MECH 6313 - Homework 1

Jonas Wagner

2021, February 1

1 Problem 1

Problem: Let

$$\begin{aligned} \dot{x}_1 &= -x + x_2 \\ \dot{x}_2 &= \frac{x_1^2}{1 + x_1^2} - 0.5x_2 \end{aligned} \tag{1}$$

Define an shifted system, linerize that system, and find the center manifold to analyze the stability properties. Then use numerical simulation to plot the phase portrait of the original coordinates and superimpose the shifted center manifold.

Solution:

1.1 Part a

Let a shifted set of state variables be defined as $\bar{x}_1 = x_1 - 1$ and $\bar{x}_2 = x_2 - 1$. The state variable equation can then be rewritten as

$$kkk \tag{2}$$

A MATLAB Code:

All code I write in this course can be found on my GitHub repository:
https://github.com/jonaswagner2826/MECH6313

Script 1: MECH6313_HW3

```
%% MECH6313 - HW 3
clear
close all

pblm1 = false;
pblm2 = false;
pblm3 = false;




if pblm1
%% Problem 1
% using ode 45 instead....
parta = true;
partb = true;
partc = true;

if parta
%% Problem 1a
% System Def
sys_func = @pblm1a;
Parms = 0.1 * [-1, 1e-10, 1];

% Simulation Setup
T = [0 100];
X_0 = 0.5 * [1, 1, -1, -1; 1, -1, 1, -1];
% X_0 = [ -0.5,0.8, -1.5, 3;
% 0.5, -0.5, 2.7, -1.9];

% Sim Phase Plots
fig = figure('position',[0,0,1500,500]);
N1 = size(Parms,2);
N2 = size(X_0,2);
simNum = 1;
for i = 1:N1
    ax(i) = subplot(1,N1,i);
    parms = Parms(i);
```

```
    for j = 1:N2
        [t,y] = ode45(@(t,y) sys_func(t,y,parms),T,X_0(:,j));
        plot(y(:,1),y(:,2));
        xlabel('x1')
        ylabel('x2')
        title(['\alpha = ', num2str(round(parms,3))])
        hold on
        simNum = simNum + 1;
    end
end
linkaxes(ax,'xy')

sgtitle('Problem 1a - Phase Portrait For Varying Parameters')
saveas(fig,fullfile([pwd '\\' 'HW2' '\\' 'fig'],'pblm1a.png'))

end

if partb
%% Problem 1b
% System Def
sys_func = @pblm1b;
Parms = 0.1 * [-1, 1e-10, 1];

% Simulation Setup
T = [0 100];
X_0 = 0.05 * [1, 1, -1, -1; 1, -1, 1, -1];

% Sim Phase Plots
fig = figure('position',[0,0,1500,500]);
N1 = size(Parms,2);
N2 = size(X_0,2);
simNum = 1;
for i = 1:N1
    ax(i) = subplot(1,N1,i);
    parms = Parms(i);
    for j = 1:N2
        [t,y] = ode45(@(t,y) sys_func(t,y,parms),T,X_0(:,j));
        plot(y(:,1),y(:,2));
        xlabel('x1')
        ylabel('x2')
        title(['\alpha = ', num2str(round(parms,3))])
        hold on
        simNum = simNum + 1;
```

```
    end
end
linkaxes([ax(1),ax(2)],'xy')


sgtitle('Problem 1b - Phase Portrait For Varying Parameters')
saveas(fig,fullfile([pwd '\\' 'HW2' '\\' 'fig'],'pblm1b.png'))

end

if partc
%% Problem 1c
% System Def
sys_func = @pblm1c;
Parms = 0.5 * [-1, 1];

% Simulation Setup
T = [0 10];
X_0 = 0.5 * [1, 1, -1, -1; 1, -1, 1, -1];

% Sim Phase Plots
fig = figure('position',[0,0,1000,500]);
N1 = size(Parms,2);
N2 = size(X_0,2);
simNum = 1;
for i = 1:N1
    ax(i) = subplot(1,N1,i);
    parms = Parms(i);
    for j = 1:N2
        [t,y] = ode45(@(t,y) sys_func(t,y,parms),T,X_0(:,j));
        plot(y(:,1),y(:,2));
        xlabel('x1')
        ylabel('x2')
        title(['\alpha = ', num2str(round(parms,3))])
        hold on
        simNum = simNum + 1;
    end
    if ax(i).XLim(1) < -5
        ax(i).XLim(1) = -5;
    end
    if ax(i).XLim(2) > 5
        ax(i).XLim(2) = 5;
    end
```

```
    if ax(i).YLim(1) < -5
        ax(i).YLim(1) = -5;
    end
    if ax(i).YLim(2) > 30
        ax(i).YLim(2) = 30;
    end
end


sgtitle('Problem 1c - Phase Portrait For Varying Parameters')
saveas(fig,fullfile([pwd '\\' 'HW2' '\\' 'fig'],'pblm1c.png'))
end
end

if pblm2
%% Problem 2
parta = true;

if parta
%% Problem 2a
disp('____________________ Problem 2: ____________________')
% sys def
sys2a = nlsys(@pblm2a)

syms x1 x2
linsys2a_sym = sys2a.linearize([x1;x2])
linsys2a = sys2a.linearize([0;0])

end
end

if pblm3
%% Problem 3
% Problem 2.20.2
syms x1 x2
A2 = [3 * x1^2 + x2^2 - 1, 2 * x1 * x2;
    2 * x1 * x2, 3 * x2^2 + x1^2 - 1]
eigA2 = eig(A2)
% x2 = sqrt((2 - 4 * x1^2)/4);
eigA2_B = subs(eigA2, x2, sqrt((2 - 4 * x1^2)/4))

% Problem 2.20.3
syms x1 x2
```

```
A3 = [-x2^2, -2 * x1 * x2; 1, 0]
eigA3 = eig(A3)
eigA3_B = subs(eigA3, x2, 0)

% Problem 2.20.4
syms x1 x2
A4 = [x2, x1; 0, 1]
eigA4 = eig(A4)
eigA4_B = subs(eigA4, x2, -1)

% Problem 2.20.4
syms x1 x2
A5 = [-x2 * sin(x1), cos(x1); cos(x1), 0]
eigA5 = eig(A5)
eigA5_B0 = subs(eigA5, [x1,x2],[0,0])
eigA5_B1 = subs(eigA5, [x1,x2],[pi/2,0])

end
%% Local Functions
function dx = pblm1a(t, x, parms)
    % pblm1a function
    arguments
        t (1,1) = 0;
        x (2,1) = [0; 0];
        parms = false;
    end

    if parms == false
        alpha = 1;
    else
        alpha = parms(1);
    end

    % State Upadate Eqs
    dx(1,1) = alpha * x(1) + x(2);
    dx(2,1) = - x(1) + alpha*x(2) - x(1)^2 * x(2);
end

function y = pblm1b(t,x,parms)
    % pblm1b function
    arguments
        t (1,1) = 0;
        x (2,1) = [0; 0];
```

```
        parms = false;
    end

    if parms == false
        alpha = 1;
    else
        alpha = parms(1);
    end

    % State Upadate Eqs
    y(1,1) = alpha * x(1) + x(2) - x(1)^3;
    y(2,1) = - x(1) + alpha*x(2) + 2 *x(2)^3;
end

function y = pblm1c(t,x,parms)
    % pblm1c function
    arguments
        t (1,1) = 0;
        x (2,1) = [0; 0];
        parms = false;
    end

    if parms == false
        alpha = 1;
    else
        alpha = parms(1);
    end
    % State Upadate Eqs
    y(1,1) = alpha * x(1) + x(2) - x(1)^2;
    y(2,1) = - x(1) + alpha*x(2) + 2 * x(1)^2;
end

function y = pblm2a(x,u)
    % pblm2 function
    arguments
        x (2,1) = [0; 0];
        u (1,1) = 0;
    end

    % Array Sizes
    n = 2;
    p = 1;

```

```
    % State Upadate Eqs
    y(1,1) = x(2) + x(1) * x(2)^2;
    y(2,1) = - x(1) + x(1)^2 * x(2);

    if nargin == 0
        y = [n;p];
    end
end
```