**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра КСУ**

отчет

**по лабораторной работе №4**

**по дисциплине «Математическое моделирование объектов и систем управления»**

Тема: **РЕАЛИЗАЦИЯ АЛГОРИТМОВ УПРАВЛЕНИЯ В СРЕДЕ MATLAB**

**Вариант 5**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 9492 |  | Викторов А.Д.  Керимов М.М. |
| Преподаватель |  | Шпекторов А.Г. |

Санкт-Петербург

2023

**Цель работы:** изучить основные принципы формирования алгоритмов управления, освоить средства моделирования систем управления в среде MATLAB.

**ЗАДАНИЕ**

Объект управления – корабль, движение которого рассматривается в горизонтальной плоскости. Управление обеспечивается с помощью вертикального руля направления с учетом инерционности привода рулей. В качестве математической модели процесса стабилизации на заданном курсе рассматривается система обыкновенных линейных дифференциальных уравнений:



где β – угол дрейфа; ω*y* – угловая скорость по рысканию; φ – угол рыскания;

δ – угол отклонения руля; *u* – управляющий сигнал. Значения параметров:

*a*11 = –0.159, *a*12 = 0.267, *b*1 = –0.0215, *a*21 = 0.103, *a*22 = –0.188, *b*2 = –0.0213.

*Содержание работы*:

1. Сформировать LTI-объект управления.

2. Сформировать регулятор в виде *u = k1β + k2ω + k3φ + k4δ*.

3. Определить коэффициенты регулятора, обеспечивающие собственные частоты замкнутой системы *s1 =* –0.1, *s2* = –0.05, *s3,4* =–0.025±0.08*j*

4. Замкнуть систему синтезированным регулятором и проверить корни характеристического полинома.

5. Увеличить *s2* в 10 раз, получить новый регулятор и сравнить с предыдущим.

**ХОД РАБОТЫ**

1. Создадим LTI-объект. Для этого запустим код, представленный в листинге 1:

Листинг 1. Создание LTI-объекта

% object

Ao = [a11 a12 0 b1;

a21 a22 0 b2;

0 1 0 0;

0 0 0 0];

Bo = [ 0;

0;

0;

1];

Co = [ 1 0 0 0;

0 1 0 0;

0 0 1 0;

0 0 0 1];

Do = [ 0;

0;

0;

0];

sys\_ob = ss(Ao, Bo, Co, Do);

Матрица B имеет размерность n на m, где n – число переменных состояния, а m – число входных управляющих воздействий. Матрица G имеет размерность n на l, где l – число возмущающих воздействий. результат выполнения кода программы представлен на рисунке 1:

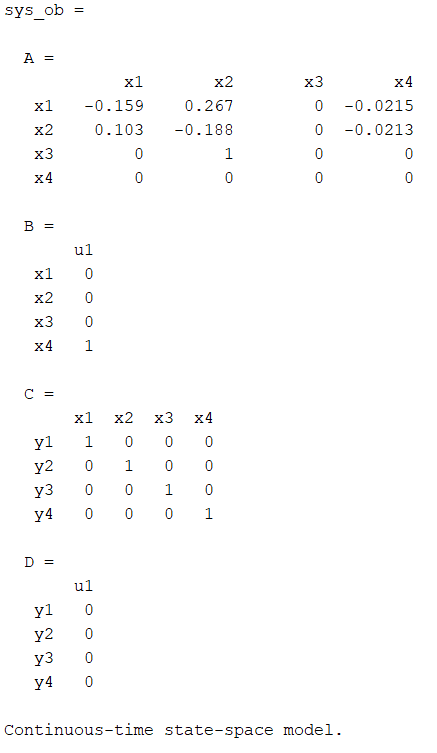


Рисунок 1 – Создание модели исследуемой системы

1. Сформируем регулятор с коэффициентами, обеспечивающими обеспечивающие собственные частоты замкнутой системы *s1 =* –0.1, *s2* = –0.05, *s3,4* =–0.025±0.08*j*,и найдем полюса системы. Для этого задействуем код из листинга 2:

Листинг 2. Формирование регулятора и поиск полюсов системы

%% modal regulator 1

s1 = -0.1;

s2 = -0.05;

s3 = -0.25-0.08i;

s4 = -0.25+0.08i;

p = [s1 s2 s3 s4];

K = -place(get(sys\_ob, 'A'), get(sys\_ob, 'B'), p);

sys\_reg = ss(K);

sys = lft(sys\_ob,sys\_reg, 1, 4);

C\_sys = [0 0 1 0];

D\_sys = 0;

B\_sys = [ 0;

0;

0;

-K(3)];

set(sys, 'C', C\_sys, 'D', D\_sys, 'B', B\_sys);

hold on

step(sys, 'r')

pole\_sys\_1 = pole(sys)

Результат выполнения программы, представленной в листинге 2, показан на рисунке 2:

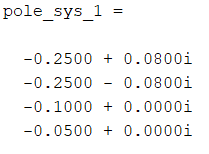


Рисунок 2 – Полюса замкнутой системы

Как видно из рисунка 2, полученная замкнутая система является устойчивой, так как все полюсы имеют отрицательную вещественную часть.

1. Увеличим параметр *s2* в 10 раз, сформируем новый регулятор и получим новые значения полюсов системы. Для этого задействуем код из листинга 3:

Листинг 2. Формирование регулятора и поиск полюсов системы при *s2* = –0.5

%% modal regulator 2

s2 = -0.5;

p = [s1 s2 s3 s4];

K = -place(get(sys\_ob, 'A'), get(sys\_ob, 'B'), p);

sys\_reg = ss(K);

sys = lft(sys\_ob,sys\_reg, 1, 4);

B\_sys = [ 0;

0;

0;

-K(3)];

set(sys, 'C', C\_sys, 'D', D\_sys, 'B', B\_sys);

step(sys,'b')

grid on

legend('s\_2 = -0.05','s\_2 = -0.5')

pole\_sys\_2 = pole(sys)

Результат выполнения программы, представленной в листинге 3, показан на рисунке 3:

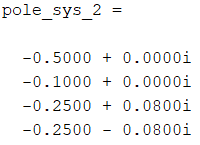


Рисунок 3 – Полюса замкнутой системы при s2 = –0.5

На рисунке 4 представлены переходные процессы системы для обоих случаев.

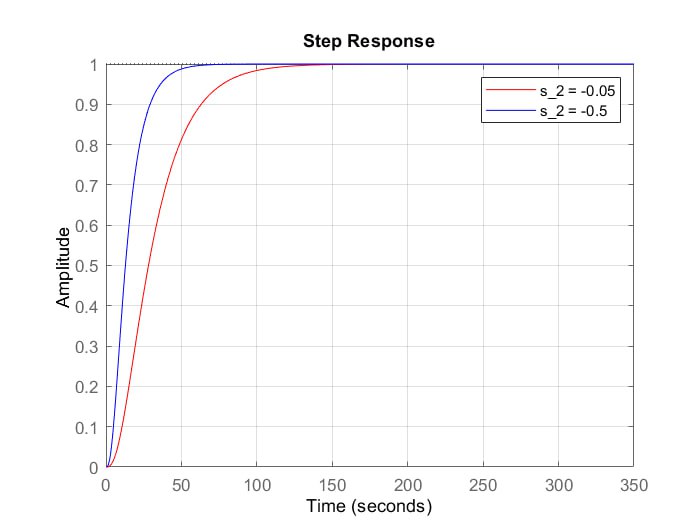
****

Рисунок 4 – Сравнительный график переходных процессов

**ПРАКТИЧЕСКОЕ ЗАДАНИЕ**

Дана система, которая описывается системой уравнений:

где Вектор состояния имеет вид , вектор выхода . Входное воздействие . При этом . Необходимо замкнуть систему регулятором и подать на вход воздействие и построить переходный процесс.

1. Создадим модель объекта управления и регулятора. Система, построенная в Simulink представлена на рисунке 5.

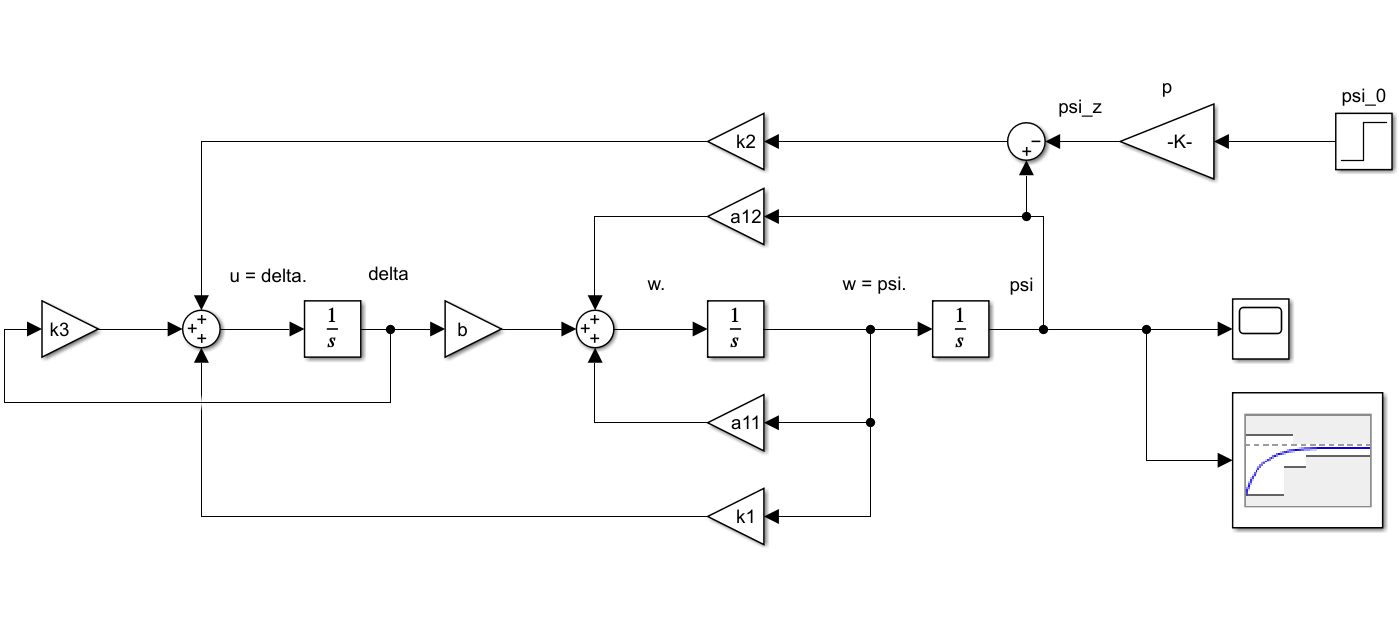


Рисунок 5 - Схема системы в Simulink

1. Запустим код программы, в котором происходит инициализация коэффициентов объекта управления, а также задание начальных значений коэффициентов регулятора. Код программы представлен в листинге 4:

Листинг 4. Исходный код программы

clc,clear

a11 = -0.1253;

a12 = -0.004637;

b = -0.002198;

k1 = 10;

k2 = 1;

k3 = -0.2;

1. Зададим в блоке *Check Step Response Characteristics* коэффициенты регулятора k в качестве изменяемых коэффициентов и зададим ограничения для этих коэффициентов как показано на рисунке 6.

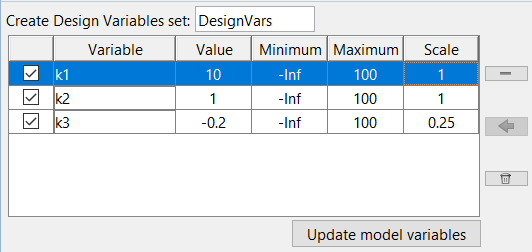


Рисунок 6 - ограничения переменных коэффициентов регулятора

1. Запустим оптимизацию и в качестве результата получим новые оптимизированные коэффициенты регулятора и график переходного процесса (рис. 7).

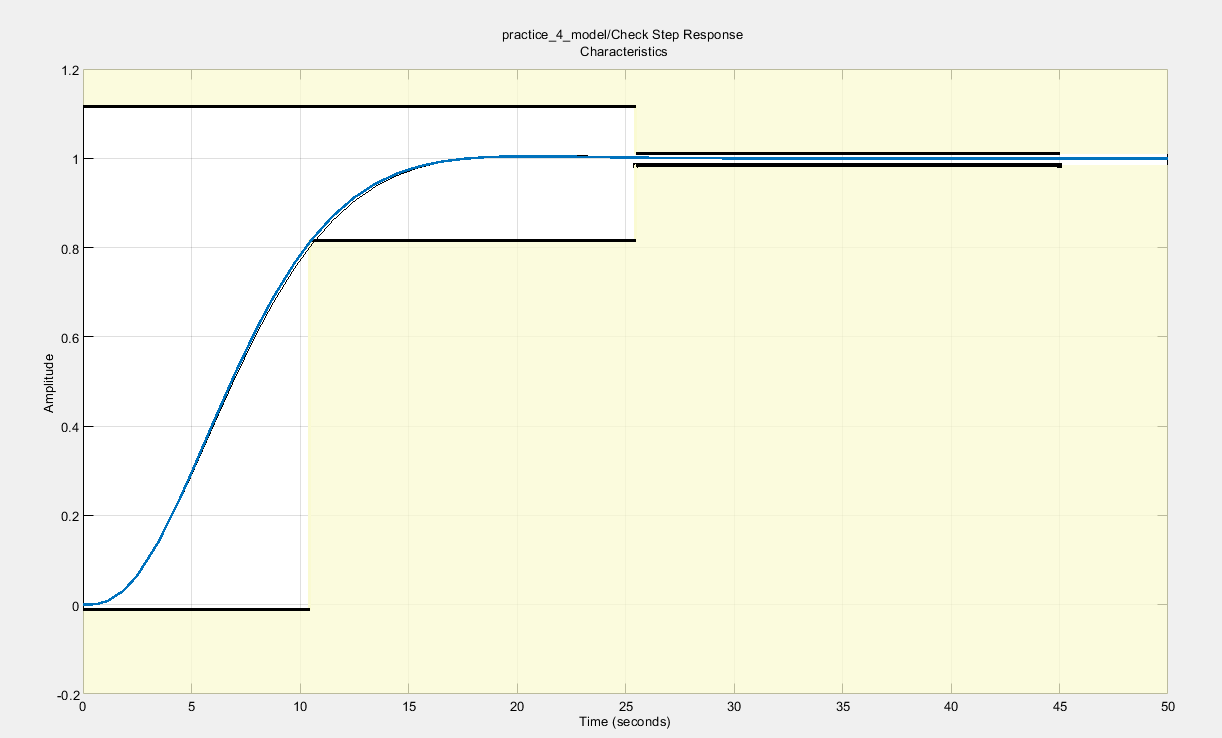


Рисунок 7 - График оптимизированного переходного процесса

Значения коэффициентов регулятора: k1 = 98.95, k2 = 18.46, k3 = -0,78.

**ВЫВОД**

В результате выполненной лабораторной работы были изучены основные принципы формирования алгоритмов управления и освоены средства моделирования систем управления в среде MATLAB.

**ПРИЛОЖЕНИЕ А. Полный исходный код программы**

clc, clear, close all

a11 = -0.159;

a12 = 0.267;

a21 = 0.103;

a22 = -0.188;

b1 = -0.0215;

b2 = -0.0213;

%x = [ beta;

% omega,

% phi

% delta];

% object

Ao = [a11 a12 0 b1;

a21 a22 0 b2;

0 1 0 0;

0 0 0 0];

Bo = [ 0;

0;

0;

1];

Co = [ 1 0 0 0;

0 1 0 0;

0 0 1 0;

0 0 0 1];

Do = [ 0;

0;

0;

0];

sys\_ob = ss(Ao, Bo, Co, Do);

%% modal regulator 1

s1 = -0.1;

s2 = -0.05;

s3 = -0.25-0.08i;

s4 = -0.25+0.08i;

p = [s1 s2 s3 s4];

K = -place(get(sys\_ob, 'A'), get(sys\_ob, 'B'), p);

sys\_reg = ss(K);

sys = lft(sys\_ob,sys\_reg, 1, 4);

C\_sys = [0 0 1 0];

D\_sys = 0;

B\_sys = [ 0;

0;

0;

-K(3)];

set(sys, 'C', C\_sys, 'D', D\_sys, 'B', B\_sys);

hold on

step(sys, 'r')

pole\_sys\_1 = pole(sys)

%% modal regulator 2

s2 = -0.5;

p = [s1 s2 s3 s4];

K = -place(get(sys\_ob, 'A'), get(sys\_ob, 'B'), p);

sys\_reg = ss(K);

sys = lft(sys\_ob,sys\_reg, 1, 4);

B\_sys = [ 0;

0;

0;

-K(3)];

set(sys, 'C', C\_sys, 'D', D\_sys, 'B', B\_sys);

step(sys,'b')

grid on

legend('s\_2 = -0.05','s\_2 = -0.5')

pole\_sys\_2 = pole(sys)