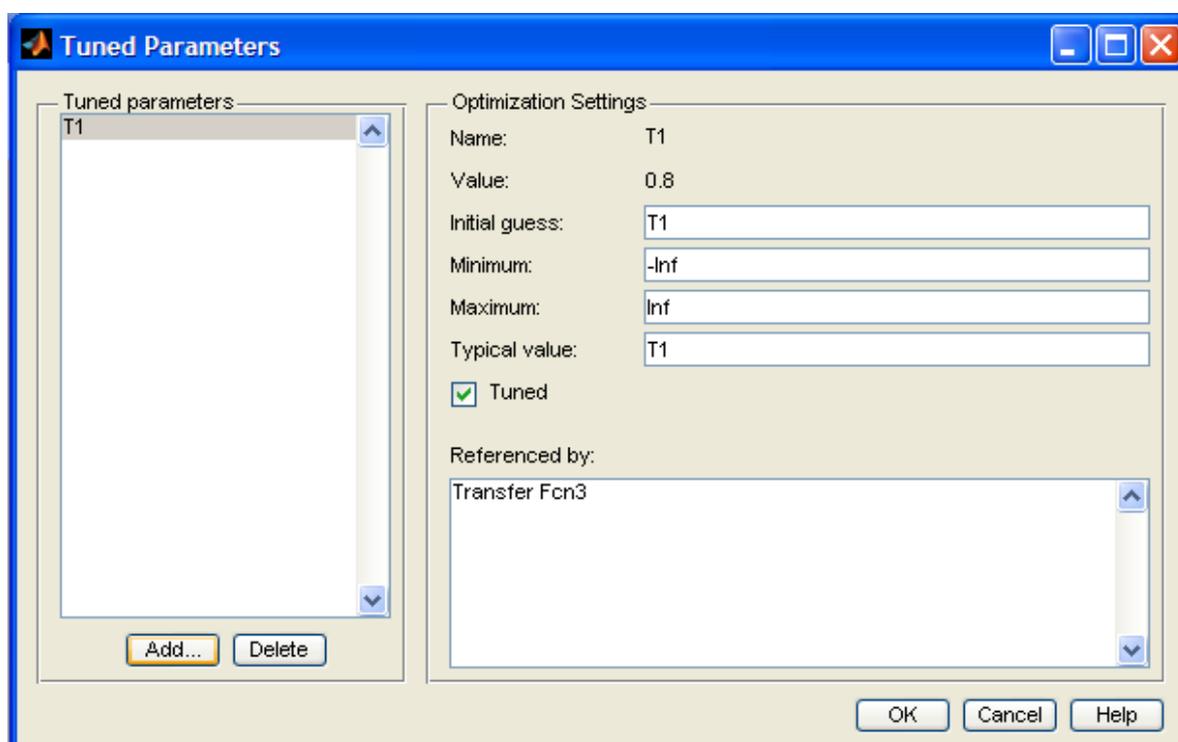


Рис. 4.5. Диалоговое окно Tuned Parameters

В его одноименное поле **Tuned parameters** формируется список настраиваемых параметров. Для этого нажимается кнопка **Add...**, в открывшемся окне **Add Parameters** (рис. 4.6) выделяются необходимые параметры и нажимается кнопка **ОК**.

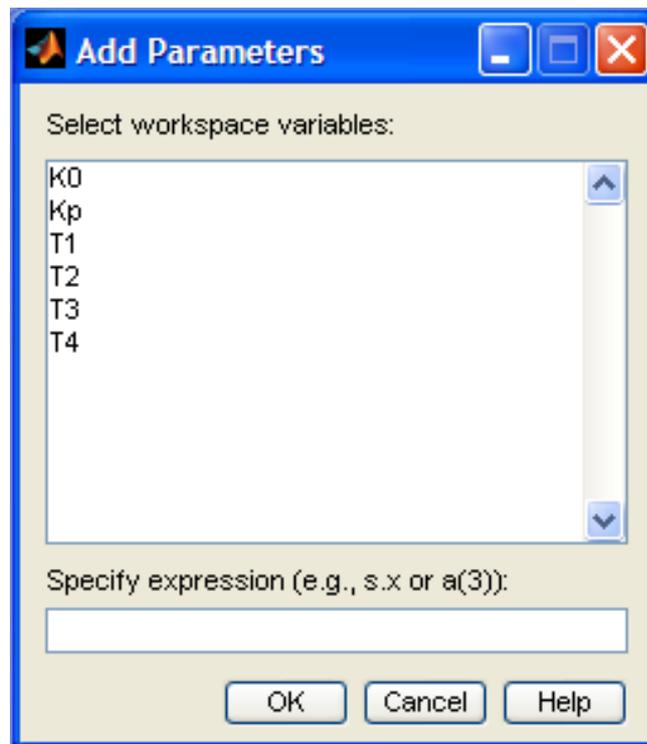


Рис. 4.6. Окно выбора настраиваемых параметров *Add Parameters*

Исключение какого-либо параметра из списка осуществляется аналогично – коэффициент выделяется и нажимается кнопка **Delete**.

В правом поле окна **Tuned Parameters** автоматически выводятся диапазон изменения и исходное значение выбранного параметра. При необходимости границы диапазона (**Minimum** и **Maximum**) можно откорректировать.

*!! Установите в качестве настраиваемого параметра коэффициент передачи корректирующего устройства  $K_p$ .*

## 4.2. Процесс параметрической настройки системы

Запуск процесса оптимизации осуществляется в основном окне **Signal Constraint** нажатием кнопки **Start** или подачей команды **Optimization/Start**. Процесс настройки системы можно наблюдать по переходным характеристикам, которые вычерчиваются в этом окне разными цветами, а также в открывающемся окне **Optimization Progress**.

*!! Запустите процесс параметрической оптимизации следящей системы. Подождите пока процесс не закончится.*

Результаты настройки в виде численных значений параметров системы можно получить также через командную строку MATLAB.

*!! Повторите параметрическую оптимизацию следящей системы на заданный переходный процесс, выбрав в качестве настраиваемых параметров  $\tau_1, \tau_2, \tau_3, \tau_4$ .*

Задача оптимизации значительно усложняется при наличии в объекте управления неопределенных параметров. Это часто встречается в практике автоматического управления, в частности, когда параметры не могут быть получены точно в результате идентификации объекта управления или когда параметры изменяются с течением времени в определенном диапазоне.

В этом случае при оптимизации системы с использованием программы RSO неопределенные параметры задаются в диалоговом окне **Uncertain Parameters** (рис. 4.7), которое вызывается командой **Optimization/ Uncertain Parameters ...**. В поле **Uncertain Parameters** этого окна вводятся в табличном виде неопределенные параметры путем нажатия кнопки **Add ...** и выбором соответствующего коэффициента из предлагаемого в окне **Add Parameters** списка. В таблице автоматически устанавливаются по умолчанию максимальные и минимальные значения неопределенных параметров. При необходимости их можно откорректировать.

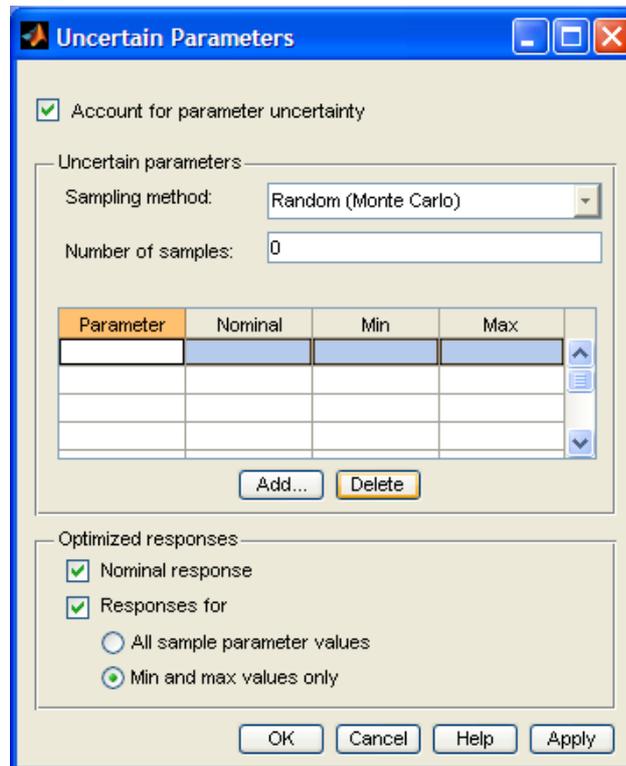


Рис. 4.7. Диалоговое окно Uncertain Parameters

*!! Введите в качестве неопределенного параметра коэффициент передачи объекта управления  $K_0$  и установите диапазон его изменения в пределах  $\pm 10\%$  от номинального значения. В качестве настраиваемого параметра возьмите коэффициент передачи регулятора  $K_p$ . Затем запустите процесс параметрической настройки.*

*!! Исключите блок ограничения из структуры S-модели и повторите процесс параметрической настройки следящей системы при неопределенном значении  $K_0$ . Сравните результаты настроек в линейной и нелинейной системах.*

### *Дополнительное упражнение*

1. Осуществить настройку параметров ПИД-регулятора в системе управления, структурная схема которой приведена на рис. 4.8.

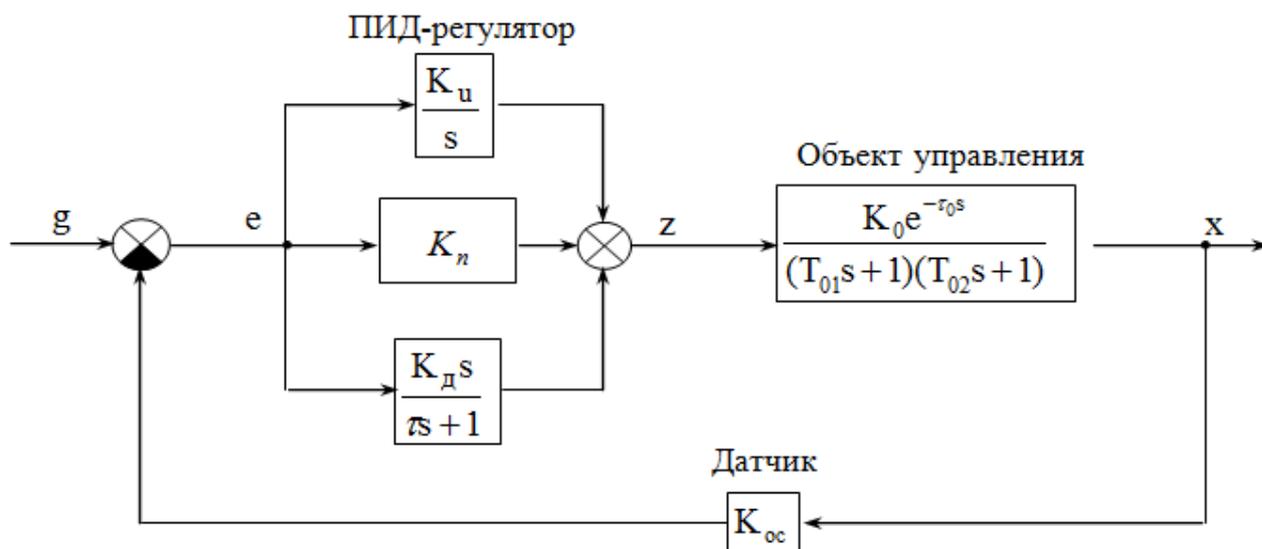


Рис. 4.8. Структурная схема системы

Передаточная функция объекта управления имеет следующие параметры:  $K_0=1.2$ ,  $T_{01}=6$  с,  $T_{02}=2.8$  с,  $\tau_0=4.5$  с. При этом запаздывание  $\tau_0$  и постоянная времени  $T_{01}$  нестабильны и изменяются соответственно в диапазонах  $\pm 5\%$  и  $\pm 10\%$  от номинального значения. Расчетные параметры регулятора и цепи обратной связи равны:  $K_u=0.4$ ,  $K_n=3.4$ ,  $K_d=7.7$ ,  $K_{oc}=0.5$ . На входе канала дифференцирования используется фильтр с постоянной времени  $\tau=K_d/10$ .

Критериями настройки являются основные показатели переходной характеристики: время переходного процесса  $t_n \leq 40$  с, перерегулирование  $\sigma \leq 25\%$ .

2. Построить  $S$ -модель данной системы управления и задать параметры ее блоков. Проверить работоспособность модели, подключив к ней виртуальный осциллограф *Scope*.

3. Подсоединить к выходной координате объекта управления блока *Signal Constraint*, в основном окне которого сформировать область желаемого расположения переходной характеристики и задать настраиваемые параметры.

4. Осуществить параметрическую оптимизацию систему. Сравнить полученные настройки регулятора с их расчетными значениями.

5. Объявить в качестве неопределенных параметров объекта время запаздывания  $\tau_0$  и постоянную времени  $T_{01}$ . Задать указанные выше диапазоны их изменений. Затем вновь запустить процесс оптимизации. Проанализировать полученные результаты настроек.

6. Повторить параметрическую настройку для более жестких ограничений на длительность переходного процесса:  $t_n \leq 20$  с.

7. Сохранить  $S$ -модель в виде  $MDL$ -файла под своей фамилией в каталоге *Work*.

### *Контрольные вопросы*

1. Для решения каких задач предназначен пакет *SRO*?
2. Какие блоки используются в *SRO* для оценки качества динамических систем?
3. Какую математическую зависимость реализуют блоки типа *RMS*?
4. В каких случаях для оценки качества системы целесообразно использовать блоки *CRMS* и *DRMS*?
5. Перечислите основные этапы подготовки САУ к параметрической оптимизации.
6. Поясните процедуру точной установки запретных границ для переходной характеристики при использовании диалогового окна *Desired Response*.
7. Поясните методику настройки границ области желаемого расположения переходной характеристики в окне *Signal Constraint*.
8. Каким образом задаются настраиваемые параметры системы?
9. Как объявляются неопределенные параметры системы управления и диапазон их изменения?
10. Что является индикатором завершения процесса параметрической оптимизации?
11. Каким образом считываются значения настраиваемых параметров, полученных в результате оптимизации?

## **5. ИСПОЛЬЗОВАНИЕ MATLAB ПРИ ПРОЕКТИРОВАНИИ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ**

### **5.1. Постановка задачи и выбор методики синтеза САУ**

Задача синтеза системы автоматического управления (САУ) заключается в выборе такой ее структуры, параметров, характеристик и способов их реализации, которые при заданных ограничениях наилучшим образом удовлетворяют требованиям, предъявляемым к системе.

Обычно определенная часть проектируемой системы задана. Она является исходной или нескорректированной САУ. Параметры ее основных функциональных элементов известны. В такой постановке задача проектирования сводится к определению управляющего устройства (УУ), обеспечивающего заданные показатели качества системы.

Для упрощения структуры УУ оно разбивается на ряд корректирующих устройств (КУ) путем организации в САУ внутренних вложенных друг в друга контуров, каждый из которых управляется соответствующим регулирующим устройством (регулятором).

Для расчета внутренних контуров целесообразно использовать методику последовательной коррекции с подчиненным регулированием координат. В её основе лежит настройка внутренних контуров по быстродействию на так называемый модульный оптимум. Последний (главный) контур синтезируется по точности путем настройки на симметричный оптимум. В результате получаются обычно простейшие КУ в виде типовых промышленных регуляторов (ПИД, ПИ и т.д.). Такие регуляторы удобны в настройке и эксплуатации.

В настоящее время при проектировании и исследовании САУ широко используются специальные программно-вычислительные системы. Наиболее применяемой в области автоматике является

среда MatLab с рядом своих приложений, в особенности, таких как *Control System Toolbox* и *Simulink*. Их использование значительно упрощает все рутинные расчеты и исключает вычислительные ошибки. Поэтому данные пакеты расширения MatLab можно использовать на всех этапах проектирования, где это целесообразно.

Далее рассматривается методика проектирования САУ с применением MATLAB на основе примера синтеза следящего электропривода, исходная схема представлена на рис. 5.1.

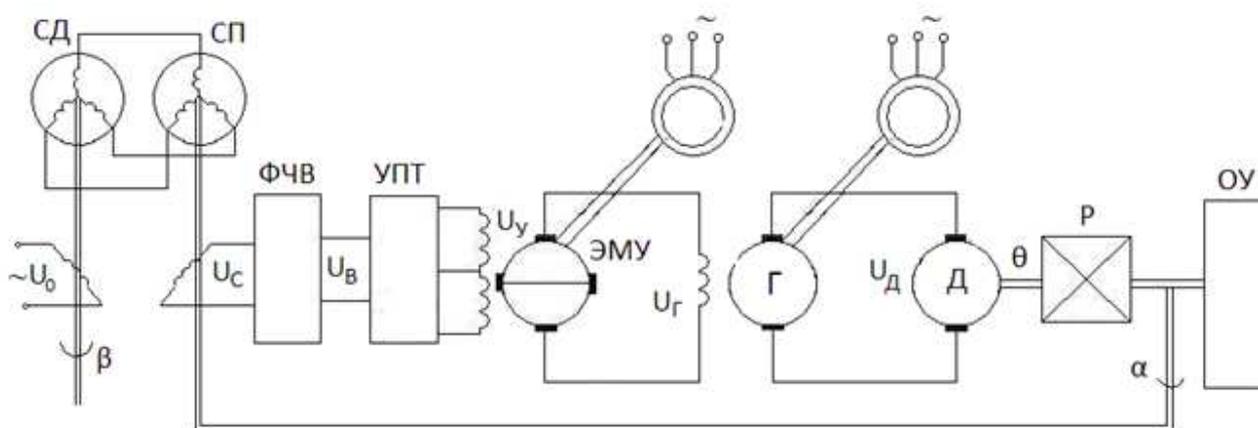


Рис. 5.1. Следящий электропривод на основе системы ЭМУ-Г-Д

В этой схеме обозначено:  $D$  – электродвигатель постоянного тока независимого возбуждения,  $ЭМУ$  – электромашинный усилитель,  $Г$  – генератор постоянного напряжения,  $СД$  – сельсин-датчик,  $СП$  – сельсин-приемник,  $ФЧВ$  – фазочувствительный выпрямитель,  $УПТ$  – усилитель постоянного тока,  $АД$  – асинхронный двигатель,  $Р$  – редуктор,  $ОУ$  – объект управления.

Задающим воздействием в данной системе является угол поворота ротора сельсина-датчика  $\beta$ , регулируемой координатой – угол поворота выходного вала редуктора  $\alpha$ , который измеряется сельсином-приемником. Пара  $СД$  и  $СП$  включены по трансформаторной схеме, которая позволяет формировать сигнал рассогласования (переменное напряжение  $U_C$ ), пропорциональный ошибке слежения  $\delta = \alpha - \beta$ . Этот сигнал подается на  $ФЧВ$ , который выпрямляет напряжение  $U_C$  с учетом его величины и фазы.

Последняя определяется знаком  $\delta$ , отсюда следует, что выходное напряжение фазо-чувствительного выпрямителя  $UB$  имеет пропорциональную зависимость от ошибки  $\delta$ .

Постоянное напряжение  $UB$  усиливается УПТ и подается в прямую и обратную обмотки управления ЭМУ в соответствии со знаком  $UB$ . Электромашинный усилитель выполняет роль усилителя мощности. По существу – это двухкаскадный генератор постоянного напряжения, который приводится в движение от асинхронного двигателя. Последний питается от трехфазной сети и вращается с некоторой постоянной скоростью. Напряжение на выходе электромашинного усилителя  $UD$  пропорционально напряжению, подаваемому в обмотку управления  $UU$ . Под действием напряжения  $UD$  двигатель вращается с определенной скоростью в прямом или обратном направлении в зависимости от знака  $UD$ .

Редуктор  $P$  в следящих системах обычно существенно понижает обороты, т.е. его коэффициент передачи значительно меньше единицы. В данной системе объектом управления по существу является двигатель с редуктором, т.к. обратная связь организована по положению вала  $P$ . Однако с учетом того, что ОУ жестко связан с валом  $P$  можно считать, что объектом управления является система  $D$  - ОУ. Работает САУ следующим образом. Если угол поворота  $\beta$  не изменяется, то  $\alpha=\beta$  и  $\delta=0$ , все напряжения  $UC$ ,  $UB$ ,  $UU$ ,  $UD$  равны нулю и двигатель не вращается. При повороте вала  $CD$  на некоторый угол в ту или иную сторону возникнет рассогласование  $\delta = \alpha - \beta$ , появятся соответствующие напряжения, включая напряжение на двигателе  $D$ . Последний начнет вращаться вместе с  $P$  и  $СП$  в том же направлении, что и  $CD$ . Как только выполнится равенство  $\alpha=\beta$  двигатель  $D$  и редуктор  $P$  остановятся. Если угол  $\beta$  изменять постоянно, то он будет отслеживаться системой с определенной ошибкой  $\delta$ . Очевидно, чем выше скорость и ускорение входного вала  $\beta$ , тем больше ошибка слежения.

Параметры нескорректированного электропривода и требуемые показатели качества представлены в табл. 5.1.

Таблица 5.1

Наименование данных	Параметры
Постоянная времени поперечной цепи ЭМУ, ТК, с	0,5
Постоянная времени якорной цепи Г и Д, ТЭ, с	0,08
Постоянная времени обмотки воз-буждения Г, ТГ, с	0,4
Электромеханическая постоянная времени Д, ТМ, с	0,3
Постоянная времени обмотки управления ЭМУ, ТУ, с	0,006
Коэффициент передачи сельсинов КС, в/рад	10
Коэффициент передачи УПТ, КУ	4
Коэффициент передачи ЭМУ, КЭ	5,5
Коэффициент передачи Д, КД, рад/с/В	1,0
Коэффициент передачи Р, КР	0,008
Коэффициент передачи ФЧВ, КВ	0,6
Коэффициент передачи Г, КГ	1,5
Максимальная скорость $\Omega_m$ , град/с	17
Максимальное ускорение $\varepsilon_m$ , град/с <sup>2</sup>	15
Максимальная ошибка ед, угл.мин	25
Перерегулирование $\sigma$ , %	18

Анализ исходных данных на проектирование и предлагаемого подхода синтеза САУ позволяет выделить следующие основные этапы проектирования:

- построение структурной схемы нескорректированной системы;
- определение показателей качества исходной системы, в том числе, оценка точности, запаса устойчивости и быстродействия;
- синтез внутренних контуров САУ методом последовательной коррекции с подчиненным регулированием координат;
- синтез основного контура регулирования с использованием настройки на симметричный оптимум;
- оценка качества скорректированной системы.

## 5.2. Построение структурной схемы исходной системы

Структурная схема САУ строится на основе её функциональной схемы с учетом принципа действия входящих в состав следящей системы узлов и блоков, а также исходных данных на проектирование. Например, для САУ, приведенной на рис. 5.1, можно предложить структурную схему, которая представлена на рис. 5.2.

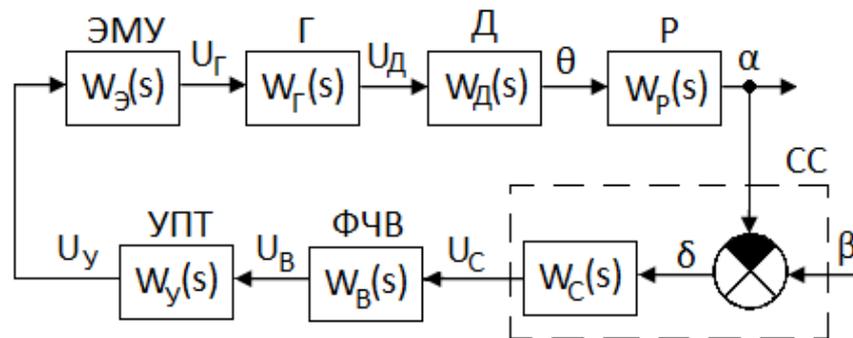


Рис. 5.2. Структурная схема следящей системы

Здесь СС – схема сравнения, реализующая операцию

$\delta = \beta - \alpha$ ;  $W_C(s)$  – ПФ преобразования ошибки  $\delta$  в напряжение  $U_C$ ;  $W_B(s)$  – ПФ фазочувствительного выпрямителя ФЧВ;  $W_Y(s)$  – ПФ усилителя постоянного тока УПТ;  $W_Э(s)$  – ПФ электромашинного усилителя ЭМУ;  $W_Г(s)$  – ПФ генератора Г;  $W_Д(s)$  – передаточная функция электродвигателя Д;  $W_Р(s)$  – ПФ редуктора Р.

Как показывает практика, наибольшую трудность при построении этой структурной схемы представляет замена функционального преобразования, выполняемого сельсинной парой (СД и СП) по соотношению  $U_C = W_C(s)\delta = W_C(s) \cdot (\beta - \alpha)$ , структурой, помеченной на рис. 5.2 пунктирной рамкой, так как рассогласование  $\delta$  измеряется косвенно.

Передаточные функции звеньев САУ определяются на основе параметров устройств САУ, приведенных в табл. 5.1. В итоге получим:

1) ПФ двигателя  $W_D(s)$

$$W_D(s) = \frac{\theta(s)}{U_D(s)} = \frac{K_D}{s(T_\Theta T_M s^2 + T_M s + 1)},$$

где  $\theta(s)$  – угол поворота выходного вала Д;

2) ПФ генератора  $W_G(s)$

$$W_G(s) = \frac{U_D(s)}{U_G(s)} = \frac{K_G}{T_G s + 1};$$

3) ПФ электромашинного усилителя  $W_\Theta(s)$

$$W_\Theta(s) = \frac{U_G(s)}{U_Y(s)} = \frac{K_\Theta}{(T_K s + 1)(T_Y s + 1)};$$

4) ПФ сельсинной пары

$$W_C(s) = \frac{U_C(s)}{\delta(s)} = K_C;$$

ПФ фазочувствительного выпрямителя  $W_B(s)$

$$W_B(s) = \frac{U_B(s)}{U_C(s)} = K_B;$$

6) ПФ электронного усилителя  $W_Y(s)$

$$W_Y(s) = \frac{U_Y(s)}{U_B(s)} = K_Y;$$

7) ПФ редуктора  $W_P(s)$

$$W_P(s) = \frac{\alpha(s)}{\theta(s)} = K_P.$$

Для упрощения расчетов и исследования характеристик САУ рекомендуется использовать пакет прикладных программ *Control System Toolbox*, который является приложением системы *MatLab* и

ориентирован на решение задач по теории автоматического управления.

Для применения функций *Control System Toolbox* в первую очередь следует ввести в среду *MatLab* все полученные выше ПФ. Наиболее удобно это делать в *tf*-форме.

Создание *tf*-модели ПФ электродвигателя с передаточной функцией для скорости его вращения

$$W_D(s) = \frac{\Omega(s)}{U_D(s)} = \frac{K_D}{T_\Omega T_M s^2 + T_M s + 1} = \frac{1}{0.024s^2 + 0.3s + 1}.$$

```
>> Wd=tf([1],[0.024,0.3,1])
```

Transfer function:

$$\frac{1}{0.024 s^2 + 0.3 s + 1}$$

Ввод в среду MATLAB передаточной функции ЭМУ

$$W_\Omega(s) = \frac{K_\Omega}{(T_K s + 1)(T_Y s + 1)} = \frac{5.5}{(0.5s + 1)(0.006s + 1)}.$$

Такую ПФ целесообразно реализовывать как произведение двух *tf*-моделей:

```
>> We=tf([5.5],[0.5,1])*tf([1],[0.006,1])
```

Transfer function:

$$\frac{5.5}{0.003 s^2 + 0.506 s + 1}$$

Ввод ПФ генератора

$$W_G(s) = \frac{K_G}{T_G s + 1} = \frac{1.5}{0.4s + 1}.$$

```
>> Wg=tf([1.5],[0.4,1])
```

Transfer function:

$$\frac{1.5}{0.4 s + 1}$$

Ввод ПФ редуктора (с перенесенным в него от электродвигателя интегратора)

$$W_P(s) = \frac{\alpha(s)}{\Omega(s)} = \frac{K_P}{s} = \frac{0.008}{s}$$

```
>> Wr=tf([0.008],[1,0])
```

Transfer function:

$$\frac{0.008}{s}$$

Ввод остальных ПФ пропорциональных звеньев

$$W_C(s)=K_C=10, W_B(s)=K_B=0.6, W_Y(s)=K_Y=4,$$

можно осуществить простым присваиванием значений коэффициентов передачи, а именно:

```
>> Wc=10      % ввод ПФ сельсинной пары
Wc = 10
```

```
>> Wv=0.6     % ввод ПФ ФЧВ
Wv = 0.6000
```

```
>> Wy=4       % ввод ПФ УПТ
Wy = 4
```

Теперь с *tf*-моделями передаточных функций можно осуществлять необходимые действия и функциональные преобразования.

### 5.3. Исследование исходной системы

Анализ исходной САУ сводится к определению её точности, устойчивости и быстродействия. Для решения этих задач необходимо в первую очередь получить ПФ разомкнутой системы  $W(s)$ , которая равна произведению ПФ всех звеньев замкнутого контура регулирования. Таким образом для структуры, приведенной на рис. 5.1, имеем:

$$W(s) = W_C(s)W_B(s)W_Y(s)W_{\Xi}(s)W_{\Gamma}(s)W_{\Delta}(s)W_P(s)$$

или

$$W(s) = \frac{K}{s(T_K s + 1)(T_V s + 1)(T_{\Gamma} s + 1)(T^2 s^2 + 2\xi T s + 1)} =$$

$$= \frac{1.584}{s(0.5s + 1)(0.006s + 1)(0.4s + 1)(0.024s^2 + 0.3s + 1)},$$

где  $K = K_C K_B K_Y K_{\Xi} K_{\Gamma} K_{\Delta} K_P = 1.584$  – коэффициент передачи разомкнутой системы.

Передаточной функции разомкнутой системы соответствует *tf*-модель, получаемая перемножением *tf*-моделей передаточных функций всех звеньев замкнутого контура САУ. В рассматриваемом случае получим:

```
>> W=Wd*We*Wg*Wf*Wc*Wv*Wy
```

Transfer function:

$$\frac{1.584}{2.88e-005s^6+0.00529s^5+0.08456s^4+0.5012s^3+1.206s^2+s}$$

## Оценка точности нескорректированной системы

Точность работы следящей системы оценивается ошибкой обработки входного воздействия  $\beta(t)$ , представленного в задании на проектирование максимальной скоростью  $\Omega_m$  и максимальным ускорением  $\varepsilon_m$ . По этим двум параметрам можно сформировать эквивалентное гармоническое входное воздействие следующего вида

$$\beta(t) = \beta_m \sin \omega_k t,$$

где  $\beta_m = \frac{\Omega_m^2}{\varepsilon_m}$  – амплитудное значение  $\beta(t)$ ;

$\omega_k = \frac{\varepsilon_m}{\Omega_m}$  – частота гармонического воздействия.

Согласно заданию  $\Omega_m = 17$  град/с = 0.2967 рад/с,  $\varepsilon_m = 15$  град/с<sup>2</sup> = 0.2618 рад/с<sup>2</sup>. Введем эти параметры в среду MatLab:

```
>> qm=0.2967          % ввод максимальной скорости
qm = 0.2967
```

```
>> em=0.2618         % ввод максимального ускорения
em = 0.2618
```

Определим амплитудное значение  $\beta_m$  и частоту эквивалентного гармонического воздействия  $\omega_k$ :

```
>> Vm=(qm^2)/em
Vm = 0.3363
```

```
>> wk=em/qm
wk = 0.8824
```

Ошибка слежения определяется уравнением

$$\delta(s) = \Phi_\beta^\delta(s)\beta(s),$$

где  $\Phi_{\beta}^{\delta}(s)$  – передаточная функция для ошибки  $\delta$  по входному воздействию  $\beta$ :

$$\Phi_{\beta}^{\delta}(s) = \frac{1}{1 + W(s)},$$

где  $W(s)$  – ПФ разомкнутой системы.

Для определения  $\Phi_{\beta}^{\delta}(s)$  целесообразно воспользоваться функцией *feedback* ( $W1, W0$ ), применяемой для вычисления ПФ встречно-параллельного соединения двух звеньев, где  $W1$  – охватываемая модель,  $W0$  – модель отрицательной обратной связи. В рассматриваемом примере *tf*-модель ПФ для ошибки определяется следующим образом. Полагая  $W1=1$ , а  $W0=W(s)$ , получим:

```
>> Fe=feedback(1,W)
```

Transfer function:

$$\frac{2.88e-005s^6+0.00529s^5+0.08456s^4+0.5012s^3+1.206s^2+s}{2.88e-05s^6+0.00529s^5+0.08456s^4+0.5012s^3+1.206s^2+s+ 1.584}$$

Учитывая, что при гармоническом входном воздействии  $\beta(t)$  рассогласование  $\delta(t)$  также изменяется гармонически, можно воспользоваться частотным методом оценки точности. Для определения значения частотной передаточной функции  $\Phi_{\beta}^{\delta}(s)$  при  $s = j\omega_K$ , удобнее всего воспользоваться функцией *freqresp* ( $Fe, wk$ ).

Для рассматриваемого примера получим следующее максимальное значение ошибки:

```
>> Em=freqresp(Fe,wk)*Bm
Em = -0.1408 + 0.3706i
```

т.е. имеем комплексное значение рассогласования  $\delta_m$ .

Для оценки амплитудного значения ошибки следует перейти к модульному соотношению для  $\delta(t)$ :

$$\delta_m = \left| \Phi_\beta^\delta(j\omega_K) \right| \beta_m.$$

В среде MatLab это осуществляется с помощью функции абсолютного значения *abs*:

```
>> Em=abs(Em)
```

```
Em = 0.3964
```

Полученное значение  $\delta_m$  сравнивается с допустимой величиной ошибки слежения и делается соответствующий вывод. В нашем случае ошибка системы оказалась значительно больше допустимой  $\epsilon_d=25$  угл.мин. = 0.0073 рад (табл. 5.1), т.е. точность работы исходной САУ не удовлетворяет техническому заданию на проектирование.

### ***Определение устойчивости исходной системы***

Устойчивость замкнутой САУ удобнее всего оценивать с помощью логарифмического критерия Найквиста. Для этого необходимо построить логарифмические частотные характеристики (ЛЧХ) разомкнутой системы. В *Control System Toolbox* для расчета ЛЧХ применяются две функции *bode* и *margin*. Удобнее использовать последнюю, т.к. результатом её выполнения является не только графики ЛЧХ, но и численные значения запаса по фазе  $Pm$ , частоты среза  $\omega_{cp}$  и запаса по амплитуде  $Gm$ .

Применим функцию *margin* для исследования запаса устойчивости рассматриваемого варианта следящей системы:

```
>> margin(W) ; grid on
```

В результате получаются ЛЧХ, представленные на рис. 5.3. Из них видно, что система имеет очень малый запас по фазе  $\gamma = Pm = 13^\circ$ , т.е. замкнутая САУ устойчива, но находится близко к границе устойчивости.

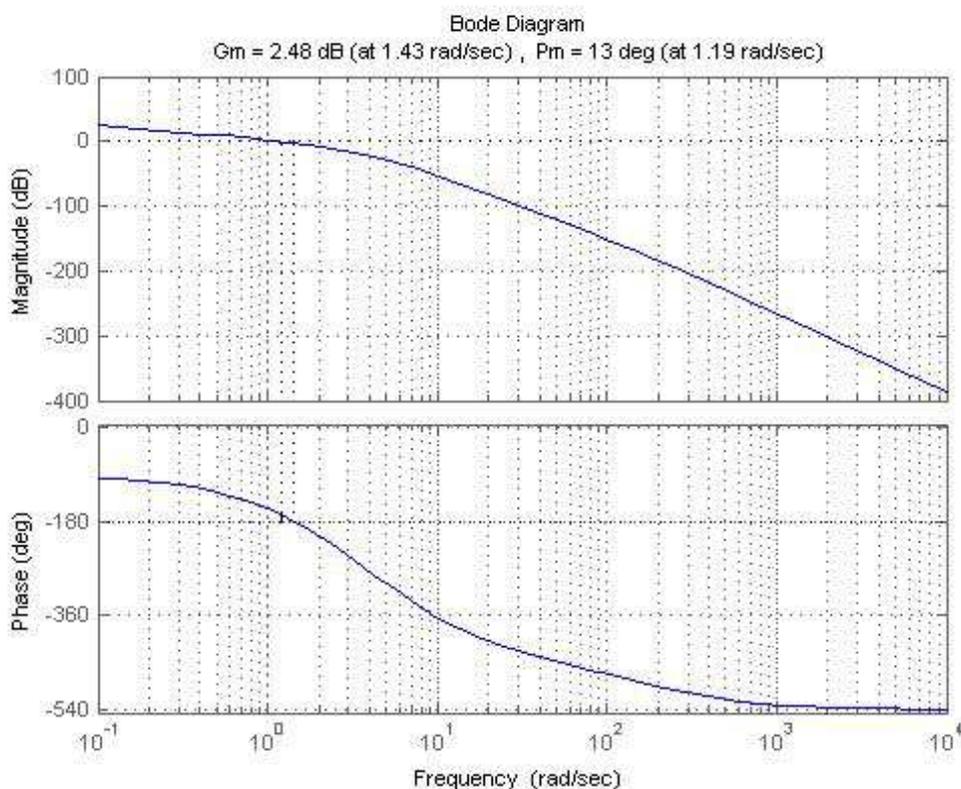


Рис. 5.3. ЛЧХ разомкнутой САУ

Следует отметить, что в удовлетворительно спроектированных системах запас по фазе лежит в диапазоне  $\gamma = 45 \div 70^\circ$ .

### ***Определение динамических показателей качества***

Для оценки динамических свойств САУ обычно используют два показателя качества: перерегулирование  $\sigma$  и длительность переходного процесса  $t_n$ , которые определяются из переходной характеристики замкнутой САУ. Для построения данной характеристики в *Control System Toolbox* применяется функция *step*,

аргументом которой является передаточная функция замкнутой системы

$$\Phi(s) = \frac{W_1(s)}{1 + W_1(s)W_0(s)}$$

где  $W_1(s)$  – передаточная функция прямого канала САУ,  
 $W_0(s)$  – ПФ цепи обратной связи.

Для получения  $\Phi(s)$  можно воспользоваться функцией *feedback* ( $W_1, W_0$ ). Для рассматриваемого примера имеем  $W_1(s) = W(s)$ ,  $W_0(s) = 1$  и соответственно:

```
>> F=feedback (W,1)
```

Transfer function:

$$\frac{1.584}{2.88e-05s^6+0.00529s^5+0.0845s^4+0.5012s^3+1.206s^2+s+1.584}$$

Далее можно рассчитать переходную характеристику с помощью функции *step*:

```
>> step (F); grid on
```

В результате получим искомый график, представленный на рис. 5.4. В данном случае получился сходящийся колебательный процесс, что подтверждает устойчивость исходной следящей системы с малым запасом по фазе  $\gamma = 13^\circ$ .

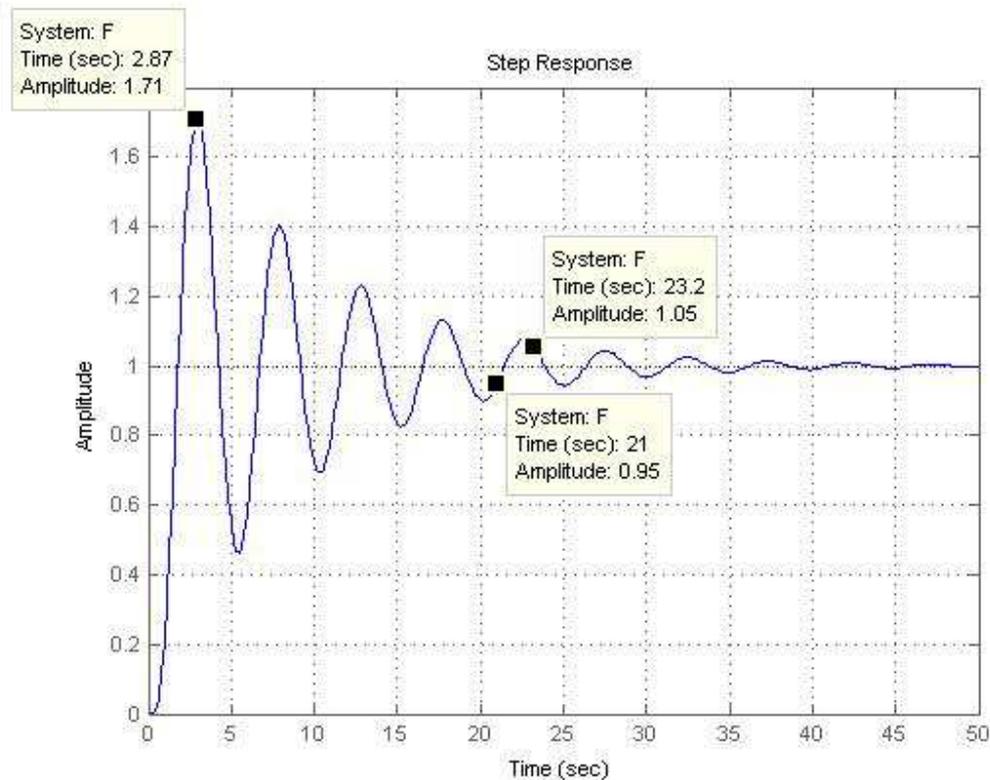


Рис. 5.4. Переходная характеристика замкнутой САУ

Из переходной характеристики с помощью известных в теории автоматического управления способов определяются два основных динамических показателя качества:

$$\text{перерегулирование } \sigma = \frac{y_{\max} - y_{\text{уст}}}{y_{\text{уст}}} \cdot 100\%,$$

где  $y_{\max}$ ,  $y_{\text{уст}}$  – соответственно максимальное и установившееся значения выходной координаты;

$t_{\Pi}$  – длительность переходного процесса, которая равна интервалу времени, за который кривая переходной характеристики входит в зону  $(0,95 \div 1,05) y_{\text{уст}}$ , в дальнейшем не выходя из неё.

Анализ полученной переходной характеристики (рис. 5.4), показывает, что  $\sigma = 70\%$ ,  $t_{\Pi} = 23.3$  с. Это говорит о неудовлетворительном качестве динамических процессов в следящей системе: перерегулирование в реальных САУ обычно лежит в диапазоне  $\sigma = 10 \div 30\%$ , а  $t_{\Pi}$  сравнимо с максимальной постоянной времени контура управления (в нашем случае  $T_K = 0.5$  с).

В заключение по результатам анализа исходной САУ можно сделать следующий общий вывод: система устойчива, но обладает неудовлетворительными показателями качества и поэтому требует соответствующей коррекции.

#### 5.4. Коррекция системы автоматического управления

При проектировании электромеханических систем управления целесообразно применять методику последовательной коррекции с подчиненным регулированием координат.

В используемой в качестве примера САУ можно ввести два внутренних вложенных друг в друга контура регулирования: один по напряжению генератора, второй по скорости электродвигателя. Организация этих обратных связей представлена на рис. 5.5.

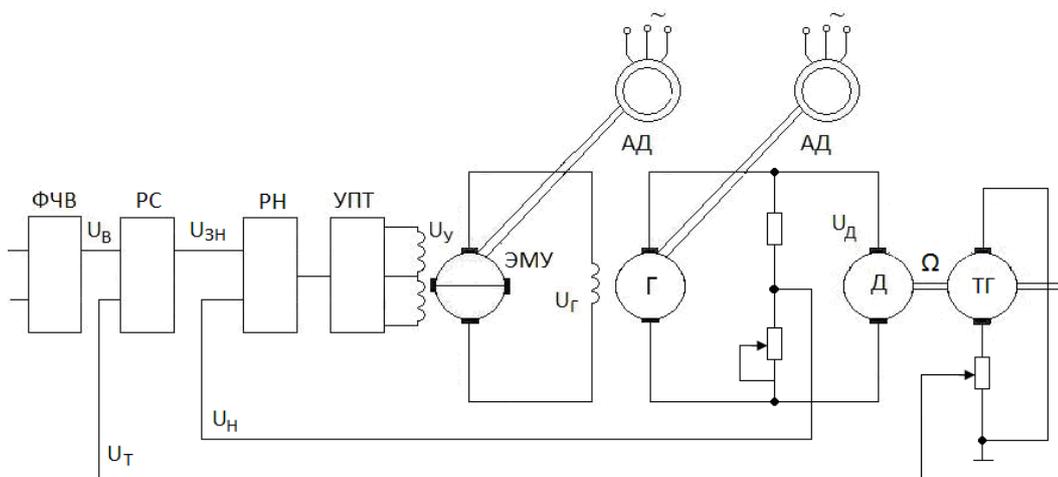


Рис. 5.5. Реализация подчиненного регулирования

В данной схеме регулятор напряжения (РН) и регулятор скорости (РС) помещены между ФЧВ и УПТ, т.к. коррекцию удобнее реализовать на постоянном токе в слаботочной части прямого канала управления.

Здесь датчиком внутреннего контура является делитель напряжения, с которого часть напряжения  $U_D$  подается на вычитающий вход РН. В качестве датчика скорости используется

тахогенератор ТГ с делителем напряжения, выходной сигнал с которого поступает на вычитающий вход РС.

### ***Расчет контура регулирования напряжения***

Структурная схема первого внутреннего контура представлена на рис. 5.6. Здесь  $W_{PH}(s)$  - передаточная функция РН,  $W_H(s) = K_H$  - ПФ делителя напряжения.

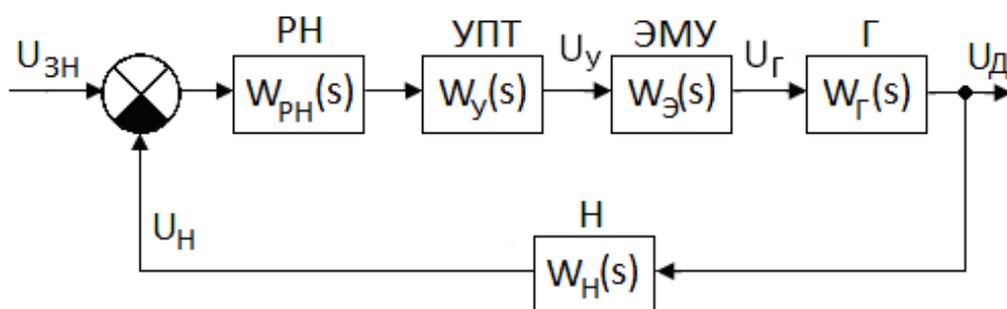


Рис. 5.6. Структурная схема контура регулирования напряжения

В соответствии с методикой подчиненного регулирования координат внутренние контуры настраиваются по быстродействию на так называемый модульный оптимум, которому соответствует желаемая ПФ разомкнутой системы следующего вида

$$W_{Ж}(s) = \frac{1}{2T_{\mu}s(T_{\mu}s + 1)},$$

где  $T_{\mu}$  – нескомпенсированная постоянная времени.

В качестве  $T_{\mu}$  берется наименьшая постоянная времени синтезируемого контура, поэтому она часто называется малой постоянной времени. По существу  $W_{Ж}(s)$  – ПФ реального интегратора. При этом желаемая ПФ замкнутой системы определяется соотношением

$$\Phi_{Ж}(s) = \frac{W_{Ж}(s)}{1 + W_{Ж}(s)} = \frac{1}{2T_{\mu}^2 s^2 + 2T_{\mu} s + 1},$$

т.е. представляет собой колебательное звено с незначительным коэффициентом демпфирования  $\xi = \sqrt{2} / 2 = 0.707$ .

Такой настройке соответствуют высокие динамические показатели качества - перерегулирование  $\sigma$  и длительность переходного процесса  $t_{II}$  достаточно малы ( $\sigma = 4.3\%$ ,  $t_{II} = 4.15 T_{\mu}$ ), а точность приемлема, так как контур обладает астатизмом первого порядка.

Возьмем в качестве некомпенсированной постоянной времени  $T_{\mu}$  постоянную времени обмотки уравнения ЭМУ, т.е. положим  $T_{\mu} = T_Y = 0.006 \text{ с}$ . Тогда желаемая ПФ разомкнутого контура будет иметь следующий вид:

$$W_{Ж1}(s) = \frac{1}{2T_Y s(T_Y s + 1)}$$

Зададимся значением коэффициента передачи датчика напряжения  $K_H$ , исходя из следующих соображений. Максимальное напряжение цепи обратной связи  $U_H$ , подаваемого на РН, ограничим значением  $U_{H \max} = 10 \text{ В}$  при максимальном значении напряжения генератора  $U_{Г \max} = 50 \text{ В}$ , которое равно максимальному напряжению, подаваемому на электродвигатель. Отсюда будем иметь

$$K_H = \frac{10}{50} = 0.2 .$$

Теперь можно определить ПФ разомкнутого контура  $W_{01}(s)$ :

$$W_{01}(s) = W_Y(s)W_{\mathcal{E}}(s)W_{Г}(s)W_H(s) = \frac{K_Y K_{\mathcal{E}} K_{Г} K_H}{(T_K s + 1)(T_Y s + 1)(T_{Г} s + 1)} .$$

После этого рассчитаем ПФ регулятора напряжения  $W_{PH}(s)$  по соотношению

$$W_{PH}(s) = \frac{W_{Ж1}(s)}{W_{01}(s)}.$$

В результате получим

$$W_{PH}(s) = \frac{(T_K s + 1)(T_\Gamma s + 1)}{T_{И1} s},$$

где  $T_{И1} = 2T_y K_y K_\varepsilon K_\Gamma K_H = 0.0792 \text{ с}$  – постоянная времени интегрирования РН.

Поделив числитель ПФ  $W_{PH}(s)$  на ее знаменатель, определим параллельную структуру РН:

$$W_{PH}(s) = K_{П1} + \frac{1}{T_{И1} s} + K_{Д1} s,$$

где  $K_{П1} = \frac{T_K + T_\Gamma}{T_{И1}} = 11.36$  – коэффициент передачи пропорционального звена;

$$K_{Д1} = \frac{T_K T_\Gamma}{T_{И1}} = 2.525 \text{ с} – \text{коэффициент передачи дифференциатора.}$$

Анализ показывает, что РН реализует пропорционально - интегрально - дифференциальный (ПИД) закон управления.

### ***Расчет контура регулирования скорости***

Первый контур регулирования является подчиненным второму и входит в канал управления последнего в виде подсистемы (звена). С учетом этого схему контура регулирования скорости для простоты расчетов целесообразно представить в виде структуры, изображенной на рис. 5.7.

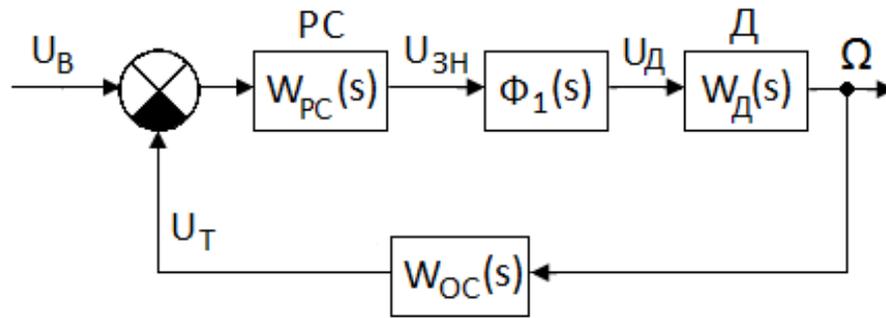


Рис 5.7. Структурная схема контура регулирования скорости

Здесь  $W_{OC}(s) = K_O$  – ПФ цепи обратной связи, включающей ТГ и его выходной делитель напряжения. Коэффициент передачи цепи обратной связи равен  $K_O = U_{T\max} / \Omega_{\max}$ .

Положим, что при максимальной скорости вращения двигателя  $\Omega_{\max} = 37$  рад/с, напряжение цепи обратной связи не должно превышать максимальное значение  $U_T = 10$  В. Чтобы система не работала в режиме насыщения, возьмем максимальную скорость с запасом  $\Omega_{\max} = 50$  рад/с.

Тогда получим

$$K_O = U_{T\max} / \Omega_{\max} = 10 / 50 = 0.2 \text{ в} \cdot \text{с/рад}.$$

Передаточная функция контура регулирования напряжения  $\Phi_1(s)$  равна

$$\Phi_1(s) = \frac{W_{O1}(s) / W_H(s)}{1 + W_{O1}(s)} = \frac{1}{K_H(2T_Y^2 s^2 + 2T_Y s + 1)}.$$

Согласно методике коррекции с подчиненным регулированием координат в знаменателе  $\Phi_1(s)$  следует пренебречь членом  $2T_Y^2 s^2$ , т.к. он очень мал. Это приводит к небольшой погрешности при расчете регулятора скорости РС, но существенно упрощает получение его передаточной функции  $W_{PC}(s)$ . Таким образом

$$\Phi_1(s) \approx \frac{1}{K_H(2T_Y s + 1)}.$$

Далее, как и для контура напряжения необходимо сформировать ПФ объекта управления для регулятора скорости. Из рис. 5.7 следует, что она равна

$$W_{02}(s) = \Phi_1(s)W_D(s)W_{OC}(s) = \frac{K_D K_O}{(2T_Y s + 1)(T_M T_\Delta s^2 + T_M s + 1)K_H}.$$

В качестве малой некомпенсированной постоянной времени  $T_{\mu 1}$  берется величина  $2T_Y$ , т.е.  $T_{\mu 1} = 2 \cdot 0.006 = 0.012$  с. Тогда желаемая ПФ контура скорости при настройке на модульный оптимум будет иметь следующий вид

$$W_{Ж2}(s) = \frac{1}{2T_{\mu 1} s (T_{\mu 1} s + 1)}.$$

Далее можно определить ПФ регулятора скорости:

$$W_{PC}(s) = \frac{W_{Ж2}(s)}{W_{02}(s)} = \frac{T_M T_\Delta s^2 + T_\Delta s + 1}{T_{И2} s},$$

где  $T_{И2} = 2T_{\mu 1} K_D K_O / K_H = 2 \cdot 0.012 \cdot 1 \cdot 0.2 / 0.2 = 0.024$  с – постоянная времени интегрирования РС.

Полученная ПФ регулятора скорости показывает, что РС также, как и РН является ПИД - регулятором:

$$W_{PC}(s) = \frac{T_M}{T_{И2}} + \frac{1}{T_{И2} s} + \frac{T_M T_\Delta}{T_{И2}} s = K_{П2} + \frac{1}{T_{И2} s} + K_{Д2} s,$$

где  $K_{П2} = \frac{2\xi T}{T_{И2}} = \frac{2 \cdot 0.97 \cdot 0.155}{0.024} = 12.53$  – коэффициент передачи

пропорционального звена;

$K_{Д2} = \frac{T^2}{T_{И2}} = \frac{0.155^2}{0.024} = 1$  с – коэффициент передачи дифференциатора.

## Синтез основного контура управления

Структурная схема контура положения представлена на рис. 5.8. Здесь  $\Phi_{02}(s)$  – ПФ замкнутого контура скорости, определяемая по методике, которая была использована для расчета ПФ контура напряжения:

$$\Phi_{02}(s) = \frac{W_{02}(s)/W_{OC}(s)}{1 + W_{02}(s)} = \frac{1}{K_O(2T_{\mu 1}^2 s^2 + 2T_{\mu 1} s + 1)} \approx$$

$$\approx \frac{1}{K_O(2T_{\mu 1} s + 1)} = \frac{1}{K_O(T_{\mu 2} s + 1)},$$

где  $T_{\mu 2} = 2T_{\mu 1} = 0.024\text{с}$  – эквивалентная малая постоянная времени.

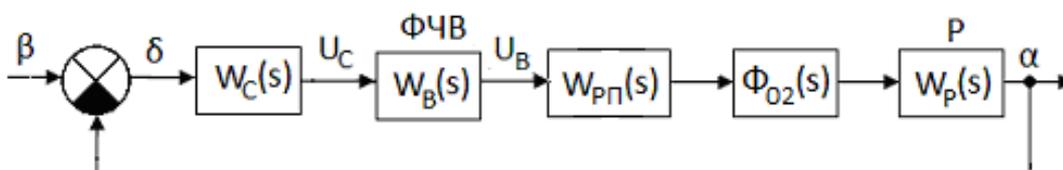


Рис. 5.8. Структурная схема контура положения

Синтез регулятора положения РП будем осуществлять путем настройки на симметричный оптимум, которому соответствует желаемая ПФ следующего вида

$$W_{ЖЗ}(s) = \frac{4T_{\mu 2} s + 1}{4T_{\mu 2} s} \frac{1}{2T_{\mu 2} s(T_{\mu 2} s + 1)}.$$

Определим предварительно ПФ исходного разомкнутого контура положения

$$W_{03}(s) = W_C(s)W_B(s)\Phi_{02}(s)W_P(s) = \frac{K_3}{s(T_{\mu 2} s + 1)},$$

где  $K_3 = K_C K_B K_P / K_O = 10 \cdot 0.6 \cdot 0.008 / 0.2 = 0.24\text{с}^{-1}$ .

Теперь можно определить ПФ регулятора положения:

$$W_{PI}(s) = \frac{W_{ЖЗ}(s)}{W_{ОЗ}(s)} = \frac{4T_{\mu 2}s + 1}{T_{ИЗ}s},$$

где  $T_{ИЗ} = 8T_{\mu 2}K_3 = 8 \cdot 0.024 \cdot 0.24 = 0.046$  с.

Таким образом, РП реализует пропорционально-интегральный закон управления (ПИ – регулятор):

Исследование скорректированной САУ можно отнести к заключительному этапу. Как и для исходной системы в итоге оцениваются все основные показатели качества, характеризующие точность, запас устойчивости и качество переходного процесса.

Для этого необходимо предварительно определить передаточные функции разомкнутой  $W(s)$  и замкнутой  $\Phi(s)$  скорректированной системы, не упрощая ПФ внутренних контуров регулирования. Указанные операции удобнее осуществлять в среде MATLAB с использованием функций приложения *Control System Toolbox*. Сначала требуется создать *tf*-модели всех реализованных регуляторов, а затем уже приступить к преобразованию структурной схемы. При этом следует исследовать каждый контур на запас устойчивости и динамические показатели качества, ибо при реализации регуляторов могли возникнуть неточности в расчетах и даже ошибки.

После получения ПФ разомкнутой  $W(s)$  и замкнутой  $\Phi(s)$  системы можно приступить к оценке качества скорректированной системы по той же методике, как и для исходной САУ.

## ЗАКЛЮЧЕНИЕ

Полученные знания и умения по использованию среды MATLAB для решения задач теории автоматического управления позволят студентам и специалистам применять их для проектирования и исследования более сложных систем, чем рассмотренная в последней главе. При этом круг используемых функций и команд может быть расширен за счет ряда дополнительных функций MATLAB. Их можно найти в специальной и справочной литературе по данному пакету [3, 5, 6]. Это касается в первую очередь команд и функций для ss-моделей, которые применяются в методе переменных состояния, являющегося основной современной теории управления.

Освоение предлагаемого материала на уровне практических умений по *Control System Toolbox* и *Simulink* открывает путь к изучению более «серьезных» приложений MATLAB, используемых для анализа и синтеза САУ. К ним можно отнести следующие *Toolbox*:

- *Frequency Domain System Identification*, предназначенный для идентификации линейных динамических систем по их временным и частотным откликам;
- *LMI Control*, используемый для решения задач оптимизации с помощью алгоритмов выпуклого и линейного программирования;
- *Model Predictive Control*, ориентированный на реализацию стратегии предиктивного управления сложными многоканальными процессами при наличии ограничений на переменные состояния и управляющие воздействия;
- $\mu$ -*Analysis and Synthesis*, используемый для проектирования устойчивых систем управления с оптимизацией в равномерной норме;

- *Robust Control*, применяемый для построения и исследования многопараметрических САУ с нестабильными или не полностью определенными параметрами;
- *System Identification*, который содержит инструменты для получения математической модели динамического объекта на основе наблюдаемых входных и выходных координат системы.

Следует также заметить, что пользователь может создавать свои функции в виде *m*-файлов и использовать их для решения определенного круга задач. Так собственно и делают опытные специалисты.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бесекерский В. А., Попов Е. П. Теория систем автоматического управления.– Изд. 4-е, перераб. и доп. – СПб, Изд-во «Профессия», 2003.– 752 с.: ил.
2. Гайдук А.Р., Беляев В.Е., Пьявченко Т.А. Теория автоматического управления в примерах и задачах с решениями в MATLAB: Учебное пособие. – СПб.: Издательство "Лань", 2011. – 464 с.
3. Дьяконов В., Круглов В. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. – СПб.: Питер, 2002.– 448.: ил.
4. Мирошник И.В. Теория автоматического управления. Линейные системы. – СПб.: Питер, 2005. – 336 с.: ил. – (Серия «Учебное пособие»).
5. Перельмуттер В.М. Пакеты расширения MATLAB. Control System Toolbox и Robust Control Toolbox. – М.: СОЛОН-ПРЕСС, 2008. – 224 с.: ил. – (Серия «Библиотека профессионала»).
6. Потемкин В.Г. Вычисления в среде MATLAB. – М.: ДИАЛОГ-МИФИ, 2004. – 720 с.
7. Теория автоматического управления: учеб. для вузов/ под ред. проф. В.Б. Яковлева. – 3-е изд., стер. – М.: Высш. шк., 2009. – 567 с.: ил.
8. Черных И.В. SIMULINK: среда создания инженерных приложений / Под общ. ред. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2004. – 496 с.
9. Hunt, Brian R. Matlab: официальный учеб. курс Кембриджского университета: [пер. с англ.] / Brian R. Hunt [и др.]М.: Изд-во ТРИУМФ, 2008. – 352 с.: ил.

## ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ.....	3
ВВЕДЕНИЕ.....	5
1. Формирование математических моделей с помощью пакета <i>Control System Toolbox</i> .....	8
1.1. Формирование tf-объектов.....	8
1.2. Формирование zpk-объектов.....	10
1.3. Формирование ss-объектов.....	12
1.4. Формирование дискретных lti-объектов.....	14
1.5. Построение lti-моделей с запаздыванием.....	18
1.6. Построение дискретных моделей непрерывных систем.....	19
1.7. Построение непрерывных моделей дискретных систем.....	22
1.8. Переопределение дискретной системы.....	23
1.9. Взаимное преобразование lti-объектов.....	24
1.10. Типовые соединения lti-моделей.....	25
1.11. Установка и извлечение значений свойств lti-объектов.....	30
2. Анализ линейных стационарных систем с помощью пакета <i>Control System Toolbox</i> .....	41
2.1. Динамические параметры lti-моделей.....	42
2.2. Частотные характеристики.....	48
2.3. Временные характеристики.....	58
3. Моделирование непрерывных систем в среде MATLAB с помощью пакета <i>Simulink</i> .....	65
3.1. Запуск <i>Simulink</i> .....	66
3.2. Интерфейс обозревателя библиотеки <i>Simulink</i> .....	68
3.3. Создание S-модели.....	70
3.4. Установка параметров блоков S-модели.....	79
3.5. Задание параметров моделирования и его выполнение.....	81
4. Параметрическая настройка динамических систем с помощью пакета <i>Simulink Response Optimization</i> .....	88
4.1. Подготовка САУ к параметрической оптимизации.....	89
4.2. Процесс параметрической настройки системы.....	94
5. Использование MATLAB при проектировании систем автоматического управления .....	99

5.1. Постановка задачи и выбор методики синтеза САУ.....	99
5.2. Построение структурной схемы исходной системы.....	103
5.3. Исследование исходной системы.....	107
5.4. Коррекция системы автоматического управления.....	114
ЗАКЛЮЧЕНИЕ.....	122
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	124

*Учебное пособие*

*БУДИН Владимир Иванович  
ДРЕМОВ Феликс Владимирович*

## **Теория автоматического управления в среде MatLab**

Редакторы:  
*Е.С. Захарова  
И. А. Назарова*

Подписано в печать 30.12.2013г.  
Формат 60x84 1/16. Бумага офсетная  
Усл. п. л. 7,4 Уч.-изд. л. 5,7  
Тираж 50 экз. Рег. № 6/14sf

---

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Самарский государственный технический университет»  
443100, г. Самара, ул. Молодогвардейская, 244. Главный корпус

Отпечатано в типографии  
Самарского государственного технического университета  
Филиал в г. Сызрани, 446001, г. Сызрань, ул. Советская 45