

University of Utah
Electrical & Computer Engineering Department
ECE 3510 Lab 9
Inverted Pendulum

M. Bodson, A. Stolp, 4/2/13
rev, 4/9/13, 4/10/19

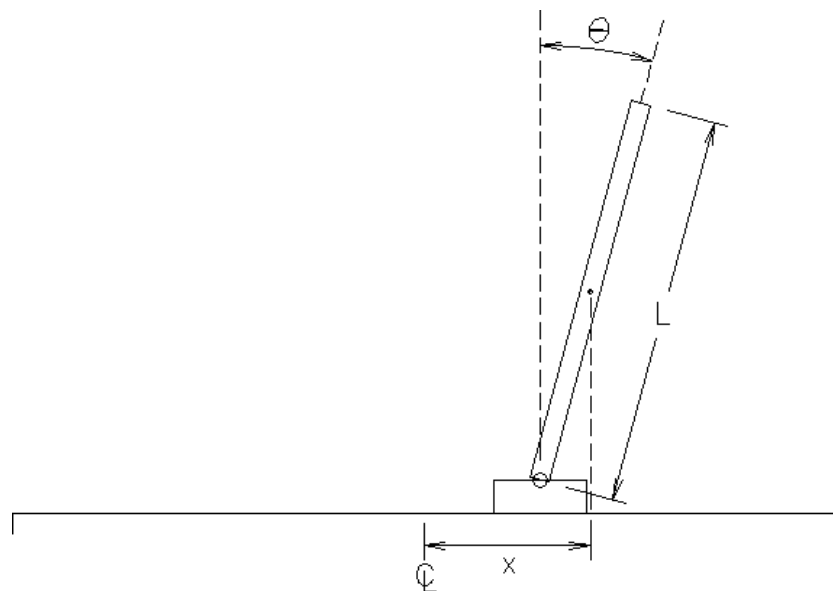
Objectives

The objective of this lab is to experiment with the stabilization of an unstable system, the inverted pendulum. First you'll play with an animation in Matlab to get a feel for the challenges of control. If you have a yardstick handy, you can try the real thing in two dimensions. (If you can stabilize something shorter than a yardstick, you've got faster reflexes than I do.) After you experience the difficulties of manual control, you'll design a linear controller using root-locus techniques, and make the controller move the inverted pendulum to different positions on the track.

Introduction

The inverted pendulum system is a favorite experiment in control system labs. Stabilization of this highly unstable "plant" can be an impressive demonstration of the capabilities of feedback systems. The inverted pendulum is considered a simplified representation of the initial flight of a rocket, when the only force is at the bottom. Once the rocket is moving fast enough its fins can stabilize the flight with no need for a feedback system until it is out of the atmosphere.

A diagram of an inverted pendulum system is shown at right. The base block can move left or right along the horizontal surface to some set limits. It will be controlled either manually by you or automatically by your controller. A vertical column (the inverted pendulum) is attached to the base by a hinge joint. The hinge only allows it to swing left and right, so this is just a one-dimensional problem. θ is the angle of the pendulum and should be kept close to zero to keep it from falling over. x is the position of the center of the pendulum.



The inverted pendulum can challenge you in several ways. The first challenge is to simply stabilize the pendulum, especially in the presence of disturbances. The second challenge is to also keep it at some position "x", and the final challenge is to move it from one x position to another. Base motion is limited to the width of the horizontal surface.

If your task requires a base motion beyond the limit, it may be impossible to achieve the task or even to recover stability.

The first step to analyzing this system is to understand the forces on the pendulum affect the angle, θ . Refer to the free-body diagram of the pendulum at right. The vertical forces are the weight of the pendulum (mg) acting at the center-of-gravity (Cg), and the balancing upward force at the base. As long as the angle, θ is small, the vertical acceleration of the pendulum will also be small and these two forces can both be considered mg . (m is the mass of the pendulum and g is the acceleration of gravity.) The distance between the hinge pin and the center-of-gravity is $L/2$, so the torque due to gravity is

$$T_g = mg \frac{L}{2} \sin(\theta)$$

The horizontal forces are a little more complex. A purely horizontal (x -direction) acceleration, a , of the pendulum will be “resisted” by its mass with a force of “ ma ” acting at its center-of-gravity. This “ ma ” also creates a moment about the hinge pin. The force at the base has to be both the balance of the moment couple (ma) as well as provide the force for the horizontal acceleration, “ ma ”, ie it’s $2ma$. The torque, however is simple

$$T_a = -ma \frac{L}{2} \cos(\theta)$$

The moment of inertia of a slender, uniform rod of length L , about one end is

$$J = \frac{1}{3} mL^2$$

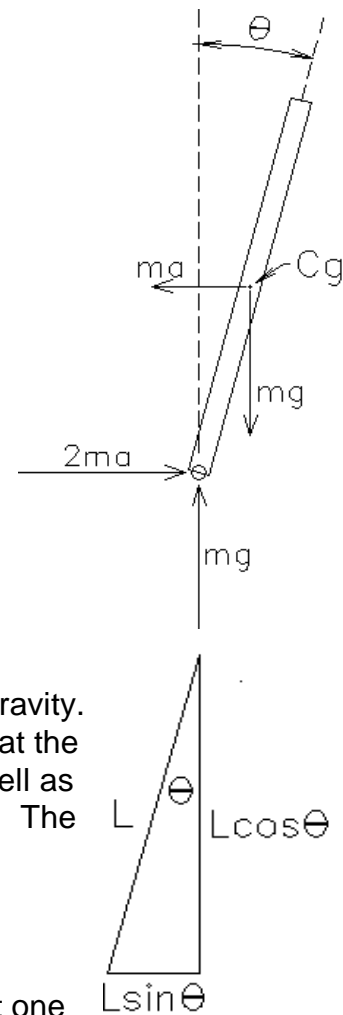
Angular acceleration, α , is related to torque in the same way that linear acceleration is related to force,

$$\sum T = J\alpha$$

Using the 4 equations above and small-angle approximations (needed for linearization), Derive the following transfer function of the pendulum in your notebook:

$$\frac{\theta(s)}{X(s)} = \frac{-s^2}{\frac{2}{3}Ls^2 - g}$$

This system has two zeros at the origin and poles at $s = \pm \sqrt{3g/2L}$. Obviously, the pole in the right-half plane is the source of this systems instability and the instability worsens when the pendulum is short (as one might expect). Another tricky problem is that the system has two zeros at $s = 0$. This is due to the fact that an x acceleration is required to impact the pendulum angle. On the other hand, this property makes stabilization possible for arbitrary base positions.



A more complex model would assume that the control variable is the force applied to the base, rather than the x position. This assumption is more realistic in many cases, but makes manual control very difficult. Instead, the simulation program of this lab will represent the dynamics of the base control system as a simple first-order system.

$$\frac{X(s)}{X_{\text{com}}(s)} = \frac{f}{s + f}$$

where x_{com} is the commanded base position, x is the base position, and $f > 0$. The overall transfer function of the system is then

$$\frac{\theta(s)}{X_{\text{com}}(s)} = \frac{f}{(s + f)} \frac{-s^2}{(\frac{2}{3}Ls^2 - g)}$$

or

$$P(s) = \frac{\theta(s)}{X_{\text{com}}(s)} = \frac{-\frac{3f}{2L}s^2}{(s + f)(s + \sqrt{\frac{3g}{2L}})(s - \sqrt{\frac{3g}{2L}})} = \frac{bs^2}{(s + f)(s + a)(s - a)}$$

$$\text{where, } a = \sqrt{\frac{3g}{2L}} \text{ and } b = -\frac{3f}{2L}$$

Manual control

Download `invpend_xxx.zip` from the class web site for the required files. Create (if you need to) a Matlab workspace folder and unzip the files into your work space. The simulation program resides in `invpend.m`. The program will ask you for the length (L) of the pendulum and then limit x to $\pm 2L$. The pendulum angle is prevented from exceeding $\pm 30^\circ$. The value of f is 5, so that the base responds relatively quickly to commands. However, even with such rapid position control of the base the system is difficult to control with the mouse for short length pendulums. Make the pendulum longer for easier control. The program will also ask allow you to choose automatic control, but it won't work yet because you haven't modified the `invpendcinit.m` file.

Choose a long pendulum and have fun trying to keep the it balanced (straight up). Now try to keep it straight up in one location (I find it easiest if I keep my eyes on the middle of the pendulum). Finally, try to move the base from the left line to the right line, while keeping the pendulum balanced (I can't). Describe in words the challenges of the control problem and your strategies for control. Try a shorter pendulum to see if it is harder or easier to control.

Automatic control

Stabilizing controller

First, make a root-locus plot of the transfer function ($P(s)$) above as though it is in a unity-feedback loop with no controller ($C(s)$). Use $g = 9.81$, $L = 2$, and $f = 5$. Is there any way to stabilize this system without a controller (additional poles and/or zeros)? Realizing that you cannot effectively cancel a pole with a zero in the right-half plane (RHP), is there any

way to stabilize this system without that controller having at least one pole in the RHP? Justify your answer with an explanation in your notebook.

You should have reached a very interesting conclusion in the previous paragraph. Although you have seen many root-locus plots in class where an inherently unstable system can be stabilized simply by placing it in a feedback loop and adjusting the gain, this may be the first time you have encountered a unstable system which can only be stabilized by a controller which is also unstable. Think of it this way. If you wished to stabilize the pendulum at any angle other than 0° , the base would have to *endlessly accelerate* in the direction of the angle. If that ain't unstable, what is? Alternatively, if you wanted to move the pendulum from one x-position to another, you have to start by moving the base in the *opposite* direction. Now you know why it was called "Rocket Science" and why so many early rockets had such short flights. By-the-way, bottle rockets avoid this problem by placing the trust above the center of gravity and model rockets avoid it by using a guide rod for the first few feet, after which the rocket is moving fast enough that the aerodynamic effects of the fins keep it stable. One of the reasons that learning to ride a bike can be difficult for children is that they are dealing with a very similar problem. In order to turn *left*, they must first steer *right!* You do too, you just don't realize it anymore.

Enter the plant transfer function ($P(s)$) in Matlab and fire-up the SISOtool. Verify your root-locus plot.

Use the following compensator to cancel out the plant poles in the left-half plane, double-up the pole in the right-half plane, and add a new pole in the left-half plane at $-c$. Let $c = 1$ for the time being.

$$C(s) = \frac{X_{com}(s)}{-\theta(s)} = k \frac{(s+a)(s+f)}{(s-a)(s+c)}$$

Document in your notebook

You will design this controller to bring the pendulum angle to zero at the zero track location. Note that, with the cancellations, the closed-loop system has three poles. The controller has one pole location ($-c$) to be adjusted. Document the following work in your notebook. First play with the value of c . Plot one case where $c > a/2$, showing that it cannot be stabilized. Notice that if you move the pole ($-c$) towards the origin, the root locus begins to curve into the left-half plane. The correct gain (k) could then place all the closed-loop poles in the left-half plane. Decide on value of c that allows you to place the closed-loop poles in desirable positions. Find the gain necessary gain factor, k .

Modify the `invpendcinit.m` file to implement your controller. Look at the comments and statements near the top of the file. They will tell you where to enter or calculate the values of a , c , and k . You may have to replace statements that already define these variables. Specify the numerator and denominator of $C(s)$ in `numc` and `denc` (yes you will have to determine to coefficients, do so in terms of a , c , f , etc. and enter them that way). Save the file and try the automatic control for a 2 meter pendulum. Most likely nothing will happen. Edit the `invpend.m` file (around line 20) and find the line that has been "commented out" which will set the initial angle to 5° . Make the line active. Run the simulation again and watch to see if it recovers.

Now back to the math for a minute. Show in your notebook that it is possible to choose the parameters c and k so that all three closed-loop poles are placed at $s = -a/2$. (Hint: assume the denominator of closed-loop transfer function is $(s + a/2)^3$ and back-calculate what c and k need to be in terms of other known parameters like g , a , f , etc..) Compute the parameters required for the system and plot the root-locus in SISOtool. What will happen if the gain is increased or decreased significantly? Determine the minimum gain for the system to be BIBO stable.

Modify the `invpendcinit.m` file to implement your controller, only now make sure that you enter a , c , and k in terms of other known parameters like g , a , f , etc.. This will allow the automatic control to work for different of pendulum lengths. Save the file and try the automatic control for a different size pendulums.

For a 2-meter pendulum, plot the responses of the pendulum angle and of the base position resulting from the initial 5° error. Check that the pendulum angle returns to zero within the expected time period and with the expected dynamics, given the pole locations.

print
figures &
include
in
notebook

Tracking controller

Modify the controller so that an arbitrary set-point of the base position can be imposed. Replace $X_{\text{com}}(s) = -C(s)\theta(s)$ by $X_{\text{com}}(s) = X_{\text{ref}}(s) - C(s)\theta(s)$. You will find the appropriate lines and commented lines in `invpendc.m`

Note that this is an unusual configuration for a feedback system, because the reference command is applied to the plant input rather than the controller input. Interestingly, the plant output cannot be set to an arbitrary position in this problem, but the plant input can. Generate steps of position reference alternating between $+L/2$ and $-L/2$ every 20 seconds (again, look for the appropriate lines and commented lines in `invpendc.m`). Save the `invpendc.m` file. Comment out the initial 5° angle in the `invpend.m` file and run the automatic control for a 2-meter pendulum. Let it run for a few cycles to accumulate enough data. Plot the responses of x , x_{com} , and θ for 60 seconds (you may find `invpendplot.m` handy). You should observe an overshoot in the response of the base. You should also observe that x_{com} initially moves (after a very small blip) in the direction opposite to the step applied. Explain why it has to do that.

Repeat your plots for a 6-meter pendulum.

Checkoff and conclusions

Show your code and demonstrate the real-time operation to the TA (any pendulum length, your choice).

Your conclusion should include some description of challenges and strategies for manual control and a qualitative description of how the automatic controller achieves this. Compare the automatic control to the manual. Include other observations and comments.