King Abdulaziz University
King Fahd Medical Research Center

Stabilization of a CUBE System

By
Ahmad Turki

Project Done at The University of Texas at San Antonio

1. Introduction

Our aim in this project is to design a State-Feedback controller with a state estimator in its feedback loop in order to balance a cube on one of its edges. Internally, a powered pendulum is used to apply a torque to the body of the cube. A piezoceramic gyroscope is used to sense the rate of rotation of the cube. No measurement of the actual cube angle is available. The only other available measurement is the relative angle between the pendulum and the cube. Hence, in order to stabilize the system, an estimator has to be constructed to estimate the other three un-measurable states. Moreover, the governing dynamics of the system are nonlinear. The first part of this project will concentrate on the design of the State-Feedback controller assuming that all output states are available for feedback. The goal in this first section is to stabilize the output of the system to zero. In the following section, the goal is to emulate a more realistic system in which only few output states are available for feedback. Therefore, the design of a state estimator in the feedback loop will be required in order to estimate the missing state and stabilize the system. MATLAB is used to design the controller and estimator, while Simulink is used to simulate the physical system behavior of the CUBE hardware.

2. Detailed Design

The design process was divided into three parts

- Task Modeling
- State-Feedback Control
- Output Feedback Control

2.1 Task 1: Modeling

In this section, the linearized model of the CUBE system characterized as a state space model, and the task is to verify the controllability and observability of the cube system to ensure whether or not our system is controllable and stable.

Our state space model of the system is:

The linearized model of the CUBE system can be described by the following state space model

$$\frac{dX}{dt} = Ax + Bu \qquad y = Cx + Du \qquad \text{(Eqn.1)}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 41.5436 \\ 41.5436 & -46.9409 & 0 & -191.6471 & -36.3892 \\ -36.3892 & -101.9403 & 0 & -786.3512 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 33.7528 \\ 138.4917 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

We verified the controllability and observability by calculating the rank of each of their matrices. To verify the controllability, we calculated the rank of the matrix $[B,AB,A^2B,A^3B,A^4B]$. If the rank of the matrix is 5, we can conclude that this system is controllable. The controllability was done using MATLAB by using the following command:

```
ControlMatrix = ctrb(A,B);
if length(A)==rank(ControlMatrix)
   sprintf('%s','The Cube system IS CONTROLLABLE')
else
   sprintf('%s','The Cube system IS NOT CONTROLLABLE')
end

ans =

The Cube system IS CONTROLLABLE
```

In a second step, the observability of the system was verified by calculating the rank of the matrix:

$$\begin{bmatrix} C \\ CA \\ C^2A \\ C^3A \\ C^4A \end{bmatrix}.$$

Similar to the controllability test, if the rank is 5, we can conclude that this system is observable. The observability was done using MATLAB by using the following command:

```
ObservMatrix = obsv(A,C);
if length(A)==rank(ObservMatrix)
   sprintf('%s','The Cube system IS OBSERVABLE')
else
   sprintf('%s','The Cube system IS NOT OBSERVABLE')
end

ans =

The Cube system IS OBSERVABLE
```

2.2 Task 2: State-Feedback Control

The second part of the project consisted of designing a state-feedback controller to stabilize the system at zero. The Simulink model was provided from Blackboard and shown in Figure 1 below.

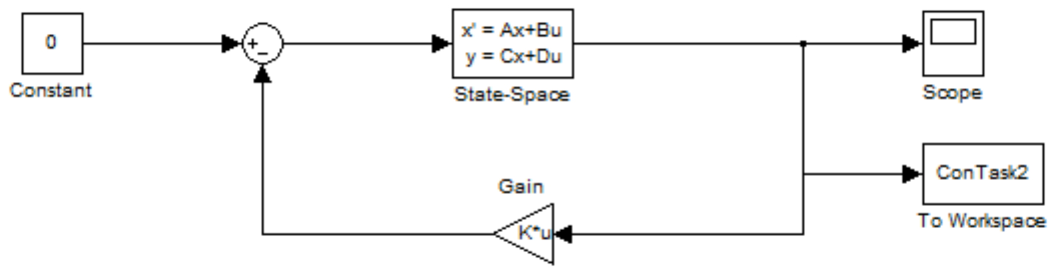

Figure 1: Task 2 Simulink Model

As noted, the output matrix C is a 5x5 identity matrix since, in this particular task, we assume we have all the information of the states. The desired eigenvalues of the closed loop system were given as (-15,-14,-13,-12,-11). In order to design a state-feedback controller $u = -Kx$ to stabilize the system to zero, we used a provided function of MATLAB command called "acker()". This function outputs the K values based on the matrix A, B and the desired eigenvalues of the closed loop system. The resulting K was then inputted in the State-Space Function Block of the Simulink tool to simulate the State-Feedback controller. We computed the value of K manually then we verified by using MATLAB.

The desired eigenvalues are [-15, -14, -13, -12, -11]

$$\det\left\|\left[sI_5 - A\right]\right\| = s^5 + \alpha_1 s^4 + \alpha_2 s^3 + \alpha_3 s^2 + \alpha_4 s + \alpha_5$$

$$\det\left\|\left[sI_5 - A\right]\right\| = \det\left\|\begin{bmatrix} s & 0 & 0 & 0 & 0 \\ 0 & s & 0 & 0 & 0 \\ 0 & 0 & s & 0 & 0 \\ 0 & 0 & 0 & s & 0 \\ 0 & 0 & 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 41.5436 \\ 41.5436 & -46.9409 & 0 & -191.6471 & -36.3892 \\ -36.3892 & -101.9403 & 0 & -786.3512 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}\right\|$$

$$= s^5 + 786.3512s^4 + 18.8501s^3 - 72309.6041s^2 - 4217.2381s + 1645536.5937$$

By comparing with the previous result we found the following:

$$\alpha_1 = 786.3512$$

$$\alpha_2 = 18.8501$$

$$\alpha_3 = -72309.6041$$

$$\alpha_4 = -4217.2381$$

$$\alpha_5 = 1645536.5973$$

Then we had to find P^{-1} to find the canonical form to find K

$$p^{-1}=[A \quad AB \quad A^2B \quad A^3B \quad A^4B]\begin{bmatrix}1 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4\\0 & 1 & \alpha_1 & \alpha_2 & \alpha_3\\0 & 0 & 1 & \alpha_1 & \alpha_2\\0 & 0 & 0 & 1 & \alpha_1\\0 & 0 & 0 & 0 & 1\end{bmatrix}$$

$$[A \quad AB \quad A^2B \quad A^3B \quad A^4B]=\begin{bmatrix}0 & 33.7528 & -2.6542\times10^{4} & 2.0866\times10^{7} & -1.6405\times10^{10}\\0 & 138.4917 & -1.0890\times10^{5} & 8.5627\times10^{7} & -6.7320\times10^{10}\\33.7528 & -2.6542\times10^{4} & 2.0866\times10^{7} & -1.6405\times10^{10} & 1.2898\times10^{13}\\138.4917 & -1.0890\times10^{5} & 8.5621\times10^{7} & -6.7316\times10^{10} & 5.2924\times10^{13}\\0 & 0 & 138.4917 & -1.0890\times10^{5} & 8.5627\times10^{7}\end{bmatrix}$$

$$p^{-1}=[A \quad AB \quad A^2B \quad A^3B \quad A^4B]\begin{bmatrix}1 & 786.3512 & 18.8501 & -72309.6041 & -4217.2381\\0 & 1 & 786.3512 & 18.8501 & -72309.6041\\0 & 0 & 1 & 786.3512 & 18.8501\\0 & 0 & 0 & 1 & 786.3512\\0 & 0 & 0 & 0 & 1\end{bmatrix}$$

$$p^{-1}=\begin{bmatrix}0 & 33.7528 & 0.0221 & -4462.3673 & -5040.5205\\0 & 138.4917 & 0 & -6981.6812 & 0\\33.7528 & 0.0221 & -4462.3673 & -5040.5205 & 0\\138.4917 & 0 & -12735.1250 & 0 & 290044.1719\\0 & 0 & 138.4917 & 0 & -6981.6812\end{bmatrix}$$

The characteristic polynomial is:

$$(s+15)(s+14)(s+13)(s+12)(s+11)=s^5+65s^4+1685s^3+21775s^2+140274s+360360$$

$\beta_1=65$, $\beta_2=1685$, $\beta_3=21775$, $\beta_4=140274$, $\beta_5=360360$

$\overline{K_i}=\beta_i-\alpha_i$ with $i=1,2,3,4,5$

$\overline{K}=[-721.3512 \quad 1666.1469 \quad 94084.6041 \quad 144491.2381 \quad -1285176.5973]$

$K=\overline{K}p=[-65.8026 \quad 28.0694 \quad -9.2902 \quad -2.9444 \quad 109.2625]$

Then to verify the K value by using MATLAB we applied the following code:

```
K=acker(A,B,desiredEig)
```

K =

-65.8026   28.0694   -9.2902   -2.9444 109.2625

After finding the K value, we were ready to run the controller Simulink model. The initial conditions were set to x(0)=[1;1;1;1;1]. By observing the system states simulation's figures, we figured that the system overshoots at the beginning then being stabilized.

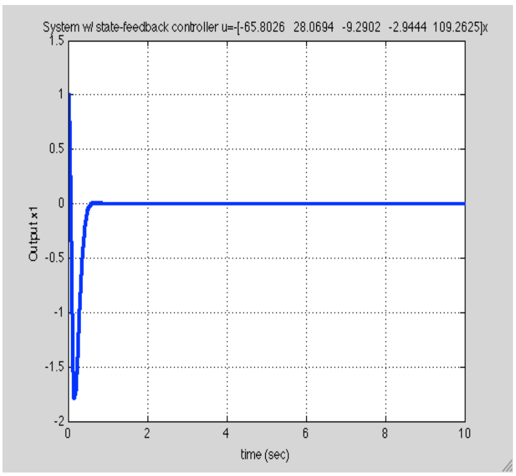

Figure 2: system state 1 simulation

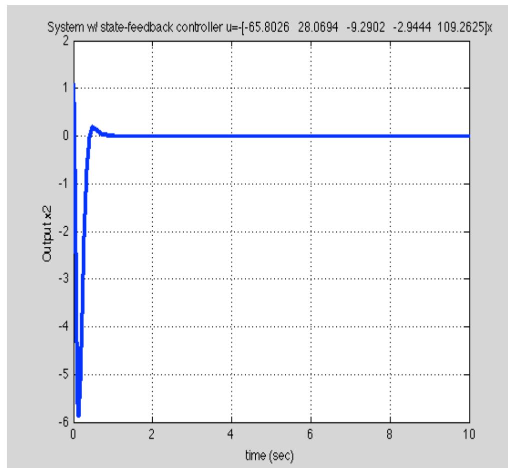

Figure 3: system state 2 simulation

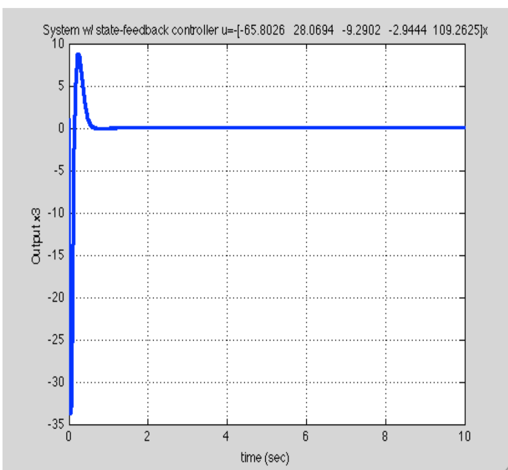

Figure 4: system state 3 simulation

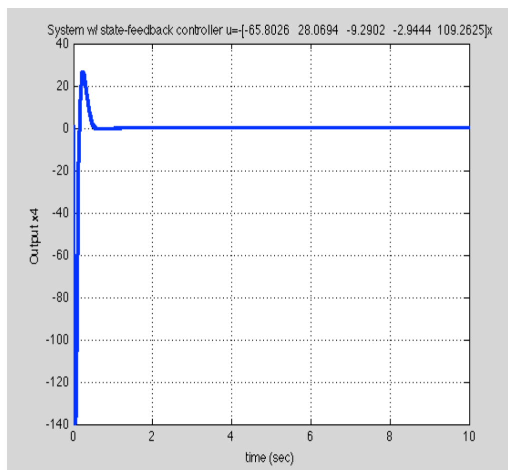

Figure 5: system state 4 simulation

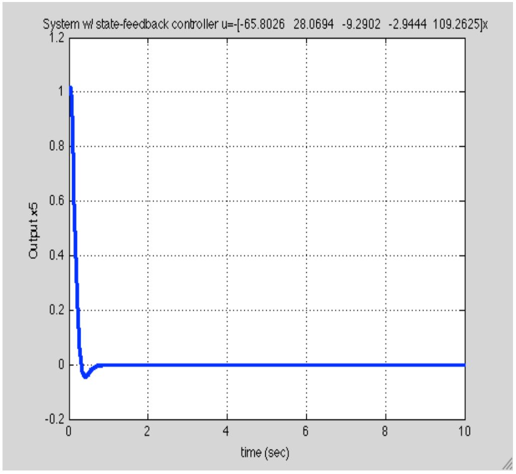

Figure 6: system state 5 simulation

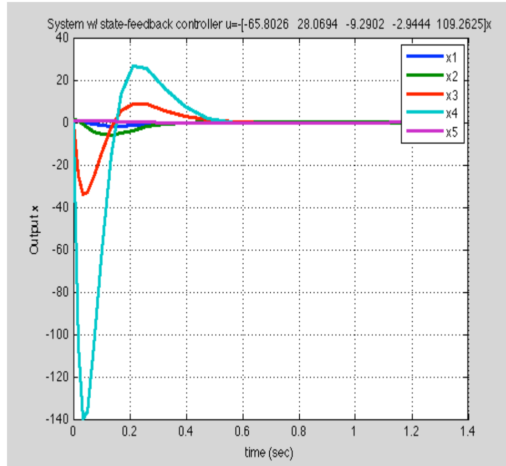

Figure 7: system all states simulation

2.3 Task3: State-Feedback Control:

The third part consisted of output feedback control of inverted pendulum. Since in the real cube system only two states are available, we need to design a controller only using the measurable states x_2 and x_3 and estimate the other three states using a State Space Estimator Function Block. The value in matrix C will remain the same as the 5x5 identity matrix in our previous Simulink model. We are asked to design a state estimator to estimate the states x based on the output y and input u of the system. Desired eigenvalues of the error dynamic are [-11 -12 -13 -14 -15]. We implemented the estimator in Simulink using a new state space model whose inputs are y and u and outputs are the estimated states. We changed the State Space Estimator Function Block to reflect the inputs and outputs of the estimator. The controller gain K, which was used in Design Task 2, was also utilized in Task 3.

The CUBE System has only two available states, therefore to design a controller we only need to use the measurable states x_2 and x_3. To design a state estimator we used the following Simulink model:

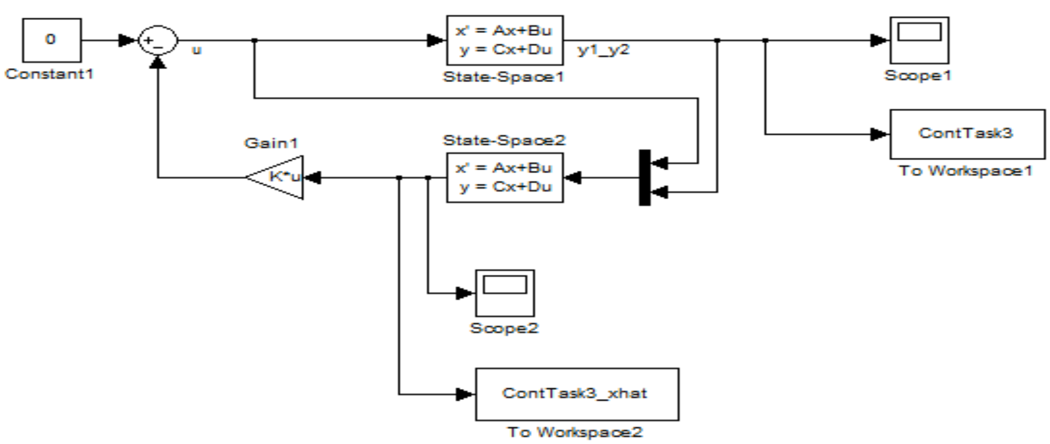

Figure 8: Task 3 Simulink Model

Designing a state estimator to estimate the state x based on the output y and input u of the system with desired eigenvalues of the error dynamic are [-11 -12 -13 -14 -15], will require calculating the value of L. L computing was done using MATLAB, by using the following command:

```
 P = [-11,-12,-13,-14,-15];
>> L = place(A',C',P)'
L =
  1.0e+003 *
    0.0000    0.0011
    0.0295    0.0041
   -0.7244   -0.7509
   -2.7902   -3.0831
    0.0051    0.0000
```

After that we implemented the estimator in the Simulink using a new state space model whose inputs are y and u and outputs are the estimated states, to reflect the input and output of the estimator. We calculated the new state space model according the following:

```
A_hat=A-L*C;
B_hat=[L,B];
C_hat=eye(5);
D_hat=zeros(5,3);
```

After running Simulink, we got the figures below. We observed from the scope figures that how outputs 1 and 2 of the CUBE system overshoot at the beginning and smooth down after that. Therefore, we can conclude that the controller was able to stabilize the system.

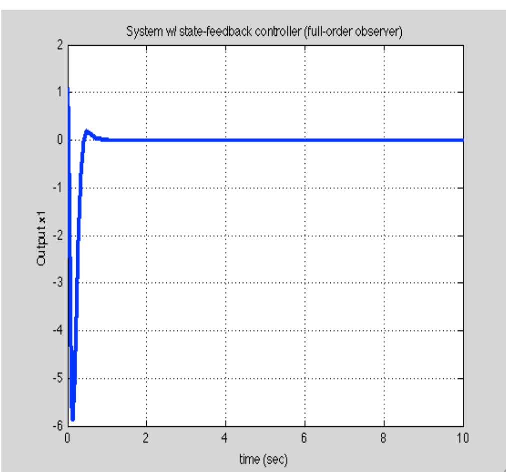

Figure 9: system output 1 simulation

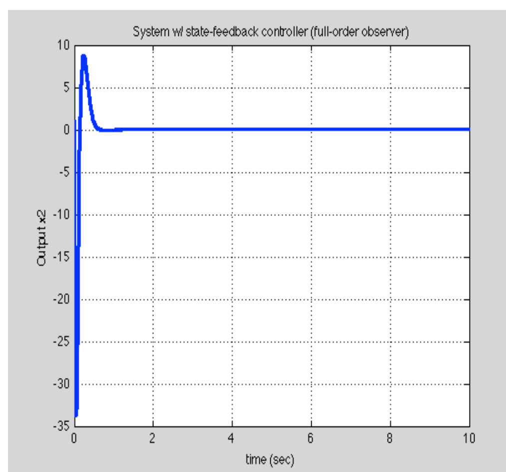

Figure 10: system output 2 simulation

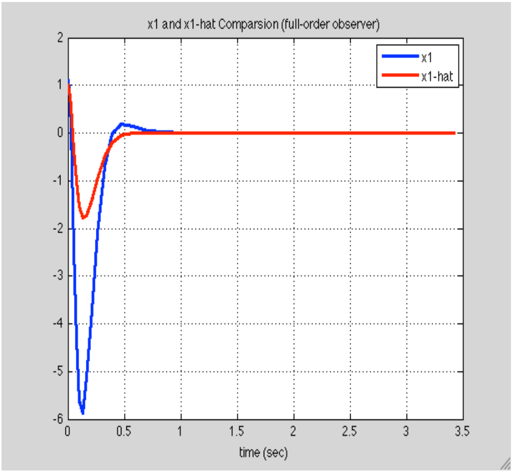

Figure 11: state 1 with observer simulation

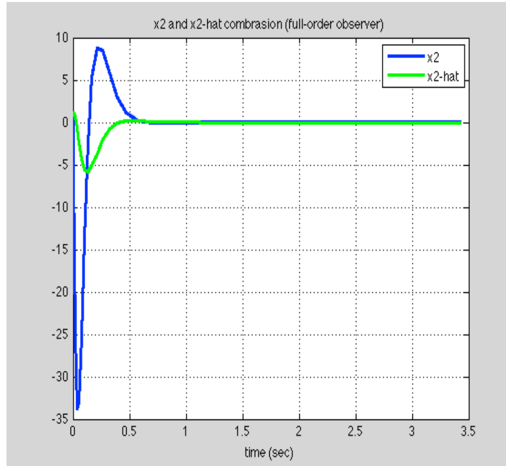

Figure 12: state 2 with observer simulation

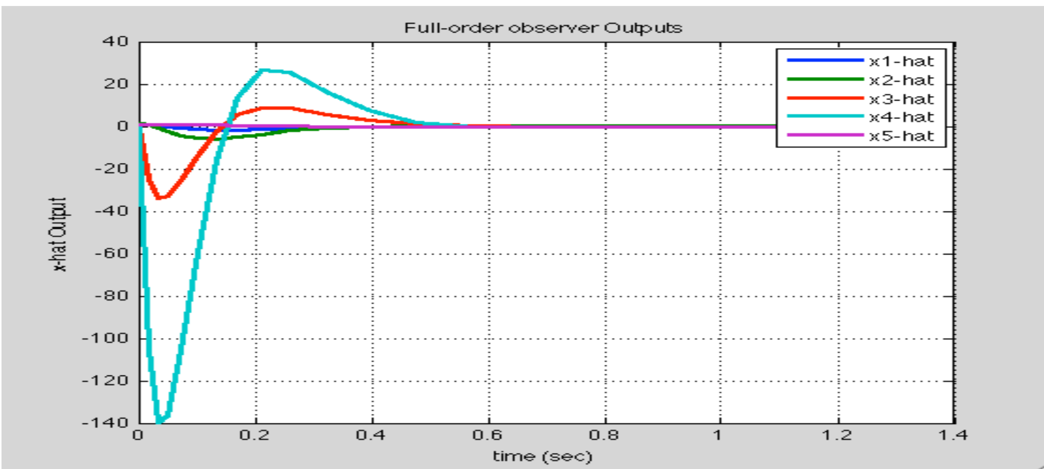

Figure 13: All states with obsever simulation

3. Conclusions:

A state feedback controller was designed to balance a cube system on one of the edges, with no direct angel measurements. Furthermore, a state estimator was designed to estimate two measurable stets of the system with desired given eigenvalues of error dynamics. All the design simulations executed using the MATLAB Simulink. From the resulted scopes figures we concluded that the cube system is balanced on one of its edge, with no any angel measurements.