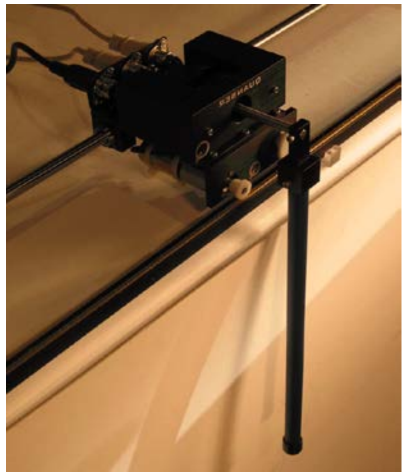
ECE:3600 CONTROL SYSTEMS – Spring 2018
The University of Iowa College of Engineering

Experiment #3 – Hanging Crane
Room 2245 SC



This experiment models several real control applications, including overhead cranes, and pick-and-place robot operations.

Objectives:

a) To design, simulate, evaluate and tune a controller that can move a cart with attached hanging pendulum to a prescribed position while minimizing pendulum swing.
b) To compare real system responses to those predicted by a linear model simulations.

Warnings:

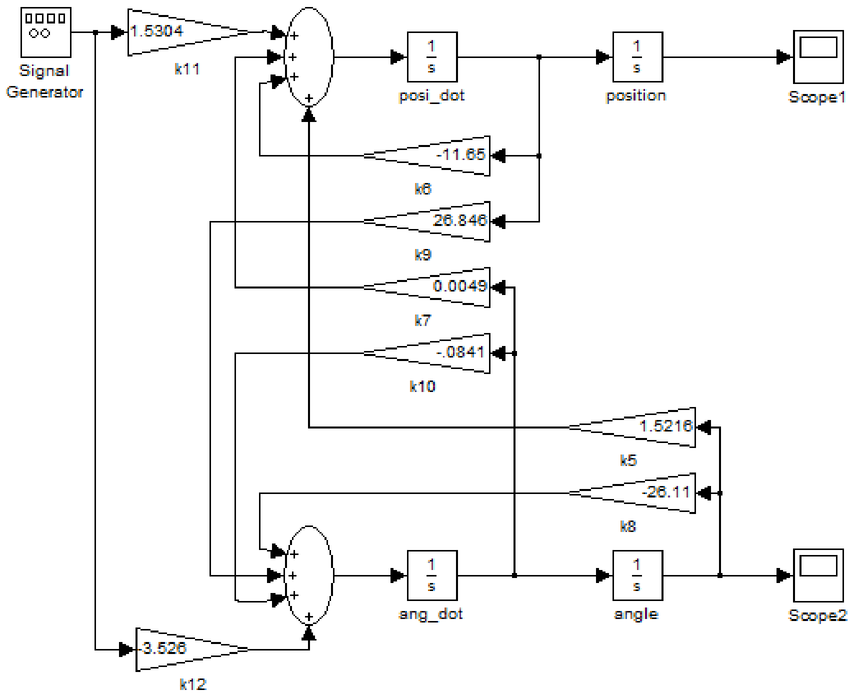
a) Do not alter any electrical connections. Lethal voltages are present.
b) Keep all foreign objects, including fingers, heads, clothes, hair, and pencils, away from the cart and pendulum. The motor generates torque sufficient to crush objects caught in its gears and move the cart and pendulum at speeds that could cause substantial injury.
c) Do not allow any commands that would send the cart off the end of the track to be issued.
d) Turn off the Universal Power Module at the end of your lab.

Your lab report (**each student submits their own)** should include:

a) The time and date of your lab and the names of your lab partners
b) A brief executive summary of the lab's objectives, procedures, and outcomes.
c) Your pre-lab analysis including: (1) verifications of the system's open-loop transfer functions, (2) the root locus plot used to complete your design, and (3) your design's *simulated* position and angle outputs for square wave and step inputs.
d) SIMULINK plots of your designs' *actual* square wave responses for your k values.
e) A brief evaluation of your simulated and experimental results, identifying possible sources of discrepancies, and assessing your design's overall performance. More specifically, your report should include: (1) an assessment of the accuracy of your simulation's position response: rise time, settling time and percent overshoot predictions (compared to observed values – each transition induces a step), (2) an assessment of the accuracy of your simulation's angle response predictions (compared to the observed responses) and (3) your conclusions: Did your original design meet its design goals when implemented? What role did theory play in development of your controller?

PART (A): Pre-lab: Design and MATLAB simulations:

A linearized open-loop model of the hanging pendulum system is shown below. Click here to download a copy.



Verify:

(1) That the transfer function from the input to the cart position $p(t)$ is

$$G_p = \frac{1.530s^2 + 0.1114s + 34.59}{s^4 + 11.73s^3 + 26.96s^2 + 263.3s}$$

(2) That the transfer function from the input to the pendulum angle $a(t)$ is

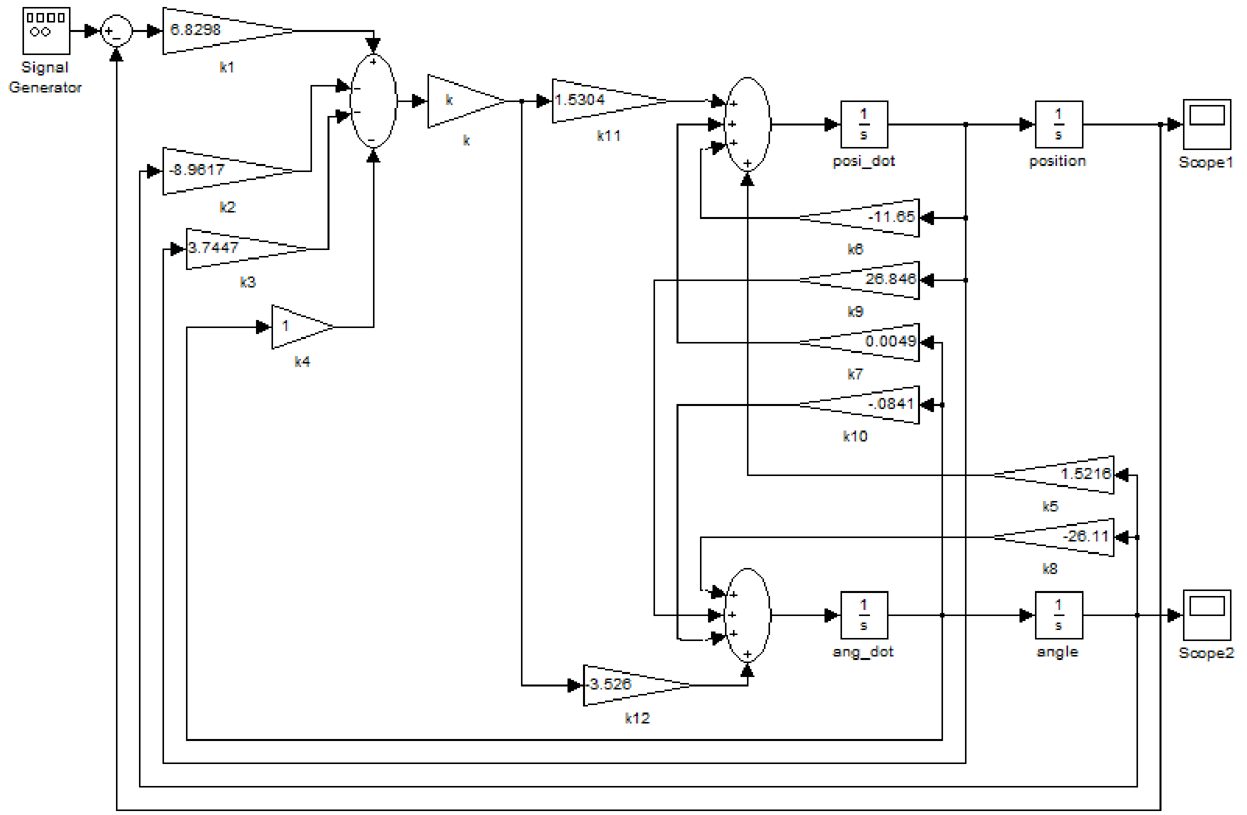
$$G_a = \frac{-3.526s + 0.007218}{s^3 + 11.73s^2 + 26.96s + 263.3}$$

(You may use the "[num,den]=linmod('filename')" SIMULINK command to find the numerator and denominator of the transfer functions. This command requires that the input and outputs of your model be labeled as inports and outports).

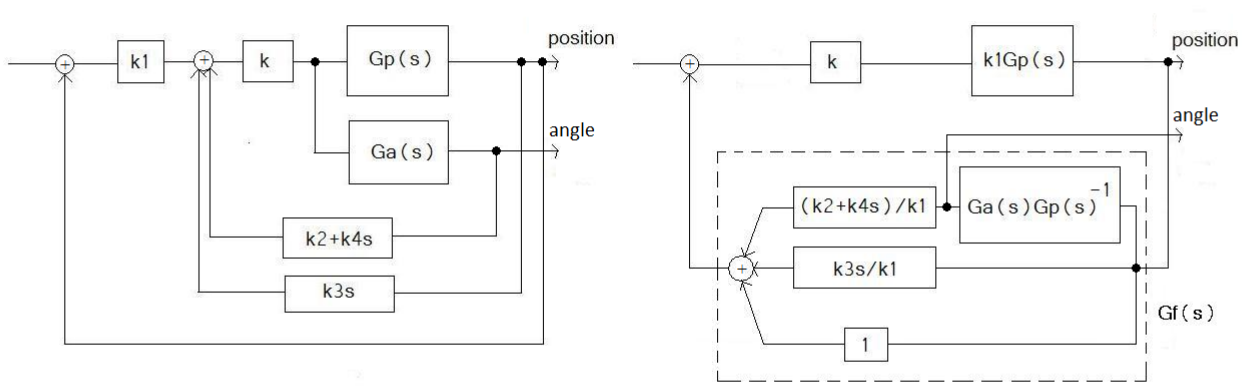
Clearly (why?), the open loop system is not stable. The control goal is:

1. To stabilize the system.
2. To force the position $p(t)$ to follow a desired position commands so that the steady state tracking error for a step input is zero and $T_r \le 1.2$ (sec), $T_s \le 2.5$ (sec) and $\%OS \le 7\%$.
3. To minimize the swing of the pendulum, i.e., to minimize the angle $a(t)$ while $p(t)$ moves from one position to another position. This can be achieved if the response of $a(t)$ to a step input is small and converges to zero.

A linearized closed-loop model of the hanging pendulum system is shown below.



Let k1 = 6.8298, k2 = - 8.9617, k3 = 3.7447 and k4 = 1. Determine the value of k so that the above control requirements are met. Verify your designs by MATLAB simulation. The following diagrams may help you in finding such a gain k.



Characteristic equation $1+kG(s)=0$, where $G(s)=k1Gp(s)Gf(s)$

Given the closed-loop system's characteristic function you can complete a root locus design using MATLAB's "rlocus" and "sgrid" commands.

Note: Manipulation of transfer function objects is most easily accomplished using MATLAB's block diagram commands "feedback," "parallel", and "series", and their equivalent arithmetic operations, e.g., "parallel(T1,T2)" can be shortened to "T1+T2" and "series(A,B)" can be shortened to "A*B".

In the lab, your controller will be excited by a 0.1 Hz square wave. To simulate your system's square wave response first create, from the block diagram at right, a MATLAB representation of the closed-loop system's position and angle transfer functions for your k. More specifically, let

T = [feedback(k*k1*Gp, Gf); feedback(k*k1*Gp, Gf)*Ga/Gp]; .

Next generate a 20 sec. segment of a square wave with a 10 sec. period (frequency 0.1 Hz) sampled every 0.1 sec. using MATLAB command

[u, t] = gensig('square', 10, 20, 0.1);

Finally, simulate the response using MATLAB command

lsim(T, u, t)

The upper plot produced depicts the cart's position response for your design, the lower, the cart's pendulum's angle response for the corresponding cart motions. Use the MATLAB command

step(T)

to assess your design's rise time, settling time, and percent overshoot performance. Repeat these simulations and performance assessments for several k values suggested by your root locus analysis (e.g., low, moderate, and high k values), print the plots, and bring these with you to lab.

PART (B): Experiments:

Step 1: Turn on the Universal Power Module using the switch on its back.

Step 2: Level and secure the track using the vacuum clamps at either end, move the cart to the track center and set the pendulum to hang motionless.

Step 3: Log on the workstation using your engineering login name and password.

Step 4: Download lab3.zip into a local hard drive folder, say D:\lab3 (putting in on the desktop won't work).

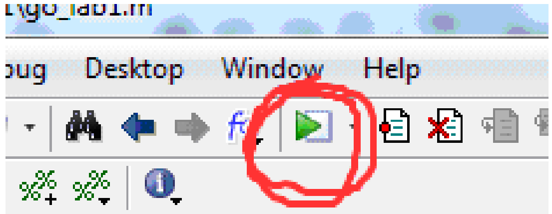
Step 5: In Windows Explorer, double-click the go_lab3.m script.

Step 6: At the MATLAB prompt, or at the top of your go_lab3.m script type, on a line by itself,

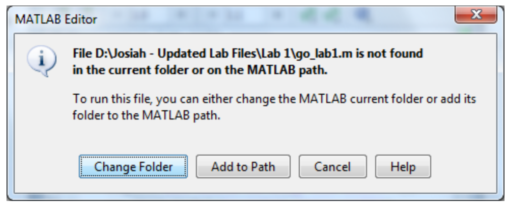
```
k = VAL
```

replacing VAL with the value you calculated for the feedback gain in the prelab.

Step 7: Click the green triangular Run button on the go_lab3.m script.



Click the “Change Folder” button and change folders if needed.



If a MEX error occurs, click OK and cancel the script

a. Type `mex -setup` in the MATLAB window
b. Type 1 to select the C++ complier
c. Answer yes to all questions
d. Rerun go_lab3.m

Step 8: When the configuration appears on screen verify that it is correct and hit enter

```
parameter_gains =

    6.8298   -8.9617    3.7447    1.0000

STATUS: manual mode
The model parameters of your Single Pendulum and IP02 system have been set.
You can now design your state-feedback position controller.
PEND_TYPE=LONG_24IN
CART_TYPE=IP02
UPM_TYPE=UPM_2405

The feedback gain is set to:

k =
    6
fx If this doesn't look right, hit Ctrl+C. Otherwise, hit enter to continue
```

Step 9: Verify that the MATLAB window contains a build output line, similar to the following, indicating a successful executable build and download.

```
### Model q_sswe_ip02 has been downloaded to target 'shmem://quarc-target:1'
fx Hit enter to start the demo. Make sure a TA has verified your gain value.
```

Step 10: After a TA has verified your gain value, hit enter to start your controller and release the seesaw. **Keep your finger near the enter key to stop the controller should the cart and seesaw start to go unstable.**

```
fx Hit enter to stop the demo.
```

Step 11: When you are finished observing the cart, hit enter in the MATLAB window to stop the controller

```
fx Hit enter to stop the demo.
```

Step 12: Modify k and repeat steps 5-11 until you have found an acceptable gain.

Step 13. Be sure to save your experimental data for comparison with your simulated plots and lab report analysis.

a. Specify the file for saving your modified SIMULINK model from part A by clicking File → Save As… → D:\lab3\lab3.mdl
b. You can save the real-time data acquired in your cart position (mm), seesaw angle (deg), and command voltage (V) plots by first making sure they are displayed in the plot windows, then running the following MATLAB command:
   ```
   >> save 'D:\lab3\lab3data.mat' data_xc data_ssw data_Vm
   ```
c. To view and access the data at a later date, load the data into MATLAB by typing
   ```
   >> load lab3data.mat
   ```

Step 14. Once you have completed your work. Turn off the Universal Power Module, close MATLAB and log out.