CORO 2021

Dynamic Model Based Control Lab (State Space approach for Linear Systems)

Author:
Ke GUO
Zixuan XU

Supervisor:
Guy LEBRET

10/04/2022

Contents

1 Introduction

In this lab we are asked to control the angle of a pendulum on a cart as demonstrated on Fig.1. The rotation of the pendulum is totally free. The motion of the cart is controlled by the force u.

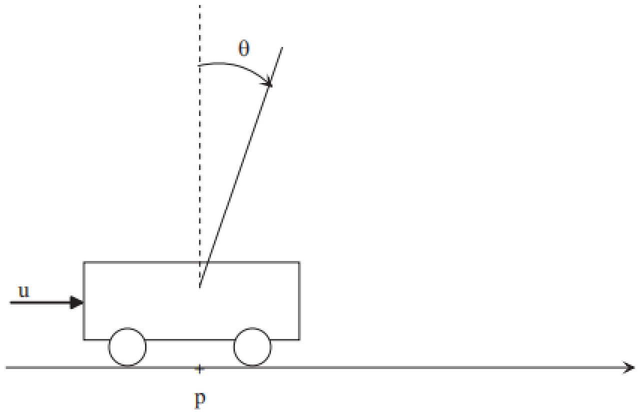


Figure 1: The inverted pendulum model

We ignore the effect of friction on this problem. The mechanical equations are shown as eq(1) and eq(2).

$$\ddot{\theta}[mL\cos^2\theta - (M+m)L] - mL\dot{\theta}^2\sin\theta\cos\theta + (M+m)g\sin\theta = u\cos\theta \quad (1)$$
$$\ddot{p}[m\cos^2\theta - (M+m)] + mL\dot{\theta}^2\sin\theta - mg\sin\theta\cos\theta = -u \quad (2)$$

The meaning of these variables:

— θ, the angle between the vertical axis and the pendulum.
— p, the position of the cart.
— m, the mass of the pendulum. It is supposed to be concentrated in its center of gravity, located at a distance L of the rotating point.
— M, the mass of the cart.

We consider some small variations around the equilibrium point defined by $\theta = 0$ and any p. Therefore we can use a linearized version of this system to design a controller.

2 State Space Model of The System

1. Show the linearized model around the equilibrium point. The equilibrium point is defined by $\theta = 0$. The initial value of θ is around 0, so we consider $\dot{\theta} = 0$, $sin\theta = \theta$ and $cos\theta = 1$. From eq(1) and eq(2) we can obtain:

$$\ddot{\theta}[mL - (M+m)L] - 0 + (M+m)g\theta = u \quad (3)$$
$$\ddot{p}[m - (M+m)] - mg\theta = -u \quad (4)$$

From eq(3) and eq(4) we can obtain:

$$\ddot{\theta} = \frac{m+M}{ML} - \frac{1}{ML}u \tag{5}$$
$$\ddot{p} = -\frac{mg}{M}\theta + \frac{1}{M}u \tag{6}$$

To build the model, we choose the state vector to be $[\theta, \dot{\theta}, p, \dot{p}]^T$. Considering that $\dot{\theta} = \dot{\theta}$ and $\dot{p} = \dot{p}$, we can get the design model (7).

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{m+M}{ML}g & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{M} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{ML} \\ 0 \\ \frac{1}{M} \end{bmatrix} u \tag{7}$$

2. Give the numerical values of the matrices A and B of the state space representation. Using Matlab, we can calculate A and B as eq(8) shows.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 10.78 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -0.98 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ -0.2 \\ 0 \\ 0.2 \end{bmatrix}, \tag{8}$$

3. What are the eigenvalues associated to this linear state space representation? We Use Matlab function $[v, D] = eig(A)$ to get the eigenvalues. The results show that the eigenvalues are $\lambda_1 = \lambda_2 = 0$, $\lambda_3 = 3.2833$, $\lambda_4 = -3.2833$.

3 State Feedback Controller

In this part we suppose that we can access the whole state as if sensors could give the angle θ, its derivative $\dot{\theta}$, the position p and its derivative $\dot{p}$.

3.1 State Feedback

First we focus on the controllability of this system. The controllability matrix is defined as eq.(9).

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{9}$$

Based on eq.(8) or we can use built-in Matlab function *ctrb(A,B)* to get the controllability matrix. The controllability matrix of this system is as eq.(10).

$$\mathcal{C} = \begin{bmatrix} 0 & -0.2 & 0 & -2.156 \\ -0.2 & 0 & -2.156 & 0 \\ 0 & 0.2 & 0 & 0.1960 \\ 0.2 & 0 & 0.1960 & 0 \end{bmatrix} \tag{10}$$

Then we can use mathematical method or built-in Matlab function *rank()* to calculate the rank of this matrix. And it equals to 4 which means this system or the pair

(A, B) is controllable and all modes are controllable whether they are stable or not. **Therefore it is possible to stabilize the pendulum and the cart around an equilibrium point with a static state feedback and can place freely all the eigenvalues**.

Then by using Matlab built-in function *-acker(A, B, eigenvalues)* we can calculate the state feedback F with eigenvalues are $-1 \pm j$ and $-2 \pm j$ as eq.(11).

$$F = \begin{bmatrix} 152.0633 & 42.2499 & 8.1633 & 12.2449 \end{bmatrix} \tag{11}$$

Then based on the control law $u = Fx + v$ we construct the Simulink block diagram of the closed loop. The diagram structure and the configuration of the subsystem are shown in Fig.2 and Fig.3 respectively.

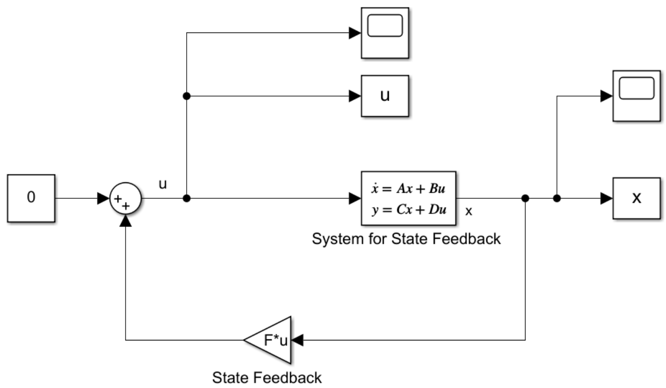


Figure 2: The Simulink block diagram of the closed loop

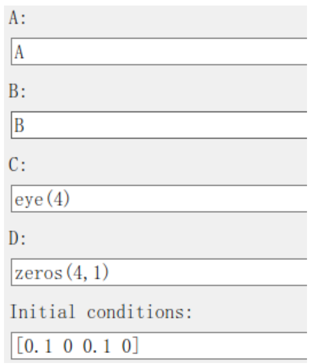

A:
A
B:
B
C:
eye(4)
D:
zeros(4, 1)
Initial conditions:
[0.1 0 0.1 0]

Figure 3: The configuration of the subsystem

In Fig.2, we can notice that we set the $v(t) = 0$ and send $x(t)$ and $u(t)$ to workspace to check the performance of this control law under initial conditions $\theta(0) = 0.1\ rd$, $\dot{\theta}(0) = 0\ rd/s$, $p(0) = 0.1\ m$ and $\dot{p}(0) = 0\ m/s$. To be able to observe x, we adjust the parameters of the subsystem so that its output becomes x instead of y. So we adjust C to $eye(4)$ and D to $zeros(4,1)$ following state vector and input dimensions. We can observe that both $x(t)$ and $u(t)$ tend to approach 0 even though the initial conditions are not 0 in Fig.4.

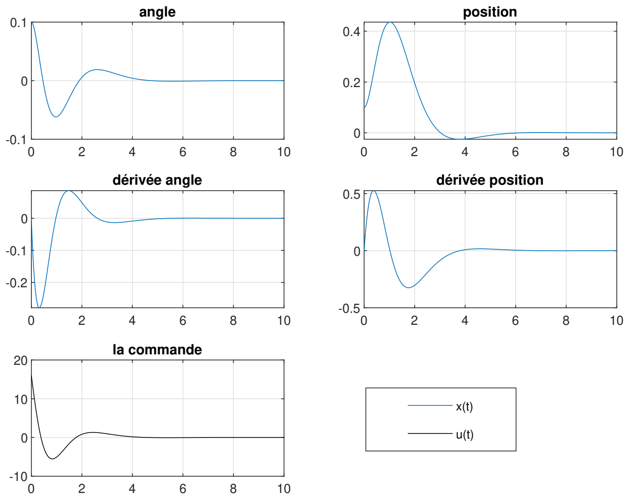


Figure 4: The plot of $x(t)$ and $u(t)$ under state feedback

3.2 Observer

In this part we consider only two sensors are available, one gives the angle θ and the other one gives the position of the cart p. Therefore the matrix C of this system can be set as eq.(12).

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{12}$$

The observability matrix is defined as eq.(13).

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{13}$$

Based on eq.(8) and eq.(12) or we can use built-in Matlab function *obsv(A,C)* to get the observability matrix of this system is as eq.(14).

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 10.78 & 0 & 0 & 0 \\ -0.98 & 0 & 0 & 0 \\ 0 & 10.78 & 0 & 0 \\ 0 & -0.98 & 0 & 0 \end{bmatrix} \tag{14}$$

Then we can use mathematical method or built-in Matlab function *rank()* to calculate the rank of this matrix. And it equals to 4 which means **this system is**

observable.

Then by using Matlab built-in function *place(A.', C.', eigenvalues).'* that this function is for MIMO situation and the controllavility and the observability is a pair of dual problem, we can calculate the observer K_1 with faster poles are $-100 \pm j$ and $-200 \pm j$ as eq.(15) and the observer K_2 with eigenvalues are are $-1 \pm j$ and $-2 \pm j$ as eq.(16).

$$K_1 = \begin{bmatrix} 295.46 & 2.30 \\ 19201 & 281.58 \\ -26.87 & 304.54 \\ -5076.4 & 20774 \end{bmatrix} \tag{15}$$

$$K_2 = \begin{bmatrix} 2.9998 & -0.9985 \\ 14.78 & 0.0043 \\ 1.0015 & 3.0002 \\ -0.9757 & 3.999 \end{bmatrix} \tag{16}$$

Then based on the previous Simulink block diagram we add an observer. The diagram structure and the configuration of the observer subsystem are shown in Fig.5 and Fig.6 respectively.

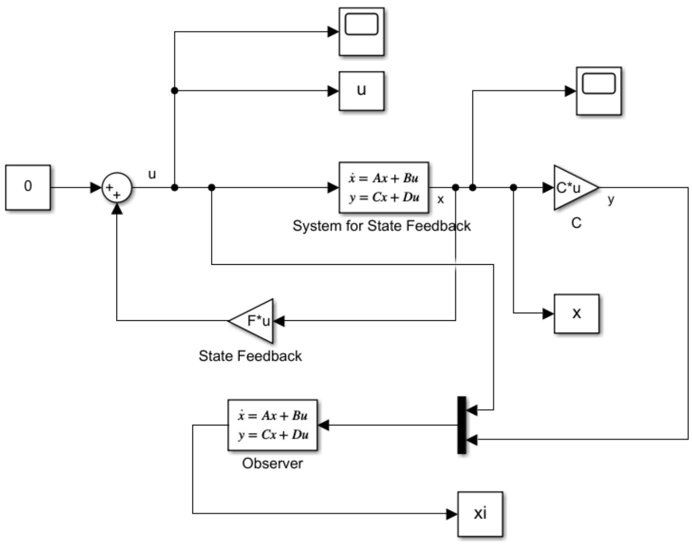


Figure 5: The Simulink block diagram after adding the observer

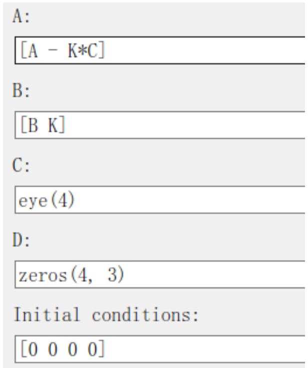


Figure 6: The configuration of the observer system

In Fig.5, it can be noticed that we use a *gain* module to get state vector x and output y respectively in this diagram and we use a tricky method that we use a state space subsystem to represent the observer. To set A and B of this subsystem we follow the eq.(17). And for C and D it is as same as the previous state feedback part but we should notice that the input of this subsystem is a composition of u and y, therefore the input dimension should be 3. And xi in Fig.5 stands for estimated state vector $\hat{x}$.

$$\dot{\hat{x}} = (A - KC)\hat{x} + Bu + Ky \tag{17}$$

We plot the responses of $x_i(t)$ and $\hat{x}_i(t)$ on the same axis and the response of $u(t)$ with different K_1 and K_2 which are shown respectively on Fig.7 and Fig.8. We found that the estimated performance of K_1 observer is better than the K_2 observer. From these two figures, it is clear that the estimated error for K_1 observer is quite small and for the other observer, it catches the real state after almost 2 seconds which is slow in this situation. Therefore during the design of the control system we should choose a faster poles of observer than the control to get a better observation.

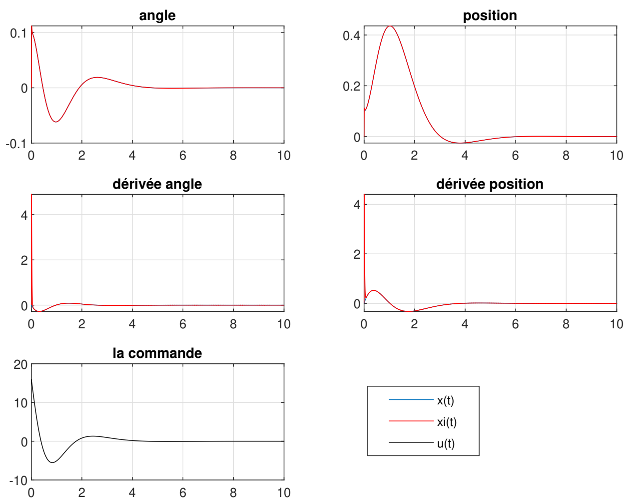


Figure 7: The plot of $x(t)$ and $u(t)$ under state feedback with K_1 observer

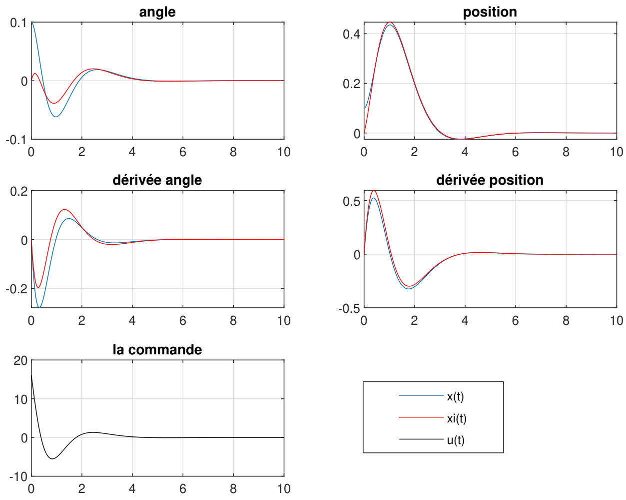


Figure 8: The plot of $x(t)$ and $u(t)$ under state feedback with K_2 observer

4 Estimated State Feedback Controller

In this section, we consider that only the angle θ and the position of the cart p are available for the control. The new output equation becomes (18).

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ p \\ \dot{p} \end{bmatrix} \tag{18}$$

We construct the New Simulink block diagram which take into account the fact that the output of the system is now just θ and p and the state feedback connected to the observer, shown in Fig.10.

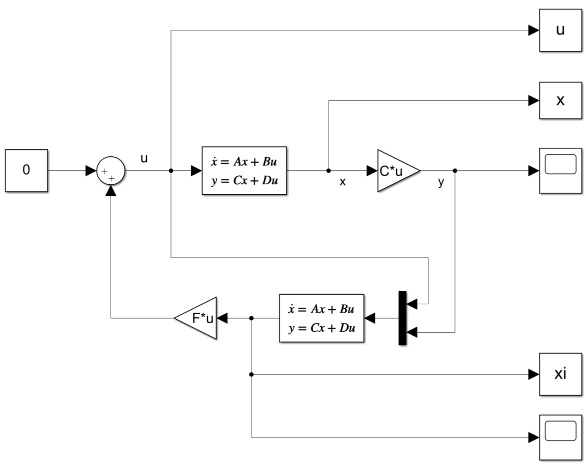


Figure 9: The Simulink block diagram of estimated state feedback controller

The configuration of the observer are shown in Fig.10. In the configuration, the input of the block is $[u, y]^T$, its dimension is 3×1, and the dimension of the state vector is 4×1, so the dimension of D is 4×3.

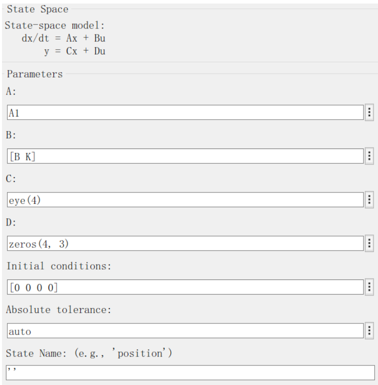


Figure 10: The configuration of the observer
b

In the diagram, we use $C = eye(4)$ first and than multiple it by new C in order to output x to the workspace. Therefore, we can compare $x_i(t)$ and $\hat{x}_i(t)$ in the plot.
(1) We choose the previous state feedback with desired eigenvalues of $A + BF = -100 \pm j$ and $-200 \pm 2j$. The comparision of $x_i(t)$ and $\hat{x}_i(t)$ is shown in Fig.11.

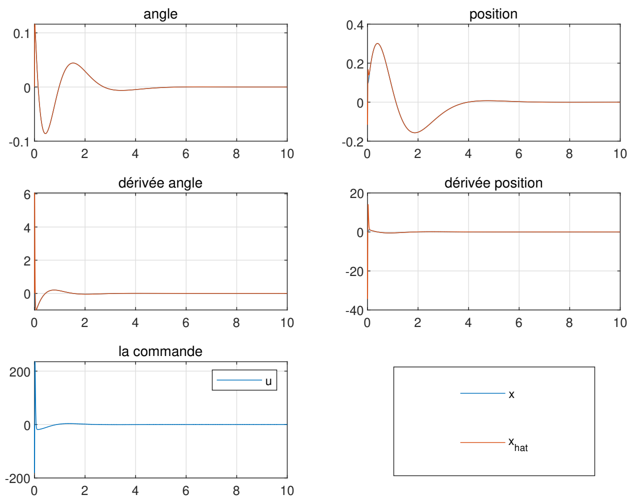


Figure 11: The comparision of $x_i(t)$ and $\hat{x}_i(t)$ with K_1 observer

(2) We choose the previous state feedback with desired eigenvalues of $A + BF = -1 \pm j$ and $-2 \pm 2j$. The comparision of $x_i(t)$ and $\hat{x}_i(t)$ is shown in Fig.12.

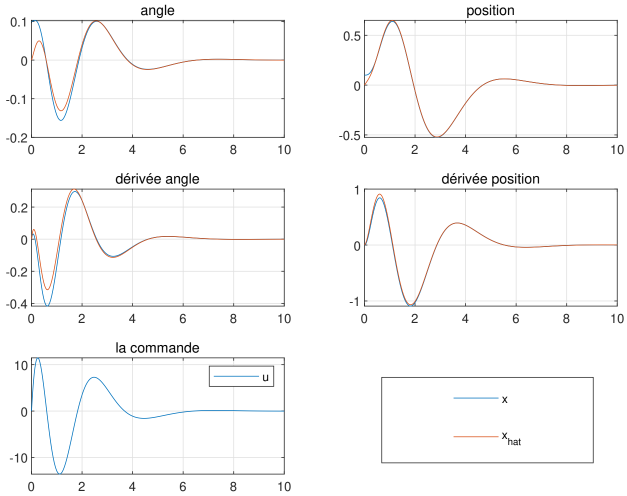


Figure 12: The comparision of $x_i(t)$ and $\hat{x}_i(t)$ with K_2 observer

Comparing two plots, we found that if the poles have very large negative part, which is fast poles, the rise time of the step response is very small. Besides, the state estimation is more accurate than "poles as fasts as" the ones used for the control. However, according to the plot, the initial value of control signal is very high in the fast pole case. It is related to the derivative action and should be reduced by adding a low-pass filter. Because zeros in the system exist a positive real part in the system, undershoot appears in the step response. It is a non minimum phase system. Because there exist a negative real part in the system, overshoot appears in the step response.

Comparing to the previous section, due to we cannot have access to the actual state, we instead use estimated state in the feedback. We have zero initial conditions in the feedback block. In the plot, we can see that the control signal in the estimated state feedback has fluctuation, especially in the "faster poles" than the ones used for the control. But the responses become stable in the end. We also notice that the error between estimated state vector $\hat{x}$ and the real state vector x with K_2 observer becomes lower in this situation which means the $\hat{x}$ catches the x more quickly. This might because the control u contains the estimated state vector $\hat{x}$ in estimated state feedback and the state will approach the previous estimated state.

5 Animation

In this part we use our personal observer and state feedback as shown in Fig.13 to fill the block $u = Fxhat + v$ in file *"lab1_dybac_L.slx"* to check the performance which is shown in Fig.14. It can show that our observer and controller well correct the initial state to 0.

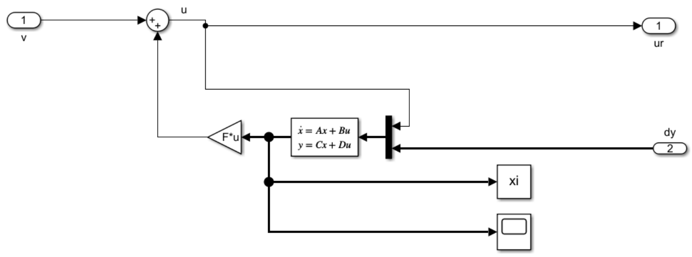
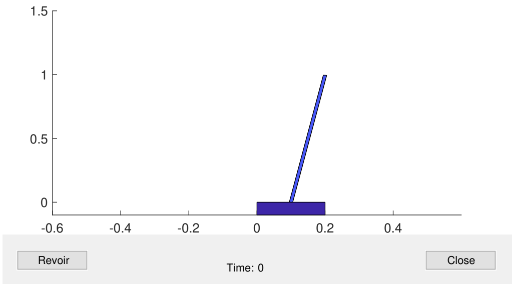
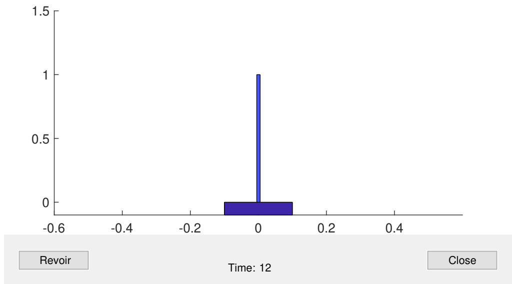


Figure 13: The block $u = Fxhat + v$



(a) The start state



(b) The final state

Figure 14: The start state and the final state of the animation

6 Conclusion

In this lab, we first construct the model of the inverted pendulum system. After apply the numerical values, we calculate the eigenvalues associated to the linear state space representation. Because the system is controllable, we are able to use a static state feed back to stabilize the pendulum and the cart around an equilibrium point. By creating Simulink blocks and send the variables to the workspace, we plot the state and the control. After that, we check if the system is observable. We use two different pole placements to compute K, in order to construct the observer. Because in some cases, we cannot have the access to the actual state, we need to estimate the state from the input and output, which requires the system to be observable. We analyze different plots in the previous sections. In the end, we use our observer and state feedback to create an animation. By these steps, we are able to control the pendulum on a cart.