Fachhochschule Südwestfalen
University of Applied Sciences

Advanced Control Technology

Bioreactor (Assignment I)

Submitted to:

Prof. Andreas Schwung

Submitted by:

Murtaza Khuzema Basuwala, Viktor Chilikin

SEEM MSc program

SS 2018

chilikin.viktor@fh-swf.de [10061806]

basuwala.murtaza@fh-swf.de [10062949]

Contents

1. Introduction

Nowadays, bioreactors are the popular kinds of equipment in the food, agricultural and medical industries. Their functioning is based upon biotechnologies, utilizing living organisms to produce goods (cheese, medicaments, fertilizers etc.). Even though in the middle of 20th century, computer technologies have been developed enough to perform systems control, the biotechnical industry was lagging behind in the introducing control strategies. The reason for that were some problems, i.e. the proper measurement of key physical and biochemical parameters [1]. The main objectives of this assignment were to understand a way the system works and then develop a solution for regulating the concentration level of substances. Respectively, the following tasks were performed:

1. Linearization of the system model in a suitable operation point;
2. Building a model of the system in MatLAB/Simulink;
3. Analysis of the system behavior: stability, controllability, observability;
4. Design of a state space controller. In addition, design of a state observer;
5. Analysis of the controller and observer design and the presentation of obtained results.

2. Initial Conditions

2.1 Description of the System

The general view of the system is presented in figure 1.

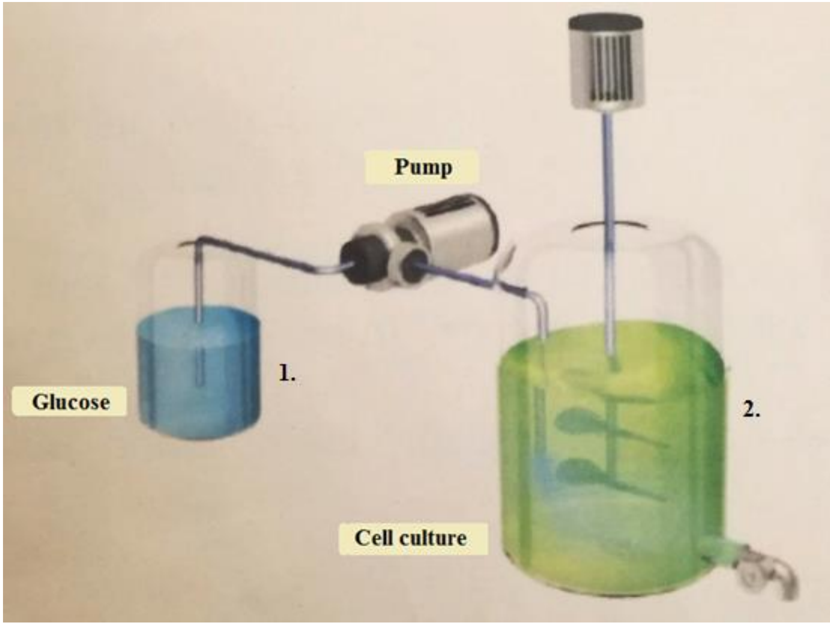


Figure 1: Bioreactor system [*]

The pump is used to transfer glucose from beaker 1 into beaker 2, which contains cell culture, or substrate. In addition, the stirrer is used to distribute glucose in the substrate more evenly. With the admixture of glucose, a biological process occurs, which leads to the generation of biomass.

2.2 State-Space Mathematical Model of the System

The model can be described as follows:

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

$$y = g(\boldsymbol{x}) = [1 \quad 0]\boldsymbol{x}$$

where: $\mu_0 = 1$ – maximal growth rate;

$k_1 = 0.06$ and $k_2 = 0.7$ – affinity constants;

$K = 5$ – feed concentration of glucose;

$\alpha = 0.7$ – yield constant;

x_1 and x_2 – the concentration of biomass and substrate respectively;

u – the input;

y – the output;

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

– the growth rate.

3. Linearization of the System Model in a Suitable Operation Point

It is possible to linearize a nonlinear system with a linear one, due to the fact that usually, the system functions around an equilibrium point, and in such case, the linear system is similar to the nonlinear one in a specific signal spectrum.

The linear system representation looks as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \qquad (3)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

where: **A** – the state matrix;

B – the input matrix;

C – the output matrix;

D – the direct transmission matrix.

With the usage of Taylor's series the nonlinear model (1) can be linearized around an operating point of a steady state. The template for the linearization of the state-space model is presented in formula 4:

$$\begin{bmatrix} \frac{d(x_1-x_{1s})}{dt} \\ \frac{d(x_2-x_{2s})}{dt} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}\Big|_{x_{1s},\,x_{2s},u_s} & \frac{\partial f_1}{\partial x_2}\Big|_{x_{1s},\,x_{2s},u_s} \\ \frac{\partial f_2}{\partial x_1}\Big|_{x_{1s},\,x_{2s},u_s} & \frac{\partial f_2}{\partial x_2}\Big|_{x_{1s},\,x_{2s},u_s} \end{bmatrix} \begin{bmatrix} x_1 - x_{1s} \\ x_2 - x_{2s} \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial u}\Big|_{x_{1s},\,x_{2s},u_s} \\ \frac{\partial f_2}{\partial u}\Big|_{x_{1s},\,x_{2s},u_s} \end{bmatrix} [u - u_s], \tag{4}$$

$$\boldsymbol{y} - \boldsymbol{y_s} = \begin{bmatrix} \frac{\partial g}{\partial x_1}\Big|_{x_{1s},\,x_{2s},u_s} & \frac{\partial g}{\partial x_2}\Big|_{x_{1s},\,x_{2s},u_s} \end{bmatrix} \begin{bmatrix} x_1 - x_{1s} \\ x_2 - x_{2s} \end{bmatrix} + \begin{bmatrix} \frac{\partial g}{\partial u}\Big|_{x_{1s},\,x_{2s},u_s} \end{bmatrix} [u - u_s]$$

where: x_{1s}, x_{2s}, u_s – the operating points;

linearizing the given matrix leads to the following equations

$$\frac{dx_1}{dt} = \dot{x}_1 = \frac{\mu_0 x_2 x_1}{k_1 + x_2 + k_2 x_2^2} - x_1 u = f_1(x_{1s}, x_{2s}, u_s), \tag{5}$$

$$\frac{dx_2}{dt} = \dot{x}_2 = \frac{-\mu_0 x_2 x_1}{\alpha(k_1 + x_2 + k_2 x_2^2)} + ku - x_2 u = f_2(x_{1s}, x_{2s}, u_s), \tag{6}$$

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}\Big|_{x_{1s},\,x_{2s},u_s} & \frac{\partial f_1}{\partial x_2}\Big|_{x_{1s},\,x_{2s},u_s} \\ \frac{\partial f_2}{\partial x_1}\Big|_{x_{1s},\,x_{2s},u_s} & \frac{\partial f_2}{\partial x_2}\Big|_{x_{1s},\,x_{2s},u_s} \end{bmatrix}, \tag{7}$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u}\Big|_{x_{1s},\,x_{2s},u_s} \\ \frac{\partial f_2}{\partial u}\Big|_{x_{1s},\,x_{2s},u_s} \end{bmatrix}, \tag{8}$$

$$\mathbf{C} = \begin{bmatrix} \frac{\partial g}{\partial x_1}\Big|_{x_{1s},\,x_{2s},u_s} & \frac{\partial g}{\partial x_2}\Big|_{x_{1s},\,x_{2s},u_s} \end{bmatrix}, \tag{9}$$

$$\mathbf{D} = 0, \tag{10}$$

The operating points x_{1s} = 3.223 and x_{2s} = 0.3955 were obtained with the following script (figure 2), executed in MatLAB. The input u_s = 0.7 was chosen arbitrarily.

The partial derivatives for the matrix **A** were obtained by means of writing a script using the command *diff* (figure 2). The results are presented in figure 3.

```
syms x1 x2 mu0 k1 k2 u a K
f1 = ((mu0*x2*x1)/(k1+x2+k2*(x2^2))) - (x1*u)
f2 = ((-mu0*x2*x1)/(a*(k1+x2+k2*(x2^2)))) + u*(K-x2)
d1 = diff(f1,x1)
d2 = diff(f1,x2)
d3 = diff(f2,x1)
d4 = diff(f2,x2)
```

Figure 2: Script for the calculation of the partial derivatives [*]

```
d1 =

(mu0*x2)/(k2*x2^2 + x2 + k1) - u


d2 =

(mu0*x1)/(k2*x2^2 + x2 + k1) - (mu0*x1*x2*(2*k2*x2 + 1))/(k2*x2^2 + x2 + k1)^2


d3 =

-(mu0*x2)/(a*(k2*x2^2 + x2 + k1))


d4 =

(mu0*x1*x2*(2*k2*x2 + 1))/(a*(k2*x2^2 + x2 + k1)^2) - (mu0*x1)/(a*(k2*x2^2 + x2 + k1)) - u
```

Figure 3: Partial derivatives calculated [*]

```
syms x1 x2 ;
u = 0.7;
mu0 = 1;
K = 5;
k1 = 0.06;
k2 = 0.7;
Alpha = 0.7;
fcn1 =((mu0*x2*x1)/(k1+x2+k2*(x2^2))) - x1*u;
fcn2 = (-(mu0*x2*x1)/(Alpha*(k1+x2+k2*(x2^2))))+(K-x2)*u;
[solx1, solx2] = solve([fcn1 == 0, fcn2 == 0], [x1,x2],'real',true);
X1_Roots = zeros;
X1_Roots = vpa(solx1);
X2_Roots = zeros;
X2_Roots = vpa(solx2);
x1_Value = X1_Roots(end);
x2_Value = X2_Roots(end);
fprintf('The operating points are x1=%f,x2=%f\n',x1_Value,x2_Value);
```

```
>> Untitled6
The operating points are x1=3.223117,x2=0.395547
>>
```

Figure 2: Script written for the obtainment of the operating points (with the results) [*]

The state-space model is:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{11}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & -0.4993 \\ -1 & -0.2970 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -3.223 \\ 4.6045 \end{bmatrix} u$$

$$\boldsymbol{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \tag{12}$$

$$\boldsymbol{y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 0$$

4. Building a model of the system in MatLAB/Simulink

The model was created in Simulink according to the state-space equation (11), is presented in figure 3.

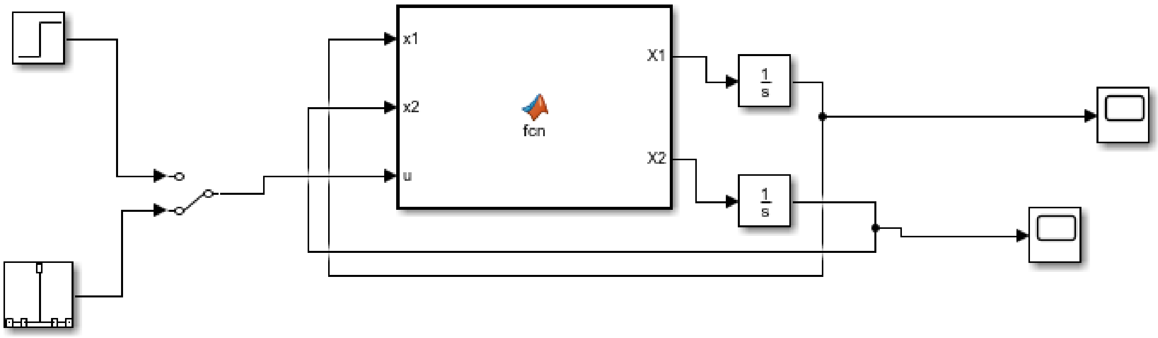


Figure 3: Simulated model of the bioreactor [*]

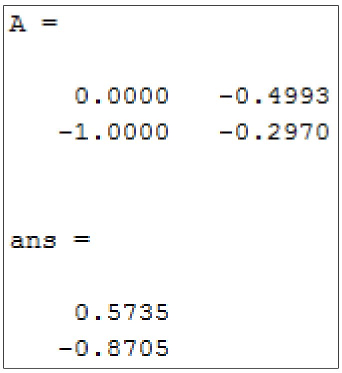
5. Analysis of the system behavior: stability, controllability, observability

5.1 Stability

There are three different categories of stability [2]:

1. *Stable* – all poles are on the left half of the S-plane;
2. *Critically stable* – at least one pole is on the left half of the S-plane;
3. *Unstable* – if all poles are on the right half of the S-plane.

The stability of the system is defined by the eigenvalues of the matrix **A**. The system can be considered as stable only in case if all the eigenvalues of this matrix are negative. The obtained eigenvalues by means of the MatLAB script are shown in figure 4.



```
A =

    0.0000   -0.4993
   -1.0000   -0.2970


ans =

    0.5735
   -0.8705
```

Figure 4: Eigenvalues obtained [*]

As it is seen from figures 4 and 5, only one eigenvalue is negative, thus the system is critically stable. It can be explained by the fact that in such systems internal dynamics takes place. A graphical representation of the position of eigen values on the real and imaginary are show in figure 5.

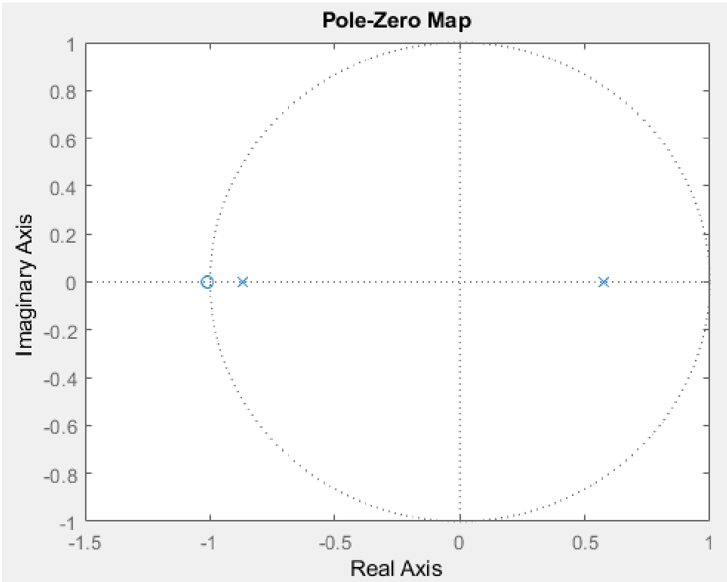


Figure 5: Eigenvalues of the state-space model [*]

5.2 Controllability

To check the controllability of the system, it is possible to use the MatLAB command *ctrb* for it (figure 6). The criteria for the controllability is the equality of the rank of the matrix [**B** | **AB**] to the order of the matrix **A**. The results are shown in figure 7.

```
Con = ctrb(A,B)
rank(Con)
```

Figure 6: Controllability command [*]

```
Con =

   -3.2230   -2.2988
    4.6045    1.8553


ans =

     2
```

Figure 7: Results of the calculation in MatLAB [*]

According to the results, the rank of the matrix **Con** (created for the controllability check) is 2, which is equal to the order of the matrix **A** (n×n), thus the system is controllable.

5.3 Observability

A state observer is predestined for the evaluation of state variables according to the measurements of control variables and output [2]. To design it, it is necessary to check the system in terms of the observability. For checking the observability of the system, it is possible to use the MatLAB command *obsv* (figure 8). The criteria for the observability is the equality of the rank of the matrix $[\mathbf{C}^{\mathbf{T}} \mid \mathbf{A}^{\mathbf{T}}\mathbf{C}^{\mathbf{T}}]$ to the order of the matrix **A**. The results are shown in figure 9.

```
Obs = obsv(A,C)
rank(Obs)
```

Figure 8: Observability command [*]

```
Obs =

    1.0000         0
    0.0000   -0.4993


ans =

     2
```

Figure 9: Results of the calculation in MatLAB [*]

According to the results, the rank of the matrix **Obs** (created for the observability check) is 2, which is equal to the order of the matrix **A** (n×n), thus the system is observable.

6. Design of a state space controller. In addition, design of a state observer

6.1 Design of the state space controller by the pole placement approach

The task was conducted by means of writing a script in MatLAB with specifying system poles arbitrarily:

$$J = [-2 + 0.5j \quad -2 - 0.5j]$$

Then, the gain matrix **K** was calculated with the command *place* (figure 10).

```
%gain matrix
Kp = place(A,B,J);
```

Figure 10: Pole placement command [*]

The **K** matrix is:

$$K = [-2.1568 \quad -0.7054]$$

With the usage of the **K** matrix, the controller was designed, the code of the script is shown in figure 11.

```
A = [0 -0.4993; -1 -0.2970];
B = [-3.2230; 4.6045];
C = [1 0];
% pole placement
J = [-2+0.5*j -2-0.5*j];
% gain matrix
Kp = place(A,B,J)
sys = ss(A-B*Kp,eye(2),eye(2),eye(2));
t = 0:0.1:10;
x = initial(sys,[3.22;0.3955],t)
x1 = [1 0]*x';
x2 = [0 1]*x';
subplot(2,1,1); plot(t,x1);grid
title('Response to initial condition')
ylabel('state variable x1')
subplot(2,1,2); plot(t,x2);grid
ylabel('state variable x2')
xlabel('t(sec)')
```

Figure 11: Script used for the controller design [*]

The obtained results with the **K** matrix are specified in figure 12.

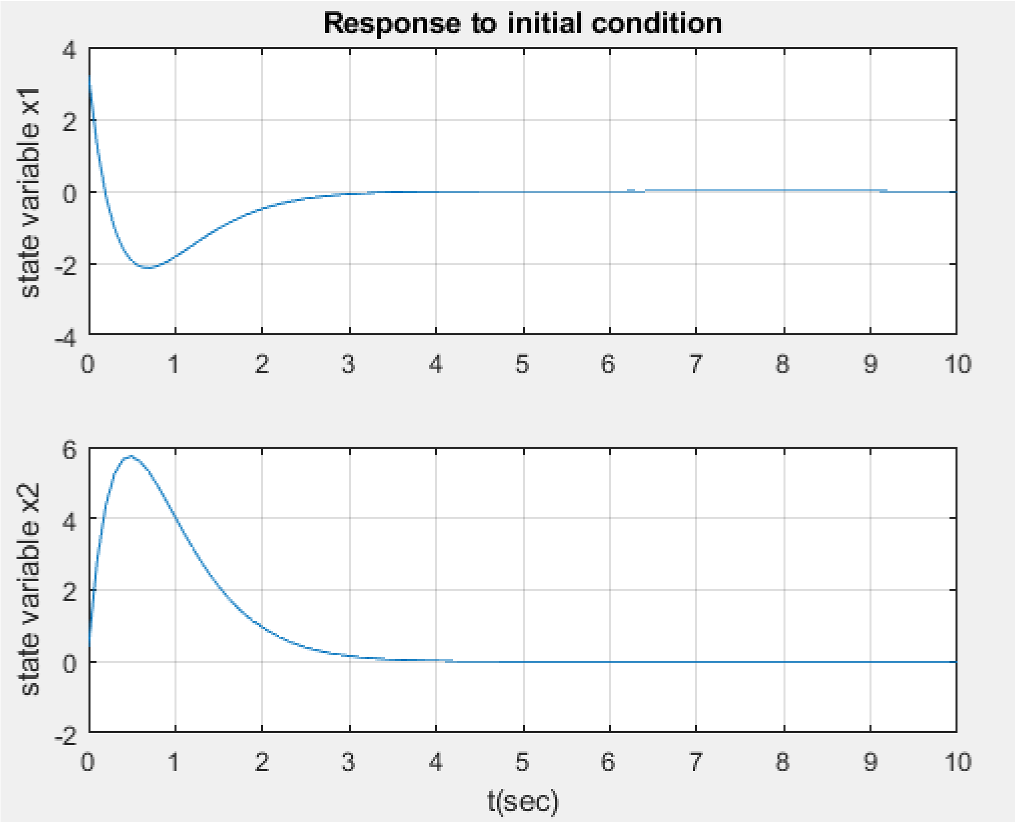


Figure 12: Result of the script with the K matrix [*]

According to the graphs, the concentrations of biomass x_1 and substrate x_2 stabilize approximately within 3 seconds. The model of the controller built in Simulink is shown in figure 13.

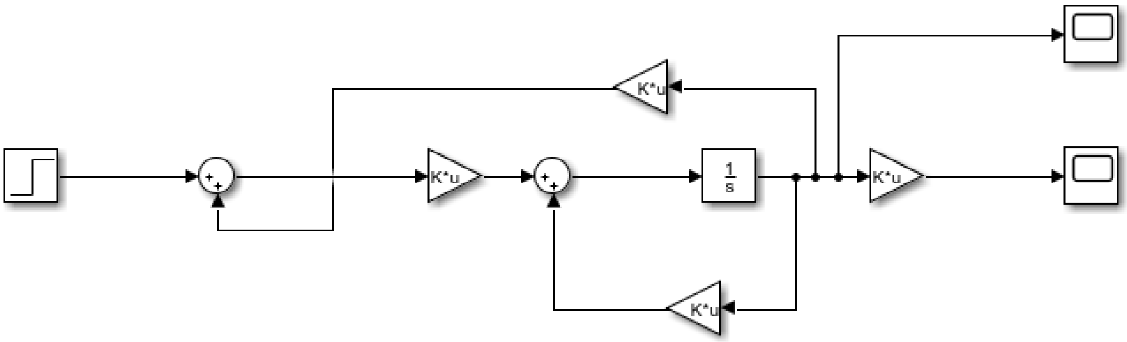


Figure 13: Model of the controller [*]

6.2 Design of the state observer by the pole placement approach

To design the observer, the following **L** matrix with the observer pole values was taken:

$$L = [-5 \quad -6]$$

Then, the gain matrix **K** was calculated with the command *place* (figure 14).

```
%gain matrix
Ke = place(A',C',L)'
```

Figure 14: Pole placement command [*]

The **K** matrix is:

$$K = [10.703 \quad -54.7176]$$

With the usage of the **K** matrix, the observer was designed, the code of the script is shown in figure 15.

```
A = [0 -0.4993; -1 -0.2970];
B = [-3.2230; 4.6045];
C = [1 0];
J = [-2+0.5*j -2-0.5*j];
L = [-5 -6];
%gain matrix
Kp = place(A,B,J)
Ke = place(A',C',L)';
sys = ss([A-B*Kp B*Kp;zeros(2,2) A-Ke*C],eye(4),eye(4),eye(4));
t = 0:0.1:10;
x = initial(sys,[3.22;0;0.3955;0],t)
x1 = [1 0 0 0]*x';
x2 = [0 1 0 0]*x';
e1 = [0 0 1 0]*x';
e2 = [0 0 0 1]*x';
subplot(2,2,1); plot(t,x1),grid
title('Response to initial condition')
ylabel('state variable x1')
subplot(2,2,2); plot(t,x2),grid
title('Response to initial  condition')
ylabel('state variable x2')
subplot(2,2,3); plot(t,e1),grid
xlabel('t(sec)'), ylabel('error state variable e1')
subplot(2,2,4); plot(t,e2),grid
xlabel('t(sec)'),ylabel('error state variable e2')
```

Figure 15: Script used for the observer design [*]

The obtained results with the **K** matrix are specified in figure 16.

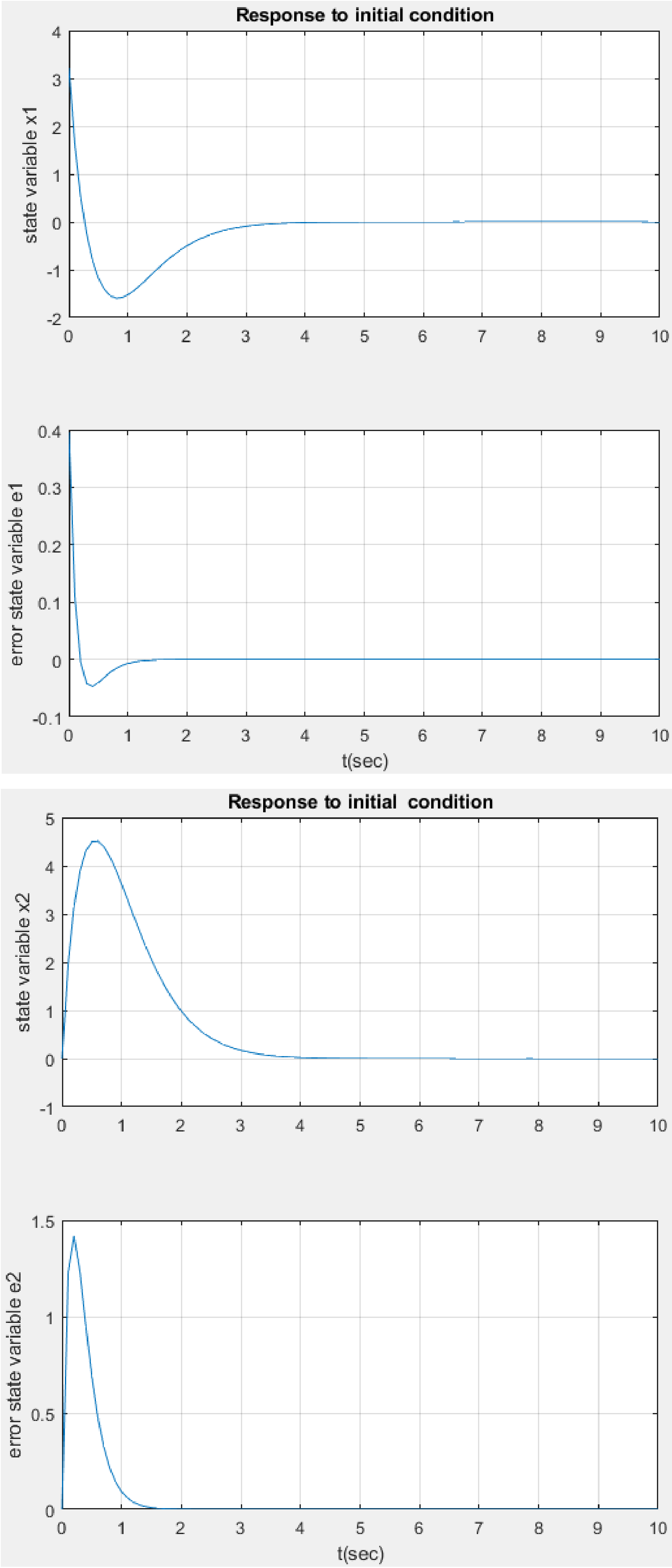


Figure 16: Result of the script with the K matrix [*]

According to the graphs, the concentrations of biomass x_1 and substrate x_2 stabilize approximately within 3 seconds with this initial error. The model of the observer built in Simulink is shown in figure 17.

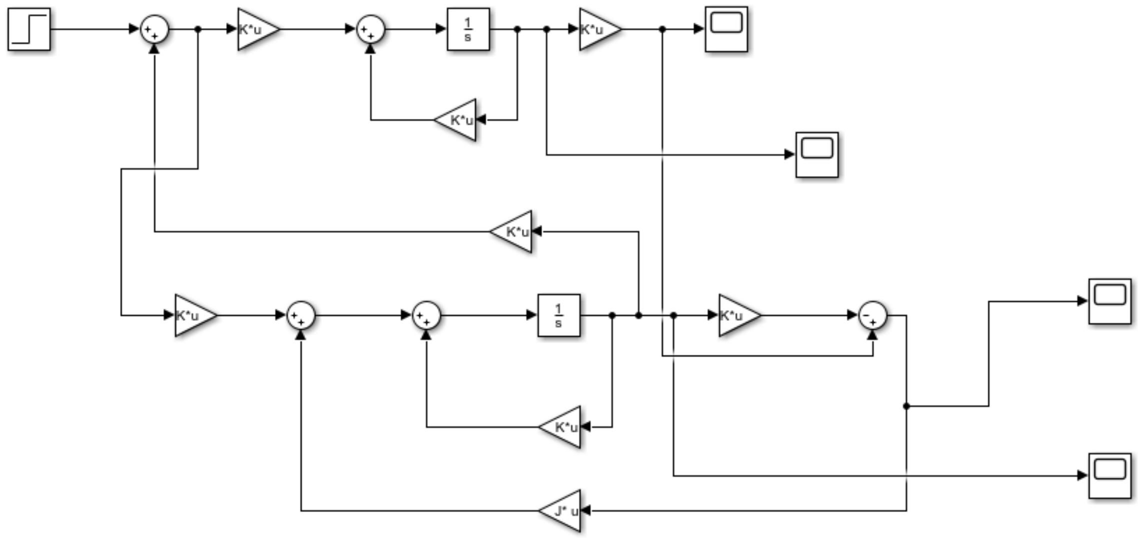


Figure 17: Model of the controller [*]

7. Analysis of the controller and observer design and the presentation of obtained results

7.1 Analysis of the state-space controller

During the analysis of the system, various poles to the left of the S-plane were considered to understand which ones are the most suitable for the controller design. If the poles are located far away from the origin the system is characterized by high disturbances until reaching the stable state (figure 18). And vice versa, if the poles are located close to the origin, the disturbances are minimized (figure 19). Thus, the poles closer to the origin are more preferable since the process goes smoother.

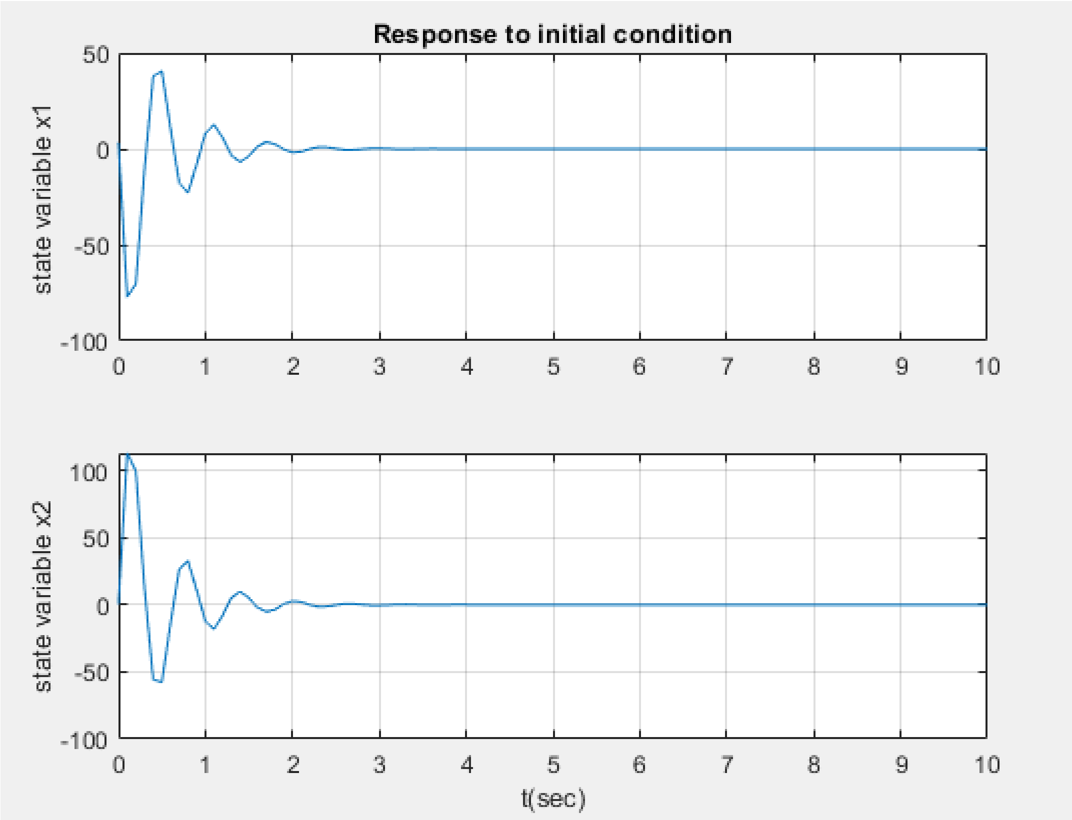


Figure 18: Controller response for the poles [-2-10j -2+10j] [*]

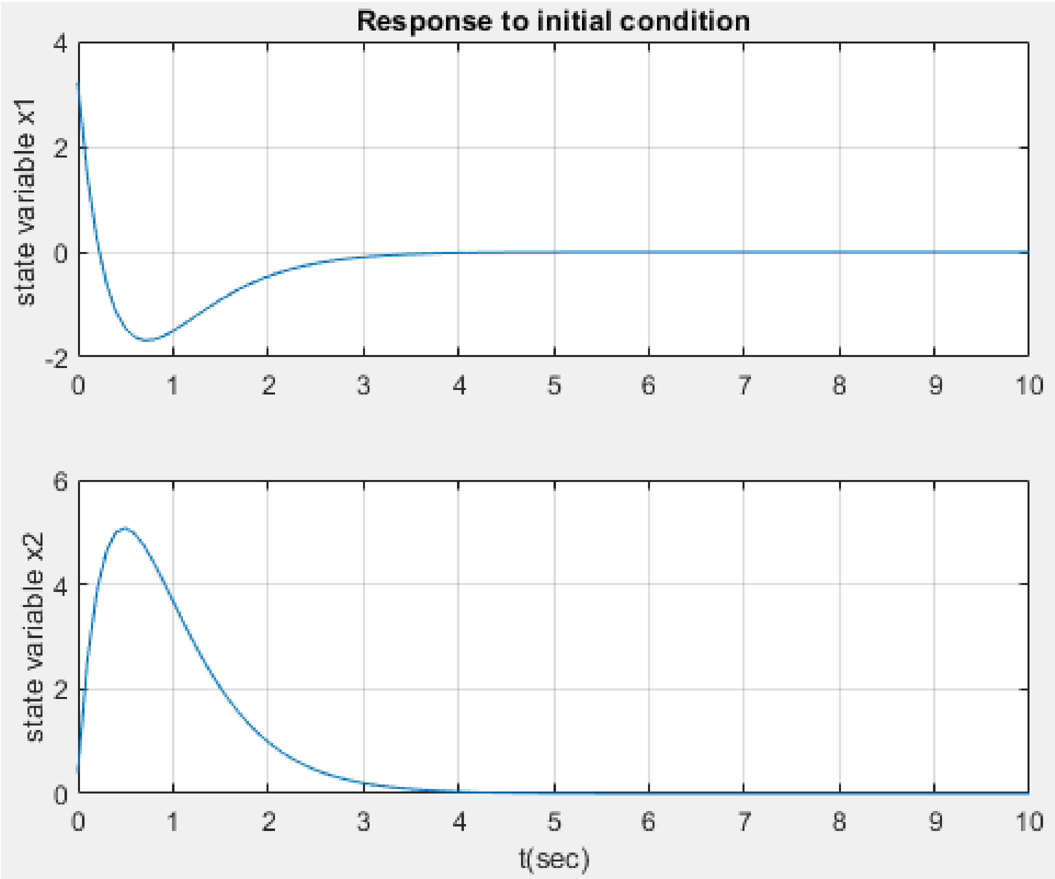


Figure 19: Controller response for the poles [-2-0.1j -2+0.1j] [*]

7.2 Analysis of the state-space observer

According to subheading 6.2, the observer stabilizes approximately within 3 seconds, which is the same with the controller, because of the fact that the observer poles are close to the origin. However, according to lecture №4, the state observer has to stabilize several times faster than the controller. Thus, the poles $[-8 \quad -9]$ were chosen, and it seen that the response time is 3 times faster than the controller (figure 20).

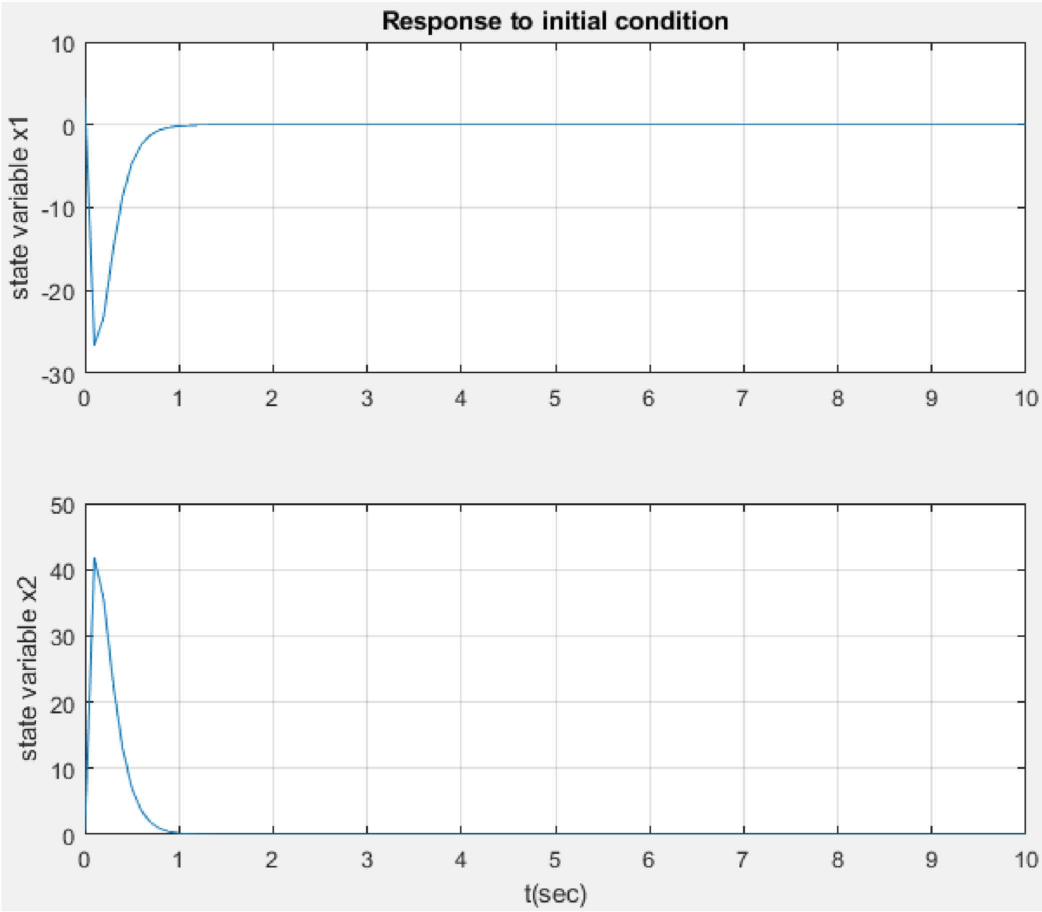


Figure 20: Controller response for the poles [-8 -9] [*]

8. Conclusion

During the assignment, the controller state-space model has been analyzed in detail, including linearization of the system at the operating points, analysis of the system behavior and with controllability and observability. Also, for the controller and observer design, the pole placement method has been applied and it should be noted that the selection of appropriate poles plays a major role in controlling the entire system.

References

[1] Bengt Carlsson, “An introduction to modeling of bioreactors”. Information Technology Uppsala University, March 24, 2009, pp. 1-2

[2] K. Ogata, Modern control engineering, 5th ed. Pearson, 2010, pp. 39-762