

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

```
1 %% MECH6313 - HW 3
2 clear
3 close all
4
5 pblm1 = false;
6 pblm2 = false;
7 pblm3 = false;
8
9
10
11
12 if pblm1
13 %% Problem 1
14 % using ode 45 instead....
15 parta = true;
16 partb = true;
17 partc = true;
18
19 if parta
20 %% Problem 1a
21 % System Def
22 sys_func = @pblm1a;
23 Params = 0.1 * [-1, 1e-10, 1];
24
25 % Simulation Setup
26 T = [0 100];
27 X_0 = 0.5 * [1, 1, -1, -1; 1, -1, 1, -1];
28 % X_0 = [ -0.5, 0.8, -1.5, 3;
29 % 0.5, -0.5, 2.7, -1.9];
30
31 % Sim Phase Plots
32 fig = figure('position', [0, 0, 1500, 500]);
33 N1 = size(Params, 2);
34 N2 = size(X_0, 2);
35 simNum = 1;
36 for i = 1:N1
37     ax(i) = subplot(1, N1, i);
38     parms = Params(i);
```

```

39     for j = 1:N2
40         [t,y] = ode45(@(t,y) sys_func(t,y,parms),T,X_0(:,j));
41         plot(y(:,1),y(:,2));
42         xlabel('x1')
43         ylabel('x2')
44         title(['\alpha = ', num2str(round(parms,3))])
45         hold on
46         simNum = simNum + 1;
47     end
48 end
49 linkaxes(ax,'xy')
50
51 sgtitle('Problem 1a - Phase Portrait For Varying Parameters')
52 saveas(fig,fullfile([pwd '\\ ' 'HW2' '\\ ' 'fig'],'pblm1a.png'))
53
54 end
55
56 if partb
57     %% Problem 1b
58     % System Def
59     sys_func = @pblm1b;
60     Params = 0.1 * [-1, 1e-10, 1];
61
62     % Simulation Setup
63     T = [0 100];
64     X_0 = 0.05 * [1, 1, -1, -1; 1, -1, 1, -1];
65
66     % Sim Phase Plots
67     fig = figure('position',[0,0,1500,500]);
68     N1 = size(Params,2);
69     N2 = size(X_0,2);
70     simNum = 1;
71     for i = 1:N1
72         ax(i) = subplot(1,N1,i);
73         parms = Params(i);
74         for j = 1:N2
75             [t,y] = ode45(@(t,y) sys_func(t,y,parms),T,X_0(:,j));
76             plot(y(:,1),y(:,2));
77             xlabel('x1')
78             ylabel('x2')
79             title(['\alpha = ', num2str(round(parms,3))])
80             hold on
81             simNum = simNum + 1;

```

```

82     end
83 end
84 linkaxes([ax(1),ax(2)],'xy')
85
86
87 sgtitle('Problem 1b - Phase Portrait For Varying Parameters')
88 saveas(fig,fullfile([pwd '\\ 'HW2' '\\ 'fig'],'pblm1b.png'))
89
90 end
91
92 if partc
93 %% Problem 1c
94 % System Def
95 sys_func = @pblm1c;
96 Params = 0.5 * [-1, 1];
97
98 % Simulation Setup
99 T = [0 10];
100 X_0 = 0.5 * [1, 1, -1, -1; 1, -1, 1, -1];
101
102 % Sim Phase Plots
103 fig = figure('position',[0,0,1000,500]);
104 N1 = size(Params,2);
105 N2 = size(X_0,2);
106 simNum = 1;
107 for i = 1:N1
108     ax(i) = subplot(1,N1,i);
109     parms = Params(i);
110     for j = 1:N2
111         [t,y] = ode45(@(t,y) sys_func(t,y,parms),T,X_0(:,j));
112         plot(y(:,1),y(:,2));
113         xlabel('x1')
114         ylabel('x2')
115         title(['\alpha = ', num2str(round(parms,3))])
116         hold on
117         simNum = simNum + 1;
118     end
119     if ax(i).XLim(1) < -5
120         ax(i).XLim(1) = -5;
121     end
122     if ax(i).XLim(2) > 5
123         ax(i).XLim(2) = 5;
124     end

```

```

125     if ax(i).YLim(1) < -5
126         ax(i).YLim(1) = -5;
127     end
128     if ax(i).YLim(2) > 30
129         ax(i).YLim(2) = 30;
130     end
131 end
132
133
134 sgtitle('Problem 1c - Phase Portrait For Varying Parameters')
135 saveas(fig,fullfile([pwd '\\' 'HW2' '\\' 'fig'],'pblm1c.png'))
136 end
137 end
138
139 if pblm2
140 %% Problem 2
141 parta = true;
142
143 if parta
144 %% Problem 2a
145 disp('----- Problem 2: -----')
146 % sys def
147 sys2a = nlsys(@pblm2a)
148
149 syms x1 x2
150 linsys2a_sym = sys2a.linearize([x1;x2])
151 linsys2a = sys2a.linearize([0;0])
152
153 end
154 end
155
156 if pblm3
157 %% Problem 3
158 % Problem 2.20.2
159 syms x1 x2
160 A2 = [3 * x1^2 + x2^2 - 1, 2 * x1 * x2;
161       2 * x1 * x2, 3 * x2^2 + x1^2 - 1]
162 eigA2 = eig(A2)
163 % x2 = sqrt((2 - 4 * x1^2)/4);
164 eigA2_B = subs(eigA2, x2, sqrt((2 - 4 * x1^2)/4))
165
166 % Problem 2.20.3
167 syms x1 x2

```

```

168 A3 = [-x2^2, -2 * x1 * x2; 1, 0]
169 eigA3 = eig(A3)
170 eigA3_B = subs(eigA3, x2, 0)
171
172 % Problem 2.20.4
173 syms x1 x2
174 A4 = [x2, x1; 0, 1]
175 eigA4 = eig(A4)
176 eigA4_B = subs(eigA4, x2, -1)
177
178 % Problem 2.20.4
179 syms x1 x2
180 A5 = [-x2 * sin(x1), cos(x1); cos(x1), 0]
181 eigA5 = eig(A5)
182 eigA5_B0 = subs(eigA5, [x1,x2],[0,0])
183 eigA5_B1 = subs(eigA5, [x1,x2],[pi/2,0])
184
185 end
186 %% Local Functions
187 function dx = pblm1a(t, x, parms)
188     % pblm1a function
189     arguments
190         t (1,1) = 0;
191         x (2,1) = [0; 0];
192         parms = false;
193     end
194
195     if parms == false
196         alpha = 1;
197     else
198         alpha = parms(1);
199     end
200
201     % State Update Eqs
202     dx(1,1) = alpha * x(1) + x(2);
203     dx(2,1) = - x(1) + alpha*x(2) - x(1)^2 * x(2);
204 end
205
206 function y = pblm1b(t,x,parms)
207     % pblm1b function
208     arguments
209         t (1,1) = 0;
210         x (2,1) = [0; 0];

```

```

211     parms = false;
212 end
213
214 if parms == false
215     alpha = 1;
216 else
217     alpha = parms(1);
218 end
219
220 % State Upadate Eqs
221 y(1,1) = alpha * x(1) + x(2) - x(1)^3;
222 y(2,1) = - x(1) + alpha*x(2) + 2 *x(2)^3;
223 end
224
225 function y = pblm1c(t,x,parms)
226 % pblm1c function
227 arguments
228     t (1,1) = 0;
229     x (2,1) = [0; 0];
230     parms = false;
231 end
232
233 if parms == false
234     alpha = 1;
235 else
236     alpha = parms(1);
237 end
238 % State Upadate Eqs
239 y(1,1) = alpha * x(1) + x(2) - x(1)^2;
240 y(2,1) = - x(1) + alpha*x(2) + 2 * x(1)^2;
241 end
242
243 function y = pblm2a(x,u)
244 % pblm2 function
245 arguments
246     x (2,1) = [0; 0];
247     u (1,1) = 0;
248 end
249
250 % Array Sizes
251 n = 2;
252 p = 1;
253

```

```
254     % State Upadate Eqs
255     y(1,1) = x(2) + x(1) * x(2)^2;
256     y(2,1) = - x(1) + x(1)^2 * x(2);
257
258     if nargin == 0
259         y = [n;p];
260     end
261 end
```