

# MECH 6313 - Homework 1

Jonas Wagner

2021, February 1

## 1 Problem 1

**Problem:** Let

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

Define an shifted system, linearize 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

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

This system can then be linearized about the origin, resulting in the system matrix

$$A = \begin{bmatrix} -1 & 1 \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}\tag{3}$$

whose eigenvalues are calculated as  $\lambda_{1,2} = 0, -\frac{3}{2}$ .

A transformation matrix

$$T = \begin{bmatrix} 1 & -2 \\ 1 & 1 \end{bmatrix}$$

can then be constructed with the associated eigenvectors to covert into

## 2 Problem 2 - S 3.7.3

**Problem:** A simple model of a fishery is given as

$$\dot{N} = rN(1 - \frac{N}{K}) - H \quad (4)$$

where  $N$  represents the fish population,  $H > 0$  is the number of fish harvested at a constant rate, and both  $r$  and  $K$  are constants.

Redefine the model in terms of  $x$ ,  $\tau$ , and  $h$ . Then plot the vector field for various values of  $h$ . Then identify  $h_c$  and classify and discuss the bifurcation.

**Solution:**

### 2.1 Part a

Let  $x = N/K$ , this can then be substituted as

$$\dot{N} = \frac{dN}{dt} = \frac{dx}{dt} = r(Kx)(1 - x) - H \quad (5)$$

$$\frac{1}{rK} \frac{dx}{dt} = x(1 - x) - \frac{H}{rK} \quad (6)$$

Let  $h = \frac{H}{rK}$  and  $\tau = rKt$ ,

$$\frac{dx}{d\tau} = x(1 - x) - h \quad (7)$$

### 2.2 Part b

Plotting in matlab

### 2.3 Part c

As is evident by observing the vector fields shown in ??, and the bifurcation diagram in ??, there exists an  $h_c$  where the bifurcation occurs.

### 2.4 Part d

The long time behavior of this model

### 3 Problem 3 - S 3.7.4

**Problem:** An improved model of a fishery is given as

$$\dot{N} = rN\left(1 - \frac{N}{K}\right) - H\frac{N}{A + N} \quad (8)$$

where  $N$  represents the fish population,  $H > 0$  is the number of fish harvested at a constant rate, and both  $r$ ,  $K$ ,  $A$  are constants.

Define the biological interpretation of the parameter  $A$ . Redefine the model in terms of  $x$ ,  $\tau$ , and  $h$ . Find and analyze various fixed points depending on the values of  $a$  and  $h$ . Then analyze the bifurcation that occurs when  $h = a$ . Then find and classify the other bifurcation that occurs at  $h = \frac{1}{4}(a + 1)^2$  for  $a < a_c$ . Finally plot the stability diagram for the system for  $(a, h)$ .

**Solution:**

#### 3.1 Part a

When a population of fish is being fished, there is a portion of fish that are not possible to catch (such as eggs or fish that are too old).

#### 3.2 Part b

Let  $x = N/K$ , this can then be substituted as

$$\dot{N} = \frac{dN}{dt} = \frac{dx}{dt} = r(Kx)(1 - x) - H\frac{Kx}{A + Kx} \quad (9)$$

$$\frac{1}{rK} \frac{dx}{dt} = x(1 - x) - \frac{H}{rK} \frac{Kx}{A + Kx} \quad (10)$$

$$= x(1 - x) - \frac{H}{rK} \frac{Kx}{K\left(\frac{A}{K} + x\right)} \quad (11)$$

$$= x(1 - x) - \frac{H}{rK} \frac{x}{\frac{A}{K} + x} \quad (12)$$

Let  $h = \frac{H}{rK}$ ,  $\tau = rKt$ , and  $a = \frac{A}{K}$

$$\frac{dx}{d\tau} = x(1 - x) - h\frac{x}{a + x} \quad (13)$$

#### 3.3 Part c

## 4 Problem 4 - K 3.8

**Problem:** Let the following system be defined:

$$\begin{aligned}\dot{x}_1 &= -x_1 + \frac{2x_2}{1+x_2^2}, \quad x_1(0) = a \\ \dot{x}_2 &= -x_2 + \frac{2x_1}{1+x_1^2}, \quad x_2(0) = b\end{aligned}\tag{14}$$

Show that this system has a unique solution for all  $t \geq 0$ .

**Solution:**

The system is known to be continuous on its domain. It is also apparent that both functions are differentiable, which results in a Jacobian of

$$\begin{bmatrix} -1 & \frac{2}{x_2^2+1} - \frac{4x_2^2}{(x_2^2+1)^2} \\ \frac{2}{x_1^2+1} - \frac{4x_1^2}{(x_1^2+1)^2} & -1 \end{bmatrix}\tag{15}$$

which indicates the systems dynamics are both differentiable and differential bounded. This also implies that the system is globally Lipschitz continuous, therefore a unique solution exists for  $t \geq 0$ .

## 5 Problem 5 - K 3.13

**Problem:** Let the following system be defined:

$$\begin{aligned}\dot{x}_1 &= \tan^{-1}(ax_1) - x_1x_2 \\ \dot{x}_2 &= bx_1^2 - cx_2\end{aligned}\tag{16}$$

Derive the sensitivity equations for the parameters vary from their nominal values of  $a_0 = 1$ ,  $b_0 = 0$ , and  $c_0 = 1$ . Then simulate the sensitivity equations and the time dependence for the initial conditions of  $x_1(0) = 1$  and  $x_2(0) = -1$ .

**Solution:**

### 5.1 Part a - Sensitivity Calculation

Let the following be defined:

$$\mu = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Let the trajectory  $x(\mu, t)$  be defined with regards to parameter changes as:

$$x(\mu, t) = x(\bar{\mu}, t) + \left. \frac{\partial x}{\partial \mu} \right|_{\bar{\mu}} \tilde{\mu}\tag{17}$$

where  $\tilde{\mu} = \mu - \bar{\mu}$ .

It can also be defined by its nonlinear definition as:

$$x(\mu, t) = x_0 + \int_0^t \dot{x}(x(\mu, \tau), \mu, \tau) d\tau\tag{18}$$

The sensativity to the parameters can then be formulated

$$S(t) = \frac{\partial x}{\partial \mu} = 0 + \frac{\partial}{\partial \mu} \int_0^t f(x(\mu, \tau), \mu, \tau) d\tau\tag{19}$$

$$= \int_0^t \frac{\partial}{\partial \mu} f(x(\mu, \tau), \mu, \tau) d\tau\tag{20}$$

$$= \int_0^t \frac{\partial f}{\partial x} \frac{\partial x}{\partial \mu} \frac{\partial f}{\partial \mu} d\tau\tag{21}$$

which can be clculated as jacobians of  $f$  resulting in

$$S(t) = \int_0^t A(\tau)S(\tau) + B(\tau) d\tau\tag{22}$$

where the matrices  $A(\tau)$  and  $B(\tau)$  are the Jacobians with respect to  $x$  and  $\mu$  respectively:

$$\begin{aligned}A(\tau) &= \left. \frac{\partial f}{\partial x} \right|_{\bar{\mu}} & B(\tau) &= \left. \frac{\partial f}{\partial \mu} \right|_{\bar{\mu}} \\ &= \begin{bmatrix} -x_2 + \frac{x_1}{x_1^2 + 1} & -x_1 \\ 0 & 1 \end{bmatrix} & &= \begin{bmatrix} \frac{x_1}{x_1^2 + 1} & 0 & 0 \\ 0 & x_1^2 & -x_2 \end{bmatrix}\end{aligned}\tag{23}$$

Finally, the evolution of sensitivity over time can be found using the Leibnitz Formula, resulting in

$$\frac{ds(t)}{dt} = A(t)S(t) + B(t)\tag{24}$$

## 5.2 Part b - Simulation

The MATLAB code in AppendixA simulates and plots the states and sensitivities for each of the parameters. These plots can be seen in Figure1.

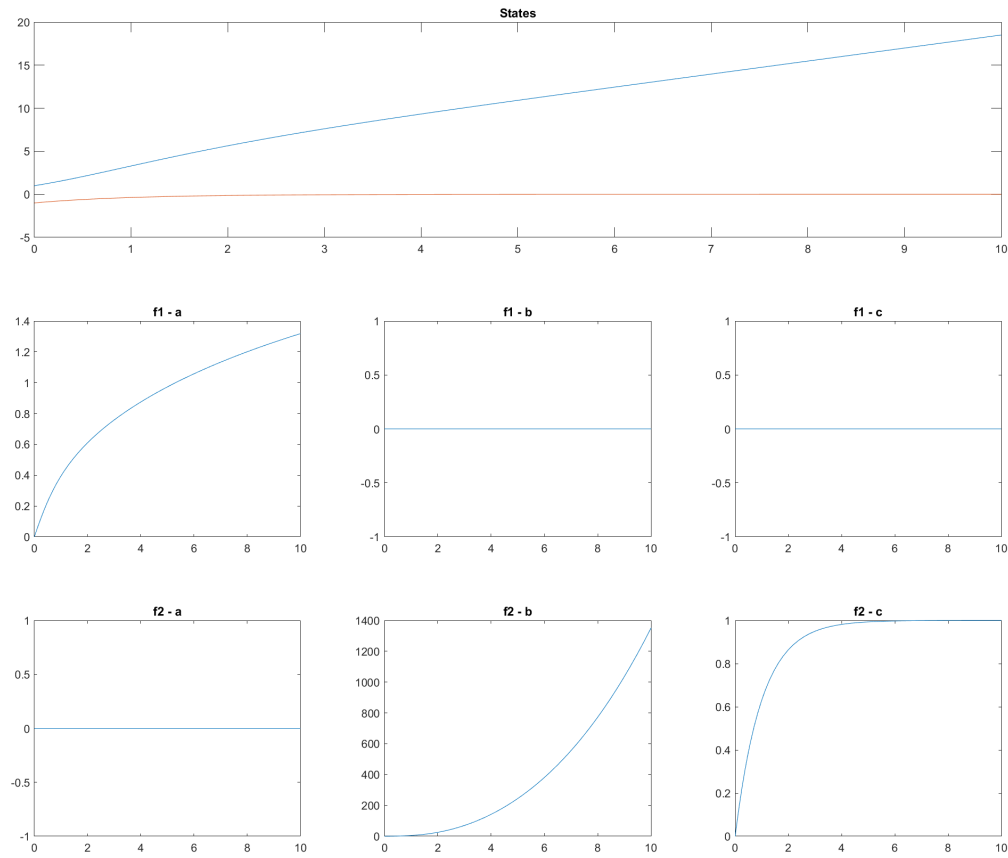


Figure 1: Simulation for Problem 5 with the evolution of sensitivities to parameters.

## 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 pblm4 = false;
9 pblm5 = true;
10
11
12 if pblm1
13 %% Problem 1
14 syms x_1 x_2
15 x_1_dot = -x_1 + x_2
16 x_2_dot = (x_1^2)/(1 + x_1^2) - 0.5 * x_2
17
18 % part a
19 syms x_1_bar x_2_bar
20
21 x_1_bar_dot = subs(x_1_dot, [x_1, x_2], [x_1_bar + 1, x_2_bar + 1]);
22 x_2_bar_dot = subs(x_2_dot, [x_1, x_2], [x_1_bar + 1, x_2_bar + 1]);
23
24 x_bar = [x_1_bar; x_2_bar];
25 x_bar_dot = [x_1_bar_dot; x_2_bar_dot]
26
27 % part b
28 % Linearize
29 A_sym = jacobian(x_bar_dot, x_bar)
30 A = subs(A_sym, [x_1_bar, x_2_bar], [0,0])
31
32 [T1, eig_A] = eig(A)
33
34 % Transform
35 syms y_sym z_sym
36 x_bar_sub = [y_sym, z_sym] * T1;
37 y_dot = subs(x_1_bar_dot, [x_1_bar, x_2_bar], x_bar_sub);
38 z_dot = subs(x_2_bar_dot, [x_1_bar, x_2_bar], x_bar_sub);
```

```

39
40 % G function Definitions
41 g1 = y_dot;
42 g2 = z_dot + 3/2 * z_sym;
43 'g1'
44 pretty(g1)
45 'g2'
46 pretty(g2)
47
48 % Coefficients of eigenvalue matrix
49 A1 = eig_A(1,1);
50 A2 = eig_A(2,2);
51
52 % w_dot substitution (from definition equation)
53 syms h_sym dh_sym
54 w_dot = A2 * h_sym + subs(g2,z_sym,h_sym) - dh_sym * (A1 * y_sym + subs(g1,z_sym,h_sym));
55
56 % Taylor's Series approximation of manifold
57 syms h2 h3
58 h = h2 * y_sym^2;% + h3 * y_sym^3;
59 dh = diff(h,y_sym);
60
61 % Attempting to solve for the h2 value...
62 w_dot = expand(subs(w_dot, [h_sym, dh_sym], [h, dh]));
63 'w_dot'
64 pretty(w_dot)
65
66 w_dot_soln = expand(w_dot);
67 w_dot_soln = subs(w_dot_soln, y_sym^4, 0);
68 w_dot_soln = subs(w_dot_soln, y_sym^3, 0);
69 syms y2
70 w_dot_soln = subs(w_dot_soln, y_sym^2, y2);
71 w_dot_soln = subs(w_dot_soln, y_sym, 0);
72 'w_dot_soln'
73 pretty(w_dot_soln)
74
75 syms h2y2
76 w_dot_soln = subs(w_dot_soln, h2*y2, h2y2)
77 solve(w_dot_soln == 0,h2y2)
78
79
80 end
81

```



```

82
83
84 if pblm4
85 %% Problem 4
86 syms x1 x2
87 x1_dot = -x1 + (2*x2)/(1 + x2^2);
88 x2_dot = -x2 + (2*x1)/(1 + x1^2);
89 f = [x1_dot; x2_dot];
90 'f'
91 pretty(f)
92 df = jacobian(f);
93 'jacobian'
94 pretty(df)
95
96
97
98
99 end
100
101
102 if pblm5
103 %% Problem 5
104 syms x1 x2 a b c
105 x1_dot = atan(a * x1) - x1 * x2;
106 x2_dot = b * x1^2 - c * x2;
107 x_dot = [x1_dot; x2_dot];
108 'x_dot'
109 pretty(x_dot)
110
111 x = [x1; x2];
112 'x'
113 pretty(x)
114
115 mu = [a; b; c];
116 mu_bar = [1; 0; 1];
117 'mu'
118 pretty(mu)
119
120
121 A_tau = jacobian(x_dot, x)
122 B_tau = jacobian(x_dot, mu)
123
124

```

```

125 sys_func = @pblm5_func;
126
127 T = [0,10];
128 x_0 = [1,-1, 0,0,0,0,0,0]';
129
130 [t,y] = ode45(@(t,y) sys_func(t,y,mu_bar,A_tau,B_tau),T,x_0);
131
132 y_states = y(:,[1,2]);
133 y_a = y(:,[3,6]);
134 y_b = y(:,[4,7]);
135 y_c = y(:,[5,8]);
136
137 fig = figure('position',[0,0,1500,1200]);
138 subplot(3,3,[1:3])
139 plot(t,y_states)
140 title('States')
141
142 titles = ["f1 - a", "f1 - b", "f1 - c", "f2 - a", "f2 - b", "f2 - c"];
143 for i = 3:8
144     subplot(3,3,i+1)
145     plot(t,y(:,i))
146     title(titles(i-2))
147 end
148
149 saveas(fig,fullfile([pwd '\\ ' 'HW3' '\\ ' 'fig'],'pblm5.png'))
150
151 end
152
153
154
155
156
157 %% Local Functions
158 function dx = pblm5_func(t, x, parms, A, B)
159     % pblm5 function
160     arguments
161         t (1,1) = 0;
162         x (8,1) = zeros(8,1); %state and 6 sensitivities
163         parms = false;
164         A = 0;
165         B = 0;
166     end
167

```

```

168     if parms == false
169         a = 1;
170         b = 0;
171         c = -1;
172     else
173         a = parms(1);
174         b = parms(2);
175         c = parms(3);
176     end
177
178     % Variable Decode
179     x1 = x(1);
180     x2 = x(2);
181     S = zeros(2,3); %[x(3), x(4), x(5); x(6), x(7), x(8)];
182
183     % State Upadate Eqs
184     x1_dot = atan(a * x1) - x1 * x2;
185     x2_dot = b * x1^2 - c * x2;
186     S_dot = subs(A * S + B);
187
188     % Variable Encode
189     dx = x;
190     dx(1) = x1_dot;
191     dx(2) = x2_dot;
192     dx(3) = S_dot(1,1);
193     dx(4) = S_dot(1,2);
194     dx(5) = S_dot(1,3);
195     dx(6) = S_dot(2,1);
196     dx(7) = S_dot(2,2);
197     dx(8) = S_dot(2,3);
198 end

```