Fachhochschule Südwestfalen

University of Applied Sciences

Hochschule für Technik und Wirtschaft

Abt. Soest

MSC SYSTEMS ENGINEERING AND ENGINEERING MANAGEMENT

SUMMER SEMESTER 2021

ADVANCED CONTROL TECHNOLOGY

ASSIGNMENT 2:BIOREACTOR (GROUP 17)

DATE OF SUBMISSION: 09.07.2021

SUBMITTED TO: DR.-ING. ANDREAS SCHWUNG

SUPERVISOR: MR. STEVE YUWONO

SUBMITTED BY: YEHIA IBRAHIM (30236730)

CONTENTS PAGE NO.

Problem Statement

OVERVIEW OF THE SYSTEM:

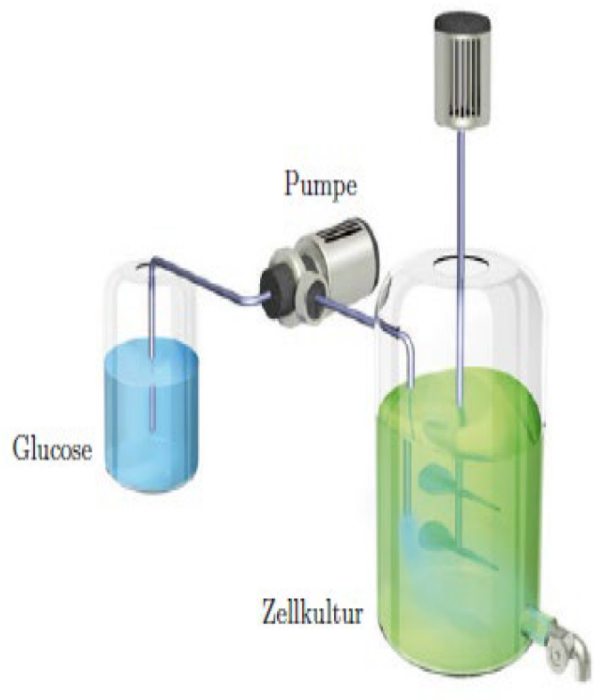

Figure1: Bioreactor System

A bioreactor is a vessel that is used to cultivate cells under sterile conditions and where environmental conditions like ph, the temperature can be controlled. They are commonly cylindrical and the size ranges from liters to cubic meters. The material widely used is stainless steel. The pump is used to transfer the glucose from the glucose container to the cell culture. The mixture in the cell culture is stirred by a stirrer to generate a chemical reaction.

The system was modeled, using the given state-space system. The constants maximal growth rate, affinity constants, feed concentration of glucose, and the yield constant are given below.

$$\dot{\boldsymbol{x}} = \boldsymbol{a}(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{x}) \cdot u = \begin{bmatrix} \mu(x_2) \cdot x_1 \\ -\dfrac{1}{\alpha}\mu(x_2) \cdot x_1 \end{bmatrix} + \begin{bmatrix} -x_1 \\ K - x_2 \end{bmatrix} u,$$

$$y = g(\boldsymbol{x}) = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}\,.$$

$$\mu(x_2) = \frac{\mu_0 x_2}{k_1 + x_2 + k_2 x_2^2}$$

Where:

- Maximal growth rate, $\mu_0 = 2$
- Affinity constant, $k_1 = 0.06$
- Affinity constant, $k_2 = 0.3$
- Feed concentration of glucose, K = 2
- Yield constant, $\alpha = 0.7$

The following tasks were performed for the given assignment.

1. A model of the system in Matlab/Simulink was built.
2. The system behaviors stability, controllability, observability were analyzed.
3. Designing the Lyapunov function for stability based on the Direct Lyapunov method.
4. Designing a controller based on Feedback Linearization.
5. Analyzing the controller design and recording the obtained results.

SYSTEM MODELLING

A model describes the reality of a specific system. They could be described pictorially or using graphs. A mathematical model describes a system through mathematical equations. Once the mathematical description of the process involved in a bioreactor is ready, it will help to understand the system better and the effect due to changing the different parameters in a system.

The following steps are involved in the mathematical model development. [3]

1. Elimination of the experimental errors, smoothing of data points, and calculation of rates and specific rates
2. Proposal of the mathematical model
3. Parameter optimization
4. Parameter Sensitivity Analysis
5. Statistical Validity

The system modeling on Simulink is shown in figure 2.

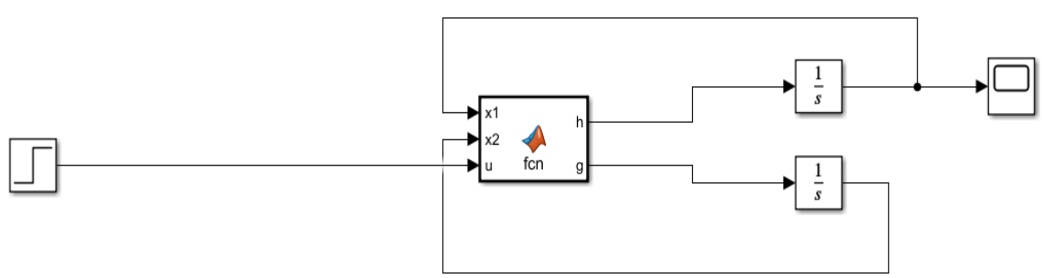

Figure2: System Modeling on Simulink [*]

SYSTEM LINEARIZATION

The section shows how to perform linearization of the system described by the nonlinear differential equations.

A linear state-space system is as follows

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$
$$y = \mathbf{C}\mathbf{x} + Du$$

Figure3: Linear System State-space Representation [1].

- **A** is n*n; **A** is the state matrix, a constant
- **B** is n*r; **B** is the input matrix, a constant
- **u** is r*1; **u** is the input, a function of time
- **C** is m*n; **C** is the output matrix, a constant
- **D** is m*r; **D** is the direct transition (or feedthrough) matrix, a constant
- **y** is m*1; **y** is the output, a function of time

The Jacobian matrix was used to get the partial derivative of each function with respect to the x1,x2, and u where the general form of the Jacobian matrix is given in Figure 3.

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{f}}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Figure 3: Jacobian Matrix [2].

Linearizing the given matrix leads to the following 2 equations

$$g=((\mu0*x2)/(k1+x2+k2*x2^{\wedge}2))*x1-(x1*u)$$
$$h=((-1/\alpha)*(\mu0*x2)/(k1+x2+k2*x2^{\wedge}2))*x1+(k-x2)*u$$

The operating points X1 and X2 were obtained by setting the input variable to 1 and then applying the Taylor series expansion to get x1 and x2.
Figure 4 shows the code used to obtain the values of X1 and X2 and figure 5 shows the results of X1 and X2.
The value of X1 does not mean that the biomass concentration of the cell culture is negative which is unrealistic but the coordinate system of this linearized system is not located on the origin but the operation point. So the value of X1 means that it is deviating negatively from the level of the operation point.
The operation point obtained is (-0.8905,3.2722) can be seen in fig

Figure 4: Obtaining the Values of X1 and X2 [*]

```
x1_value =

-0.8905


x2_value =

3.2722
```

Figure 5: Values of X1 and X2[*]

The code used for obtaining the state-space system and then substituting the values of x1, x2, and the given variables is shown in figure 6.

```
%clearing the workspace, command window and previous plots
clear all
close all
clc

%Assigning symbolic variables
syms x1 x2; % defining the state variables
u = 1; % the input variable chosen arbitrarily
u0 = 2; k1=0.06; k2=0.3; k =2; a=0.7; %given Variables
g= ((u0*x2*x1)/(k1+x2+k2*(x2^2)))-x1*u %x1 dot
h= (-(u0*x2*x1)/(a*(k1+x2+k2*(x2^2))))+(k-x2)*u; %x2 dot
[solve_x1,solve_x2]=solve([g==0,h==0],[x1,x2],"real",true);


x1_roots=solve_x1;
x2_roots=solve_x2;
x1_value=round(x1_roots(end),4) % rounding the values of x1 and x2 to the nearest 4th decimal number
x2_value=round(x2_roots(end),4)
disp(x1_value,x2_value);
```

Figure 6: Obtaining the State-Space System

The state-space model obtained is shown in figure 7.

```
A =

         0    0.1311
   -1.4286   -1.1873


B =

    0.8905
   -1.2720


C =

     1     0


D =

     0
```

Figure 7: The Obtained State-Space Model

A=

$$\begin{bmatrix} 0 & 0.1311 \\ -1.4286 & -1.1873 \end{bmatrix}$$

B=

$$\begin{bmatrix} 0.8905 \\ -1.2720 \end{bmatrix}$$

C= $[\,1 \quad 0\,]$

D= $[\,0\,]$

The step response of the system is shown in figure 8

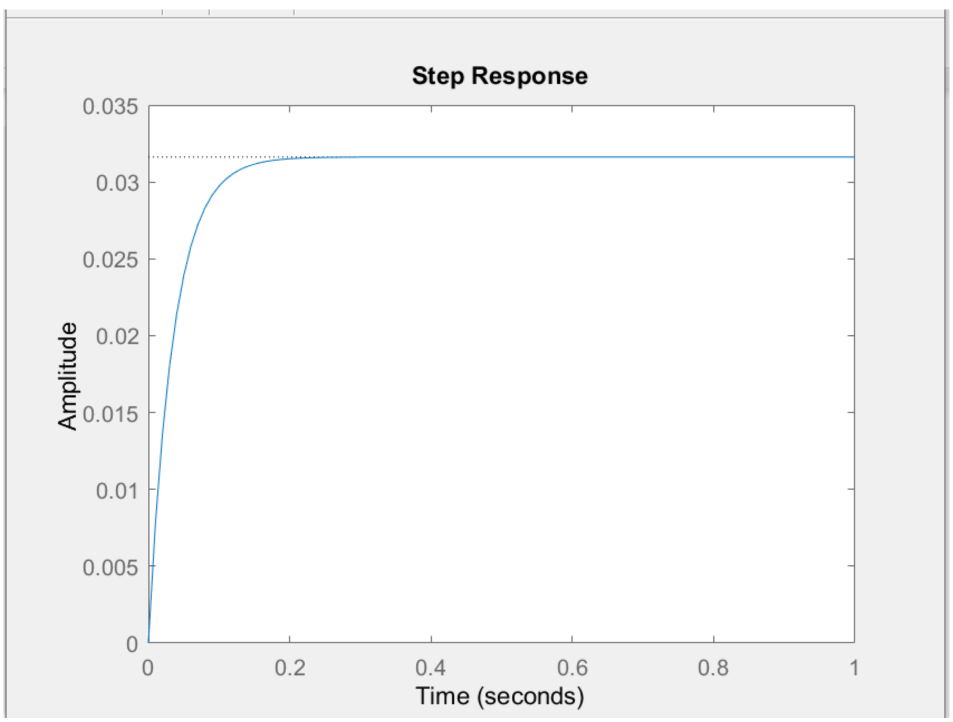

Figure8: Step response of the given system[*]

ANALYSIS OF SYSTEM BEHAVIOUR

This section illustrates how stability, controllability, and observability were analyzed in Matlab. The stability of the system was checked by analyzing the poles.

The stability of the system can be determined from the eigenvalues by the following table

Eigenvalue Type	Stability	Oscillatory Behavior	Notation
All Real and +	Unstable	None	Unstable Node
All Real and -	Stable	None	Stable Node
Mixed + & - Real	Unstable	None	Unstable saddle point
+a + b*i*	Unstable	Undamped	Unstable spiral
-a + b*i*	Stable	Damped	Stable spriral
0 + b*i*	Unstable	Undamped	Circle
Repeated values	Depends on orthogonality of eigenvectors		

Table 1: The stability corresponding to each type of eigenvalue

The Code used in determining the stability of the system in Matlab is shown in figure 9.

```
%Checking stability
rlocus(sys); % to check the poles of the system on the graph
pole(sys) % to see the exact numerical values of the poles
Stability=eig(A);
if Stability<0
    disp('System is stable')
else
    disp('System is unstable')
end
```

Figure9: Stability Checking in Matlab [*]

The controllability of a system can be checked by evaluating the rank of [B: AB] and checking if it is equal to the order of matrix A to check if there are any uncontrollable states in the system.
The Bioreactor controllability checking in Matlab is shown in figure 9

```
%Checking controllability
Controllability=ctrb(A,B);
if (length(A) - rank(Controllability)==0) % checking whether the rank of the controllabelity is equal to the length of A

      disp('System is controllable')
else
    disp('System is uncontrollable')
end
unco = length(A) - rank(Controllability);
display(unco); % double checking to see if there is any uncontrollable state
```

Figure10: Matlab Code for Controllability [*]

The observability criteria of the system are based on getting the rank of the matrix [C^T: A^T C^T] and equating the result to the order of matrix A and if they are the same that means that system is observable which is the case in the designed bioreactor system.

Figure 10 shows the observability code that was used in Matlab.

```
%Checking observability
Observability=obsv(A,C);
if (length(A) - rank(Observability)==0)% checking whether the rank of the observability is equal to the length of A

    disp('System is observable')
else
    disp('System is unobservable')
end
```

Figure11: Matlab Code for Observability[*]

The analysis of the system resulted in the following characteristics of the system which are:

- The system is stable
- The system is controllable
- The system is observable

Figure 11 shows these characteristics on Matlab.

```
System is stable
System is observable
System is controllable
```

Figure12: System Characteristics on Matlab[*]

Designing The Lyapunov Function

(Lyapunov Stability Theorem) Let x = 0 be an equilibrium point of ± = f (x), f : D -+ R", and let V : D -> R be a continuously differentiable function such that

(i) V(0) = 0,

(ii) V(x) > 0 for X≠0,

(iii) V'(x) < 0 for X≠0,

Thus x = 0 is stable.

Before designing the Lyapunov function the system must be stable and in our case the system is stable.

Using the Quadratic function $V(X)= X_1^2 + X_2^2$ to determine the stability of the nonlinear system at the origin. It is clear that the quadratic equation is positive definite as it satisfies the first two conditions of the Lyapunov Theorem.

$V'(x)= x'^T$ grad $V(x) = [x'_1 \quad x'_2] \begin{bmatrix} 2x1 \\ 2x2 \end{bmatrix}$

$V'(x) = (2\mu(x_2).x_1 \hat{} 2) / (K_1 + x_2 + k2x_2 \hat{} 2) - 2x_1u + (-2\mu(x_2 \hat{} 2).x_1) / (K1+x_2+k2x_2) + 2Ku - 2x_2 \hat{} 2u)$

By Applying this equation in Matlab, there are several spaces where $V'(x) < 0$ but three of these spaces are mentioned below

$V'(x) < 0$ for $x_1 \geq 0.8$ and $x_2 \geq 0.9$ or $x_1 \leq -0.7$ and $x_2 \leq -0.8$ or $x_1 \leq 0$ and $x_2 \leq -1.5$. (note : all of these numbers are approximated to the first decimal).

This means that the system is asymptotically stable at the equilibrium state.

V'(x) = -8.23423 at the operation point (-0.8905, 3.2722) which means that the system is stable at the operation point in the sense of Lyapunov Function.

Matlab was used to plot $V'(x)$ against x_1 and x_2 but in the code, they are represented as ($V'(x) \rightarrow z$, $x_1 \rightarrow x$, and $x_2 \rightarrow y$).

The code written for obtaining the results of V'(x) on Matlab can be seen in figure 13.

```
a= [-3:.1:3];
b= [-3:.1:3];
[x,y]=meshgrid(a,b);
z= (4*y.*(x.^2))./(0.06 + y + 0.3*(y.^2)) - 2*(x.^2) + (-5.714*x.*(y.^2))./(0.06 + y + 0.3*(y.^2))+4-(2*(y.^2))
surf(x,y,z);
xlabel('x axis');
ylabel('y axis');
zlabel('z axis');
```

Figure 13: code written for plotting V'(x)[*]

The 3d plotting of V'(x) against x_1 and x_2 and the values of x_1 and x_2 at which the system is stable can be seen in figure 14

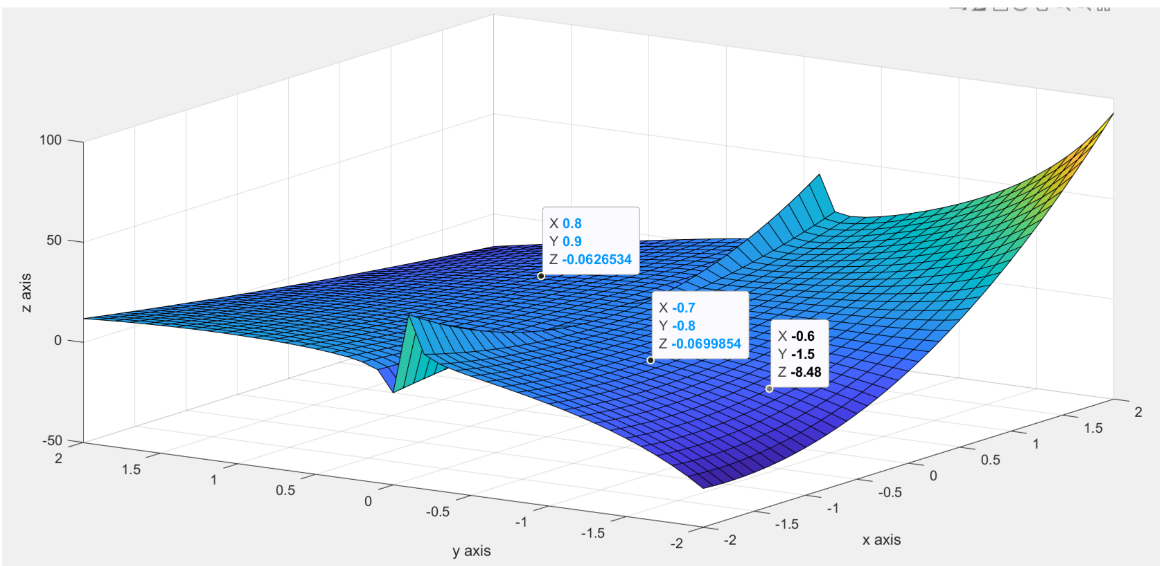

Figure 14The value of V'(x) at the operation point (-0.8905, 3.2722) can be seen in figure 15.

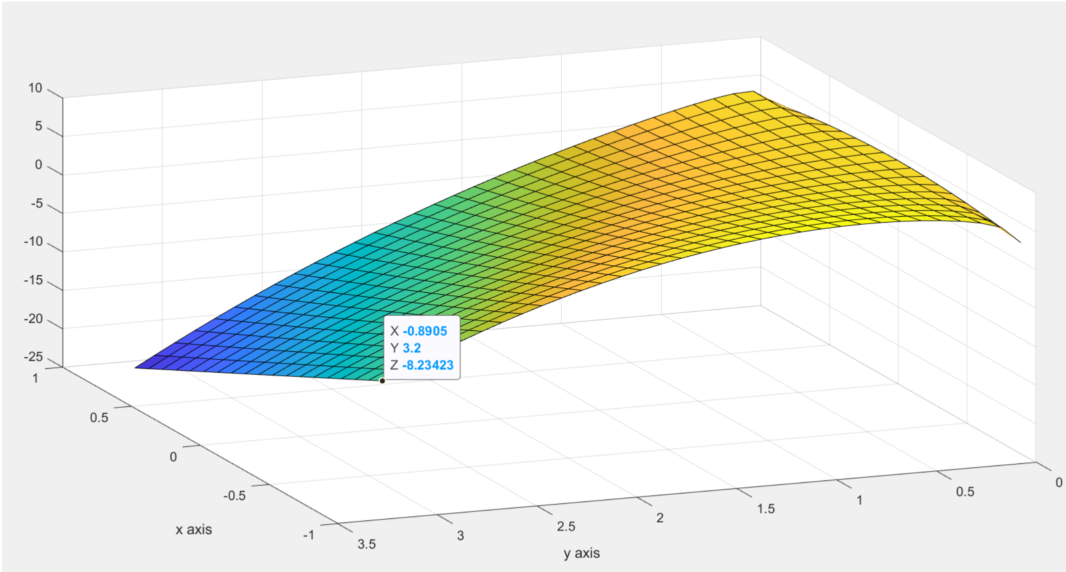

Figure 15: the value of V'(x) at the operation point [*]

Feedback Linearization

Basic steps for Feedback Linearization

1- Take derivatives of y

$Y' = \partial c/\partial t = \partial c(x)/\partial x * x' = \partial c(x)/\partial x * [a(x)+b(x).u]$

2- Check for $L_b c(x) = 0 \rightarrow$ if 0 then continue with y'' and if $\neq 0$ then stop (relative degree of the system)

In our system

$X' = a(x) + b(x).u$

$Y = g(x)$

$y' = \partial c(x)/\partial x * [a(x)+b(x).u]$

$L_b g(x) = \bigtriangledown g.b = [\ 1\ \ 0\ ] \begin{bmatrix} -x1 \\ k - x2 \end{bmatrix}$

$L_b g(x) = -x_1 \neq 0$

The relative degree is 1 and since the order of the system is 2 then the system has internal dynamics of order n-r which is 1

$Y = \Psi_1 = x_1$

$Y' = \Psi'_1 = x_1' = \mu(x_2).x_1 - x_1.u$

$\Psi'_1 = v$

$\partial \varphi/\partial x * b(x) = 0$ guarantees that the internal dynamics does not depend on the input u

$\partial \varphi/\partial x * b(x) = [\partial \varphi/\partial x_1\ \partial \varphi/\partial x_2] \begin{bmatrix} -x1 \\ k - x2 \end{bmatrix}$

Then $-x_1 \partial \varphi/\partial x_1 + [k - x_2]\, \partial \varphi/\partial x_2 = 0 \rightarrow 1$

Assume $\varphi = f_1(x_1) + f_1(x_2)$

Consequently $\partial\varphi/\partial x_1$= df_1/dx_1 & $\partial\varphi/\partial x_2$= df_2/dx_2

By substituting in eqn.1 x_1 * $df1/dx_1$ = [k-x_2] $df2/dx_2$

Assuming $df_2/dx_2 = x_1$ and df_1/dx_1= [k-x_2]

Then $f_1 = (k-x_2).x_1 + C1$ & $f2 = x_1.x_2 + C2$

$\varphi = f_1(x_1) + f_2(x_2) =(k-x_2).x_1 + x_1.x_2 + C1+ C2$

Assuming C1 & C2 = 0

Then

$$T(x) = \begin{bmatrix} \varphi \\ \psi 1 \end{bmatrix} = \begin{bmatrix} (k-x2)x1 + x1x2 \\ x1 \end{bmatrix} = \begin{bmatrix} Z1 \\ Z2 \end{bmatrix}$$

Since the $\partial t/\partial x$ matrix is singular, the second term of the T(x) matrix was chosen arbitrarily as x_2 so that the matrix is not singular and it is a global diffeomorphism.

Then the new T(x) is = $\begin{bmatrix} Kx1 \\ x2 \end{bmatrix}$

Applying the Lyapunov:

Using the Quadratic function $V(X)= X_1^2 + X_2^2$ to determine the stability of the system.

$V'(x) = x'$^T grad $V(x) = [x'_1 \quad x'_2] \begin{bmatrix} 2x1 \\ 2x2 \end{bmatrix}$

$V'(x) = [K \quad 1] \begin{bmatrix} 2x1 \\ 2x2 \end{bmatrix} = 2Kx_1 + 2x_2$

$V'(x) \le 0$ if $x_1 \le 0$ & $x_2 \le 0$

This means that the system is globally stable at the equilibrium state

Deriving the controller according to the formula in figure …

$$r(\boldsymbol{x}) = \frac{L_{\boldsymbol{a}}^{\delta} c(\boldsymbol{x}) + a_{\delta-1} L_{\boldsymbol{a}}^{\delta-1} c(\boldsymbol{x}) + \ldots + a_0 c(\boldsymbol{x})}{L_{\boldsymbol{b}} L_{\boldsymbol{a}}^{\delta-1} c(\boldsymbol{x})}.$$

In our case $r(x) = [L_a g(x) + a_0 g(x)] / L_b L^0_a g(x)$

$L_a g(x) = \bigtriangledown g.a = [1\ 0] \begin{bmatrix} \mu(x2)x1 \\ -\mu(x2)x1/\alpha \end{bmatrix} = \boldsymbol{\mu}(\boldsymbol{x}_2)\boldsymbol{x}_1$

$L_b L^0_a g(x) = L_b g(x) = L_b g(x) = \bigtriangledown g.b = [1\ 0] \begin{bmatrix} -x1 \\ k - x2 \end{bmatrix} = -\boldsymbol{x}_1$

$r(x) = [\boldsymbol{\mu}(\boldsymbol{x}_2)\boldsymbol{x}_1 + a_0] / -\boldsymbol{x}_1$

Deriving the Prefilter according to the formula in figure ………

$$v(\boldsymbol{x}) = \frac{V}{L_{\boldsymbol{b}} L_{\boldsymbol{a}}^{\delta-1} c(\boldsymbol{x})}.$$

$v(\boldsymbol{x}) = V / -\boldsymbol{x}_1$

Analyzing The Controller Design

The calculated prefilter, controller, and nonlinear system were modeled on simulating and the modeled system can be seen in figure 16.

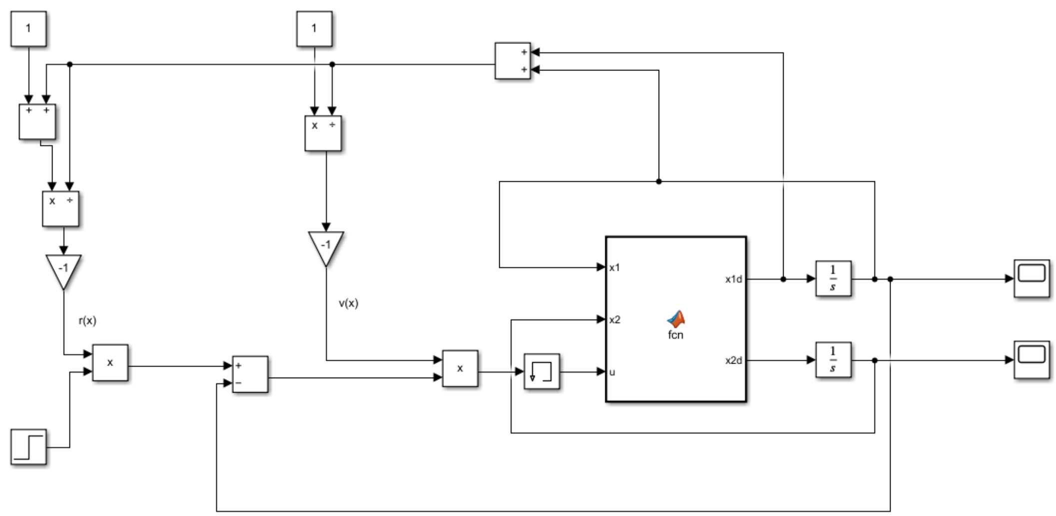

Figure 16: Prefilter, Controller, and the nonlinear system [*].

The values of a_0 and V were chosen arbitrarily as 1 and 1 and the results from the scope can be seen in figure 17.

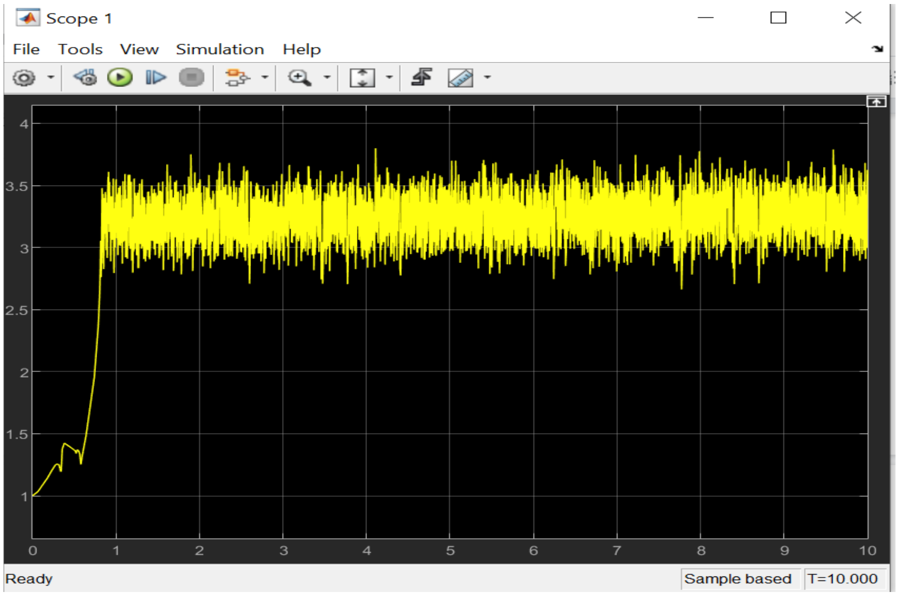

Figure 17: the output of the system over 10 seconds [*].

It is seen from the figure that the system turns from instability to critically stable after 0.9 seconds approximately and when increasing the value of a_0 to 5 and decreasing the value of V to 0.1 the system takes more time to turn into the critical stable state and vice versa and this can be seen in figure 18 and 19 respectively.

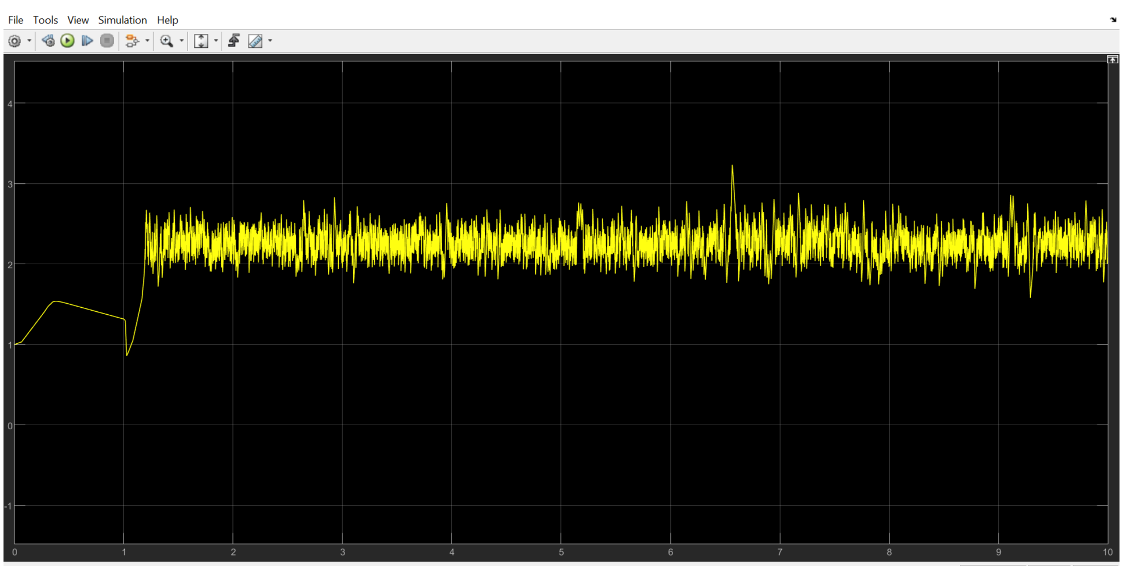

Figure 18: the output after decreasing V and increasing a_0

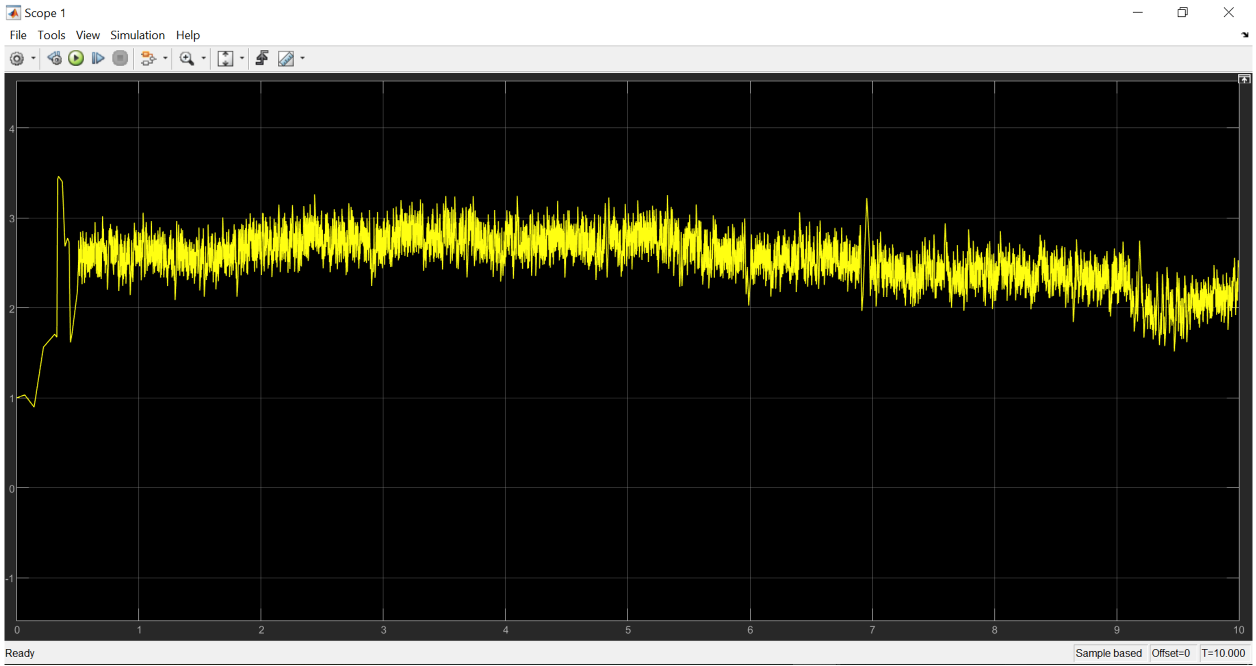

Figure 19: the output after increasing V and decreasing a_0

It can also be noticed that after increasing V and decreasing a_0 the output values tend to decay over time.

By increasing both of V and a_0 it can be seen that the values of the output have a larger range. And by decreasing the values of V and a_0 it can be seen that the time taken for the system to reach the instability state is much more than before and this can be seen in figures 20 and 21 respectively.

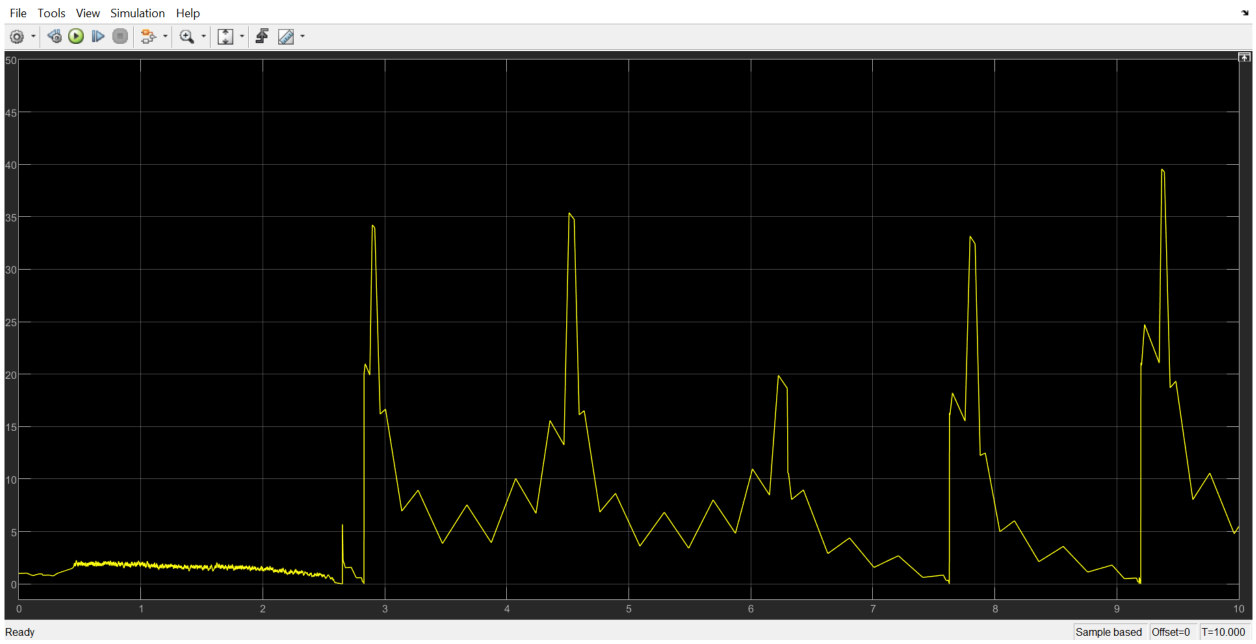

Figure 20: the output value after increasing both of V and a_0 [*]

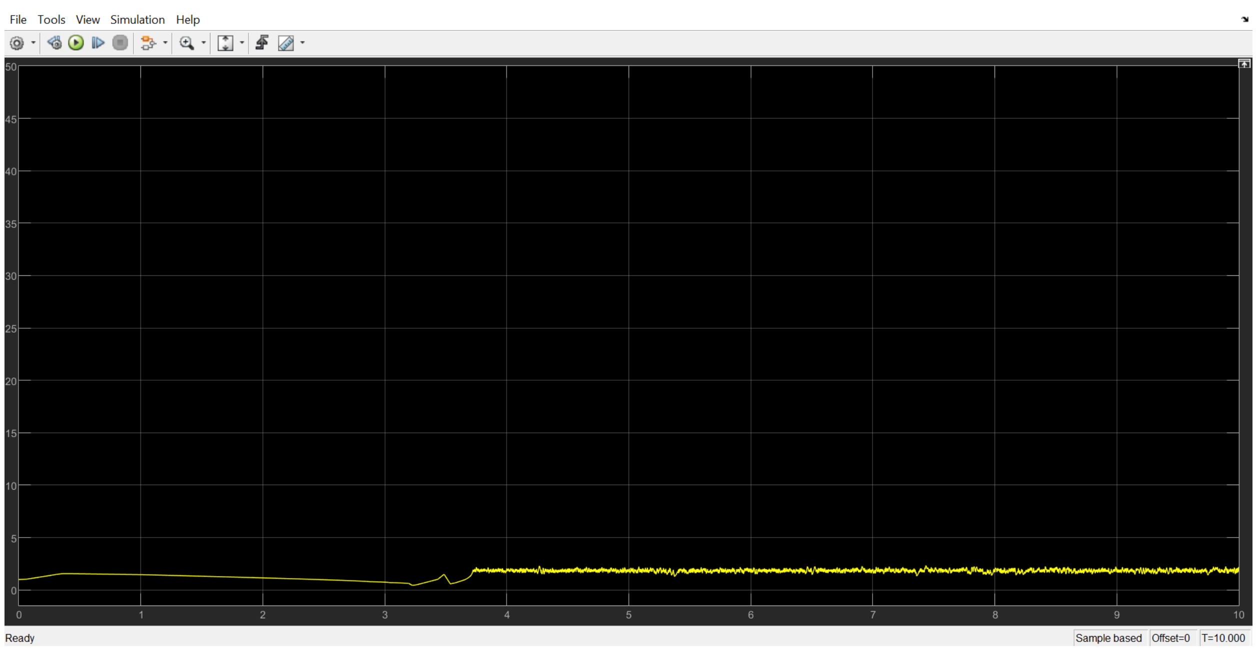

Figure 20: the output value after increasing both of V and a_0 [*]

REFERENCES

[1]K. Ogata, Modern control engineering, 5th ed. Pearson, 2010
[2]https://www.scienceabc.com/pure-sciences/what-is-a-jacobian-matrix.html.
[3]Development of mathematical model (Steps in Modelling): Bioreactor Modeling & Simulation lab: Biotechnology and Biomedical EngineeringResponse: Amrita Vishwa Vidyapeetham Virtual Lab
[4]https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Book%3A_Chemical_Process_Dynamics_and_Controls_(Woolf)/10%3A_Dynamical_Systems_Analysis/10.04%3A_Using_eigenvalues_and_eigenvectors_to_find_stability_and_solve_ODEs?fbclid=IwAR2FsW9d-_sGYGBxc7EfAOOe00pzgTClHFXm9cyxrRLXY3eBys4BjA5q9C4
[5] Hassan K. Khalil, "Nonlinear Systems"
[*] Own Figures and Tables.