МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра КСУ

ОТЧЕТ

по лабораторной работе № 3

по дисциплине «Математическое моделирование объектов и систем управления»

Тема: АНАЛИЗ УСТОЙЧИВОСТИ ДИНАМИЧЕСКИХ СИСТЕМ СРЕДСТВАМИ MATLAB

Вариант 5

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

Санкт-Петербург

2023

Цель работы: изучить методы оценки устойчивости линейных динамических систем, освоить машинные средства оценки устойчивости, провести исследование динамических объектов на устойчивость.

Задание на лабораторную работу

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 $u = k_1\beta + k_2\omega + k_3\varphi + k_4\delta$.

3. Провести анализ зависимости степени устойчивости замкнутой системы от величины коэффициента $k2$. Сделать то же самое для запаса устойчивости по амплитуде и по фазе.

Ход работы

1. Формирование LTI - объекта управления реализовано в следующем программном коде (листинг 1):

Листинг 1 – Формирование LTI - объекта

```
clc, clear
a11 = -0.159;
a12 = 0.267;
a21 = 0.103;
a22 = -0.188;
b1 = -0.0215;
b2 = -0.0213;

% 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);
```

2. Формирование регулятора вида $u = k_1\beta + k_2\omega + k_3\varphi + k_4\delta$ будет производится в цикле с итерационным изменением коэффициента k_2. Для оценки степени устойчивости системы с конкретным коэффициентом регулятора будем использовать величину удаления наиболее близкого к мнимой оси корня ХП системы. Так же в цикле итерационно будет строится график зависимости степени устойчивости системы от величины коэффициента k_2. Программный код, реализующий описанный алгоритм представлен в листинге 2.

Листинг 2 – Код итерационного алгоритма

```
hold on
for i = -100:100
    k2 = i;

    % regulator
    k1 = 10; k3 = 5; k4 = -1;
    k = [k1, k2, k3, k4];
    sys_reg = ss(k);

    % closed loop system
    sys = lft(sys_ob,sys_reg);
    C_sys = [0 0 1 0];
    D_sys = 0;
    B_sys = [0; 0; 0; -k3];
    set(sys, 'C', C_sys, 'D', D_sys, 'B', B_sys);
    sys_tf = tf(sys);

    % open loop system
    sys_tf_raz = tf(sys_tf.numerator,(sys_tf.denominator{1,1} -
sys_tf.numerator{1,1}));

    % stability
    [Gm,Pm] = margin(sys_tf_raz);
    figure(1);
    plot(k2,-1*max(real(eig(sys))), '.k')
    figure(2)
    hold on
    plot(k2, Gm, '.r')
    figure(3)
    hold on
    plot(k2, Pm, '.b')

end

% plots config
figure(1)
fplot(0, "red")
xlabel('k_2')
ylabel('Степень устойчивости')
grid on

figure(2)
xlabel('k_2')
ylabel('Запас по амплитуде')
xlim([-100 100])
grid on

figure(3)
xlabel('k_2')
ylabel('Запас по фазе')
grid on
```

В результате работы программы получаем следующий график (рис. 1):

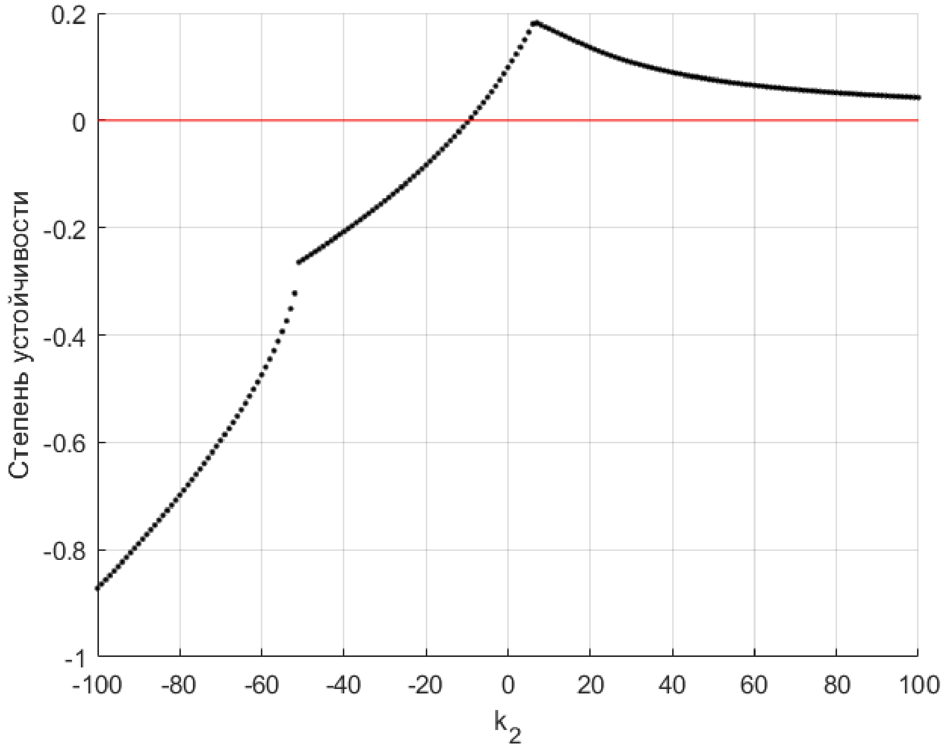

Рисунок 1 - График зависимости степени устойчивости системы от величины коэффициента регулятора

Из графика следует, что коэффициент $k_2 < -9$ выводит систему из зоны устойчивости (корни переходят в левую полуплоскость), а увеличение коэффициента $k_2 \to \infty$ асимптотически приближает систему к границе устойчивости.

По результатам проведенного исследования изменение коэффициента k_2 влияет на запас по фазе и амплитуде так, как показано на графиках на рисунках 2 и 3. На рисунке 2 отображен запас по амплитуде системы, только при $k_2 > -17$. Запас по амплитуде растет линейно с увеличением коэффициента.

Запас по фазе становится положительным только при значениях коэффициента $k_2 > -9$.

Таким образом систему можно считать устойчивой при значениях коэффициента $k_2 > -9$.

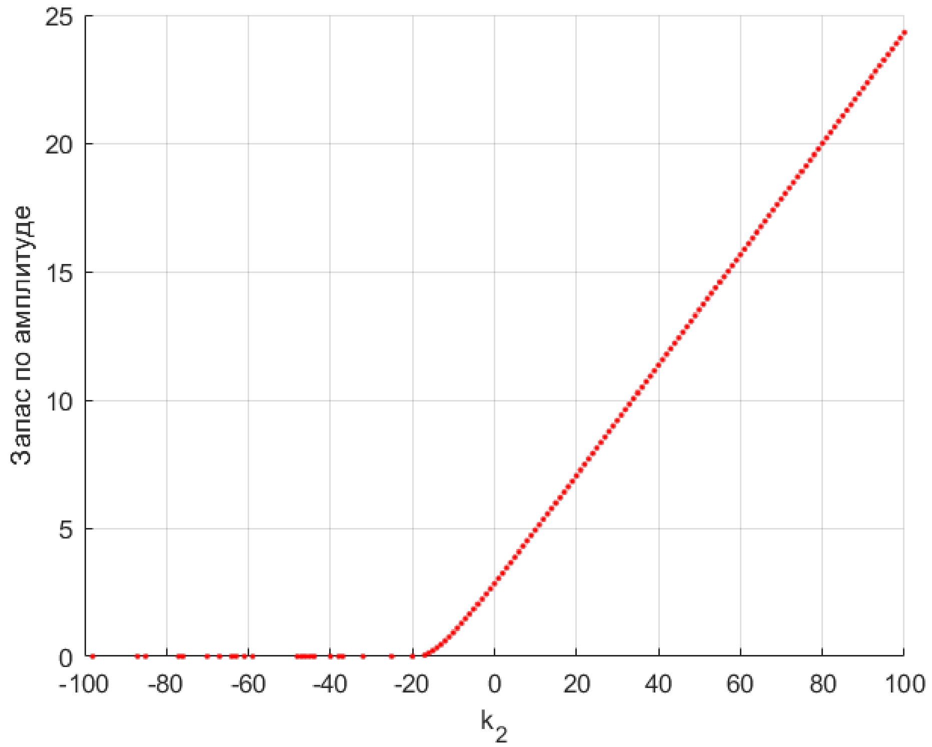

Рисунок 2 - График зависимости запаса по амплитуде системы от величины коэффициента регулятора

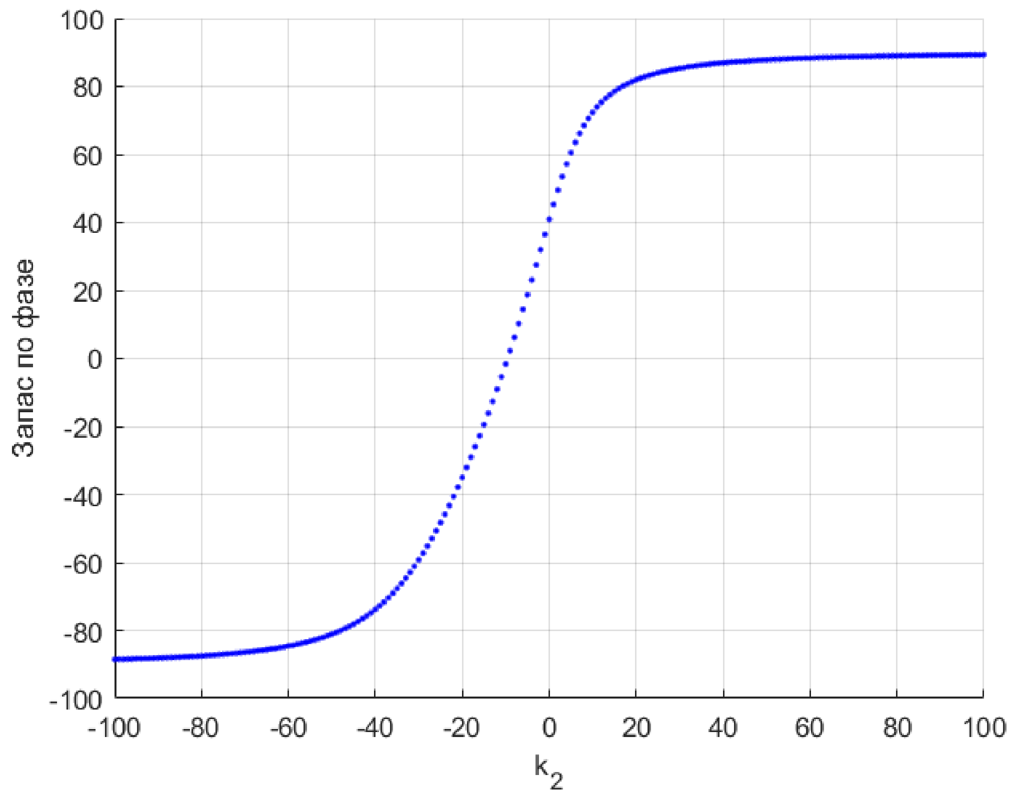

Рисунок 3 - График зависимости запаса по фазе системы от величины коэффициента регулятора

Вывод

В входе выполнения данной лабораторной работы были изучены методы оценки устойчивости линейных динамических систем, освоены машинные средства оценки устойчивости, проведено исследование динамического объекта на устойчивость.

Было получено множество коэффициентов k_2 при которых система имеет запас по фазе, запас по амплитуде и корни ее ХП расположены в левой полуплоскости.

Приложение

Освоение численных методов решения СЛАУ

Задача: написать программу, реализующую метод минимальных невязок.

Описание задачи

Дано: система линейных алгебраических уравнений

$$Au = f,$$

A – известная матрица, f – известный вектор правой части, u – точное решение (недостижимое), обращающее уравнение в тождество.

Вместо точного решения вводится вектор **y** приближенного решения, вычисляемого итерационно.

Основные определения:

Невязка на k-м шагу – мера приближения численного решения к точному:

$$r_k = Ay_k - f \tag{1}$$

Поправка – взвешенная невязка:

$$\omega_k = B^{-1} r_k \tag{2}$$

B – весовая матрица простой структуры (диагональная), при удачном выборе увеличивает скорость метода (по умолчанию можно принять единичной);

Погрешность: ошибка численного решения

$$z_k = y_k - u \tag{3}$$

Описание методов:

1) Двухслойные градиентные методы (ДГМ).

К ДГМ относятся: метод скорейшего спуска, метод минимальных невязок, метод минимальных поправок, метод минимальных погрешностей.

Эти методы имеют общую итерационную формулу:

$$B\frac{y_{k+1} - y_k}{\tau_{k+1}} + Ay_k = f, \quad k = 0, 1, \ldots, \tag{4}$$

метод минимальных невязок:

$$\tau_{k+1} = \frac{(A\omega_k, r_k)}{(A\omega_k, A\omega_k)}, \quad k = 0, 1, \ldots \tag{6}$$

Код функции вычисляющий вектор приближенного значения по методу минимальных невязок представлен в листинге 1.

Листинг 1 – Код функции

```
function [y, i] = min_nevaz (A1, f1, y, B)
    % A1 - system of linear algebraic equations
    % f1 - right vector
    % y  - approximate value vector
    % B  - weight matrix

    switch nargin
        case 2  % default y and B parameters
            B = eye(4);
            y = [6;1;6;1];
        case 3  % default B parameters
            B = eye(4);
        case 4  % got all parameters
            % do nothing
        case 0  % default parameters
            rng(9492)
            A1 = rand(4)*10 - 5;
            f1 = [9;4;9;2];
            B = eye(4);
            y = [6;1;6;1];
    end

    A = A1' * A1;
    f = A1' * f1;

    i = 0;  % iteration counter
    e = 1;  % precision
    while e > 0.001
        r = A * y - f;
        w = B\r;
        t = ((A*w)' * w) / ((B^-1 * A * w)' * (A*w));
        y = y + inv(B) * (f - (A*y))* t;
        e = norm(r);
        i = i + 1;
    end
end
```

В листинге 1 представлен код программы Matlab для тестирования работы вышеописанной функции. В коде программы происходит вызов функции с двумя аргументами и без аргументов, функционал работы с разным числом аргументов реализован и поддерживает от двух до четырех аргументов. Результат выполнения программы представлен в листинге 3.

Листинг 2 – Код программы

```
clc, clear
disp('Вызов функции с двумя аргументами:')
rng(9492)
A = rand(4)*10 - 5;
f = [9;4;9;2];
[y, i] = min_nevaz(A, f)


disp('Вызов функции без аргументов:')
[y, i] = min_nevaz()
```

Листинг 3 – Результат работы программы

```
Вызов функции с двумя аргументами:
y =
  -60.9862
  -42.8966
  -25.9887
   42.6864

i =
       16817

Вызов функции без аргументов:
y =
  -60.9862
  -42.8966
  -25.9887
   42.6864

i =
       16817
```

Изменение матрицы B не влияет на количество итераций вычисления приближенного вектора.