

Министерство образования и науки Российской Федерации
ФГАОУ ВПО «УрФУ имени первого Президента России Б.Н. Ельцина»



А.В. Малов, Е.Э. Страшинин, А.В. Цветков

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ УПРАВЛЕНИЯ ТЕХНИЧЕСКИМИ СИСТЕМАМИ

Учебное электронное текстовое издание

Подготовлено кафедрой автоматики

Методические указания к практическим занятиям
для магистрантов направления
090900.68 – Информационная безопасность

© УрФУ, 2015

Екатеринбург

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. РАСЧЕТ МАТРИЧНЫХ ВЕСОВЫХ И ПЕРЕХОДНЫХ ФУНКЦИЙ	3
1.1. Цель занятия.....	3
1.2. Подготовительная часть.....	4
1.3. Компьютерное моделирование	5
1.4. Оформление материалов.....	15
1.5. Контрольные вопросы.....	15
2. ИЗУЧЕНИЕ ТИПОВЫХ ЗВЕНЬЕВ	15
2.1. Цель занятия.....	15
2.2. Подготовительная часть занятия	16
2.3. Компьютерное моделирование	18
2.4. Оформление материалов.....	23
2.5. Контрольные вопросы.....	23
3. ЧАСТОТНЫЕ ХАРАКТЕРИСТИКИ СИСТЕМ И КРИТЕРИЙ УСТОЙЧИВОСТИ НАЙКВИСТА	24
3.1. Цель занятия.....	24
3.2. Подготовительная часть.....	25
3.3. Компьютерное моделирование	25
3.4. Оформление материалов.....	29
3.5. Контрольные вопросы.....	29
4. МОДАЛЬНЫЙ СИНТЕЗ УПРАВЛЕНИЯ В ЛИНЕЙНЫХ НЕПРЕРЫВНЫХ СИСТЕМАХ	30
4.1. Цель занятия.....	30
4.2. Подготовительная часть.....	30
4.3. Компьютерное моделирование	31
4.4. Оформление материалов.....	37
4.5. Контрольные вопросы.....	38
СПИСОК ЛИТЕРАТУРЫ.....	39
ПРИЛОЖЕНИЕ. Список команд, операторов и функций Matlab, использованных при создании m-файлов	40
1. Команды, операторы и функции общего назначения.....	40
2. Функции создания и преобразования моделей линейных систем с постоянными параметрами (linear time-invariant models), функции извлечения данных о моделях	41
3. Функции синтеза контуров управления с обратной связью	43
4. Функции, используемые для построения графиков.....	43

ВВЕДЕНИЕ

Программа дисциплины предусматривает выполнение практических занятий. В данных указаниях представлены материалы по 4 темам. Занятия проводятся с использованием пакета Matlab. Для сохранения результатов занятий каждому студенту необходимо создать на компьютере в папке, предназначенной для работы пользователей, папку группы, внутри нее папку студента, где и следует размещать поддиректории с результатами занятий.

1. РАСЧЕТ МАТРИЧНЫХ ВЕСОВЫХ И ПЕРЕХОДНЫХ ФУНКЦИЙ

1.1. Цель занятия

В данном занятии средствами пакета Matlab (с использованием его расширения – пакета моделирования динамических систем Simulink) должно быть выполнено моделирование линейной системы, зафиксированы процессы, соответствующие элементам матричной весовой и переходной функций и проведено их сравнение с аналитически полученными зависимостями. Структурная схема системы представлена на рис. 1.1, коэффициенты структурной схемы – в табл. 1.1. Номер варианта указывается преподавателем.

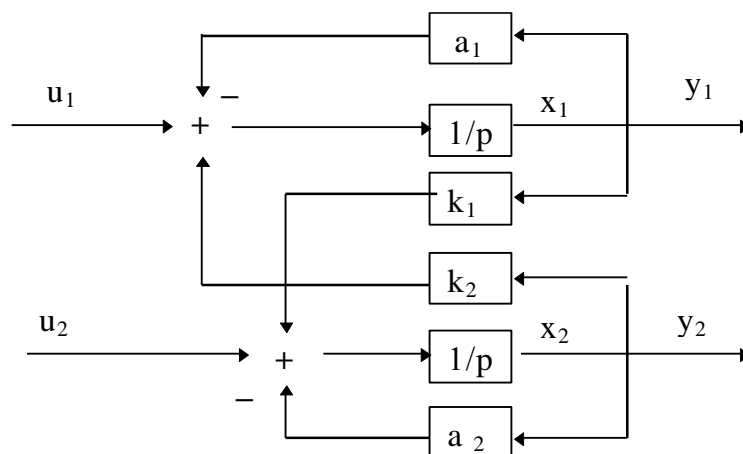


Рис. 1.1. Структурная схема системы

Таблица 1.1

Значения параметров структурной схемы

Номер варианта	1	2	3	4	5	6	7	8	9	10
a ₁	0,1	0,1	3	0,3	0,4	0,8	1	2	2	2
a ₂	0,1	0,3	0,2	0,5	0,6	1,2	3	3	4	6
k ₁	0,1	0,015	0,4	5	3	0,48	1,5	1	4	24
k ₂	0,1	2	1	0,03	0,08	2	2	6	2	0,5
Номер варианта	11	12	13	14	15	16	17	18	19	20
a ₁	3	0,8	0,9	0,9	0,9	1,2	3	4	4	5
a ₂	7	0,4	0,5	0,7	0,9	1,8	2	3	5	6
k ₁	0,5	0,5	2	0,12	0,5	0,1	4	2	6	4
k ₂	42	0,24	0,025	0,25	0,02	1,6	0,5	3	2	5

1.2. Подготовительная часть

1.2.1. Составить векторно-матричное описание системы.

1.2.2. Вычислить передаточную функцию $W_{uy}(p)$ (матрицу 2×2) с использованием резольвенты матрицы динамики A :

$$W_{uy}(p) = C \cdot (pE - A)^{-1} \cdot B = C \cdot \frac{I(p)}{\varphi_A(p)} \cdot B,$$

где B – матрица входа; C – матрица выхода; $I(p)$ – присоединенная матрица для матрицы A ; $\varphi_A(p)$ – характеристический полином матрицы A ($I(p)$ и $\varphi_A(p)$ могут быть определены по методу Фаддеева-Леве́рье).

Найти элементы матричной весовой функции по формуле

$$w_{ij}(t) = L^{-1} \{ W_{ij}(p) \},$$

где $w_{ij}(t)$ – элемент i -й строки и j -го столбца матричной весовой функции, интерпретируемый как реакция i -й координаты вектора выхода \vec{y} на дельта-функцию в j -й координате вектора входа \vec{u} .

1.2.3. Вычислить матричную весовую функцию по формуле

$$w_{uy}(t) = C \cdot e^{At} \cdot B = C \cdot \left(\sum_{k=1}^n e^{\lambda_k t} \vec{v}_k \cdot \vec{d}_k^T \right) \cdot B,$$

где \vec{v}_k и \vec{d}_k – соответственно k-й правый и k-й левый собственные векторы матрицы A. Убедиться в идентичности результатов, полученных в пп. 1.2.2 – 1.2.3.

1.2.4. Вычислить элементы матричной переходной функции по формуле

$$h_{ij}(t) = \int_0^t w_{ij}(\tau) d\tau,$$

где $h_{ij}(t)$ – элемент i-й строки и j-го столбца матричной переходной функции, интерпретируемый как реакция i-й координаты вектора выхода \vec{y} на единичную функцию в j-й координате вектора входа \vec{u} .

1.3. Компьютерное моделирование

В ходе проведения занятия требуется в среде Matlab подготовить схему моделирования исследуемой системы, провести модельный эксперимент и зафиксировать его результаты.

Изучаемая в данном занятии система описана двумя способами: при помощи структурной схемы (см. рис. 1.1) и в виде векторно-матричных уравнений, полученных в ходе подготовки к работе. Поэтому предлагается провести моделирование для двух вариантов описания системы и сравнить его результаты (процессы, соответствующие элементам матричной весовой и переходной функций) с аналитически полученными зависимостями.

Рекомендуется создать две отдельных модели: одну – для получения и фиксации 4 процессов, соответствующих элементам матричной весовой функции, вторую – для 4 процессов, соответствующих элементам матричной переходной функции.

Таким образом, должно быть зафиксировано 8 процессов, причем каждый из них будет представлен в трех вариантах, совмещенных на одном графике (т.к. моделируется система, описанная, во-первых, в виде структурной схемы, во-вторых, в векторно-матричной форме, и, в-третьих, полученными аналитическими зависимостями для $w_{ij}(t)$ и $h_{ij}(t)$).

1.3.1. Для проведения моделирования должны быть созданы 5 файлов.

Первый файл-сценарий `w_h_init.m` содержит определения всех необходимых переменных в моделях:

```
%Файл определения переменных
%
%Параметры моделирования для mdl-файла
t_end = 1;           %максимальный шаг моделирования
h_max = 0.01;        %время завершения моделирования

%Параметры исследуемой системы
a1 = 2;
a2 = 3;
k1 = 1;
k2 = 6;
```

Время моделирования `t_end` и максимальный шаг моделирования `h_max` должны быть выбраны такими, чтобы все процессы смогли достигнуть своих установившихся значений, а графики не имели изломов и искажений.

Второй и третий файлы с моделями Simulink `w.mdl` и `h.mdl` предназначены для расчета и визуализации элементов соответственно весовой и переходной функций (рис. 1.2 и 1.3).

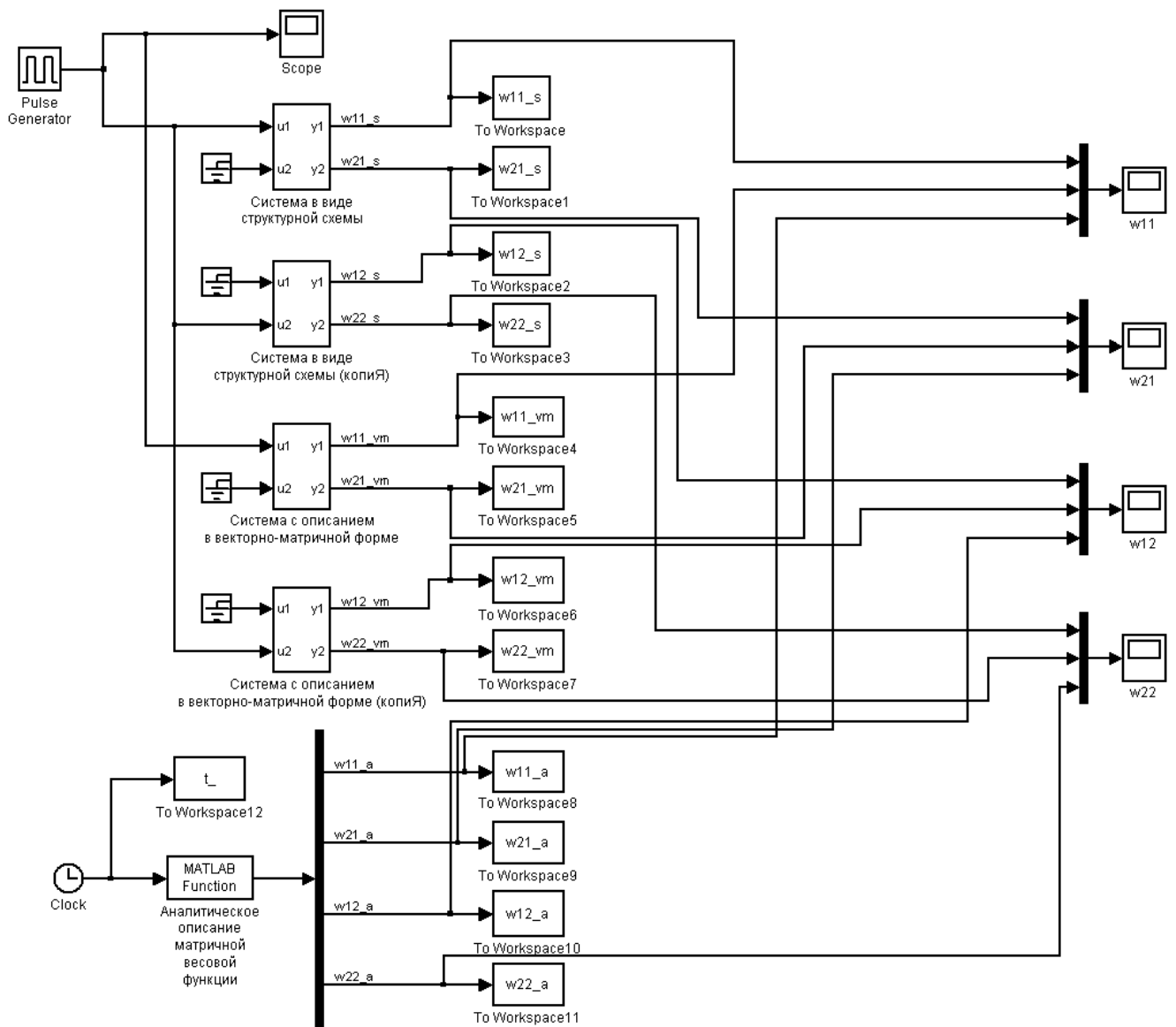


Рис. 1.2. Схема моделирования, используемая для расчета и визуализации элементов матричной весовой функции

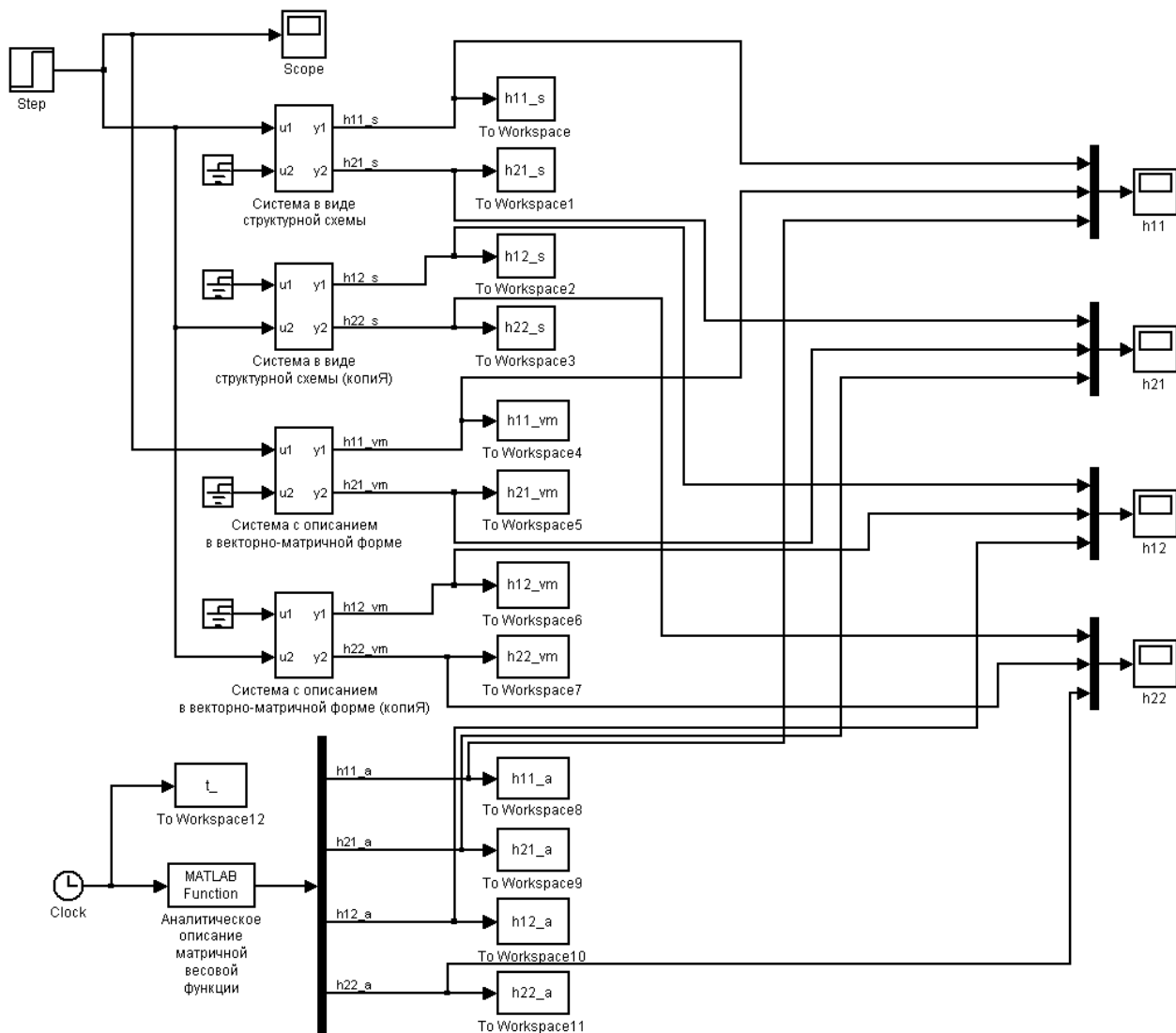


Рис. 1.3. Схема моделирования, используемая для расчета и визуализации элементов матричной переходной функции

Параметры моделирования следует задать в окне Simulation Parameters, доступном через меню Simulation\Simulation parameters окна, в котором открыт mdl-файл (рис. 1.4).

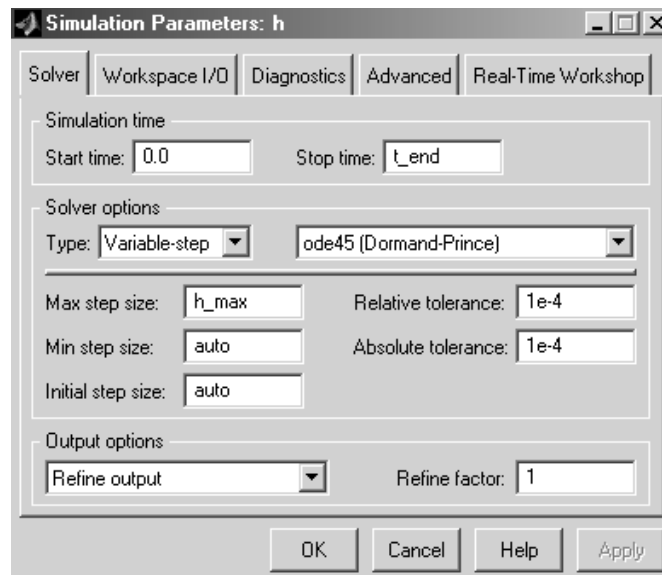


Рис. 1.4. Настройка параметров моделирования

Промоделировать систему, описанную в векторно-матричной форме, позволяет блок State-Space раздела Continuous основной библиотеки блоков Simulink. Структурная схема системы также может быть создана на основе блоков этой же библиотеки. Начальные условия следует принять нулевыми. Полученные структуры рекомендуется объединить в подсистемы, выделив все их элементы и выполнив команду Create Subsystem меню Edit или нажав на сочетание клавиш Ctrl+G.

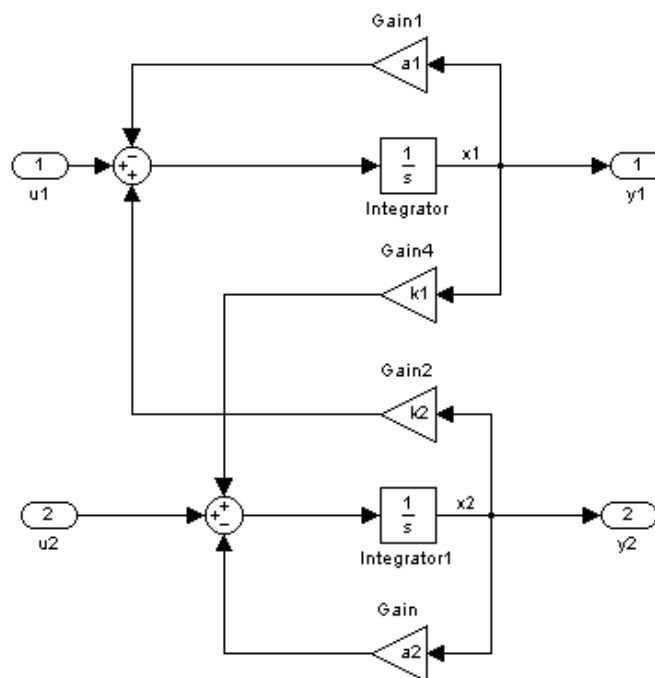


Рис. 1.5. Подсистема «Система в виде структурной схемы»

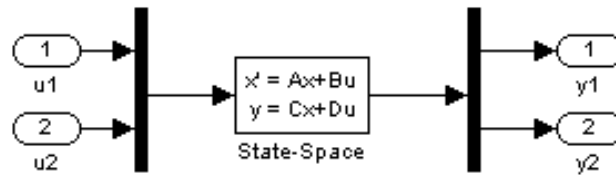
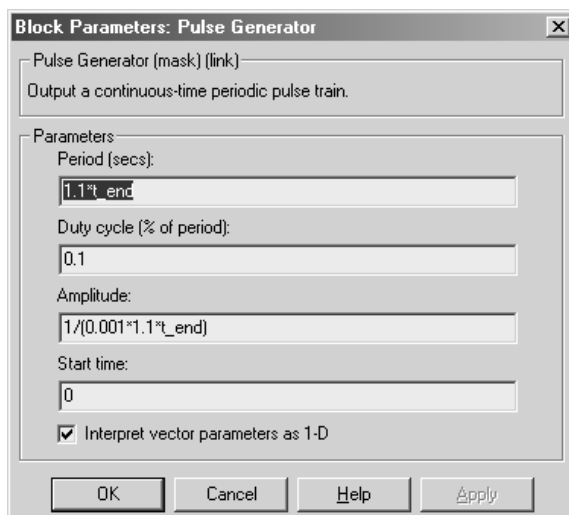
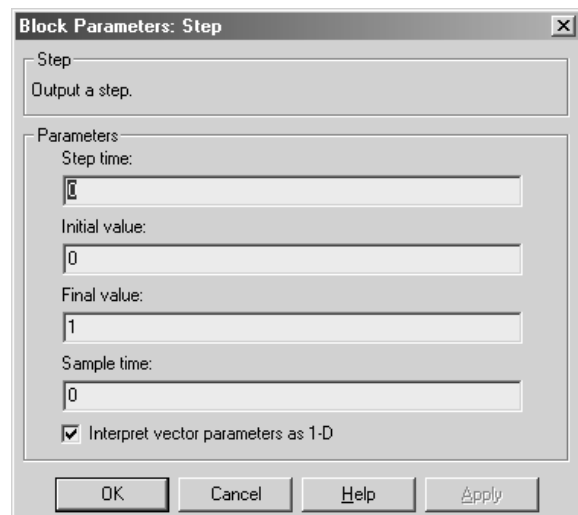


Рис. 1.6. Подсистема «Система с описанием в векторно-матричной форме»

При организации источника входного воздействия (единичной или дельта-функции) следует использовать блоки Step и Pulse Generator раздела Sources. Обратите внимание на то, что необходимо подобрать длительность входного импульса, который бы воспринимался системой как дельта-функция (**площадь импульса должна быть равна 1**), т.е. значение длительности должно быть таким, чтобы его уменьшение уже не приводило к изменению отклика системы.



а



б

Рис. 1.7. Настройки блоков:

а – Pulse Generator; б – Step

Графики найденных аналитически временных зависимостей можно получить при помощи блока MATLAB Fcn раздела Function&Tables и источника времени моделирования Clock.

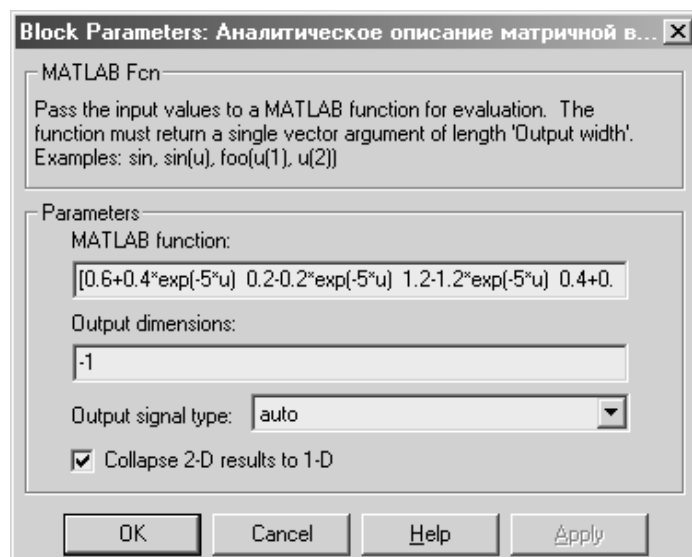


Рис. 1.8. Настройки блока Matlab Fcn, содержащего аналитическое описание элементов матричной весовой функции

Для визуализации процессов, соответствующих элементам матричной весовой и переходной функций, рекомендуется выводить их, а также текущее время моделирования в рабочую область памяти Matlab при помощи блоков To Workspace раздела Sinks, а затем строить их графики, используя команды построения графиков функций одной переменной. В этом случае не возникает проблем с переносом полученных результатов в электронную версию практических занятий и имеется возможность соблюсти все необходимые правила оформления графиков.

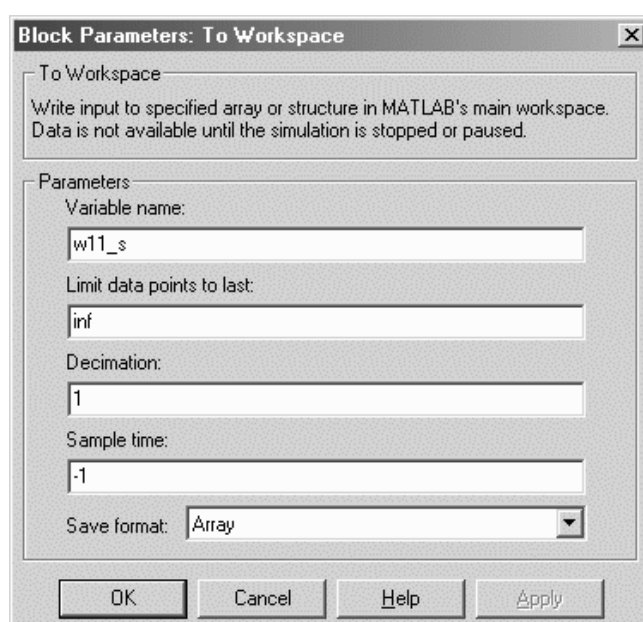


Рис. 1.9. Настройки блока To Workspace

Вывод графиков может быть продублирован с использованием осциллографов (блоков Scope раздела Sinks).

Четвертый и пятый файлы-сценарии `w_stop.m` и `h_stop.m` предназначены для построения графиков процессов, соответствующих элементам матричной весовой и переходной функций, в стандартных графических окнах Figure.

```
%Построение графиков элементов матричной весовой функции (файл w_stop.m)
```

```
%
```

```
close all
```

```
figure
```

```
plot(t_,w11_s,'r-',t_,w11_vm,'b--',t_,w11_a,'m-.')
```

```
grid on
```

```
xlabel('t, c')
```

```
ylabel('w11')
```

```
title('Графики элемента w11 матричной весовой функции')
```

```
legend('w11-struct','w11-VM','w11-analit',0)
```

```
figure
```

```
plot(t_,w21_s,'r-',t_,w21_vm,'b--',t_,w21_a,'m-.')
```

```
grid on
```

```
xlabel('t, c')
```

```
ylabel('w21')
```

```
title('Графики элемента w21 матричной весовой функции')
```

```
legend('w21-struct','w21-VM','w21-analit',0)
```

```
figure
```

```
plot(t_,w12_s,'r-',t_,w12_vm,'b--',t_,w12_a,'m-.')
```

```
grid on
```

```
xlabel('t, c')
```

```
ylabel('w12')
```

```
title('Графики элемента w12 матричной весовой функции')
```

```
legend('w12-struct','w12-VM','w12-analit',0)
```

```
figure
```

```
plot(t_,w22_s,'r-',t_,w22_vm,'b--',t_,w22_a,'m-.')
```

```
grid on
```

```
xlabel('t, c')
```

```

ylabel('w22')
title('Графики элемента w22 матричной весовой функции')
legend('w22-struct','w22-VM','w22-analit',0)

%Построение графиков элементов матричной переходной функции (файл h_stop.m)
%
close all

figure
plot(t_,h11_s,'r-',t_,h11_vm,'b--',t_,h11_a,'m-.')
grid on
xlabel('t, c')
ylabel('h11')
title('Графики элемента h11 матричной переходной функции')
legend('h11-struct','h11-VM','h11-analit',0)

figure
plot(t_,h21_s,'r-',t_,h21_vm,'b--',t_,h21_a,'m-.')
grid on
xlabel('t, c')
ylabel('h21')
title('Графики элемента h21 матричной переходной функции')
legend('h21-struct','h21-VM','h21-analit',0)

figure
plot(t_,h12_s,'r-',t_,h12_vm,'b--',t_,h12_a,'m-.')
grid on
xlabel('t, c')
ylabel('h12')
title('Графики элемента h12 матричной переходной функции')
legend('h12-struct','h12-VM','h12-analit',0)

figure
plot(t_,h22_s,'r-',t_,h22_vm,'b--',t_,h22_a,'m-.')
grid on
xlabel('t, c')
ylabel('h22')
title('Графики элемента h22 матричной переходной функции')
legend('h22-struct','h22-VM','h22-analit',0)

```

Назначение всех функций пакета Matlab, использованных при создании программ (m-файлов), приводится в приложении.

Организовать все указанные файлы следует самостоятельно (файлы-примеры выдаваться не будут) с целью получения навыков работы с редактором-отладчиком m-файлов и редактором схем Simulink.

1.3.2. Открыть и запустить файл `w_h_init.m` для задания значений всем необходимым переменным в рабочей области памяти Matlab (Workspace).

1.3.3. Открыть и запустить модель из файла `w.mdl` и проверить соответствие результатов расчета элементов матричной весовой функции результатам компьютерного моделирования. В случае несоответствия найти и устранить ошибки.

1.3.4. По завершении моделирования в Simulink открыть и запустить файл `w_stop.m`. Скопировать информацию, выведенную в графическое окно путем выполнения команды меню "Edit\Copy Figure", после чего сохранить ее при помощи какого-либо приложения, например текстового редактора MS Word.

1.3.5. Выполнить пп. 1.3.3 и 1.3.4, используя файлы `h.mdl` и `h_stop.m`.

Замечание. Запуск файлов `w_h_init.m` и `w_stop.m` (`h_stop.m`) до и после проведения моделирования можно автоматизировать, связав mdl-файл с соответствующими m-файлами. Для этого необходимо:

- убедиться в том, что mdl- и m- файлы находятся в одной директории и она является текущей;

- открыть mdl-файлы;

- в командном окне выполнить следующие команды:

```
set_param('w','InitFcn','w_h_init'),  
set_param('w','StopFcn','w_stop'),  
set_param('h','InitFcn','w_h_init'),  
set_param('h','StopFcn','h_stop');
```

- в Simulink для каждого mdl-файла выполнить команду меню Edit\Update Diagram и сохранить файлы.

1.4. Оформление материалов

1.4.1. Исходные данные занятия: название занятия, цель занятия, структурная схема исследуемой системы, номер варианта и соответствующие ему значения параметров схемы.

1.4.2. Используемые схемы моделирования с необходимыми пояснениями (например, по выбору параметров того или иного блока).

1.4.3. Графики элементов $w(t)$ и $h(t)$, количество – 8 графиков.

Материалы оформляются в электронном виде.

1.5. Контрольные вопросы

1. Дать определение переходной матрицы. Привести алгоритм вычисления.
2. Дать определение матричных весовой и переходной функций. Привести алгоритм вычисления.

2. ИЗУЧЕНИЕ ТИПОВЫХ ЗВЕНЬЕВ

2.1. Цель занятия

Целью занятия является расчет весовых и переходных функций, логарифмических амплитудно-частотных и логарифмических фазо-частотных характеристик (ЛАЧХ и ЛФЧХ), амплитудно-фазовых характеристик (АФХ) типовых динамических звеньев систем автоматического управления, а также анализ влияния их параметров на перечисленные выше характеристики. На занятии предусматривается исследование звеньев, перечень которых приведен в табл. 2.1.

Таблица 2.1

Перечень типовых звеньев

Номер звена	Звено	Передаточная функция
1	Интегрирующее	$\frac{k}{p}$
2	Инерционное	$\frac{k}{T_p p + 1}$
3	Реальное дифференцирующее	$\frac{k p}{T_p p + 1}$
4	Интегро-дифференцирующее	$\frac{T_1 p + 1}{T_2 p + 1}$
5	Колебательное	$\frac{k}{T^2 p^2 + 2\xi T_p p + 1}$
6	Неминимально-фазовое первого порядка	$\frac{1 - T_1 p}{1 + T_2 p}$
7	Неустойчивое первого порядка	$\frac{k}{T_p p - 1}$
8	Неустойчивое второго порядка	$\frac{k}{T^2 p^2 - 2\xi T_p p + 1}$
9	Режекторный фильтр	$\frac{T^2 p^2 + 2\xi_1 T_p p + 1}{T^2 p^2 + 2\xi_2 T_p p + 1}$

Обязательными для изучения являются звенья 2 и 5, указанные в табл. 2.1, а также одно дополнительное по указанию преподавателя.

2.2. Подготовительная часть занятия

Для каждого типового звена необходимо выполнить следующее.

2.2.1. Вывести и записать дифференциальные уравнения.

2.2.2. Вывести и записать выражения для модуля и фазы комплексных передаточных коэффициентов.

2.2.3. Вывести и записать выражения для весовых и переходных функций и привести их примерное графическое изображение.

2.2.4. Построить (эскизно) логарифмические асимптотические амплитудно-частотные характеристики и фазовые частотные характеристики. Обратить внимание на масштабирование по оси абсцисс.

2.2.5. Привести примерное изображение амплитудно-фазовых характеристик.

2.2.6. Спланировать проведение экспериментов.

Все рассматриваемые звенья, за исключением интегратора, имеют 2 или 3 настраиваемых параметра. В отдельных случаях при изучении влияния параметра на характеристики звена достаточно варьировать его независимо от остальных, а в других ситуациях имеет смысл учитывать соотношения нескольких параметров и варьировать их совместно. Поэтому для каждого исследуемого звена следует провести предварительный анализ необходимого количества и содержания экспериментов.

При проведении подготовки рекомендуется использовать учебную и справочную литературу.

Особое внимание следует уделить информационному сопровождению приводимых в подготовительной части занятия эскизных рисунков: для каждого процесса необходимо отметить его характерные особенности, которые зависят от каких-либо параметров исследуемого звена. Так, для временных процессов – это начальное и установившееся значения, постоянная времени, период колебаний, перерегулирование. Для ЛАЧХ указываются сопрягающие частоты, наклон в децибелах на декаду (дБ/дек) для наклонных участков и значения модуля комплексного передаточного коэффициента для горизонтальных участков. Для ЛФЧХ – сопрягающие частоты, асимптоты, точки экстремума (если есть). Для АФХ – точки на комплексной плоскости ($\operatorname{Re}(W(j\omega))$, $\operatorname{Im}(W(j\omega))$, ω), соответствующие $\omega = 0$, $\omega = \infty$, пересечению с мнимой или действительной осями, точка максимального значения модуля комплексного передаточного коэффициента.

2.3. Компьютерное моделирование

2.3.1. Описать передаточные функции изучаемых звеньев в файле-функции Matlab. Например, создать файл TF_zv.m следующего содержания:

```
%Функция описания звена в виде передаточной функции
%
function W = TF_zv(N_zv,inp_param)
    p = tf('p');
    switch N_zv
        case 1, %интегрирующее звено, N_zv = 1, inp_param = [k]
            k = inp_param(1);
            W = k/p;
        case 2, %инерционное звено, N_zv = 2, inp_param = [k,T]
            k = inp_param(1);
            T = inp_param(2);
            W = k/(T*p+1);
        case 3, %реальное дифференцирующее звено, N_zv = 3, inp_param = [k,T]
            k = inp_param(1);
            T = inp_param(2);
            W = (k*p)/(T*p+1);
        case 4, %интегро-дифференцирующее звено, N_zv = 4, inp_param = [T1,T2]
            T1 = inp_param(1);
            T2 = inp_param(2);
            W = (T1*p+1)/(T2*p+1);
        case 5, %колебательное звено, N_zv = 5, inp_param = [k,T,ksi]
            k = inp_param(1);
            T = inp_param(2);
            ksi = inp_param(3);
            W = k/(T^2*p^2+2*ksi*T*p+1);
        case 6, %неминимально-фазовое звено 1-го пор., N_zv = 6, inp_param = [T1,T2]
            T1 = inp_param(1);
            T2 = inp_param(2);
            W = (1-T1*p)/(1+T2*p);
        case 7, %неустойчивое звено 1-го пор., N_zv = 7, inp_param = [k,T]
            k = inp_param(1);
            T = inp_param(2);
            W = k/(T*p-1);
        case 8, %неустойчивое звено 2-го пор., N_zv = 8, inp_param = [k,T,ksi]
            k = inp_param(1);
```

```

T = inp_param(2);
ksi = inp_param(3);
W = k / (T^2*p^2 - 2*ksi*T*p + 1);
case 9, %режекторный фильтр, N_zv = 9, inp_param = [T,ksi1,ksi2]
T = inp_param(1);
ksi1 = inp_param(2);
ksi2 = inp_param(3);
W = (T^2*p^2 + 2*ksi1*T*p + 1) / (T^2*p^2 + 2*ksi2*T*p + 1);
end
%end of function TF_zv

```

Назначение всех использованных при создании этого файла функций Matlab приводится в приложении.

В программе звенья пронумерованы в соответствии с табл. 2.1, указание на номер моделируемого звена выполняется с помощью параметра `N_zv` рассматриваемой функции `TF_zv`. Второй параметр `inp_param` является векторным и предназначен для передачи значений параметров звена; в зависимости от вида звена он может содержать от одного до трех элементов. Для организации выбора описания звена служит оператор `switch...case`, при помощи которого производится анализ значения переменной `N_zv` и переход на вычисление соответствующей передаточной функции.

2.3.2. Для каждого звена написать файл-сценарий с целью получения графиков требуемых временных и частотных характеристик. Например, для изучения колебательного звена следует создать файл `kol_zv.m` следующего содержания:

```

%Изучение типовых динамических звеньев: колебательное звено

%Очистка всех переменных в памяти
clear all

%Очистка командного окна
clc

%Закрытие всех предыдущих рисунков

```

```

set(0,'ShowHiddenHandles','on')
delete(get(0,'Children'))

%Описание колебательного звена (N_zv = 5) через его передаточную функцию
%при различных значениях параметров. Параметры колебательного звена
%задаются вектором inp_param = [k,T,ksi] (см. текст файла TF_zv.m)

%изменяем k
W_11 = TF_zv(5,[1,1,0.707]);
W_12 = TF_zv(5,[2,1,0.707]);
W_13 = TF_zv(5,[3,1,0.707]);

%изменяем T
W_21 = TF_zv(5,[1,0.354,0.707]);
W_22 = TF_zv(5,[1,0.707,0.707]);
W_23 = TF_zv(5,[1,1.414,0.707]);

%изменяем ksi
W_31 = TF_zv(5,[1,0.5,0.3]);
W_32 = TF_zv(5,[1,0.5,0.5]);
W_33 = TF_zv(5,[1,0.5,0.7]);

%изменяем T и ksi
W_41 = TF_zv(5,[1,0.3,0.3]);
W_42 = TF_zv(5,[1,0.5,0.5]);
W_43 = TF_zv(5,[1,0.7,0.7]);

%Построение требуемых характеристик при различных k
%ЛАЧХ и ЛФЧХ
ltiview({'bode'},W_11,'b-',W_12,'r-',W_13,'k-')
%АФХ
ltiview({'nyquist'},W_11,'b-',W_12,'r-',W_13,'k-')
%весовая функция w(t)
ltiview({'impulse'},W_11,'b-',W_12,'r-',W_13,'k-')
%переходная функция h(t)
ltiview({'step'},W_11,'b-',W_12,'r-',W_13,'k-')

%Построение требуемых характеристик при различных T
%ЛАЧХ и ЛФЧХ

```

```

ltiview({'bode'},W_21,'b-',W_22,'r-',W_23,'k-')
%АФХ
ltiview({'nyquist'},W_21,'b-',W_22,'r-',W_23,'k-')
%весовая функция w(t)
ltiview({'impulse'},W_21,'b-',W_22,'r-',W_23,'k-')
%переходная функция h(t)
ltiview({'step'},W_21,'b-',W_22,'r-',W_23,'k-')

%Построение требуемых характеристик при различных ksi
%ЛАЧХ и ЛФЧХ
ltiview({'bode'},W_31,'b-',W_32,'r-',W_33,'k-')
%АФХ
ltiview({'nyquist'},W_31,'b-',W_32,'r-',W_33,'k-')
%весовая функция w(t)
ltiview({'impulse'},W_31,'b-',W_32,'r-',W_33,'k-')
%переходная функция h(t)
ltiview({'step'},W_31,'b-',W_32,'r-',W_33,'k-')

%Построение требуемых характеристик при различных T и ksi
%ЛАЧХ и ЛФЧХ
ltiview({'bode'},W_41,'b-',W_42,'r-',W_43,'k-')
%АФХ
ltiview({'nyquist'},W_41,'b-',W_42,'r-',W_43,'k-')
%весовая функция w(t)
ltiview({'impulse'},W_41,'b-',W_42,'r-',W_43,'k-')
%переходная функция h(t)
ltiview({'step'},W_41,'b-',W_42,'r-',W_43,'k-')

```

Назначение всех использованных при создании этой программы функций Matlab приводится в приложении.

Обратите внимание: в данном m-файле используется внешняя функция TF_zv, описанная в п. 2.3.1, в связи с этим файлы kol_zv.m и TF_zv.m должны находиться в одной директории.

Для построения временных и частотных характеристик динамического звена используется команда `ltiview`, первый параметр которой – строковая переменная, заключенная в фигурные скобки, – служит для указания типа отоб-

ражаемой характеристики, а следующие за ним пары параметров – для указания имени системы и свойств выводимой линии (цвета, типа линии и т. д. аналогично оформлению двумерных графиков при использовании команды plot).

2.3.3. Для каждого изучаемого звена произвести запуск программы и зафиксировать результаты моделирования.

При выполнении команды `ltiview` появляется графическое окно, в котором отображается график указанной характеристики. Дополнительные настройки доступны при нажатии правой клавиши мыши и через команды меню. Необходимой настройкой является включение сетки `grid`.

Для сохранения содержимого графического окна LTI Viewer необходимо воспользоваться командой меню `File\Print to Figure`, после чего открывается новое окно `Figure`, содержащее тот же самый рисунок, скопировать который можно путем выполнения команды меню `Edit\Copy Figure`.

При изучении влияния какого-либо параметра на характеристики рассматриваемого звена он (отдельно или в сочетании с другими параметрами) должен варьироваться не менее трех раз. Это позволяет впоследствии достаточно легко выявить основную тенденцию в изменении вида характеристик. Таким образом, результатом эксперимента с варьированием какого-либо параметра звена является четыре рисунка (ЛАЧХ и ЛФЧХ, АФХ, весовые функции, переходные функции), содержащие не менее трех графиков.

Замечания.

1. При оформлении результатов компьютерного моделирования необходимо, как и в подготовительной части, обратить внимание на информационное сопровождение рисунков: оси должны быть снабжены обозначениями, рисунки иметь подрисуночные надписи, каждому графику должен быть поставлен в соответствие тот набор параметров, при котором проводилось моделирование звена. Кроме того, на графиках должны быть отмечены характерные точки (с указанием числовых значений по осям) и показано их соответствие теоретически рассчитанным характерным точкам.

2. На АФХ проставить оцифровку (не менее 8 значений ω).

2.4. Оформление материалов

2.4.1. Исходные данные практического занятия: название занятия, цель занятия, список выбранных для изучения звеньев с их передаточными функциями.

2.4.2. Подготовительная часть, оформленная в соответствии с требованиями п. 2.2.

2.4.3. Результаты компьютерного моделирования, оформленные в соответствии с требованиями п. 2.3.3.

2.4.4. Качественный анализ влияния параметров каждого из рассмотренных звеньев на вид частотных и временных характеристик.

Материалы оформляются в электронном виде.

2.5. Контрольные вопросы

1. Записать выражения для модуля и фазы комплексного передаточного коэффициента указанных преподавателем динамических звеньев.

2. Перечислить операторы Matlab для описания динамических звеньев.

3. Провести анализ влияния вариации параметров динамических звеньев на частотные характеристики (вид динамического звена указывается преподавателем).

4. Установить связь между параметрами колебательного звена и:

- расположением полюсов передаточной функции;
- временем переходного процесса;
- перерегулированием.

5. Изобразить оси для построения ЛЧХ с оцифровкой (в конкретном масштабе).

6. Перевести значения модулей $W(j\omega)$, указанных преподавателем, из натуральных чисел в децибелы и обратно.

3. ЧАСТОТНЫЕ ХАРАКТЕРИСТИКИ СИСТЕМ И КРИТЕРИЙ УСТОЙЧИВОСТИ НАЙКВИСТА

3.1. Цель занятия

Целью занятия является анализ частотных характеристик разомкнутых и замкнутых систем, получение навыков по использованию критерия устойчивости Найквиста.

В занятии предусматривается исследование двух систем, различающихся видом передаточной функции (ПФ) разомкнутого контура. Варианты значений параметров ПФ приведены в табл. 3.1. Замкнутая система построена по типу классической следящей системы, ее структурная схема представлена на рис. 3.1.

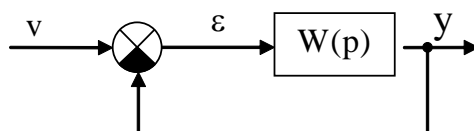


Рис. 3.1. Структурная схема исследуемой замкнутой системы

Варианты передаточной функции разомкнутого контура:

$$\text{система № 1 } W(p) = \frac{T_1 p + 1}{T_2 p + 1} \cdot \frac{1}{T_3 p - 1} \cdot \frac{k}{p},$$

$$\text{система № 2 } W(p) = \left(\frac{T_1 p + 1}{T_2 p + 1} \right)^2 \cdot \frac{k}{p^3}.$$

Таблица 3.1

Значения параметров передаточной функции

Номер варианта		1	2	3	4	5	6	7	8	9	10
Система № 1	T ₁	0,5	0,5	1,0	1,0	2,0	2,0	3,0	3,0	5,0	5,0
	T ₂	0,1	0,1	0,2	0,2	0,4	0,4	0,6	0,6	1,0	1,0
	T ₃	0,5	1,0	1,0	2,0	2,0	4,0	3,0	6,0	5,0	10,0
Система № 2	T ₁	1	1	2	2	4	4	6	6	8	8
	T ₂	0,02	0,1	0,04	0,2	0,08	0,4	0,12	0,6	0,16	0,8
Номер варианта		11	12	13	14	15	16	17	18	19	20
Система № 1	T ₁	0,25	0,25	0,5	0,5	1,0	1,0	2,0	2,0	4,0	4,0
	T ₂	0,1	0,1	0,2	0,2	0,4	0,4	0,8	0,8	1,6	1,0
	T ₃	2,0	3,0	4,0	6,0	8,0	12,0	16	24	32	48
Система № 2	T ₁	10	10	20	20	40	40	60	60	80	80
	T ₂	0,2	1	0,4	2	0,8	4	1,2	6	1,6	8

3.2. Подготовительная часть

Для каждой из двух систем в соответствии с заданным вариантом произвести следующее:

3.2.1. Записать уравнения для модуля и фазы комплексного передаточного коэффициента.

3.2.2. Построить (эскизно) логарифмическую асимптотическую амплитудную и фазовую частотные характеристики разомкнутой системы.

3.2.3. Построить (эскизно) амплитудно-фазовую характеристику разомкнутой системы, провести ее расширение, если это необходимо.

3.2.4. Проанализировать устойчивость замкнутой системы, применяя критерий устойчивости Найквиста. По критерию Гурвица найти критическое значение передаточного коэффициента k разомкнутого контура системы, а также его значения (в виде неравенств), приводящие замкнутую систему в устойчивое или неустойчивое состояние.

3.3. Компьютерное моделирование

3.3.1. Создать файл-сценарий, содержащий описание исследуемых систем и обеспечивающий построение частотных характеристик и переходных про-

цессов. Также необходимо предусмотреть расчет значений полюсов для каждой из систем в замкнутом состоянии. Рекомендуется организовать файл следующим образом:

```
%Исследование устойчивости систем (частотные характеристики систем  
%и критерий устойчивости Найквиста), файл prog.m
```

```
%Очистка всех переменных в памяти  
clear all
```

```
%Очистка командного окна  
clc
```

```
%Закрытие всех предыдущих рисунков  
set(0,'ShowHiddenHandles','on')  
delete(get(0,'Children'))
```

```
%Параметры систем для варианта #20
```

```
T1_s1 = 4;  
T2_s1 = 1;  
T3_s1 = 48;  
K_s1 = 1;
```

```
T1_s2 = 80;  
T2_s2 = 8;  
K_s2 = 1;
```

```
%Описание разомкнутой и замкнутой систем через передаточные функции  
p = tf('p');
```

```
R_s1 = K_s1*(T1_s1*p+1);  
Q_s1 = (T2_s1*p+1)*(T3_s1*p-1)*p;  
W_s1_r = R_s1/Q_s1;  
W_s1_z = R_s1/(Q_s1+R_s1);
```

```
R_s2 = K_s2*(T1_s2*p+1)^2;  
Q_s2 = (T2_s2*p+1)^2*p^3;  
W_s2_r = R_s2/Q_s2;
```

```

W_s2_z = R_s2/(Q_s2+R_s2);

%Выбор номера исследуемой системы
sys_num = 1;

%Построение требуемых характеристик
switch sys_num
    case 1,
        %ЛАЧХ и ЛФЧХ разомкнутой системы
        ltiview({'bode'},W_s1_r);
        %АФХ разомкнутой системы
        ltiview({'nyquist'},W_s1_r);
        %переходный процесс в замкнутой системе
        ltiview({'step'},W_s1_z);
        %собственные числа замкнутой системы - полюсы ее ПФ
        [zeros_s1_z,poles_s1_z,koef_s1_z] = zpndata(zpk(W_s1_z),'v');
        poles_s1_z
    case 2,
        %ЛАЧХ и ЛФЧХ разомкнутой системы
        ltiview({'bode'},W_s2_r);
        %АФХ разомкнутой системы
        ltiview({'nyquist'},W_s2_r);
        %переходный процесс в замкнутой системе
        ltiview({'step'},W_s2_z);
        %собственные числа замкнутой системы - полюсы ее ПФ
        [zeros_s2_z,poles_s2_z,koef_s2_z] = zpndata(zpk(W_s2_z),'v');
        poles_s2_z
end

```

Назначение всех использованных функций Matlab приводится в приложении.

Переменной `sys_num` необходимо присвоить значение номера той системы, изучение которой производится в данный момент. В результате для этой системы при выбранных значениях ее параметров будут построены все требуемые характеристики с использованием LTI Viewer, а в командное окно будут выведены значения полюсов ПФ замкнутой системы.

3.3.2. Запустить созданный файл-сценарий, предварительно выбрав систему № 1 и задав значения ее параметров в соответствии со своим вариантом (значение коэффициента усиления k разомкнутой системы принять равным 1). Сравнить полученные ЛАЧХ, ЛФЧХ и АФХ разомкнутой системы с результатами подготовки. По совмещенным ЛАЧХ и ЛФЧХ разомкнутой системы вычислить значение k , при котором:

- замкнутая система будет находиться на границе устойчивости ($k=k_{кр}$);
- замкнутая система будет неустойчива ($k=k_{нy}$);
- замкнутая система будет устойчива ($k=k_y$) с запасом по модулю не менее 10 дБ.

Проверить соответствие значений коэффициента, рассчитанных при подготовке и по результатам компьютерного моделирования.

3.3.3. Для каждого из трех найденных значений k провести повторный запуск программы и убедиться в том, что замкнутая система находится в соответствующем значению k состоянии. Для этого:

- проверить значения запасов устойчивости по модулю, используя ЛЧХ разомкнутой системы, оценить состояние замкнутой системы;
- проанализировать взаимное расположение АФХ разомкнутой системы и точки $(-1, j0)$ на комплексной плоскости, определить количество переходов расширенной АФХ разомкнутой системы, сделать вывод об устойчивости системы в замкнутом состоянии;
- проанализировать расположение полюсов замкнутой системы на комплексной плоскости, сделать вывод о ее устойчивости;
- проверить состояние замкнутой системы по ее переходной характеристике.

Эксперимент по изучению устойчивости рассматриваемой системы считается завершенным только в случае непротиворечивости всех полученных результатов.

3.3.4. Сохранить в файл совмещенные ЛАЧХ и ЛФЧХ разомкнутой системы при $k=1$, также сохранить все частотные характеристики разомкнутой

системы, переходный процесс в замкнутой системе и ее собственные значения при $k = k_{кр}, k_{ну}, k_y$ (т. е. всего $1+3 \times 3$ рисунка + 3 текстовых блока для рассматриваемой системы). Наличие сетки на графиках обязательно. На ЛАЧХ должны быть отмечены сопрягающие частоты и ее наклон на каждом из участков. На АФХ должна присутствовать оцифровка.

3.3.5. Повторить пп. 3.3.2-3.3.4 для системы № 2.

3.4. Оформление материалов

3.4.1. Исходные данные практического занятия: название занятия, цель занятия, перечень ПФ исследуемых систем с указанием значений параметров, соответствующих номеру варианта.

3.4.2. Подготовительная часть, оформленная в соответствии с требованиями п. 3.2.

3.4.3. Результаты компьютерного моделирования, оформленные в соответствии с требованиями п. 3.3.4.

3.4.4. Анализ полученных результатов – на основании выводов, сделанных в пп.3.3.2 и 3.3.3.

Отчет оформляется в электронном виде.

3.5. Контрольные вопросы

1. Формулировка необходимого условия устойчивости систем.
2. Формулировка критерия устойчивости Гурвица.
3. Правила расширения частотных характеристик.
4. Формулировка частотного критерия устойчивости (критерия Найквиста).
5. Понятие запасов устойчивости.
6. Решение задачи на нахождение коэффициента усиления разомкнутой системы для осуществления сдвига ЛАЧХ по оси ординат на заданное количество децибел.
7. Способы описания объектов в Matlab.

4. МОДАЛЬНЫЙ СИНТЕЗ УПРАВЛЕНИЯ В ЛИНЕЙНЫХ НЕПРЕРЫВНЫХ СИСТЕМАХ

4.1. Цель занятия

Данное занятие предназначено для первого практического ознакомления с применением модального метода для синтеза управления. Его целью является исследование взаимосвязи между расположением собственных чисел и качеством процессов и выработка обоснованного подхода к выбору желаемых полюсов системы при осуществлении модального синтеза.

В качестве неизменяемой части системы следует принять систему № 1, заданную передаточной функцией в практическом занятии № 3, с теми же значениями параметров.

4.2. Подготовительная часть

4.2.1. По передаточной функции разомкнутой системы (объекта) записать соответствующие векторно-матричные уравнения.

4.2.2. Записать в общем виде уравнение обратной связи, т.е. выражение для синтезируемого управляющего сигнала (входного сигнала объекта) в векторно-матричной форме. Дать определение и вычислить размерности всех входящих в уравнение переменных.

4.2.3. Записать векторно-матричные уравнения замкнутой системы.

4.2.4. По результатам выполнения пп. 4.2.1 и 4.2.2 составить подробную структурную схему замкнутой системы (т.е. системы «регулятор + объект управления») с показом структуры объекта и обратных связей по каждому элементу вектора состояния.

4.2.5. Дать описание алгоритма вычисления матрицы обратной связи.

4.2.6. Привести описание алгоритма вычисления коэффициента усиления по командному сигналу.

4.2.7. Изучить материалы лекций по темам «Основные показатели качества», «Обратная связь по состоянию, обеспечивающая заданное (желаемое) расположение собственных чисел в замкнутой системе с одним (скалярным)

входом», а также проанализировать данные, полученные в процессе выполнения практических занятий № 2 и 3, где требовалось проводить анализ влияния собственных чисел систем на их динамические свойства.

4.3. Компьютерное моделирование

4.3.1. Подготовить файл-сценарий, реализующий процедуру модального синтеза управления для изучаемой системы. Реализовать в Simulink структурную схему замкнутой системы, предусмотрев возможность наблюдения сигнала на входе и выходе объекта. Пример модели для системы № 1 (sys1_mod.mdl) представлен на рис. 4.1 и 4.2, а содержание m-файла (sys1.m) будет следующим:

```
%Модальный синтез управления в линейных непрерывных системах
%Система #1
%Вывод информации - в командное и графические окна

%Очистка всех переменных в памяти
clear all

%Очистка командного окна
clc

%Закрытие всех предыдущих рисунков
set(0, 'ShowHiddenHandles', 'on')
delete(get(0, 'Children'))

%Установка параметров моделирования для mdl-файла
h_max = 0.01; %максимальный шаг моделирования
t_end = 20;   %время завершения моделирования

%задание переменной преобразования Лапласа для последующего описания
%системы в виде передаточной функции
p = tf('p');

%Параметры системы #1 для варианта #20
T1_s1 = 4;
T2_s1 = 1;
T3_s1 = 48;
```

```

%Описание разомкнутой системы через передаточную функцию
disp('Передаточная функция системы #1 в разомкнутом состоянии');
disp(['          ' num2str(T1_s1) '*p+1'])
disp('-----');
disp(['(' num2str(T2_s1) '*p+1)*(' num2str(T3_s1) '*p-1)*p']);
R_s1 = T1_s1*p+1;
Q_s1 = (T2_s1*p+1)*(T3_s1*p-1)*p;
W_s1_r = R_s1/Q_s1

%Преобразование описания в ss-форму (получение описания через матрицы
%A,B,C,D)
sys1_r = ss(W_s1_r);
[A_s1_r,B_s1_r,C_s1_r,D_s1_r] = ssdata(sys1_r);
disp('Матрицы A,B,C,D разомкнутой системы');
A_s1_r
B_s1_r
C_s1_r
D_s1_r

%Вычисление нулей и полюсов ПФ разомкнутой системы
[zeros_s1_r,poles_s1_r,kcoef_s1_r] = zpkmdata(zpk(W_s1_r),'v');
disp('Нули и полюсы ПФ разомкнутой системы');
zeros_s1_r
poles_s1_r

%Проверка правильности вычисления матрицы динамики при домашней подготовке
%путем вычисления ее собственных чисел и сравнения их с poles_s1_r
disp('Собств. числа матрицы динамики разомк. сис-мы (из домашней подготовки)');
poles_s1_r_dp = eig([-47/48 1/48 0; 1 0 0; 0 1 0])

%Желаемые полюсы замкнутой системы (задаются после анализа расположения
%полюсов разомкнутой системы)
disp('Желаемые полюсы замкнутой системы');
poles_s1_z_g = [-2+2*j; -2-2*j; -0.25]

%Расчет матрицы обратных связей L, реализующей желаемое расположение
%полюсов замк. сис.
[L_s1,PREC_s1,MESSAGE_s1] = PLACE(A_s1_r,B_s1_r,poles_s1_z_g);
disp('Матрица обратных связей');
L_s1

```



```

%Матрица динамики замкнутой системы
A_sl_z = A_sl_r-B_sl_r*L_sl;

%Матрицы B,C,D замкнутой системы
B_sl_z = B_sl_r;
C_sl_z = C_sl_r;
D_sl_z = D_sl_r;

%Описание замкнутой системы в ss-форме
sysl_z = ss(A_sl_z,B_sl_z,C_sl_z,D_sl_z);

%Расчет коэффициента усиления по командному сигналу (должен обеспечивать
%единичную статику)
W_sl_z = tf(sysl_z);
[num_sl,den_sl] = tfdata(W_sl_z,'v');
disp('Коэффициент усиления по командному сигналу');
%Вычисление коэффициента усиления по командному сигналу
%как отношения свободных членов полиномов знаменателя и числителя,
%передаточной функции замкнутой системы, т.е. последних элементов
%в векторах коэффициентов знаменателя и числителя
Kv_sl = den_sl(length(den_sl))/num_sl(length(num_sl))

%Описание итоговой замкнутой системы в ss-форме
B_sl_z = Kv_sl*B_sl_r;
sysl_z = ss(A_sl_z,B_sl_z,C_sl_z,D_sl_z);

%Фактические собственные числа итоговой замкнутой системы
%(сравнить с заказанными)
[zeros_sl_z_f,poles_sl_z_f,koef_sl_z_f] = zpndata(zpk(sysl_z),'v');
disp('Нули и полюсы замкнутой системы с синтезированным управлением');
zeros_sl_z_f
poles_sl_z_f

%Рисунок, отображающий расположение нулей и полюсов до и после
%замыкания обратных связей
figure
plot(real(poles_sl_r),imag(poles_sl_r),'rs',real(zeros_sl_r),imag(zeros_sl_r),'ro')
hold on
plot(real(poles_sl_z_f),imag(poles_sl_z_f),'bs',real(zeros_sl_z_f),...

```

```

imag(zeros_sl_z_f),'bo','MarkerSize',14)
grid on
title('Расположение нулей и полюсов до и после замыкания обратных связей')
xlabel('real(z)');
ylabel('imag(z)');
legend('poles of sys1 (razomk)', 'zeros of sys1 (razomk)', ...
       'poles of sys1 (zamkn)' , 'zeros of sys1 (zamkn)', 0);

%переходный процесс в замкнутой системе с синтезированным управлением
ltiview({'step'},sys1_z,t_end);
grid on

%вызов модели
open_system('sys1_mod.mdl');
%запуск моделирования
sim('sys1_mod');

```

Назначение всех использованных при создании данной программы функций Matlab приводится в приложении.

Как следует из текста вышеприведенного файла, процедура модального синтеза управления предполагает выполнение следующих шагов:

- описание объекта (ввод исходных данных);
- преобразование описания объекта к ss-форме и нахождение матриц A, B, C, D;
- вычисление нулей и полюсов ПФ разомкнутой системы;
- анализ расположения нулей и полюсов разомкнутой системы и назначение желаемых полюсов замкнутой системы;
- расчет матрицы обратных связей, реализующей желаемое расположение полюсов замкнутой системы;
- вычисление передаточной функции полученной замкнутой системы и коэффициента усиления по командному сигналу $k^v = \frac{1}{W_{vy}(0)}$ (т. е. как отношения свободных членов полиномов знаменателя и числителя передаточной функции замкнутой системы).

Результаты вычислений на каждом из этапов выводятся в командное окно.

Также в программе предусмотрено проведение ряда проверок (с использованием данных из командного окна): сравнение собственных чисел матрицы динамики разомкнутой системы, полученной в ходе предварительной подготовки, с собственными числами разомкнутой системы, рассчитанными в Matlab; проверка правильности работы процедуры расчета матрицы обратных связей – вычисление фактических значений полюсов полученной замкнутой системы и сравнение их с желаемыми полюсами.

Кроме того, по завершении процедуры модального синтеза будет выведено графическое окно с расположением нулей и полюсов системы на комплексной плоскости до и после замыкания обратных связей, построен график переходного процесса для замкнутой системы и запущена модель Simulink, содержащая структурную схему замкнутой системы.

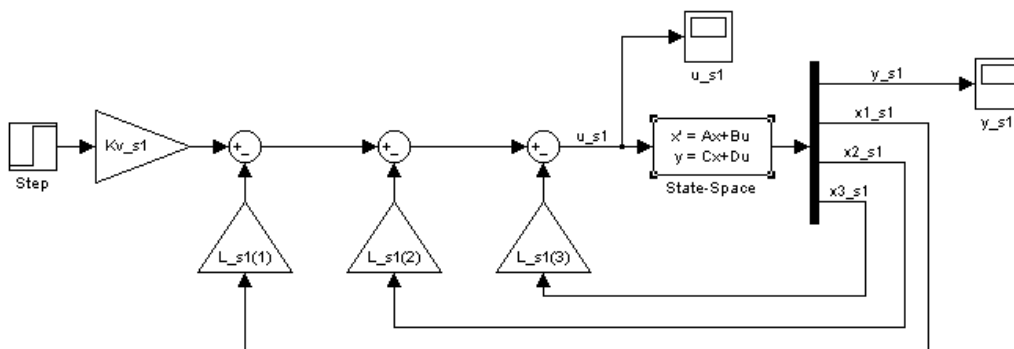
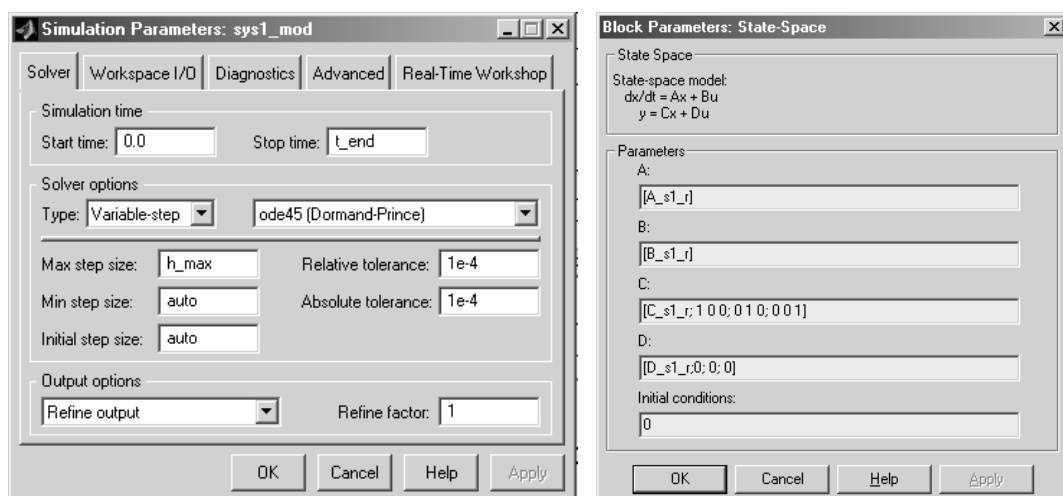


Рис. 4.1. Структурная схема замкнутой системы в Simulink



а)

б)

Рис. 4.2. Настройка параметров: а – моделирования; б – в блоке State-Space

4.3.2. В тексте описанного выше файла-сценария скорректировать исходные данные в соответствии с номером варианта, задать матрицу динамики, рассчитанную при домашней подготовке, и задать некоторое желаемое расположение полюсов замкнутой системы. Скорректировать описание объекта в блоке State-Space модели, если это необходимо. Запустить программу на выполнение, после чего проверить правильность предварительных расчетов и корректность работы встроенной в Matlab процедуры расчета матрицы обратных связей place. Убедиться в правильности описания замкнутой системы при помощи структурной схемы в Simulink, сравнив переходный процесс в графическом окне и на экране осциллографа. В случае отсутствия ошибок сохранить полученные файл-сценарий и структурную схему в файле.

4.3.3. Выявить взаимосвязь между расположением нулей и полюсов замкнутой системы на комплексной плоскости и качеством процессов управления, варьируя желаемые собственные значения и фиксируя при помощи встроенных средств LTI Viewer (описание LTI Viewer см. в приложении) значения времени регулирования t_r и перерегулирования σ (основных показателей качества переходного процесса). Величину допуска («трубки») принять равной $\pm 5\%$.

Выполнение этого пункта занятия может существенно облегчить следующий подход: стремиться располагать полюсы замкнутой системы таким образом, чтобы приблизить ее к системе 2-го порядка, так как обеспечить требуемое качество управления в такой системе значительно проще; принимать во внимание тот факт, что значения показателей качества зависят от расположения на комплексной плоскости полюсов, ближайших к мнимой оси.

Тогда, во-первых, необходимо обеспечить расположение всех полюсов слева от мнимой оси (система должна быть устойчива в замкнутом состоянии); во-вторых, ближайшей к мнимой оси должна быть пара комплексно сопряженных полюсов, не считая полюсов, компенсирующих нули системы; в-третьих, все остальные полюсы должны находиться значительно левее, чтобы не оказывать влияния на характер переходной функции (соответствующие им моды будут обладать меньшим временем переходного процесса и вносить меньший

вклад в выходной сигнал). Однако следует помнить, что чрезмерный сдвиг полюсов влево может привести к значительному увеличению коэффициентов обратных связей, росту пиковых значений управляющего воздействия и появлению нелинейных эффектов в реальных системах, поскольку для них всегда производится ограничение управляющего сигнала.

Изучение поведения систем второго порядка на примере колебательного звена показывает, что время переходного процесса определяется расстоянием пары комплексно сопряженных полюсов $p_{1,2} = -\alpha \pm j\beta$ до мнимой оси (это время равно приблизительно утроенной обратной величине их действительной части, $t_p \approx 3/|\alpha|$), а от соотношения $|\beta|/|\alpha|$ мнимой и действительной частей зависит колебательность процесса.

4.3.4. Рассчитать управление, обеспечивающее в замкнутой системе значительное улучшение показателей качества по сравнению с полученными в занятии № 3 для данной системы.

Привести на графике в одних координатах переходные функции системы из занятия № 3 и полученного на данном занятии при модальном синтезе, сравнить показатели качества.

4.4. Оформление материалов

4.4.1. Исходные данные практического занятия: название занятия, цель занятия, ПФ исследуемой системы с указанием значений параметров, соответствующих номеру варианта.

4.4.2. Подготовительная часть, оформленная в соответствии с требованиями п. 4.2.

4.4.3. Текст файла-сценария и структурная схема модели в Simulink.

4.4.4. Результаты компьютерного моделирования, оформленные в соответствии с требованиями пп. 4.3.3 и 4.3.4.

4.4.5. Аргументированные выводы о влиянии собственных значений на характер переходных процессов.

Материалы оформляются в электронном виде.

4.5. Контрольные вопросы

1. Уравнения системы до и после синтеза управления.
2. Приведите операторы Matlab для описания линейных систем.
3. Приведите операторы Matlab для преобразования описания систем.
4. Приведите операторы Matlab для извлечения информации о параметрах систем.
5. Какая функция обеспечивает расчет матрицы обратных связей?
6. Какими средствами обеспечивается единичная статика в замкнутой системе?
7. Чем определяется количество степеней свободы в модальном синтезе?

СПИСОК ЛИТЕРАТУРЫ

1. Ануфриев, И. Е. Самоучитель Matlab 5.3/6.x / И.Е. Ануфриев. – СПб. : БХВ-Петербург, 2002. – 736 с.
2. Гайдук, А.Р. Теория автоматического управления в примерах и задачах с решениями в Matlab : учеб. пособие для студентов вузов / А.Р. Гайдук. – СПб. : Лань, 2011. – 464 с.
3. Дьяконов, В. Matlab : учебный курс / В. Дьяконов. – СПб. : Питер, 2001. – 560 с.
4. Дьяконов, В. Simulink 4 : специальный справочник / В. Дьяконов. – СПб.: Питер, 2002. 528 с.
5. Ерофеев, А.А. Теория автоматического управления : учебник для студентов вузов / А.А. Ерофеев. – СПб. : Политехника, 2008. – 302 с.
6. Коновалов, Б.И. Теория автоматического управления / Б.И. Коновалов, Ю.М. Лебедев. – М. : Лань, 2010. – 218 с.
7. Малов, А.В. Пакет математического моделирования MATLAB v6.0 : краткое справочное руководство к лабораторным работам по дисциплине «Теория автоматического управления» / А.В. Малов, Е.Э. Страшинин. – Екатеринбург : УГТУ-УПИ, 2010. – 52 с.
8. Медведев, В.С. Control System Toolbox. MATLAB 5 для студентов / В.С. Медведев, В.Г. Потемкин ; под общ. ред. В.Г. Потемкина. – М. : ДИАЛОГ-МИФИ, 1999. – 287 с.
9. Первозванский, А.А. Курс теории автоматического управления: учеб. пособие / А.А. Первозванский. – СПб. : Лань, 2010. – 624 с.
10. Страшинин, Е.Э. Основы теории автоматического управления. Часть 1: Линейные непрерывные системы управления: учеб. пособие / Е.Э. Страшинин. – Екатеринбург : УГТУ-УПИ, 2000. – 214 с.
11. Теория автоматического управления : учебник для вузов / С.Е. Душин, Н.С. Зотов, Д.Х. Имаев и др.; под ред. В.Б. Яковлева. – М. : Высшая школа, 2009. – 567 с.

ПРИЛОЖЕНИЕ

Список команд, операторов и функций Matlab, использованных при создании m-файлов

1. Команды, операторы и функции общего назначения

`clc` – команда очистки командного окна.

`clear x` – команда удаления переменной `x` из рабочей области памяти;
команда `clear all` служит для удаления всех переменных из памяти.

`close` – команда удаления текущего окна с рисунком Figure;
команда `close all` закрывает все рисунки.

`switch...case` – условный оператор; в зависимости от значений `case_expr1`, `case_expr2`, `case_expr3` или других значений переменной `switch_expr` обеспечивает выполнение ряда предусмотренных в каждой ситуации действий `statements`. Синтаксис:

```
switch switch_expr
case case_expr1
    statement,...,statement
case case_expr2
    statement,...,statement
...
otherwise
    statement,...,statement
end
```

`disp` – команда вывода текста в командное окно.

Синтаксис: `disp(s)` ,

где `s` – строковая переменная (скаляр, вектор или матрица) или массив числовых значений. Пример: `disp('abc')`.

`num2str` – функция преобразования числа в символьную строку.

Синтаксис: `num2str(x)`

`length` – функция вычисления длины вектора.

Синтаксис: `length(v)`

`real` – функция вычисления действительной части комплексного числа.

Синтаксис: `real(z)`

`imag` – функция вычисления мнимой части комплексного числа.

Синтаксис: `imag(z)`

`open_system` – команда открытия модели в Simulink.

Синтаксис: `open_system('abc')`,

где `abc` – имя файла модели `abc.mdl`.

`sim` – команда запуска моделирования открытой в Simulink модели.

Синтаксис `sim('abc')`,

где `abc` – имя файла модели `abc.mdl`.

`close_system` – команда закрытия модели в Simulink.

Синтаксис: `close_system('abc',saveflag)`,

где `abc` – имя файла модели `abc.mdl`, `saveflag` – флаг разрешения сохранения файла перед закрытием (0 – не сохранять файл, 1 – сохранять файл).

2. Функции создания и преобразования моделей линейных систем с постоянными параметрами (linear time-invariant models), функции извлечения данных о моделях

`sys = ss(A, B, C, D)` – функция формирования непрерывной модели в ss-форме (в форме пространства состояний state-space), описываемой четверкой матриц $\{A, B, C, D\}$:

$$\begin{aligned}\dot{\bar{x}} &= A\bar{x} + B\bar{u} \\ \bar{y} &= C\bar{x} + D\bar{u}\end{aligned}$$

`ss_sys = ss(sys)` – функция преобразования lti-модели `sys` произвольного подкласса в модель `ss_sys` подкласса `ss`.

`[A, B, C, D] = ssdata(sys)` – функция, возвращающая значения четверки матриц $\{A, B, C, D\}$ для lti-объекта `sys` произвольного подкласса.

$p = tf('p')$ – функция, служащая для задания переменной преобразования Лапласа при описании lti-объекта в виде передаточной функции.

$tf_sys = tf(sys)$ – функция преобразования произвольной lti-модели sys в модель tf_sys подкласса tf . Модель подкласса tf – это модель, описанная при помощи передаточной функции (transfer function)

$$W(p) = \frac{N_1 p^m + N_2 p^{m-1} + \dots + N_m p + N_{m+1}}{D_1 p^n + D_2 p^{n-1} + \dots + D_n p + D_{n+1}}, \quad m \leq n,$$

а точнее вектором с коэффициентами полинома числителя $num = [N_1 \ N_2 \ \dots \ N_{m+1}]$ и вектором с коэффициентами полинома знаменателя $den = [D_1 \ D_2 \ \dots \ D_{n+1}]$.

$[num, den] = tfdata(sys, 'v')$ – функция, возвращающая числитель num и знаменатель den передаточной функции lti-объекта sys в виде числовых значений векторов-строк, а не в виде описания размерности массива ячеек.

$z_sys = zpks(sys)$ – функция преобразования произвольной lti-модели sys в модель подкласса $zpks$. Модель подкласса $zpks$ – это модель, описанная при помощи нулей, полюсов и коэффициента усиления:

$$W(p) = K \cdot \frac{(p - Z_1)(p - Z_2) \dots (p - Z_m)}{(p - Pl_1)(p - Pl_2) \dots (p - Pl_n)}, \quad m \leq n,$$

где Z_1, Z_2, \dots, Z_m – нули системы;

Pl_1, Pl_2, \dots, Pl_n – полюсы системы;

K – обобщенный коэффициент передачи.

$[z, pl, k] = zpksdata(sys, 'v')$ – функция, возвращающая нули z , полюсы pl и коэффициент усиления k lti-объекта sys , причем z и pl – в виде числовых значений векторов-столбцов, а не в виде описания размерности массива ячеек.

$eig(A)$ – функция нахождения собственных чисел матрицы A .

3. Функции синтеза контуров управления с обратной связью

`[L, prec, message] = place(A, B, pl)` – функция, рассчитывающая матрицу коэффициентов обратных связей L , которая обеспечивает желаемое расположение pl полюсов системы; также она возвращает оценку $prec$ близости собственных значений найденной замкнутой системы к их желаемым значениям (число точных десятичных цифр у полюсов найденной замкнутой системы) и в случае, если некоторые ненулевые полюсы замкнутой системы отличаются от желаемых значений более чем на 10%, выдает предупреждающее сообщение `message`. Уравнение управляющего воздействия для системы в замкнутом состоянии имеет вид

$$u = -Lx ,$$

где u – управляющее воздействие;

x – вектор состояния управляемого объекта.

4. Функции, используемые для построения графиков

`set(0, 'ShowHiddenHandles', 'on'), delete(get(0, 'Children'))` – последовательность команд для удаления всех графических окон; первая команда позволяет найти все (и даже скрытые) графические окна, а вторая удаляет их.

`figure` – команда, которая служит для создания пустого графического окна и отображения его на экране. Окно становится текущим, т. е. все последующие графические функции будут осуществлять построение графиков в этом окне. Для получения нового графического окна следует снова использовать `figure`.

`plot(X1, Y1, LineSpec1, X2, Y2, LineSpec2, ... 'MarkerSize', M_size)` – позволяет строить один или несколько графиков, изменять цвет и стиль линий и добавлять маркеры на каждый график, устанавливать размер маркеров. Дополнительный аргумент `LineSpec`, помещаемый за каждой парой векторов, – это заключенный в апострофы набор из трех символов, которые определяют цвет, тип маркера и тип линии (могут быть использованы как

все три позиции, так и одна или две, в зависимости от требуемых изменений). В табл. А приведены возможные значения данного аргумента с указанием результата.

Таблица А

Свойства линии

Сим-вол	Цвет	Сим-вол	Тип маркера	Сим-вол	Тип линии
y	Желтый	.	Точка	-	Сплошная
m	Розовый	o	Кружок	:	Пунктирная
c	Голубой	x	Крестик	-.	Штрихпунктирная
r	Красный	+	Знак «плюс»	--	Штриховая
g	Зеленый	*	Звездочка		
b	Синий	s	Квадрат		
w	Белый	d	Ромб		
k	Черный	v	Треугольник вершиной вниз		
		^	Треугольник вершиной вверх		
		<	Треугольник вершиной влево		
		>	Треугольник вершиной вправо		
		p	Пятиконечная звезда		
		h	Шестиконечная звезда		

`hold on` – команда, позволяющая выводить несколько графиков в одно окно.

`xlabel('s'), ylabel('s')` – функции создания подписей для осей графика.

`title('s')` – функция создания заголовка графика.

`legend('s1', 's2', ..., position)` – функция размещения на рисунке легенды с информацией о линиях, которыми построены графики. Ее применение необходимо в случае наличия на рисунке нескольких графиков. Порядок и количество аргументов должны соответствовать порядку вывода графиков и их количеству. Последним дополнительным аргументом `legend` может быть положение легенды в графическом окне.

`grid on` – команда включения сетки.

`ltiview('plottype', sys1, LineSpec1, sys2, LineSpec2, ...)` – функция вызова LTI Viewer (средства графического интерфейса пользователя,

используемого при расчете и анализе lti-систем). В качестве первого аргумента данной функции указывается тип выводимой на график характеристики:

- 'bode' – совмещенные в одном графическом окне АЧХ и ФЧХ системы;
- 'nyquist' – АФХ системы;
- 'impulse' – импульсная характеристика системы;
- 'step' – переходная характеристика системы.

При указании нескольких типов характеристик в фигурных скобках, например {'bode'; 'nyquist'}, графическое окно будет разбито на подокна, содержащие характеристики этих типов.

В одно графическое окно или подокно может быть выведено несколько однотипных характеристик, принадлежащих разным системам `sys1`, `sys2`, ... Можно указать спецификацию линии каждого графика `LineStyle` (цвет и тип линии, тип маркеров).

При построении переходного процесса для системы `sys` можно отказаться от автоматического определения времени наблюдения и указать дополнительный аргумент `Tfinal`, содержащий конечное значение времени наблюдения: `ltiview('step', sys, Tfinal)`.

Настройки LTI Viewer доступны через команды меню в верхней части графического окна, а также через дополнительное меню, вызываемое нажатием правой клавиши мыши при установке указателя на нужном рисунке. Например, при выполнении команды `Properties` дополнительного меню появляется окно "Property Editor", где во вкладке `Characteristics` можно выбрать опцию вычисления максимума переходного процесса (`Peak Response`), времени переходного процесса (`Setting Time`), времени нарастания (`Rise Time`), установившегося значения (`Steady State`). Включение сетки, масштабирование и копирование рисунка, ряд других операций также доступны через команды меню.

Учебное электронное текстовое издание

Малов Андрей Викторович
Страшинин Евгений Эрастович
Цветков Александр Владимирович

**ТЕОРЕТИЧЕСКИЕ ОСНОВЫ УПРАВЛЕНИЯ
ТЕХНИЧЕСКИМИ СИСТЕМАМИ**

Методические указания к практическим занятиям

Редактор:
Компьютерная верстка:

Я.О. Смирнова
авторская

Рекомендовано Методическим советом
Разрешено к публикации 04.06.2015
Электронный формат – pdf
Объем 1,31 уч.-изд. л.



620002, Екатеринбург, ул. Мира, 19

Информационный портал УрФУ
<http://www.urfu.ru>