

UNIVERSITY OF CALIFORNIA, SANTA BARBARA
Department of Electrical & Computer Engineering
ECE 147b: Digital Control

Lab 3: Balancing the Inverted Pendulum

Overview

In this lab you will attempt to balance the inverted pendulum. The pendulum is a classic example in differential equations in general, and control systems in specific. This is due to the fact that it has two equilibria, one stable and one unstable. The stable equilibrium—when the pendulum hangs down—is not as interesting as the unstable one, but it is a lot easier to use for debugging your implementation. The ultimate objective is to get a balanced upright pendulum and this will require a good state-feedback/estimator based design.

Modeling

A schematic of the inverted pendulum is shown in Figure 1. You are already familiar with the motor cart. The pendulum is attached to the pivot on top of the cart and the angle measurements are obtained via a quadrature encoder, similar to the one used to obtain cart position measurements.

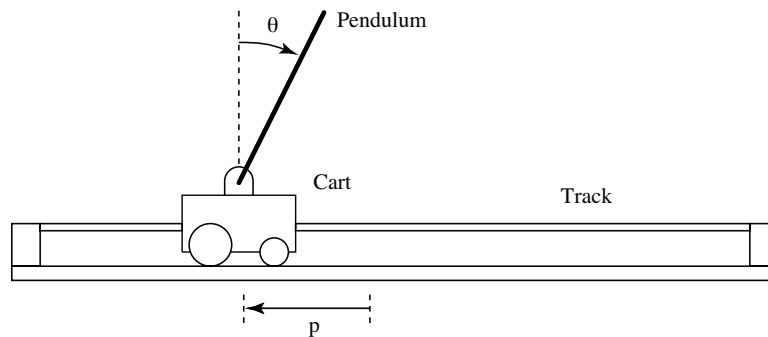


Figure 1: Inverted Pendulum

A free body diagram of the system is shown in Figure 2, and defines the forces and moments considered below. Applying Newton's second law to the motor cart yields

$$m_c \ddot{p} = F - N . \tag{1}$$

Applying Newton's second law to the pendulum in both the horizontal and vertical directions

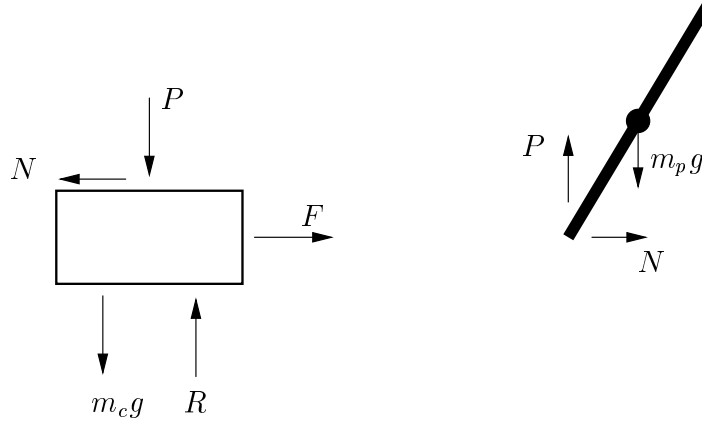


Figure 2: Free Body Diagram for Cart/Pendulum System

we obtain

$$\begin{aligned} N &= m_p \frac{d^2}{dt^2} (p + l \sin(\theta)) \\ &= m_p (\ddot{p} + l \cos(\theta) \ddot{\theta} - l \sin(\theta) (\dot{\theta})^2) \end{aligned} \quad (2)$$

$$\begin{aligned} P - m_p g &= m_p \frac{d^2}{dt^2} (l \cos(\theta)) \\ &= m_p g + m_p l (-\sin(\theta) \ddot{\theta} - \cos(\theta) (\dot{\theta})^2) . \end{aligned} \quad (3)$$

Balancing the moments about the pendulum's center of mass gives

$$I \ddot{\theta} = Pl \sin(\theta) - Nl \cos(\theta) . \quad (4)$$

By substituting Equation 2 into Equation 1 we obtain

$$F = (m_c + m_p) \ddot{p} + m_p l \cos(\theta) \ddot{\theta} - m_p l \sin(\theta) (\dot{\theta})^2 . \quad (5)$$

Substituting Equations 2 and 3 into Equation 4 yields

$$\ddot{\theta} (I + m_p l^2) = m_p g l \sin(\theta) - m_p l \ddot{p} \cos(\theta) . \quad (6)$$

For simplicity, define the constants

$$\begin{aligned} M &= m_c + m_p \\ L &= \frac{I + m_p l^2}{m_p l} . \end{aligned}$$

Recall that the relationship between force and voltage for the motor cart is

$$F = \frac{K_m K_g}{Rr} V - \frac{K_m^2 K_g^2}{Rr^2} \dot{p} .$$

Eliminating $\ddot{\theta}$ from Equation 5, \ddot{p} from Equation 6, and substituting M, L , and the equation for force yields

$$\ddot{p} \left(M - \frac{m_p l \cos^2(\theta)}{L} \right) = \frac{K_m K_g}{Rr} V - \frac{K_m^2 K_g^2}{Rr^2} \dot{p} - \frac{m_p l g}{L} \cos(\theta) \sin(\theta) + m_p l \sin(\theta) (\dot{\theta})^2 \quad (7)$$

$$\ddot{\theta} \left(L - \frac{m_p l \cos^2(\theta)}{M} \right) = g \sin(\theta) - \frac{m_p l (\dot{\theta})^2}{M} \cos(\theta) \sin(\theta) - \frac{\cos(\theta)}{M} \left(\frac{K_m K_g}{Rr} V - \frac{K_m^2 K_g^2}{Rr^2} \dot{p} \right). \quad (8)$$

Let us define the state-vector as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} p \\ \dot{p} \\ \theta \\ \dot{\theta} \end{bmatrix}.$$

We can linearize the system about the equilibrium $[0 \ 0 \ 0 \ 0]^T$. Note that $\theta = 0$ corresponds to the pendulum being upright. You should check that this is indeed an equilibrium for the cart/pendulum system. The linearization is

$$\begin{bmatrix} \delta \dot{x}_1 \\ \delta \dot{x}_2 \\ \delta \dot{x}_3 \\ \delta \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{M - \frac{m_p l}{L}} \frac{K_m^2 K_g^2}{Rr^2} & -\frac{g m_p l}{L \left(M - \frac{m_p l}{L} \right)} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{1}{M \left(L - \frac{m_p l}{M} \right)} \frac{K_m^2 K_g^2}{Rr^2} & \frac{g}{L - \frac{m_p l}{M}} & 0 \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \delta x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M - \frac{m_p l}{L}} \frac{K_m K_g}{Rr} \\ 0 \\ -\frac{1}{M \left(L - \frac{m_p l}{M} \right)} \frac{K_m K_g}{Rr} \end{bmatrix} V$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \delta x_4 \end{bmatrix}.$$

Substituting values from Table 1 yields the linearized system

$$\begin{bmatrix} \delta \dot{x}_1 \\ \delta \dot{x}_2 \\ \delta \dot{x}_3 \\ \delta \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -15.14 & -3.04 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 37.23 & 31.61 & 0 \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \delta x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 3.39 \\ 0 \\ -8.33 \end{bmatrix} V \quad (9)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \delta x_3 \\ \delta x_4 \end{bmatrix}. \quad (10)$$

Having obtained a linear model, we are now ready to design a controller to balance the inverted pendulum.

Design Example

For the following design, a plant different from Equation 9 is considered. You will need to replicate what follows for the inverted pendulum system. Note that there may be some specific implementation issues not dealt with below that you will need to take care of. For the following design, we will assume our plant is given by

$$\dot{x} = Ax + Bu \tag{11}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -40 & -20 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 25 & 17 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 7 \\ 0 \\ -12 \end{bmatrix} V$$

$$y = Cx \tag{12}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}.$$

If we define the feedback input to be $u = -Kx$, then Equation 11 becomes

$$\dot{x} = Ax + B(-Kx) = (A - BK)x.$$

By examining the above equation we can see that it is possible to arbitrarily place the system poles by selecting the feedback gain matrix K . (This is subject to some technical conditions which will not be addressed here. For a full discussion, look up the subject of *controllability* in any linear systems text.) This is referred to as full-state feedback control, since we assume all the states are available for measurement. Note that this works for discrete-time systems as well (simply replace \dot{x} by $x(k+1)$).

The zero-order hold equivalent, using a sample period of $T_s = 0.01$ seconds, for Equations 11 and 12 has poles at $z = 0.6682, 0.9806, 1, \text{ and } 1.0230$. This system is obviously unstable (since it has a pole located outside the unit circle). Using the Matlab command **place**, we obtain a feedback gain matrix,

$$K = [6.62 \quad -0.35 \quad -3.53 \quad -0.87]$$

which places the closed-loop system poles at $z = 0.67, 0.96, \text{ and } 0.98 \pm j0.01$. You should check this.

Observer Design

The above approach has an implementation problem; we do not have all of the states available for measurement. A simple, and somewhat crude, fix would be to notice that the states we don't have available (x_2 and x_4) are derivatives of the states we do have available (x_1 and x_3). Therefore, we could use numerical differentiation to obtain an estimate of the unmeasured states.

For example, denote the cart velocity by $v(k)$. We can get a crude estimate of $v(k)$ via,

$$v(k) \approx \frac{p(k) - p(k-1)}{T}, \quad (13)$$

where $p(k)$ is the measured cart position and T is the sample period. This estimate is susceptible to noise.

However, we have quite a bit of information about the system which may allow us to do something more intelligent. We know the system, the system input, and the system output.

We can construct a copy of the system given in Equations 11 and 12, driven by an extra term:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad (14)$$

$$\hat{y} = C\hat{x} \quad (15)$$

where L is the “observer gain matrix”. We will refer to the system described by Equations 14 and 15 as an observer or estimator. The term $L(y - \hat{y})$ forces the states of the estimator to converge to the states of the real system. This can be seen as follows: consider the error between the real states and the estimated states as

$$\tilde{x} = x - \hat{x}.$$

Differentiating the above equation, we obtain an expression for the *error* system (i.e., the system which describes how the difference between the states and their estimates evolves):

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} = (A - LC)\tilde{x}.$$

You should write this out to convince yourself the above equation is correct. From the above equation, we can see that \tilde{x} will go to zero as time goes to infinity if $A - LC$ is a stable matrix (i.e., all the poles are in the right half plane for a continuous-time system, or all the poles are in the unit circle for a discrete-time system). This appears similar to the problem of state feedback, in that we can now arbitrarily place the poles of the *error* system (again under some technical conditions on *observability*). Note that fast poles for the error system will cause the difference between the actual states and the estimates to decay quickly. However, fast poles also generally lead to high gain, so there is a trade-off which needs to be made. For the system described in Equations 11 and 12, we will place the poles of the error system at $z = 0.5$, 0.8 , and $0.85 \pm j0.05$ by using the observer gain matrix

$$L = \begin{bmatrix} 0.3246 & -0.0665 \\ -1.9277 & 0.8108 \\ -0.0838 & 0.3472 \\ 0.6868 & 2.3669 \end{bmatrix}.$$

This matrix is calculated using the Matlab command **place** with A^T and C^T . Take the time to understand why this works.

Having constructed the estimator of Equations 14 and 15, we can use the estimated states for our state feedback control (i.e., let $u = -K\hat{x}$). (For those of you concerned about such things, the interconnection of the observer and state feedback controller is indeed stable. A full discussion of the topics of controllability, observability, and interconnections is undertaken in ECE 230AB.) A diagram of a system with a controller and observer is shown in Figure 3.

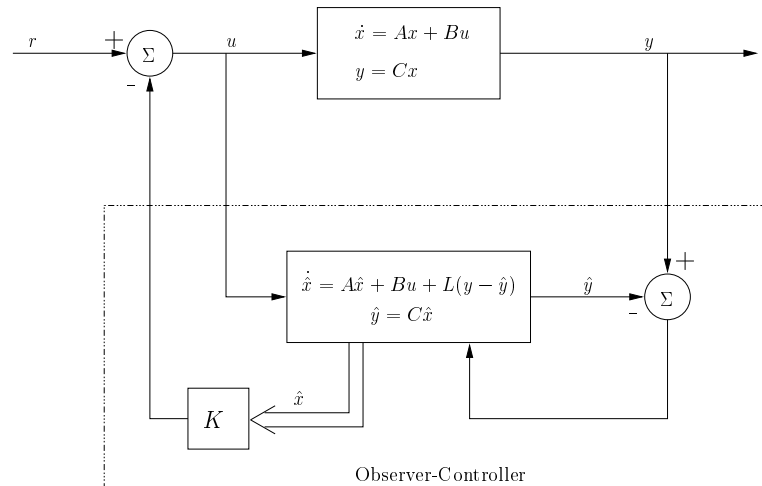


Figure 3: Observer-Controller Configuration

Figure 4 shows both the measured states (x_1 and x_2) and their estimates (\hat{x}_1 and \hat{x}_2). The estimated states all have zero initial conditions (this is standard practice). The actual system was started with an initial position of 5 centimeters and an angle of 5 degrees. This is a reasonable error to expect as it is difficult to start the pendulum any closer to vertical than 5 degrees. Notice that the estimates quickly converge to the actual states.

Lab Schedule: Week 1

The first week will focus on trying to get a simple state-feedback controller to work.

Implementing a state-feedback controller requires measurements of all four states (cart position, cart velocity, pendulum angle and pendulum angular velocity). Use a simple differencing scheme (as in Equation 13) to estimate the unmeasured values.

It's much easier to debug the system if the plant is stable. So we will first try to design a controller for driving the pendulum to the down position.

When you have that working you can start looking at trying to control to the pendulum-up position. Note that these two control problems will require different plant models, as the linearizations are taken at different operating points.

Prelab: week 1

You will need to write MATLAB scripts to do the design, analysis and simulations. Attach copies of these scripts to your prelab.

1. Create a linearization of the pendulum system with the pendulum hanging down as the operating point.
2. Design a discrete-time state-feedback controller for the pendulum-down case. Check the position of the closed-loop poles. Use $T_s = 0.0005$ seconds as your sampling rate.

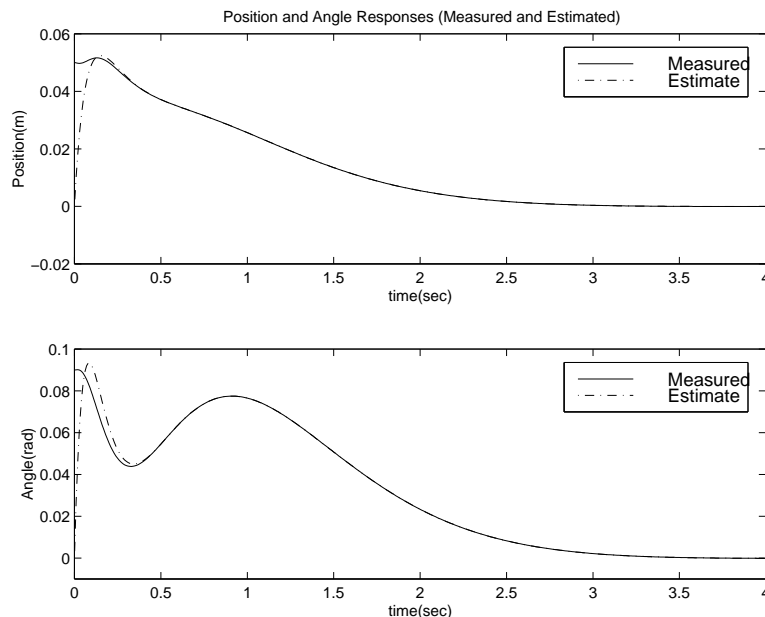


Figure 4: Response for Sample System Design

3. Simulate your design for the pendulum-down case. Make sure that the actuator voltage does not saturate if you start the controller when the cart is at either end of the track, or when the pendulum is up to 45 degrees off of vertical. If your simulation shows significant saturation, repeat step two.
4. Also do a simulation with the cart position at zero and the pendulum 45 degrees off of vertical with the controller turned *off*. This is an open-loop response that you can use for comparison with your closed-loop system.
5. Repeat the steps 1 through 3 above for the pendulum-up case. Test it with the cart about 30 cm from center (and the pendulum vertical) and the pendulum 5 degrees of off vertical (and the cart at zero).

Lab: week 1

1. Build (in SIMULINK) differencing systems to estimate the two states that are not measured.
2. Implement and test your pendulum-down controller. You should test it by starting it from a non-zero position on the track, and from an off-vertical pendulum position. Take data that allows you compare your design to your prelab simulations (including the open-loop case in step 4 of the prelab).
3. Repeat step 2 for the pendulum-up case. You will need to start the pendulum very close to vertical to get a working controller. If it works, take enough data to compare this response with your simulations.

Note: When you start the controller, WinCon takes the current encoder readings as zero. That is, the position of the cart and the angle of the pendulum, *when you click the start button*, are defined as zero. Therefore, you want to hold the pendulum as near vertical as possible when starting the controller.

Lab Schedule: Week 2

We will now try to get an estimator running for the system. From the two measurements (cart position and pendulum angle) we will create an estimator for the entire state. Again, debugging is much easier if the system is stable. Design and implement an estimator for the pendulum-down case. You can then compare the measured position and angle against the estimated values and use this information to fine tune your estimator gains. The combined estimator-controller can be used to control the pendulum in the upright position.

Prelab: week 2

1. Design an estimator for pendulum-down case. In simulation you should plot the estimated states and the plant states. Use a small amount of noise on your measurements (say 5 millivolts) in the simulation. In designing the estimator, make sure that not much noise appears on the estimated states. If they are too noisy, your estimator gain (L) is too big. Modify it by backing off on the pole positions. Test the estimator with the state-feedback controller off (i.e. in the open-loop case).
2. Simulate your estimator with the state feedback controller used for the pendulum-down case. If you had to modify your state feedback gains in the previous lab, use the modified gains. You may need to adjust the estimator and/or state-feedback gains to get a nice combined response.
3. Repeat the estimator design for the pendulum-up case. Here you should chose estimator pole positions that are about 2 to 4 times faster than your closed-loop pole positions for the pendulum-up state-feedback controller.

Lab: week 2

1. Implement your estimator in the pendulum-down case, with the controller *off*. Record plots of the estimated states and the two measured states. Try to use experimental situations which match those in step 1 of your prelab simulations.
2. Now connect your pendulum-down estimator to your pendulum-down controller. Again take data for the experiments which give a close match to your step 2 prelab simulations.
3. Try connecting your pendulum-up estimator to your pendulum-up state feedback controller. This might require some adjustment of gains to get right. Debugging the estimator is difficult because the system may be going unstable and you only get a record of the first second or two. Try turning the controller off and seeing if the

estimator tracks the states as the pendulum falls. This will only be good until the angle reaches 45 degrees. Beyond this point the linearization is not even close to valid.

4. If you get it balanced, try tapping it lightly and recording the data. Try also starting the pendulum slightly off-vertical. In your lab report you should try and simulate these two experiments and compare the data.

Lab Schedule: Week 3

The objective here is to design a reduced order estimator to estimate only the two velocities. The two measured states will be used directly in the state-feedback design.

Prelab: week 3

1. Design a reduced order estimator for the pendulum-up case.
2. Analyze the closed-loop pendulum-up control system with the reduced-order estimator. Use unfiltered position and angle measurements for those parts of the state vector in calculating the state-feedback. Include a small amount of noise in the simulations.

Lab: week 3

1. Implement your reduced-order pendulum-up estimator and your pendulum-up state feedback controller.
2. Run a similar set of closed-loop experiments to those performed for the full order pendulum-up estimator/controller in week 2. The objective is to collect data to compare these two control schemes in your lab report.

More Challenging Problems

Once you have balanced the inverted pendulum, you may wish to try the following experiments:

1. Try to find the minimum sampling time needed to balance the system. Assuming you created an m-file to obtain your observer and feedback gain matrices, it shouldn't be difficult to calculate values for different sampling periods.
2. Try closing a feedback loop around the cart position so that the cart will track a reference signal. You must simulate this before attempting it.

System Parameters

These parameters should be checked for your particular system. They may differ slightly.

Parameter	Symbol	Value	Units
Motor Torque Constant	K_m	0.00767	$\frac{Nm}{Amp}$
Gear-box Ratio	K_g	3.7	N/A
Motor Armature Resistance	R	2.6	Ω
Motor Pinion Radius	r	0.00635	m
Cart Mass	m_c	0.455	kg
Pendulum Mass	m_p	0.210	kg
Rotational Inertia	I	0.00651	kg m ²
Half-Length of Pendulum	l	0.305	m

Table 1: System Parameters for Cart/Pendulum System