Oscillating Movement via PID

Task #2
The goal is the same as task #1. Somehow make the device oscillate position between 45° and -45°. But this time use a parallel PID controller.

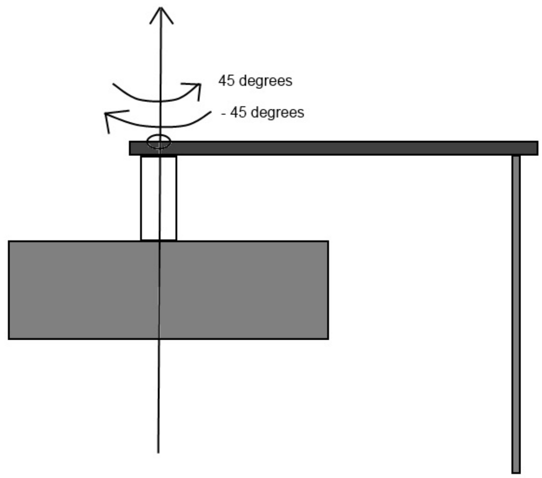

Figure 1: Overview of oscillating movement around center pivot between 45° and -45°

Background: Systems and PID Controllers
A system can be modeled using a mathematical equation with an input and an output. This mathematical equation with an input and output is called a transfer function. In the last lab, the system was formed by creating arbitrary input voltages and viewing the resultant positional outputs.

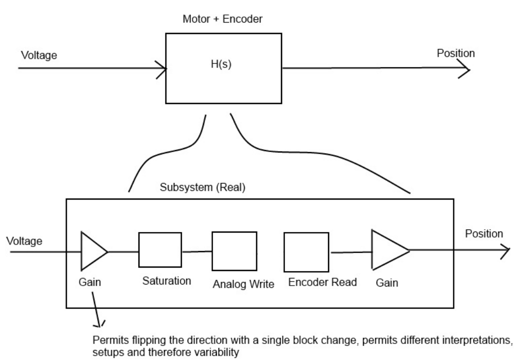

Figure 2: System from last lab.

This however is fixed, the input voltage does not respond dynamically given the output position. To improve upon its design we can implement some form of controller to form a more competent overall system. A simple type of controller is a closed loop negative feedback controller. A desired position (in our case an angle) is input to overall system. The input (desired position) is compared against the actual output position by calculating the difference between the two. We call the result of this difference the "error." **The error feeds into the compensator (or controller) which in turn dynamically calculates a new control voltage to feed into the transfer function we created before.** This closed loop negative feedback system is inherently stable. Think of it like a ball in a bowl (as compared to a ball on a hill).

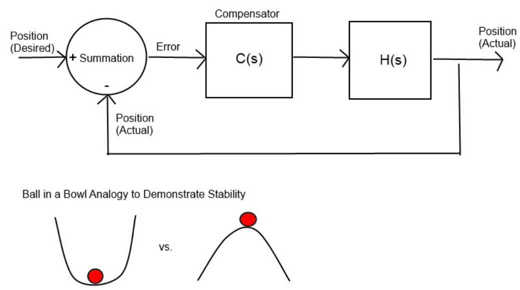

Figure 3: Diagram of a simple negative feedback closed loop system.

The compensator (or controller) can be expanded in many ways. In lab #3, the implemented choice is a PID controller. A PID controller calculates an error value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable.A PID controller provides variability and dynamic control. P,I and D stand for proportional, integral and derivative terms respectively. These values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. The weighted sum of the three values is used to adjust the control parameter (in our case voltage). It is possible to obtain only the effects of certain values in the controller... for example, setting both "I" and "D" to 0 would yield a simple P controller.
The proportional term is easy to think of setting the control parameter proportional to the error. In a simple water tap controller, the tap could be set in proportion to the error-¿temps under the desired temp would need hot water and temps over the desired temp would need cold water. The derivative term considers the rate of change, ie: adding extra hot water if the temperature is falling, and less on rising temperature. Lastly the integral action uses the average temperature in the past to detect whether the temperature of the container is settling out too low or too high and set the tap proportional to the past errors. [Sources: Wikipedia]

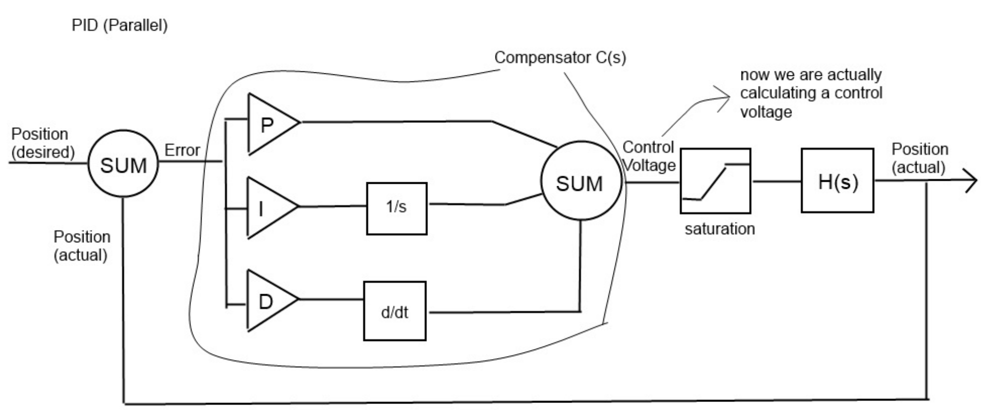

Figure 4: Diagram of a simple parallel PID controller.

A controller is defined by how it reaches setpoint after a step change... this includes how closely it reaches the setpoint, how quickly it gets there and if it overshoots or oscillates in doing so. Tuning the parameters of the PID controller enable the system to hone in on the best response to a step change.

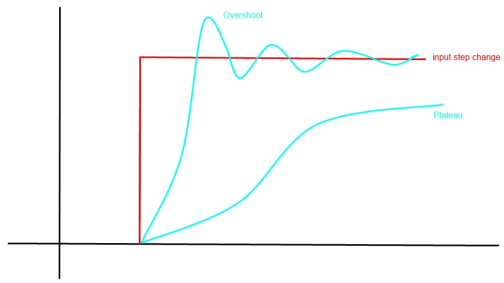

Figure 5: Different system step responses.

"P" term: a proportionality term that is too high will lead to instability as it will overshoot too dramatically. A proportionality term that is too low will lead to an unresponsive system or at least a slow one.
"I" term: an integral term is responsible for accelerating the movement of the process towards the setpoint and eliminating residual steady state error inherent to purely proportional controllers. If the integral term is too large the system will overshoot, but if the integral term is too small the system will take a longer to reach setpoint.
"D" term: a derivative term improves settling time and stability of the system... ie: if a system is oscillating with decreasing amplitude,the proper derivate term will lead to a faster settling to setpoint.

Performance of a controller is often defined by its rise time and its settling time. There are constraints in certain conditions, for example some systems cannot allow overshoot at all while others must optimize output voltage for energy consumption etc. In every case a proper balance is result of fine tuning.

Steady State Error the resultant constant error when a system cannot reach a desired value.
Transient Error the error during the process of a system getting to the desired setpoint

Task #2: Method & Observations
Created a simple PID controller with a simple repeating step input.

Simulink Palette:

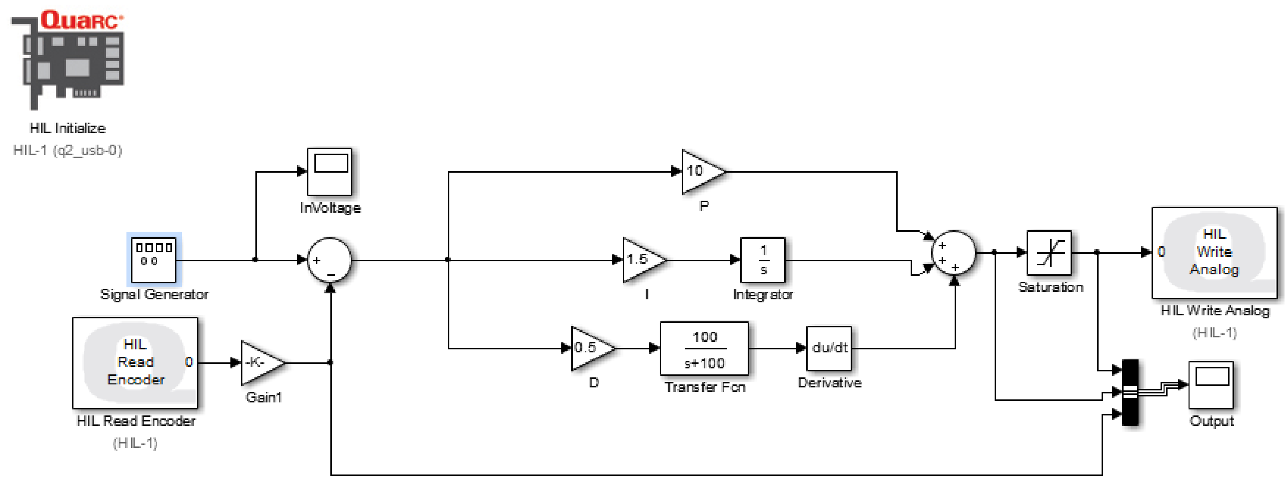

Figure 6: Simple PID Controller Simulink Palette Schematic with repeating step input.

Then changed around different values for PID and looked at the resultant scope readings. In each scope reading...
blue line = position
purple line = pre saturation control output voltage
yellow line = post saturation control output voltage

1. P,I,D = 1, 0, 0:

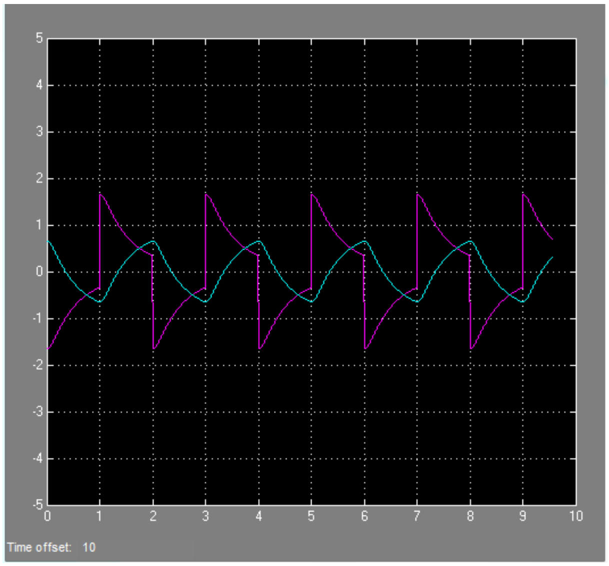

Figure 7: Step Response of PID controller with values 1,0,0.

2. P,I,D = 5, 0, 0:

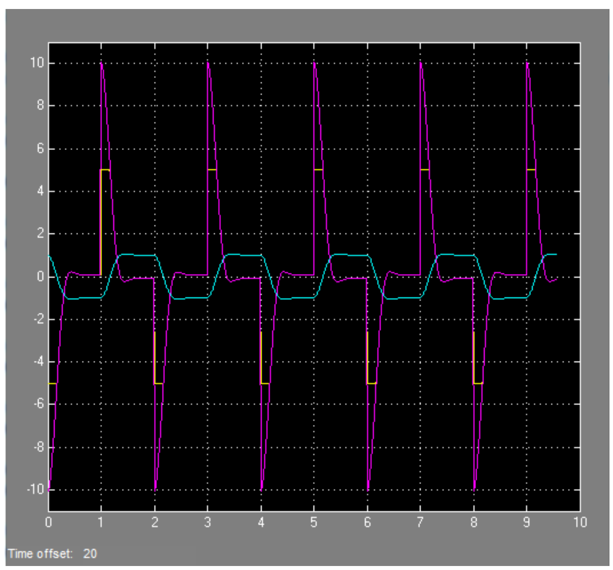

Figure 8: Step Response of PID controller with values 5,0,0.

3. P,I,D = 1, 1, 0:

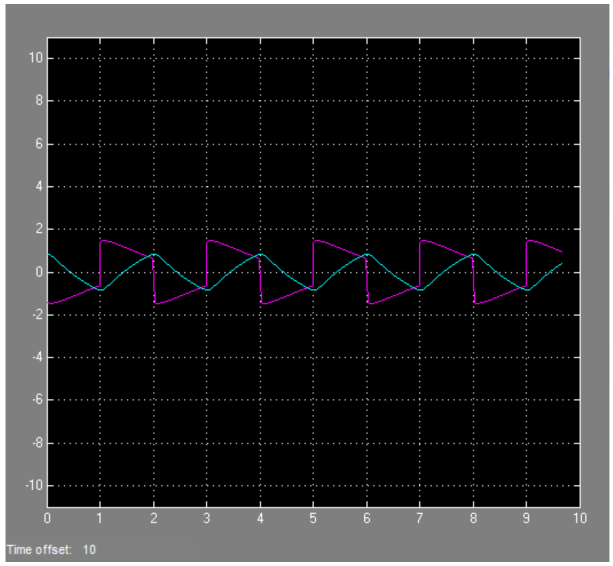

Figure 9: Step Response of PID controller with values 1,1,0.

4. P,I,D = 1, 5, 0:

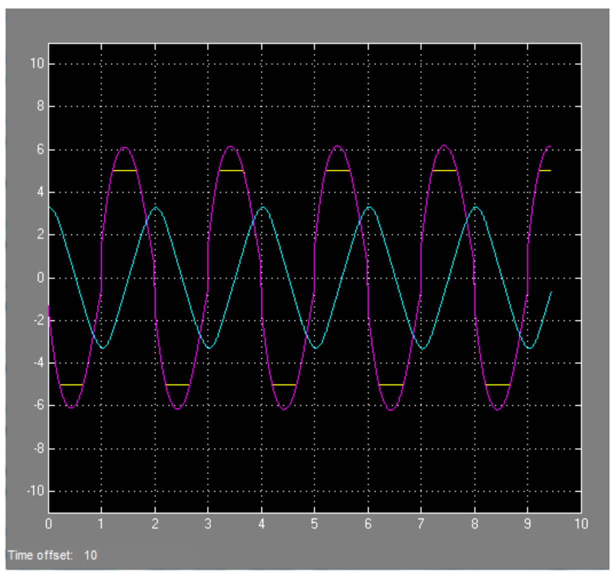

Figure 10: Step Response of PID controller with values 1,5,0.

5. P,I,D = 1, 1, 1:

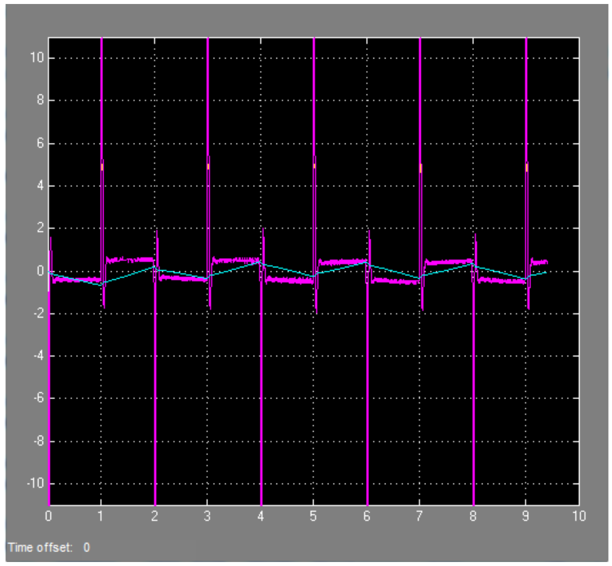

Figure 11: Step Response of PID controller with values 1,1,1.

6. P,I,D = 10, 1.5, 0.5:

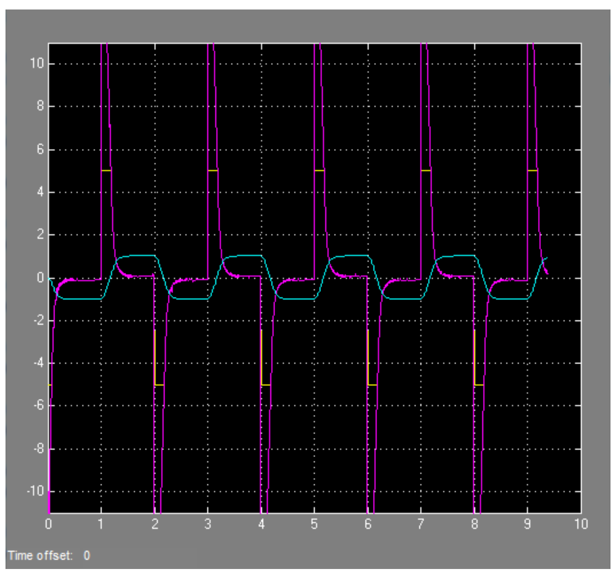

Figure 12: Step Response of PID controller with values 10,1.5,0.5.

7. Physically shaking and fighting the movements of the robot demonstrated the true dynamic behavior of the control voltage and the benefits of a PID controller. See how the output control voltage fluctuates dramatically to keep the position relatively on track.

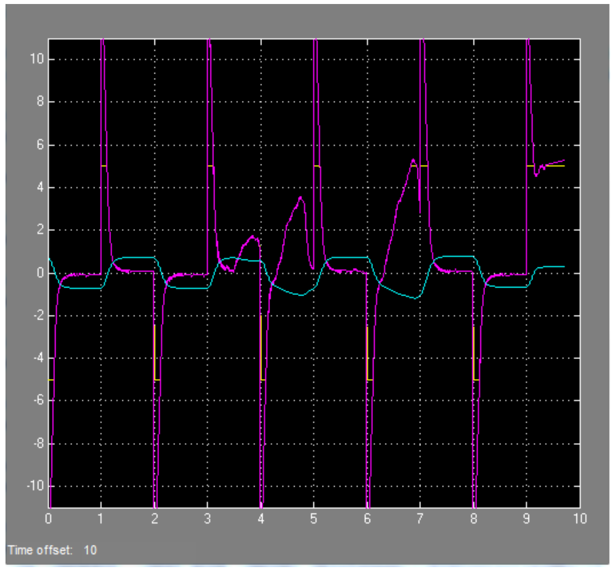

Figure 13: Response of PID controller under physical perturbations affecting movement demonstrating dynamic output voltage capabilities.

Task #3: Method & Observations
The objective of task#3 was to oscillate the system with various waveforms.

1. System Response to Sinusoidal input.

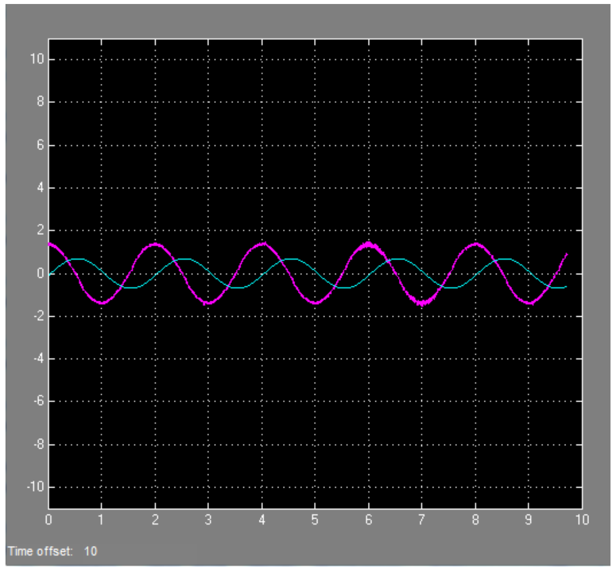

Figure 14: Response of PID controller to sinusoidal input

2. System Response to Sawtooth input.

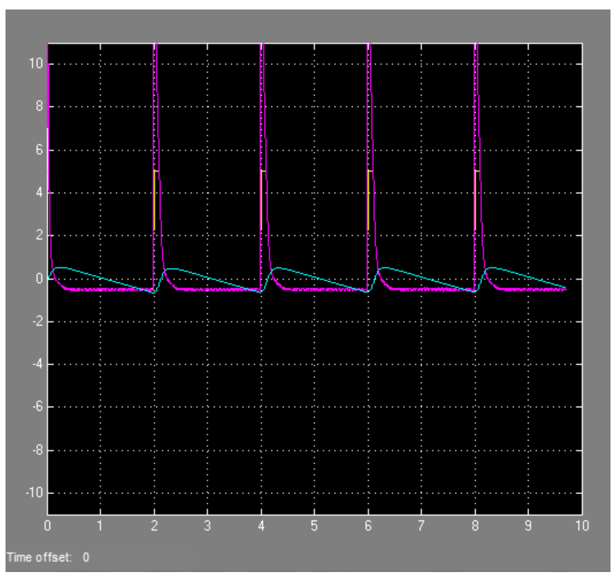

Figure 15: Response of PID controller to sawtooth input

3. System Response to Triangular input.

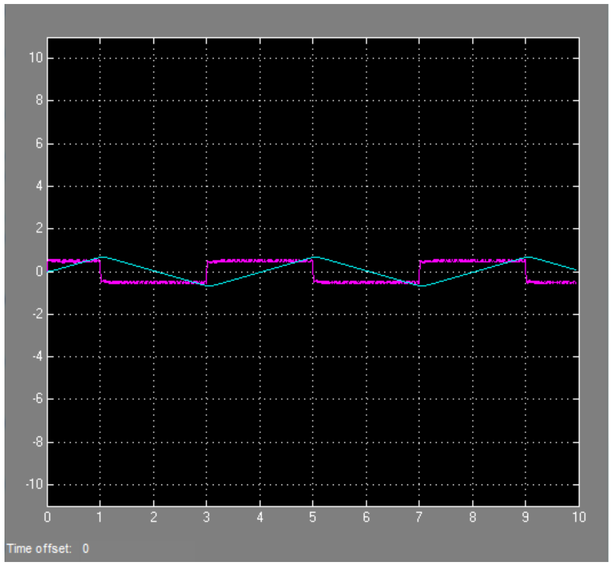

Figure 16: Response of PID controller to triangular input

Task #4: Method & Schematic
The objective of task#4 was to move the link to any given θ (position) and hold the position. To accomplish this, a constant value was fed into a slider gain w/ a range of -π : π yielded positional control simply by selecting the angle on the slider.

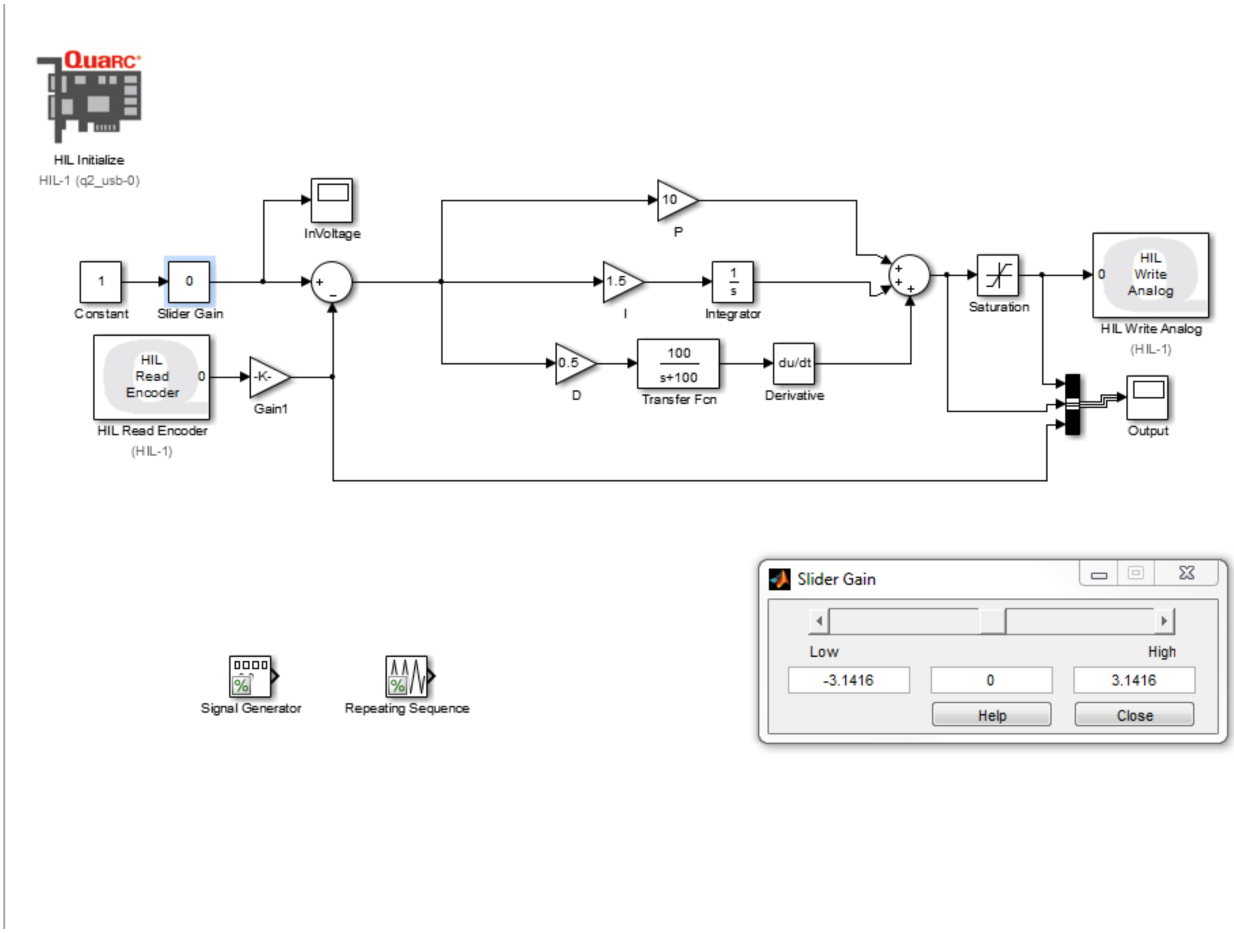

Figure 17: PID controller with a slider gain to hold any position.

Task #4: Method & Schematic

The objective of task#4 was to create a control screen with three selectable modes with controls as listed below:

1. Move to zero position.
2. Move to position "X°".
3. Oscillate between "$\pm X°$

To accomplish this, a set of cascading switches were implemented.

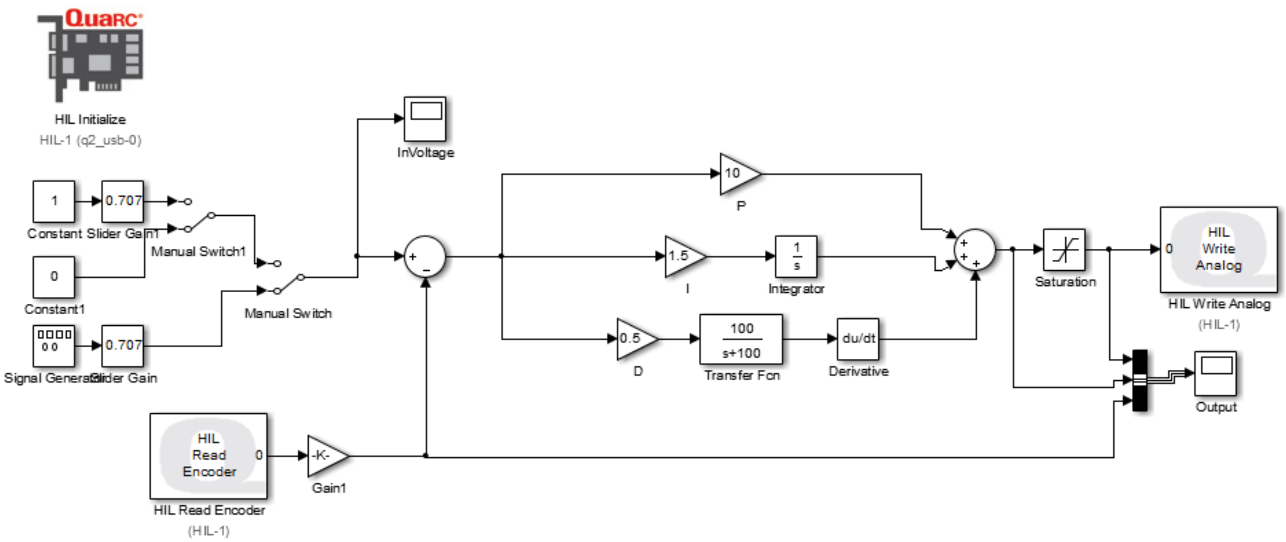

Figure 18: PID controller selectable input.