

Лабораторная работа: Анализ динамических объектов с использованием программных средств системы MATLAB и SIMULINK

Цели работы: Ознакомление с пакетом прикладных программ Control System Toolbox системы MATLAB, предназначенным для работы с линейными стационарными системами и интерактивным пакетом SIMULINK, предназначенным для моделирования нелинейных динамических систем.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ РАБОТЫ

Классы вычислительных объектов в MATLAB

Классом в MatLAB принято называть определенную форму представления вычислительных объектов в памяти компьютера в совокупности с правилами (процедурами) их преобразования. Класс определяет тип переменной, а правила — операции и функции, которые могут быть применены к этому типу. В свою очередь, тип определяет объем памяти, которая отводится под запись переменной в память и структуру размещения данных в этом объеме. Операции и функции, которые могут быть применены к определенному типу переменных, образуют методы этого класса.

Основные классы объектов в MatLAB

В системе MatLAB определены 6 встроенных классов вычислительных объектов:

double	Числовые массивы и матрицы действительных или комплексных чисел с плавающей запятой в формате двойной точности, наиболее распространенный тип переменных в системе MatLAB, с которым оперирует большинство функций и процедур
sparse	Двумерные комплексные разреженные матрицы двойной точности. Разреженная структура применяется для компактного хранения матриц с незначительным количеством ненулевых элементов. Разреженные матрицы требуют применения специальных методов для решения задач
char	Массивы символов - переменные, являющиеся совокупностью символов, каждый символ занимает 16 битов, эту совокупность часто называют строкой
struct	Массивы записей (структуры). Объекты класса struct состоят из нескольких составляющих, которые называются полями, каждое из которых имеет собственное имя
cell	Массивы ячеек. Переменные класса cell (ячейки) являются совокупностью некоторых других массивов. Массивы ячеек позволяют объединить связанные данные (возможно, разных типов и размеров) в единую структуру
uints	Массивы 8-разрядных целых чисел без знаков, позволяет хранить целые числа от 0 до 255 в 1/8 части памяти, необходимой для чисел двойной точности. Никакие математические операции для этого класса данных не определены.

Каждому типу данных соответствуют собственные функции и операторы обработки, т.е. методы.

Производные классы MatLAB

Рассмотренные выше классы объектов построены таким образом, что на их основе могут быть созданы новые классы объектов.

В языке MatLAB отсутствует необходимость и возможность предварительного объявления типа или класса переменных, которые будут использованы. То же самое относится и к объектам любых вновь создаваемых классов.

Объекты класса создаются в виде структур (записей), т.е. относятся к потомкам (наследникам) класса **struct**. Поля структуры и операции с полями являются доступными только внутри методов данного класса.

Все М-файлы, определяющие методы для объектов данного класса, должны размещаться в специальном каталоге, который называется каталогом класса и обязательно имеет имя, состоящее из знака @ (коммерческое "эт") и имени класса, т.е. имеет вид @<имя класса>. Каталог класса должен быть подкаталогом одного из каталогов, описанных в путях доступа системы MatLAB, но не самим каталогом. Каталог класса обязательно должен содержать М-файл с именем, совпадающим с именем класса. Этот файл называют конструктором класса. Назначение такого М-файла — создавать объекты этого класса, используя данные в виде массива записей (структуры) и приписывая им метку класса.

В системе MatLAB на этой основе создан и используется встроенный класс объектов **sym**, с которым работает пакет символьных вычислений Symbolic Math Toolbox, и который позволяет выполнять вычисления с символьными переменными и матрицами. Пакет Control System Toolbox использует класс объектов LTI и три его дочерних подкласса **tf**, **zpk**, **ss**, которые поддерживают алгоритмы анализа линейных стационарных систем автоматического

Классы пакета CONTROL

Пакет прикладных программ (ППП) Control System Toolbox (сокращенно —CONTROL) сосредоточен в подкаталоге CONTROL каталога TOOLBOX системы MatLAB.

Основными вычислительными объектами этого ППП являются:

родительский объект (класс) LTI (Linear Time-Invariant System — линейные, инвариантные во времени системы); в русскоязычной литературе за этими системами закрепилось название линейные стационарные системы (ЛСС).

дочерние объекты (классы), т.е. подклассы класса LTI, соответствующие трем разным представлениям ЛСС:

- TF-объект (Transfer Function — передаточная функция);
- ZPK-объект (Zero-Pole-Gain — нули-полюсы-коэффициент передачи);
- SS-объект (State Space — пространство состояния).

Объект LTI, как наиболее общий, содержит информацию, независимую от конкретного представления ЛСС (непрерывного или дискретного), а также от имен входов и выходов. Дочерние объекты определяются конкретной формой представления ЛСС, т.е. зависят от модели представления. Объект класса TF характеризуется векторами коэффициентов числителя и знаменателя рациональной передаточной функции. Объект класса ZPK характеризуется векторами, содержащими значения нулей, полюсов передаточной функции системы и коэффициента передачи системы. Наконец, объект класса SS определяется четверкой матриц, описывающих динамическую систему в пространстве состояния. Ниже приведены основные атрибуты этих классов, их обозначения и смысл.

Атрибуты (поля) LTI-объектов

Ниже NU, NY и NX определяют число входов (вектор U), выходов (вектор Y) и переменных состояния (вектор X) ЛСС соответственно; OM (SISO) — одномерная система, т.е. система с одним входом и одним выходом; MM (MIMO) — многомерная система (с несколькими входами и выходами).

	Специфические атрибуты передаточных функций (TF-объектов)
num	Числитель Вектор-строка для OM-систем; для MM-систем — массив ячеек из векторов-строк размером NY на NU (например, {[1 0] 1 ; 3 [1 2 3]})
den	Знаменатель Вектор-строка для OM-систем; для MM-систем — массив ячеек из векторов-строк размером NY на NU. Например: <code>tf([-5 ; [1 -5 6]] , {[1 -1] ; [1 1 0]})</code>

	определяет систему с одним входом и двумя выходами $[-5 / (s-1)]$ $[(s^2-5s+6) / (s^2+s)]$
Variable	Имя (тип) переменной (из перечня) Возможны варианты: s, p, z, gl-1 или q. По умолчанию принимается s (для непрерывных переменных) и z (для дискретных). Имя переменной влияет на отображение и создает дискретную ПФ для дискретных сигналов
	<u>Специфические атрибуты ZPK-объектов</u>
z	Нули Вектор-строка для ОМ-систем; для ММ-систем — массив ячеек из векторов-строк размером NY на NU
p	Полюсы Вектор-строка для ОМ-систем; для ММ-систем — массив ячеек из векторов-строк размером NY на NU
k	Коэффициенты передачи Число — для ОМ-систем, матрица NY на NU для ММ-систем
Variable	Имя (тип) переменной (из перечня) То же, что и для TF-объекта (см. выше)
	<u>Специфические атрибуты SS-объектов (моделей пространства состояния)</u>
a,b,c,d	A,B,C,D — матрицы, в соответствии с уравнениями в переменных состояния: $x = Ax + Bu, y = Cx + Du$
e	E — матрица для систем Descriptor'a (описателя). По умолчанию $E = eye(size(A))$
StateName	Имя переменной состояния (не обязательное). Массив ячеек NX на I из строк (используйте " для состояний без имени). Пример: {'положение'; 'скорость'}
	<u>Атрибуты, общие для всех LTI-моделей</u>
Ts	Дискрет по времени (в секундах) Положительный скаляр (период дискретизации) для дискретных систем. $Ts = -1$ для дискретных систем с неустановленной частотой дискретизации. $Ts = 0$ для непрерывных систем.
Td	Задержки входов (в секундах) Вектор 1 на NU промежутков времен задержек входов. Установка Td как скаляра определяет единую задержку по всем входам. Используется только для непрерывных систем. Используйте D2D для установки задержек в дискретных системах. $Td = []$ для дискретных систем
InputName	Имена входов Строка для систем с одним входом. Массив ячеек NU на 1 из строк для систем с несколькими входами (используйте " для переменных без имени). Примеры: 'момент' или {'напор1'; 'отклонение элеронов1'}
OutputName	Имена выходов Строка для систем с одним выходом. Массив ячеек NY на 1 из строк для систем с несколькими выходами (используйте " для переменных без имени). Пример: 'мощность' или {'скорость'; 'угол атаки'}
Notes	Заметки Любая строка или массив ячеек из строк символов. Пример: 'Эта модель создана в январе 2000'
Userdata	Дополнительная информация или данные. Может быть любого типа MATLAB

Функции из перечня методов класса LTI:

`tfddata, ssdata, zpkdir, step, impulse, rlocus, pade, series, parallel, bode, margin, nichols, nyquist, ss2ss, augstate, damp, get, issiso, Iticheck, trange, balreal, display, gram, kalman, set, tzero, dssdata, kalmd, modred, pzmap, sigma, uplus, canon, eig, inherit, Iqgreg, quickset, connect, estim, initial, Iqry, norm, reg, covar, evalfr, isct, Isim, rlocfind, ctrb, fgrid, isempty, Iti, obsv,`

Перечень основных процедур пакета CONTROL, сгруппированных по функциональному назначению

	<u>Создание LTI-моделей</u>
<code>ss</code>	Создает модель пространства состояния
<code>zpk</code>	Создает модель нули/полюсы/коэффициенты передачи
<code>tf</code>	Создает модель передаточной функции
<code>dss</code>	Специфицирует описатель модели пространства состояния
<code>filt</code>	Специфицирует цифровой фильтр
<code>set</code>	Установка/модификация атрибутов LTI-модели
<code>Itiprops</code>	Детальная справка об атрибутах LTI-моделей
	<u>Извлечение данных</u>
<code>ssdata</code>	Извлечение матриц пространства состояния
<code>zpkdir</code>	Извлечение данных о нулях/полюсах/КП
<code>tfddata</code>	Извлечение числителя(-лей) и знаменателя(-лей) ПФ
<code>dssdata</code>	Получение информации о версии описателя SSDATA
<code>get</code>	Получение информации о значениях свойств LTI-модели
	<u>Получение информации об отдельных характеристиках модели</u>
<code>class</code>	Информация о типе модели (ss, zpk или tf)
<code>size</code>	Информация о размерах матриц входа и выхода
<code>isempty</code>	Проверка, является ли LTI-модель пустой
<code>isct</code>	Проверка, является ли модель непрерывной
<code>isdt</code>	Проверка, является ли модель дискретной
<code>isproper</code>	Проверка, является ли модель правильной
<code>issiso</code>	Проверка, имеет ли модель один вход и один выход
<code>isa</code>	Проверка, является ли LTI-модель моделью заданного типа
	<u>Преобразование системы</u>
<code>ss</code>	Преобразование в пространство состояния
<code>zpk</code>	Преобразование в нули/полюсы/КП

Работа с передаточными функциями с использованием функций пакета CONTROL

Передаточная функция формируется функцией **tf**, в качестве аргументов задаются коэффициенты полиномов числителя и знаменателя в порядке убывания степени оператора **s**:

```
>> tf([3 1], [1 1 1])
```

```
Transfer function:
 3 s + 1
-----
s^2 + s + 1
```

Целесообразно выполнить присвоение модели, представляемой данной передаточной функцией некоторой переменной:

```
>> w1 = tf([3 1], [1 1 1])
```

```
Transfer function:
 3 s + 1
-----
s^2 + s + 1
```

Теперь данную переменную можно использовать при обращении к функциям или выполняя необходимые преобразования, используя операторы Matlab.

Например, передаточная функция модели **SYS**, состоящей из параллельно соединенных блоков **SYS1** и **SYS2**, рис. 1 может быть получена следующим образом:

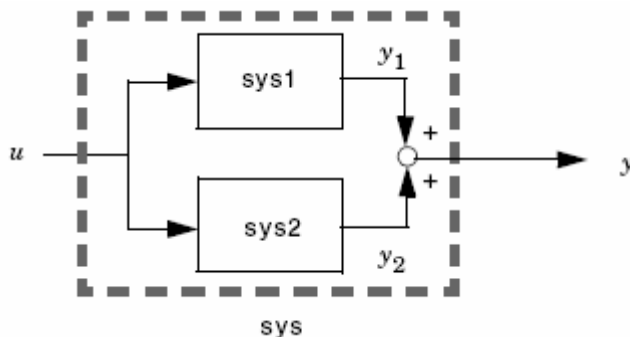


Рис. 1

В первом варианте результат получен путем операции сложения, а во втором – с использованием функции **parallel**.

И в том и в другом случае получается идентичный результат, рис. 2.

```
sys1 = tf([1 1],[1 2 1])
sys2 = tf([1],[3 1])
sys = sys1 + sys2
```

ИЛИ

```
sys1 = tf([1 1],[1 2 1])
sys2 = tf([1],[3 1])
parallel(sys1,sys2)
```

```
Transfer function:
```

```
  s + 1
-----
s^2 + 2 s + 1
```

```
Transfer function:
```

```
  1
-----
3 s + 1
```

```
Transfer function:
```

```
  4 s^2 + 6 s + 2
-----
3 s^3 + 7 s^2 + 5 s + 1
```

Рис. 2

Для преобразования моделей предназначены следующие функции:

parallel	параллельное соединение
feedback	соединение с обратной связью
series	последовательное соединение
append, connect	различные соединения блоков

Выполняя преобразования систем с использованием операторов и функций необходимо внимательно следить за получаемыми результатами. Например, передаточную функцию системы с единичной отрицательной обратной связью можно получить с использованием операторов Matlab:

```
>> w1 = tf([3 1], [1 1 1])
Transfer function:
   3 s + 1
-----
s^2 + s + 1

>> W1 = w1/(1+w1)
Transfer function:
   3 s^3 + 4 s^2 + 4 s + 1
-----
s^4 + 5 s^3 + 7 s^2 + 6 s + 2
```

Однако результат не вполне понятен, т.к. увеличился порядок системы. Преобразование с помощью функции **feedback** дает иной результат:

```
>> W2 = feedback(w1,1)

Transfer function:
   3 s + 1
-----
s^2 + 4 s + 2
```

Выполним преобразование передаточной функции $W1 = w1/(1+w1)$ в **zpk** форму:

```
>> zpk(W1)

Zero/pole/gain:
   3 (s+0.3333) (s^2 + s + 1)
-----
(s+3.414) (s+0.5858) (s^2 + s + 1)
```

Из полученного результата видно, что в числителе и знаменателе присутствует множитель второго порядка $(s^2 + s + 1)$, который соответствует знаменателю передаточной функции прямой цепи и подлежит сокращению в результирующей формуле. Проверяем:

```
>> zpk(W2)

Zero/pole/gain:
   3 (s+0.3333)
-----
(s+3.414) (s+0.5858)
```

Результат соответствует формуле для **W1** при сокращении сомножителя $(s^2 + s + 1)$.

Анализ свойств динамической системы

При анализе системы, состоящей из нескольких блоков, возникает задача получения передаточной функции для всей системы или для подсистемы, включающей в себя ряд блоков. Для выполнения преобразований целесообразно задать передаточные функции всех блоков системы, а затем выполнять расчеты для подсистем и системы в целом.

Например, для расчета параметров передаточной функции силового гиросtabilизатора можно написать следующую программу.

```
% исходные данные
H = 2000;
Jp = 0.8;
Jst = 125;
Dst = 300;
Dp = 0;
Kr = 50000;
% расчет параметров
Omega = H * sqrt(1+Dst*Dp/H^2)/sqrt(Jst * Jp);
t0 = 1/Omega;
dzeta =
((Dp/(2*H))/sqrt(Jst/Jp) + (Dst/(2*H))*sqrt(Jp/Jst)) * (1/sqrt(1+(Dst*Dp)/H^2));

t1 = 2*t0;
% формирование передаточных функций
w1 = tf([Kr/H],[1 0]);
w2 = tf([1],[t0^2 2*t0*dzeta 1]);
w3 = tf([1],[t1 1])*tf([1],[t1 1]); % корректирующее звено
% передаточная функция гиросtabilизатора с коррекцией
W = w1 * w2 * w3;
```

В результате получена передаточная функция

```
>> W
Transfer function:
                    25
-----
2.5e-009 s^5 + 5.06e-007 s^4 + 0.0001262 s^3 + 0.02006 s^2 + s
```

Или, в форме **zpk**:

```
>> zpk(W)

Zero/pole/gain:
          10000000000
-----
s (s+100)^2 (s^2 + 2.4s + 4e004)
```

В Matlab предусмотрены функции извлечения данных из модели. С помощью функции **tfdata** получим коэффициенты полиномов числителя и знаменателя передаточной функции гиросtabilизатора:

```
>> [n,d] = tfdata(W, 'v')
n =
    0    0    0    0    0    25
d =
 2.5000e-009 5.0600e-007 1.2620e-004 2.0060e-002 1.0000e+000    0
```

С помощью функции **roots** получим значения корней полинома знаменателя

```
>> roots(d)
ans =
         0
-1.2000e+000 +2.0000e+002i
-1.2000e+000 -2.0000e+002i
-1.0000e+002
-1.0000e+002
```

С помощью функции **pzmap** получим значения полюсов и нулей системы:

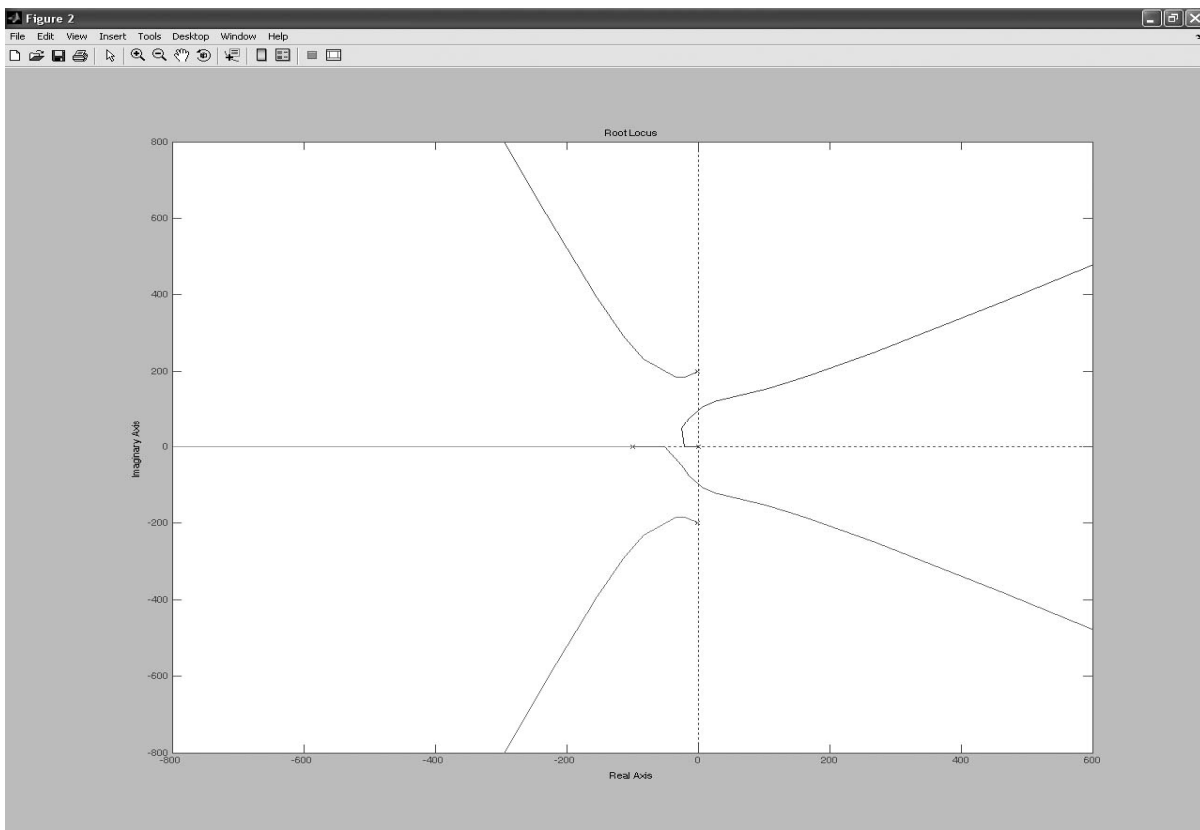
```
>> [np,dz] =pzmap(W)
np =
      0
-1.2000e+000 +2.0000e+002i
-1.2000e+000 -2.0000e+002i
-1.0000e+002
-1.0000e+002
dz =
Empty matrix: 0-by-1
```

Функция **damp** вычисляет показатели демпфирования и собственные частоты системы:

```
>> damp(W)
```

Eigenvalue	Damping	Freq. (rad/s)
0.00e+000	-1.00e+000	0.00e+000
-1.00e+002	1.00e+000	1.00e+002
-1.00e+002	1.00e+000	1.00e+002
-1.20e+000 + 2.00e+002i	6.00e-003	2.00e+002
-1.20e+000 - 2.00e+002i	6.00e-003	2.00e+002

Функция **rlocus** строит корневой годограф.



Основы работы с системой Simulink

SIMULINK является программной системой интерактивного моделирования и позволяет использовать вычислительные возможности MATLAB для решения широкого круга задач моделирования и анализа систем.

Для вызова подсистемы **Simulink** необходимо задать команду **simulink** в командной строке MATLAB или нажать кнопку **Simulink** на панели инструментов MATLAB. При этом открывается окно **Simulink Library Browser** – библиотека Simulink рис. 1. Окно библиотеки содержит позиции меню **File Edit View Help** и панель инструментов

Для дальнейшей работы есть возможность создать новую модель или открыть файл модели, созданный ранее, выполнив команду из меню или нажав кнопку панели инструментов.

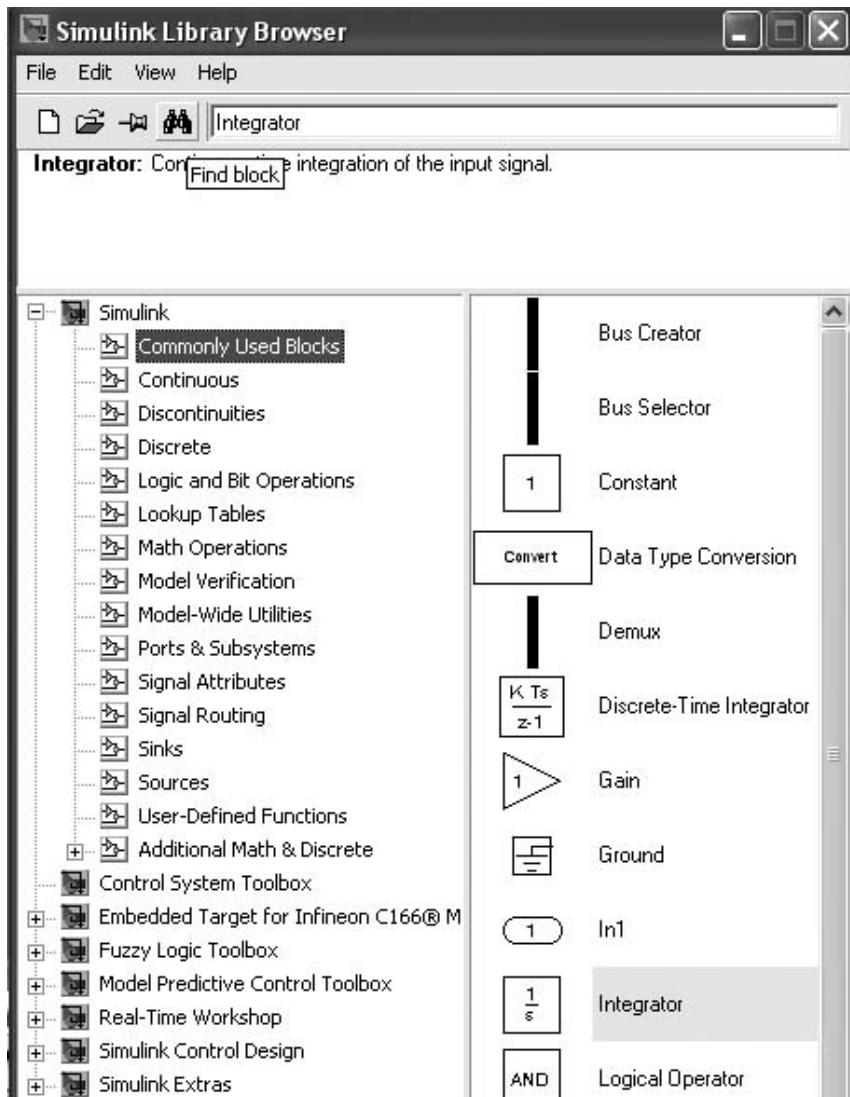


Рис. 1

Модель состоит из блоков и связей, или линий передачи сигналов. Блоки сгруппированы по назначению, группы отображаются в виде иерархической структуры в окне библиотеки. Нужный блок можно найти, набрав его название в строке поиска и нажав кнопку **Find Block** в строке инструментов (бинокль). На рис. 1 показано, как система нашла блок **Integrator**. Для того, чтобы создать новую модель, следует нажать кнопку **Create a new model** на панели инструментов библиотеки или выполнив соответствующую команду из меню **File**. При этом открывается новое окно – поле сборки модели. Если выполняется команда открытия ранее созданного файла, открывается окно с моделью, собранной ранее, рис. 2.

При вызове Simulink открывается окно **Simulink Library Browser**. Затем можно создать новую модель или открыть файл модели, созданный ранее.

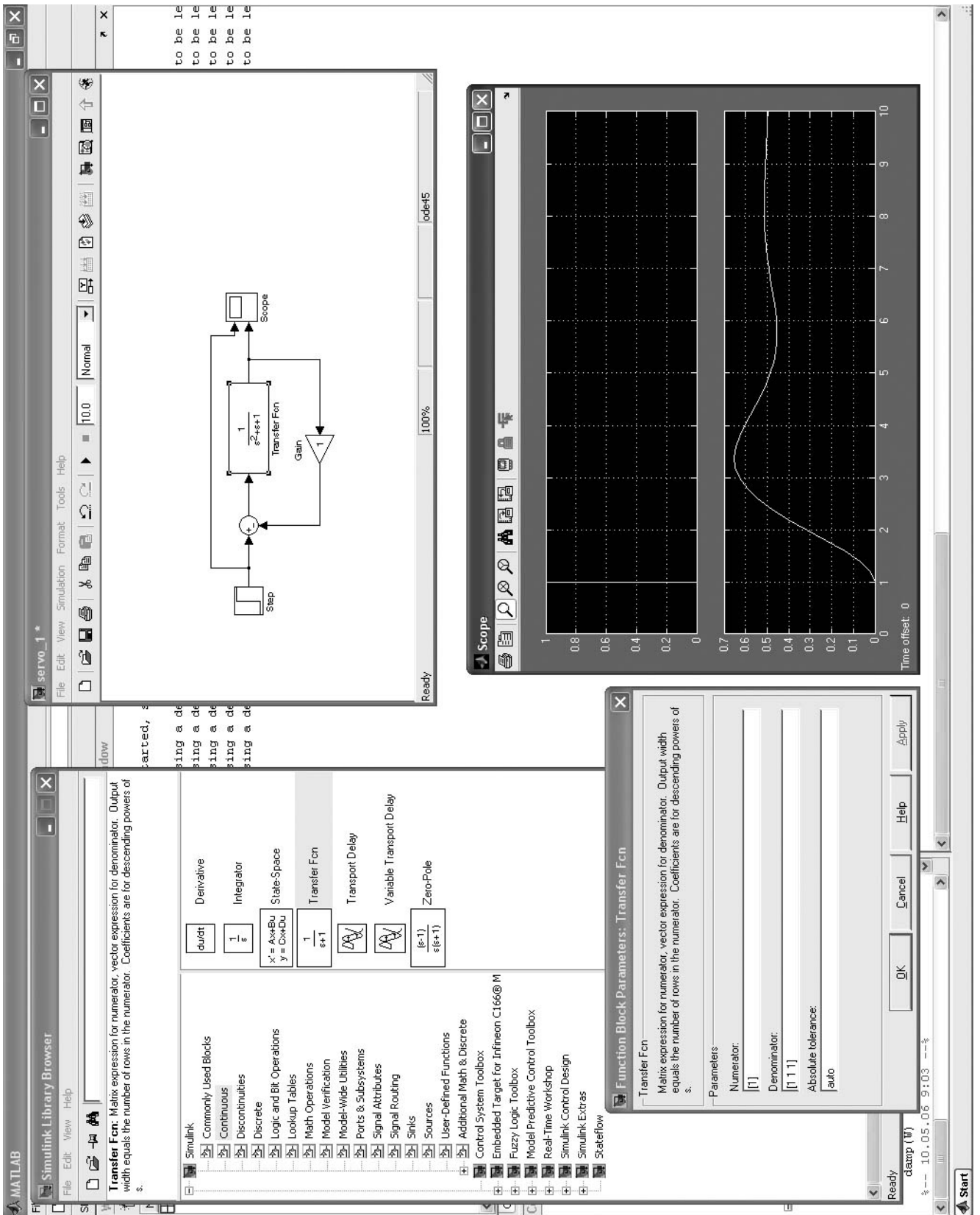


Рис. 2.

Для "сборки" схемы модели блок из окна библиотеки необходимо перетащить в окно (поле сборки) новой модели. Двойной щелчок по блоку открывает окно установки параметров. На рис. 3 показан блок модели – источник постоянного сигнала с единичным значением. Более сложные блоки требуют задания большего числа параметров, например, для генератора необходимо ввести форму колебаний, амплитуду и частоту, причем частоту можно задать в единицах Герц или рад/с рис. 4.

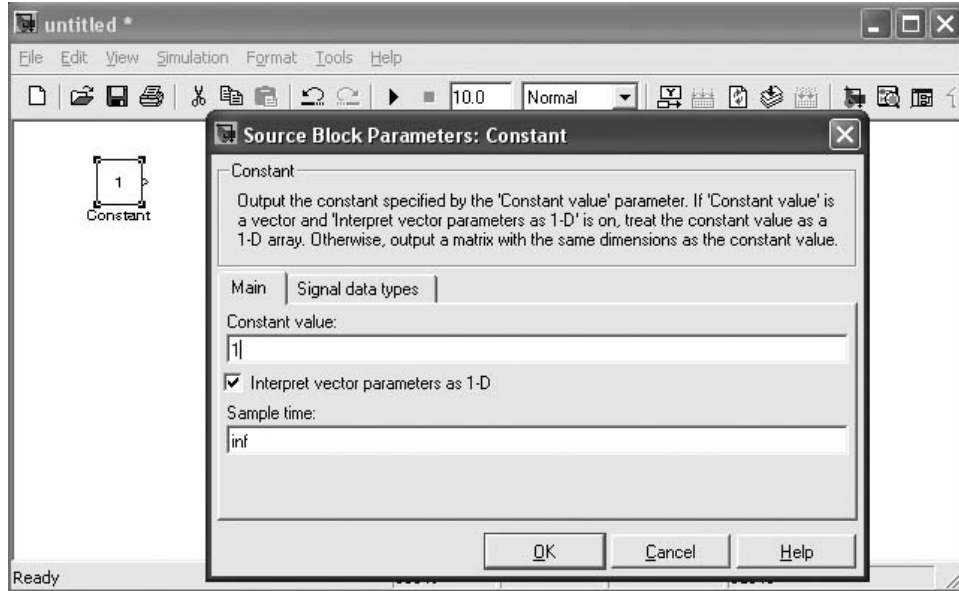


Рис. 3

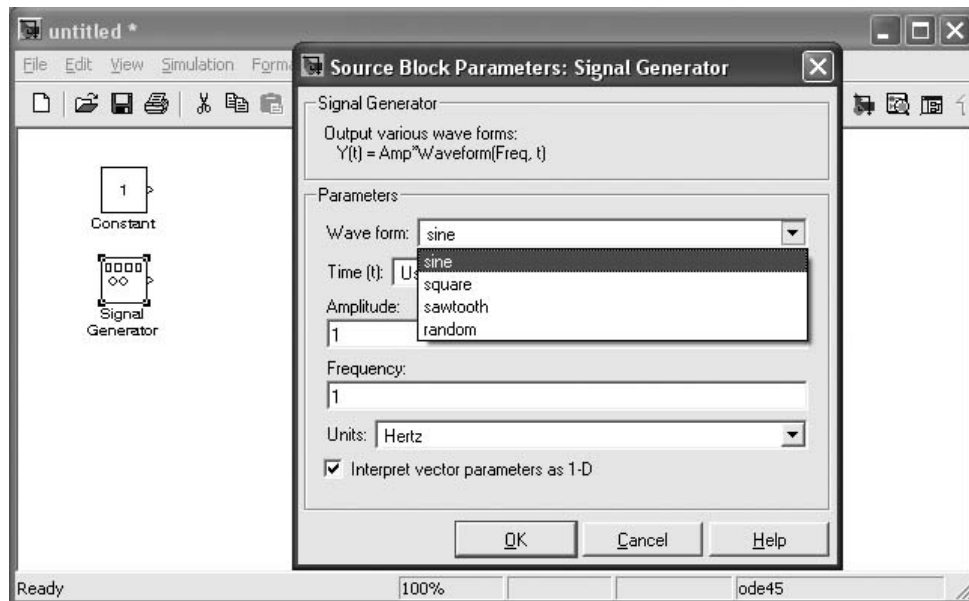


Рис. 4

Анализ свойств динамической системы

Для анализа свойств динамической системы, созданной в Simulink можно применить LTI-viewer. Для этого необходимо ввести в схему модели блоки **In** и **Out** на вход и выход, соответственно, рис. 5.

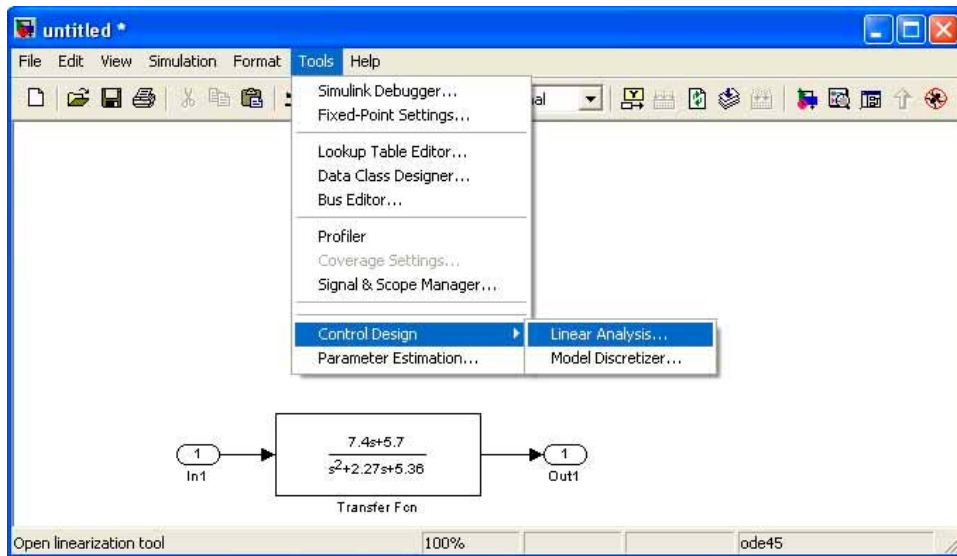


Рис. 5

Для обозначения входа и выхода можно также указать на точку схемы (линию связи блоков) мышью и нажав правую кнопку. При этом выпадает меню, в котором следует выбрать позицию **Linearisation Points** и далее **Input** или **Output** рис. 6. Затем, через меню **Tools-Control Design-Linear Analysis** запускаем LTI-viewer.

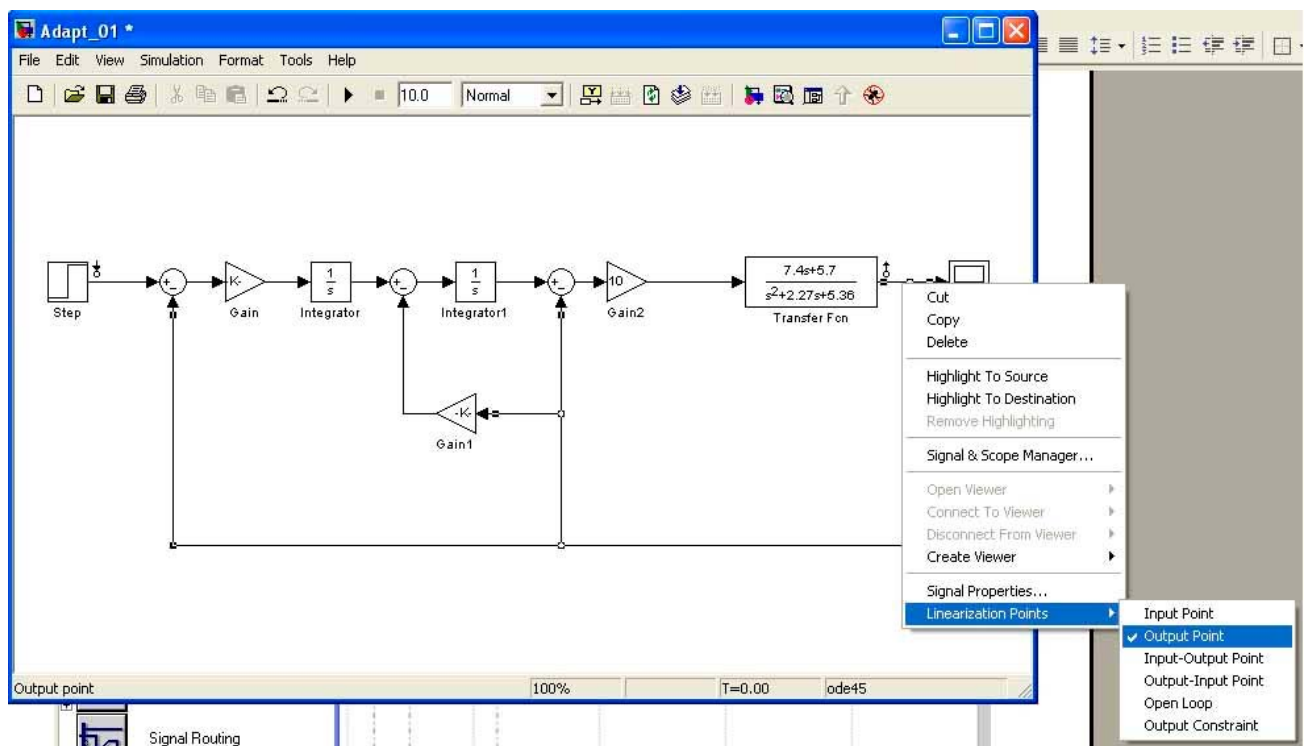


Рис. 6

Создание подсистемы

При создании больших и сложных моделей можно собирать схему по частям, из подсистем. Если схема модели полностью готова, можно обвести с помощью мышки группу блоков рис. 7и в меню **Edit** задать команду **Create Subsystem**, рис. 8.

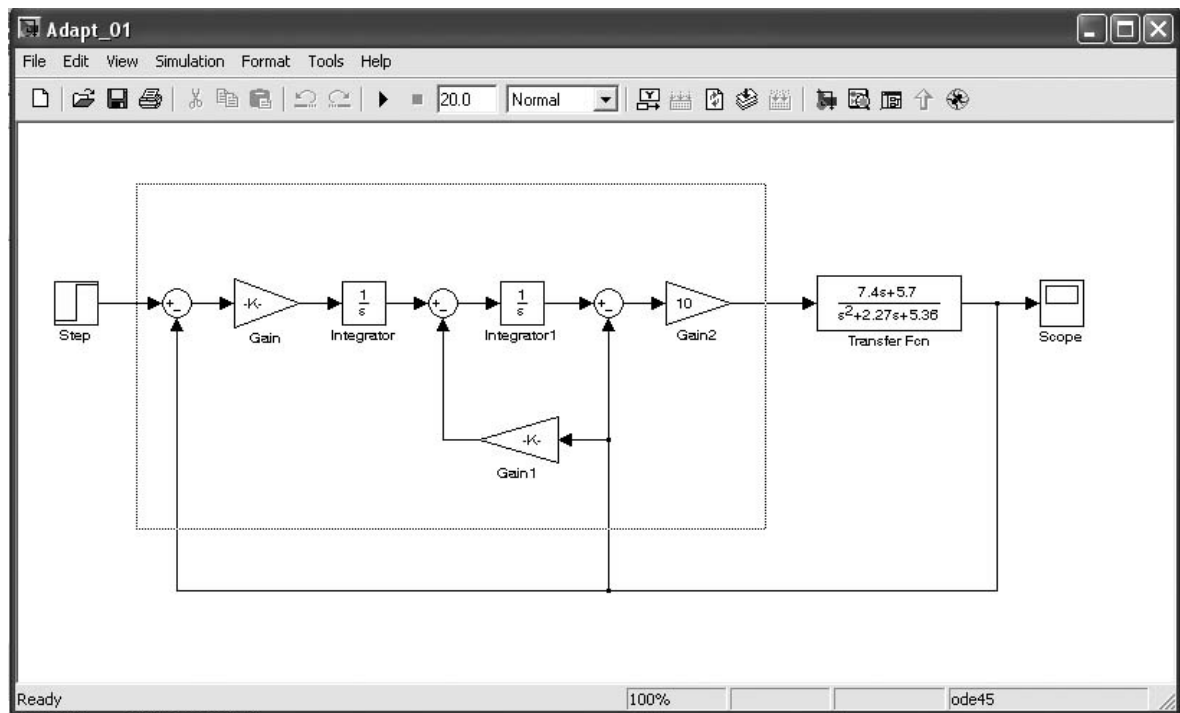


Рис. 7

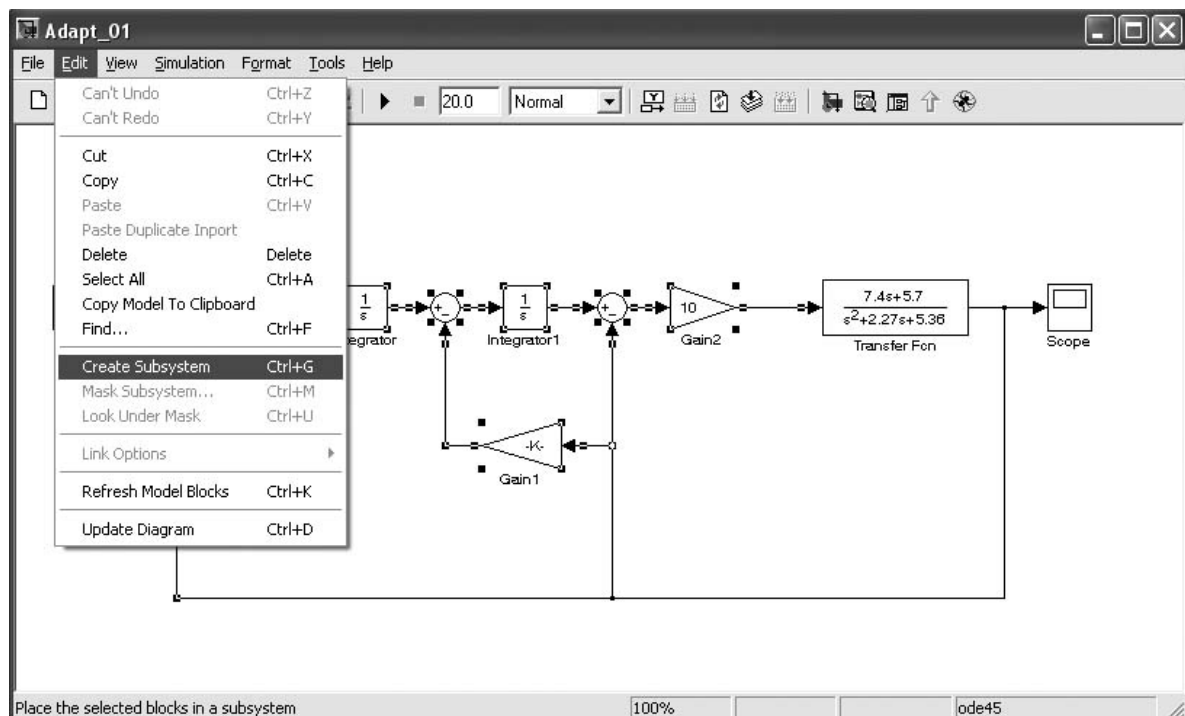


Рис. 8

В результате этого выделенная группа блоков объединяется в один блок (Subsystem), который хранится в том же файле и раскрывается при щелчке левой кнопкой мышки по блоку на схеме всей модели, рис. 9.

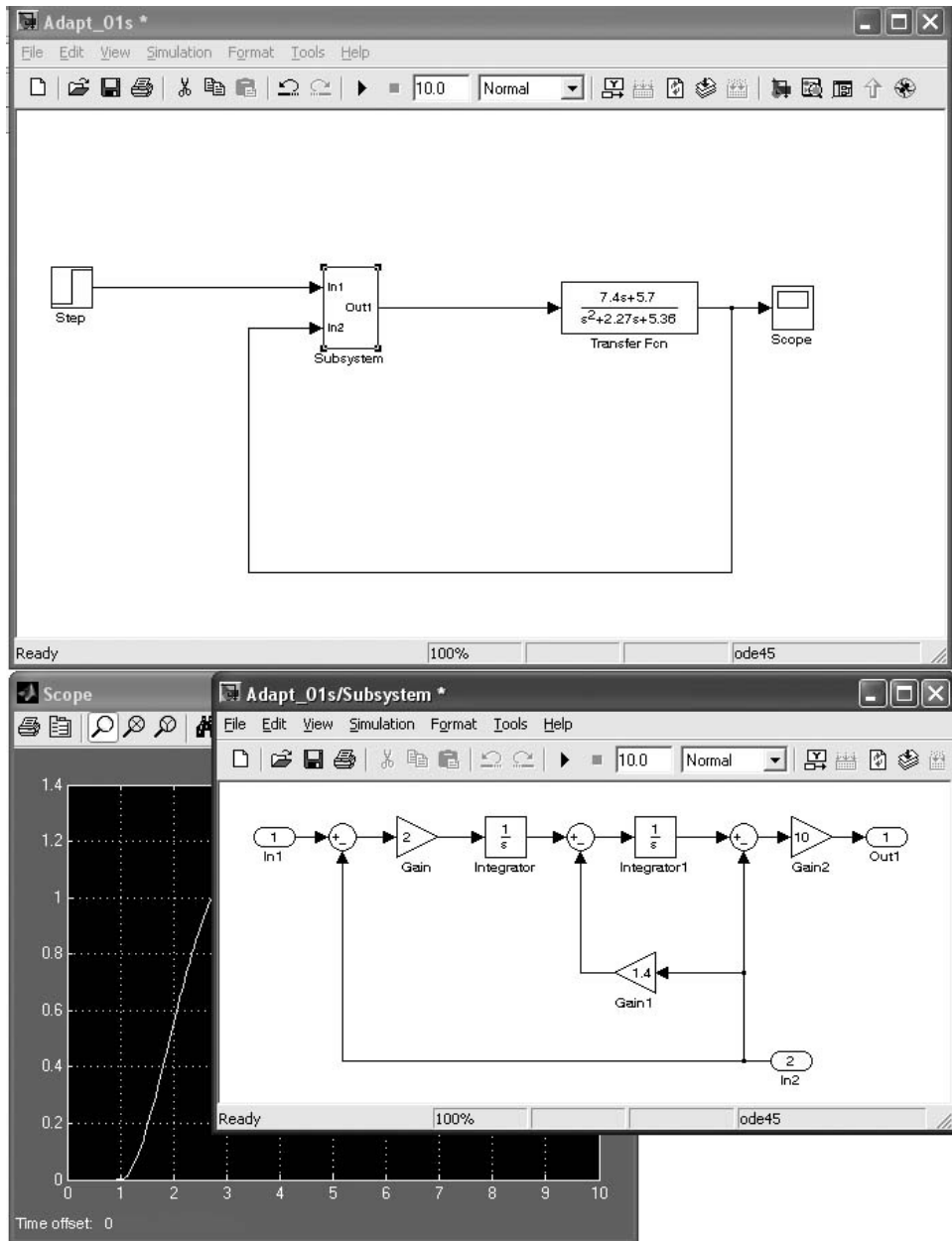


Рис. 9

Из группы блоков можно сразу набирать подсистему, для этого следует поместить на наборное поле необходимое количество блоков **In** и **Out**, рис. 10.

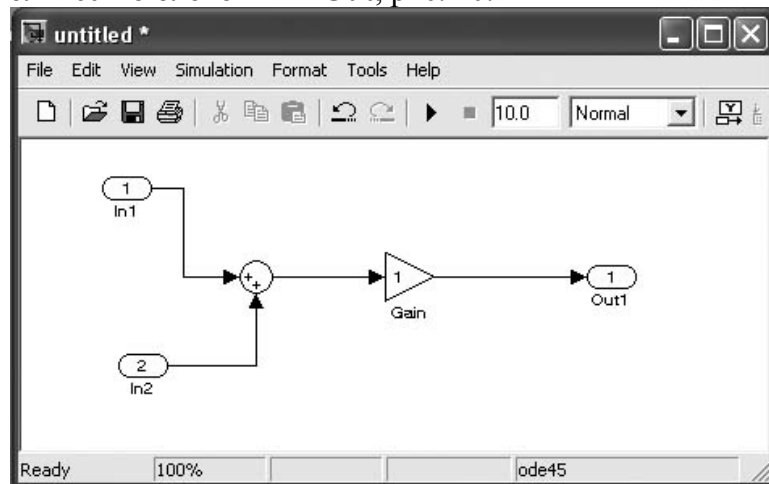


Рис. 10

Передача данных из Simulink в рабочую среду MATLAB

Результаты расчетов, выполняемые в **Simulink** не отображаются в рабочем пространстве MATLAB. Для того, чтобы присвоить значения параметров, например, данные переходного процесса, какой-либо переменной MATLAB, необходимо ввести в схему блок **To Workspace**, в окне его параметров задать имя переменной и формат вывода – **Array**, рис. 11. Теперь, при выполнении моделирования, в рабочей среде MATLAB появится переменная с заданным именем, также появится переменная **tout**, с дискретами времени.

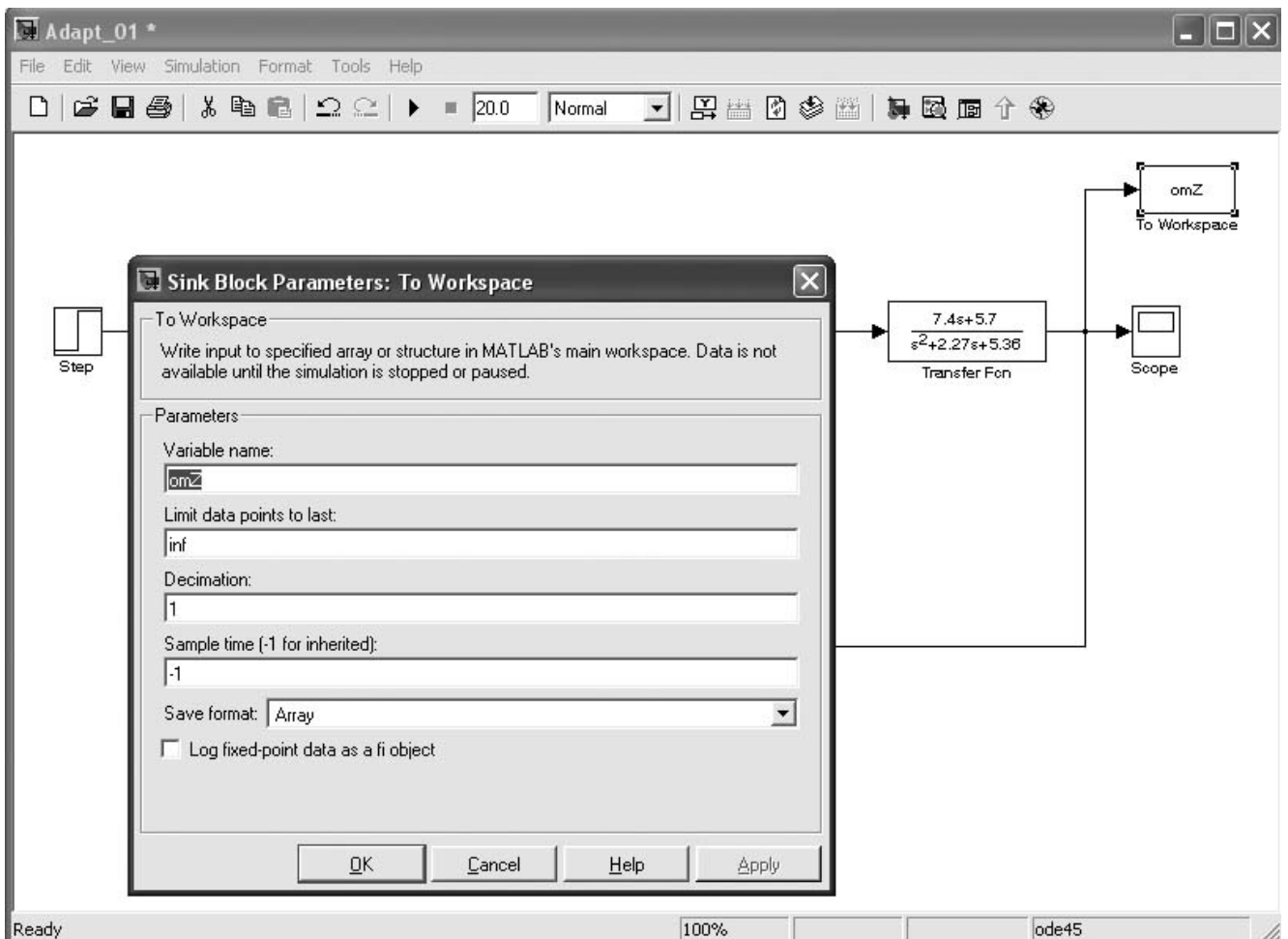


Рис. 11

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2: анализ линейных стационарных систем в MATLAB, обработка передаточных функций, создание и анализ моделей динамических объектов в Simulink.

ЗАДАНИЕ НА ЛАБОРАТОРНЫЙ ПРАКТИКУМ

1. Изучить способы задания параметров линейных стационарных систем (LTI) в MATLAB. Задать параметры передаточных функций блоков из схемы гиросtabilизатора, сформировать передаточную функцию замкнутой и разомкнутой систем.
2. Выполнить анализ свойств гиросtabilизатора (параметры взять из курсовой работы), получить корни полиномов передаточных функций, построить ЛАХ и корневой годограф, переходный процесс (использовать LTIviewer).
3. Выполнить анализ свойств гироскопического стабилизатора с использованием Simulink. Параметры переходного процесса вывести в рабочую среду MATLAB.
4. Выполнить настройку и анализ свойств динамической системы ЛА-САУ с использованием Simulink. Сформировать подсистему из блоков эталонной модели в прямой цепи системы. Построить графики переходных процессов для заданных переменных системы. Определить:
 - время регулирования;
 - перерегулирование.

Задача: выполнить настройку канала управления угловой скоростью тангажа самолета с адаптивным алгоритмом управления на основе эталонной модели в прямой цепи. Модель объекта управления – самолета, задана в виде передаточной функции

$$W_{\omega_z}(p) = \frac{b_1 p + b_0}{p^2 + a_1 p + a_0} = \frac{\omega_z(p)}{\delta_\beta(p)} \quad (1)$$

Где ω_z и δ_β – приращения угловой скорости тангажа и угла отклонения руля высоты; a_0, a_1, b_0, b_1 – коэффициенты, определяемые аэродинамическими свойствами самолета. Алгоритм управления построен на основе эталонной модели, динамические свойства которой определены дифференциальным уравнением второго порядка:

$$\ddot{\omega}_z^* + g_1 \dot{\omega}_z^* + g_0 \omega_z^* = g_0 \omega_z^0 \quad (2)$$

Числовые значения параметров g_0 и g_1 назначаются такими, чтобы динамические параметры системы самолет–САУ соответствовали заданным требованиям к качеству переходных процессов, протекающих при поступлении на вход заданного сигнала $\omega_z^0 = const$.

Структурная схема системы ЛА–САУ представлена на рис.12

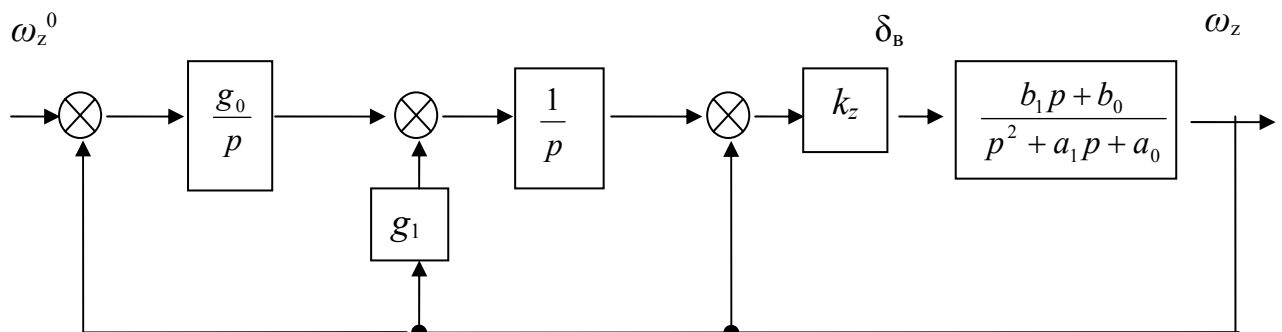


Рис. 12

Параметры g_0 и g_1 эталонной модели рассчитываются по заданным значениям времени регулирования $t_{\text{пер}}$ и коэффициента демпфирования ξ (постоянная времени $T \approx t_{\text{пер}}/3$,

собственная частота $\omega = 1/T$, время регулирования $t_{\text{рег}}$ выбирается большим, чем постоянная времени объекта регулирования, в данном случае $t_{\text{рег}} = 1.5 - 5$ с, $\xi = 0.5 - 1$).

Величину коэффициента k_z следует уточнить экспериментально. Теоретически, при $k_z \rightarrow \infty$, переходный процесс в системе ЛА-САУ в точности соответствует заданным параметрам эталонной модели. Но большие значения коэффициентов невозможно реализовать средствами аналоговой техники, а в цифровом вычислителе усложняется алгоритм контроля. Приемлемое качество регулирования в системе данного вида достигается при $k_z = 1 - 10$.

Модель системы для данного примера, сформированная в среде Matlab-Simulink, приведена на рис. 13.

Варианты задания

Номер режима полета	a_0	a_1	b_0	b_1
1.	5.36	2.27	5.7	7.4
2.	89.48	6.9	159.0	57.0
3.	121.4	4.66	96.6	42.0
4.	8.52	1.66	7.78	13.8
5.	18.11	0.65	3.4	10.0

Вариант	Номер режима полета	коэффициент демпфирования ξ	время регулирования $t_{\text{рег}}$
1.	1	0.4, 0.7, 1.0	3 с
2.	1	0.4, 0.7, 1.0	4 с
3.	1	0.4, 0.7, 1.0	6 с
4.	2	0.4, 0.7, 1.0	3 с
5.	2	0.4, 0.7, 1.0	4 с
6.	2	0.4, 0.7, 1.0	6 с
7.	3	0.4, 0.7, 1.0	3 с
8.	3	0.4, 0.7, 1.0	4 с
9.	3	0.4, 0.7, 1.0	6 с
10.	4	0.4, 0.7, 1.0	3 с
11.	4	0.4, 0.7, 1.0	4 с
12.	4	0.4, 0.7, 1.0	6 с
13.	5	0.4, 0.7, 1.0	3 с
14.	5	0.4, 0.7, 1.0	4 с
15.	5	0.4, 0.7, 1.0	6 с

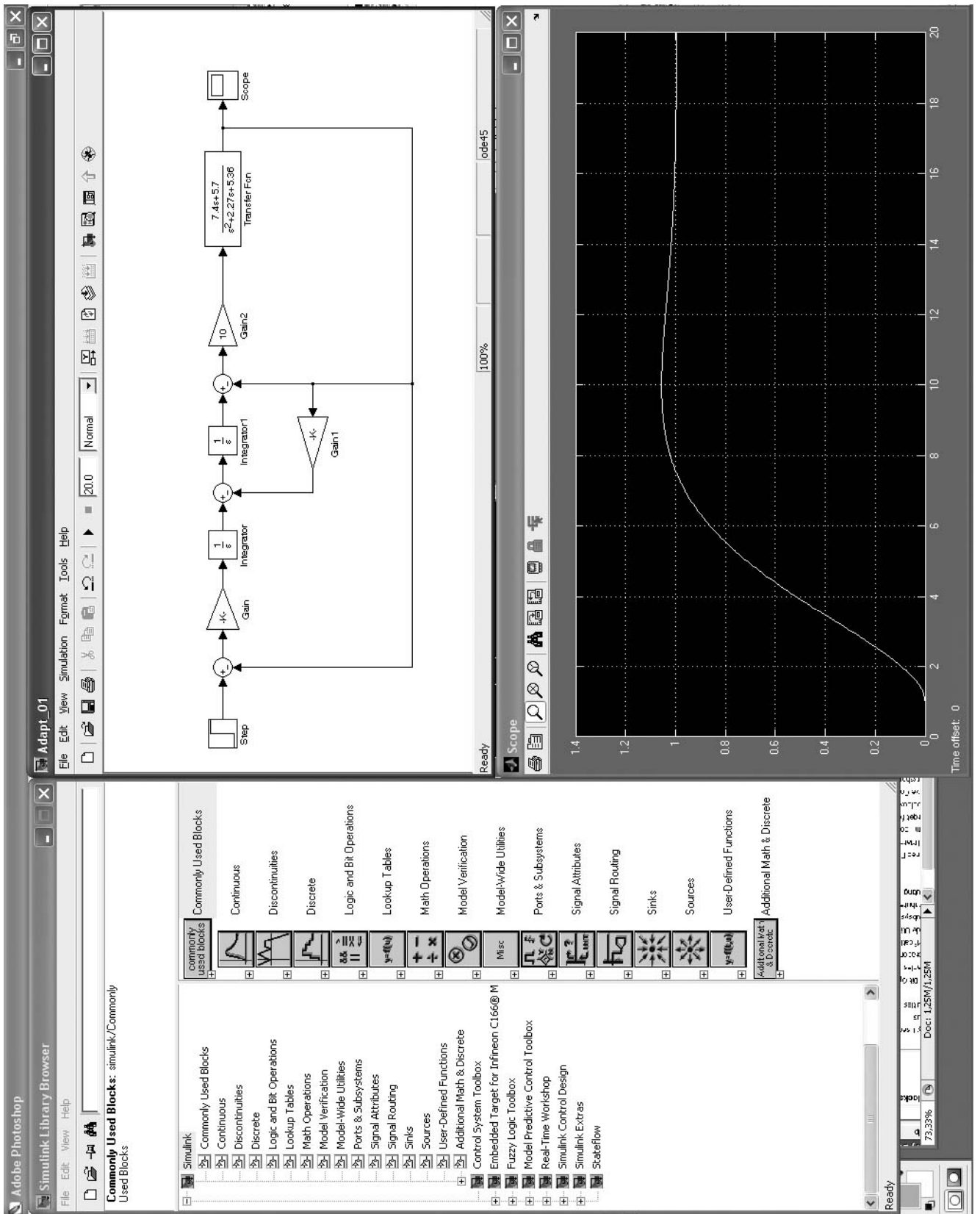


Рис. 13