Control Systems Lab

Exp:7 Measurement of servo speed and Pendulum moment of inertia

- Measurement of servo speed (**FILTERING)**
- Topics Covered in this exp. using an encoder to measure speed, and Low-pass filters.
- The low-pass filter is used to block out the high-frequency components of a signal.
- First-order filter transfer function has the form: $G(s) = \dfrac{\omega_f}{s + \omega_f},$

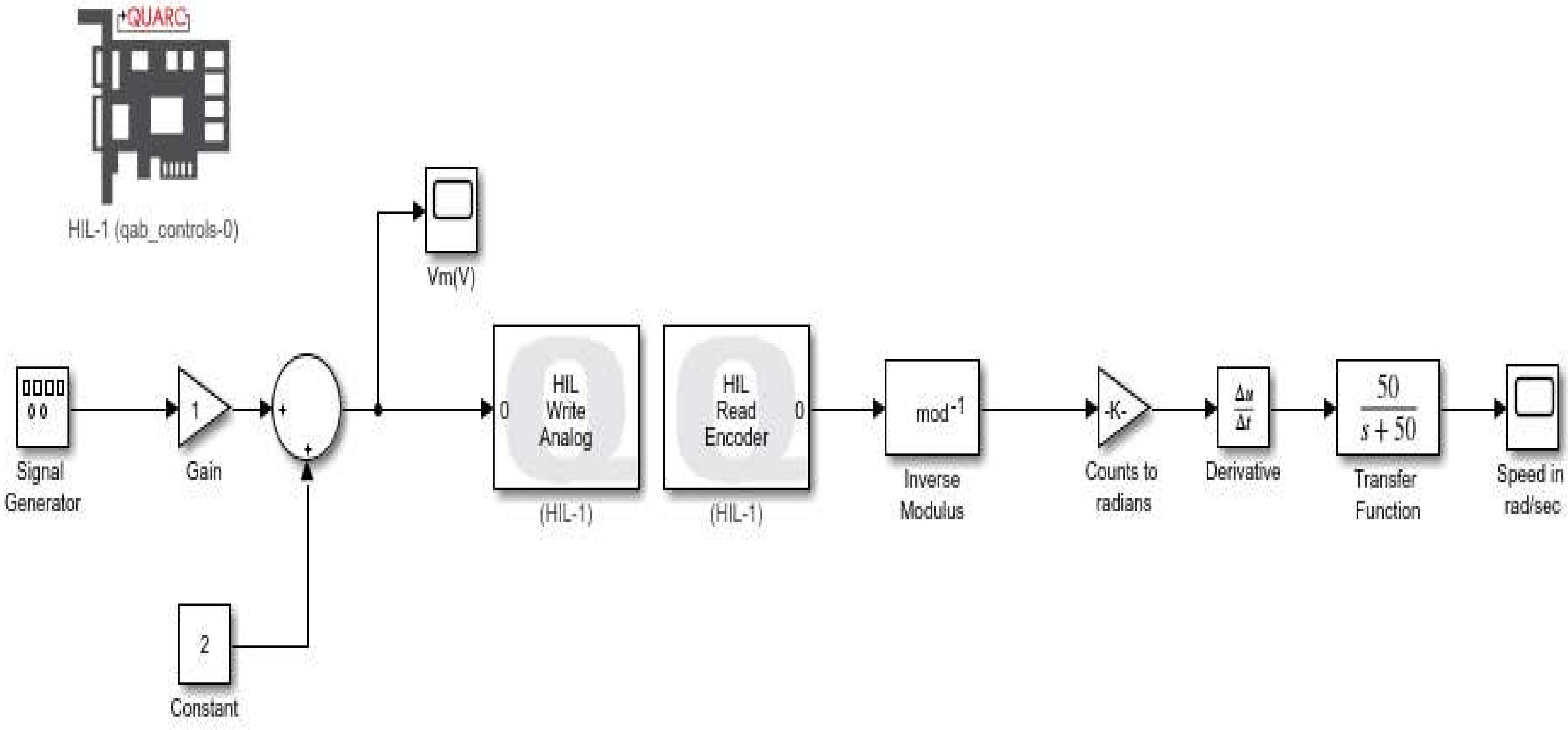
where, ω_f is the cut-off frequency of the filter in radians per seconds (rad/s).

In Lab Exercise (Hardware interfacing with MATLAB)

Open the MATLAB Window

Create a blank Simulink Model

In-Lab Exercise, design a MATLAB/Simulink diagram as shown in Figure, to measure the servo velocity using the encoder



- Build the Simulink Model, without Transfer Function block.
- Add the **HIL-1 Initialize** from Simulink Library/QUARC Targets/Data Acquisition/Generic/Configuration. (HIL: Hardware In Loop)
- Add a **Signal Generator** from Simulink Library / Simulink / Sources.
- Setup the Signal Generator to output a square voltage 3 V at 0.4 Hz.
- Add the **Gain** from Simulink Library / Simulink / Commonly used.
- Add the **Sum** from Simulink Library / Simulink / Math Operation.
- Add the **Constant** from Simulink Library / Simulink / Commonly used.
- Add the **Scope** from Simulink Library / Simulink / Commonly used.
- **HIL Write Analog** from Simulink Library/QUARC Targets/Data Acquisition/Generic/Immediate I/O
- **HIL Read Encoder** from Simulink Library/QUARC Targets/Data Acquisition/Generic/Immediate I/O

- Add the **Inverse Modulus** block: Simulink Library / QUARC Targets / Discontinuous category into the Simulink diagram. Set the field to 65536 to the Modulus terminal.
- QUBE-Servo DAQ has 16-bit counters, the valid count range is $2^{16} = 65536$.
- To eliminate the discontinuous jump that would occur when the encoder reaches the limits of this range.
- Add the **Gain** from Simulink Library / Simulink / Commonly used.
- Change the encoder calibration gain to measure the gear position in radians, i.e., instead of degrees.
- To measure the gear in radians, set the gain to $2\pi/2048 = 0.003068$ (instead of 360/2048).
- Add a **Derivative** block from Simulink Library / Simulink / Continuous to the encoder calibration gain output to measure the gear speed using the encoder (in rad/s).
- Connect the Scope at output of the derivative
- Add the **Scope** from Simulink Library / Simulink / Commonly used.
- Now, do not include the Transfer Function block (will be added later).

- **Hardware Settings:**

- Hardware Implementation: Hardware board / Determine by code generation system target file.
- Device type: x86-32(windows32)
- Click on system target file and browse and select:

quarc_linux_rt_armv7.tlc *QUARC Linux RT ARMv7 Target (NI ELVIS III, my RIO)*

- Solver:

- Simulation time: Set the start time is ‘0’ and stop time is ‘5’ sec.
- Solver selection: Fixed-step and Solver: ode1(Euler)

- Solver details: Fixed-step size is 0.002

- Click on apply and ok.

- Click on Hardware and Run the Simulation.
- Examine the encoder speed response, attach sample responses in observations.

- From the response, we are observed the encoder-based measurement is noisy.
- Measure the encoder position measurement using a new Scope.
- Zoom up on the position response and remember that this later enters derivative.
- As shown in scope, the position measurement is not continuous.
- The measurement is segmented into small steps.
- Because the encoder outputs 2048 small steps per turn, each step being $2\pi/2048 = 0.003067$ rad.
- Differentiating these small steps results in large values in the response.
- Then encoder-based measurement is noisy.
- One of the way to remove some of the high-frequency components is adding a low-pass filter (LPF) to the derivative output.

- Add a Transfer Function block after the derivative output and connect LPF to the scope.
- From the Simulink Library / Simulink / Continuous.
- Set the Transfer Function block to 50/(s + 50).
- Run the Simulation.
- Show the filtered encoder-based speed response and the motor voltage.
- The filtered response is a less noisy.
- Cut off frequency of the LPF, 50/(s + 50), ωf = 50 rad/s, or f = 50/(2π) = 7.96 Hz.
- Vary the cut off frequency, ωf between 10 to 200 rad/s (or 1.6 to 32 Hz).
- Lowering the cut off frequency, remove more noise from the signal but causes it to slow down.
- At higher cut off frequency, allows for more high-frequency components (i.e., noise) , but the signal has less delay.

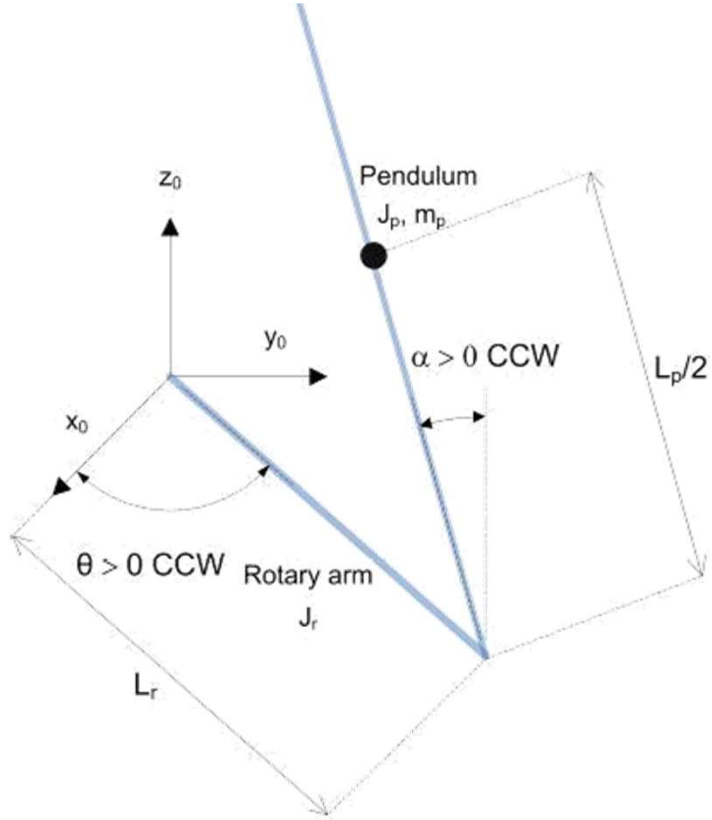
Observations and results

- Measured servo speed using encoder: **The output signal was obtained without using filter, we observed that noise is very high**
- The waveform measured before the derivative block was found to be **continuous**, hence, we conclude that noise is introduced in the waveform after the derivative block.
- The filtered encoder-based speed response has **improved**
- The cut off frequency of the low-pass filter 50/(s + 50) is **50 rad/s., 7.957Hz**
- If Vary the cut off frequency w_f , between 10 to 200 rad/s: **At lower cut off frequencies the response is cleaner and improved as the noise of higher frequency has been removed. At higher cut off frequencies the response is very unclear due to a lot of noise.**

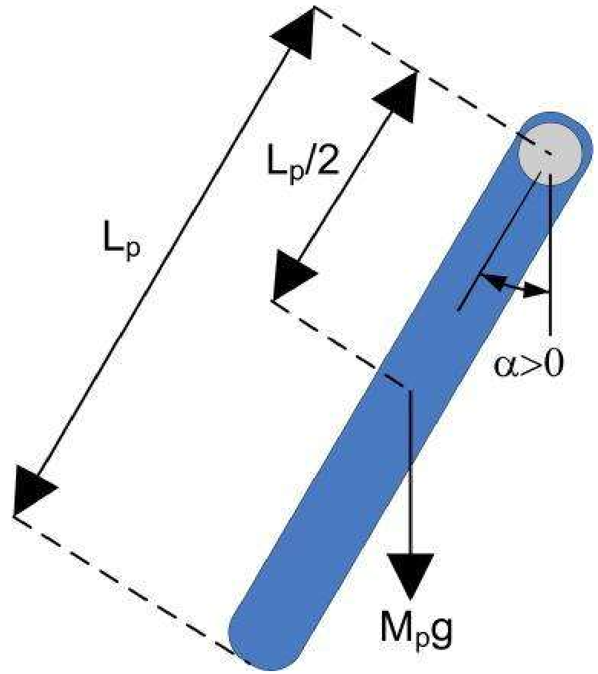
In-Lab Exercises: PENDULUM MOMENT OF INERTIA

QUBE-Servo with pendulum module

Rotary inverted pendulum conventions



- Finding moment of inertia analytically and experimentally.
- Before starting this lab make sure: QUBE-Servo Integration Lab and
- Rotary pendulum module is attached to the QUBE.
- The free-body diagram of the QUBE-Servo pendulum:



- From the free-body diagram of the pendulum, the resulting nonlinear equation of motion of the pendulum is: $J_p \ddot{\alpha}(t) = M_p\, g\, l_p \sin \alpha(t)$
- where Jp is the moment of inertia of the pendulum at the pivot axis,
- Mp is the total mass of the pendulum.
- Lp is the length of the pendulum (from pivot to end).
- The centre of mass position is at Lp/2.
- The moment of inertia of the pendulum can be found experimentally.

$$J_p = \frac{M_p\, g\, l_p}{(2\pi f)^2}$$

- where f is the measured frequency of the pendulum from the scope:
- The frequency is calculated using: $f = \frac{n_{cyc}}{\Delta t}$

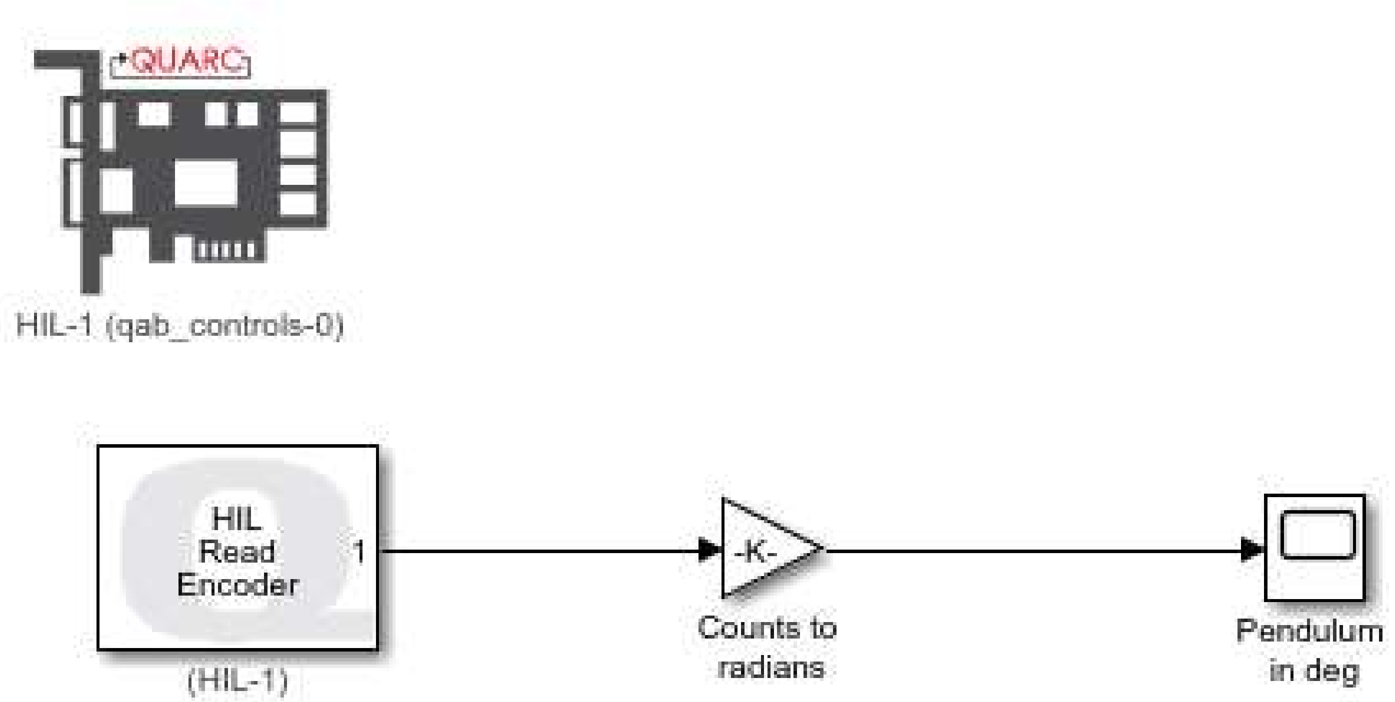
- where n_{cyc} is the number of cycles and Δt is the duration of these cycles.
- Alternatively, Jp can be calculated using the moment of inertia expression:

$$J = \int r^2 dm$$

- where r is the perpendicular distance between the element mass, dm and the axis of rotation.
- Rotary pendulum details:

Rotary Pendulum Module		
m_r	Rotary arm mass	0.095 kg
L_r	Rotary arm length (pivot to end of metal rod)	0.085 m
m_p	Pendulum link mass	0.024 kg
L_p	Pendulum link length	0.129 m
Motor and Pendulum Encoders		
	Encoder line count	512 lines/rev
	Encoder line count in quadrature	2048 lines/rev
	Encoder resolution (in quadrature)	0.176 deg/count

In-Lab Exercises: Design a Simulink diagram as shown in Figure below:



- Find the moment of inertia using Equation of the pendulum free-body diagram:

$$d_m = \frac{M_p}{L_p}\, dr \qquad J_p = \int_0^{L_p} r^2 dm = \frac{M_p}{L_p}\int_0^{L_p} r^2 dr = \frac{1}{3} M_p L_p^2$$

- The moment of inertia, Jp measured using the formula is **1.33128*10^{-4} kg-m^2**
- After performing the experiment, frequency and moment of inertia of the pendulum using the observed results: **1.53846 hz. and 1.625*10^{-4}kg-m^2**
- The moment of inertia found analytically and experimentally are reasonably close.
- The discrepancy may be due to an inaccuracy when measuring the pendulum frequency or the fact that this frequency is the damped frequency (not the undamped natural frequency that the equation to compute the inertia uses).

Thank you