Robotics Lab Notes

David Young

December 7, 2014

Abstract

Introduction and Filters

Definitions & Useful Information

Servos: A motor with a position encoder capable of knowing its own position. You always want to put the encoder on the component that is actually moving but most servos have encoders on the shaft.

Encoders: Digital input devices. Digital, so plugged straight into little block box. Has a little plastic disk w/ slits and a light source/detector. The detector sees a square pulse train as the disk spins and light filters through the slits. Position can be calculated incrementally or absolutely. Incremental position is relative to the starting position when the servo is turned on.

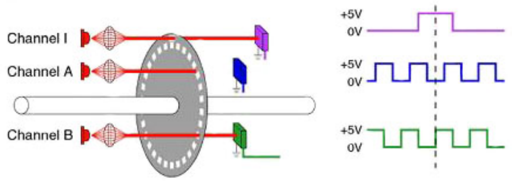

Figure 1: Rotary Encoder with two detectors and resulting pulse trains.

Quadrature: Commonly encoders use two detectors to increase accuracy without increasing the # of slits. Additionally the detectors note both the rising edge and falling edge of the pulses, therefore doubling the markers from a single pulse. With two detectors seeing different pulse trains, and noting both the rising and falling edge of each pulse, one can tell the direction by the order of signals between A+,A-,B+ and B-.

Low Pass Filter: Using a low pass filter will help remove noise from a waveform. See the bodeplot figure.

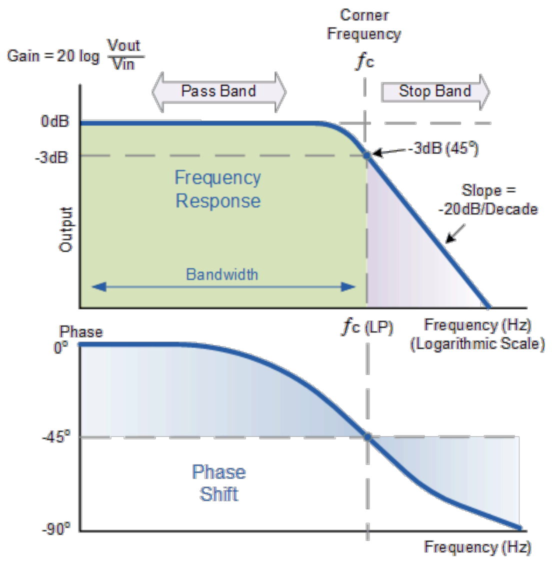

Figure 2: BodePlot of a Low Pass Filter: demonstrates how higher frequencies that can contribute to noise are attenuated.

Simulink Pallet Familiarization

Run Types

"Normal":

Runs within simulink only.
Interpreted.
Simply a simulation

"External":

Runs on real external hardware.
Compiled... in order to run quickly enough the code must be compiled, assembled, linked and loaded.
Accomplished by hitting "build", after which one should see "Code downloaded to target"

Time Basis

Note: "inf" = infinite, otherwise ex: "10sec" = 10 seconds

Connect to Target

Only if in external mode

Basic Workflow Overview

Process

Build >Connect to Target >Play
Note: the code remains running in the box until terminated by a command, even if the computer shuts down.

Procedure

1. **Place an "HIL Initialize" block onto the pallet**
   This doesn't need to be connected to anything, but must be on the pallet. It tells all the other blocks what hardware to connect to. For our use, set the appropriate hardware on the HIL initialize by changing the type to Q2 sub (meaning 2 ports over a usb connection).
2. **Place a "Write Analog" Block from the "Immediate I/O" onto the pallet**
   This will be used to write data to the motor.

   Note: Analog signals generally require going through the power supply.
3. **Place a "Constant" Block as input to the HIL Write Analog**
   Set the constant value to 3 Volts.
4. **Place a "Saturation" Block between the constant & the Write Analog**
   Set the limits to + or - 5 Volts. This will serve as a safety.

   Observations: The sign of the constant values dictates the direction of movement. A larger constant value dictates the speed of the movement, but only until saturation and then the speed plateaus.
5. **Add a "Sine Wave" input w/ a "Manual Switch" to select between the two inputs.**

Procedure 2

1. **Set the motor to constant input w/ value of 0**
   This is simply to prevent the motor from moving during the next steps.
2. **Place an "HIL Read Encoder" Block onto the Pallet**
   Select the channel with the chord plugged into the robot.

3. **Place a "Scope" Block onto the output of the "HIL Read Encoder"**
   This will provide a visual representation of the position of the servo.
   To pretty up the pallet, simply place the motor write and the encoder read inside a single box.

   Observation: Zooming in on the scope shows that the position measurements are in discrete steps, despite the continuous movement of the arm. This means that the derivative (used for speed) is a constant flip flop between 0 and infinity...ie: useless noise. We see a waveform similar to the left side of the figure, but we'd prefer to see something similar to the right side.

Figure 3: Sinusoidal Waveforms before and after application of low pass filter.

   To make an average trend line, we need to use a filter to remove the high frequency components. This relates directly to the Fourier series (infinite series of sine waves), where getting rid of high frequencies w/ a low pass filter would help prevent noise.

4. **Find the number of slits per rotation**
   Found by averaging the # of discrete steps per single revolution. If one rotation of the scope demonstrates a step function to a certain height on the scope, then the total height reached in one rotation divided by the height of a single discrete step yields the number of steps in one rotation.
   Observation: The number of slits per rotation was found to be 4096 $\frac{\text{counts}}{\text{revolutions}}$. That is equivalent to 2^{12}.

5. **Convert the $\frac{\textbf{counts}}{\textbf{rev}}$ to $\frac{\textbf{radians}}{\textbf{rev}}$ with a "Gain" Block.**

$$2\pi \text{ rads} = 4096 \text{ counts}$$
$$\frac{\text{rad}}{\text{counts}} = \frac{2\pi}{4096}$$

6. **Use a sine input w/ a good frequency to make a sinusoid scope reading**

7. **Place a "Transfer Function" Block between the gain and the scope to act as a low pass filter**
   Use a value of $\frac{100}{s+100}$ for the transfer function which represents a low pass filter w/ a gain of 1. This is because "s" is the same as "jw" and if the frequency "w" is 0, the gain would go to $\frac{100}{100} = 1$.

8. **Use a "Bus" to input both the unfiltered and filtered signal into the scope**
   Observation: Both signals were visible if one zoomed in close enough, the unfiltered was the same discrete stepping sinusoid, and the filtered was a smoothed average. See the figure.

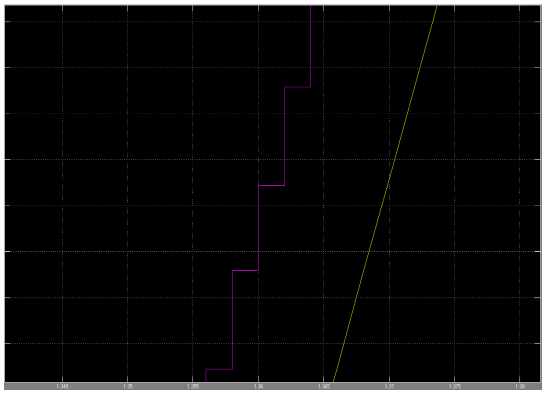

Figure 4: Scope Read comparing the filtered (yellow) and unfiltered (pink) signals.

Final Palette Schematic:

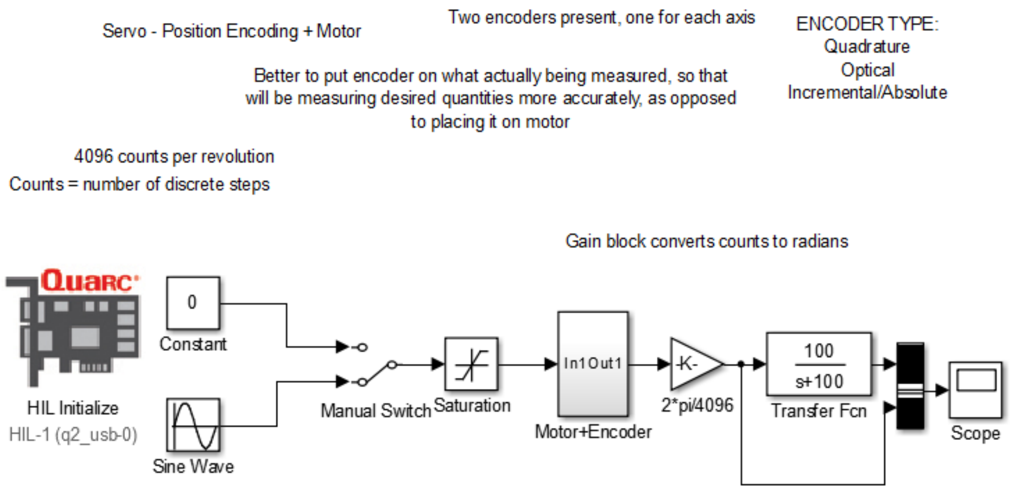
Servo - Position Encoding + Motor
Two encoders present, one for each axis
ENCODER TYPE:
Quadrature
Optical
Incremental/Absolute
Better to put encoder on what actually being measured, so that will be measuring desired quantities more accurately, as opposed to placing it on motor
4096 counts per revolution
Counts = number of discrete steps
Gain block converts counts to radians
QUARC
HIL Initialize
HIL-1 (q2_usb-0)
0
Constant
Sine Wave
Manual Switch
Saturation
In1 Out1
Motor+Encoder
-K-
2*pi/4096
100
s+100
Transfer Fcn
Scope

Figure 5: Final Palette Schematic in simulink.

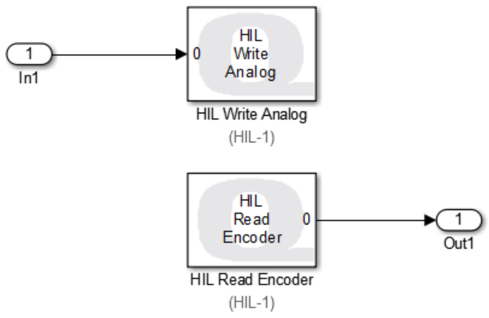

Figure 6: Contents of the motor and encoder box in the full schematic.