**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САУ**

отчет

**по лабораторной работе № 2**

**по дисциплине «Современные методы теории управления»**

Тема: **ПОСТРОЕНИЕ ФАЗОВОГО ПОРТРЕТА СИСТЕМЫ ВТОРОГО ПОРЯДКА**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9492 |  | Викторов А.Д. |
| Преподаватель |  | Бельский Г.В. |

Санкт-Петербург

2023

**Цель работы:** нахождение особых точек системы второго порядка, определение их типа и построение фазового портрета с помощью Matlab.

**Ход работы**

Работа заключается в реализации двух независимых программ, первая из которых служит для нахождения особых точек системы второго порядка и определения их типа, а вторая предназначена для построения фазового портрета.

Первая программа является дополнением для функции, реализованной в первой лабораторной работе. Функция автоматически определяет порядок системы, по переданным в качестве аргумента уравнениям, поддерживаются системы первого и второго порядка.

Вторая программа является отдельной функцией и способна реализовать построение фазового портрета, принимая на вход координаты точек равновесия, систему в виде функции и стиль оформления.

Полный код первой функции представлен в листинге 1, второй функции в листинге 2.

*Листинг 1 – Исходный код первой функции*

function [points, type] = special\_points(func, arg)

point\_struct = struct2cell(solve(func, arg, 'Real',true, 'ReturnConditions',true)); % roots structure

switch size(arg,2)

case 1

points = point\_struct{1, 1};

if size(point\_struct{2, 1}, 2) > 0 % check if func is periodic

period\_counter = 0; % periodic roots counter

j = 0; % integer iterator for periodic roots

points = [];

while period\_counter < size(point\_struct{1, 1},1)

for i = 1:size(point\_struct{1, 1},1)

if subs(point\_struct{1, 1}(i),point\_struct{2, 1},j) < 2\*pi && subs(point\_struct{1, 1}(i),point\_struct{2, 1},j) >= 0 % check if the root within the peroid

points = [points; subs(point\_struct{1, 1}(i),point\_struct{2, 1},j)]; % append new roots

else

if subs(point\_struct{1, 1}(i),point\_struct{2, 1},j) >= 0

period\_counter = period\_counter + 1;

end

end

end

j = j + 1;

end

end

points = eval(points);

df = diff(func);

type = subs(df, arg, points);

for i = 1:size(type,1)

if eval(type(i)) < 0

type(i) = 'stable';

else

type(i) = 'unstable';

end

end

case 2

points = [point\_struct{1, 1} point\_struct{2, 1}];

points = eval(points);

A(1,1) = diff(func(1),arg(1));

A(1,2) = diff(func(1),arg(2));

A(2,1) = diff(func(2),arg(1));

A(2,2) = diff(func(2),arg(2));

a = [0; 0];

for j = 1:size(points,1)

A1 = subs(A,arg(1),points(j,1));

A1 = eval(subs(A1,arg(2),points(j,2)));

a(:,j) = eig(A1);

end

eigenvalues = a';

type = string(zeros(size(eigenvalues, 1),1));

for i = 1:size(eigenvalues,1)

if imag(eigenvalues(i,1)) == 0 % check if imaginary part is zero

if real(eigenvalues(i,1)) \* real(eigenvalues(i,2)) < 0

type(i) = 'saddle';

else

if real(eigenvalues(i,1)) >= 0

*Листинг 1 – Продолжение*

type(i) = 'unstable node';

else

type(i) = 'stable node';

end

end

else

if real(eigenvalues(i,1)) == 0

type(i) = 'center';

else

if real(eigenvalues(i,1)) >= 0

type(i) = 'unstable focus';

else

type(i) = 'stable focus';

end

end

end

end

end

end

*Листинг 2 – Исходный код второй функции*

function portrait\_plotter(sys, points, style)

n = 20;

plot(points(:,1),points(:,2), '.', 'MarkerSize', 15, color = 'b')

hold on

switch style

case "points"

Tspan = [0 1];

options = odeset('RelTol',1e-1,'AbsTol',1e-1);

for i = linspace(min(points(:,1)) - 1, max(points(:,1)) + 1, n)

for j = linspace(min(points(:,2)) - 1, max(points(:,2)) + 1, n)

X0 = [i, j];

[~, x] = ode45(sys,Tspan,X0, options);

plot(x(:,1),x(:,2), ':', color = 'r');

end

end

xlim([min(points(:,1)) - 1 max(points(:,1)) + 1])

ylim([min(points(:,2)) - 1 max(points(:,2)) + 1])

case "arrows"

for i = linspace(min(points(:,1)) - 1, max(points(:,1)) + 1, n)

for j = linspace(min(points(:,2)) - 1, max(points(:,2)) + 1, n)

X0 = [i, j];

u = sys(0, X0);

quiver(i, j, u(1), u(2), 0.05, 'b', AutoScale='on')

axis equal

end

end

xlim([min(points(:,1)) - 1 max(points(:,1)) + 1])

ylim([min(points(:,2)) - 1 max(points(:,2)) + 1])

end

end

В качестве демонстрации в листинге 3 приведен пример использования обеих функций. На рисунке 1 показан фазовый портрет, полученный с помощью соответствующей функции.

*Листинг 3 – Пример вызова функций*

clc, clear, close all

syms x y

x = [x y];

sys = @ (t,x) [x(1)^2 - x(2)^2 - 5; x(1)^2 + x(2)^2 - 13];

f = [x(1)^2 - x(2)^2 - 5; x(1)^2 + x(2)^2 - 13];

[points, type] = special\_points(f, x)

portrait\_plotter(sys, points, "points")

hold on

portrait\_plotter(sys, points, "arrows")

> > >

points =

-3 -2

3 -2

-3 2

3 2

type =

"stable focus"

"saddle"

"saddle"

"unstable focus"

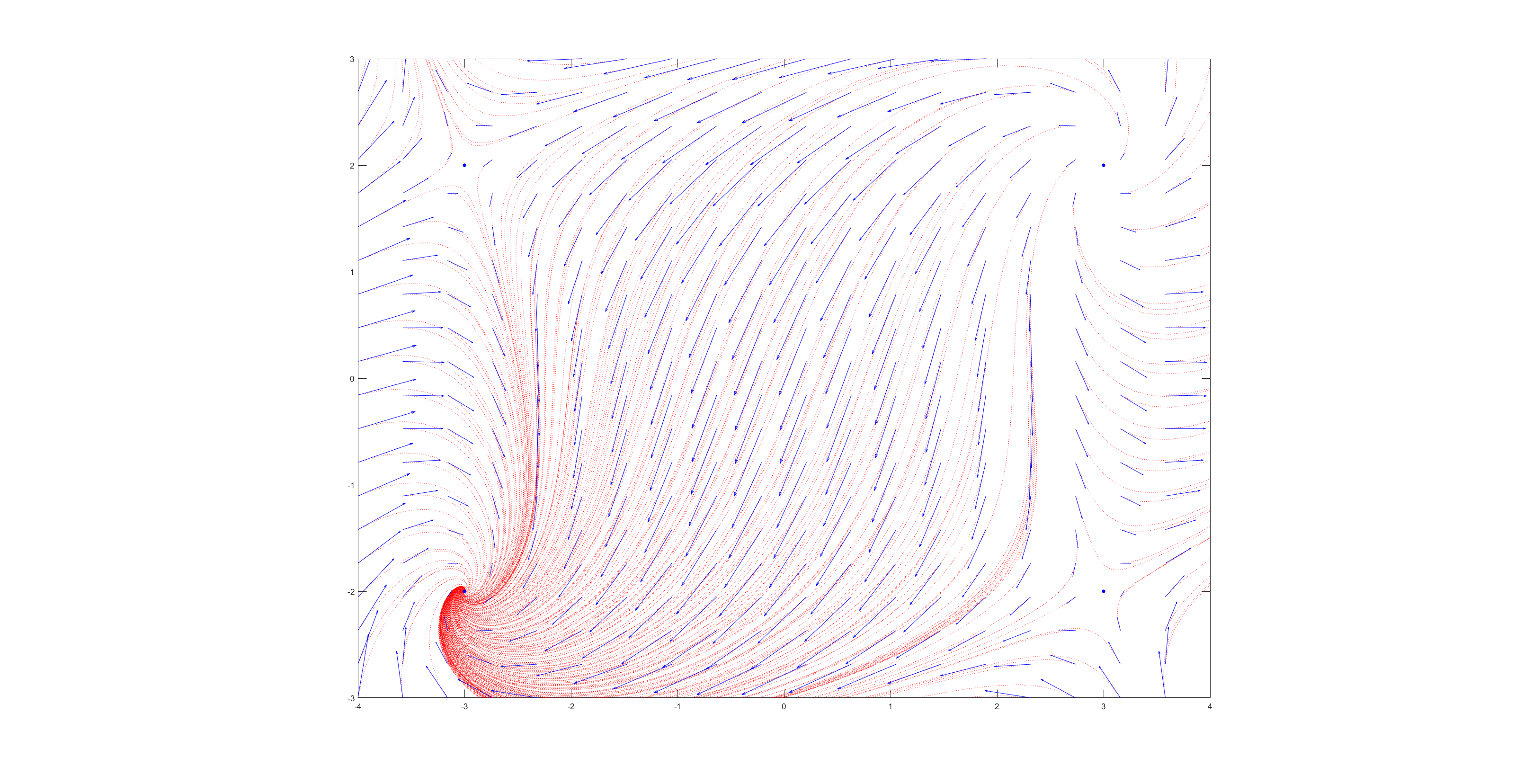


Рисунок - Фазовый портрет системы

**Вывод**

В ходе выполнения данной лабораторной работы была реализована функция, определяющая координаты и тип особых точек системы второго порядка, функция, изображающая фазовый портрет системы а также продемонстрирована работа реализованных функций.