PID Tuning

10 - Step Process

*** with Rate-Control Example ***

Step–by–step recipe for success

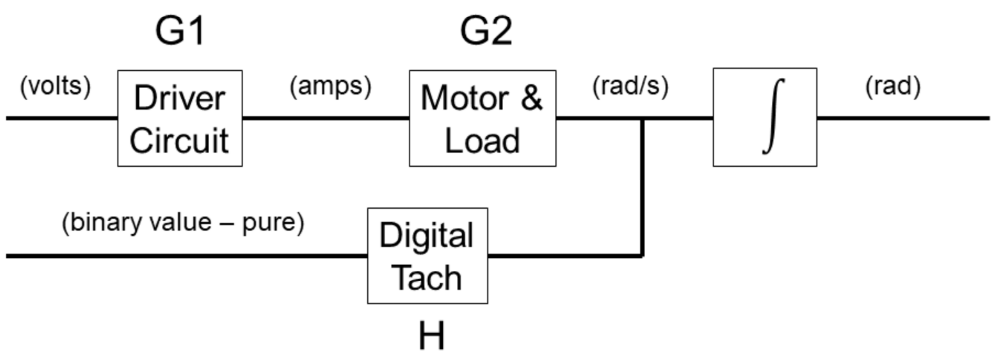
Each step demonstrated by example

- Current Driver Circuit (1^{st} order system)
- Motor & Load (2^{nd} order system)
- Optical Encoder with discretization error
- Typical u-controller with average clock speed

Prof. Leo Stocco
Electrical & Computer Engineering
UBC

Step 1: System ID & Linearize

$$\frac{K_{mech}}{Ms^2 + Bs + K} \qquad \frac{K_{elec}}{Cs^2 + {}^{s}/_{R} + {}^{1}/_{L}}$$



Break system into sub-systems

ID all signals

Develop model of each sub-system

- Most elec / mech systems can be APPROXIMATED as 2^{nd} order system + gain
- Calculate Xfer Functions “G1, G2, …” from Data Sheets (**DS**) & known values

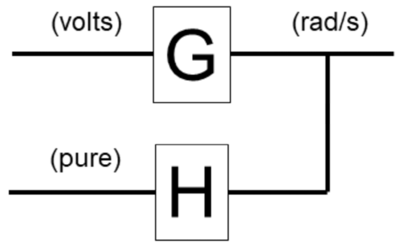
If no data sheet, record step response at reasonable OPERATING POINT and APPROXIMATE:

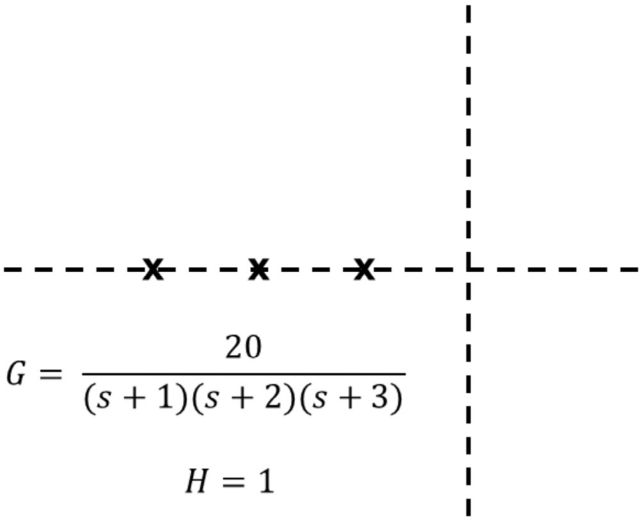
- 0-order linear approximation
- 1^{st} or 2^{nd} order approximation
- Additional (3^{rd}) poles if shape “unusual”

<u>Hint</u>

- Linear sub-system modeled by constant (gain)
- DELAY modeled by POLE
- Neglect non-linearities (discontinuities, noise, etc.)

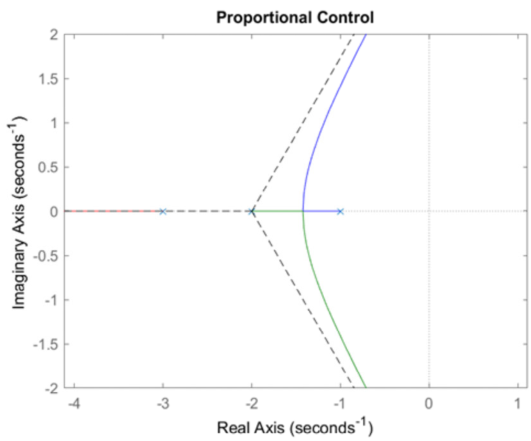
Step 2: System P&Z





$$G = \frac{20}{(s+1)(s+2)(s+3)}$$

$$H = 1$$

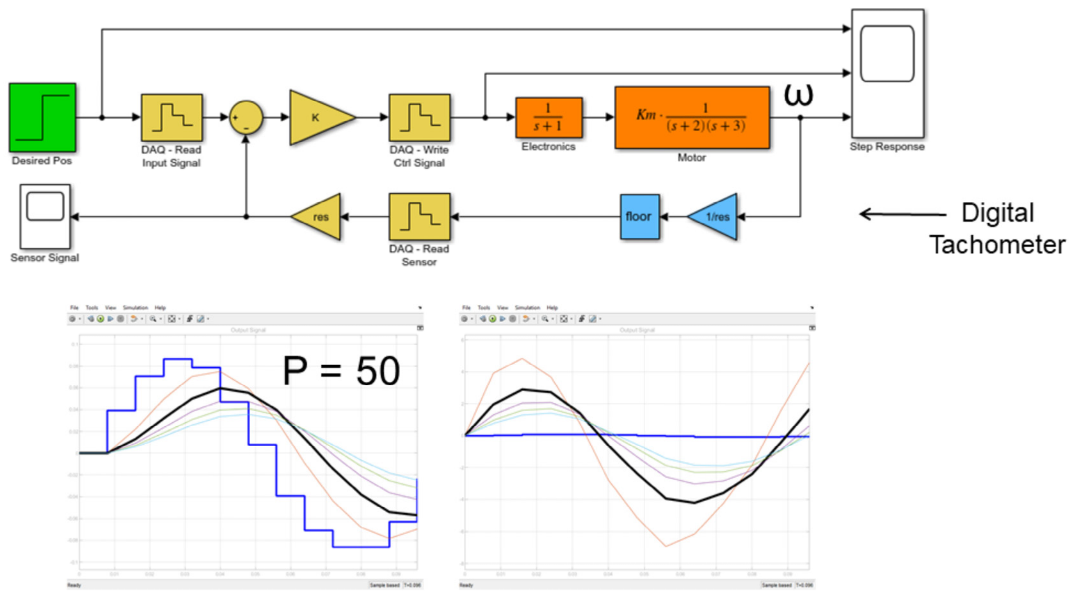


Combine & compute Loop Gain (GH)

- Identify Poles & Zeros

Plot Root-Locus

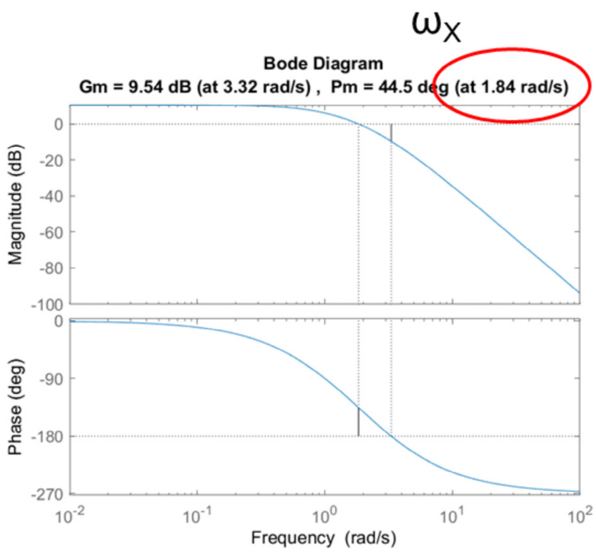
Step 3: Model System & Design Filter

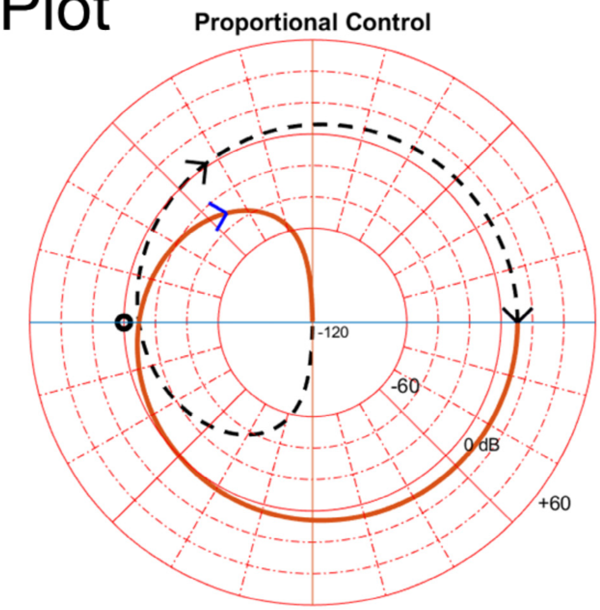


Design Filter

- Generate sensor noise model
- Optimize P (filter pole location)

Step 4: System Nyquist Plot





Generate Bode & Nyquist plot

- “margin” function shows Gm, Pm & X-over frequencies

Stable System:

- Positive margins
- Gm X-Over Freq > Pm X-Over Freq

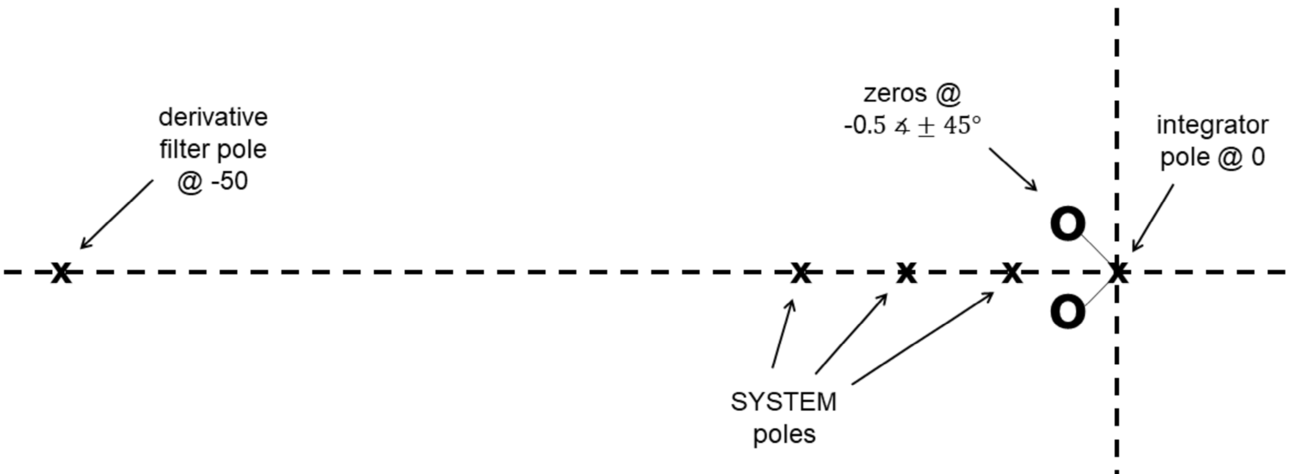
Unstable System:

- Negative margins
- Gm X-Over Freq < Pm X-Over Freq

Choose SMALLER X-Over Freq

- Sytem is stable
- wx = Pm X-Over Freq = 1.84 rad/s

Step 5: Controller P&Z



$$KG = K\frac{(s+Z_1)(s+Z_2)}{s(s+P)}\;\frac{20}{(s+1)(s+2)(s+3)}$$

Add controller poles & zeros to Root-Locus

<u>Controller Poles</u>

- 1 Pole @ 0 (integrator)
- 1 Pole @ P (filter pole)

<u>Controller Zeros</u>

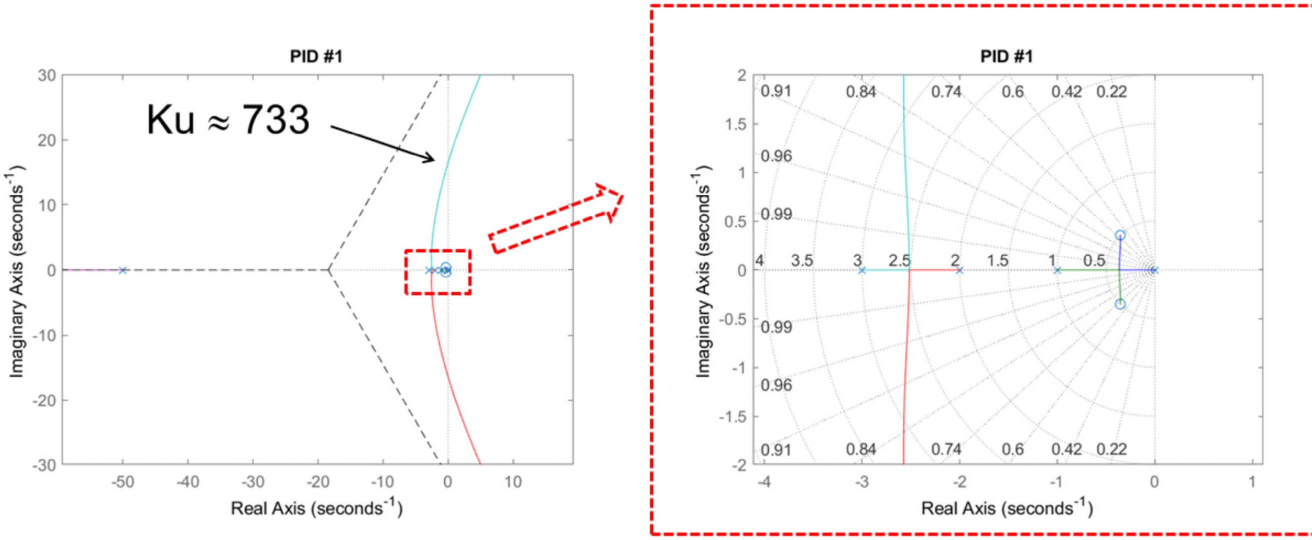
- 2x zeros
    - Mag = ½ the magnitude of right-most non-zero pole
    - Ang = 45 deg (reasonable value)

<u>PID Controller</u>

- Zeros must trap dominant poles
- Send less dominant poles along asymptotes

- Stabilize by reducing zero Mag or Increasing zero Ang
- Effectively shifts asymptote centre to left

Controller Root Locus

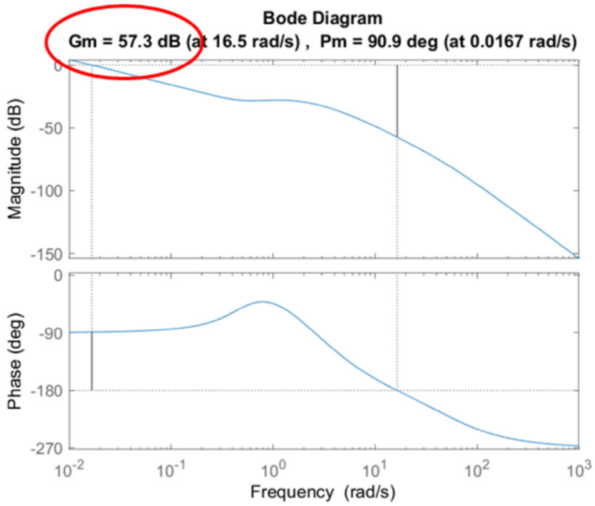


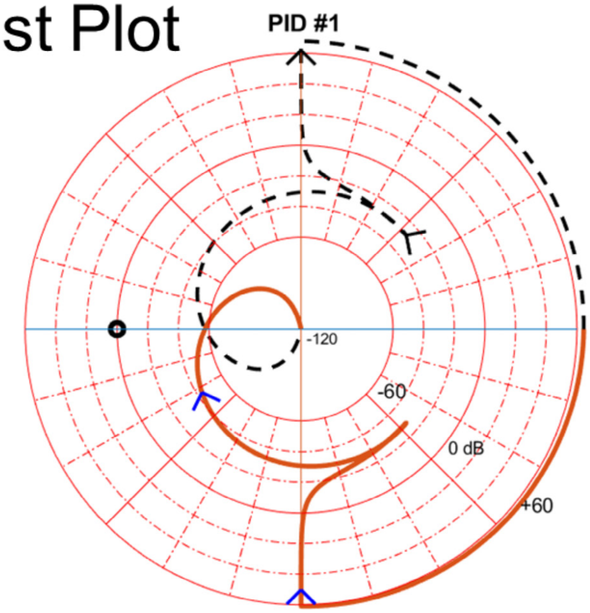
Zoom in on Dominant Poles

- Filter pole is Non-Dominant (typical)
- Moves further to left as Gain increases

Step 6: Controller Nyquist Plot

K_U (in dB)





$$K_U = 10^{Gm/_{20}} = 10^{57/_{20}} = 733$$

Generate Bode & Nyquist plots

- Unity gain (K=1)
- Find Ku = Gain Margin
- Convert from dB to pure units

Step 7: Tune PID Gains

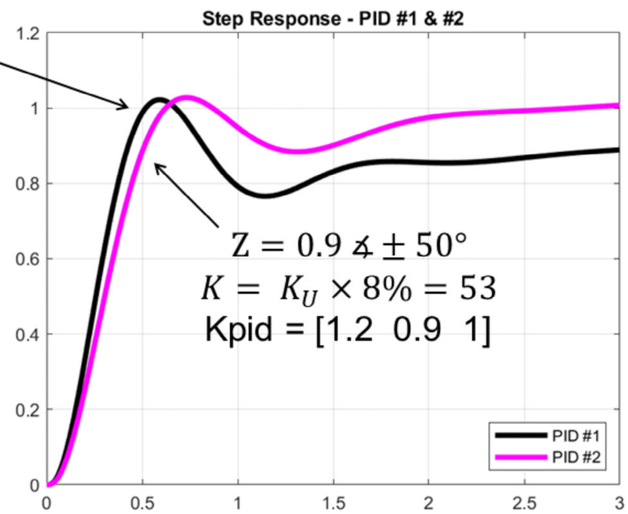
$$K = K_U \times 10\% = 73$$

$$K_I = \frac{K Z_1 Z_2}{P}$$

$$K_P = \frac{K(Z_1+Z_2) - K_I}{P}$$

$$K_D = \frac{K - K_P}{P}$$

Kpid = [1 0.4 1.4]



Select reasonable K value

- K = Ku * 10% → 50%

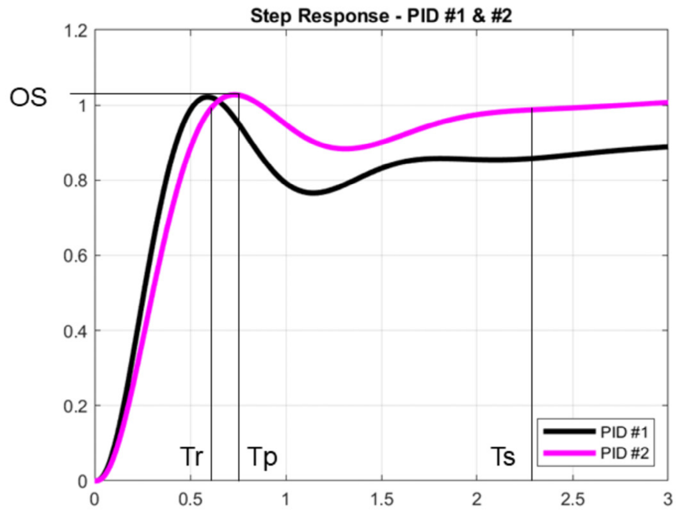
Plot step response

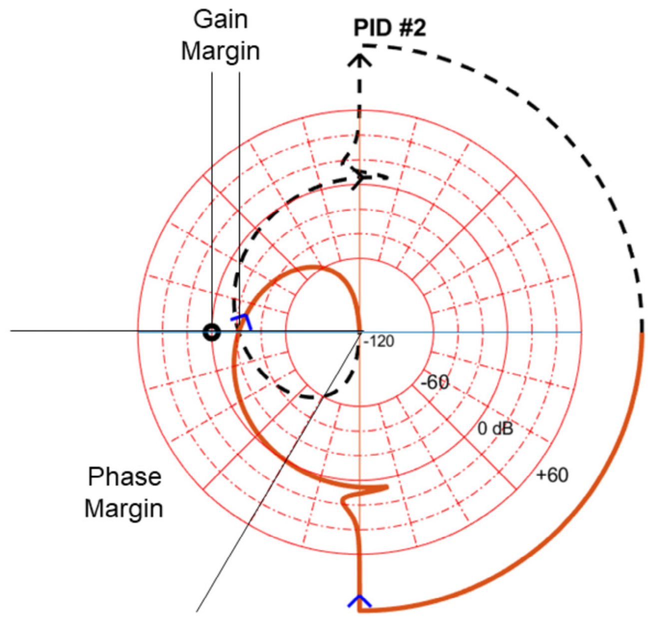
- Compute Gains Kpid = [Kp   Ki   Kd]
- Apply gains and filter to PID controller

Adjust design parameters & repeat

- Z mag = wx * 10% → 100%
- Z ang = 0 → 90 deg
- K = Ku * 2% → 80%

Step 8: Evaluate



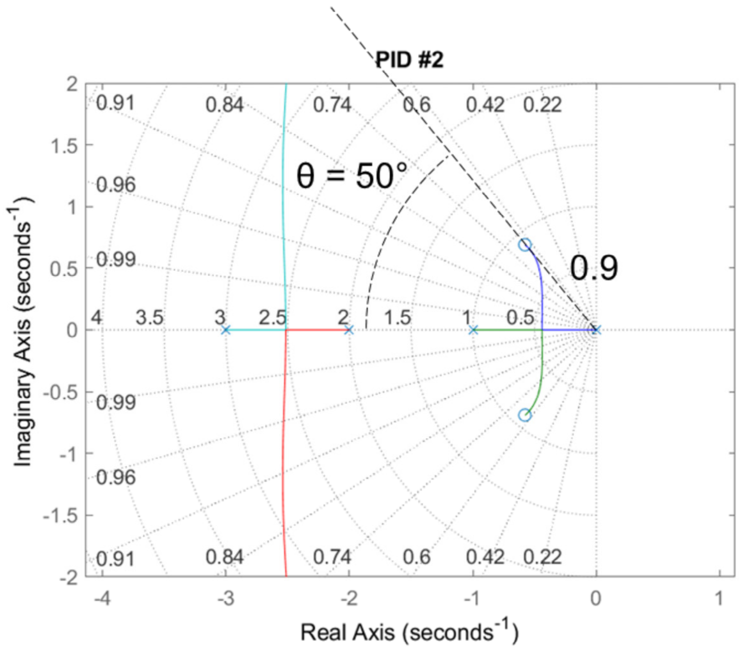


Measure

- Rise Time (Tr)
- Peat Time (Tp)
- Settle Time (Ts)
- Over-Shoot or % Over-Shoot (OS)
- Gain Margin
- Phase Margin

If RCGs not satisfied, Re-Tune

Check Work

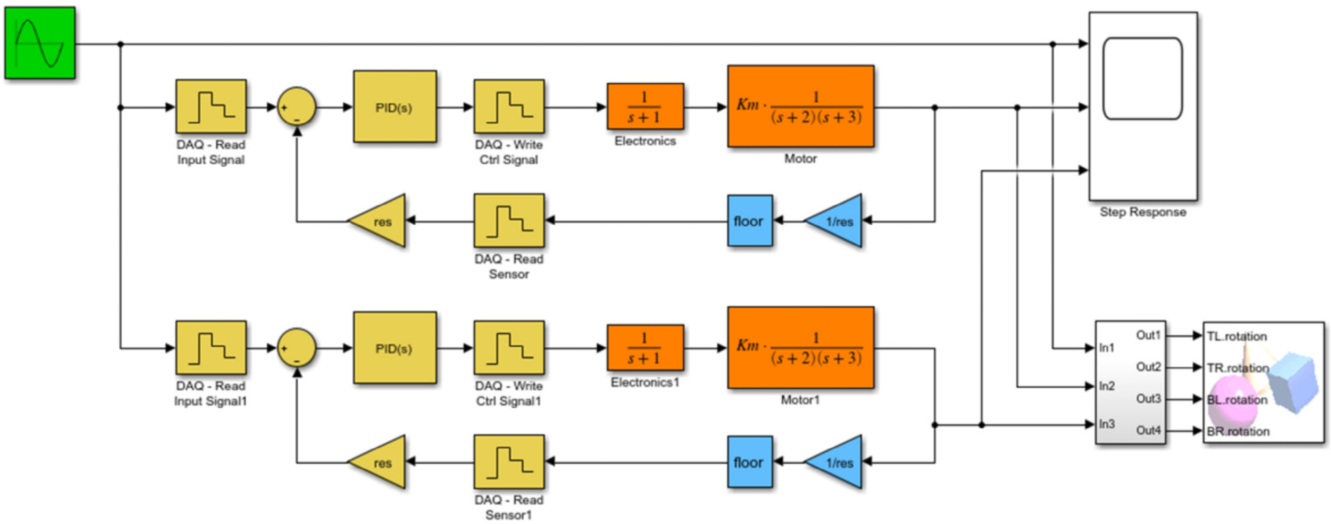


Plot Root Locus

<u>Ensure zeros are located as expected</u>

- If not, check for calculation errors

Step 9: Implement



Implement in Simulink

- Implement filters using Matlab code, not X-fer function block
- Compare results
- Any difference = Software Bug

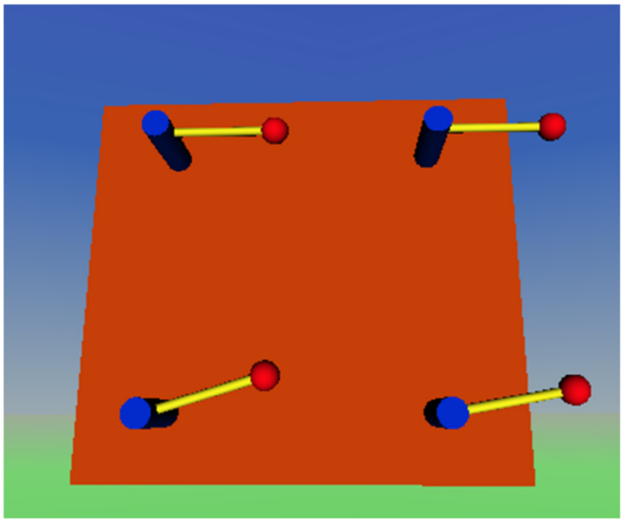
Port to Micro-Controller

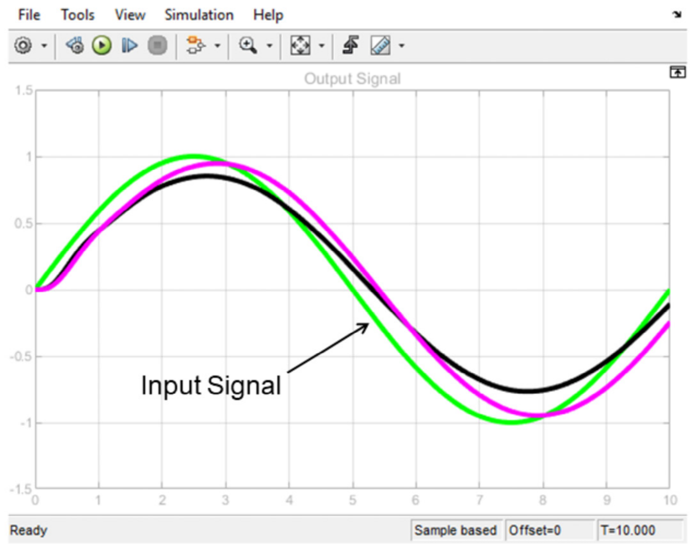
- Implement filter in Micro-Controller
- Set PD gains in Micro-Controller
- Generate & measure step response (on test bench)
- Compare to simulated results (from Step 8)

<u>If Simulation does not match Experiment</u>

- Identify software bugs
- Identify modeling errors
- Go to Step 1

Step 10: Fine-Tune





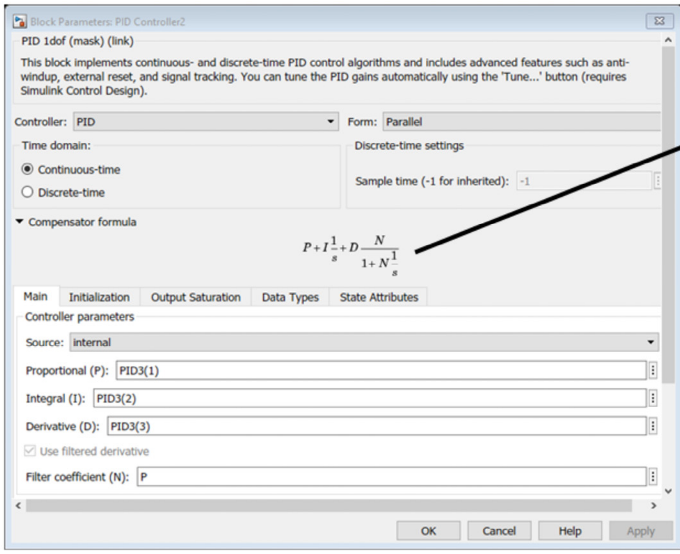
Fine-tune individual PD gains

- Account for non-linearities
- Attempt to restore theoretical response (Step 9)
- Static friction & gravity are compensated by increasing Ki

<u>Trade off:</u>

- PID #1 → greater amplitude difference
- PID #2 → greater phase shift

Notes



$$K_P + K_I \frac{1}{s} + K_D \frac{Ns}{s+N}$$

Unity Gain Filter

Gain	Stability	SS Error
K_p	↓	↓
K_i	↓	0
K_d	↑	n/a

“N” in Simulink PID block is “P” in these notes