# Лабораторная работа 3

# ПРОЕКТИРОВАНИЕ ДИСКРЕТНЫХ СИСТЕМ УПРАВЛЕНИЯ С НАБЛЮДАТЕЛЯМИ СОСТОЯНИЯ

# *Цель работы:* научиться создавать динамические модели дискретных систем с наблюдающими устройствами; овладеть навыками построения цифровых систем с наблюдателями полного порядка.

# 3.1. Основные сведения

При решении практических задач управления методами размещения полюсов часто встречаются случаи, когда часть переменных вектора состояния оказываются неизмеримыми. Если известна математическая модель системы, то можно восстановить состояние системы по измеряемым сигналам входа и выхода.

Восстановление вектора состояния (*k*) называется его оценкой, а устройство, обеспечивающее получение оценки по измерениям управления u(*k*) и **k**-мерного (**k**<n) вектора выхода y(*k*) на конечном интервале времени, ̶ наблюдателем.

Пусть стационарный объект описывается традиционной системой разностных уравнений

(3.1)

Предположим, что матрицы *A, B, C* известны, тогда вектор x(*k*) можно аппроксимировать состоянием (*k*) модели

(*k+1*) = (3.2)

которая имеет тот же вход, что и объект (3.1), то есть u(*k*).

Если модель (3.2) является идеальной аппроксимацией системы (3.1), в том смысле, что их параметры и начальные условия идентичны, то состояния x(*k*) и (*k*) также совпадают. Таким образом, модель (3.2) восстанавливает состояния объекта (3.1). При этом в модели (3.2) не используется измеряемый выход.

Качество восстановления улучшается, если ввести в модель разность измеренного выхода и его оценки *(k)* в виде обратной связи (предложено американским ученым Люенбергером):

*(k+1)* = (3.3)

Здесь *L* − матрица коэффициентов размерностью **n**×**k**, обеспечивающая требуемую динамику сходимости оценки к вектору состояния объекта.

Введем ошибку восстановления

. (3.4)

Вычитая (3.3) из дифференциального уравнения (3.1), получим

*=*

(3.5)

Мы видим, что модель (3.5) восстанавливает все составляющие вектора состояния, поэтому она называется наблюдателем состояния полного порядка.

Общая схема системы управления с наблюдателем состояния полного порядка будет иметь вид, представленный на рис 3.1.

Рис. 3.1. Схема системы управления с наблюдателем полного порядка

При синтезе наблюдателей используют ***принцип разделимости***.

**Принцип разделимости**. Допустим, что объект описывается уравнениями (3.1), регулятор состояния (см. рис. 3.1)

(3.6)

а наблюдатель полного порядка - уравнением (3.3).

Тогда уравнения состояния всей системы имеют вид:

. (3.7)

Если ввести ошибку наблюдения (3.4), то с помощью линейного преобразования

(3.8)

получим

(3.9)

Из (3.9) согласно правилам вычисления определителей блочных матриц (определитель блочной матрицы равен произведению определителей ее диагональных блоков) следует, что характеристическое уравнение всей системы имеет вид,

то есть равно произведению характеристических полиномов системы с полными измерениями и собственно наблюдателя. Из этого следует, что характеристические полиномы замкнутой системы управления и наблюдателя независимы и, следовательно, мы можем назначать полюсы наблюдателю независимо от полюсов системы управления.

Процесс, описываемый уравнениями (3.7), как и (3.9), состоит из двух взаимно независимых процессов: из процесса оценки вектора и из собственно процесса управления. Разделение процессов в наблюдателе и регуляторе принято называть ***принципом разделимости***.

С учетом принципа разделимости синтез наблюдателя состояния выполняется ***независимо*** от синтеза регулятора состояния.

Очевидно, что для выполнения операций синтеза наблюдателей состояния полного порядка, могут быть использованы те же алгоритмы, которые используются для синтеза регуляторов состояния.

Перед тем, как проектировать наблюдатель, необходимо проверить наблюдаемость системы. Это можно сделать с помощью следующего фрагмента программы:

A = […];

C = […];

No = (obsv(A,C)); %Построение матрицы наблюдаемости

unno = length (A) - rank (No) ; % Число ненаблюдаемых мод

if unno == 0

disp ( 'Система полностью наблюдаема' )

else

M = 'Число ненаблюдаемых мод равняется ' ;

disp ([M unno])

end

Матрица наблюдаемости может быть построена с помощью функции obsv, которая вызывается в одном из вариантов:

>> No = obsv(A, С);

>> No = obsv(sys);

>> No = obsv(sys.A, sys.С);

Для расчета коэффициентов обратных связей наблюдателя могут быть применены функции acker и place, в которых, в данном случае, в качестве параметров следует использовать транспонированные матрицы A', С' и pn − вектор желаемых полюсов наблюдателя.

Например:

>> A=[0 1; 2 3];

>> C=[1 0];

>> pn=[ -5 -5];

>> Lt=acker(A', C', pn);

>> L=Lt';

L=

13

66

**3.2. Методический пример**

**Пример.** Построить дискретные модальный регулятор и наблюдатель полного порядка для исходного непрерывного объекта управления второ­го порядка (рис. 3.2).

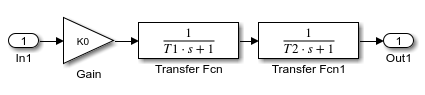


Рис. 3.2. Объект управления

Работа с примером начинается с запуском на исполнение m-файла сценария **vipolnenie\_lr\_3.m**. Удобнее всего запускать этот файл с точками останова (см. комментарии в файле), либо по шагам (режим Step) при выполнении отдельных этапов программы.

Пусть исходный непрерывный объект управления имеет следующие параметры: Kо =5, T1 =0.3, T2 =0.05.

Представим объект управления в виде общей передаточной функции, изображенной на рис. 3.3, и выполним его дискретизацию с периодом Ts = 0.01c (файл **ish\_blok.slx**).

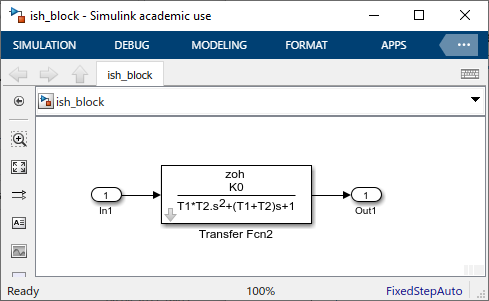


Рис. 3.3. Преобразование и дискретизация объекта управления

Получим описание дискретного объекта управления в пространстве состояний в рабочей области MATLAB с помощью следующих команд:

**[a,b,c,d]=dlinmod('ish\_block',Ts)**

**SYS=ss(a,b,c,d,Ts)**

**step(SYS,2.5)**

Уравнения состояния, полученные для объекта на рис. 3.3, имеют матрицы *A*, *B* и *C*:

*A*=; *B*=; *C*=.

Составим структурную схему объекта управления в программе Simulink (файл **ishodnaya\_model.slx**, см. рис. 3.4) и выполним ее симуляцию (см. рис. 3.5).

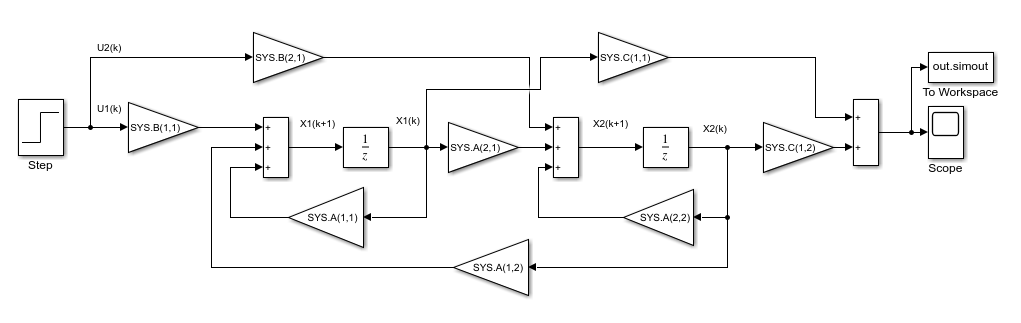


Рис. 3.4. Модель объекта управления и результат моделирования

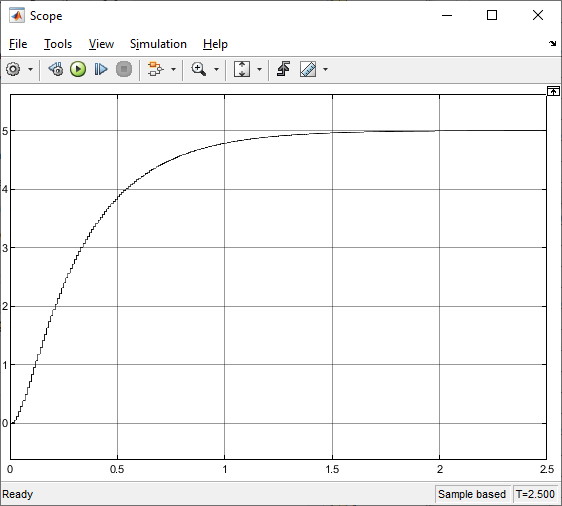


Рис. 3.5. Переходная характеристика ОУ, показанного на рис 3.4

Вычислим сначала полюса для непрерывной системы 2 порядка с желаемым временем регулирования 0.5 с по распределению Баттерворта, получим:

%Вводим порядок системы и желаемое время ПП для сценария

**>>batterwort** %(файл **batterwort.m**), n=2, tgel=0.5, получаем:

p =

-4.1295 + 4.1295i

-4.1295 - 4.1295i

Для вычисления желаемого расположения полюсов дискретной системы в z-области используем команду

pd=exp(p\*Ts)

Получим:

pd =

0.9587 + 0.0396i

0.9587 - 0.0396i

Вычислим коэффициенты модального регулятора и нормирующий коэффициент:

Kd = place(SYS.A,SYS.B,pd)

ydu=dcgain(SYS);

Adk = SYS.A - SYS.B\*Kd;

Fd =ss(Adk,SYS.B,SYS.C,SYS.D,Ts);

yduk=dcgain(Fd);

kdnorm = ydu/yduk

Получим:

Kd = - 0.1315 0.1288

kdnorm = 0.5507

Составим структурную схему полученной дискретной системы в Simulink (файл **model\_s\_mod\_reg.slx**), показанную на рис. 3.6.

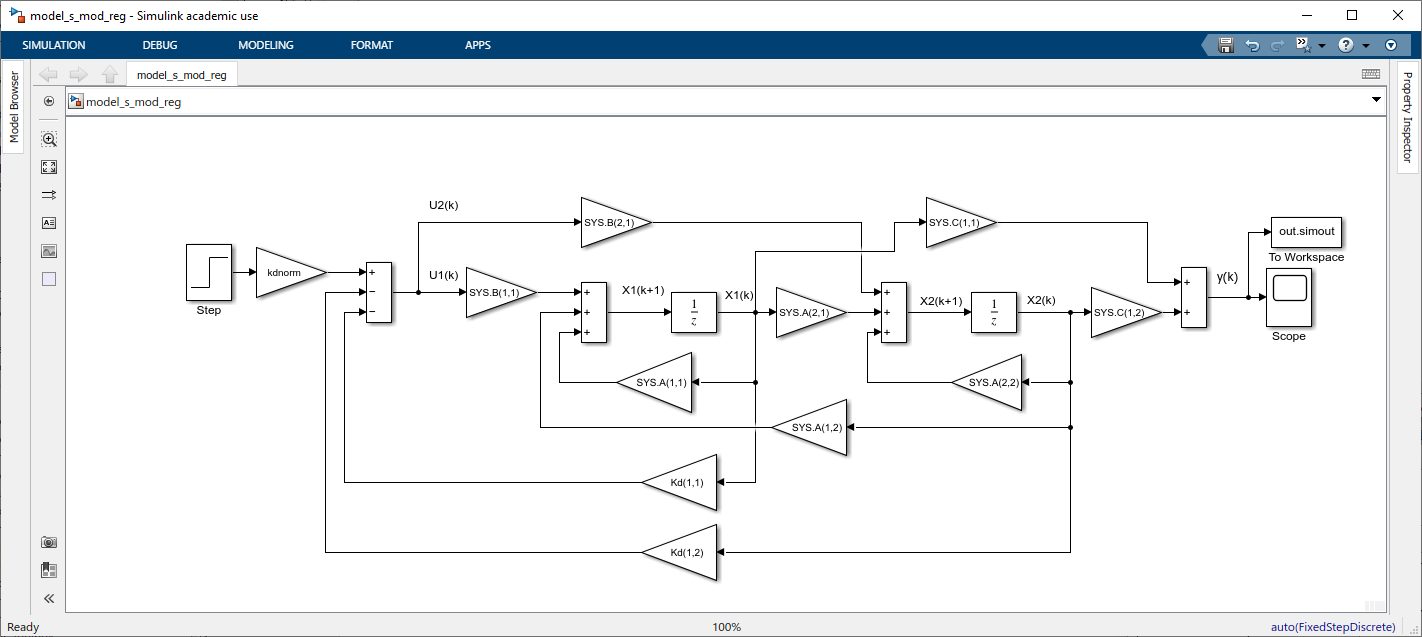


Рис. 3.6. Модель дискретной системы управления с модальным регулятором

Произведем расчет наблюдателя (коэффициентов матрицы *L*) с учетом того факта, что динамика замкнутого наблюдателя (собственные значения матрицы *A-LC*) должна быть примерно в 3-10 раз быстрее (обычно в 3 раза), чем динамика системы с модальным управлением.

Выберем желаемое время переходного процесса наблюдателя, равное 0,1 с.

%Вводим порядок системы и желаемое время ПП для сценария

**>>batterwort** %(файл **batterwort.m**), n=2, tgel=0.1, получаем:

Получим:

tgel = 0.1000

Вектор желаемых полюсов непрерывного наблюдателя

p = -20.6475 +20.6475i

-20.6475 +20.6475i

Вектор желаемых полюсов дискретного наблюдателя

>> pnd=exp(p\*Ts)

pnd =

0.7962 + 0.1668i

0.7962 - 0.1668i

Рассчитаем коэффициенты наблюдателя

>> l=place(SYS.A', SYS.C', pnd);

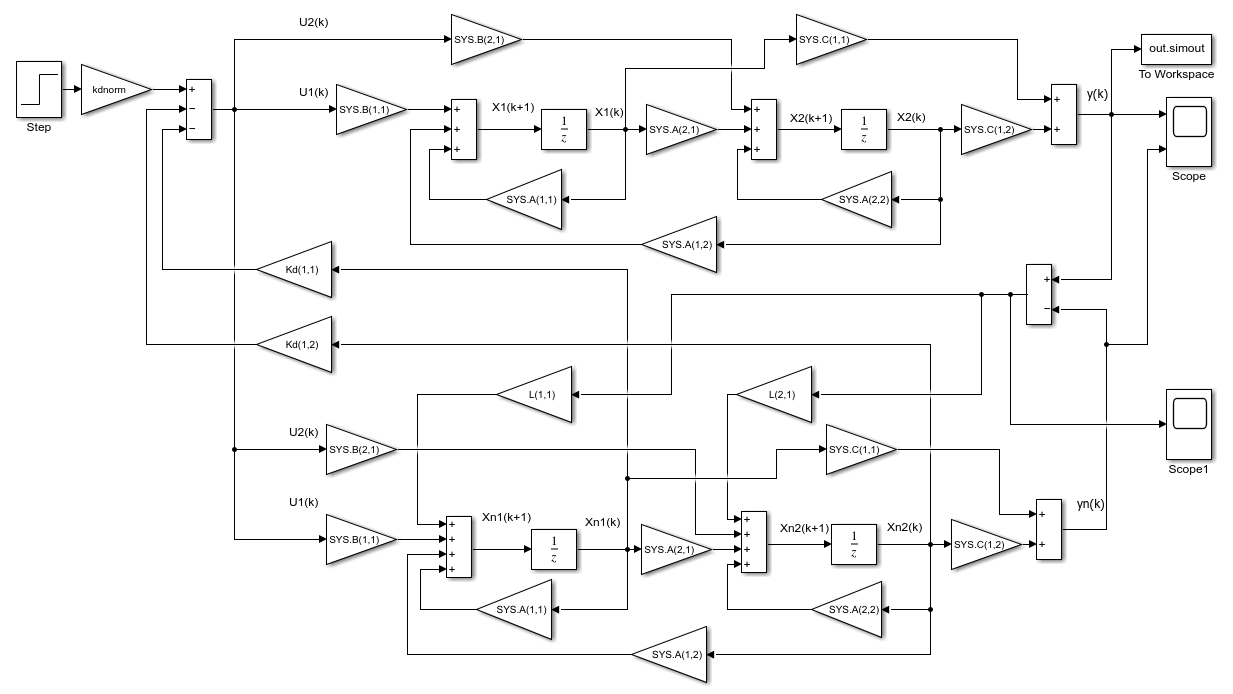
L = l'

L = 6.9258

6.0728

Ознакомьтесь с работой наблюдателя, как с отдельной подсистемой, используя файл **nabludatel.slx**.

Составим структурную схему дискретной системы, замкнутую с помощью рассчитанного ранее модального регулятора и наблюдателя, показанную на рис. 3.7 (файл **model\_mod\_reg\_nabl.slx**).



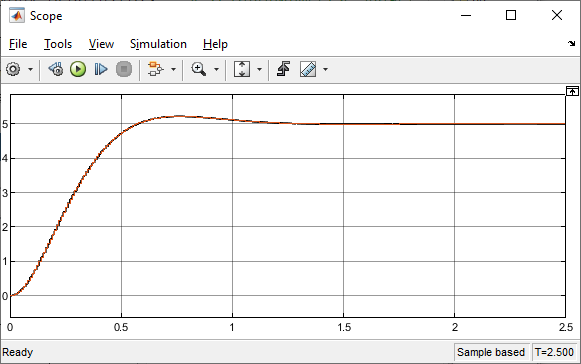
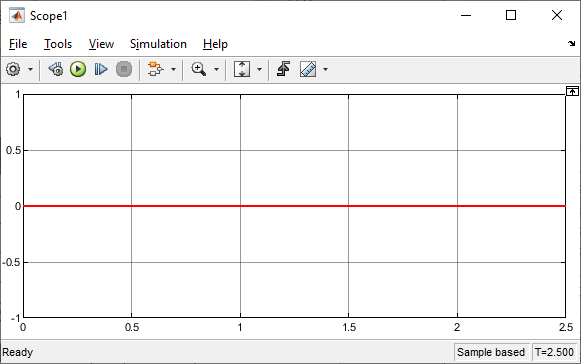
 

Рис. 3.7. Система с модальным регулятором и наблюдателем и результаты моделирования

Как видно из результатов моделирования (см. рис. 3.7), ошибка наблюдения, в том числе и на начальном интервале, равна нулю! Почему? Поскольку математические описания объекта и наблюдателя в точности совпадают, то при одинаковом входном сигнале будут совпадать как сигналы векторов состояния, так и сигналы на выходах. Такой идеализированный случай конечно же не встречается на практике.

С целью проверки работы системы с наблюдателем, изменим параметры объекта управления. Пусть Kо1 =4.95, T11 =0.33, T21 =0.06. Отличия исходного объекта управления от объекта управления с новыми параметрами иллюстрируются рис. 3.8.

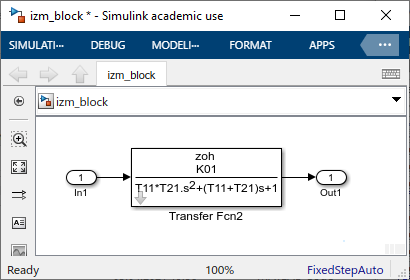
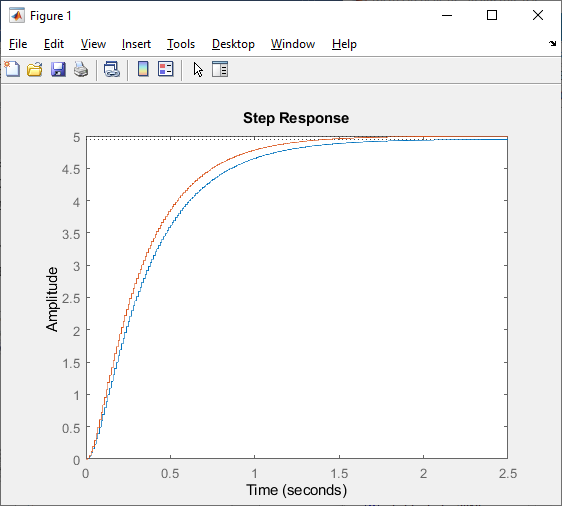
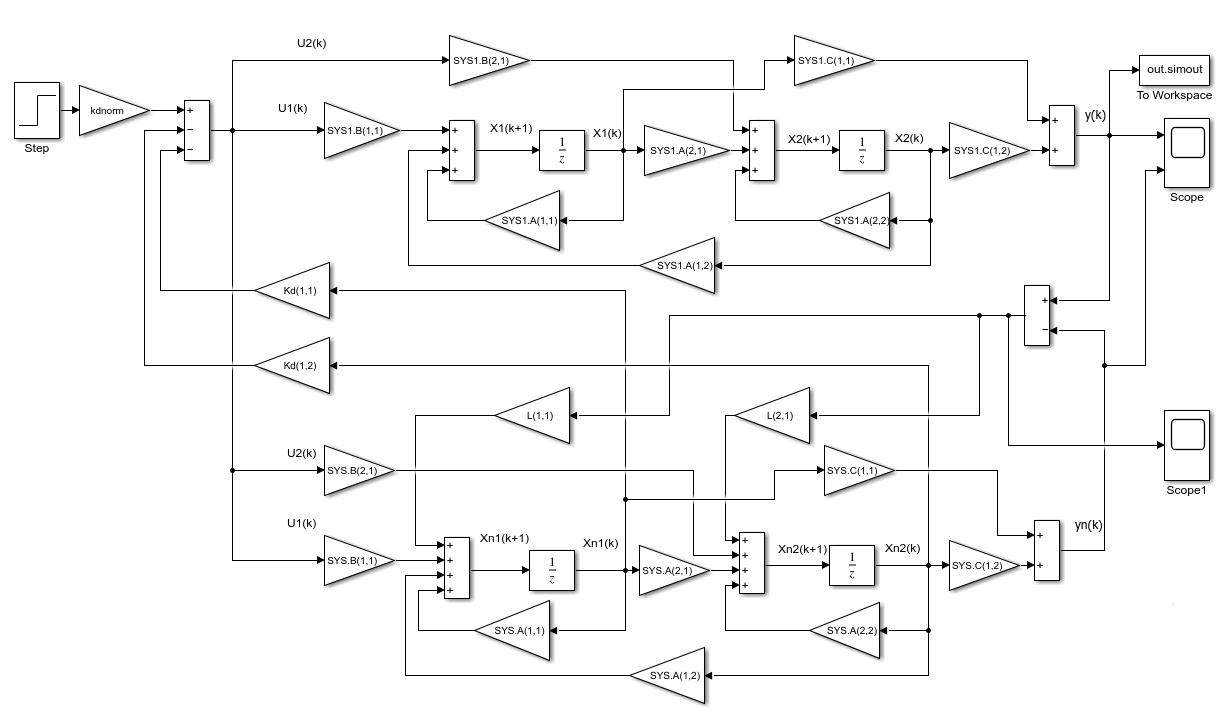
 

Рис. 3.8. Изменение параметров объекта управления и график

измененного переходного процесса (синяя линия)



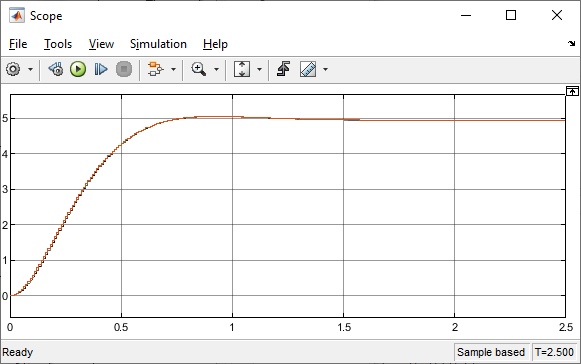
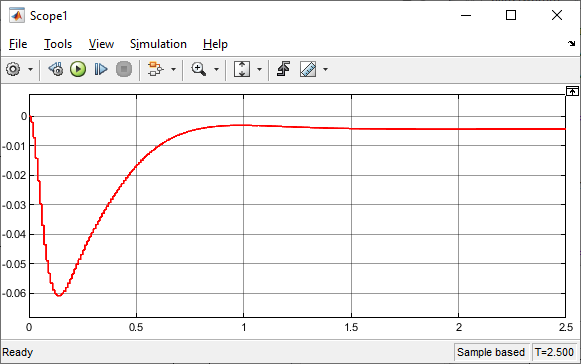
 

Рис. 3.9. Система управления с измененными параметрами объекта и результаты моделирования

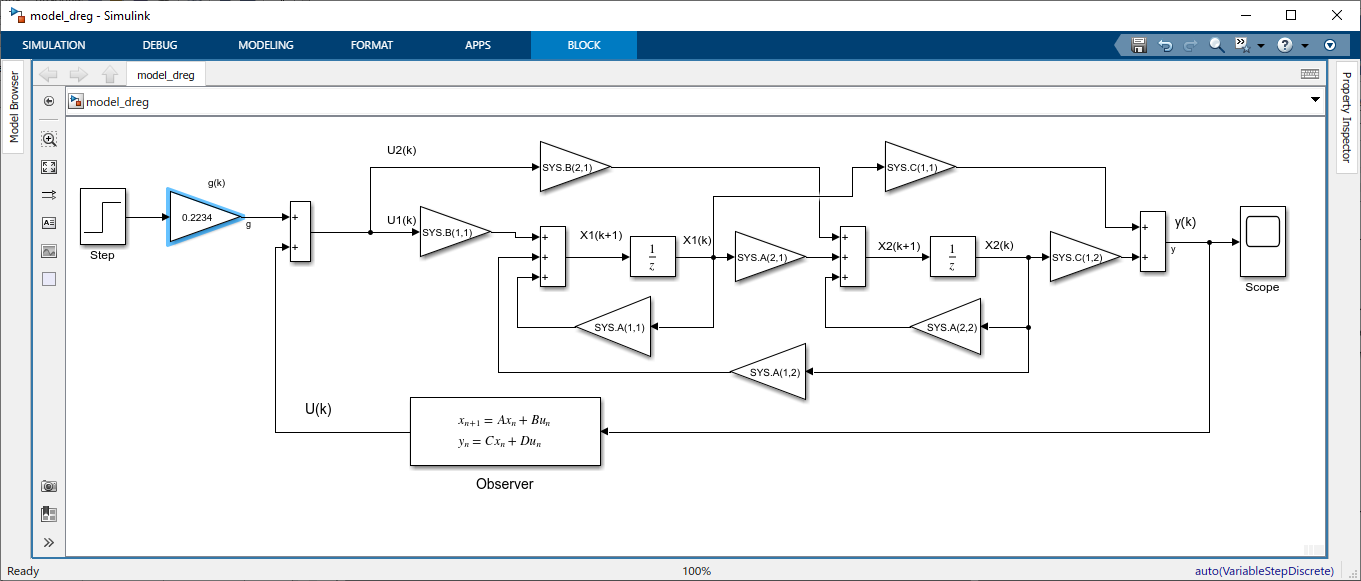
Сформируем новое описание объекта управления и запишем его компоненты в переменную SYS1, после чего модифицируем его Simulink-модель. Система управления с измененными параметрами (файл **model\_mod\_reg\_nabl\_izm.slx**) и результаты моделирования представлены на рис. 3.9. Видно, что появляется как динамическая, так и статическая ошибка (ее легко убрать нормирующим коэффициентом), не превышающие в сумме 1.2% от установившегося значения.

Построение наблюдателя состояния совместно с регулятором состояния можно осуществить с использованием функции **dreg()**. В случае построения наблюдателя полного порядка функция записывается следующим образом:

**>> [An,Bn,Cn,Dn]=dreg(SYS.A,SYS.B,SYS.C,SYS.D,Kd,L),**

где SYS.A, … , SYS.D − матрицы пространства состояний дискретного объекта управления (исходной системы); kd b L − матрицы (векторы) цифровых модального регулятора и наблюдателя соответственно; An, … , Dn − соответствующие матрицы обобщенного дискретного регулятора. На рис. 3.10 представлена схема Simulink-модели системы управления с обобщенным дискретным регулятором (регулятор + наблюдатель) и результаты симуляции (файл **model\_dreg.slx**).

Формулу для расчета коэффициента нормализации для данной системы выведите самостоятельно.



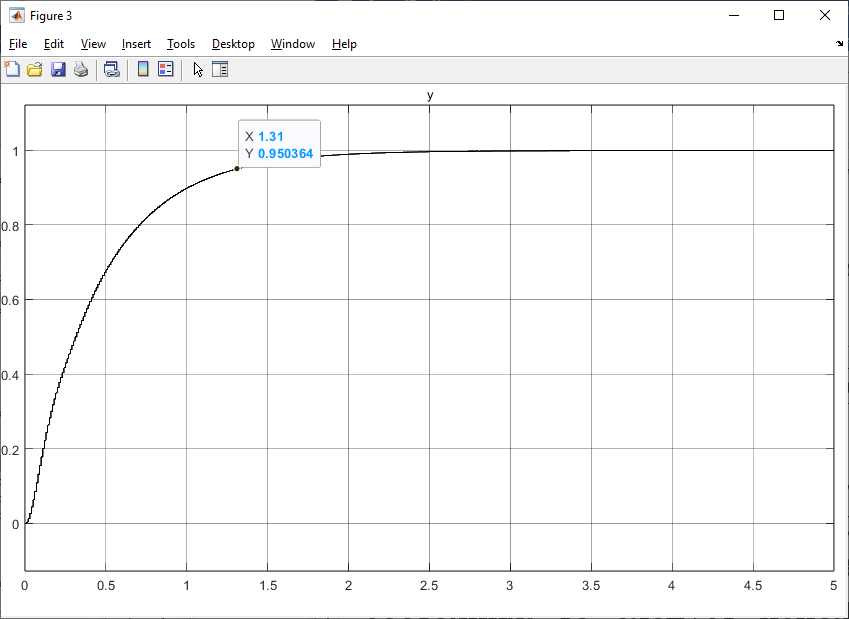


Рис. 3.10. Система управления с обобщенным дискретным регулятором **dreg** и результат моделирования

Как видно из рис. 3.10 результат не совпадает с предыдущим результатами (см. рис. 3.7 и 3.9).

Приведенный пример иллюстрирует возможность использования наблюдателя состояния в том случае, когда переменные состояния объекта управления оказываются недоступными для непосредственного измерения. Если не измеряемыми оказываются только часть компонентов вектора состояния, то следует использовать редуцированный наблюдатель для их восстановления в том числе и редуцированную форму функции **dreg()**.

**Примечание.** Для описанного методического примера имеются файлы моделей (их имена указаны в тексте) и файл-сценарий выполнения всех пунктов работы с именем **vipolnenie\_lr\_3.m**. Файлы приведены в редакции версии MATLAB R2021b.

**3.3. Порядок выполнения работы**

1. Изучить методику построения дискретных модальных регуляторов и наблюдателей по описанному примеру, используя прилагаемые файлы моделей.

2. Построить дискретные модальный регулятор и наблюдатель полного порядка для следящей системы управления на базе двигателя постоянного тока в соответствии с вашим вариантом задания.

Предварительно получите дискретную математическую модель вашей исходной системы в пространстве состояний с помощью функции **dlinmod():**

**>>[Ad,Bd,Cd,Dd]=dlinmod('SYS', Ts),**

где SYS − имя файла Simulink-модели исходной передаточной функции разомкнутой по положению вашей следящей системы Wсс = Wdpt/s, преобразованной в цифровую форму (см. рис. 3.3), а Ts − период дискретизации.

3. Построить дискретный наблюдатель в виде отдельной Simulink-модели и убедиться в его работоспособности. При выборе собственной динамики наблюдателя использовать рекомендации, приведенные в п. 3.2.

4. Использовать полученный дискретный наблюдатель **для исходного непрерывного объекта управления**, построив соответствующую непрерывно-дискретную модель системы управления.

5. Построить дискретный наблюдатель состояния, используя стандартную функцию **dreg**(…). Сравнить его работу с наблюдателем состояния, построенным в п. 2.

6. Сравнить полученные результаты с результатами предыдущих практических и лабораторных работ.

7. На основании проведенных исследований и полученных результатов составить отчет. Сделать выводы по каждому этапу работы и по работе в целом.

Примечание: При построении наблюдателя и проверки его работоспособности, то есть проверки его "подстройки под объект управления", ввести в математическое описание ОУ (либо наблюдателя) невязку, либо по значениям параметров, либо по начальным условиям, либо и то и другое одновременно. Отличие параметров наблюдателя от соответствующих параметров объекта примите равными в пределах 5÷10%. Невязку по начальным условиям (ввод начальных условий отличных от нуля) целесообразно ввести в непрерывную часть непрерывно-дискретной модели вашей системы. Здесь выбор полностью за вами.

**3.4. Контрольные вопросы**

1. Какую задачу решает наблюдатель?

2. В чем заключается особенность наблюдателя Люенбергера?

3. Что означает принцип разделимости для построения наблюдателя?

4. Какие функции выполняют наблюдатели полного и пониженного (редуцированного) порядка?

5. Каким образом выбирается собственная динамика наблюдателя?

6. Что означает понятие "полная наблюдаемость"? Как проверяется наблюдаемость системы?

7. С какой целью вводится невязка по параметрам объект/наблюдатель? Какова ее практическая значимость?

8. Какие задачи выполняют функции reg/dreg?