2.004 Lab 3 Intro
P and PI Control of Velocity

Fall, 2021

Today's Tasks

MITMECHE

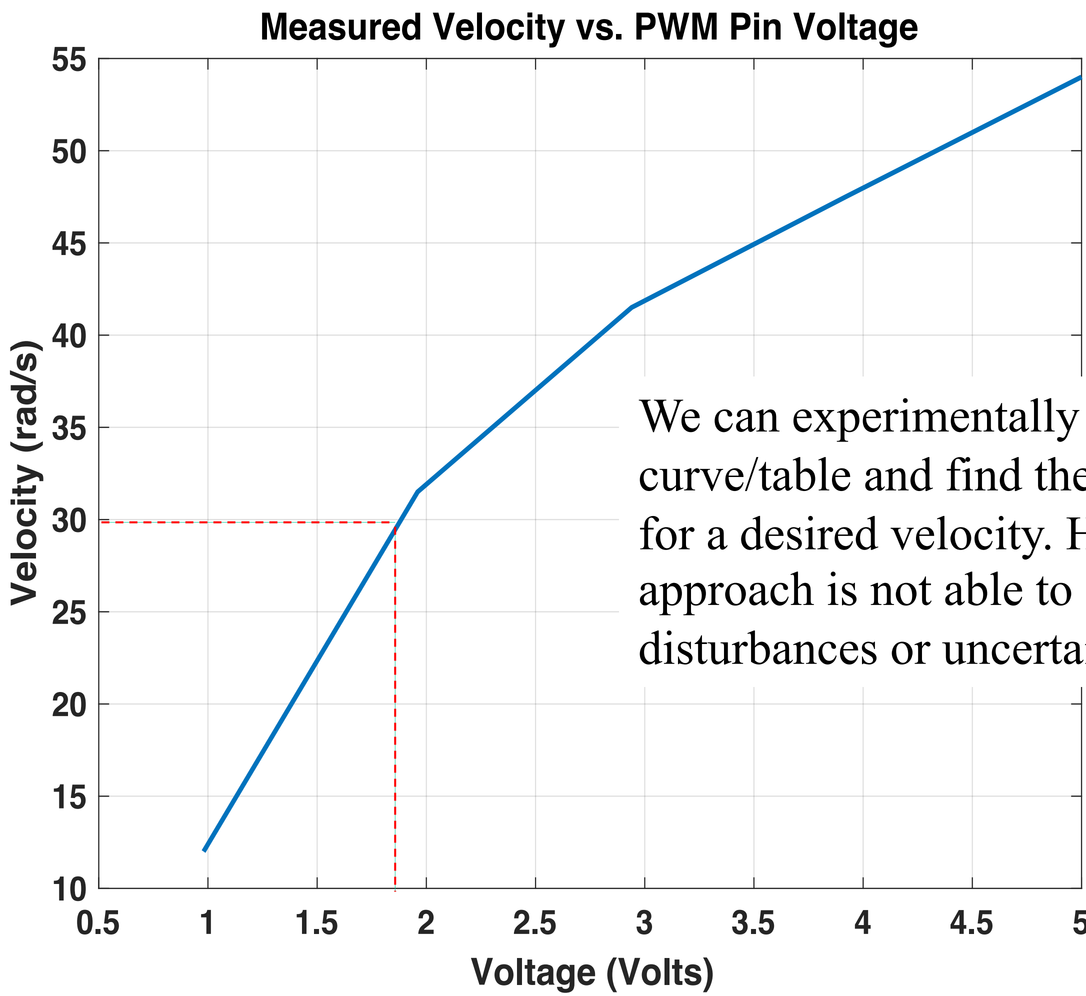
- Implement PID code.
- Closed-loop velocity control using proportional feedback, and study the effect of controller gain on transient and steady-state behavior.
- Elimination of steady-state error through the use of *integral* (I), and *proportional plus integral* (PI) control.
- Design and implement a PI controller according to given specifications.
- Extra Credit Task: Simulink simulation.

- Deliverable:
  - Lab 3 report (use the report template) to Gradescope by midnight.

Control System Comparison

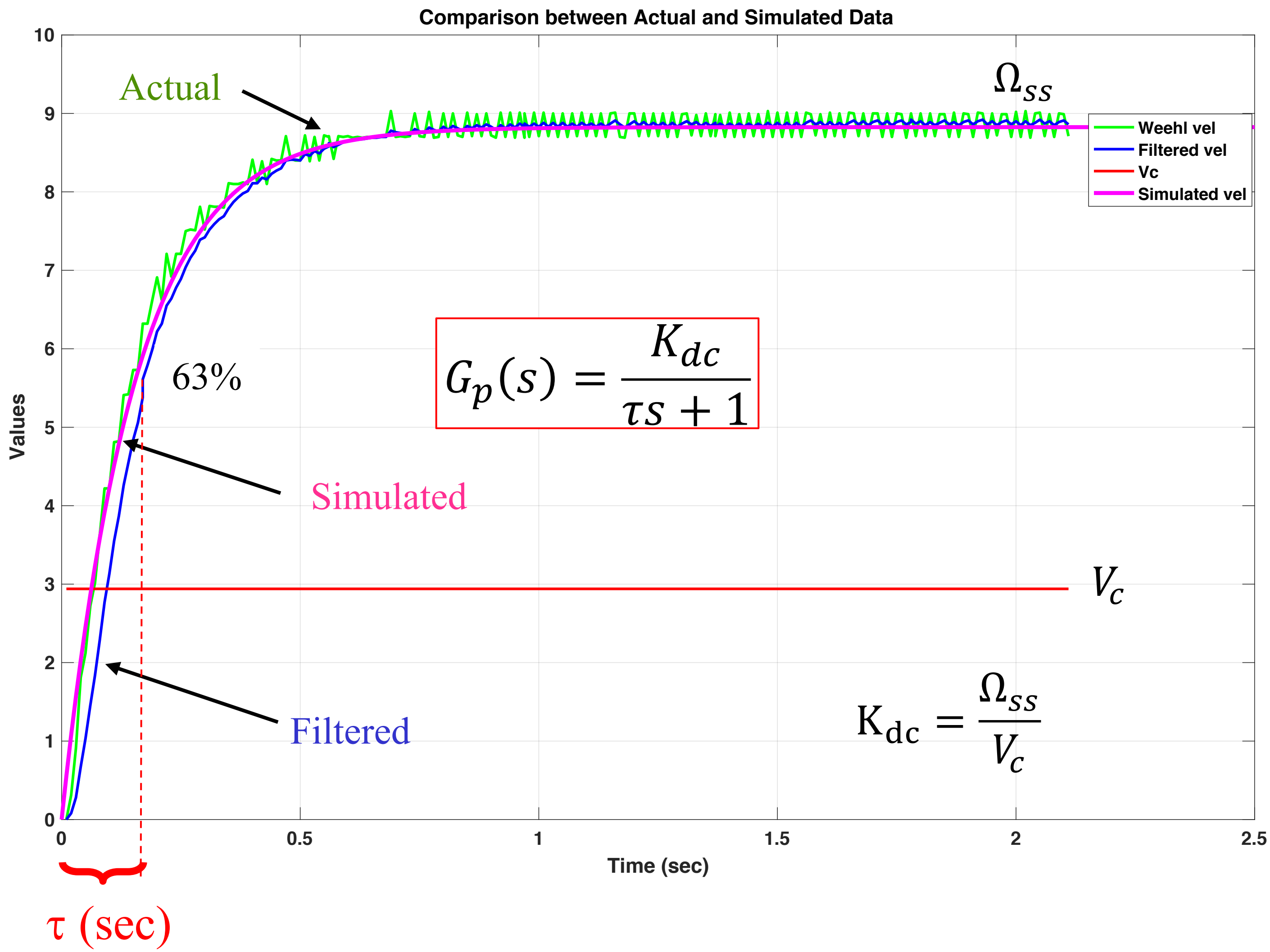
- **Open-loop:**
  - The output variables do not affect the input variables
  - The system will follow the desired reference commands if no unpredictable effects occur
  - It can compensate for disturbances that are taken into account
  - It does not change the system stability
- **Closed-loop:**
  - The output variables do affect the input variables in order to maintain a desired system behavior
  - Requires measurements (controlled variables or other variables)
  - Requires control error computed as the difference between the reference command and the controlled variable
  - Computes control input based on the control error such that the control error is minimized
  - Able to reject the effect of disturbances
  - Can make the system unstable, where the controlled variables grow without bound

Open Loop Angular Velocity as a Function of PWM

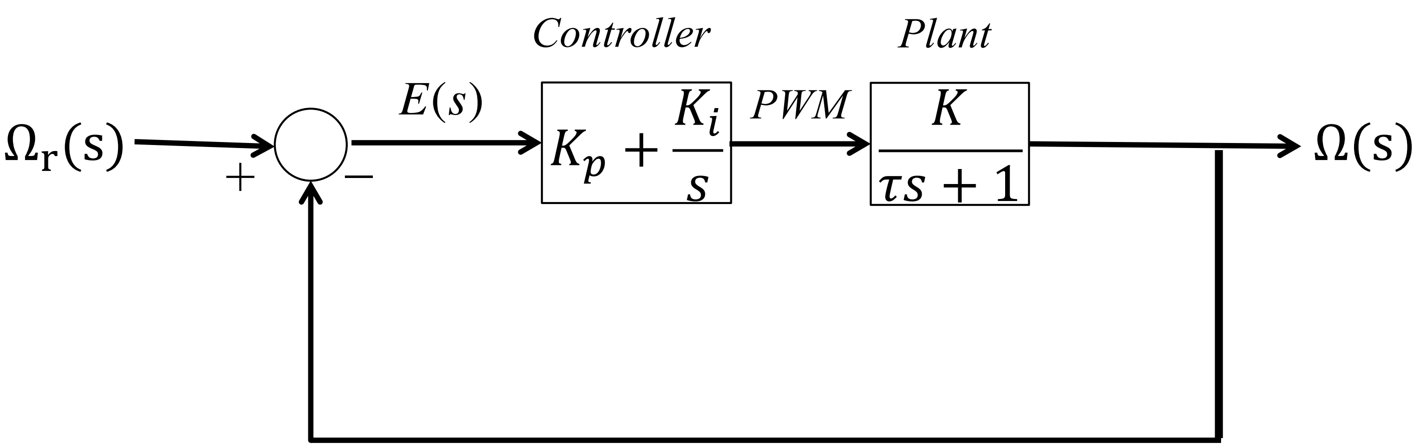


We can experimentally obtain a mapping curve/table and find the required PWM for a desired velocity. However this approach is not able to deal with external disturbances or uncertainties.

Open Loop Step Responses



Closed-Loop Block Diagram

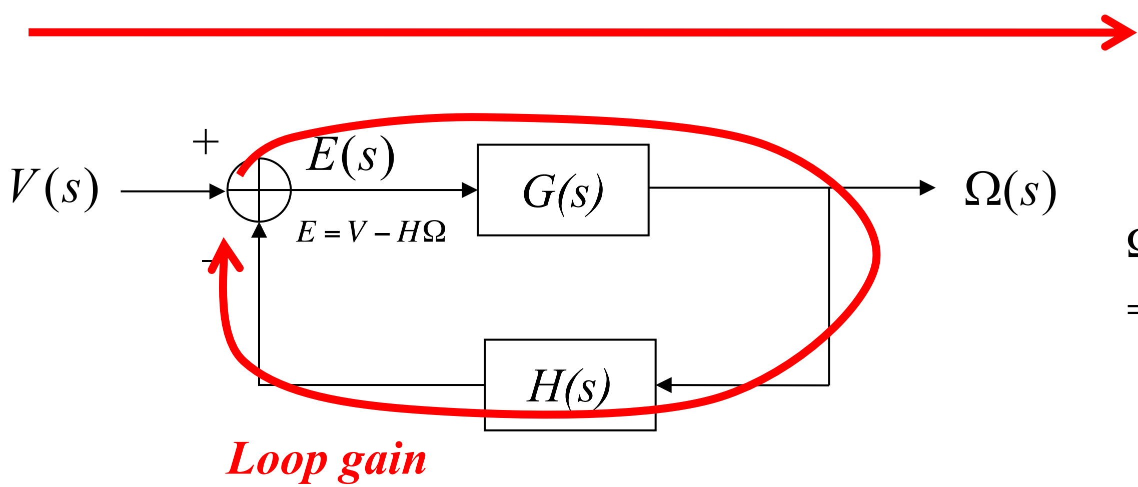


$$K = \frac{K_{dc}}{vc2pwm}$$

Voltage to PWM conversion factor: $vc2pwm = \dfrac{255\ [PWM]}{5\ [V]} = 51$

Closed-Loop Transfer Function

- Closed-loop transfer function: The gain of a <u>single-loop feedback system</u> is given by the forward gain divided by 1 plus the loop gain (for negative feedback).



Forward gain

$$\Omega = GE = G(V - H\Omega) = GV - GH\Omega$$

$$\Rightarrow \Omega + GH\Omega = (1 + GH)\Omega = GV$$

$$\boxed{\frac{\Omega(s)}{V(s)} = \frac{G(s)}{1 + G(s)H(s)} = G_{cl}(s)}$$

Closed-Loop Transfer Function: Proportional Control

$$G_{cl}(s) = \frac{\Omega(s)}{\Omega_r(s)} = \frac{K_pK}{\tau s + (1 + K_pK)} = \frac{\frac{K_pK}{(1 + K_pK)}}{\frac{\tau}{(1 + K_pK)}s + 1}$$

A new time constant

Step Response Characteristics:

Transient

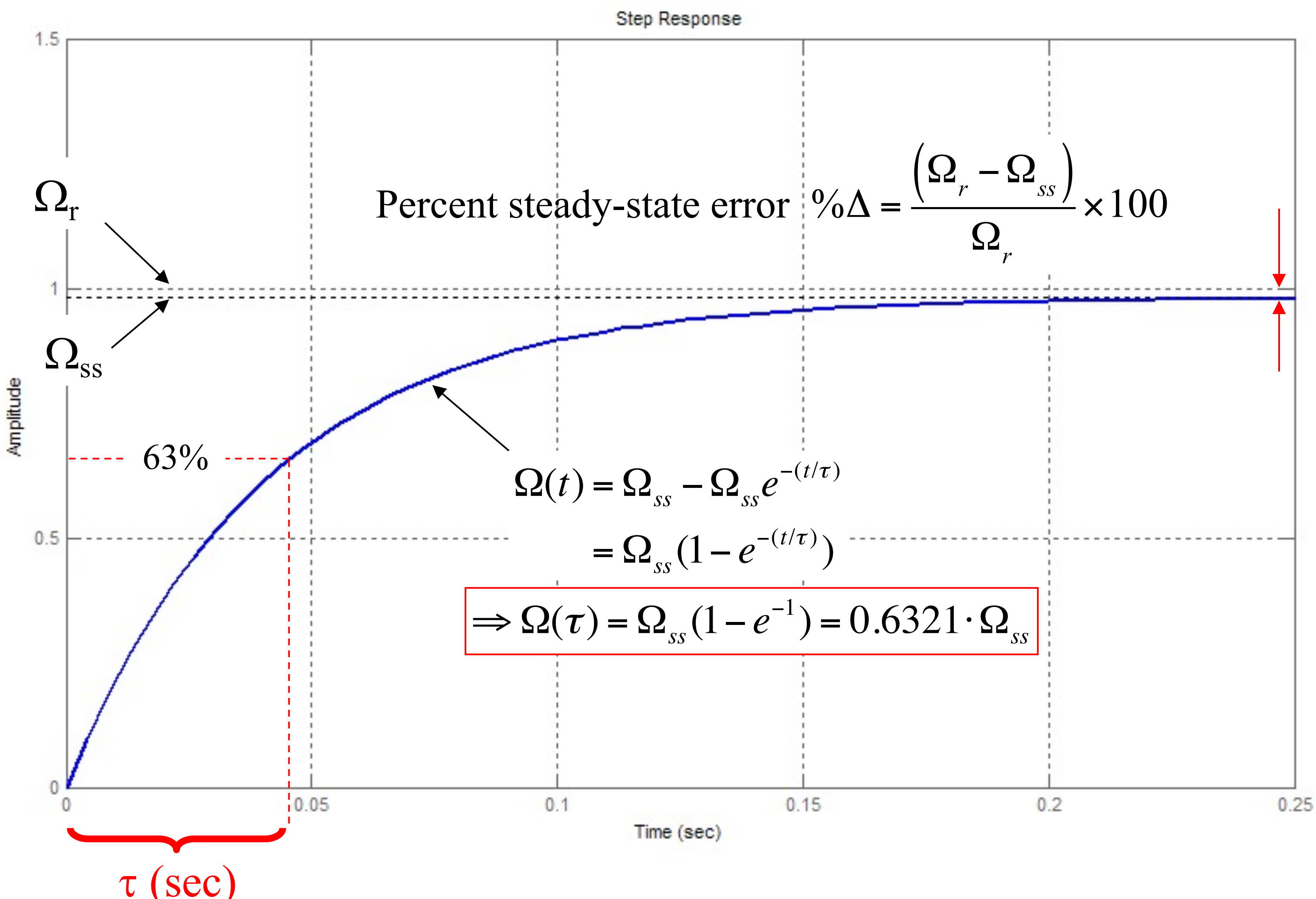
$$\tau' = \frac{\tau}{1 + K_pK}$$

Steady-state

$$\Omega_{ss} = \lim_{t\to\infty} \Omega(t) = \lim_{s\to 0} s\left(\frac{A}{s}\right) G_{cl}(s) = A\left(\frac{K_pK}{1 + K_pK}\right)$$

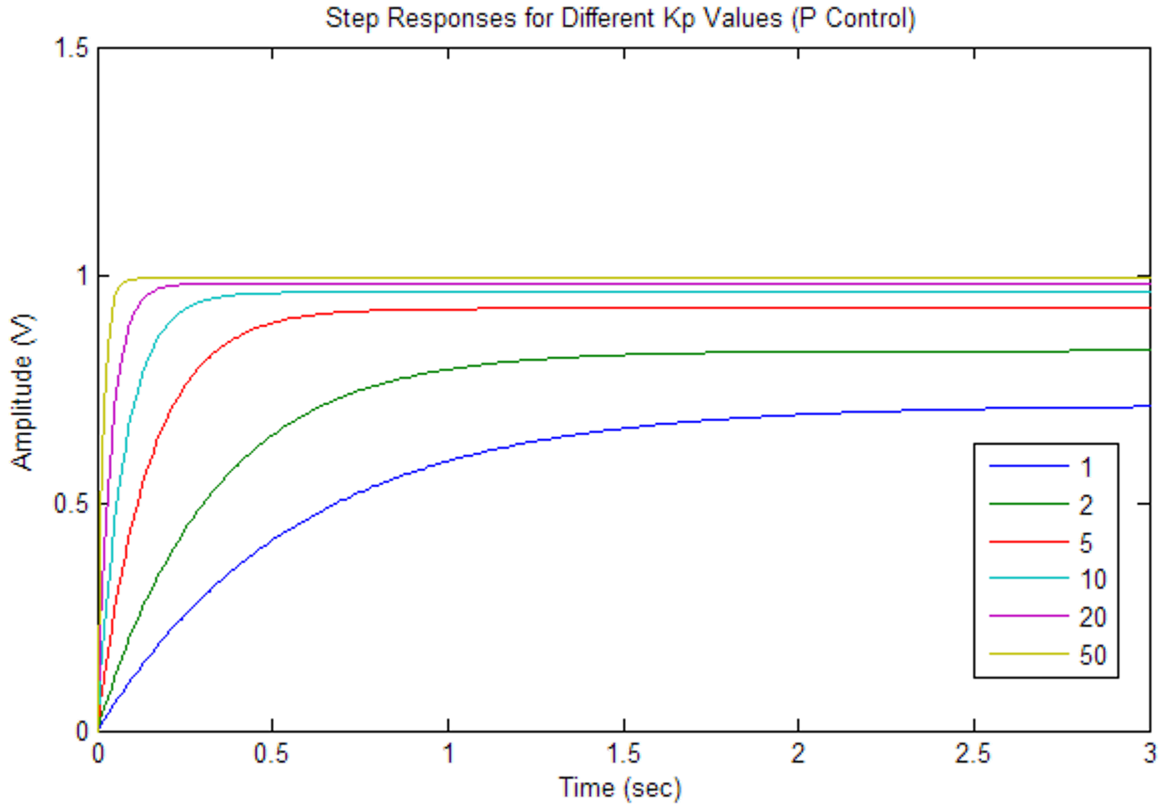
A = 1 for a unit step input

Measuring Time Constant and SS Error



Drawbacks of P Control

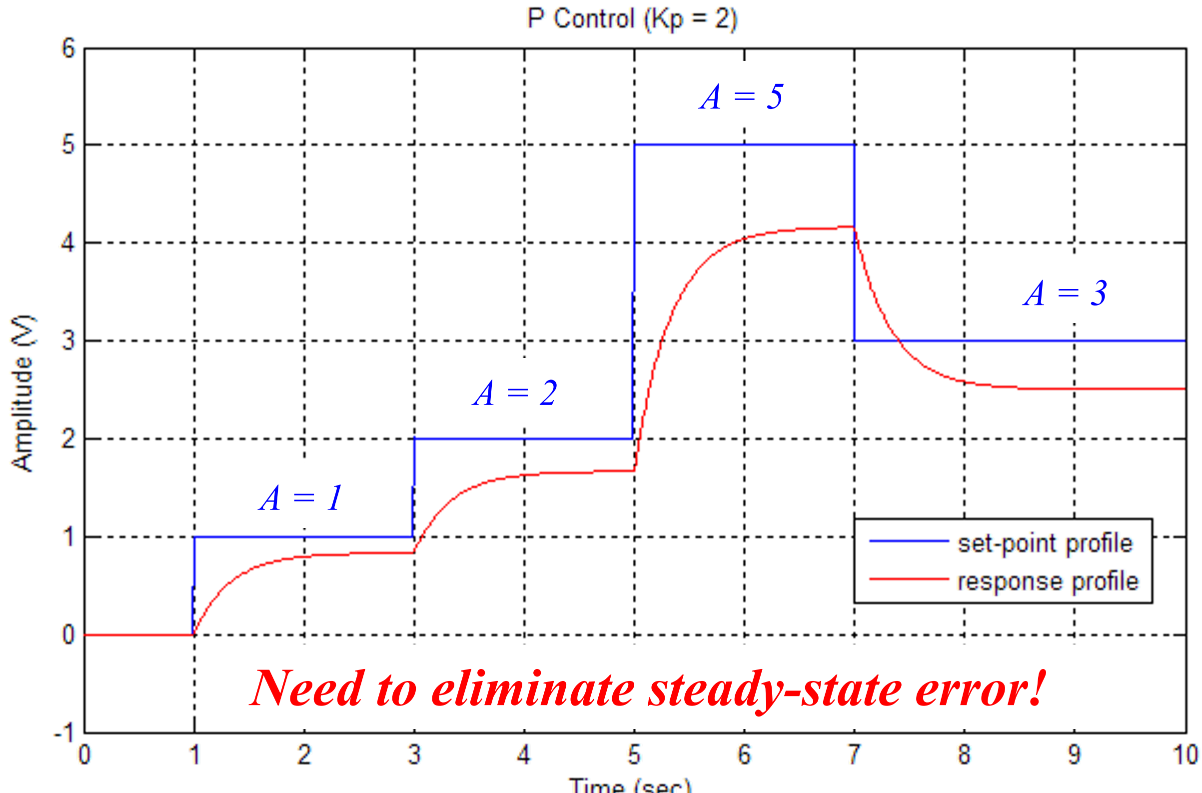
- Steady-state error
- Kp is limited by the saturation limit of the system
- Large Kp will also amplify noise and/or disturbances that may lead to instability



P Control Steady-Sate Errors

- In real world a set-point profile is often more complex than a simple step input.

$$\Omega_{ss} = \lim_{t\to\infty} \Omega(t) = \lim_{s\to 0} s\left(\frac{A}{s}\right) G_{cl}(s) = A\left(\frac{K_p K}{1 + K_p K}\right)$$



Steady-State Error and System Types

- **The servomotor plant is a "Type 0" system that will always have steady-state error with proportional control.**

$$e_{ss} = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s\left[\Omega_r(s) - G(s)\Omega_r(s)\right] = \lim_{s\to 0} s\left[1 - G(s)\right]\Omega_r(s)$$

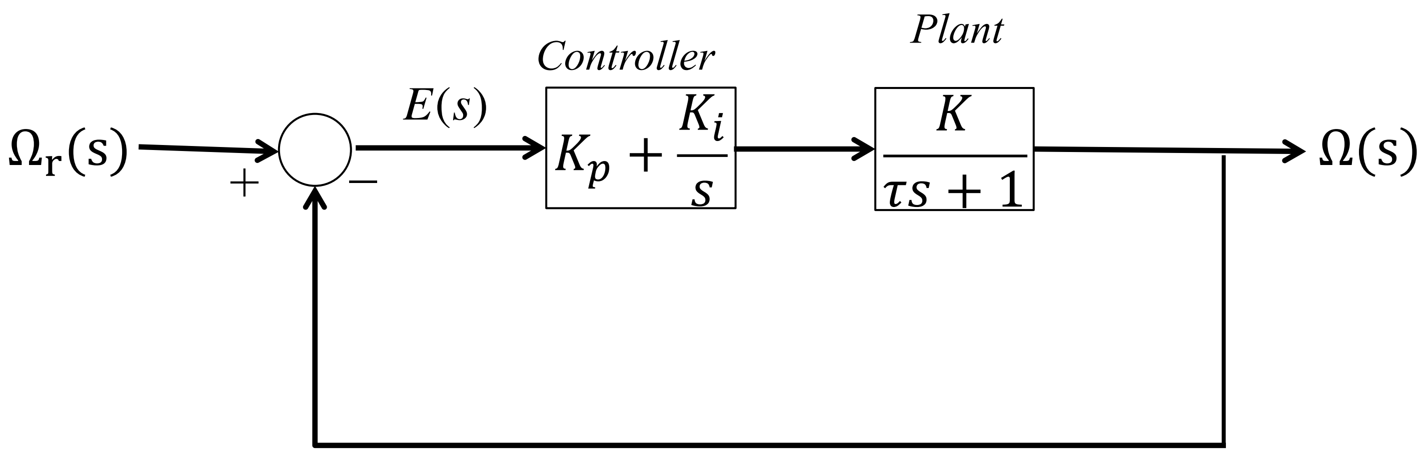
$$\text{Closed-loop P control} \quad \Longrightarrow 1 - G_{cl}(s) = 1 - \frac{K_p K}{\tau s + (1 + K_p K)} \neq 0$$

- **A system type can be determined by the number of "free" integrations in the open-loop transfer function.**

$$\boxed{G_{ol}(s) = \frac{N(s)}{s^n D(s)} \qquad \text{“Type } n\text{” system}}$$

- **Type *n* system will produce zero steady-state error for input of the same order or less. For example,**
  - Zero steady-state error for Type 1 system when given a step input.

I Control to Eliminate Steady-State Error



$$G_{cl}(s) = \frac{K(K_p s + K_i)}{\tau s^2 + (1 + K_p K)s + K_i K}$$

One zero

Two poles

Steady-state

$$\Omega_{ss} = \lim_{t \to \infty} \Omega(t) = \lim_{s \to 0} s\left(\frac{A}{s}\right) \mathrm{G_{cl}}(s) = A\left(\frac{K_i K}{K_i K}\right) = A$$

A = 1 for a unit step input

Differential Equation Analysis

P Control: $$\tau\frac{d\Omega(t)}{dt} + \Omega(t) = K_pK(\Omega_r(t) - \Omega(t))$$

At steady state $\frac{d\Omega(t)}{dt} = 0 \Rightarrow \Omega_{ss} = K_pK(\Omega_r - \Omega_{ss}) \Rightarrow \boxed{\Omega_{ss} = \frac{K_pK}{1+K_pK}\Omega_r}$

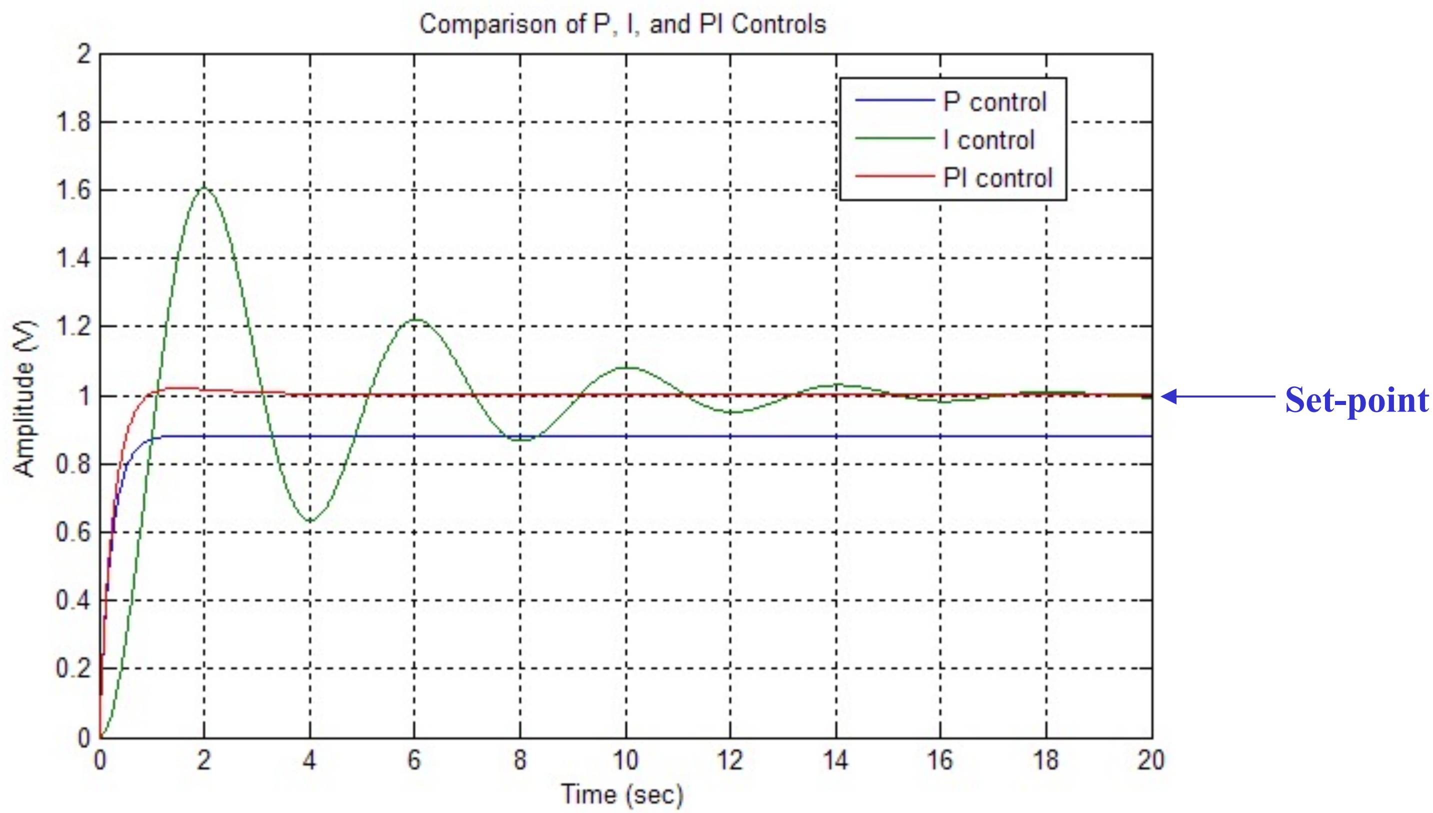
PI Control: $$\tau\frac{d\Omega(t)}{dt} + \Omega(t) = K_pK\big(\Omega_r(t) - \Omega(t)\big) + K_iK\int\big(\Omega_r(t) - \Omega(t)\big)dt$$

Take derivative of above $$\tau\frac{d^2\Omega}{dt^2} + \frac{d\Omega}{dt} = K_pK\frac{d(\Omega_r - \Omega)}{dt} + K_iK(\Omega_r - \Omega)$$

At steady state all derivatives go to zero $\Rightarrow 0 = K_iK(\Omega_r - \Omega_{ss}) \Rightarrow \boxed{\Omega_{ss} = \Omega_r}$

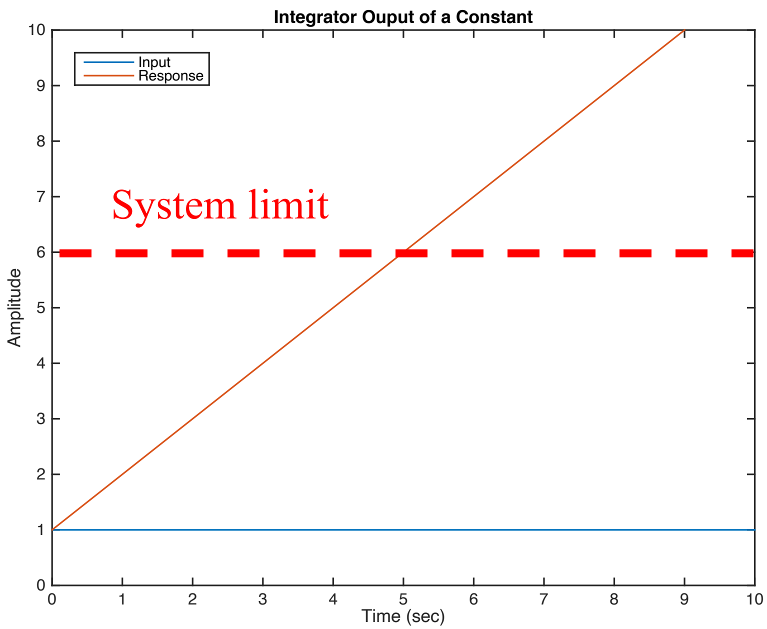
P, I, and PI Comparison

- **P Control: steady-state error**
- **I Control: overshoot, longer transient, integrator windup**



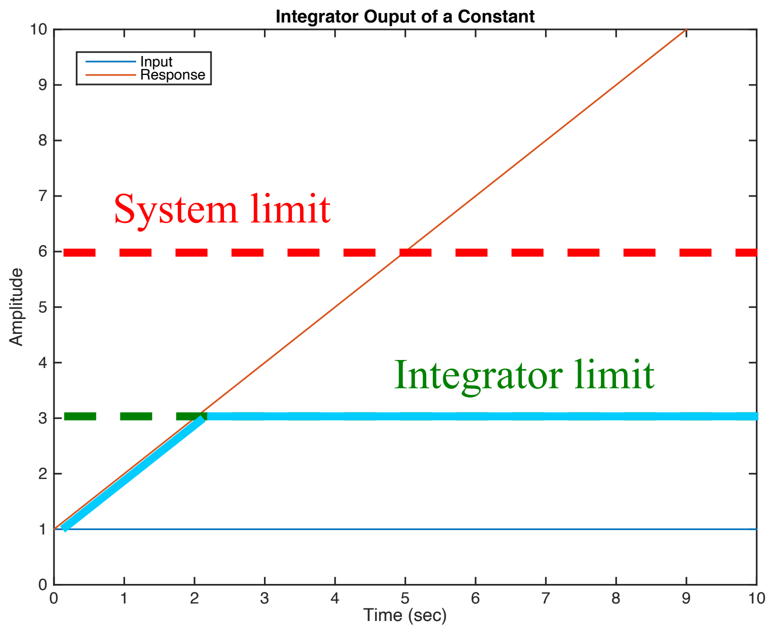
Integrator Windup

- Caused by the interaction of integral action and saturations, etc.
- Actuators have limitations such as: torque/speed of a motor, opening/closing of a valve, etc.
- When saturation occurs
  - Controller output reaches the actuator limits
  - Feedback loop is inactive
  - System operates in open loop (i.e., actuator output is fixed at its saturation value and independent of the system states/outputs)
- A controller with integral action
  - Error continues to be integrated
  - Integral term may become very large (it "winds up")
  - System may exhibit large transients when saturation occurs



Anti-Windup

- Limit set-point range so the actuator does not saturate.
  - Sets conservative bounds
  - Results in poor performance
  - Does not avoid windup caused by disturbances
- Avoid by maintaining the integral term to an appropriate value when the actuator saturates.

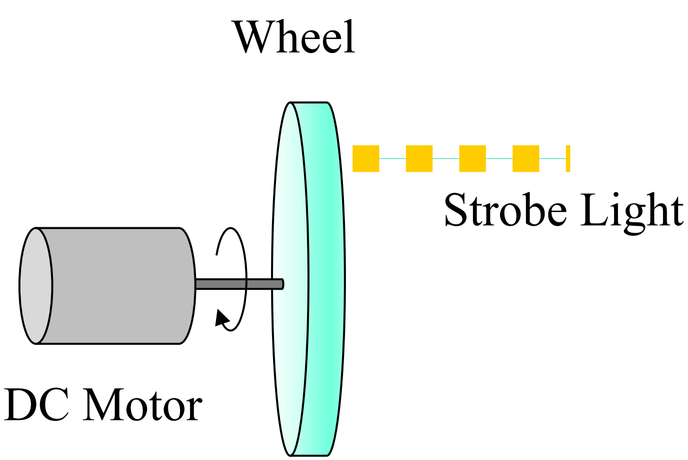


PI Controller Design

- **Design a PI controller based on given transient specifications:**
  - Critically damped
  - Natural frequency = 10 rad/s
- **Implement your controller gains on the Arduino and acquire an actual response with a reference angular velocity $\boldsymbol{\Omega_r = 15\, rad/s}$.**

Test The Controller

- Test the controller by observing the Stroboscopic effect.
- Determine the desired wheel velocity.
- Use your phone's camera app to observe the spokes of the wheel. Verify or set the frame rate to 30 frames per second (fps) in video mode.



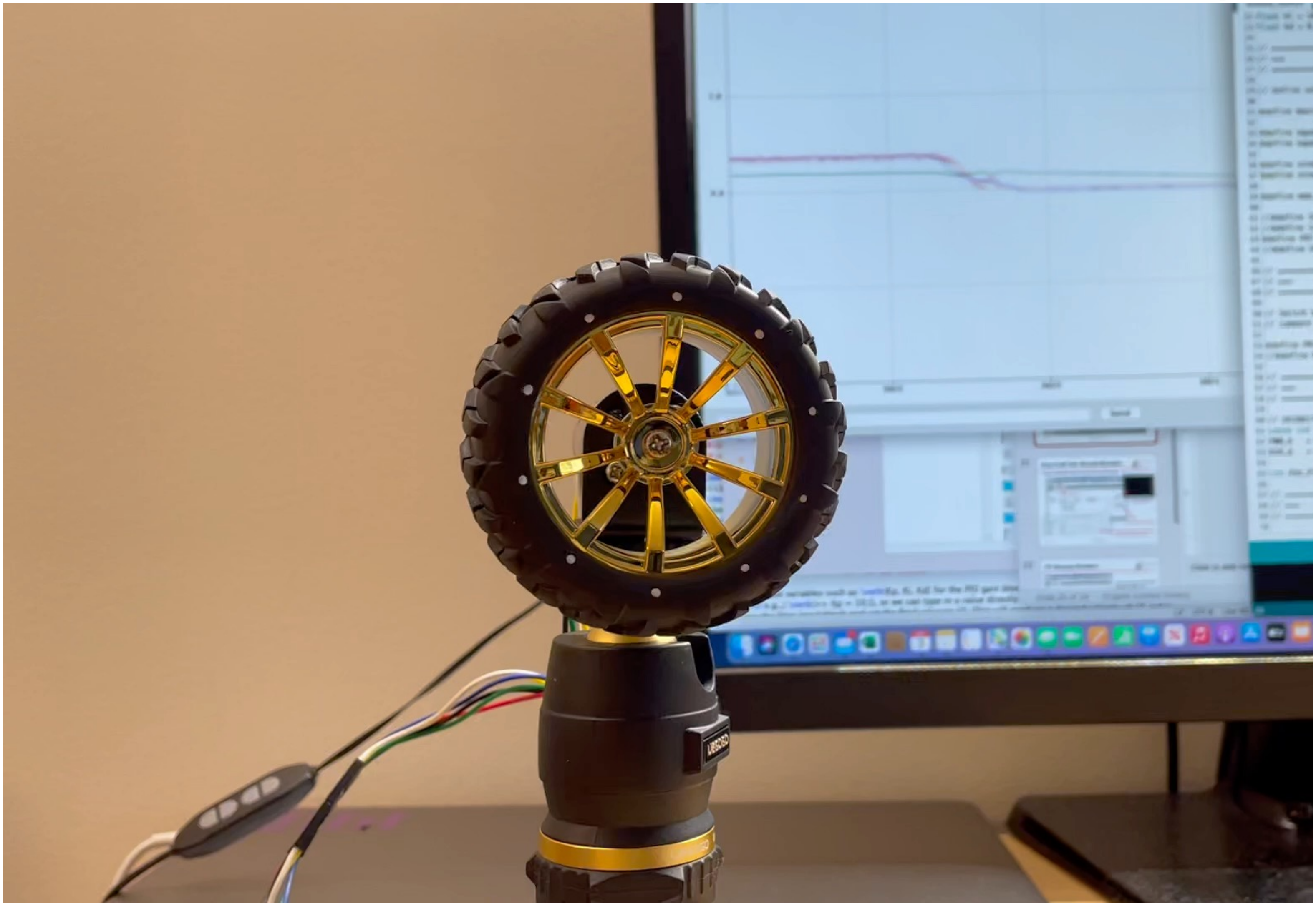
Strobe light frequency

$$\Omega_{\mathrm{r}} = \frac{2\pi f_{strobe}}{m}$$

Desired velocity

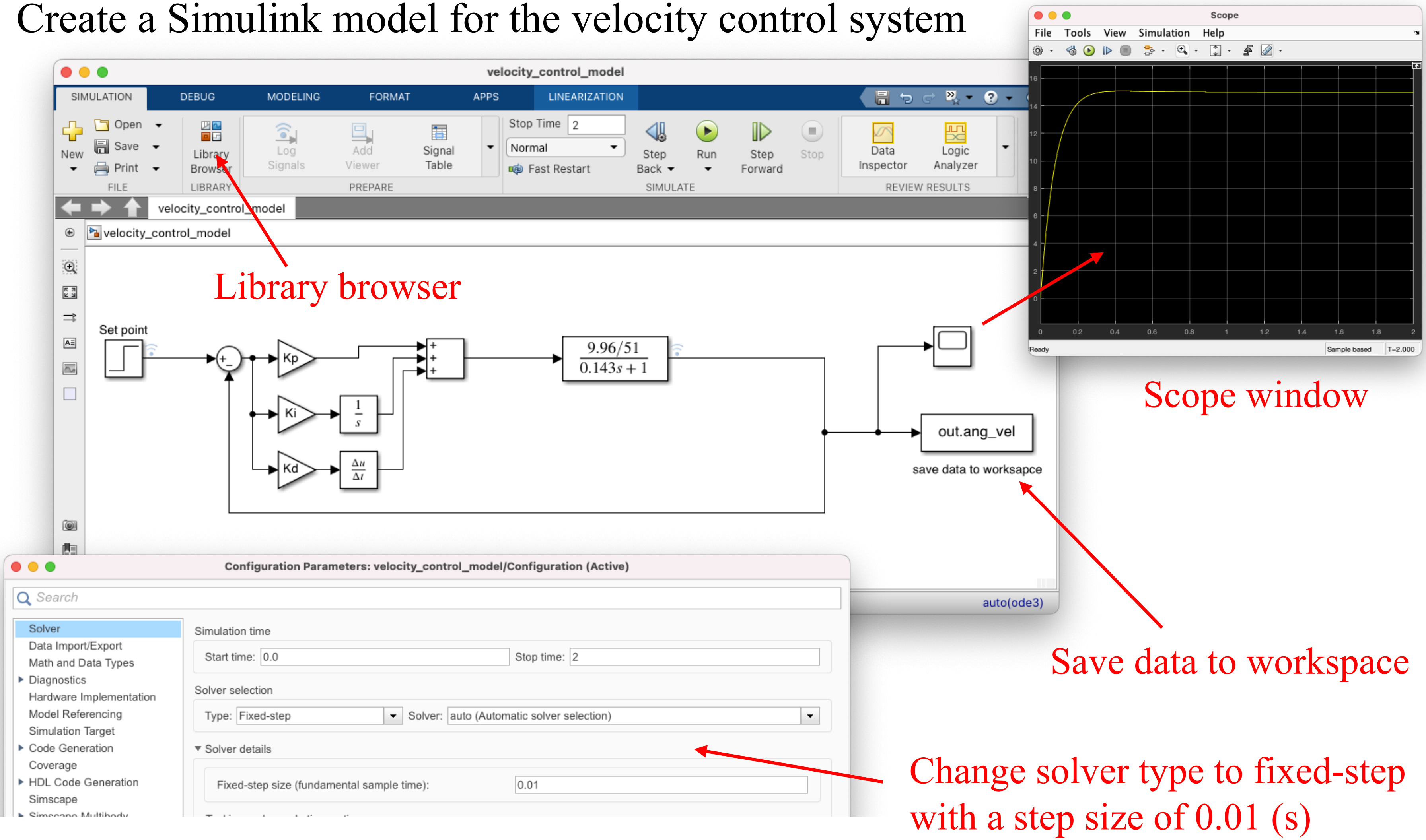
# of equally spaced patterns

Strobe Effect Example

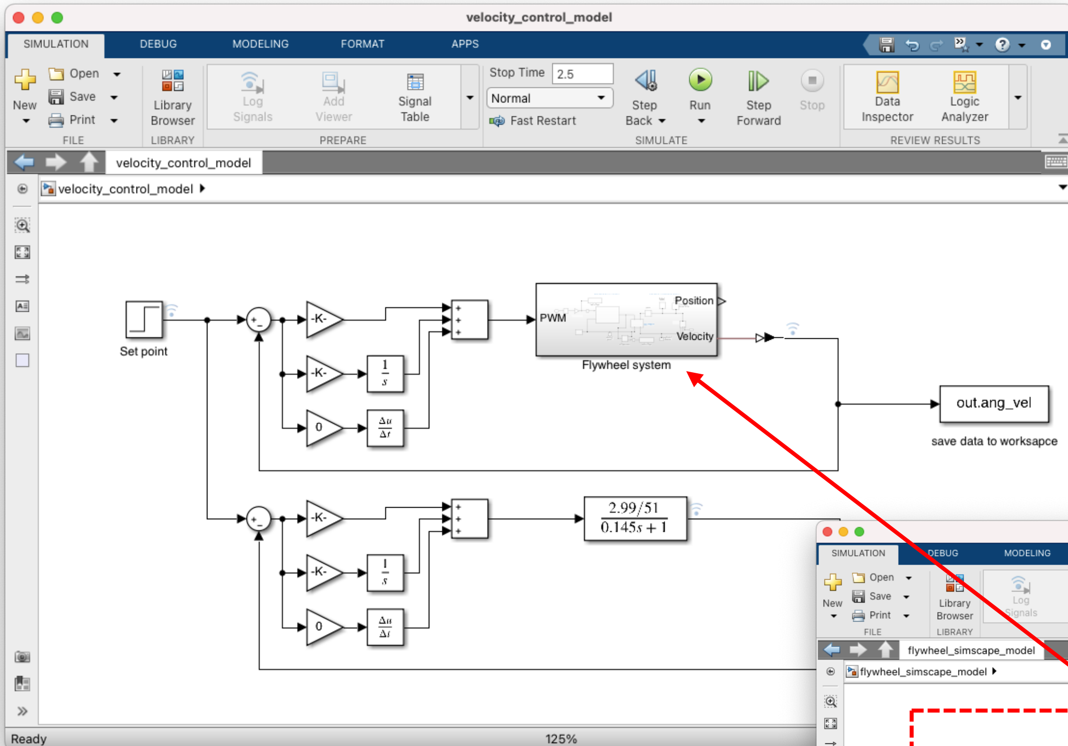


Extra Credit Task: Simulink Simulation

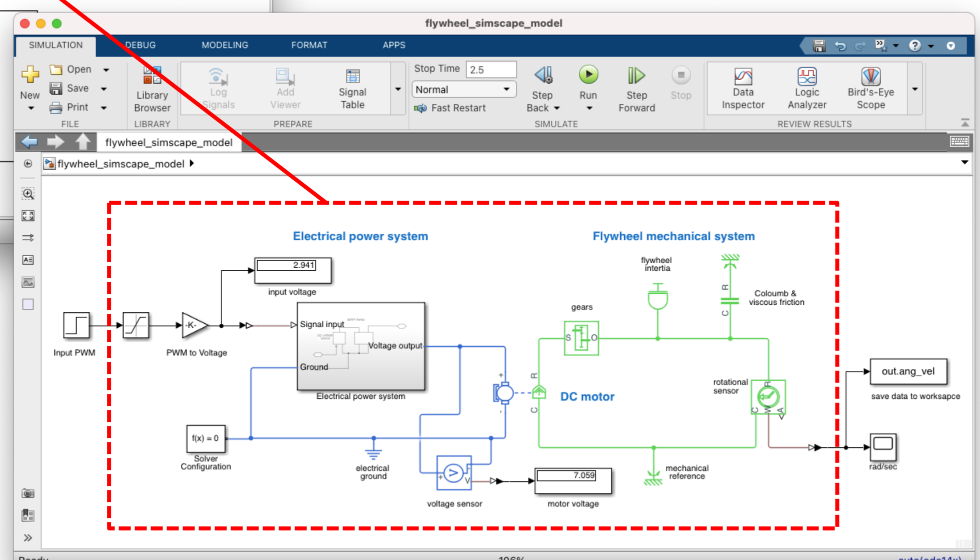
Create a Simulink model for the velocity control system



FYI: Simscape Simulation

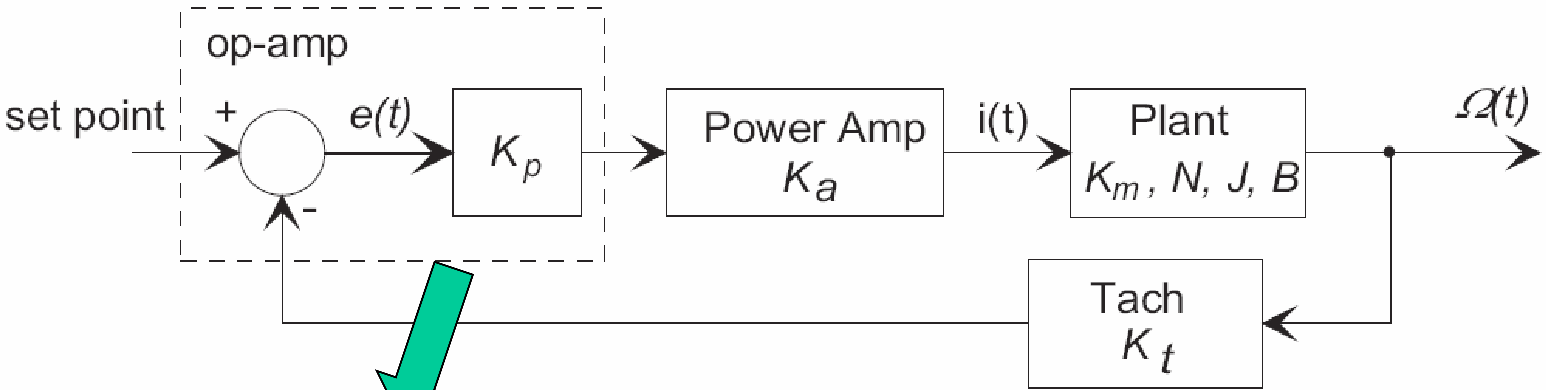


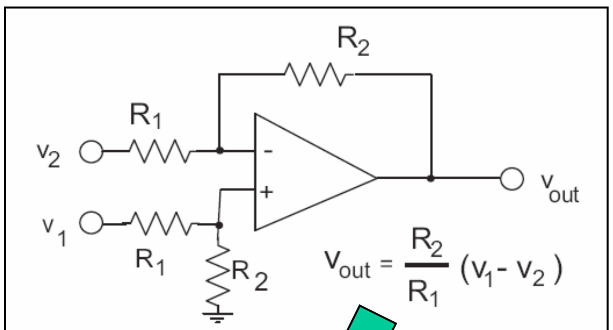
You can also use the flywheel Simscape model to do feedback control simulation.



Select the blocks in the Simscape model and make it a subsystem block.

FYI: Analog Proportional Controller



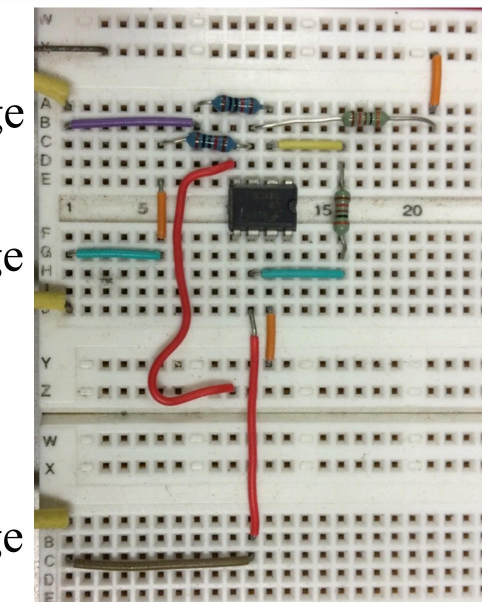


$$\boxed{K_P = \frac{R_2}{R_1}}$$

Tach voltage

Set point voltage

Controller voltage



FYI: Analog PID Controller

$$V_{out} = K_p V_{in} + K_d \frac{dV_{in}}{dt} + K_i \int_0^t V_{in} dt$$