|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Факультет: «Специальное машиностроение»

Кафедра: «Подводные аппараты и роботы»

**Лабораторная работа № 2**

по курсу «Теория автоматического управления»

Вариант 13

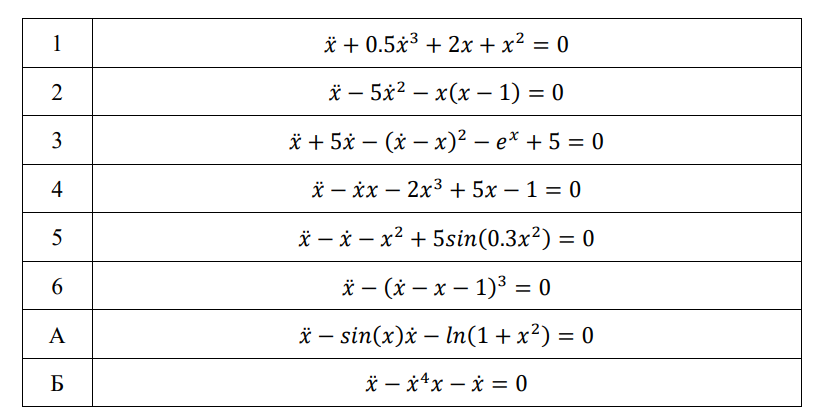
Выполнил: Петров Илья

Группа: СМ11-61Б

Проверил(а):

Москва, 2024 г.

Таблица 1. – Задания для выполнения по вариантам (мой 5)



Работа с фазовыми портретами двумерных систем

Данное задание состоит из трех пунктов:

1) Построить фазовый портрет системы в соответствии с вашим

вариантом:

a. Определить фазовые переменные естественным образом

(𝑥1 = 𝑥; 𝑥2 = 𝑥˙; …);

b. Построить фазовый портрет с помощью кода;

c. Построить фазовый портрет с помощью Simulink модели;

d. Найти особые точки (любым способом) и определить их тип;

e. Получить локальные фазовые портреты в окрестности каждой

особой точки (так, чтобы по нему можно было определить тип

особой точки);

2) Построить фазовый портрет системы с вырожденной особой точкой

(строка А таблицы 1) и попытаться проанализировать его;

3) Построить фазовый портрет системы с континуумом особых точек

(строка Б таблицы 1) и попытаться проанализировать его

.

Условие:



1. Определить фазовые переменные естественным образом

x1 = x , x2 =

1. Построить фазовый портрет с помощью кода

Строим фазовый портрет с помощью Matlab

|  |
| --- |
| clc;  clear all;  close all;  tic; % запуск секундомера  XMAX = 5; % размер сетки рисования  STEP = 0.5; % шаг сетки  TMAX = 10; % время моделирования  BETA = 0.3; % коэффициент вязкого трения  [x1, x2] = meshgrid(-XMAX: STEP: XMAX);  dx = @(t, x) pendulumGrad(t, x, BETA);  event\_out\_of\_bounds = @(t, z) outOfBounds(t, z, XMAX+0.01);  plotLocus(x1, x2, dx, event\_out\_of\_bounds, TMAX); % строим фазовые траектории  plotQuiver(x1, x2, dx); % строим стрелочки  toc; tic; % остановка и запуск секундомера  plotLocus(x1, x2, @vanderpolGrad, event\_out\_of\_bounds, TMAX); % строим фазовые траектории  plotQuiver(x1, x2, @vanderpolGrad); % строим стрелочки  toc; % остановка секундомера |

|  |
| --- |
| function dxdt = pendulumGrad(t,x,BETA)  % функция, возвращающая градиент x(2, :) + x(1, :).^2 - 5 \* sin(0.3 \* x(1, :).^2)  % в ней необходимо задать уравнения системы  dxdt(1, :) = x(2, :);  dxdt(2, :) = x(2, :) + x(1, :).^2 - 5 \* sin(0.3 \* x(1, :).^2);  end |

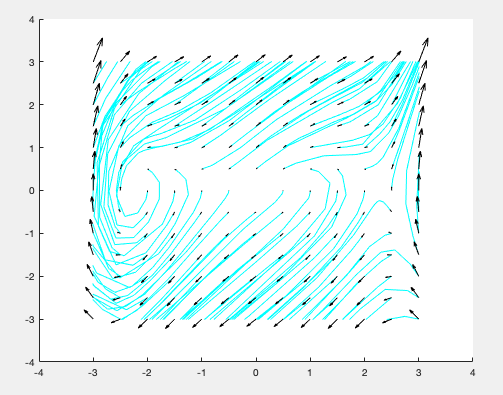


Рис. 1 – фазовый портрет полученный после моделирования

с) Построить фазовый портрет с помощью Simulink модели

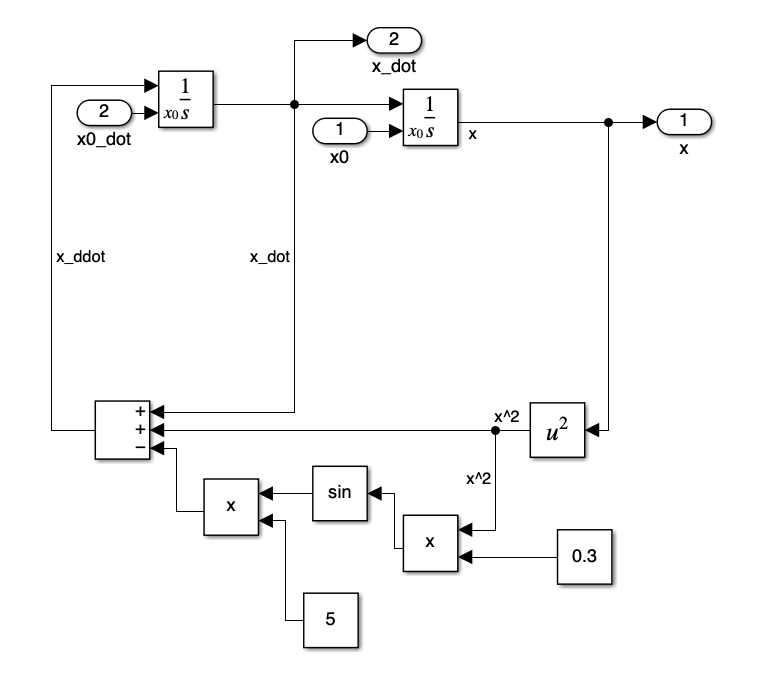


Рис.2 – схема в Simulink дифференциального уравнения

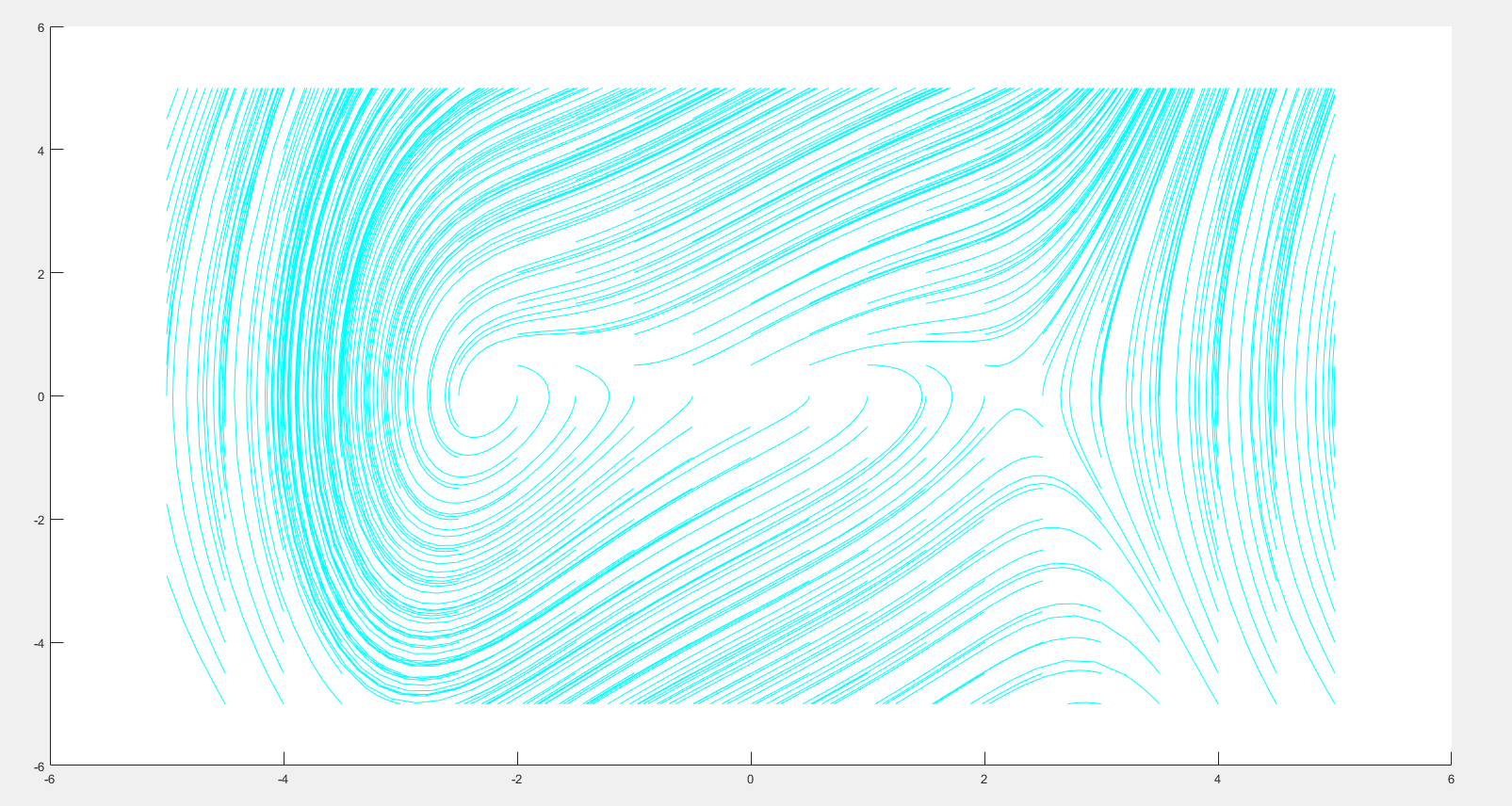


Рис.3 – фазовый портрет, построенный с помощью Simulink

d) Найти особые точки (любым способом) и определить их тип:

Исходное дифференциальное уравнение:

Найдем особые точки ()

:

Получим в результате особые точки: (0;0), (2.23;0), (-2.23;0).

е) Получить локальные фазовые портреты в окрестности каждой

Линеаризуем дифференциальное уравнение около каждой точки и запишем в функцию

Точка (0;0)

|  |
| --- |
| function dxdt = pendulumGrad(t,x,BETA)  % функция, возвращающая градиент % в ней необходимо задать уравнения системы  dxdt(1, :) = x(2, :);  dxdt(2, :) = x(2, :); |

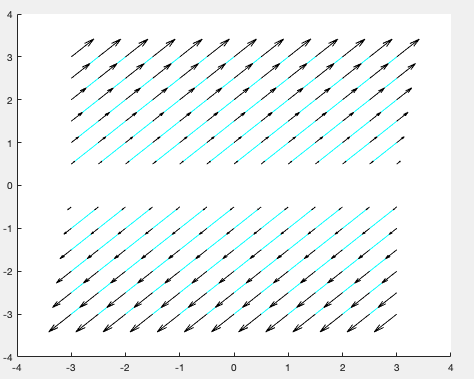


Рис.4 – фазовый портрет в окрестности (0;0)

Точка (2.23;0)

|  |
| --- |
| function dxdt = pendulumGrad(t,x,BETA)  % функция, возвращающая градиент  % в ней необходимо задать уравнения системы  dxdt(1, :) = x(2, :);  dxdt(2, :) = 3.96 .\* x(1, :) + x(2, :);  end |

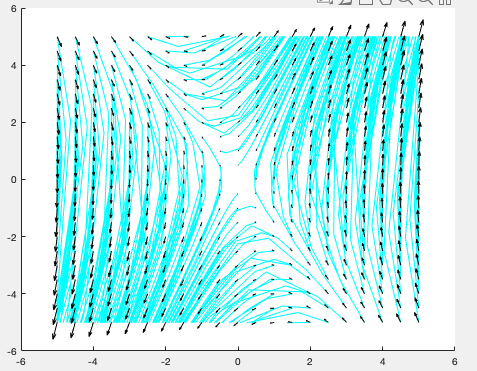


Рис.5 - фазовый портрет в окрестности (2.23;0)

Точка (-2.23;0)

|  |
| --- |
| function dxdt = pendulumGrad(t,x,BETA)  % функция, возвращающая градиент  % в ней необходимо задать уравнения системы  dxdt(1, :) = x(2, :);  dxdt(2, :) = - 3.96 .\* x(1, :) + x(2, :);  end |

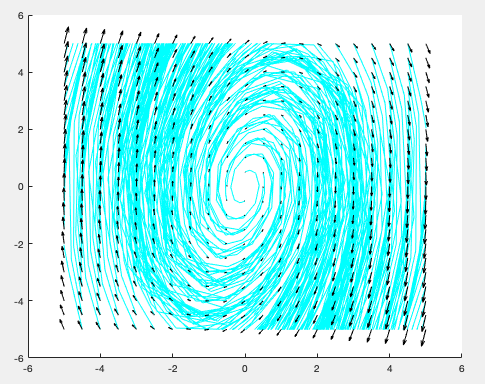


Рис.6 - фазовый портрет в окрестности (-2.23;0)

2) Уравнение А

|  |
| --- |
| function dxdt = pendulumGrad(t,x,BETA)  % функция, возвращающая градиент  % в ней необходимо задать уравнения системы  dxdt(1, :) = x(2, :);  dxdt(2, :) = sin(x(1, :)).\* x(2, :) + log(1 + x(1, :).^ 2);  end |

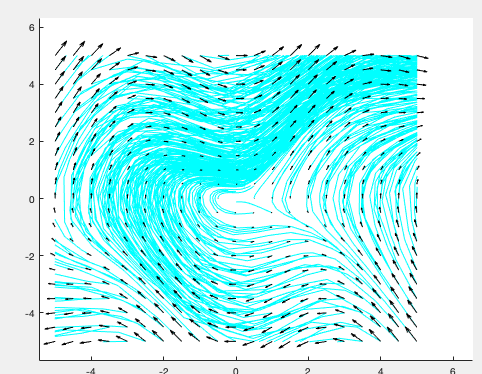


Рис.7 - фазовый портрет А

3) Уравнение Б

|  |
| --- |
| function dxdt = pendulumGrad(t,x,BETA)  % функция, возвращающая градиент  % в ней необходимо задать уравнения системы  dxdt(1, :) = x(2, :);  dxdt(2, :) = (x(2, :).^ 4).\* x(1, :) - x(2, :);  end |

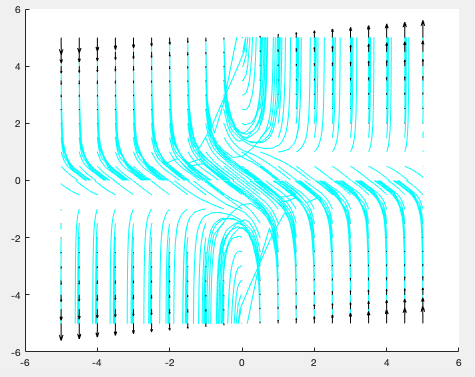


Рис.8 - фазовый портрет Б