**Министерство образования Российской Федерации**

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ**

**им. Н.Э. БАУМАНА**

Факультет: Информатика и системы управления

Кафедра: Информационная безопасность (ИУ8)

**ОСНОВЫ ТЕОРИИ УПРАВЛЕНИЯ**

**Лабораторная работа №1 на тему:**

«Исследование динамики линейных систем, описываемых обыкновенными дифференциальными уравнениями»

Вариант 4

**Преподаватель:**

Чернега Е.В.

**Студент**:

Девяткин Е.Д.

**Группа:**

ИУ8-44

**Репозиторий работы**: <https://github.com/ledibonibell/Module04-BMT>

Москва 2024

**Цель работы**

Ознакомиться с пакетом моделирования MatLab. Освоить основные приемы моделирования САУ, описываемых при помощи обыкновенных дифференциальных уравнений.

**Порядок выполнения работы**

1. Ознакомиться с пакетом прикладных программ MatLab.
2. Записать дифференциальное уравнение по исходным данным.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Порядок модели |  |  |  |  |  |
| 3 | 5 | 4 | 2 | 1 | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| Порядок модели |  |  |  |
| 3 | 1 | 0.5 | -1 |

1. Привести исходную систему к нормальной форме Коши.
2. Написать скрипт и осуществить моделирование системы дифференциальных уравнений в математическом пакете MatLab , используя численный метод интегрирования Рунге-Кутта и задавая н.у. векторами.
3. Проанализировать системы при двух видах входных воздействий:
4. На каждое из воздействий найти решение для двух различных начальных условий:
   1. нулевые н.у.

На дисплей выводить графики сигналов (синий цвет, сплошная линия) и (зеленый цвет, пунктирная линия). Продолжительность интервалов наблюдения выбрать равной .

**Ход работы**

Исходная система:

А также приведем ее к нормальной форме Коши:

Теперь рассмотрим различные входные воздействия на систему (листинги кодов в конце отчета):

1. Входное воздействие -

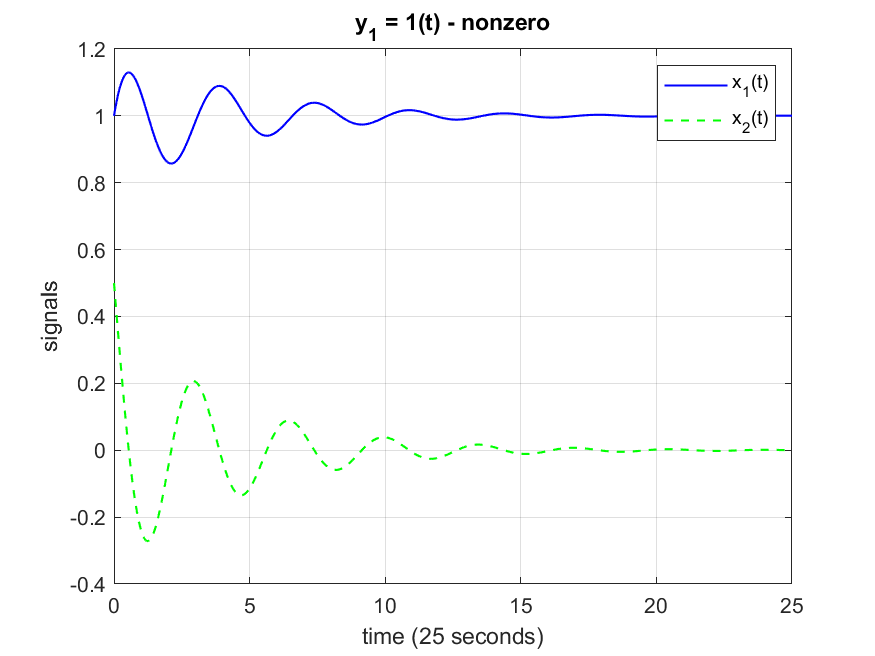


Рис. 1 – Реакция системы на единичное воздействие при ненулевых начальных условиях

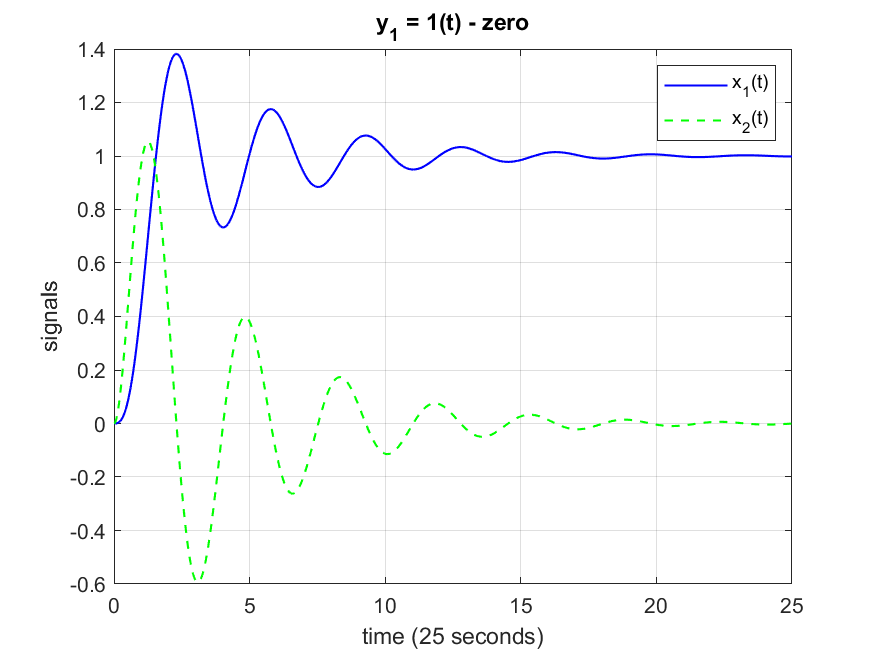


Рис. 2 – Реакция системы на единичное воздействие при нулевых начальных условиях

1. Входное воздействие -

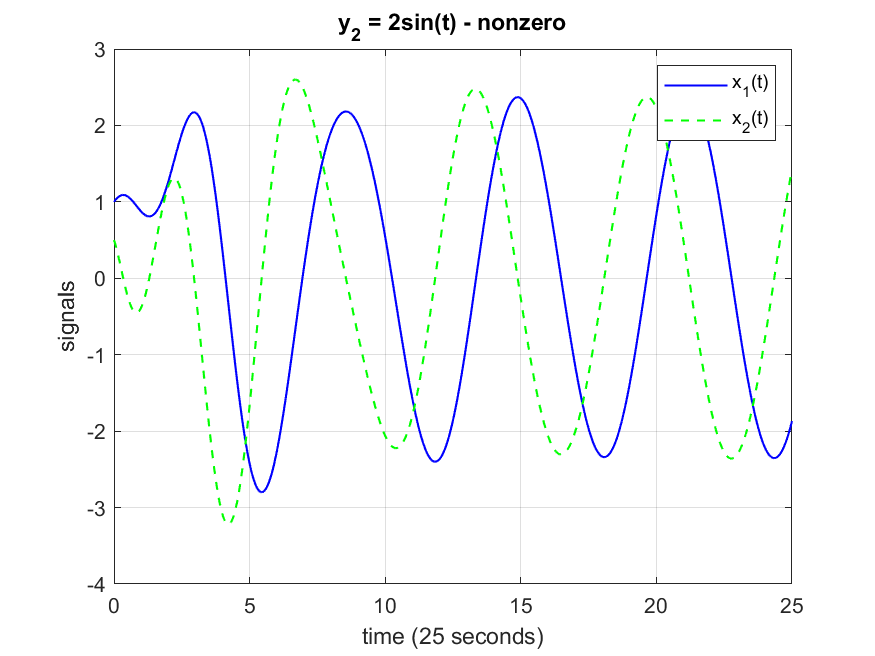


Рис. 3 – Реакция системы на синусоидальное воздействие при ненулевых начальных условиях

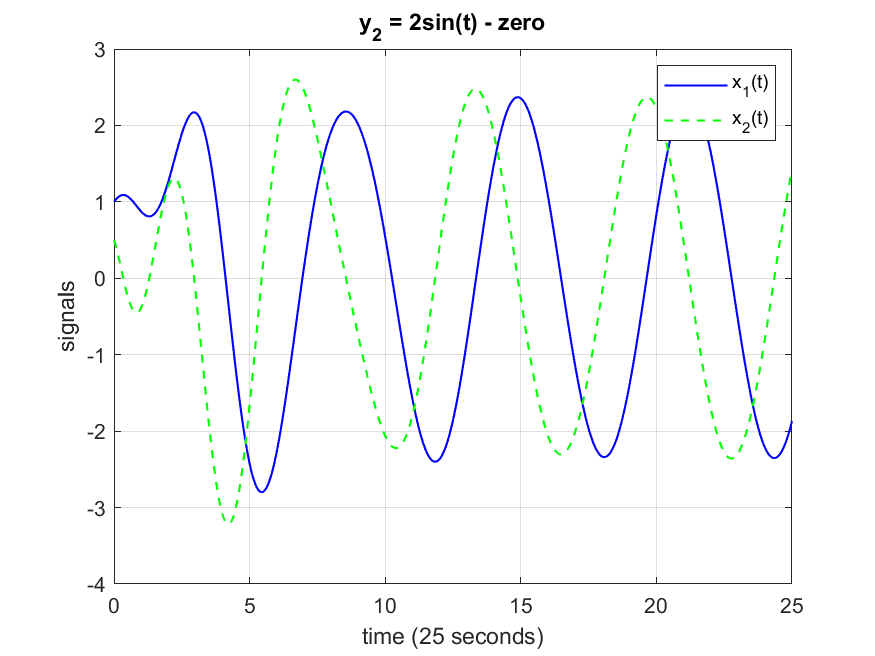


Рис. 4 – Реакция системы на синусоидальное воздействие при нулевых начальных условиях

**Вывод**

В ходе лабораторной работы были получены начальные (базовые) умения работы со средой программирования MatLab и моделирования САУ.

Были решены данные в условии уравнения с помощью метода понижения порядкам дифференциального уравнения, а также смоделирована система д.у. с применением численного метода Рунге-Кутта.

Также был произведен анализ входных воздействий на различные уравнения, с различными начальными условиями, в ходе которого был сделан вывод, о его компенсации функцией и ее последующем возвращении к исходному состоянию.

**Листинг 1**

Приведение дифференциального уравнения к нормальной форме Коши при единичном воздействии:

**options:**

n = 3;

a0 = 5;

a1 = 4;

a2 = 2;

a3 = 1;

b0 = 5;

**model:**

function xp = model(t, x, y)

% x - вектор длины 3

% y - некоторая функция

% t - время

xp = zeros(3, 1);

in = y(t);

xp(1) = x(2);

xp(2) = x(3);

xp(3) = b0 \* in - a2 \* x(3) - a1 \* x(2) - a0 \* x(1);

**Листинг 2.1**

Решение дифференциального уравнения методом Рунге-Кутта при ненулевых начальных условиях и единичном воздействии:

**main:**

ode\_opts = odeset('AbsTol',[1e-5,1e-5,1e-5],'RelTol',1e-5);

y = @(t) 1;

model\_1 = @(t, x) model(t, x, y);

x\_nonzero = [ 1, 0.5, -1];

T = 25;

[time, x] = ode45(model\_1, [0, T], x\_nonzero, ode\_opts);

**Листинг 2.2**

Решение дифференциального уравнения методом Рунге-Кутта при нулевых начальных условиях и единичном воздействии:

**main:**

ode\_opts = odeset('AbsTol',[1e-5,1e-5,1e-5],'RelTol',1e-5);

y = @(t) 1;

model\_1 = @(t, x) model(t, x, y);

x\_zero = [0, 0, 0];

T = 25;

[time, x] = ode45(model\_1, [0, T], x\_zero, ode\_opts);

**Листинг 3**

Приведение дифференциального уравнения к нормальной форме Коши при синусоидальном воздействии:

**options:**

n = 3;

a0 = 5;

a1 = 4;

a2 = 2;

a3 = 1;

b0 = 5;

**model:**

function xp = model(t, x, y)

% x - вектор длины 3

% y - некоторая функция

% t - время

xp = zeros(3, 1);

in = y(t);

xp(1) = x(2);

xp(2) = x(3);

xp(3) = b0 \* in - a2 \* x(3) - a1 \* x(2) - a0 \* x(1);

**Листинг 4.1**

Решение дифференциального уравнения методом Рунге-Кутта при ненулевых начальных условиях и синусоидальном воздействии:

**main:**

ode\_opts = odeset('AbsTol',[1e-5,1e-5,1e-5],'RelTol',1e-5);

y\_sin = @(t) 2 \* sin(t);

model\_sin = @(t, x) model(t, x, y\_sin);

x\_nonzero = [ 1, 0.5, -1];

T = 25;

[time, x] = ode45(model\_sin, [0, T], x\_nonzero, ode\_opts);

**Листинг 4.2**

Решение дифференциального уравнения методом Рунге-Кутта при нулевых начальных условиях и синусоидальном воздействии:

**main:**

ode\_opts = odeset('AbsTol',[1e-5,1e-5,1e-5],'RelTol',1e-5);

y\_sin = @(t) 2 \* sin(t);

model\_sin = @(t, x) model(t, x, y\_sin);

x\_zero = [0, 0, 0];

T = 25;

[time, x] = ode45(model\_sin, [0, T], x\_zero, ode\_opts);

**Листинг программы**

**main.m:**

options;

ode\_opts = odeset('AbsTol',[1e-5,1e-5,1e-5],'RelTol',1e-5);

y = @(t) 1;

y\_sin = @(t) 2 \* sin(t);

model\_1 = @(t, x) model(t, x, y);

model\_sin = @(t, x) model(t, x, y\_sin);

graphic(model\_1, x\_zero, 'y\_1 = 1(t) - zero', 'graphics/signal - 1 - zero.png');

graphic(model\_1, x\_nonzero, 'y\_1 = 1(t) - nonzero', 'graphics/signal - 1 - nonzero.png');

graphic(model\_sin, x\_nonzero, 'y\_2 = 2sin(t) - zero', 'graphics/signal - sin - zero.png')

graphic(model\_sin, x\_nonzero, 'y\_2 = 2sin(t) - nonzero', 'graphics/signal - sin - nonzero.png')

pause

**model.m:**

function xp = model(t, x, y)

% x - vector of length 3

% y - some function

% t - time

options;

disp(t);

xp = zeros(3, 1);

in = y(t);

xp(1) = x(2);

xp(2) = x(3);

xp(3) = b0 \* in - a2 \* x(3) - a1 \* x(2) - a0 \* x(1);

**options.m:**

n = 3;

a0 = 5;

a1 = 4;

a2 = 2;

a3 = 1;

b0 = 5;

x\_zero = [0, 0, 0];

x\_nonzero = [ 1, 0.5, -1];

T = 25; %time of plot's interval

**graphic.m:**

function graphic(model, zero, uptitle, name)

% model - equation with 1(t) or 2 \* sin(t)

% zero - initional conditions

% uptitle - titles

% name - name of plot in directory

options;

ode\_opts = odeset('AbsTol',[1e-5,1e-5,1e-5],'RelTol',1e-5);

[time, x] = ode45(model, [0, T], zero, ode\_opts);

figure;

plot(time, x(:,1), 'b-', time, x(:,2), 'g--', 'LineWidth', 1);

legend('x\_1(t)', 'x\_2(t)');

title(uptitle);

xlabel('time (25 seconds)');

ylabel('signals');

grid on;

saveas(gcf, name);