# MECH 6313 - Homework 3

Jonas Wagner

2021, March 13

## 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, 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

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

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 using

$$\begin{bmatrix} y \\ z \end{bmatrix} = T^{-1}x$$

to transform into the diagonalized system

$$\begin{aligned}\dot{y} &= A_1 y + g_1(y,z) \\ \dot{z} &= A_2 z + g_2(y,z)\end{aligned} \tag{4}$$

where $A_1 = 0$, $A_2 = -\frac{3}{2}$, $g_1(y,z) = 3z$, and $g_2(y,z) = \dfrac{(y-2z+1)^2}{(y-2z+1)^2+1} + \dfrac{2z-y+1}{2}$.

An invariant manifold can then be defined as

$$\omega = z - h(y)$$

with $\dot{\omega}$ calculated as

$$\dot{\omega} = \dot{z} - \frac{\partial h}{\partial y}\dot{y}$$

To satisfy invarience, $z = h(y)$, which implies $\dot{\omega} = \omega = 0$. This implies that for an invarient manifold to exist the following must be true:

$$\dot{\omega} = 0 = A_2 h(y) + g_2(y, h(y)) - \frac{\partial h}{\partial y}[A_1 y + g_1(y, h(y))] \tag{5}$$

## 2 Problem 2 - S 3.7.3

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

$$\dot{N} = rN(1 - \frac{N}{K}) - H \tag{6}$$

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

$$\dot{N} = \frac{\mathrm{d}N}{\mathrm{d}t} = \frac{\mathrm{d}x}{\mathrm{d}t} = r(Kx)(1-x) - H \tag{7}$$

$$\frac{1}{rK}\frac{\mathrm{d}x}{\mathrm{d}t} = x(1-x) - \frac{H}{rK} \tag{8}$$

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

$$\frac{\mathrm{d}x}{\mathrm{d}\tau} = x(1-x) - h \tag{9}$$

### 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

$$\dot{N} = rN(1 - \frac{N}{K}) - H\frac{N}{A+N} \tag{10}$$

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

$$\begin{aligned}
\dot{N} = \frac{\mathrm{d}N}{\mathrm{d}t} = \frac{\mathrm{d}x}{\mathrm{d}t} &= r(Kx)(1-x) - H\frac{Kx}{A+Kx} && (11)\\
\frac{1}{rK}\frac{\mathrm{d}x}{\mathrm{d}t} &= x(1-x) - \frac{H}{rK}\frac{Kx}{A+Kx} && (12)\\
&= x(1-x) - \frac{H}{rK}\frac{Kx}{K\left(\frac{A}{K}+x\right)} && (13)\\
&= x(1-x) - \frac{H}{rK}\frac{x}{\frac{A}{K}+x} && (14)
\end{aligned}$$

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

$$\frac{\mathrm{d}x}{\mathrm{d}\tau} = x(1-x) - h\frac{x}{a+x} \tag{15}$$

### 3.3 Part c

# 4 Problem 4 - K 3.8

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

$$\begin{aligned} \dot{x}_1 &= -x_1 + \frac{2x_2}{1+x_2^2},\ x_1(0) = a \\ \dot{x}_2 &= -x_2 + \frac{2x_1}{1+x_1^2},\ x_2(0) = b \end{aligned} \tag{16}$$

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

$$\begin{bmatrix} -1 & \frac{2}{x_2^2+1} - \frac{4x_2^2}{\left(x_2^2+1\right)^2} \\ \frac{2}{x_1^2+1} - \frac{4x_1^2}{\left(x_1^2+1\right)^2} & -1 \end{bmatrix} \tag{17}$$

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

# 5 Problem 5 - K 3.13

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

$$\begin{aligned}\dot{x}_1 &= \tan^{-1}(ax_1) - x_1x_2 \\ \dot{x}_2 &= bx_1^2 - cx_2\end{aligned} \tag{18}$$

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:

$$x(\mu, t) = x(\bar{\mu}, t) + \left.\frac{\partial x}{\partial \mu}\right|_{\bar{\mu}} \tilde{\mu} \tag{19}$$

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

It can also be defined by its nonlinear definition as:

$$x(\mu, t) = x_0 + \int_0^t \dot{x}(x(\mu, \tau), \mu, \tau)\, \mathrm{d}\tau \tag{20}$$

The sensativity to the parameters can then be formulated

$$\begin{aligned} S(t) = \frac{\partial x}{\partial \mu} &= 0 + \frac{\partial}{\partial \mu} \int_0^t f(x(\mu, \tau), \mu, \tau)\, \mathrm{d}\tau && (21) \\ &= \int_0^t \frac{\partial}{\partial \mu} f(x(\mu, \tau), \mu, \tau)\, \mathrm{d}\tau && (22) \\ &= \int_0^t \frac{\partial f}{\partial x} \frac{\partial x}{\partial \mu} \frac{\partial f}{\partial \mu}\, \mathrm{d}\tau && (23) \end{aligned}$$

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

$$S(t) = \int_0^t A(\tau)S(\tau) + B(\tau)\, \mathrm{d}\tau \tag{24}$$

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

$$\begin{aligned} A(\tau) &= \left.\frac{\partial f}{\partial x}\right|_{\bar{\mu}} & B(\tau) &= \left.\frac{\partial f}{\partial \mu}\right|_{\bar{\mu}} \\ &= \begin{bmatrix} -x_2 + \dfrac{x_1}{x_1^2 + 1} & -x_1 \\ 0 & 1 \end{bmatrix} & &= \begin{bmatrix} \dfrac{x_1}{x_1^2 + 1} & 0 & 0 \\ 0 & x_1^2 & -x_2 \end{bmatrix} \end{aligned} \tag{25}$$

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

$$\frac{\mathrm{d}s(t)}{\mathrm{d}t} = A(t)S(t) + B(t) \tag{26}$$

## 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.

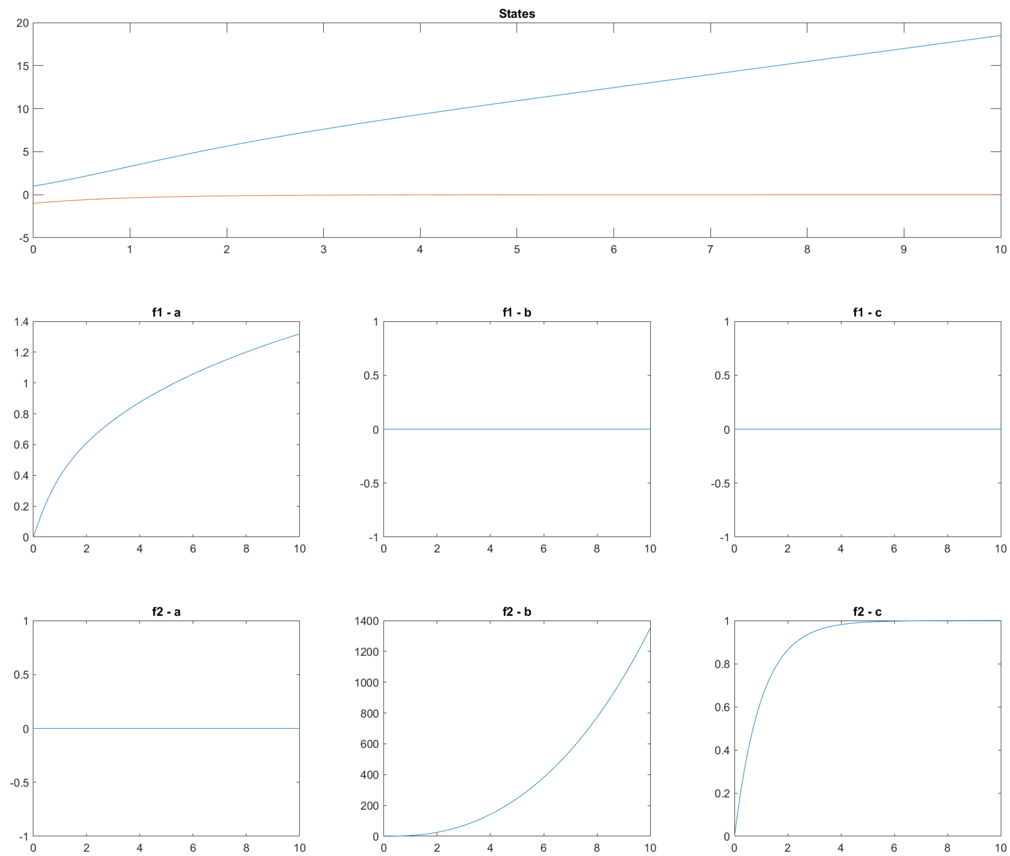


Figure 1: Simulation for Problem 5 with the evolution of sensitivities to parameters.

# 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

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

pblm1 = false;
pblm2 = false;
pblm3 = false;
pblm4 = false;
pblm5 = true;


if pblm1
%% Problem 1
syms x_1 x_2
x_1_dot = -x_1 + x_2
x_2_dot = (x_1^2)/(1 + x_1^2) - 0.5 * x_2

% part a
syms x_1_bar x_2_bar

x_1_bar_dot = subs(x_1_dot, [x_1, x_2], [x_1_bar + 1, x_2_bar + 1]);
x_2_bar_dot = subs(x_2_dot, [x_1, x_2], [x_1_bar + 1, x_2_bar + 1]);

x_bar = [x_1_bar; x_2_bar];
x_bar_dot = [x_1_bar_dot; x_2_bar_dot]

% part b
% Linearize
A_sym = jacobian(x_bar_dot, x_bar)
A = subs(A_sym, [x_1_bar, x_2_bar], [0,0])

[T1, eig_A] = eig(A)

% Transform
syms y_sym z_sym
x_bar_sub = [y_sym, z_sym] * T1;
y_dot = subs(x_1_bar_dot, [x_1_bar, x_2_bar], x_bar_sub);
z_dot = subs(x_2_bar_dot, [x_1_bar, x_2_bar], x_bar_sub);
```

```

% G function Definitions
g1 = y_dot;
g2 = z_dot + 3/2 * z_sym;
'g1'
pretty(g1)
'g2'
pretty(g2)

% Coefficents of eigenvalue matrix
A1 = eig_A(1,1);
A2 = eig_A(2,2);

% w_dot substitution (from definition equation)
syms h_sym dh_sym
w_dot = A2 * h_sym + subs(g2,z_sym,h_sym) - dh_sym * (A1 * y_sym + subs(g1,z_sym,h_sym));

% Taylor's Series approximation of manifold
syms h2 h3
h = h2 * y_sym^2;% + h3 * y_sym^3;
dh = diff(h,y_sym);

% Attempting to solve for the h2 value...
w_dot = expand(subs(w_dot, [h_sym, dh_sym], [h, dh]));
'w_dot'
pretty(w_dot)

w_dot_soln = expand(w_dot);
w_dot_soln = subs(w_dot_soln, y_sym^4, 0);
w_dot_soln = subs(w_dot_soln, y_sym^3, 0);
syms y2
w_dot_soln = subs(w_dot_soln, y_sym^2, y2);
w_dot_soln = subs(w_dot_soln, y_sym, 0);
'w_dot_soln'
pretty(w_dot_soln)

syms h2y2
w_dot_soln = subs(w_dot_soln, h2*y2, h2y2)
solve(w_dot_soln == 0,h2y2)



end

```

```
if pblm4
%% Problem 4
syms x1 x2
x1_dot = -x1 + (2*x2)/(1 + x2^2);
x2_dot = -x2 + (2*x1)/(1 + x1^2);
f = [x1_dot; x2_dot];
'f'
pretty(f)
df = jacobian(f);
'jacobian'
pretty(df)




end


if pblm5
%% Problem 5
syms x1 x2 a b c
x1_dot = atan(a * x1) - x1 * x2;
x2_dot = b * x1^2 - c * x2;
x_dot = [x1_dot; x2_dot];
'x_dot'
pretty(x_dot)

x = [x1; x2];
'x'
pretty(x)

mu = [a; b; c];
mu_bar = [1; 0; 1];
'mu'
pretty(mu)


A_tau = jacobian(x_dot, x)
B_tau = jacobian(x_dot, mu)
```

```
sys_func = @pblm5_func;

T = [0,10];
x_0 = [1,-1, 0,0,0,0,0,0]';

[t,y] = ode45(@(t,y) sys_func(t,y,mu_bar,A_tau,B_tau),T,x_0);

y_states = y(:,[1,2]);
y_a = y(:,[3,6]);
y_b = y(:,[4,7]);
y_c = y(:,[5,8]);

fig = figure('position',[0,0,1500,1200]);
subplot(3,3,[1:3])
plot(t,y_states)
title('States')

titles = ["f1 - a", "f1 - b", "f1 - c", "f2 - a", "f2 - b", "f2 - c"];
for i = 3:8
    subplot(3,3,i+1)
    plot(t,y(:,i))
    title(titles(i-2))
end

saveas(fig,fullfile([pwd '\\' 'HW3' '\\' 'fig'],'pblm5.png'))

end




%% Local Functions
function dx = pblm5_func(t, x, parms, A, B)
    % pblm5 function
    arguments
        t (1,1) = 0;
        x (8,1) = zeros(8,1); %state and 6 sensitivities
        parms = false;
        A = 0;
        B = 0;
    end

```

```
    if parms == false
        a = 1;
        b = 0;
        c = -1;
    else
        a = parms(1);
        b = parms(2);
        c = parms(3);
    end

    % Variable Decode
    x1 = x(1);
    x2 = x(2);
    S = zeros(2,3);%[x(3), x(4), x(5); x(6), x(7), x(8)];

    % State Upadate Eqs
    x1_dot = atan(a * x1) - x1 * x2;
    x2_dot = b * x1^2 - c * x2;
    S_dot = subs(A * S + B);

    % Variable Encode
    dx = x;
    dx(1) = x1_dot;
    dx(2) = x2_dot;
    dx(3) = S_dot(1,1);
    dx(4) = S_dot(1,2);
    dx(5) = S_dot(1,3);
    dx(6) = S_dot(2,1);
    dx(7) = S_dot(2,2);
    dx(8) = S_dot(2,3);
end
```