

# MECH 6313 - Homework 4

Jonas Wagner

2021, March 26

## 1 Problem 1

### 1.1 Part a

**Problem:** Let the plant

$$\frac{1}{s^2}$$

be defined with no input and the following state-space representation:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= 0\end{aligned}\tag{1}$$

What can be said about the equilibrium stability of the system?

**Solution:** The defined system is a linear system with now input defined with

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

From this it is clear that there exists a single Jordan block with two poles at  $\lambda_{1,2} = 0$ .

The system is therefore unstable as the multiplicity of the roots on the  $j\omega$ -axis is greater than one which disqualifies the system from marginal stability.

### 1.2 Part b

**Problem:** Let the magnetically suspended ball system be defined as

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{-c\bar{u}^2}{mx_1^2} + g\end{aligned}\tag{2}$$

with the input

$$\bar{u} = \sqrt{\frac{mg}{c}}Y$$

defined as a constant.

What can be said about the stability of the system at its equilibrium?

**Solution:** The equilibrium points of the system can be found by solving for the points where the state equations are equal to zero. It is clear that all equilibrium points occur when  $\dot{x}_2 = 0$ .

$$\dot{x}_2 = 0 = \frac{-c\bar{u}^2}{mx_1^2} + g \quad (3)$$

$$\frac{c\bar{u}^2}{mx_1^2} = g \quad (4)$$

$$c\bar{u}^2 = mgx_1^2 \quad (5)$$

$$x_1^2 = \frac{c\bar{u}^2}{mg} \quad (6)$$

$$x_1 = \sqrt{\frac{c\bar{u}^2}{mg}} \quad (7)$$

By substituting the defined steady-state input, the following can be obtained:

$$x_1 = \sqrt{\frac{c\sqrt{\frac{mg}{c}}Y^2}{mg}} = \sqrt{Y} \quad (8)$$

Thus the equilibrium point is at

$$\begin{aligned} x_1 &= \sqrt{Y} \\ x_2 &= 0 \end{aligned} \quad (9)$$

The linearization of the system can then be obtained by evaluating the jacobian matrix at the equilibrium point:

$$A = \left[ \begin{array}{cc} 0 & 1 \\ \frac{2c\bar{u}^2}{mx_1^3} & 0 \end{array} \right] \bigg|_{x_1=\sqrt{Y}, x_2=0} \quad (10)$$

$$= \left[ \begin{array}{cc} 0 & 1 \\ \frac{2c\sqrt{\frac{mg}{c}}Y^2}{m\sqrt{Y}^3} & 0 \end{array} \right] \quad (11)$$

Which can be simplified into the linear dynamics defined by the A matrix:

$$A = \left[ \begin{array}{cc} 0 & 1 \\ \frac{2g}{\sqrt{Y}} & 0 \end{array} \right] \quad (12)$$

This system linearization around the equilibrium point can be analyzed to see that the characteristic polynomial is

$$s^2 - \frac{2g}{\sqrt{Y}}$$

The poles of the linearized system can then be found as

$$\lambda_{1,2} = \pm \sqrt{\frac{2g}{\sqrt{Y}}} = \pm \frac{\sqrt{2g}}{\sqrt[4]{Y}} = \pm \sqrt{2g} (Y)^{-\frac{1}{4}}$$

From this it can be seen that the system is unstable at the equilibrium point due to the positive pole at  $\lambda_1 = \sqrt{2g}(Y)^{\frac{1}{4}}$ .

## 2 Problem 2

An unforced morse oscillator is governed by the following equations:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\mu(e^{-x_1} - e^{-2x_1})\end{aligned}\tag{13}$$

### 2.1 Part a

**Problem:** Find the equilibrium points of the system.

**Solution:** The equilibrium points of the system can be found by solving for the points where the state equations are equal to zero. It is clear that all equilibrium points occur when  $\dot{x}_2 = 0$ .

$$\dot{x}_2 = 0 = -\mu(e^{-x_1} - e^{-2x_1})\tag{14}$$

$$-\mu e^{-x_1} = e - \mu e^{-2x_1}\tag{15}$$

$$-x_1 = -2x_1\tag{16}$$

$$x_1 = 0\tag{17}$$

Thus, the only equilibrium point occurs at the origin.

### 2.2 Part b

**Problem:** Assess the stability properties of the equilibrium point.

**Solution:** Linearization of the model around the origin produces the following dynamic matrix:

$$A = \begin{bmatrix} 0 & 1 \\ -\mu & 0 \end{bmatrix}\tag{18}$$

The associated characteristic polynomial is given as

$$s^2 + \mu$$

thus the eigenvalues of the system are

$$\lambda_{1,2} = \pm j\sqrt{\mu}$$

From this it can be said that the linearized system is marginally stable as the roots of the system are purely imaginary. However this does not explicitly provide that the nonlinear system is stable.

The alternative method of proving this marginal stability is to directly use a Lyapunov Function defined by

$$V(x) = \frac{1}{2}x_2^2 + \int_0^{x_1} \mu(e^{-\xi} + e^{-2\xi})d\xi\tag{19}$$

and show the negative semi-definiteness of  $\dot{V}(x)$ :

$$\dot{V}(x) = x_2\dot{x}_2 + \mu(e^{-x_1} + e^{-2x_1})\dot{x}_1\tag{20}$$

$$= x_2(-\mu(e^{-x_1} - e^{-2x_1})) + \mu(e^{-x_1} + e^{-2x_1})(x_2)\tag{21}$$

$$= 0 \leq 0\tag{22}$$

thus the system is shown to be stable in the sense of lyapunov (marginally stable).

### 3 Problem 3

A nonlinear system is given as

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -g(k_1 x_1 + k_2 x_2), \quad k_1, k_2 > 0\end{aligned}\tag{23}$$

where  $g(\cdot)$  is known to satisfy the following

$$\begin{aligned}yg(y) &> 0, \quad \forall y \neq 0 \\ \lim_{|y| \rightarrow \infty} \int_0^y g(\xi) \, d\xi &= +\infty\end{aligned}\tag{24}$$

#### 3.1 Part a

**Problem:** Use an appropriate lyapunov function to show that the equilibrium point at  $x = 0$  is globally asymptotically stable.

**Solution:**

#### 3.2 Part b

**Problem:** Show that the saturation function

$$\text{sat}(y) = \text{sign}(y) \min\{1, |y|\}$$

satisfies the above assumptions for  $g(\cdot)$ .

What is the exact form of your Lyapunov function for this saturation nonlinearity?

**Solution:**

### 3.3 Part c

**Problem:** Demonstrate that a double integrator with a saturation actuator, given as

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \text{sat}(u)\end{aligned}\tag{25}$$

can be saturated with the state-feedback controller

$$u = -k_1x_1 - k_2x_2$$

and design  $k_1$  and  $k_2$  such that the eigenvalues of the linearization are placed at  $\lambda_{1,2} = -1 \pm j$ .

Simulate the closed loop with and without saturation and compare the resulting trajectories vs time.

**Solution:** MATLAB and Simulink were used to calculate gains as well as simulate the stabilization of the saturated double integrator system.

The negative feedback controller gains to achieve the desired poles was calculated to be:

$$K = \begin{bmatrix} 2 & 2 \end{bmatrix}\tag{26}$$

The simulink model that was used to simulate consisted of an LTI system representing the double integrator, a saturation block, and a gain for feedback. The model can be seen in Figure 3.3. The results for

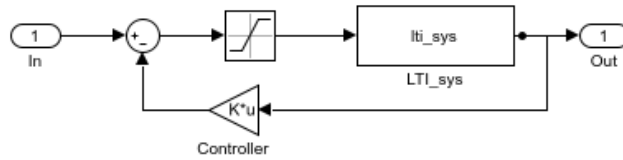


Figure 1: Simulink model used to simulate the saturated double integrator.

stabilization of the system with initial conditions  $x_0 = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$  is shown in Figure 3.3.

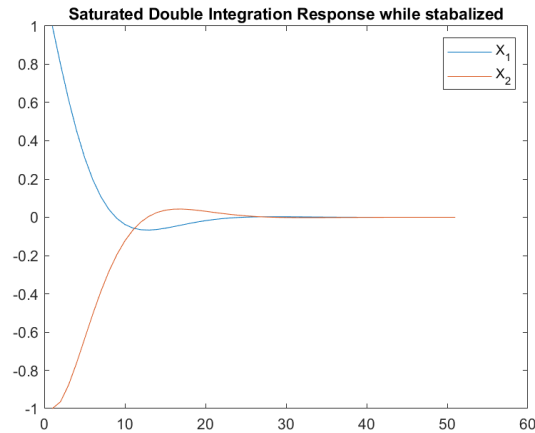


Figure 2: Simulation results of the stabilization of the saturated double integrator.

## 4 Problem 4: K4.14

A nonlinear system is given as

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -g(x_1)(x_1 + x_2)\end{aligned}\tag{27}$$

where  $g(\cdot)$  is known to be locally Lipschitz continuous and satisfies  $g(y) \geq 1$ ,  $\forall y \in \mathbb{R}$ .

**Problem:** Verify that

$$V(x) = \int_0^{x_1} yg(y) \, dy + x_1x_2 + x_2^2\tag{28}$$

is positive definite  $\forall x \in \mathbb{R}^2$  and radially unbounded. Next use  $V(x)$  to shown that  $x = 0$  is globally asymptotically stable.

**Solution:**

## 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\_HW4

```
1 % MECH 6313 - Homework 4
2 % Jonas Wagner
3 % 2021-03-26
4
5 clear
6 close all
7
8 %% Problem 3
9 % Part c
10 % Saturation Block
11 a = -1;
12 b = 1;
13
14 % Linearized System
15 A = [0, 1;
16      0, 0];
17 B = [0;
18      1];
19 C = eye(2);
20 D = [0;
21      0];
22 lti_sys = ss(A,B,C,D);
23 x0 = [1;
24       -1];
25
26
27 % Controller Gain
28 K = place(A,B, [-1+j,-1-j])
29 A_BK = A - B * K;
30 eig_A_BK = eig(A_BK);
31
32
33 %% Simulink Creation In Code
34 % Simulink Settings -----
35 % Specifiy Subfolder
36 subfolder = 'Homework\HW4';
37 % Get the current configuration
38 cfg = Simulink.fileGenControl('getConfig');
```

```

39 % Changes Code Save Location
40 cfg.CacheFolder = [pwd, '\', subfolder];
41 cfg.CodeGenFolder = [pwd, '\', subfolder];
42 cfg.CodeGenFolderStructure = 'TargetEnvironmentSubfolder';
43 % Apply new Config
44 Simulink.fileGenControl('setConfig', 'config', cfg, 'keepPreviousPath',true, 'createDir',
    true);
45
46 %% Specify the name of the model to create
47 fname = 'pblm3_model';
48
49 % Check if the file already exists and delete it if it does
50 if exist(fname,'file') == 4
51     % If it does then check whether it's open
52     if bdIsLoaded(fname)
53         % If it is then close it (without saving!)
54         close_system(fname,0)
55     end
56     % delete the file
57     delete([fname, '.slx']);
58 end
59
60 % Create Simulink Model
61 new_system; %fname not used... saved later
62
63 % Create Simiple Input
64 add_block('simulink/Sources/In1', [gcs, '/In']);
65
66 % Create Sum block
67 add_block('simulink/Commonly Used Blocks/Sum', [gcs, '/Sum'],...
68     'inputs', '|+-');
69 add_line(gcs, 'In/1', 'Sum/1','autorouting','on');
70
71 % Saturation Block
72 add_block('simulink/Commonly Used Blocks/Saturation', [gcs, '/Saturation'], ...
73     'LowerLimit','a', ...
74     'UpperLimit','b');
75 add_line(gcs, 'Sum/1', 'Saturation/1');
76
77 % State-Space System
78 add_block('cstblocks/LTI System', [gcs, '/LTI_sys'],...
79     'sys','lti_sys',...
80     'IC', 'x0');

```



```

81 add_line(gcs, 'Saturation/1', 'LTI_sys/1');
82
83 % Controller (just a feedback gain)
84 add_block('simulink/Commonly Used Blocks/Gain', [gcs, '/Controller'],...
85     'Gain', 'K',...
86     'Multiplication', 'Matrix(K*u)',...
87     'Orientation', 'left');
88 add_line(gcs, 'LTI_sys/1', 'Controller/1');
89 add_line(gcs, 'Controller/1', 'Sum/2');
90
91 % Create Simple Scope/Output
92 add_block('simulink/Sinks/Out1', [gcs, '/Out']);
93 add_line(gcs, 'LTI_sys/1', 'Out/1');
94
95
96 % Auto Arrange
97 Simulink.BlockDiagram.arrangeSystem(gcs) %Auto Arrange
98
99 %% Save and Open System
100 save_system(gcs,[subfolder, '/', fname]);
101 print(['-s', gcs], '-dpng',... % Print model to figure
102     [pwd, '\', subfolder, '\fig\', 'pblm3_c_model.png'])
103 % open(fname); % Don't need to open to run
104
105
106
107 %% Simulate System
108 simConfig.SaveState = 'on';
109 simOut = sim(fname, simConfig);
110
111 % Sim Data
112 Xout = simOut.xout{1}.Values.Data; %Only works by grabbing states of first block (LTI_sys
    )
113
114
115 %% Plot Results
116 fig = figure;
117 plot(Xout(:,1))
118 hold on
119 plot(Xout(:,2))
120 legend('X_1', 'X_2')
121 title('Saturated Double Integration Response while stabalized')
122 saveas(fig, [pwd, '\', subfolder, '\fig\', 'pblm3_c_plot.png'])

```

123

124 `close all`