

**МИНИСТЕРСТВО ТРАНСПОРТА И КОММУНИКАЦИЙ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**Кафедра «Микропроцессорная техника  
и информационно-управляющие системы»**

**А.А. КОРОЛЕВ, К.Ф. ИЗМАЙЛОВ**

**Автоматизированное проектирование микропроцессорных систем**

**Учебно-методическое пособие по выполнению лабораторных работ  
для студентов электротехнического факультета**

**Гомель 2016**

## СОДЕРЖАНИЕ

СОДЕРЖАНИЕ .....	2
Краткая характеристика системы компьютерной математики (СКМ) MatLab .....	3
Математическое моделирование и параметрическая оптимизация технических систем.....	10
Приемы моделирования в Simulink.....	15
1 Общие сведения о пакете <i>Simulink</i> .....	15
2 Интерфейс пакета <i>Simulink</i> .....	16
3 Создание и редактирование модели в <i>Simulink</i> .....	18
Элементы работы с ЛТИ-объектами.....	21
Лабораторная работа № 1 .....	30
Лабораторная работа № 2.....	32
Лабораторная работа № 3.....	33
Лабораторная работа № 4.....	34
Лабораторная работа № 5.....	35
Лабораторная работа №6.....	35
Лабораторная работа № 7.....	36

# Краткая характеристика системы компьютерной математики (СКМ) MatLab

## 1. Пакеты СКМ MatLab

СКМ MatLab является мощным средством для исследования различных физических процессов и, в частности, систем автоматического управления. Среди возможностей MatLab отметим три пакета: пакет Control System Toolbox (CST), Simulink и Power System (PS), которые целесообразно применять при изучении и исследовании систем автоматического управления, используя на каждом этапе исследования сильные стороны одного из предложенных пакетов.

Пакет Control System Toolbox позволяет исследовать линейные системы с привлечением современного математического аппарата. Он представляет широкий набор процедур, осуществляющих анализ систем регулирования с самых различных точек зрения и, прежде всего, определения откликов системы на внешние воздействия, как во временной, так и в частотной областях. По своим возможностям этот пакет является наглядной иллюстрацией достижений современной теории управления, уделяя внимание не только анализу и синтезу систем, но и корректности математических преобразований.

Пакет Simulink позволяет осуществлять исследование (моделирование) линейных и нелинейных систем. Создание моделей в пакете Simulink основано на технологии Drag-and-Drop (схвати и перетащи). В качестве «кирпичиков» для построения моделей используются модули или блоки, хранящиеся в библиотеке Simulink. По сравнению с пакетом Control System Toolbox пакет Simulink работает с нелинейными системами и больше приближен к физическим моделям.

Пакет Power System позволяет создавать виртуальные модели систем регулирования, что дает возможность исследовать не только саму систему, но и отдельные аппараты, из которых она состоит. Причем, детализация аппаратов настолько глубокая, что характеристики математической модели в пакете Power System практически не отличаются от физических моделей.

## 2. Пакет Control System Toolbox

Все объекты, используемые в пакете Control System Toolbox, объединены в один класс – lti-объекты. В зависимости от вида математического описания lti-объекты, могут быть заданы либо парой многочленов числитель/знаменатель передаточной функции (команда tf), либо нулями, полюсами и обобщенным коэффициентом передачи (команда zpk), либо четверкой матриц ( $A$ ,  $B$ ,  $C$ ,  $D$ ), задающих модель системы в пространстве состояния (команда ss или dss). Команда dss также описывает линейную систему четверкой матриц ( $A$ ,  $B$ ,  $C$ ,  $D$ ), но используется тогда, когда уравнения переменных не разрешены относительно производных.

Указанные процедуры позволяют создать непрерывные и дискретные модели. Для дискретных моделей к числу входных параметров следует добавить значение параметра  $T_{\Pi}$  - шага дискретизации, а вводимые значения коэффициентов должны задавать параметры дискретных передаточных функций (для команд tf и zpk) либо матриц конечно-разностных уравнений пространства состояния – при использовании команд ss и dss.

Известно, что  $z$ -передаточные функции могут быть представлены в двух видах. В теории управления используют переменную  $z$  и располагают многочлены числителя и знаменателя в порядке убывания степени этой переменной. Например  $F(z) = (z + 5)/(z^2 + 2z + 1)$ . При цифровой обработке сигналов предпочитают записывать эту передаточную функцию как функцию переменной  $z^{-1}$ . Например  $F(z^{-1}) = (z^{-1} + 5z^{-2})/(1 + 4z^{-1} + z^{-2})$ .

При положительных степенях переменной  $z$  используются команды `tf` и `zpk` с указанием параметра  $T_{II}$ , а при отрицательных степенях используется команда `filt` с указанием параметра  $T_{II}$ , либо команды `tf` и `zpk` с указанием указателя переменной 'variable', 'z^-1'.

Если значение  $T_{II} = -1$ , то период дискретности для дискретной системы не специфицирован, т.е. при выполнении команд `tf`, `zpk`, `ss` и `filt` периоду дискретности не присваивается никакого значения.

Ниже приведено несколько примеров, иллюстрирующих применение описанных команд.

**% Создание LTI-моделей**

```

H1=tf([1,4],[1,2,200])           %Создание модели в форме tf
H2=tf([1,4],[1,2,200],0.1)      %Создание дискретной модели
H3=tf([1,4],[1,2,200],-1)      %с периодом дискретности 0,1.
                                %Время дискретности не определено.
H4=tf([1,4],[1,2,200],'variable','z^-1')
H5=filt([1,4],[1,2,200])

```

**% Преобразование моделей**

```

H1=tf([1,3,2],[1,7,12,0])      %Исходные данные.
F1=zpk(H1)                     %Переход от формы tf к zpk.
F2=ss(H1)                      %Переход от формы tf к ss.
F3=tf(F2)                      %Переход от формы ss к tf.
F4=tf(F1)                      %Переход от формы zpk к tf.

```

Результаты выполнения приведенных выше команд.

```

Transfer function:              %Передаточная функция H1
      s + 4                    %непрерывной системы
-----
s^2 + 2 s + 200

Transfer function:              %Передаточная функция H2
      z + 4                    %дискретной системы с TII = 0,1 с.
-----
z^2 + 2 z + 200

Sampling time: 0.1

Transfer function:              %Передаточная функция дискретной
      z + 4                    %системы H3.Время дискретизации
-----                    %не определено (не специфицирована)
z^2 + 2 z + 200              %TII = -1

Sampling time: unspecified

Transfer function:              %Передаточные функции H4, H5
      1 + 4 z^-1              %дискретных систем записаны
-----                    %по убывающим степеням z
1 + 2 z^-1 + 200 z^-2

```

Sampling time: unspecified

Transfer function:

$$1 + 4 z^{-1}$$

-----  
 $1 + 2 z^{-1} + 200 z^{-2}$

Sampling time: unspecified

Transfer function:

$$s^2 + 3 s + 2$$

-----  
 $s^3 + 7 s^2 + 12 s$

Zero/pole/gain:

$$(s+2) (s+1)$$

-----  
 $s (s+4) (s+3)$

a =

	x1	x2	x3
x1	-7	-1.5	0
x2	8	0	0
x3	0	2	0

b =

	u1
x1	1
x2	0
x3	0

c =

	x1	x2	x3
y1	1	0.375	0.125

d =

	u1
y1	0

Continuous-time model.

Transfer function:

$$s^2 + 3 s + 2$$

-----  
 $s^3 + 7 s^2 + 12 s$

Transfer function:

$$s^2 + 3 s + 2$$

-----  
 $s^3 + 7 s^2 + 12 s$

%Передаточная функция H1  
%записана в формате tf

%Переход от формы tf к форме zpk

%Переход от формы tf к форме ss

%Матрица коэффициентов a

%Матрица управления b

%Матрица выхода c

%Переход от формы ss к форме tf

%Переход от формы zpk к форме tf

Возможности пакета Control System Toolbox не ограничиваются приведенными командами. Для ознакомления с новыми командами и лучшего их использования целесообразно провести классификацию команд.

Функции пакета Control System Toolbox, описанные выше, могут быть классифицированы следующим образом.

Функции, создающие lti-модели.

Таб.1

set	Установка свойств
ss	Формирование ss-модели системы в явной форме Коши
dss	Формирование ss-модели системы в неявной форме Коши
tf	Формирование tf-модели в форме передаточной функции
zpk	Формирование zpk-модели
filt	Формирование модели в форме дискретного фильтра

Функции, преобразующие lti-модели.

Таб.2

c2d	Построение дискретной модели непрерывной системы
d2c	Построение непрерывной модели дискретной системы
d2d	Изменение периода дискретности
ss	Преобразование моделей к ss-форме
tf	Преобразование моделей к tf-форме
zpk	Преобразование моделей к zpk-форме

Функции, соединения lti-моделей.

Таб.3

parallel	Параллельное соединение (сложение lti-моделей)
series	Последовательное соединение (умножение lti-моделей)
feedback	Соединение с обратной связью
append	Объединение lti-моделей с добавлением входов и выходов
connect	Объединение lti-моделей с использованием матрицы соединений
star	Объединение взаимосвязанных многомерных моделей

Кроме приведенных выше функций к блоку команд формирования и преобразования lti-моделей следует отнести команды по извлечению информации об lti-моделях и команды, переопределяющие базисные функции. Однако при первом рассмотрении используются только команды, приведенные в таб. 1, 2 и 3.

Для создания модели нужно предварительно либо привести уравнения всей системы к форме уравнений пространства состояний, либо найти передаточные функции системы. В общем случае это довольно сложная и громоздкая задача. В то же время реальные системы автоматического управления (САУ) состоят из соединенных между собой отдельных блоков (динамических звеньев), уравнения которых обычно достаточно просты. Поэтому в практике проектирования САУ принято использовать структурные методы, когда САУ задается как определенная схема соединения отдельных элементарных динамических звеньев, и фактически проектируется одно или несколько из этих звеньев таким образом, чтобы обеспечить заданное качество всей системы. В соответствии с этим в MatLab предусмотрена возможность "набирать" программно "схему" САУ путем предварительного ввода моделей звеньев, составляющих САУ, и последующего "соединения" этих звеньев в

единую структуру. К процедурам, осуществляющим расчет характеристик соединений отдельных звеньев, относятся:

`plus (minus)` – выполняет "параллельное соединение" указанных при обращении звеньев, т.е. определяет характеристики модели системы из параллельно соединенных звеньев; особенностью является то, что вызов этих процедур может быть осуществлен не только обычным способом: путем указания имени процедуры и перечисления (в скобках после имени) идентификаторов соединенных звеньев, но и простым указанием идентификаторов звеньев, которые должны быть объединены, с постановкой между ними знаков "+" (при суммировании выходных сигналов звеньев) или "-" (при вычитании выходных сигналов).

`parallel` – осуществляет ту же процедуру параллельного соединения звеньев; в отличие от предыдущей процедуры может использоваться для многомерных систем и осуществления параллельного соединения лишь по некоторым входам и выходам;

`mtimes` (или знак "\*" между именами звеньев) – осуществляет последовательное соединение звеньев, имена которых указаны; применяется для одномерных систем;

`series` – последовательное частичное соединение многомерных систем;

`feedback` – такое соединение двух звеньев, когда второе указанное звено составляет цепь отрицательной обратной связи для первого звена;

`append` – формальное объединение не связанных между собой систем (добавление выходов и входов второй системы к выходам и входам первой);

`connect` – установление соединений выходов и входов многомерной системы, созданной предварительно формальным объединением процедурой `append`; схема соединений задается матрицей  $Q$  соединений, указываемой как один из входных параметров процедуры;

`inv` – рассчитывает САУ, обратную указанной, т.е. такую, у которой выходы и входы поменяли местами;

`vertcat` – производит так называемую вертикальную конкатенацию (сцепление) систем (звеньев), т.е. такое их объединение, когда входы этих систем становятся общими, а выходы остаются независимыми; для такого объединения необходимо, чтобы число входов объединяемых систем было одинаковым; тогда число входов в результирующей системе останется таким же, как и в каждой из объединяемых систем, а число выходов будет равно сумме выходов объединяемых систем;

`horzcat` – осуществляет "горизонтальное сцепление" указанных систем, при котором выходы становятся общими, а входы добавляются.

Проиллюстрируем применение некоторых из этих процедур. Создадим модель управления двигателем постоянного тока состоящего из трех последовательно соединенных звеньев: регулятора  $W_{rt}(s)$ ; усилителя мощности  $W_{um}(s)$ ; объекта регулирования  $W_{or}(s)$ . В цепи обратной связи стоит звено -  $K_{DT}$ , а регулятор тока состоит из параллельного соединения усилительного звена -  $W_{II}(s) = K_{II}$  и интегрирующего  $W_{II}(s) = K_{II}/s$ .

Параллельное соединение  $W_{II}(s)$  и  $W_{II}(s)$  можно осуществить двумя способами: либо используя команду `parallel`.

`Wn=tf(0.2,1)`

`%Исходные данные`

```
Transfer function:
0.2
```

```
Wi=tf(23.8,[1,0])
```

```
%Исходные данные
```

```
Transfer function:
23.8
```

```
-----
s
```

```
Wrt=parallel(Wn,Wi)
```

```
%Переменные команды parallel
```

```
Transfer function:
0.2 s + 23.8
```

```
%Результат выполнения
%команды parallel
```

```
-----
s
```

либо применяя операцию "Сложение" моделей

```
Wrt=Wn+Wi
```

```
Transfer function:
0.2 s + 23.8
```

```
%Результат применения
%операции «Сложение»
```

```
-----
s
```

Последовательное соединение звеньев *Wrt* , *Wum* и *Wor* можно так же осуществить двумя способами - применением процедур *series*.

```
Wum=tf(22,[0.003,1])
```

```
%Исходные данные
```

```
Transfer function:
22
```

```
-----
0.003 s + 1
```

```
Wor=tf(92,[0.008,1])
```

```
%Исходные данные
```

```
Transfer function:
92
```

```
-----
0.008 s + 1
```

```
Wps=series(Wrt,Wum*Wor)
```

```
Transfer function:
```

```
404.8 s + 4.817e004
-----
2.4e-005 s^3 + 0.011 s^2 + s
```

```
%Результат выполнения
%команды series
```

ИЛИ

```
Wps=series(Wrt*Wum,Wor)
```

```
Transfer function:
```

```
404.8 s + 4.817e004  
-----  
2.4e-005 s^3 + 0.011 s^2 + s
```

```
%Результат выполнения  
%команды series
```

либо просто перемножив передаточные функции моделей

```
Wps=Wrt*Wum*Wor
```

```
Transfer function:
```

```
404.8 s + 4.817e004  
-----  
2.4e-005 s^3 + 0.011 s^2 + s
```

Теперь найдем модель замкнутой системы управления двигателем постоянного тока по управлению, используя команду `feedback`. В цепи обратной связи стоит усилительное звено  $K_{DT}$ .

```
KDT=0.0345
```

```
%Коэффициент обратной связи
```

```
Wzu=feedback(Wps,KDT)
```

```
%Передаточная функция замкнутой  
%системы по управлению
```

```
Transfer function:
```

```
404.8 s + 4.817e004  
-----  
2.4e-005 s^3 + 0.011 s^2 + 14.97 s + 1662
```

Аналогично определяется передаточная функция замкнутой системы по возмущению.

```
UZB=feedback(Wor,KDT*Wrt*Wum)
```

```
Transfer function:
```

```
%Передаточная функция замкнутой  
%системы по возмущению
```

```
0.276 s^2 + 92 s  
-----  
2.4e-005 s^3 + 0.011 s^2 + 14.97 s + 1662
```

В приведенных примерах команды MatLab использовались для определения передаточных функций простых систем и их применение, во-первых, очевидно, а, во-вторых, не очень экономит время. Эффективность применения команд MatLab, которые приведены в урезанном виде, значительно возрастает при исследовании сложных многомерных систем с перекрестными связями.

## Математическое моделирование и параметрическая оптимизация технических систем

Рассмотрим процесс создания математической модели технической системы и оптимизацию её параметров в соответствии со сформулированным критерием на примере теплотехнической системы.

В настоящее время всё более актуальной становится задача проектирования и использования теплотехнических систем (ТС) небольшой и средней мощности, которые позволяют создавать автономные системы отопления, оптимизированные по энергопотреблению и которые могут использовать различные виды топлива. При этом современные подобные системы зачастую способны адаптироваться и перестраиваться под различные виды топлива, а система автоматики способна поддерживать оптимальные режимы энергопотребления. В силу того, что потребность в таких системах возрастает, актуальной становится задача автоматизации процесса расчета основных параметров оборудования теплотехнических систем и дальнейшего выбора компонент оборудования из числа известных.

В настоящее время разработаны, известны и описаны в литературе алгоритмы оптимального расчета и выбора параметров теплотехнических систем. Тем не менее, на рынке стран СНГ практически отсутствуют доступные и надежные компьютерные программы подобных автоматических расчетов с последующим автоматизированным выбором элементов системы и их математическим описанием, на основе которого проектировщик мог бы выполнить моделирование и оптимизацию параметров проектируемой системы.

Необходимость в подобном моделировании продемонстрируем на следующем примере системы автоматического регулирования температуры в помещении (рис.1):

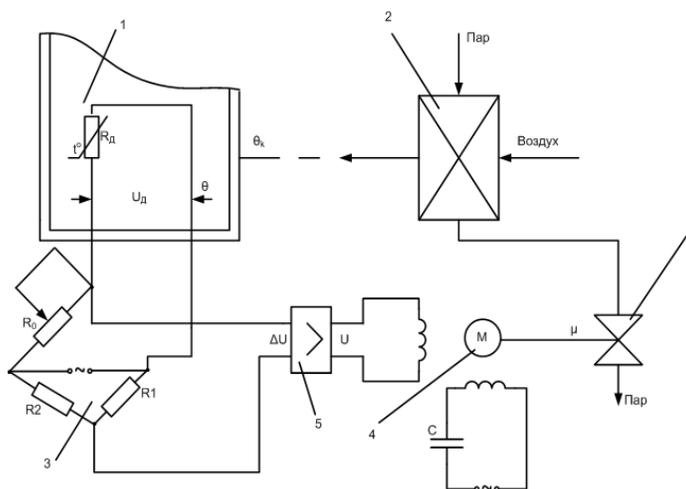


Рисунок 1. Схема системы автоматического регулирования температуры в помещении: 1 – помещение; 2 – теплообменник (калорифер); 3 – измерительная мостовая схема; 4 – двухфазный исполнительный двигатель; 5 – дифференциальный магнитный усилитель; 6 – клапан (заслонка).

Математическую модель объекта регулирования (помещения) определяют на основе уравнения теплового баланса:

$$C_{п} m_{п} \frac{d\theta}{dt} + \theta = \Phi_{п..п.} + \Phi_{о..п.},$$

где  $C_n$  – усредненная теплоемкость, определяемая теплоемкостями воздуха и предметов, находящихся в помещении;  $m_n$  – усредненная масса, определяемая массами воздуха и предметов, находящихся в помещении;  $\Phi_{п.п.}$  – поток теплоты, поступающий в помещение от калорифера;  $\Phi_{о.п.}$  – поток теплоты, отводимый из помещения через стенки в окружающую среду.

Запишем это уравнение иначе:

$$T_0 \frac{d\theta}{dt} + \theta = \theta_k + kf,$$

где  $T_0$  – постоянная времени, с;  $\theta$  – значение температуры воздуха в помещении, °С;  $\theta_k$  – значение температуры воздуха на выходе калорифера, °С;  $k$  – коэффициент передачи;  $f$  – возмущающее воздействие на объекте регулирования, т.е. в рассматриваемой в данном случае технической системе – это изменение температуры внешнего воздуха, которая, очевидно, влияет на температуру в помещении.

Математическую модель регулирующего органа (калорифера) определяют аналогичным образом:

$$C_k m_k \frac{d\theta_k}{dt} + \theta_k = \Phi_{кл.},$$

где  $C_k$  – усредненная теплоемкость для калорифера;  $m_k$  – усредненная масса для калорифера;  $\Phi_{кл.}$  – поток теплоты, поступающий в калорифер при открытии клапана.

Данное уравнение можно записать следующим образом:

$$T_3 \frac{d\theta_k}{dt} + \theta_k = k_3 \mu,$$

где  $T_3$  – постоянная времени, с;  $\theta_k$  – значение температуры воздуха на выходе калорифера, °С;  $k_3$  – коэффициент передачи;  $\mu$  – линейное перемещение клапана, см.

Математическую модель воспринимающего органа (датчика) можно записать в виде следующего соотношения:

$$U_d = k_1 \theta,$$

где  $\theta$  – значение температуры воздуха в помещении, °С;  $k_1$  – коэффициент передачи;  $U_d$  – падение напряжения на термодатчике, В.

Таким образом, полная модель подобной теплотехнической системы и динамические свойства объекта регулирования и элементов системы описываются следующими уравнениями [1]:

$$T_0 \frac{d\theta}{dt} + \theta = \theta_k + kf \text{ - объект регулирования;}$$

$$U_d = k_1 \theta \text{ - датчик;}$$

$$T_2 \frac{d^2 \mu}{dt^2} + \frac{d\mu}{dt} = k_2 U \text{ - двигатель совместно с клапаном;}$$

$$T_3 \frac{d\theta_k}{dt} + \theta_k = k_3 \mu \text{ - калорифер;}$$

$$\Delta U = U_0 - U_d \text{ - сравнивающий орган;}$$

$$T_4 \frac{dU}{dt} + U = k_4 \Delta U - \text{магнитный усилитель,}$$

где  $T_0, T_2, T_3, T_4$  – постоянные времени, с;  $\theta$  – значение температуры воздуха в помещении, °С;  $\theta_k$  – значение температуры воздуха на выходе калорифера, °С;  $k, k_1, k_2, k_3, k_4$  – коэффициенты передачи;  $f$  – возмущающее воздействие на объекте регулирования;  $U_d$  – падение напряжения на термодатчике, которое пропорционально температуре на объекте, т.е. в помещении, В;  $\Delta U$  – напряжение на выходе мостовой схемы (сигнал рассогласования), которое, очевидно, пропорционально разнице между требуемой и имеющейся температурой в помещении, В;  $\mu$  – линейное перемещение клапана, см;  $U_0$  – задающий сигнал, т.е. уровень напряжения на мостовой схеме, который пропорционален требуемой температуре в помещении, В (например, коэффициент пропорциональности может быть установлен так, что напряжение 4В будет соответствовать 20°С).

В результате решения системы уравнений в СКМ *Maple* при нижеприведённых параметрах:  $T_0=1000$  с;  $T_2=0.06$  с;  $T_3=20$  с;  $T_4=0.5$  с;  $k=0.2$ ;  $k_1=0.2$  В/°С;  $k_2=0.002$  см/(В·с);  $k_3=10$ °С/см;  $k_4=2$ ;  $f = -11$ °С (возмущающее воздействие, т.е. температура воздуха на улице равна -11°С);  $U_0 = 4$ В (что соответствует требуемой температуре в помещении равной 20°С), был получен следующий график (рис.2) переходного процесса нагрева температуры в комнате от 5 до 20 градусов при температуре наружного воздуха -11°С.

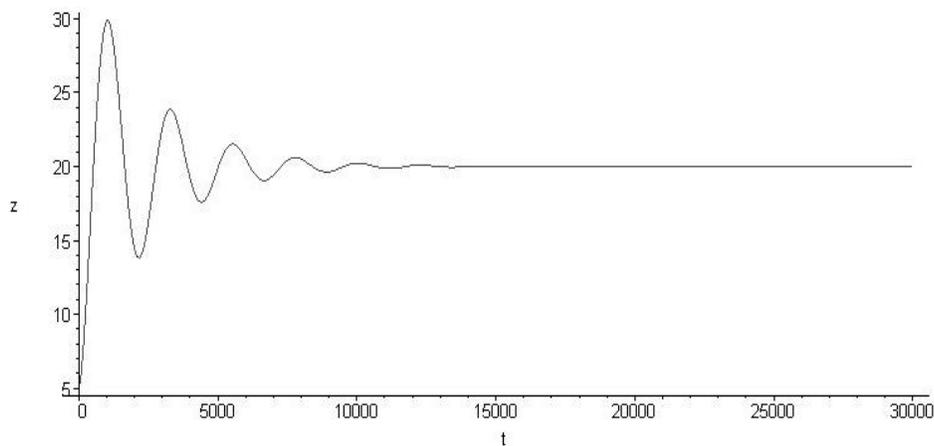


Рисунок 2. График нагрева температуры в комнате

Для этой же системы визуализированная модель в виде передаточных функций в пакете *Simulink* СКМ *MatLab* будет выглядеть следующим образом (рис.3):

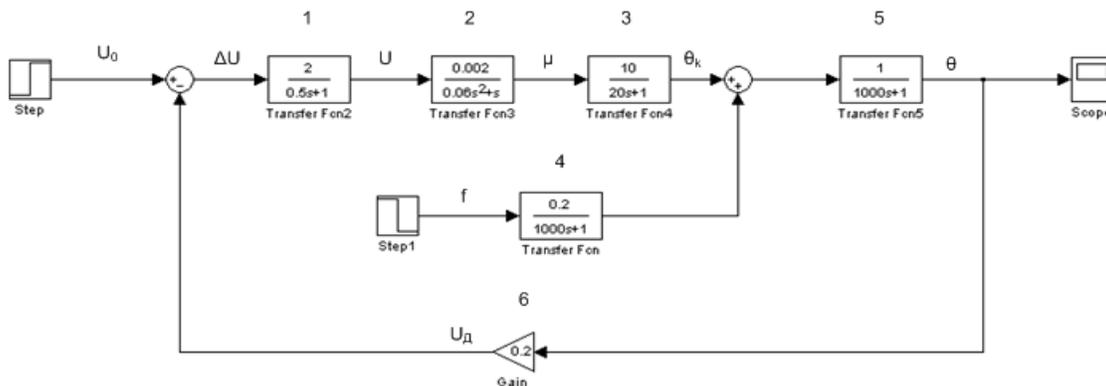


Рисунок 3. Модель ТС в пакете *Simulink*

Результаты моделирования оказались аналогичными с моделированием в СКМ *Maple*. Математическое моделирование показывает, что система управления ТС с серийными параметрами приводит её в требуемое состояние, т.е. в помещении устанавливается температура 20°C. Но при этом очевиден колебательный режим процесса, который не так просто отследить в эксплуатационном режиме. Подобное замечание ещё более актуально при существенных случайных возмущающих воздействиях.

Таким образом, наличие системы автоматики не гарантирует оптимизацию (в данном случае - минимизацию) энергопотребления. Таким образом, очевидно, что при проектировании ТС необходимо рассчитывать и по возможности оптимизировать параметры её элементов. На рис.4 показан процесс нагрева при оптимизированных параметрах системы. Критериями оптимизации графика изменения температуры на объекте регулирования (в помещении) являются время переходного процесса и величина перерегулирования. Расчет показывает, что при подобном режиме потери теплоносителя для нагрева помещения уменьшаются не менее чем на 10% по сравнению с неоптимизированной системой рассматриваемого примера.

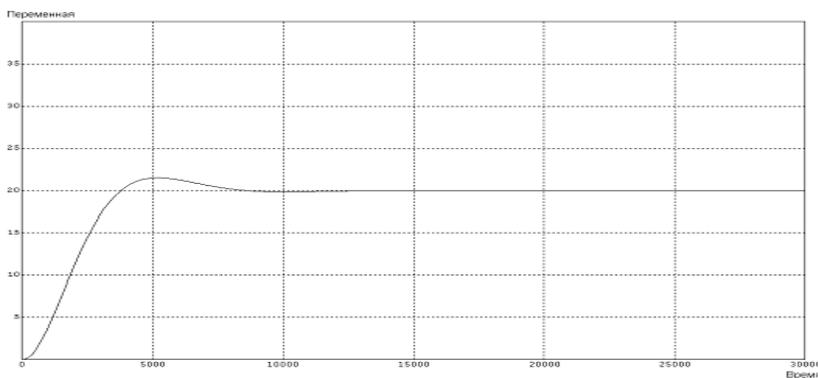


Рисунок 4. Процесс нагрева при оптимизированных параметрах системы

Еще более актуальной решение подобной проблемы оптимизации возникает при распределенных ТС. Моделирование показывает, что неоптимизированные системы автоматики не только не экономят энергоносители, но приводят к неоправданным их потерям.

Применение дискретных и цифровых систем регулирования не решают автоматически проблем неоптимального управления, при этом их некорректное применение может

даже ухудшить выходные параметры ТС. Рассмотрим пример дискретной системы регулирования температуры воздуха в помещении, заменив дифференциальный магнитный усилитель (блок 1 схемы рис.3) релейным элементом с зоной нечувствительности. Модель системы, подготовленная в пакете *MBTU*, показана на рис.5.

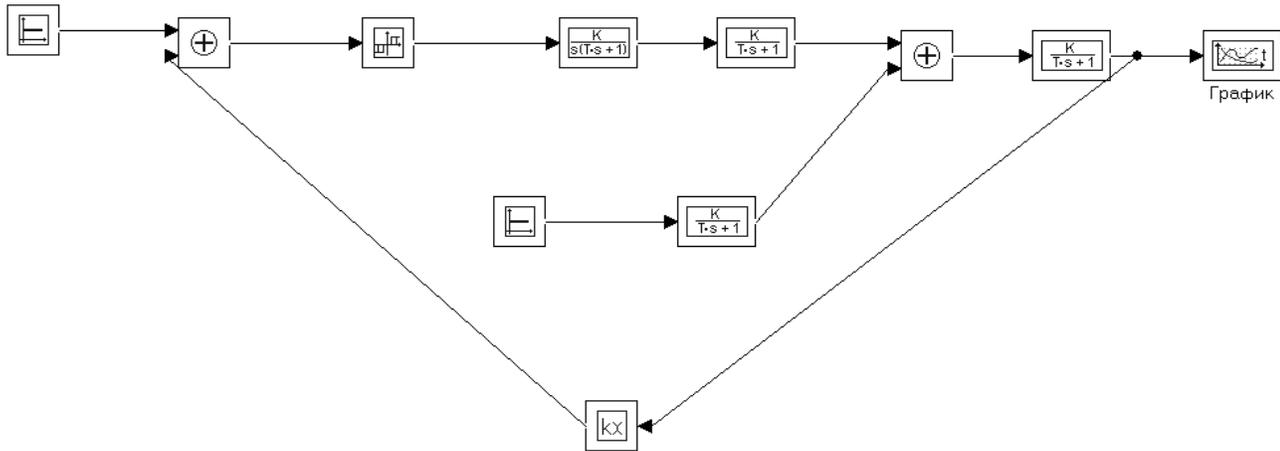


Рисунок 5. Модель ТС в пакете *MBTU*

Результат моделирования приведён на рис.6.

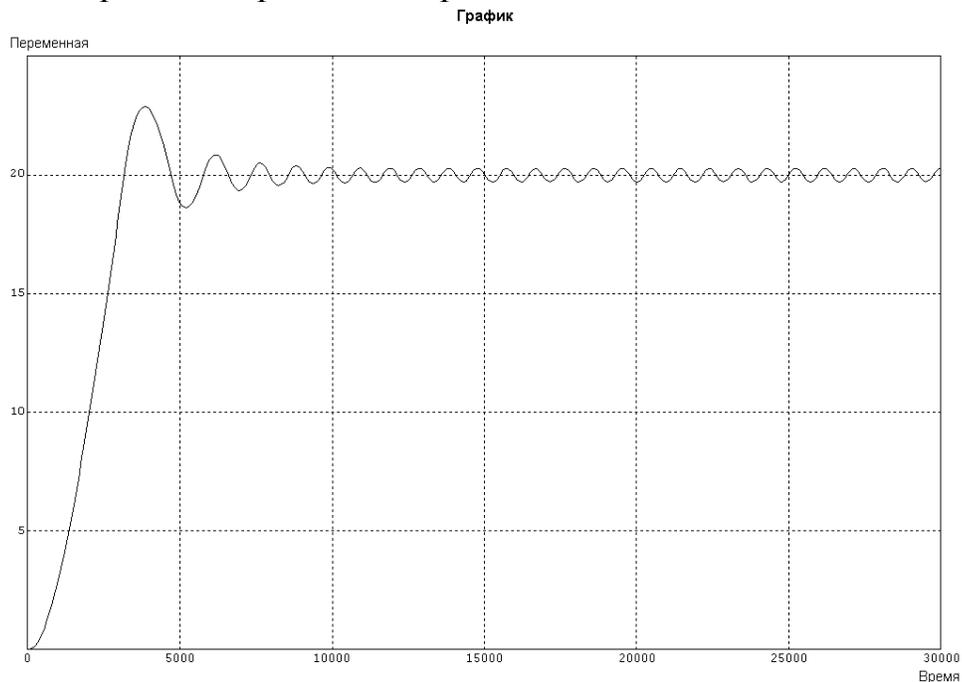


Рисунок 6. График изменения температуры в комнате при дискретной системе управления

Моделирование показывает, что дискретизация системы приводит к появлению устойчивых периодических колебаний в окрестностях установившегося выходного параметра регулирования – температуры воздуха в помещении, равной 20°C. Такие колебания, близкие гармоническим в окрестностях точки в 20°C с амплитудой, равной примерно 0.5°C, возможно, будут незаметны и несущественны для офисного помещения, но могут быть недопустимы в технологическом производственном помещении.

Заметим, что замена релейного элемента в контуре управления на сложную микропроцессорную систему приводит к улучшению параметров ТС только при корректной её настройке и программировании с учётом законов управления техническими системами. Так, например, серийные микропроцессорные контролеры управления климатическими системами зачастую позволяют их программирование, которое приводит сложную систему управления к модели релейного элемента с настраиваемыми параметрами. Эмпирический подбор параметров при натурной отладке программ требует существенных временных и материальных затрат, которых, очевидно, позволяет избежать этап математического моделирования.

Важно отметить, что в ряде случаев только тщательное математическое моделирование позволяет на ранних стадиях проектирования выявить недостатки в качестве регулирования ТС, в том числе цифровых. Пример с регулированием температуры воздуха в помещении подсказывает, что некорректная работа ТС может быть в ряде случаев незаметна в процессе эксплуатации.

## Приемы моделирования в *Simulink*

### 1 Общие сведения о пакете *Simulink*

Пакет *Simulink* является приложением к системе *MATLAB*. При моделировании с использованием *Simulink* реализуется принцип визуального программирования, в соответствии с которым, пользователь из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты.

*Simulink* является автономным пакетом *MATLAB*. Модели, созданные в *Simulink*, построены на основе внутреннего языка *MATLAB*, которые могут быть откорректированы в *MATLAB*, и данные из *MATLAB* могут быть переданы для обработки в *Simulink*. Часть входящих в состав пакетов имеет инструменты, встраиваемые в *Simulink* (например, LTI-Viewer приложения Control System Toolbox – пакета для разработки систем управления). Имеются также дополнительные библиотеки блоков для разных областей применения (например, Power System Blockset – моделирование электротехнических устройств, Digital Signal Processing Blockset – набор блоков для разработки цифровых устройств и т.д.).

При работе с *Simulink* пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков.

При моделировании пользователь может выбирать метод решения дифференциальных уравнений, а также способ изменения модельного времени (с фиксированным или переменным шагом). В ходе моделирования имеется возможность следить за процессами, происходящими в системе. Для этого используются специальные устройства наблюдения, входящие в состав библиотеки *Simulink*. Результаты моделирования могут быть представлены в виде графиков или таблиц.

Преимущество *Simulink* заключается также в том, что он позволяет пополнять библиотеки блоков с помощью подпрограмм написанных как на языке MATLAB, так и на языках C++, Fortran и Ada.

## 2 Интерфейс пакета *Simulink*

Для запуска программы необходимо предварительно запустить пакет MATLAB. После открытия основного окна программы MATLAB нужно запустить пакет *Simulink*. Это можно сделать одним из трех способов:

- Нажать кнопку (Simulink) на панели инструментов командного окна MATLAB.
- В командной строке главного окна MATLAB напечатать >>Simulink и нажать клавишу Enter на клавиатуре.
- Выполнить команду Open... в меню File и открыть файл модели (mdl - файл).

Последний вариант удобно использовать для запуска уже готовой и отлаженной модели, когда требуется лишь провести расчеты и не нужно добавлять новые блоки в модель. Использование первого и второго способов приводит к открытию окна обозревателя разделов библиотеки *Simulink* (рисунок 1).

Рисунок 1. Окно обозревателя разделов библиотеки *Simulink*

Окно обозревателя библиотеки блоков содержит следующие элементы (рисунок 1):

- Заголовок, с названием окна – Simulink Library Browser.

- Меню, с командами File, Edit, View, Help.

2

- Панель инструментов, с ярлыками наиболее часто используемых команд.

3

- Окно комментария для вывода поясняющего сообщения о выбранном блоке.

- Список разделов библиотеки, реализованный в виде дерева.

4

- Окно содержимого раздела библиотеки (список вложенных разделов библиотеки или блоков)

5

6

- Строка состояния, содержащая подсказку по выполняемому действию.

7

На рисунке 1 выделена основная библиотека Simulink (в левой части окна) и показаны ее разделы (в правой части окна).

Библиотека Simulink содержит следующие основные разделы:

Continuous – линейные блоки.

Discrete – дискретные блоки.

Functions & Tables – функции и таблицы.

Math – блоки математических операций.

Nonlinear – нелинейные блоки.

Signals & Systems – сигналы и системы.

Sinks - регистрирующие устройства.

Sources — источники сигналов и воздействий.

Subsystems – блоки подсистем.

При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое (рисунок 2).

### 3 Создание и редактирование модели в Simulink

Для создания модели в среде *SIMULINK* необходимо последовательно выполнить ряд действий.

Создание файла модели. Создать новый файл модели с помощью команды File/New/Model, или используя кнопку на панели инструментов.

Выбор блоков модели. Расположить блоки в окне модели. Для этого необходимо открыть соответствующий раздел библиотеки (Например, Sources - Источники). Далее, указав курсором на требуемый блок и нажав на левую клавишу “мыши” - “перетащить” блок в созданное окно. *Клавишу мыши нужно держать нажатой.* На рисунке 3 показано окно модели, содержащее блоки.

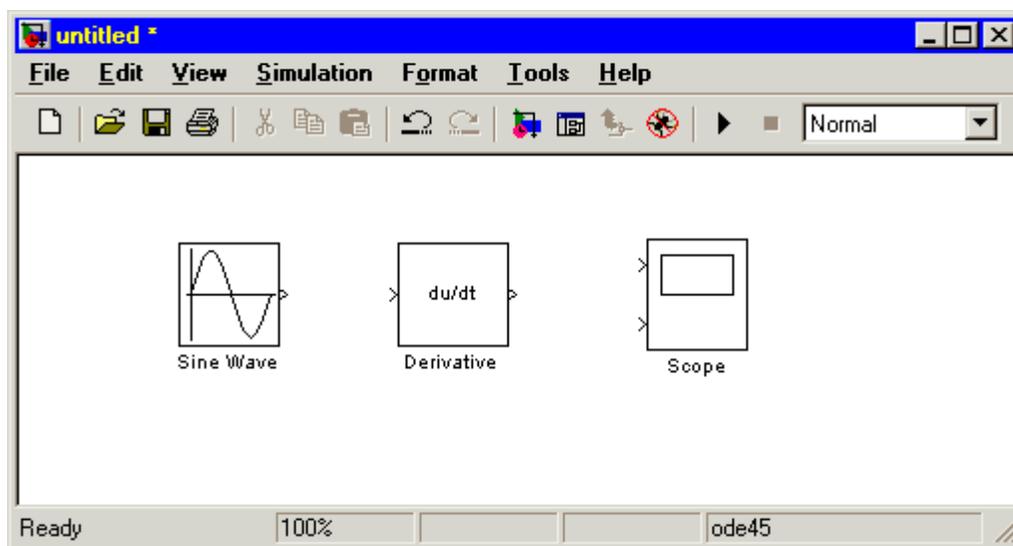


Рисунок 3. Окно модели, содержащее блоки

Редактирование блоков модели. Для удаления блока необходимо выбрать блок (указать курсором на его изображение и нажать левую клавишу “мыши”), а затем нажать клавишу Delete на клавиатуре.

Для изменения размеров блока требуется выбрать блок, установить курсор в один из углов блока и, нажав левую клавишу “мыши”, изменить размер блока (курсор при этом превратится в двухстороннюю стрелку).

Для редактирования блока, нужно изменить параметры блока, установленные пакетом “по умолчанию”. Для этого необходимо дважды щелкнуть левой клавишей “мыши”, указав курсором на изображение блока. Откроется окно редактирования параметров данного блока. При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая. После внесения изменений нужно закрыть окно кнопкой ОК. На рисунке 4 в качестве примера показаны блок, моделирующий синусоидальный сигнал с заданной частотой, амплитудой, фазой и смещением, и окно редактирования параметров данного блока.

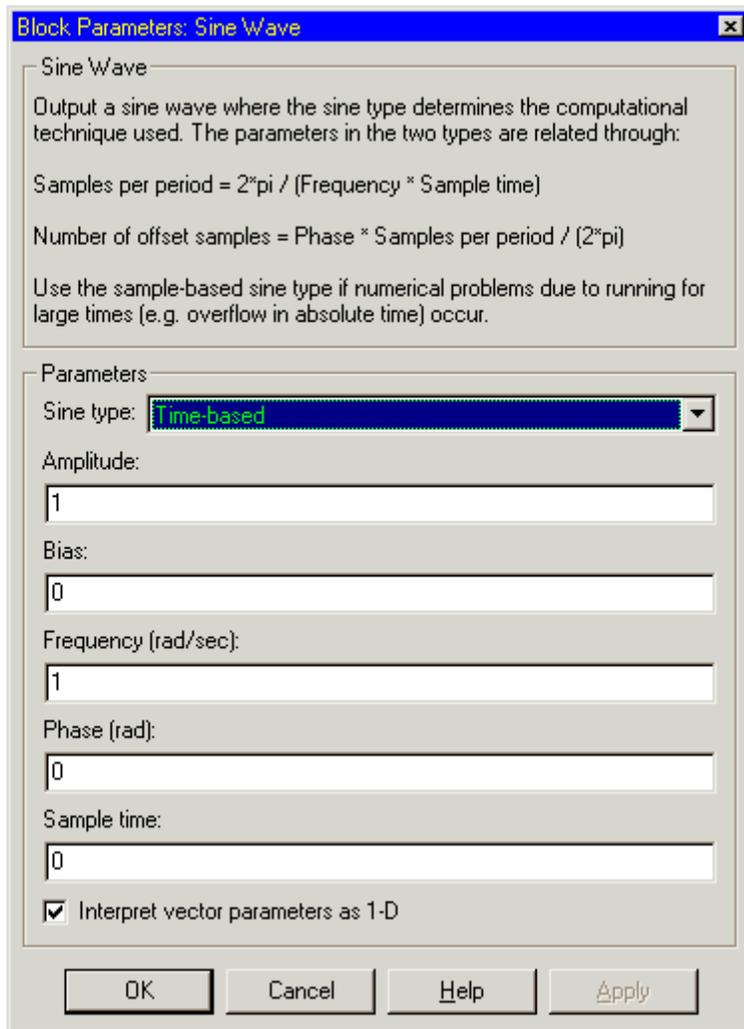
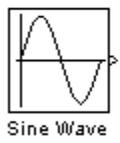


Рисунок 4. Блок, моделирующий синусоидальный сигнал с заданной частотой, амплитудой, фазой и смещением, и окно редактирования параметров блока

Соединение блоков модели. После установки на схеме всех блоков из требуемых библиотек нужно выполнить соединение элементов схемы. Для соединения блоков необходимо указать курсором на “выход” блока, а затем, нажав и, не отпуская левую клавишу “мыши”, провести линию к входу другого блока. После чего отпустить клавишу. В случае правильного соединения изображение стрелки на входе блока изменяет цвет. Для создания точки разветвления в соединительной линии нужно подвести курсор к предполагаемому узлу и, нажав *правую* клавишу “мыши”, протянуть линию. Для удаления линии требуется выбрать линию (так же, как это выполняется для блока), а затем нажать клавишу Delete на клавиатуре. Схема модели, в которой выполнены соединения между блоками, показана на рисунке 5.

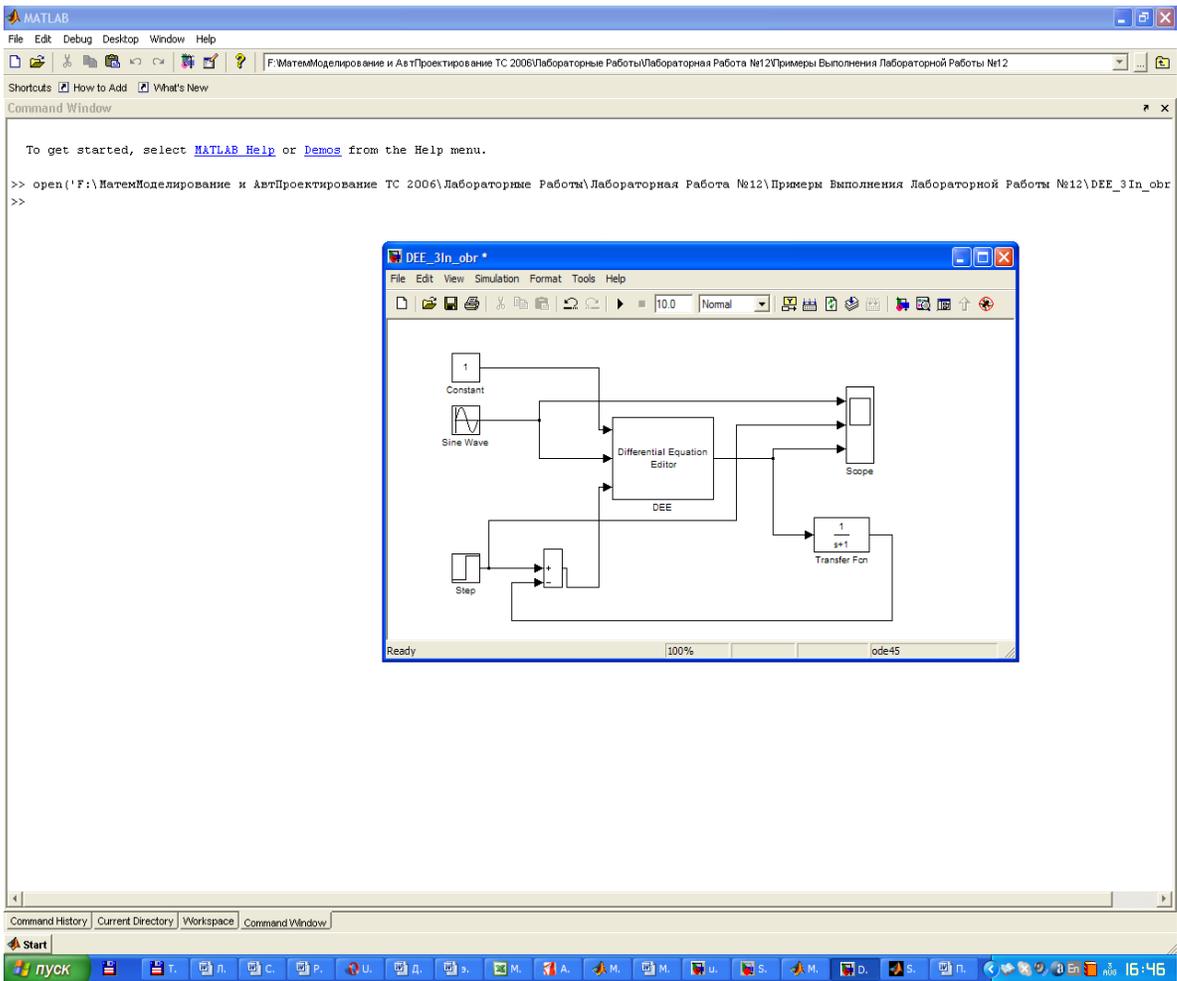
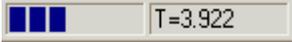


Рисунок 5. Схема модели

#### 4 Запуск модели, анализ результатов

Запуск модели на выполнение осуществляется нажатием на панели инструментов кнопки  (Start simulation). При этом в строке состояния демонстрируется текущее время расчета и процентное состояние расчета 

Получение результатов модели сводится к записи в рабочую область MATLAB - Workspace данных в численном виде, или отображение исходных данных и результата в виде графика рисунок 6.

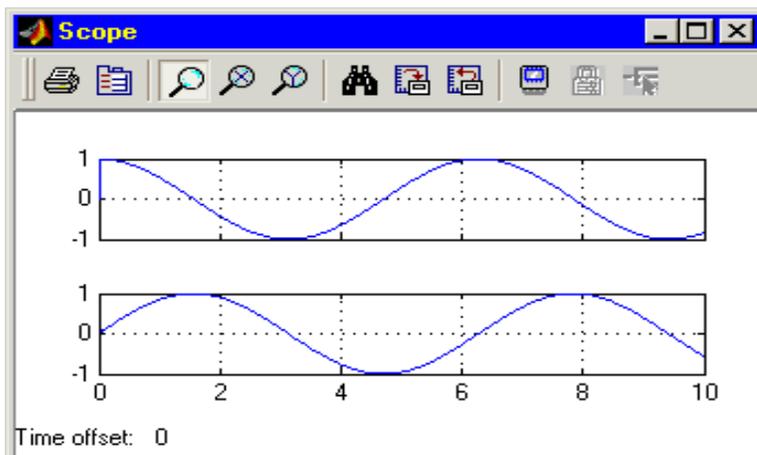
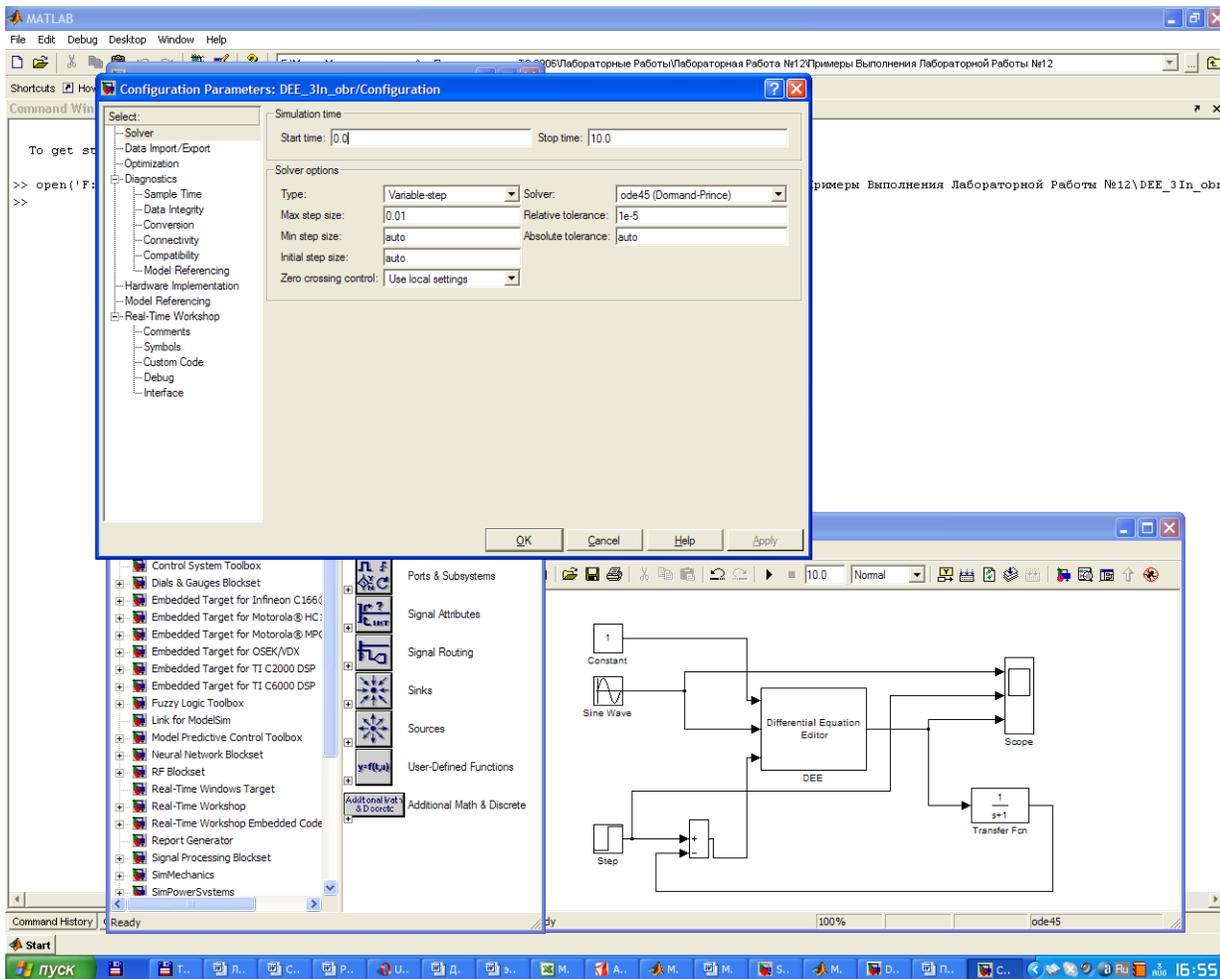


Рисунок 6 – Графическое отображение исходных данных и результата

Опцией *Simulation* -> *Configuration Parameters* вызывается окно настройки параметров моделирования, в котором можно указать время начала и конца моделирования, минимальный и максимальный шаг интегрирования, требуемая точность, метод интегрирования и ряд других параметров:



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

## Элементы работы с ЛТИ-объектами

## 1 Постановка задачи

В качестве объектов исследования могут быть линейные (линеаризованные) динамические стационарные системы управления с одним входом и одним выходом. При этом модель одномерной САУ задана в виде комплексной передаточной функции, записанной как отношение полиномов

$$W(s) = \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0}.$$

MatLab позволяет:

1. Определить полюса и нули передаточной функции

$$s_j^*, (j = \overline{1, n}), \quad s_j^0, (j = \overline{1, m}).$$

2. Записать дифференциальное уравнение, определяющее функционирование САУ.
3. Построить графики переходной и импульсной характеристик:

$$h(t), w(t).$$

4. Построить логарифмические частотные характеристики

$$L(\omega).$$

5. Построить частотный годограф Найквиста

$$W(i\omega), \omega = [0, \infty].$$

6. Построить карту нулей и полюсов передаточной функции (при выполнении анализа в режиме *LTIVIEW*).

## 2 Сведения по работе с ППП *Control System Toolbox*

ППП предназначен для работы с *LTI*-моделями (Linear Time Invariant Models – линейными инвариантными моделями) технических систем и систем управления.

*MatLab* позволяет выполнить требуемый анализ несколькими способами:

- в интерактивном режиме, задавая команды ППП в командном режиме из *Command Window* ;
- в программном режиме, организовав *m*-файл или пользовательский *GUI*-интерфейс;
- использование графического интерфейса ППП *CST – LTI viewer*;
- в режиме визуального моделирования пакета *Simulink* , создав файл моделирования *Model*.

В *Control System Toolbox* имеется тип данных, определяющих динамическую систему в виде комплексной передаточной функции. Синтаксис команды *tf*, создающий в интерактивном режиме из *Command Window* *LTI*-систему с одним входом и одним выходом в виде передаточной функции:

```
>> W = tf([bm, ..., b1, b0], [an, ..., a1, a0])
```

$b_m, \dots, b_1$  – значения коэффициентов полинома  $B$  числителя передаточной функции,

$a_n, \dots, a_1$  – значения коэффициентов полинома  $A$  знаменателя.

В переменную  $W$  записывается *LTI*-система в виде передаточной функции. В окне вывода *Workspace* можно посмотреть созданную переменную.

Для выполнения работы необходимо применяться команды, приведенные в таблице 1.1.

Таблица 1.1. Некоторые команды *Control System Toolbox*

Синтаксис	Описание
<code>pole(&lt;LTI-объект&gt;)</code>	Вычисление полюсов передаточной функции
<code>zero(&lt;LTI-объект&gt;)</code>	Вычисление нулей передаточной функции
<code>step(&lt;LTI-объект&gt;)</code>	Построение графика переходной характеристики
<code>impulse(&lt;LTI-объект&gt;)</code>	Построение графика импульсной характеристики
<code>bode(&lt;LTI-объект&gt;)</code>	Построение логарифмических частотных характеристик (диаграммы Боде)
<code>nyquist(&lt;LTI-объект&gt;)</code>	Построение частотного годографа Найквиста

Пример применения команды *step*:

```
>> step(W)
```

Для определения корней полиномов степени  $k$ , может, также, применяться команда *MATLAB*

```
roots(P),
```

которая, в качестве аргумента  $P$ , получает матрицу коэффициентов полинома  $[p_k, \dots, p_0]$ .

Приведённые выше команды могут быть организованы в программный фрагмент, записаны в *m*-файл и затем запускаться на выполнение как программа.

Использование графического интерфейса ППП *CST – LTI viewer* осуществляется вызовом команды из *Command Window*:

```
>> ltiview
```

В качестве параметра команды можно указать имя переменной, содержащей **LTI**-объект, например >>ltiview (W)

В результате выполнения команды на экран дисплея выводится интерактивное окно анализа характеристик ТС. Работа с ним вполне очевидна. Отметим возможность задания опцией *Edit – Plot configuration* конфигурации выводимых окон и анализируемых параметров в каждом из них.

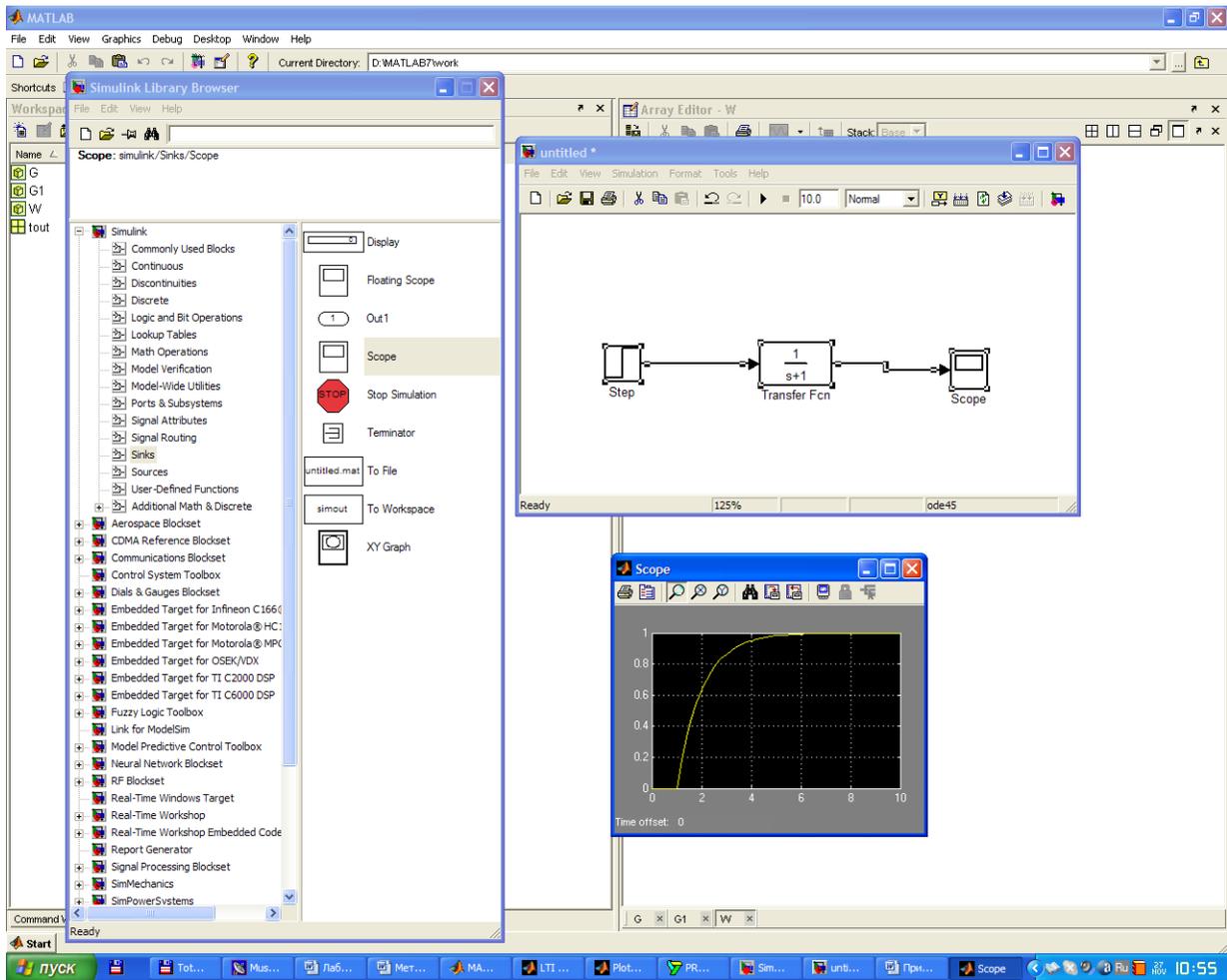
Для выполнения анализа в *Simulink* рекомендуется предварительно создать файл моделирования *Model*, затем вызвать *Simulink*. Описание работы с *Simulink* приведено в файле «ПРИЕМЫ МОДЕЛИРОВАНИЯ В SIMULINK», здесь отметим некоторые графические блоки, которые понадобятся для выполнения работы, и библиотеки, в которых они расположены:

1. Блоки задания передаточных функций *Transfer Fnc*, а также интегрирования и дифференцирования функций находятся в библиотеке *Continuous*.
2. Блок задания функции Хевисайда *Step* расположен в библиотеке *Sources*.
3. Блок визуализации результатов моделирования *Scope* – библиотека *Sinks*.

Параметры моделирования устанавливаются опциями *Simulation – Configuration Parameters* окна моделирования.

Требуемые блоки устанавливаются переносом их мышки в окно моделирования, соединяются между собой, затем устанавливаются параметры блоков и моделирования. Вызов окна задания параметров блока осуществляется двойным щелчком мыши на его изображении.

После запуска на моделирование, если система не выдала ошибок, двойным щелчком мыши на блок *Scope* вызывается график моделируемой кривой:



Рассмотрим работу с ППП на примере.

Задана передаточная функция САУ

$$W = \frac{s + 2}{3s^3 + 4s^2 + 5s + 3}.$$

Найдем ее динамические и частотные характеристики. Пример приведён для варианта выполнения задания в командном режиме.

1. Создадим ЛТИ-объект с именем w, для этого выполним:

```
>> w=tf([1 2],[3 4 5 3])
```

```
Transfer function:
```

```
  s + 2
```

```
-----
```

```
3 s^3 + 4 s^2 + 5 s + 3
```

2. Найдем полюса и нули передаточной функции с использованием команд pole, zero.

```
>> pole(w)

ans =

-0.2639 + 1.0825i
-0.2639 - 1.0825i
-0.8055

>> zero(w)

ans =

-2
```

3. Построим переходную функцию командой `step(w)`. Результат ее выполнения приведен на рис. 1.1.

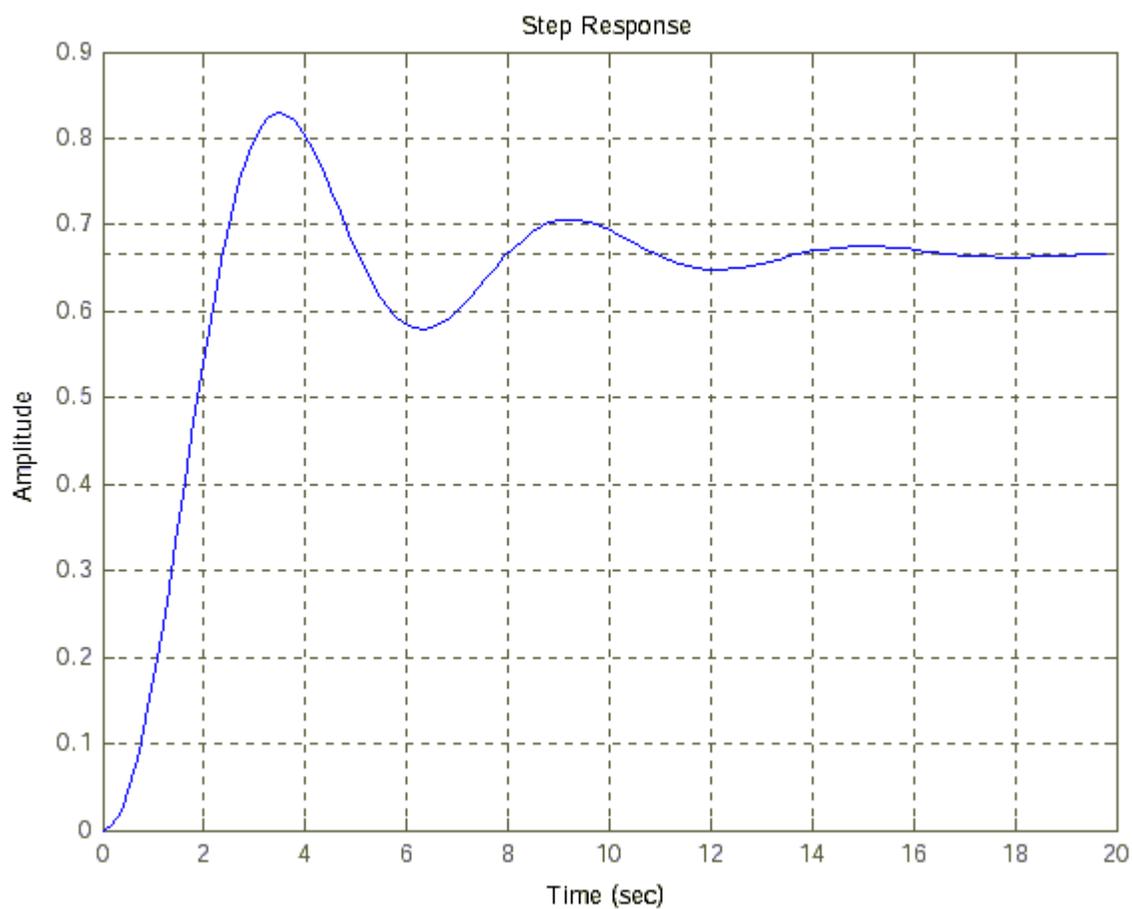
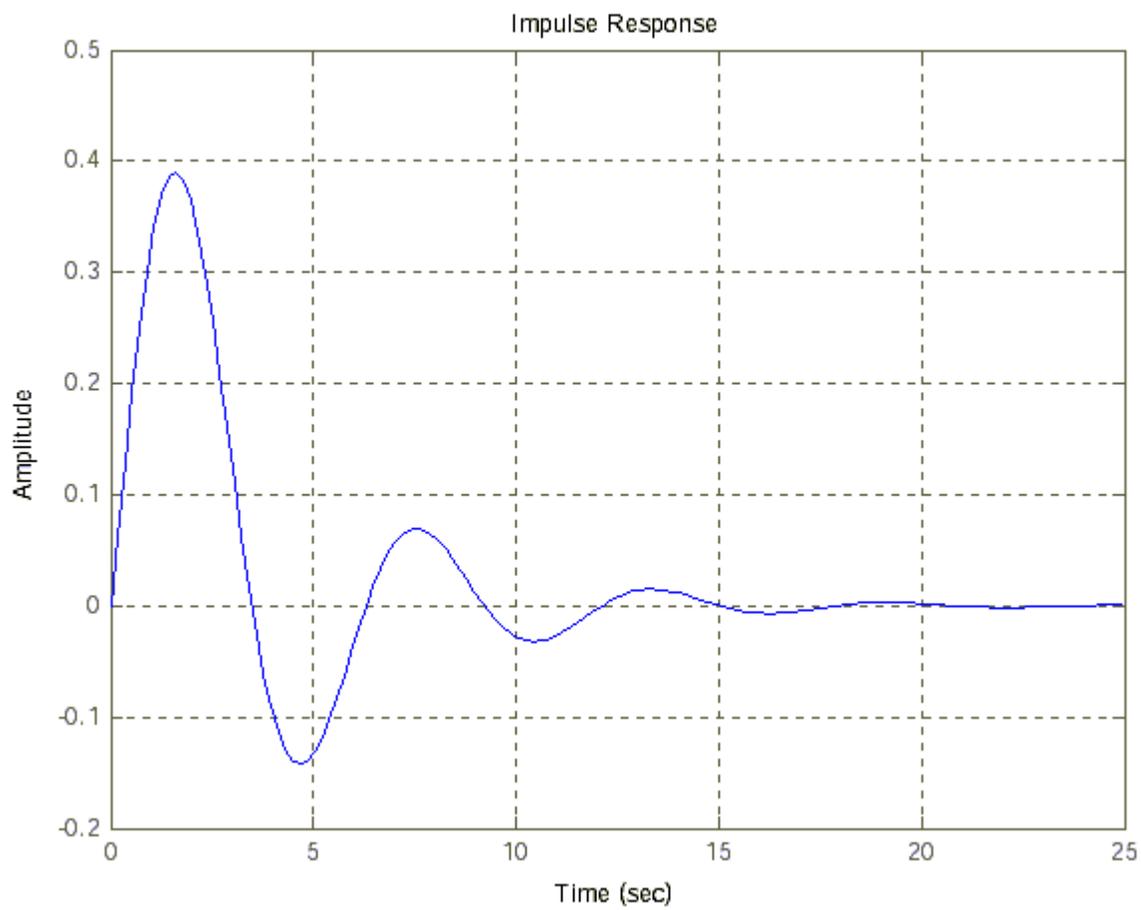


Рис. 1.1. Переходная характеристика  $h(t)$ .

4. Построим импульсную функцию командой `impulse(w)`. Результат показан на рис. 1.2.



*Рис. 1.2. Импульсная характеристика.*

5. Диаграмму Боде получим, используя команду `bode(w)` – рис. 1.3.

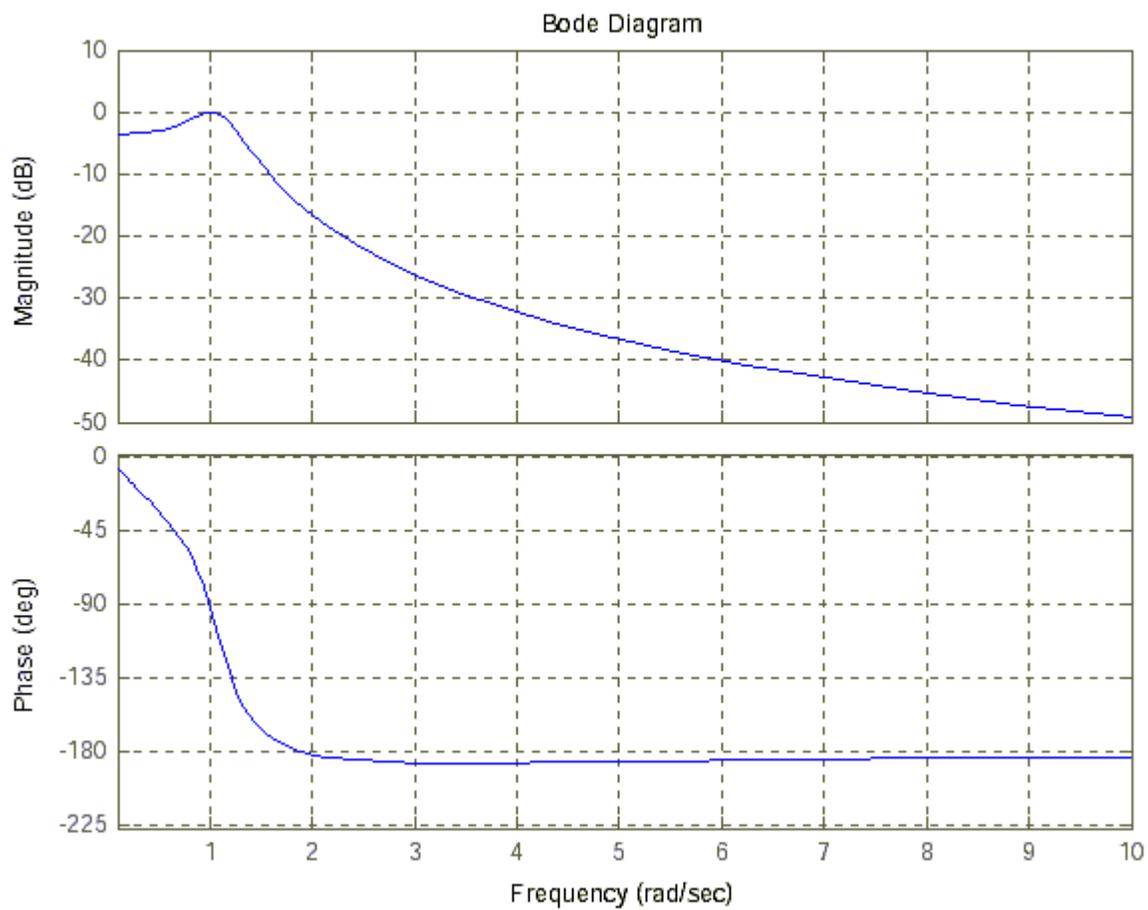
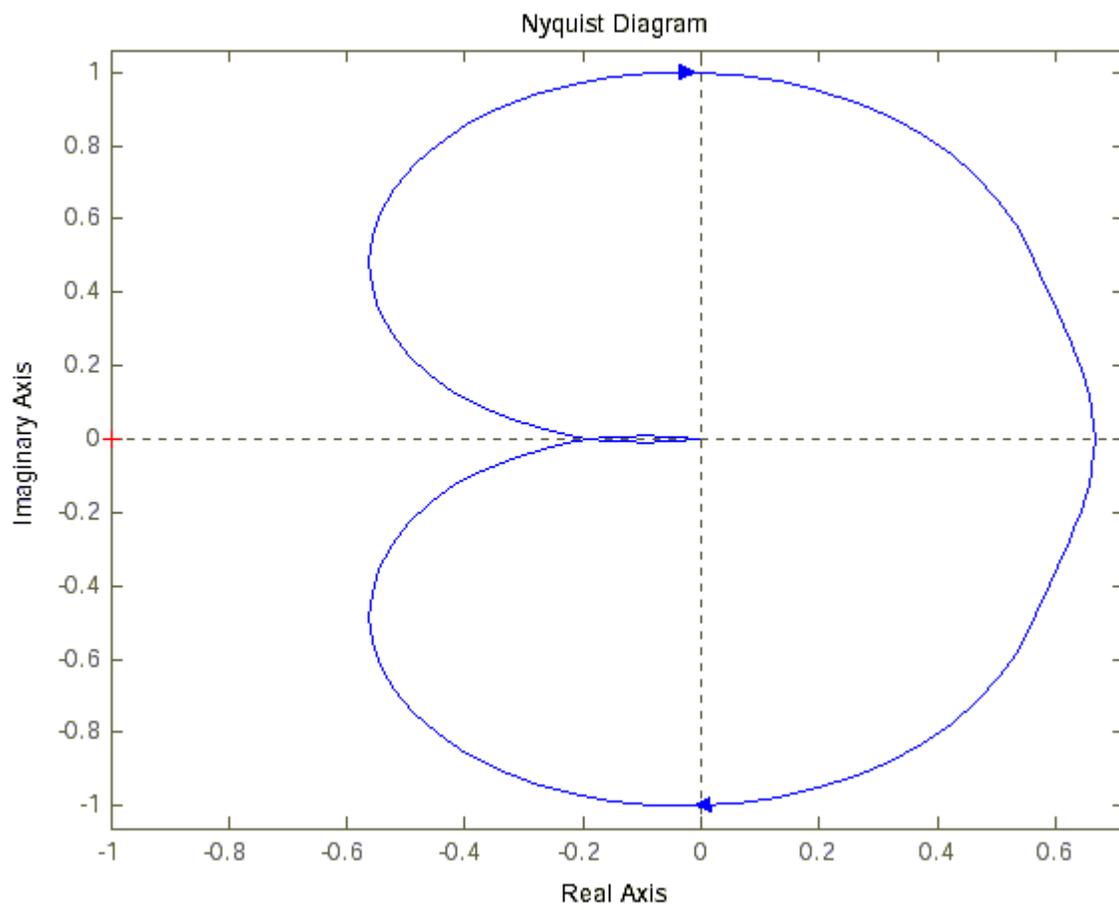


Рис. 1.3. Логарифмические частотные характеристики.

6. Определим частотный годограф Найквиста, выполнив команду `nyquist(w)` – рис. 1.4.



*Рис. 1.4. Частотный годограф.*

Аналогичные результаты (рис. 1.5) можно получить, используя команду `ltiview(w)`, с соответствующими настройками в меню "Plot Configuration".

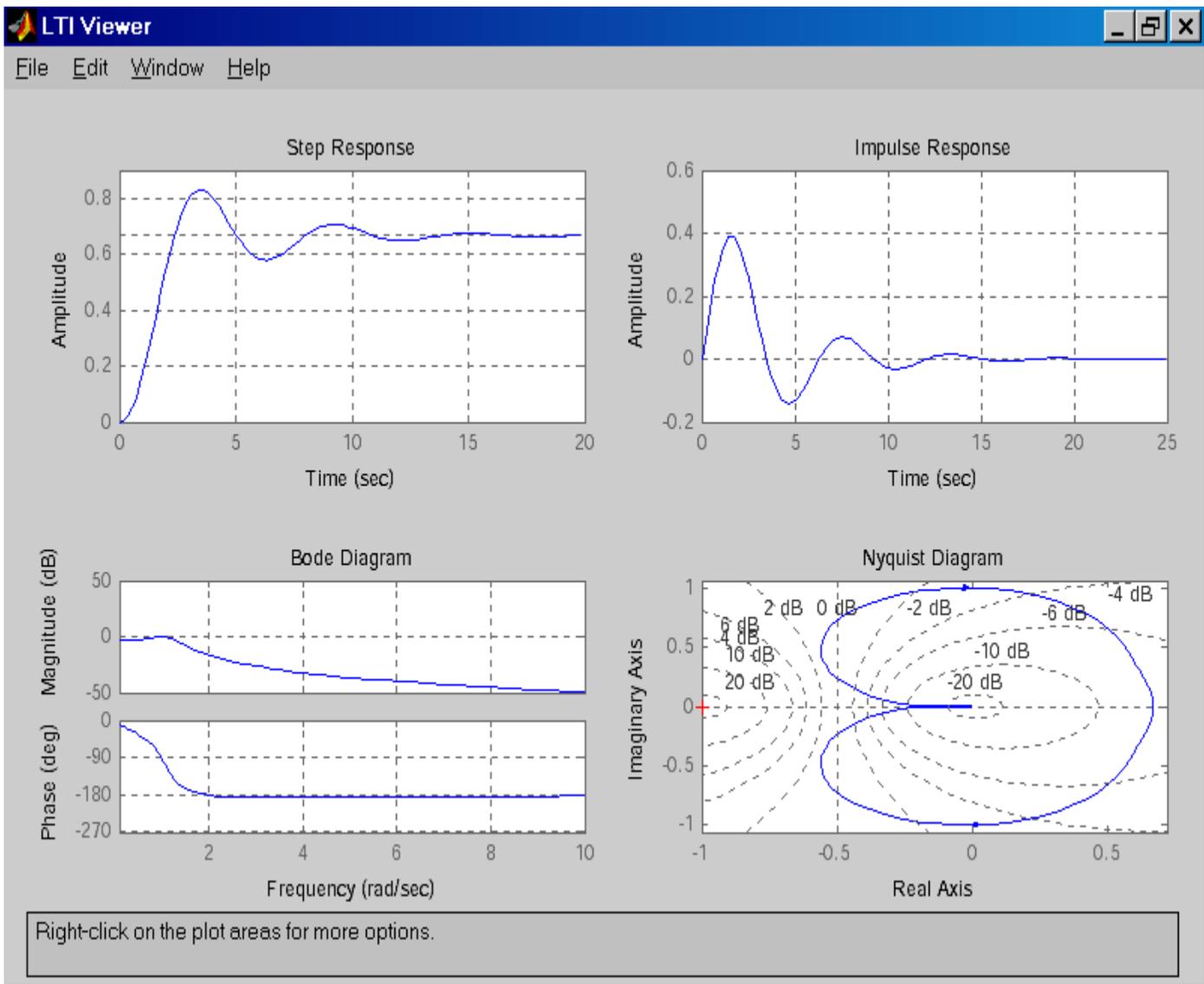


Рис. 1.5. LTI-viewer.

Каждая из построенных характеристик полностью и однозначно определяет рассматриваемую систему управления.

## Лабораторная работа № 1

### ***Автоматизация моделирования и оптимизация линейных стационарных систем автоматического регулирования технических объектов***

Цель работы: *Овладеть навыками автоматизации математического моделирования систем автоматического управления технических объектов, а также принципами оптимизации параметров систем*

1. Теоретические сведения к работе представлены в поддиректории «Учебно-методические Материалы к Лабораторной Работе №1» настоящей лабораторной работы.
2. По принципиальной схеме системы автоматического регулирования (САР) и объекта управления подготовить её структурную схему, на которой необходимо указать все функциональные элементы системы с входными и выходными сигналами, а также управляющие и возмущающие воздействия.
3. По системе алгебро-дифференциальных уравнений получить в символьном виде:
  - а) передаточные функции каждого элемента САР;
  - б) передаточную функцию всей системы по управляющему воздействию.
4. Рассчитать корни характеристического полинома. Оценить устойчивость САР. Все вычисления оформить в виде m-файла.
5. Получить переходную, импульсную и частотные характеристики САР, а также построить её годограф, воспользовавшись модулем *lti* СКМ *MatLab*.
6. В пакете *Simulink* системы компьютерной математики (СКМ) *MatLab* подготовить визуализированную модель САР технической системы. Первоначальные параметры системы задать в соответствии с указанным вариантом.
7. Выполнить моделирование работы САР при ступенчатом изменении управляющего и возмущающего воздействий. Пределы изменения воздействий выбрать самостоятельно и обосновать выбор.
8. Самостоятельно задать произвольные функции изменения управляющего и возмущающего воздействий, отличные от ступенчатой, и выполнить моделирование работы САР.
9. Изменяя коэффициенты усиления элементов САР (в первую очередь блоков усиления сигнала рассогласования и цепи обратной связи) оптимизировать параметры САР, т.е. подобрать параметры системы, при которых в системе будет происходить:
  - а) колебательный затухающий процесс;
  - б) затухающий процесс с одним перерегулированием.
 Проследить, как при этом изменяется расположение нулей и полюсов (корни характеристического полинома) на карте нулей и полюсов. Использовать функции расчёта корней полинома и модуль *lti* СКМ *MatLab*.
10. Определить параметры процесса регулирования для неоптимизированной и оптимизированной систем.
- 11. Выполнить факультативно.** *Определить параметры САР, при которых система находится на границе устойчивости.*
12. Подготовить отчёт по работе. Отчёт должен включать исходную схему и описание её работы; систему алгебро-дифференциальных уравнений, моделирующую работу САР и объекта управления; передаточные функции элементов и всей системы управления; результаты моделирования и характеристики САР, полученные в ходе выполнения лабораторной работы; оптимизированные параметры САР; обоснованные выводы по работе.

## Лабораторная работа № 2

### Параметрическая оптимизация линейной САР

**Цель работы:** Овладеть навыками параметрической оптимизации систем автоматического управления техническими объектами, выполненными на основе ПИ- и ПИД-регуляторов.

#### Ход выполнения работы

1. В *Simulink*-модели САР, полученной в ходе выполнения лабораторной работы № 1, заменить звенья управляющей цепи (от сумматора цепи обратной связи до объекта управления) ПИ-регулятором.
2. В окне *Workspace* объявить 2 переменные  $k_{prop}$  и  $k_{int}$ , которые будут определять коэффициенты пропорционального и интегрального звеньев ПИ-регулятора. Начальные значения этих переменных задать равными 1.
3. В *Simulink*-модели в поле коэффициента усиления усилителя пропорционального звена указать имя переменной  $k_{prop}$ , а в соответствующем поле усилителя интегрального звена – имя переменной  $k_{int}$ .
4. Величину возмущающего воздействия задать равной 0.
5. К выходу САР подключить компонент *Check Custom Bounds* из библиотеки *Simulink Design Optimization*.
6. В окне параметров блока задать границы изменения выходного сигнала САР. Точность поддержания заданного выходного значения принять равной 1 %. При задании границ учесть результаты моделирования САР, полученные в ходе выполнения лабораторной работы № 1.
7. В качестве переменных, значение которых подлежит изменению в процессе оптимизации, указать переменные  $k_{prop}$  и  $k_{int}$ .
8. Запустить процесс оптимизации. По достижении результатов, удовлетворяющих заданным ограничениям, задать более жесткие границы и повторить процесс оптимизации.
9. Модель САР с ПИ-регулятором, а также результаты моделирования и конечное значение переменных  $k_{prop}$  и  $k_{int}$  поместить в отчет.
10. Рассчитать параметры процесса регулирования и сравнить их с параметрами, полученными в лабораторной работе № 1.
11. Добавить в состав регулятора дифференцирующее звено таким образом, чтобы получился ПИД-регулятор.

12. В окне *Workspace* объявить переменную  $k_{diff}$  с начальным значением 1. Имя этой переменной добавить в поле коэффициента усиления усилителя дифференцирующего звена, а также добавить ее в список переменных, подлежащих изменению в процессе оптимизации.

13. Выполнить п. 8 для САР с ПИД-регулятором.

14. Модель САР с ПИД-регулятором, а также результаты моделирования и конечное значение переменных  $k_{prop}$ ,  $k_{int}$  и  $k_{diff}$  поместить в отчет.

15. Рассчитать параметры процесса регулирования и сравнить их с соответствующими параметрами ПИ-регулятора.

16. Выбрать регулятор, обладающий лучшими параметрами регулирования. В случае равенства параметров отдать предпочтение ПИ-регулятору как более простому в реализации.

17. Сделать выводы по работе.

### Лабораторная работа № 3

#### *Автоматизация моделирования цифровых стационарных систем автоматического регулирования технических объектов (часть 1)*

**Цель работы:** *Овладеть навыками автоматизации математического моделирования цифровых систем автоматического управления технических объектов.*

#### **Ход выполнения работы**

Для выполнения данной лабораторной работы используется модель САР, полученная в ходе выполнения предыдущей работы.

1. При помощи компонента *Scope* определить диапазон изменения сигналов:

- на выходе источника задающего воздействия (положительный вход сумматора цепи ОС);
- обратной связи (отрицательный вход сумматора цепи ОС);
- на входе объекта управления по задающему воздействию.

2. В разрыв указанных в п. 1 цепей включить последовательно блоки *Zero Order Hold* и *Quantizer*.

3. Используя результаты расчетов параметров процесса регулирования, полученные в предыдущей лабораторной работе, выполнить оценку периода дискретизации системы. Если процесс регулирования периодический с несколькими перерегулированиями, то период дискретизации следует принимать не более половины периода колебаний. Иначе, если процесс регулирования носит апериодический характер, либо имеет одно перерегулирование, период дискретизации следует принимать не более половины времени регулирования. Выбранное значение периода дискретизации задать для всех блоков *Zero Order Hold*.

4. По результатам измерений, полученным в п. 1, рассчитать величины шагов квантования для всех блоков *Quantizer* по следующей формуле:

$$\delta = \frac{1,02(X_{\max} - X_{\min})}{2^N},$$

где  $X_{\max}$ ,  $X_{\min}$  – максимальное и минимальное значения сигнала в соответствующей точке;  $N$  – предполагаемая разрядность АЦП/ЦАП. Для АЦП принять разрядность 10 бит, а для ЦАП – 8 бит. Рассчитанные значения  $\delta$  задать для соответствующих блоков *Quantizer*.

5. Преобразовать интегрирующее и дифференцирующее звенья ПИД-регулятора (либо только интегрирующее звено, если в системе используется ПИ-регулятор) в дискретную форму.

6. Выполнить моделирование полученной дискретной системы. Если процесс регулирования значительно отличается от процесса регулирования в непрерывной системе, необходимо уменьшить период дискретизации системы, выполнить пересчет коэффициентов ПИД(ПИ)-регулятора и повторить моделирование.

7. Рассчитать параметры процесса регулирования и сравнить их с соответствующими параметрами непрерывной системы.

Результаты, полученные в каждом из пунктов поместить в отчет.

## Лабораторная работа № 4

### Моделирование непрерывной САУ, содержащей ПИ(ПИД)-регулятор, в пакете Proteus

*Цель работы:* Освоить принципы создания и моделирования электрических схем в пакете Proteus. Выполнить моделирование непрерывной системы автоматического управления, содержащей ПИ(ПИД)-регулятор.

#### Порядок выполнения лабораторной работы

1. Освоить принципы создания электрических схем в редакторе Isis пакета Proteus.
2. Изучить компоненты библиотеки Laplace Primitives.
3. Из компонентов библиотеки Laplace Primitives составить схему САУ, структура и параметры которой получены в ходе выполнения лабораторной работы № 2. Для формирования задающего и возмущающего воздействий использовать источник сигнала PULSE, находящийся на вкладке GENERATORS.
4. Выполнить моделирование полученной системы. Для отображения результатов моделирования использовать компонент ANALOGUE, находящийся на вкладке GRAPHS.
5. Сравнить результаты моделирования с аналогичными результатами из лабораторной работы № 2.
6. Сделать выводы по работе.

#### Содержание отчета

1. Схема САУ, полученная в ходе выполнения лабораторной работы № 2 и результаты ее моделирования.
2. Схема САУ, построенная в пакете Proteus и результаты ее моделирования.

3. Сравнительная характеристика результатов, полученных в данной лабораторной работе и лабораторной работе № 2.

4. Выводы по работе.

Архив должен содержать файл отчета (\*.doc) и файлы проекта.

## Лабораторная работа № 5

### Разработка управляющего программного обеспечения ПИ(ПИД)-регулятора для микроконтроллера на языке С

*Цель работы:* Освоить принципы программирования микроконтроллеров на языке С. Реализовать управляющее программное обеспечение ПИ(ПИД)-регулятора.

#### Порядок выполнения работы:

1. Выбрать микроконтроллер, на котором будет реализовываться ПИ(ПИД)-регулятор. Контроллер должен удовлетворять следующим требованиям:

- модель контроллера должна присутствовать в пакете Proteus;
- содержать в своем составе аппаратный умножитель;
- иметь модуль АЦП выбранной в ходе выполнения предыдущей лабораторной разрядности;
- обеспечивать возможность подключения внешнего ЦАП;
- иметь количество линий ввода/вывода, достаточное для параллельного ввода сигнала задающего воздействия;
- должен содержать модуль USART (UART).

2. Разработать алгоритм функционирования программного обеспечения ПИ(ПИД)-регулятора.

3. На языке С для выбранного контроллера реализовать управляющее программное обеспечение ПИ(ПИД)-регулятора. ПО должно выполнять следующие функции:

- выполнять дискретизацию аналогового сигнала обратной связи с помощью модуля АЦП;
- выполнять дискретизацию цифрового сигнала задающего воздействия, представленного в параллельном коде;
- вычислять сигнал ошибки (разность сигналов задающего воздействия и обратной связи);
- вычислять величину управляющего сигнала, поступающего на внешний ЦАП, по схеме, полученной в ходе выполнения предыдущей лабораторной работы;
- выдавать управляющий сигнал на внешний ЦАП.

При реализации ПО необходимо максимально использовать аппаратные возможности выбранного МК.

## Лабораторная работа №6

### Моделирование работы микропроцессорной САР в пакете Proteus

*Цель работы:* Изучить принципы моделирования смешанных аналогово-цифровых систем, содержащих микроконтроллеры, в пакете Proteus.

Для выполнения лабораторной работы необходимо программное обеспечение микроконтроллера, написанное в ходе лабораторной работы № 4, а также схема САР из лабораторной работы №5.

1. На основе схемы из лабораторной работы № 5 построить микропроцессорную САР. Для этого выполнить следующее:

- в схему добавить выбранный ранее микроконтроллер и ЦАП;

- на аналоговый канал микроконтроллера подать предварительно промасштабированный сигнал обратной связи. Этот сигнал должен быть смещен в положительную сторону относительно нулевого уровня на величину, равную половине опорного напряжения АЦП;

- между выходом управляющего сигнала микроконтроллера и входом объекта управления включить ЦАП выбранной ранее разрядности. Сигнал на выходе ЦАП должен быть сдвинут в отрицательную область на величину, равную половине опорного напряжения ЦАП и промасштабирован.

- на входные линии задающего воздействия подать цифровой код заданной разрядности. Код формировать отдельными источниками цифрового сигнала.

2. Задать необходимые параметры микроконтроллера и выполнить загрузку программы из файла с расширением .cof.

3. Выполнить моделирование работы САР. Сравнить полученные результаты с результатами моделирования САР в лабораторной работе № 3.

4. Сделать выводы по работе.

## Лабораторная работа № 7

### *Разработка протокола обмена данными с микропроцессорной системой управления*

*Цель:* Овладеть навыками разработки протоколов обмена данными, а также освоить принципы реализации программ взаимодействия микроконтроллеров по последовательному каналу связи.

Для выполнения лабораторной работы используется система, полученная в ходе лабораторной работы №7. Дополнительные данные берутся из таблицы в соответствии с вариантом.

#### Порядок выполнения работы

1. Разработать ASCII протокол обмена данными между микроконтроллером систему управления и терминалом. Посылка от терминала к контроллеру должна иметь следующий формат:

[**признак начала посылки**]  
[**команда**]  
<[**блок данных**]>  
[**контрольный байт**]  
[**признак конца посылки**]

Блок данных является не обязательным и зависит от команды. Команда представляет собой символ, определяющий действия, которые должен выполнять контроллер. В лабораторной работе нужно преду-

смотреть 2 типа команд: прием контроллером величины задающего воздействия и передача контроллером текущего значения сигнала обратной связи (выходной код АЦП).

При приеме команды второго типа контроллер должен сформировать посылку, имеющую следующий формат:

[**признак начала посылки**]  
[**блок данных**]  
[**контрольный байт**]  
[**признак конца посылки**]

Блок данных в этом случае представляет собой шестнадцатеричное значение выходного кода АЦП, закодированное ASCII символами.

Во всех посылках старший байт блока данных передается первым.

Контрольный байт представляет собой значение *CRC* или *LRC*, вычисленное для команды и блока данных.

Тип контрольной суммы, признак начала посылки и символы для кодирования команд выбираются из таблицы в соответствии с вариантом.

Признак конца посылки представляет собой последовательность символов *CR* (возврат каретки 0x0D), *LF* (перевод строки 0x0A).

### Пример протокола.

Исходные данные:

- признак начала строки: ‘:’;
- символ команды первого типа (прием контроллером величины задающего воздействия): ‘K’;
- символ команды второго типа (передача контроллером текущего значения сигнала обратной связи): ‘L’;
- тип контрольной суммы: *LRC MODBUS*.

Текущие данные:

- величина задающего воздействия: 500 (0x1F4);
- величина кода на выходе АЦП: 215 (0x0D7);
- значение *LRC* для команды первого типа и блока данных: 168 (0xA8);
- значение *LRC* для команды второго типа: 156 (0x9C);
- значение *LRC* для блока данных ответа: 41 (0x29).

Таким образом, посылка установки задающего воздействия (от терминала к контроллеру) имеет вид:

**‘:’K’1’F’4’A’8’CR LF**

Ответ на эту посылку контроллером не формируется.

Посылка получения текущего значения сигнала обратной связи (от терминала к контроллеру) имеет вид:

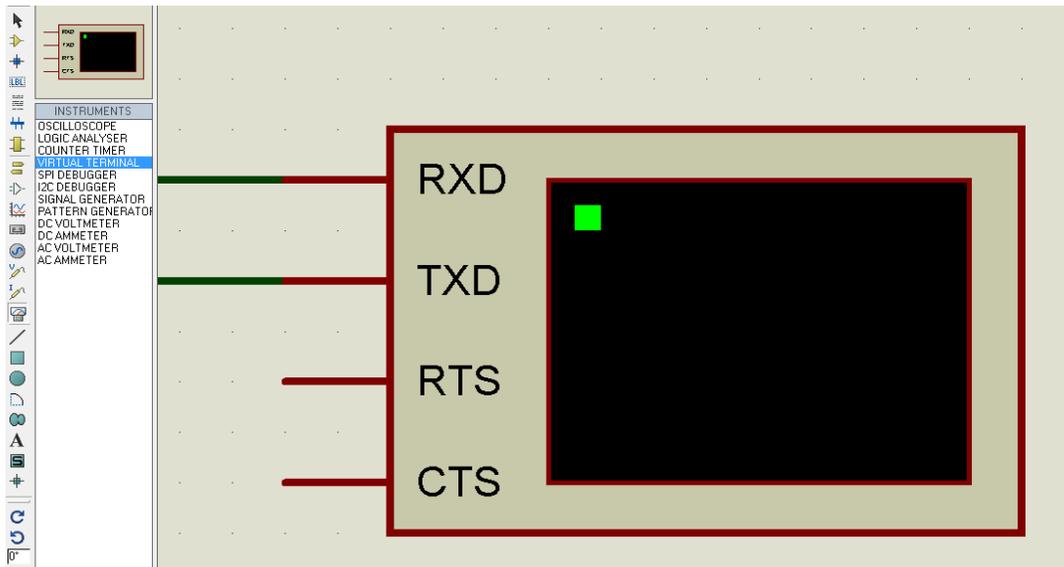
**‘:’L’9’C’CR LF**

В ответ контроллер отсылает текущее значение кода на выходе АЦП:

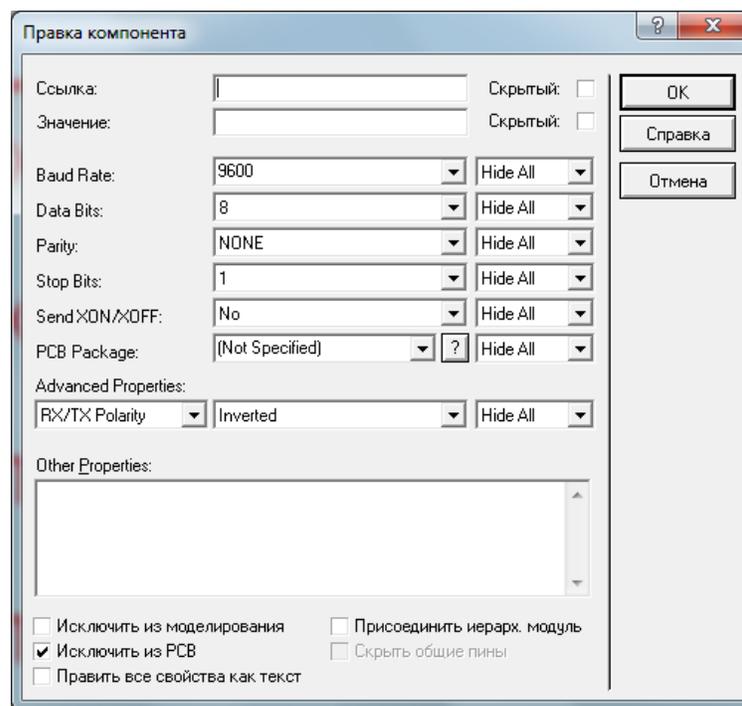
**‘:’0’D’7’2’9’CR LF**

**2. В основной цикл (не в обработчик прерывания!) управляющей программы микроконтроллера добавить код, реализующий полученный протокол обмена данными по последовательному каналу (с использованием аппаратного модуля USART микроконтроллера). Программа также должна выполнять проверку целостности пакета путем вычисления величины контрольной суммы и сравнения ее с принятым значением. В случае несоответствия значений пакет отбрасывается.**

**3. Выполнить моделирование работы системы в пакете *Proteus*. К линиям *Tx* и *Rx* микроконтроллера подключить компонент «VIRTUAL TERMINAL»:**



В настройках компонента задать скорость обмена в соответствии с вариантом:



Остальные опции оставить неизменными.

Моделирование следует выполнять в режиме реального времени.

В процессе моделирования проанализировать реакцию системы на управляющие послыки, а также убедиться в том, что система не реагирует на послыки с неправильной контрольной суммой.

**4.** Оформить отчет по работе. Отчет должен содержать описание протокола, исходный код управляющей программы, результаты моделирования работы системы и вывод по работе.