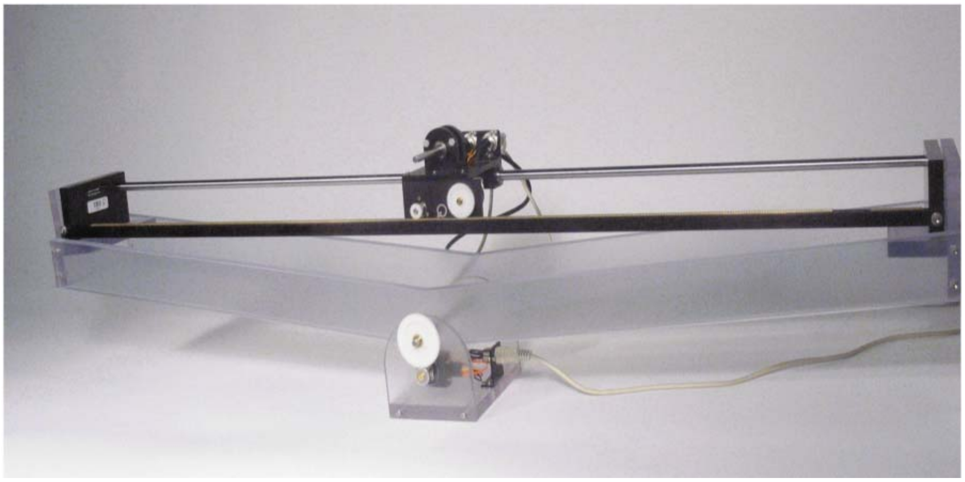
ECE:3600 CONTROL SYSTEMS – Spring 2018

The University of Iowa College of Engineering

Experiment #2 – SeeSaw Balancer

Room 2245 SC



Objectives:

a) To design, simulate, evaluate and tune a controller that can maintain seesaw balance by driving a motorized cart back and forth on the seesaw.
b) To compare real system responses to those predicted by a linear model simulations.

Warnings:

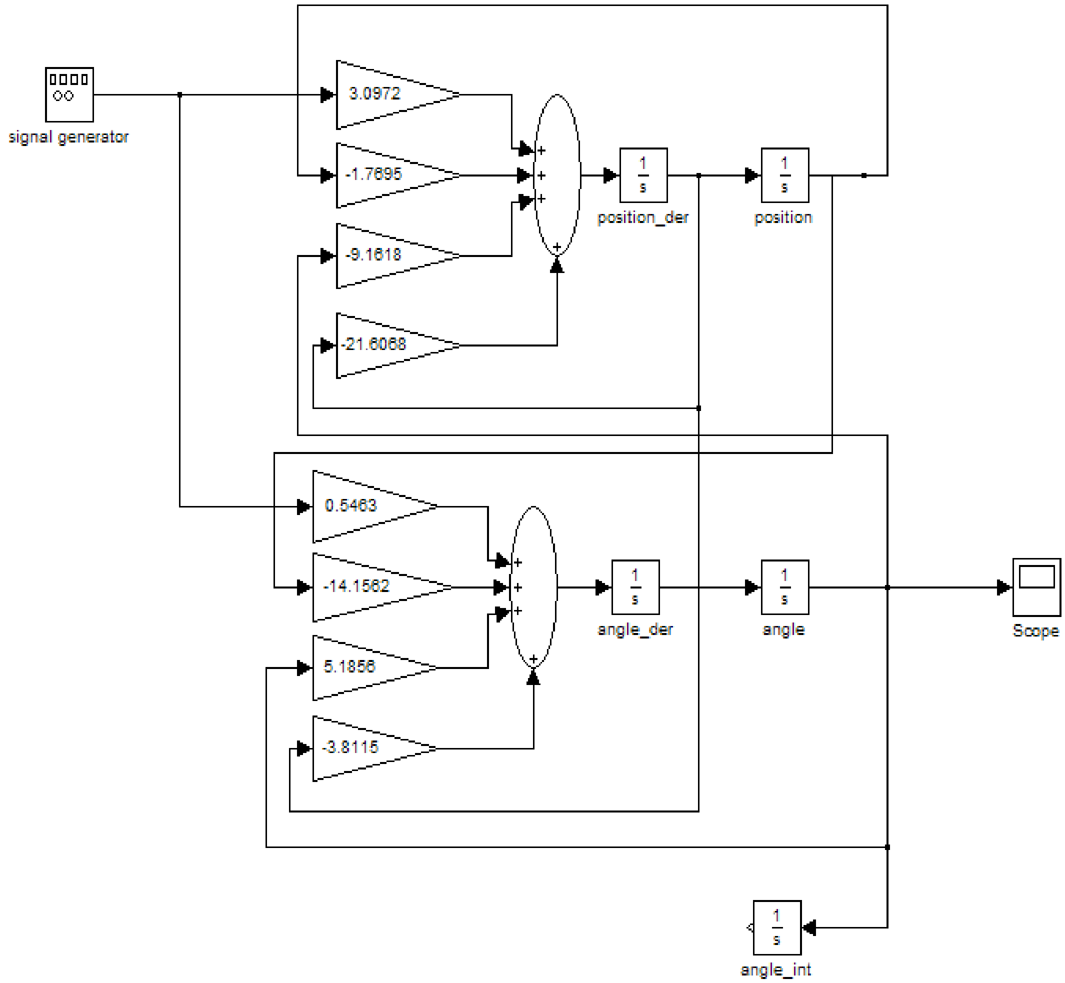
a) Do not alter any electrical connections. Lethal voltages are present.
b) Keep all foreign objects, including fingers, heads, clothes, hair, and pencils, away from the cart and pendulum. The motor generates torque sufficient to crush objects caught in its gears and move the cart and pendulum at speeds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

Each student must submit their own lab report. The report should include:

a) The time and date of your lab and your partners' names.
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 and closed-loop transfer functions, (2) the root locus plot used to complete your design, and (3) SIMULINK plots of your design's ***simulated*** output angle for random and square wave inputs.
d) SIMULINK plots of your initial and final designs' ***actual*** **(i.e., measured)** output angles in steady-state, AND immediately following a 1° (tap induced) balance-angle disturbance. As possible superimpose your design's actual plots *on top of* its Malab simulations. **You will need to save your lab data to a file to do this.**
e) A brief evaluation of your simulated and experimental results, identifying possible causes of discrepancies, and assessing (both qualitatively and quantitatively) your design's overall performance. Issues addressed should include: (1) Steady-state behavior: Does system achieve balance? Are its motions random? Periodic? Why might it behave this way? (2) Transient behavior: Can we learn anything about the system's response to a tap disturbance from its square-wave driven simulation? Are the response shapes similar? Why does the simulated response (generated by "step" or a simulink square wave simulation) initially move in the opposite direction of the initial step? Could the settling time have been predicted? (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 model of the seesaw system is shown below. Click here to download a copy.



Verify:

(1) That the transfer function from the signal generator voltage input (modeling the output of the UPM amplifier) to the seesaw angle $a(t)$ is

$$G_{va}(s) = \frac{0.5463s^2 - 0.001183s - 42.88}{s^4 + 21.61s^3 - 3.416s^2 - 147s - 138.9}$$

(2) That the transfer function from the signal generator voltage input to the cart position $p(t)$ is

$$G_{vp}(s) = \frac{3.097s^2 - 21.07}{s^4 + 21.61s^3 - 3.416s^2 - 147s - 138.9}$$

(3) That the system's open loop poles are -21.46, -1.78, -1.26 and 2.89.

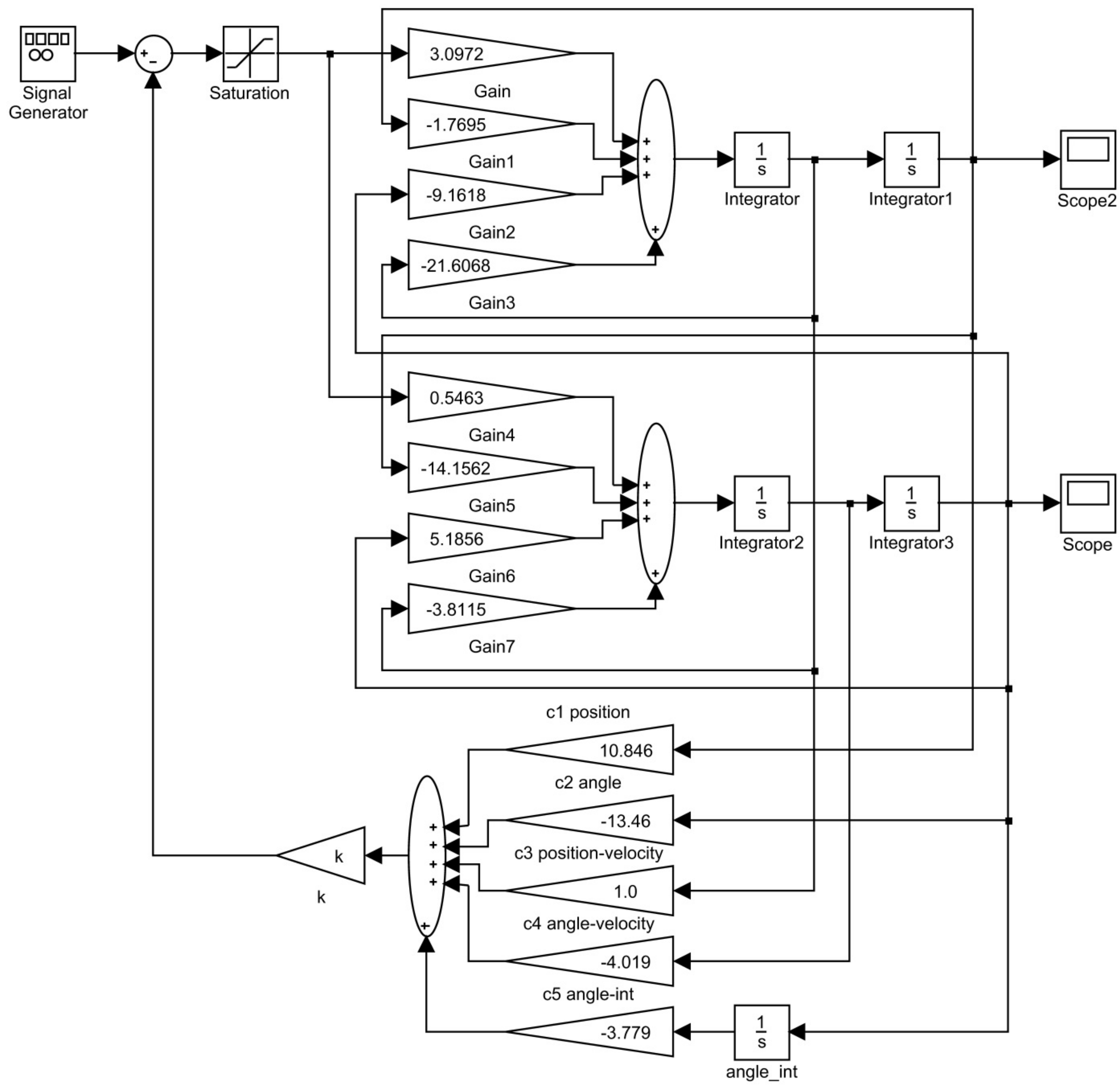
(You may use the "[num,den]=linmod('filename')" SIMULINK command to find the numerator and denominator of the transfer functions. This command requires that inputs and outputs be labeled as inports and outports as has been done in downloadable model).

Due to the pole at 2.89 the seesaw system is open-loop unstable (i.e., it is naturally unbalanced). To maintain seesaw balance, we need to design and implement a controller that ensures that the closed-loop seesaw system is stable (i.e., a controller that ensures that all closed-loop poles are left-hand plane poles).

Step 1: The signals that you have access to are $p(t)$ and $dp(t)/dt$, the cart's position and velocity, and $a(t)$, $da(t)/dt$, and $\int a(t)dt$, the seesaw's angular position, velocity and position integral. Your challenge is to design a feedback control

$$u(t) = -k*\left[c1*p(t) + c2*a(t) + c3*\frac{dp(t)}{dt} + c4*\frac{da(t)}{dt} + c5*\int a(t)dt\right],$$

as shown in the following model, ensuring that the closed-loop system is stable, and that $p(t)$ and $a(t)$ are maintained close to zero, i.e., that the seesaw is balanced.



By hand, and/or using block diagram reduction, the MATLAB commands parallel and series, and the $G_{va}(s)$ and $G_{vp}(s)$ computed earlier using MATLAB, verify that the closed-loop transfer function from the signal generator input to the seesaw angle is given by

$$\frac{0.5463s^3 - 0.001183s^2 - 42.88s}{s^5 + p_1 s^4 + p_2 s^3 + p_3 s^2 + p_4 s + p_5}$$

where

$$p_1 = 21.6068 + 3.09772 * k * c3 + 0.54638 * k * c4$$
$$p_2 = -0.0012 * k * c4 - 3.4161 + 0.5463 * k * c2 + 3.0972 * k * c1$$
$$p_3 = -21.0659 * k * c3 - 42.8779 * k * c4 - 0.0012 * k * c2 + 0.5463 * k * c5 - 146.9644$$
$$p_4 = -21.0659 * k * c1 - 138.8722 - 42.8779 * k * c2 - 0.0012 * k * c5$$
$$p_5 = -42.8779 * k * c5$$

and $c1$=10.8458, $c2$= -13.4583, $c3$=1; $c4$= -4.0192 and $c5$= -3.7792.

Step 2: Find *k* so that the closed-loop system is stable and its dominant poles satisfy

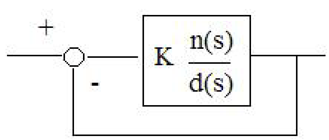
$$t_r \le 3\,(\text{sec}),\ t_s \le 7.5\,(\text{sec}) \text{ and } \zeta \ge 0.78\,.$$

To find *k*, you may use the root locus method. To this end, notice that

$$s^5 + p_1 s^4 + p_2 s^3 + p_3 s^2 + p_4 s + p_5 = d(s) + kn(s) = 0$$

$$d(s) = s^5 + 21.6068s^4 - 3.4161s^3 - 146.9644s^2 - 138.8722s$$
$$n(s) = 0.9012s^4 + 26.2525s^3 + 149.2205s^2 + 348.5348s + 162.0442$$

As roots of the above equation are exactly the closed-loop poles of the unity feedback system with the open loop transfer function $kn(s)/d(s)$,



plotting this system's root locus using MATLAB command "rlocus" allows graphical determination of *k* via ltiview data points or the MATLAB command "rlocus" and the "+" data cursor..

Notes: 1. You do not want to choose values of *k* that are too large as this will increase the system's sensitivity to noise and under-modeling errors arising from linearization. 2. While the above unity feedback system has the same poles as the original closed-loop system *it does not* have the same closed-loop transfer function.

Step 3: Download a SIMULINK model of closed-loop system by clicking here. Double click on the feedback gain element *k* to enter your design value. Double click on the angle scope to view the seesaw angle and double click on the signal generator to select your input. Begin by setting the generator waveform to "random" and the amplitude to "2". Run the simulation by clicking on the back arrow in the green circle near the top of the screen in the center. Autoscale the scope display by right clicking the scope and selecting "autoscale". When satisfied click on the printer icon to print. Next, set the signal generator waveform to a "square wave" with frequency 0.1 (Hz) and magnitude 2. Once again run the simulation, autoscale the resulting scope display, and print the display when satisfied. Bring these plot prints to your lab and submit them with your report.

PART (B): Experiments:

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

Step 2. Verify: (1) that the load has been removed from the top of the cart, (2) that the pendulum has been removed from the front of the cart and (3) that the clamps have been detached from the ends of the seesaw. Carefully move the cart to the middle of the seesaw track while your lab partner holds one end of the seesaw to keep it balanced.

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

Step 4: Download [lab2.zip]() into a local hard drive folder, say D:\lab2 (putting in on the desktop won't work).

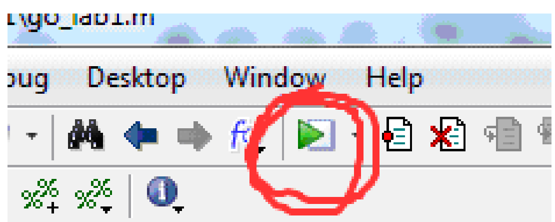
Step 5: In Windows Explorer, double-click the go_lab2.m script.

Step 6: At the MATLAB prompt, or at the top of your go_lab2.m script type, on a line by itself,

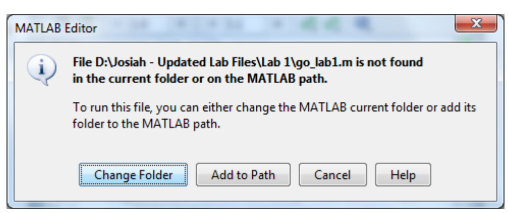
```
k = VAL
```

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_lab2.m script.



Click the "Change Folder" button and change folders if needed.



If a MEX error occurs, click OK and cancel the script and: (a) Type `mex -setup` in the MATLAB window, (b) Type 1 to select the C++ complier, (c) Answer yes to all questions and (d) Rerun go_lab2.m.

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

```
tilt_angle_correction_factor =

    0.6870
parameter_gains =

   10.8458  -13.4583    1.0000   -4.0192   -3.7792

STATUS: manual mode
The model parameters of your Seesaw with IP01 or IP02 system have been set.
You can now design your state-feedback position controller.

IP01_2_WEIGHT_TYPE=NO_WEIGHT
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 6-11 until you have found a gain that balances the seesaw. Save a copy of the autoscaled steady-state response for this *k* using the commands listed below.

Step 13: While in steady-state, introduce a 1° disturbance by gently tapping on one end of the see-saw at the beginning of a scope trace. Immediately before the trace completes hit return to stop the experiment (and trace). Once again save a copy of the autoscaled steady-state response using the commands listed below.

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

To save your simulated and experimental data (as opposed to printing it):

a. Specify the file for saving your modified SIMULINK model from part A by clicking File → Save As… → D:\lab2\lab2.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:\lab2\lab2data.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 lab2data.mat
```