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:

$$\begin{cases} \dot{\beta} = a_{11}\beta + a_{12}\omega + b_1\delta \\ \dot{\omega} = a_{21}\beta + a_{22}\omega + b_2\delta \\ \dot{\varphi} = \omega \\ \dot{\delta} = u \end{cases}$$

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 $u = k1\beta + k2\omega + k3\varphi + k4\delta$.
3. Определить коэффициенты регулятора, обеспечивающие собственные частоты замкнутой системы s_1 = –0.1, s_2 = –0.05, $s_{3,4}$ = –0.025±0.08j
4. Замкнуть систему синтезированным регулятором и проверить корни характеристического полинома.
5. Увеличить s_2 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:

```
sys_ob =

  A = 
              x1         x2         x3         x4
   x1     -0.159      0.267          0    -0.0215
   x2      0.103     -0.188          0    -0.0213
   x3          0          1          0          0
   x4          0          0          0          0

  B = 
       u1
   x1   0
   x2   0
   x3   0
   x4   1

  C = 
       x1  x2  x3  x4
   y1   1   0   0   0
   y2   0   1   0   0
   y3   0   0   1   0
   y4   0   0   0   1

  D = 
       u1
   y1   0
   y2   0
   y3   0
   y4   0

Continuous-time state-space model.
```

Рисунок 1 – Создание модели исследуемой системы

2. Сформируем регулятор с коэффициентами, обеспечивающими обеспечивающие собственные частоты замкнутой системы $s_1 = -0.1$, $s_2 = -0.05$, $s_{3,4} = -0.025 \pm 0.08j$, 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:

```
pole_sys_1 =

  -0.2500 + 0.0800i
  -0.2500 - 0.0800i
  -0.1000 + 0.0000i
  -0.0500 + 0.0000i
```

Рисунок 2 – Полюса замкнутой системы

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

3. Увеличим параметр s_2 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 $s_2 = –0.5$

```
%% 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:

```
pole_sys_2 =

  -0.5000 + 0.0000i
  -0.1000 + 0.0000i
  -0.2500 + 0.0800i
  -0.2500 - 0.0800i
```

Рисунок 3 – Полюса замкнутой системы при $s_2 = -0.5$

На рисунке 4 представлены переходные процессы системы для обоих случаев.

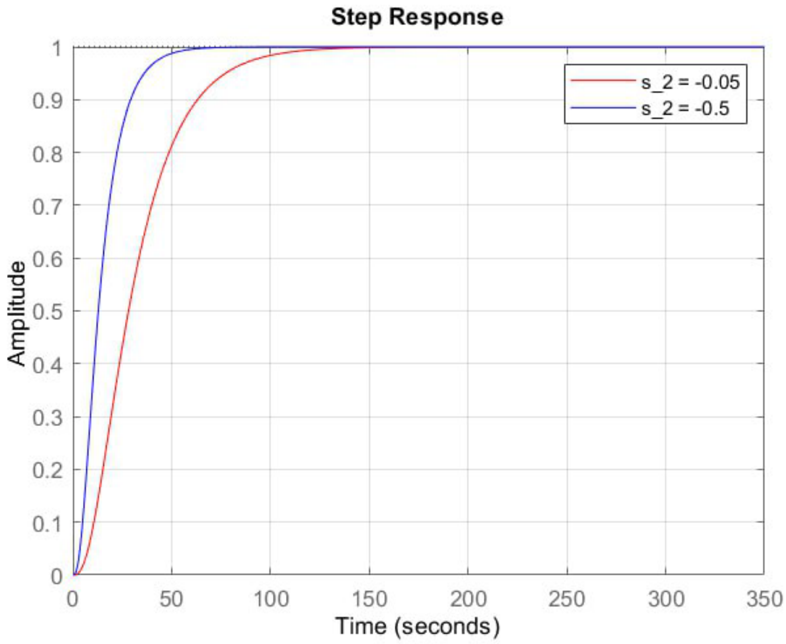


Рисунок 4 – Сравнительный график переходных процессов

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Дана система, которая описывается системой уравнений:

$$\begin{cases} \dot{\omega} = a_{11}\omega + a_{12}\psi + b\delta, \\ \dot{\psi} = \omega, \\ \dot{\delta} = u \end{cases}$$

где $a_{11} = -0.1253, a_{12} = 0.004637, b = -0.002198$. Вектор состояния имеет вид $x = \begin{bmatrix} \omega \\ \psi \\ \delta \end{bmatrix}$, вектор выхода $y = \psi$. Входное воздействие $u = k_1\omega + k_2(\psi_0 - \psi_z) + k_3\delta$. При этом $\psi_z = \psi_0 * \frac{\mathrm{bk}_2 - a_{12}k_3}{\mathrm{bk}_2}$. Необходимо замкнуть систему регулятором и подать на вход воздействие $\psi_0 = 10°$ и построить переходный процесс.

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

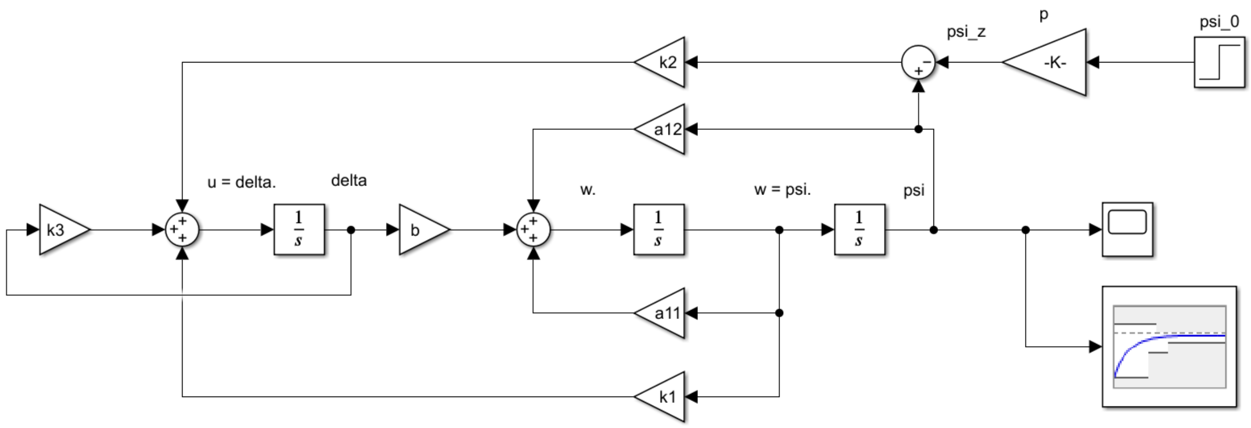


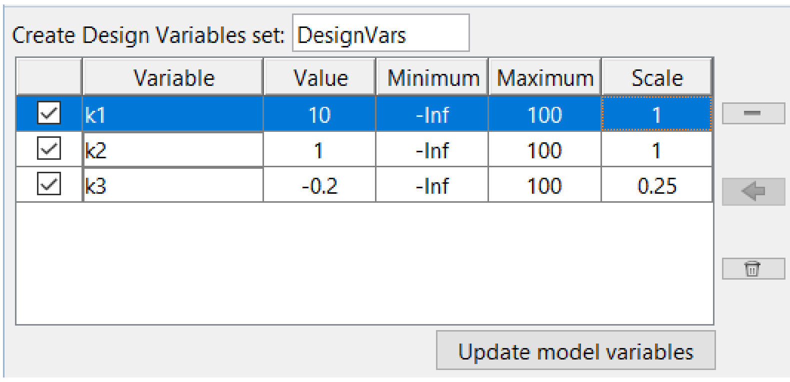
Рисунок 5 - Схема системы в Simulink

2. 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;
```

3. 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.



Create Design Variables set: DesignVars

	Variable	Value	Minimum	Maximum	Scale
☑	k1	10	-Inf	100	1
☑	k2	1	-Inf	100	1
☑	k3	-0.2	-Inf	100	0.25

Update model variables

Рисунок 6 - ограничения переменных коэффициентов регулятора

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

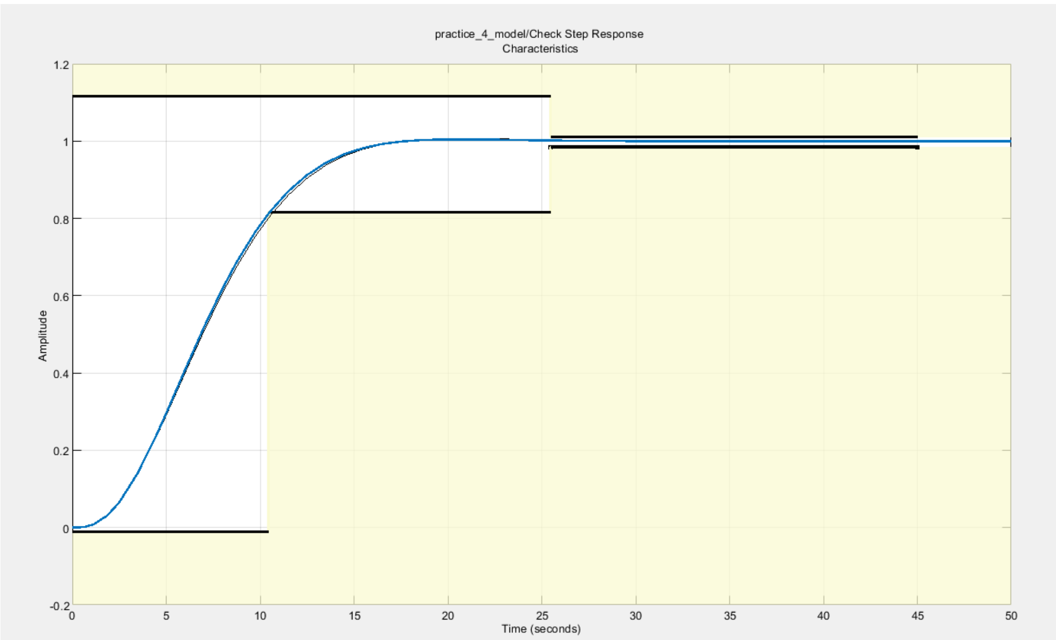


Рисунок 7 - График оптимизированного переходного процесса

Значения коэффициентов регулятора: $k_1 = 98.95$, $k_2 = 18.46$, $k_3 = -0,78$.

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)
```