**Лекция 3**

# Проектирование систем на основе заданного расположения полюсов

Суть **модального управления** состоит в нахождении численных значений коэффициентов передачи безынерционных обратных связей (ОС) по всем переменным состояния объекта для того, чтобы обеспечить заданное распределение корней характеристического уравнения замкнутой САУ. Распределение корней в свою очередь полностью определяет движение (динамику) системы.

Каждая составляющая такого движения, соответствующая отдельному корню pi (или паре комплексно сопряженных корней), за рубежом называется модой – отсюда и название "модальное управление".

Корни характеристического уравнения однозначно зависят от его коэффициентов, поэтому модальное управление можно понимать как целенаправленное изменение коэффициентов характеристического уравнения объекта с помощью безынерционных ОС.

Корни характеристического уравнения или полюса замкнутой системы определяют следующие характеристики переходного процесса:

* время достижения пер­вого максимума,
* время установления переходного процесса,
* показатели ко­лебательности.

Таким образом, обеспе­чение требуемых динамических свойств проектируемой системы **полностью определяется** расположением ее полюсов на комплексной плоскости.

Проектная процедура синтеза, использующая *метод заданного расположения полюсов* предполагает следующий порядок действий:

* исходя из требований к переходному процессу, **выбрать расположение  
  полюсов** передаточной функции замкнутой системы на комплексной  
  плоскости (обычно с использованием стандартных полиномов);
* **рассчитать значения** коэффициентов обратных связей, при которых такое расположение полюсов достигается (используются библиотечные функции).

Этот метод требует, чтобы была известна ss-модель системы. Для непрерывных систем эта модель имеет вид:

 (3.1)

где u − вектор входов и у − вектор измерений; *A*, *B*, *C*, *D* − матрицы системы, соответствующих размерностей. Матрица *D* в технических системах обычно нулевая.

# Расчет коэффициентов обратных связей

При использовании обратной связи по переменным состояния ***u = − Кх*** динами­ка замкнутого контура задается уравнением

*** = (А− ВК)х***

и полюсы замкнутой системы являются собственными значениями матрицы

***А− ВК***.

Схема системы с модальным регулятором приведена на рис. 3.1.

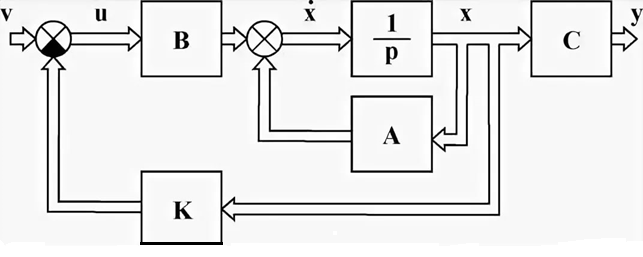


Рис. 3.1. Схема системы с модальным регулятором

Применяя алгоритмы, реализующие метод заданного расположения полюсов, можно вычислить матрицу коэффициентов обратных связей *K*, ко­торая обеспечит ***любое*** желаемое расположение этих полюсов на комплекс­ной плоскости при условии, что система является ***управляемой***.

Последовательность проектирования здесь следующая:

1. Проверяется управляемость пары матриц {*A*, *B*}, например

**>>A = […];**

**>>B = […];**

**Co = ctrb (A, B) %Вычисление матрицы управляемости**

**unco = length (A) − rank (Co) ;% Число неуправляемых мод**

**if unco == 0**

**disp ( 'Система полностью управляема' )**

**else**

**T = 'Число неуправляемых мод равняется ';**

**disp ([T unco])**

**end**

1. Исходя из требований к переходному процессу, определяется желаемый характеристический полином и желаемое расположение полюсов замкнутой системы, например, с использованием метода стандартных полиномов (см. рис. 3.2).
2. Вычисляется матрица коэффициентов обратных связей *K*, обеспечивающая заданное расположение полюсов в проектируемой системе. При этом используют следующие функции:

**K = acker(A, B, p)** % Для одномерной системы

**K = place(A, B, p)** % Для одномерных и многомерных систем

Здесь аргументы A и B – матрицы ss-модели, p – вектор желаемых полюсов. Возвращаемая величина K – матрица коэффициентов обратных связей

Функция **acker** предназначена для расчета одномерных систем с не­большим числом переменных состояния (обычно до 8). Функция **place** может быть приме­нена как для одномерных, так и для многомерных систем и использует специальный алгоритм, который гарантирует высокую точность вычислений.

*Примечание:*

*В технической литературе приводятся различные наборы стандартных характеристических полиномов 1-8 порядков и соответствующие им графики переходных процессов с указанными на них показателями качества. Наиболее часто используются следующие полиномы:*

***- Полином Ньютона;***

***- Полином Баттерворта;***

***- Полином Бесселя;***

*- Полином, доставляющий минимум интегралу от квадрата ошибки;*

*- Полином, минимизирующий интеграл*I3 (интеграл модуля ошибки)*;*

*- Полином с 5%-ным перерегулированием;*

*- Полином Грехема-Летропа;*

*- Полином, сформированный методом двойных пропорций.*

***Подробнее см.:*** [***http://web.snauka.ru/issues/2013/11/28937***](http://web.snauka.ru/issues/2013/11/28937) ***(действующая ссылка)***

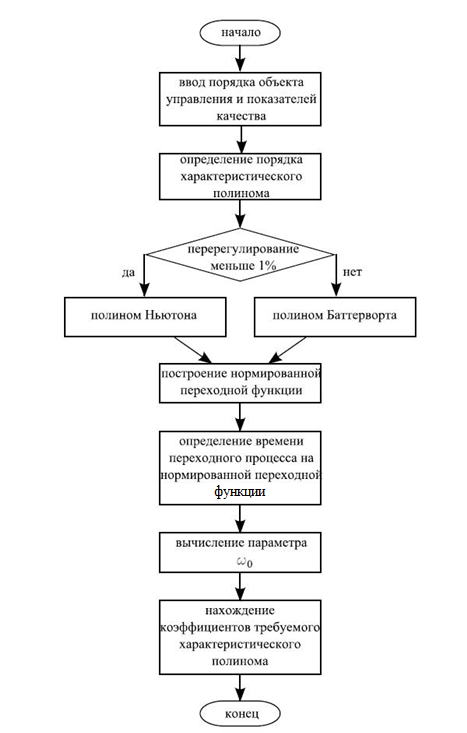
**

Рис. 3.2. Алгоритм нахождения требуемого характеристического полинома

**Полиномы Ньютона** D(p) = (p+ω0)*n*

Таблица 3.1

|  |  |
| --- | --- |
| Порядок системы n | Стандартный полином Ньютона |
| 2 | p2 +2ω0 p+ ω02 |
| 3 | p3 +3ω0 p2+ 3ω02 p+ ω03 |
| 4 | p4 + 4ω0 p3+ 6ω02 p2+ 4ω03 p+ ω04 |
| 5 | p5+5ω0 p4+ 10ω02 p3+ 10ω03 p2+ 5ω04 p+ ω05 |

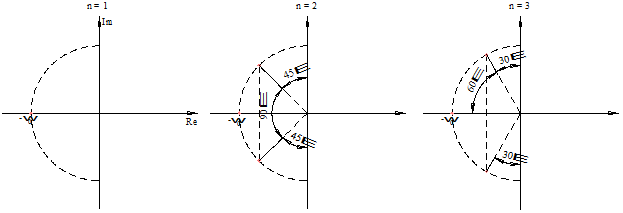
pi = −ɷ0, i = . (кратные корни)

**Полиномы Баттерворта**

Таблица 3.2

|  |  |
| --- | --- |
| Порядок системы n | Стандартный полином Баттерворта |
| 2 | p2 +1,414ω0 p+ ω02 |
| 3 | p3 +2ω0 p2+ 2ω02 p+ ω03 |
| 4 | p4 + 2,613ω0 p3+ 3,414ω02 p2+ 2,613ω03 p+ ω04 |
| 5 | p5+3,24ω0 p4+ 5,24ω02 p3+ 5,24ω03 p2+ 3,24ω04 p+ ω05 |

Корни этих полиномов располагаются на окружности радиусом ω0 на одинаковом условном расстоянии друг от друга симметрично оси вещественных.



При синтезе модальных регуляторов электроприводов, по мнению многих авторов, целесообразно использовать стандартные полиномы Бесселя.

**Полиномы Бесселя**

Таблица 3.3

|  |  |
| --- | --- |
| Порядок системы n | Стандартный полином Бесселя |
| 2 | p2 +1,73ω0 p+ ω02 |
| 3 | p3 +2,43ω0 p2+ 2,47ω02 p+ ω03 |
| 4 | p4 + 3,12ω0 p3+ 4,39ω02 p2+ 3,2ω03 p+ ω04 |
| 5 | p5+3,81ω0 p4+ 6,78ω02 p3+ 6,89ω03 p2+ 3,94ω04 p+ ω05 |

Для того, чтобы облегчить синтез непрерывных систем с модальным регулятором с использованием полиномов Баттерворта, предлагается использовать следующую программу (файл **gel\_polusa\_Batterwort.m**):

**%Программа рассчитывает вектор желаемых полюсов**

**%проектируемой системы в соответствии со стандартными**

**%настройками по Баттерворту.**

**%Входными данными являются порядок системы и желаемое**

**%время переходного процесса.**

**%Автор: А.В. Никоза, СПбГЭТУ(ЛЭТИ) 13.11.2018**

**n= input('Введите порядок системы n = ');**

**[z,p,k]=buttap(n);% Обращение к функции buttap(n)**

**[b,a]=zp2tf(z,p,k);**

**SYS=tf(b,a);**

**step(SYS), title('Нормированный переходный процесс по Баттерворту');**

**grid;% Нормированный переходный процесс**

**[Y,T]=step(SYS, 0:0.01:20);**

**T\_dyn = T(Y>1+0.05 | Y < 1-0.05 ); % Пятипроцентная зона**

**tau = T\_dyn(end);%Нормированное значение времени переходного процесса**

**tgel= input('Введите желаемое время переходного процесса tgel = ')**

**w0 = tau/tgel;% Значение среднегеометрического корня**

**for i=1:n % Расчет коэффициентов желаемого полинома**

**a(i+1)=a(i+1).\*w0^(i);**

**end**

**b=a(n+1); % Расчет коэффициента числителя ПФ**

**SYS=tf(b,a);**

**[z,p,k]=zpkdata(SYS,'v');% Векторы нулей, полюсов и коэффициент усиления желаемой системы**

**figure**

**step(SYS), title('Переходный процесс в желаемой системе');**

**grid% Переходный процесс в желаемой системе**

**disp('Вектор желаемых полюсов')**

**p**

**disp('Коэффициент числителя ПФ')**

**b**

***Примечание 1: В графических окнах при выборе Characteristics/Settling Time по умолчанию установлена 2% зона. Измените это значение на 5% с помощью Properties…/Options/Show settling time within.***

***Примечение 2:***

***Для того, чтобы использовать настройку системы в соответствии с полиномом Бесселя, в приведенной выше программе следует воспользоваться командой* [z,p,k]=besselap(n).**

***Стандартная переходная характеристика в соответствии с полиномом Ньютона n-го порядка может быть построена следующим образом:***

***%Текст скрипта newton***

**n= input('Введите порядок системы n = ');**

**r(1:n)=-1**

**p=poly(r) %Вектор коэффициентов полинома**

**W=tf(1,p)**

**step(W), title('Нормированный переходный процесс');**

**grid;% Нормированный переходный процесс**

**или**

**n= input('Введите порядок системы n = ');**

**z = [];**

**k = 1;**

**p = ones(1,n)\*(-1);**

**[b,a]=zp2tf(z,p,k);**

**SYS=tf(b,a);**

**step(SYS), title('Нормированный переходный процесс');**

**grid;% Нормированный переходный процесс**

**Пример расчета**

В качестве примера рассмотрим следящую систему автоматического управления с двигателем постоянного тока. Параметры двигателя соответствуют одному из двигателей, приведенной в лекции 2, табл. 1. Дополнительно учтен момент инерции нагрузки, приведенный к валу двигателя. Детализированная Simulink-модель исходной разомкнутой системы приведена на рис. 3.3 (файл **primer\_lec3.slx**). На рис. 3.4 показан переходный процесс в скоростном контуре следящей системы. Контур положения разомкнут.

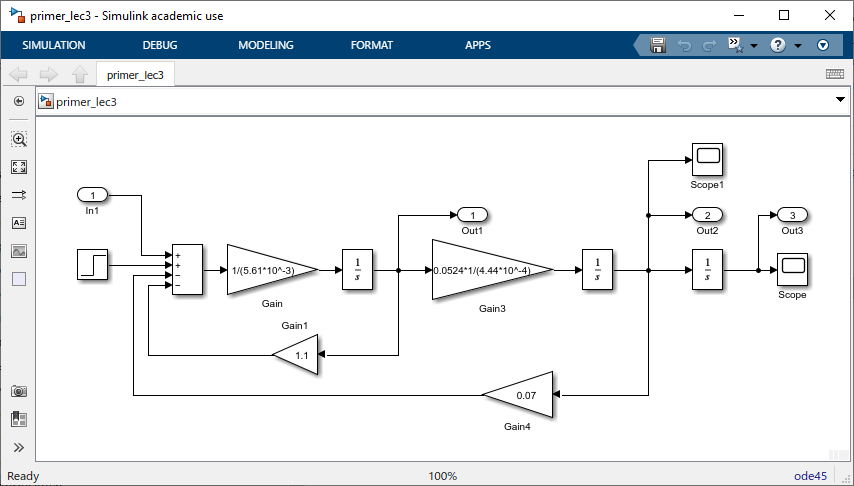


Рис. 3.3. Исходная модель двигателя с нагрузкой

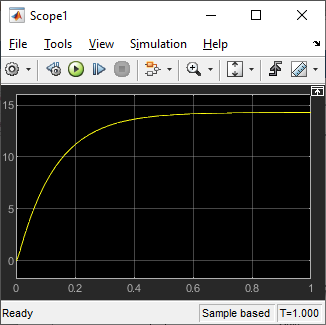


Рис. 3.4. Переходный процесс по скорости в исходной системе

Получим описание системы в пространстве состояний:

>>[A,B,C,D]=linmod('primer\_lec3')

A =

-196.0784 -12.4777 0

118.0180 0 0

0 1.0000 0

B = 178.2531

0

0

C =

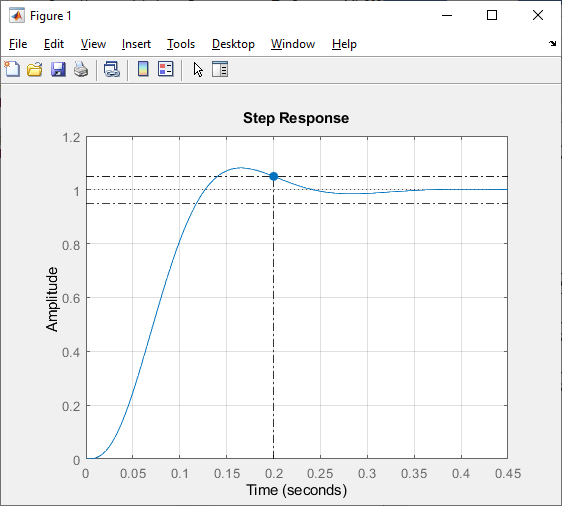
1 0 0

0 1 0

0 0 1

После получения матриц системы проверяем ее управляемость (см. выше). В данном случае система полностью управляема.

Воспользуемся приведенной выше программой для расчета желаемого размещения полюсов. Зададим порядок системы n=3 и требуемое время переходного процесса tпп=0.2 с, получим:



p =

-29.8000

-14.9000 +25.8076i

-14.9000 -25.8076i

Рис. 3.5. Желаемый ПП

Найдем значения коэффициентов обратных связей, используя функцию place(…):

**>> K = place(A,B,p)**

K = −0.7656 0.0144 1.2579

Проверим полученный результат, замкнув систему обратными связями с найденными значениями коэффициентов (см. рис. 3.6 и 3.7 и файл **primer1\_lec3.slx**).

**Замечание** (***важно!***): Коэффициент передачи полученной замкнутой системы отличен от единицы (нет единичной обратной связи). Для обеспечения равенства коэффициента передачи замкнутой системы единице, на входе по задающему сигналу добавлен усилитель с коэффициентом передачи kу = 1.2579.

В общем случае для того, чтобы замкнутая система имела такой же коэффициент передачи, как и разомкнутая (либо некоторое заданное пользователем значение yu, например, единица), следует дополнительно в цепь входного сигнала ввести нормирующий коэффициент. Установившееся значение может быть вычислено с помощью функции **dcgain(sys)**.

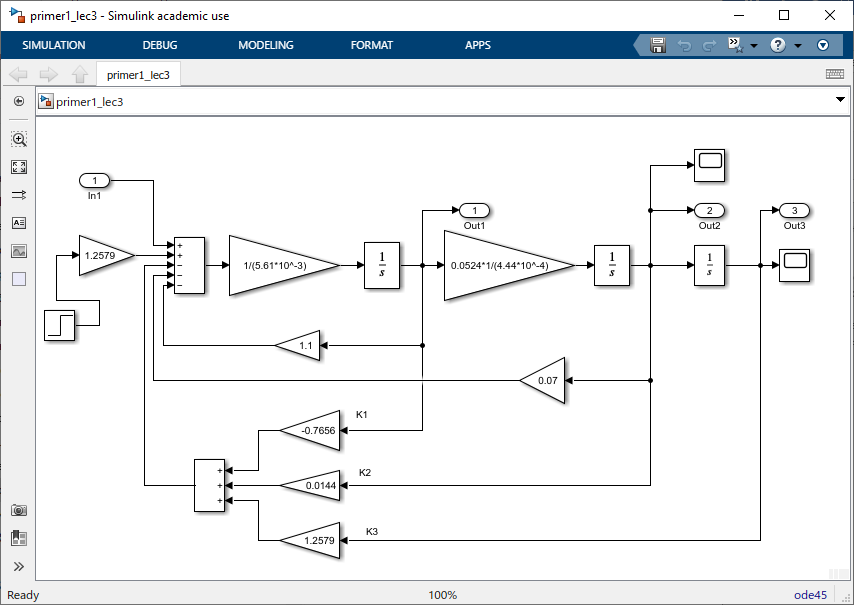


Рис. 3.6. Скорректированная система

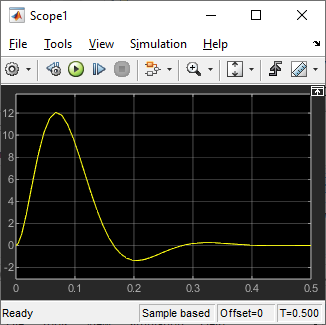
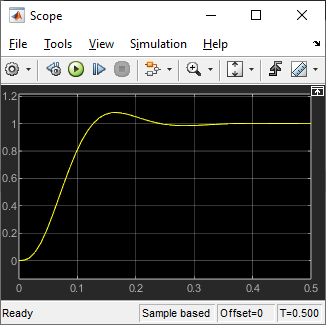
 

Рис. 3.7. Переходные процессы в скорректированной системе

В MATLAB перечисленные процедуры могут быть выполнены с помощью следующих операций (файл **norm\_koeff.m**):

**Sss=ss(A,B,C,D); % Описание разомкнутой системы**

**Ak = A - B\*K; % Матрица Ak замкнутой системы**

**F =ss(Ak,B,C,D); % Описание замкнутой системы**

**yuk=dcgain(F); % Установившееся значение на выходе**

**% замкнутой системы**

**yu=1; % Требуемое установившееся значение на выходе системы**

**knorm = yu/yuk % Нормирующий коэффициент для системы**

Результат:

knorm =

0 0 1.2579

Следует указать особенность функции **place**. Если кратность желаемых полюсов (например, полином Ньютона) оказывается больше ранга матрицы B, то функция **place** не работает. При этом выдается ошибка:

**>> The "place" command cannot place poles with multiplicity greater than**

**rank(B)**

Для систем с единственным управляющим входом в случае кратных полюсов (полином Ньютона) следует использовать функцию **acker**.

Для автоматизации проектирования непрерывных и цифровых систем с модальным управлением на нашей кафедре была разработана программа в среде графического приложения AppDesigner (авторы доц. С.Е. Голик, доц. А.В. Никоза, магистрант А. Фещенко). Также разработана программа для вычисления желаемых полюсов замкнутой непрерывной системы с использованием метода стандартных полиномов. Эти программы кратко описаны в дополнительном материале к данной лекции.

**Задание**. Ознакомьтесь с функциями MATLAB **poly**() и **pole**() и примером расчета:

>> r(1:n)=-1 %Вектор корней полинома

>> p=poly(r) %Вектор коэффициентов полинома

>> SYS=tf(1,p) %Передаточная функция

>> pg=pole(SYS) % Корни характеристического полинома (полюса) ПФ

**Задание к практической работе 3.**

Ознакомьтесь с приведенными в лекции программами и примерами расчета. Ознакомьтесь с программами автоматизации расчетов желаемых полюсов и проектирования непрерывных и цифровых систем с модальным управлением. Методические примеры и программы находятся в архивном файле, прилагаемом к лекции. В качестве примера выполните расчеты с использованием программ автоматизации расчетов (см. папку PROGRAM\_AUTOMATIZ) для объекта управления, который описан в лекции. Сравните полученные вами данные с соответствующими данными, приведенными в лекции. Оформите краткий отчет и сделайте выводы. Используйте данные программы при выполнении лабораторной работы 2, а также в вашей дальнейшей работе.

***Заключение.*** В лекции был рассмотрен синтез модального управления для непрерывных систем. Методика синтеза для дискретных систем в целом совпадает с рассмотренной, но производится с дискретными моделями ОУ. В таких моделях непрерывные матрицы ОУ заменяются их дискретными эквивалентами, а желаемые полюса замкнутой системы их отображениями на z-плоскость. Синтез цифровых систем будет рассмотрен в одной из следующих лекций.

**ДОПОЛНИТЕЛЬНЫЙ МАТЕРИАЛ К ЛЕКЦИИ**

В библиотеке функций Control System Toolbox имеется функция lqr

lqr Linear-quadratic (LQ) - линейный квадратичный критерий

state-feedback regulator for state-space system

Синтаксис:

[K,S,e] = lqr(SYS,Q,R,N)

[K,S,e] = LQR(A,B,Q,R,N)

Эта функция позволяет вычислить матрицу K, содержащую оптимальные коэффициенты, такую, что закон обратной связи u = − Kx по переменным состояния линейной стационарной САУ вида dx/dt=Ax+Bu минимизирует квадратичный функционал качества

Qx + ,

где Q, N и R – матрицы соответствующих размерностей (по умолчанию N – нулевая матрица).

Дополнительно к матрице оптимальных коэффициентов K функция lqr возвращает матрицу S решения уравнения Риккати и вектор собственных значений матрицы A−B\*K. Необходимым условием возможности расчета матрицы K является требование полной управляемости исходной САУ. Для цифровых систем используется функия dlqr. Нужно сказать, что в практике проектирования эта функция находит ограниченное применение, т. к. стратегия поиска оптимального решения требует достаточно высокой квалификации пользователя.

**Управляемость и наблюдаемость линейных стационарных непрерывных систем**

**Определение ранга матрицы в Matlab**

Рассмотрим некоторую прямоугольную матрицу A. Если в этой матрице выделить произвольно *k* строк и *k* столбцов, то элементы, стоящие на пересечении выделенных строк и столбцов, образуют **квадратную** матрицу k-го порядка. Определитель этой матрицы называется *минором k-го порядка*матрицы А. Очевидно, что матрица А(nxn) обладает минорами любого порядка от 1 до наименьшего из чисел m и n. Среди всех отличных от нуля миноров матрицы А найдется по крайней мере один минор, порядок которого будет наибольшим. Наибольший из порядков миноров данной матрицы, отличных от нуля, называется*рангом*матрицы. Если ранг матрицы А равен *r*, то это означает, что в матрице А имеется отличный от нуля минор порядка *r*, но всякий минор порядка, большего чем *r*, равен нулю.

Ранг матрицы А вычисляется в Matlab с помощью функции **rank**(A).

**Управляемость**

Состояние непрерывной системы (3.1) управляемо, ***если и только*** ранг матрицы управляемости Qy равен размерности пространства состояний системы, где

http://www.masters.donntu.edu.ua/2005/eltf/krasik/library/st5.files/i16.gif

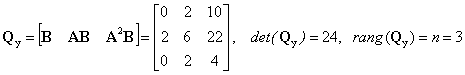
− матрица управляемости.  
 Пример. Определим управляемость системы

|  |
| --- |
| http://www.uran.donetsk.ua/%7Emasters/2005/eltf/krasik/library/st5.files/i19.gif |

a)

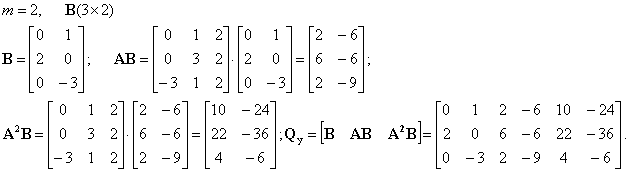
|  |
| --- |
| http://www.uran.donetsk.ua/%7Emasters/2005/eltf/krasik/library/st5.files/i20.gif |

матрица управляемости



система управляема.

б)



Вычисляются n(m−1)+1=4 определителя для матриц Q1, ..., Q4 порядка n=3. Матрица для расчета каждого последующего определителя формируется путем отброса первого (в матрицах Q1, ..., Q3) и захвата следующего столбца матрицы управляемости Qу. Если хотя бы один из определителей отличен от нуля, то система управляема.

Здесь Q1=[0 1 2; 2 0 6; 0 −3 2];

det(Q1)= −16,

значит, система, рассматриваемая в примере, управляема, и другие определители можно не вычислять.

**Наблюдаемость**

Аналогично формулируется и условие наблюдаемости для линейных непрерывных стационарных систем.

Система (3.1) наблюдаема, если и только если ранг матрицы наблюдаемости системы равен размерности пространства состояний.

|  |
| --- |
| http://www.uran.donetsk.ua/%7Emasters/2005/eltf/krasik/library/st5.files/i32.gif |

Свойства управляемости и наблюдаемости систем необходимо рассматривать совместно для того, чтобы задача об управлении была корректно поставлена и принципиально разрешима.

В Matlab для вычисления матриц управляемости и наблюдаемости и их рангов могут быть использованы следующие функции:

rank(ctrb(A,B));

rank(obsv(A,C)),

естественно, что значения матриц A, B, C должны быть предварительно определены.

## Для дискретных стационарных систем свойства управляемости и наблюдаемости выглядят аналогично. Однако, при анализе управляемости и наблюдаемости дискретных систем желательно ознакомиться с соответствующей литературой, например, книгой Куо Б. Теория и проектирование цифровых систем управления.

## ПРИМЕРЫ ПРОГРАММ АВТОМАТИЗАЦИИ РАСЧЕТОВ, РАЗРАБОТАННЫХ НА КАФЕДРЕ

**Программа расчета желаемых корней непрерывных систем на основе метода стандартных полиномов**

**(файл metod\_std\_polynom\_App.mlapp или metod\_std\_polynom.m)**

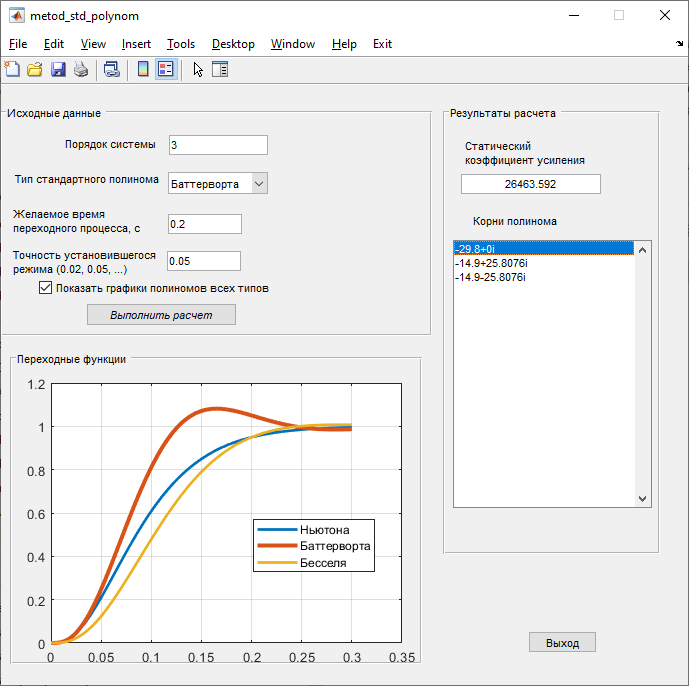


Рис. 3.8. Графический интерфейс программы

**Программа расчета модальных регуляторов для одномерных непрерывных и цифровых систем управления**

**(файл program\_modal.mlapp)**

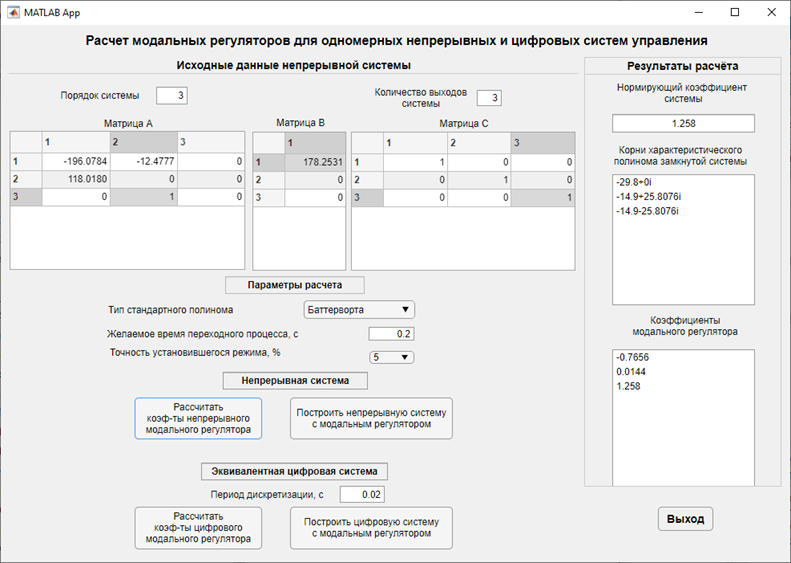
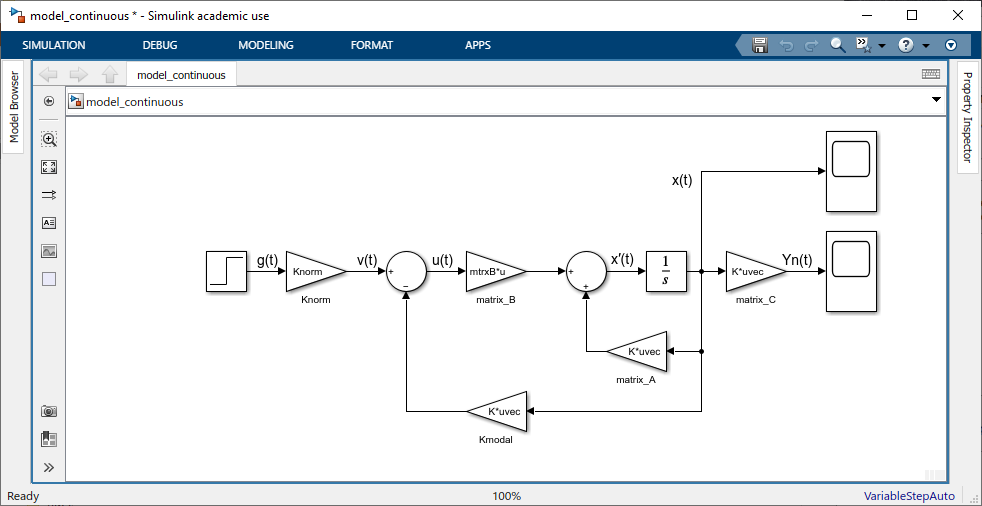


Рис. 3.9. Графический интерфейс программы

**Результаты работы программы**



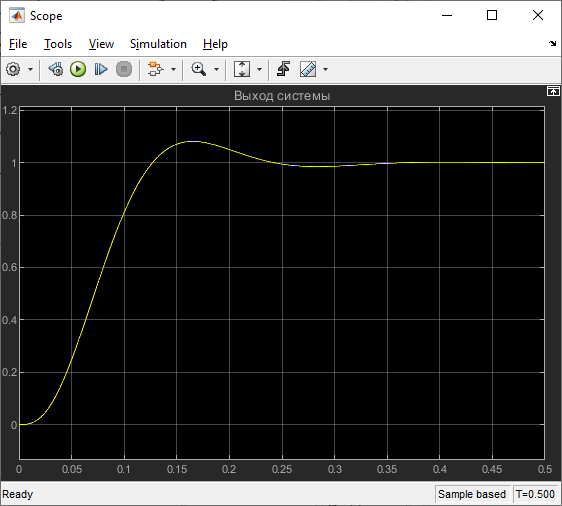
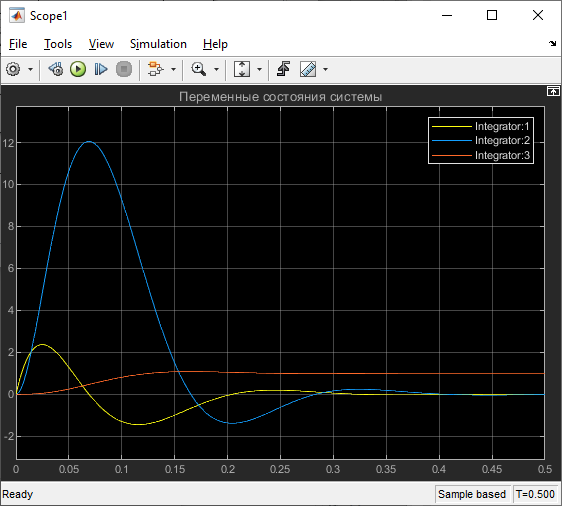
 

Рис. 3.10. Векторно-матричная Simulink-модель системы с модальным управлением

и результаты симуляции