

Proportional-Integral-Derivative (PID) Control

Course 1, Module 5, Lesson 1



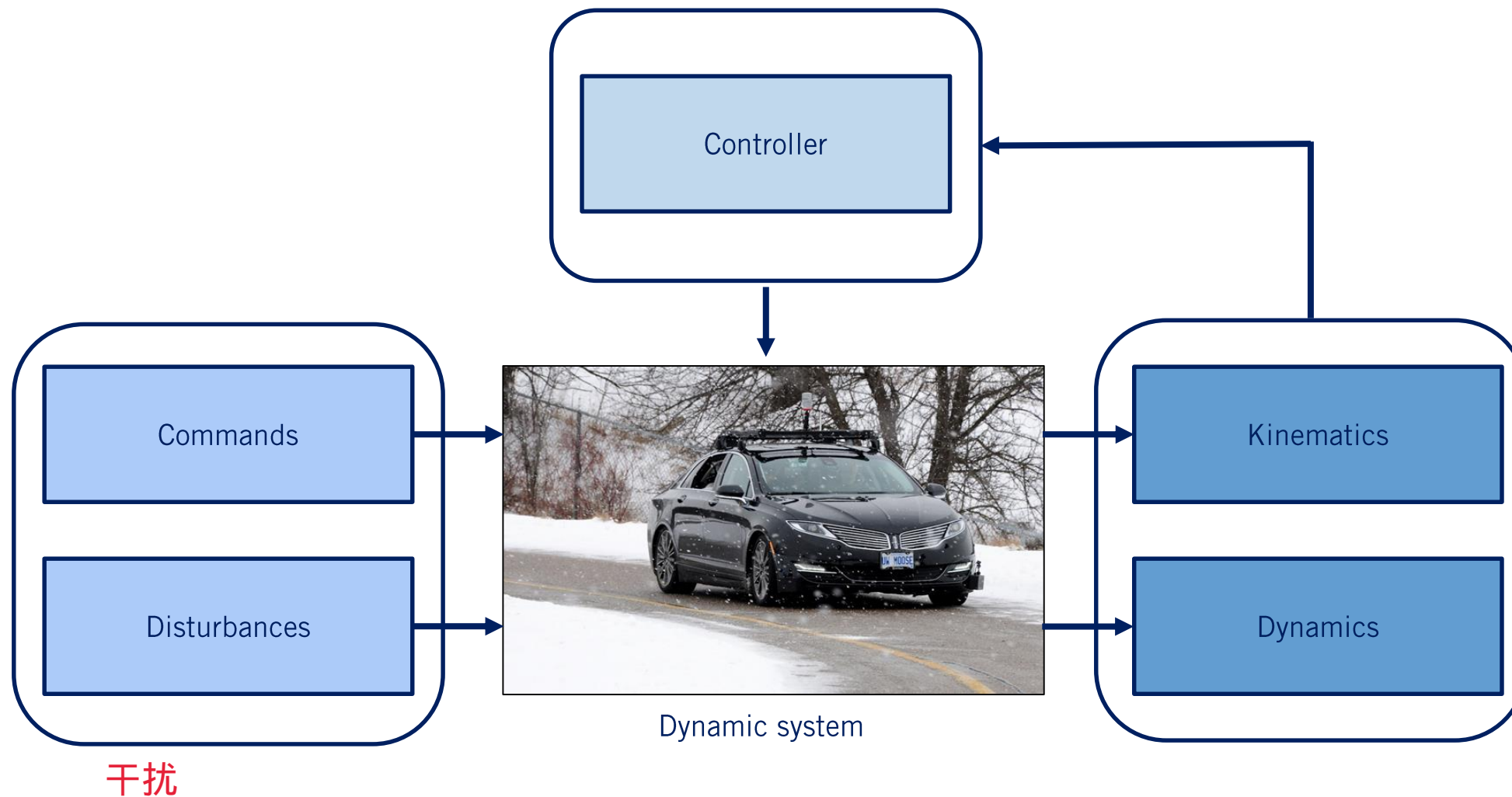
UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Learning Objectives

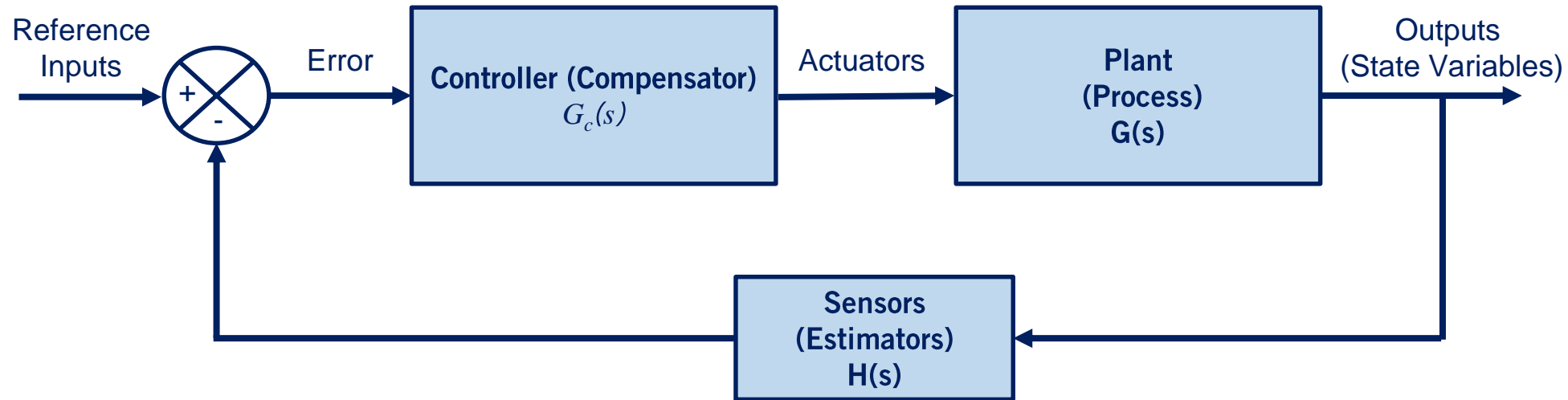
PID:比例积分微分

- In this module, you'll
 - Review basics of linear time-invariant (LTI) control 线性时不变控制
 - Develop proportional-integral-derivative (PID) controllers for longitudinal vehicle control
 - Combine feedforward and feedback control to improve speed tracking performance 前馈和反馈控制
- In this video, you'll
 - Review the basics of LTI control
 - Explore the design of PID controllers

Control Development



Typical Feedback Control Loop



Plant System or Process

- System Representation:

- The plant system could be linear or nonlinear
- Plant representation: state-space form and transfer functions
- Linear time-invariant systems can be expressed using transfer functions

- Transfer Function:

- A transfer function G is a relation between input U and output Y

$$Y(s) = G(s)U(s) \quad s = \sigma + j\omega$$

- Expressed in the Laplace domain, as a function of s , a complex variable

$$Y(s) = G(s)U(s) = \frac{N(s)}{D(s)}U(s)$$

- Zeros – roots of numerator , Poles – roots of denominator

Controller or Compensator 控制器，补偿器

- Control algorithms can vary from simple to complex
- Some simple algorithms, widely used in industry:
 - Lead-lag controllers 超前滞后控制器
 - PID controllers
- More complex algorithms
 - Nonlinear methods: Feedback linearization 反馈线性化, Backstepping 反推, Sliding mode 滑动模式
 - Optimization methods: Model predictive control 模型预测控制

PID Controller

对作用域进行变换，由时域变为Laplace域

- In the time domain:

$$u(t) = K_P e(t) + K_I \int_0^t e(t) dt + K_D \dot{e}(t)$$

where K_P , K_I , K_D are the proportional, integral and derivative gains 比例，积分和微分收益

- In the Laplace domain:

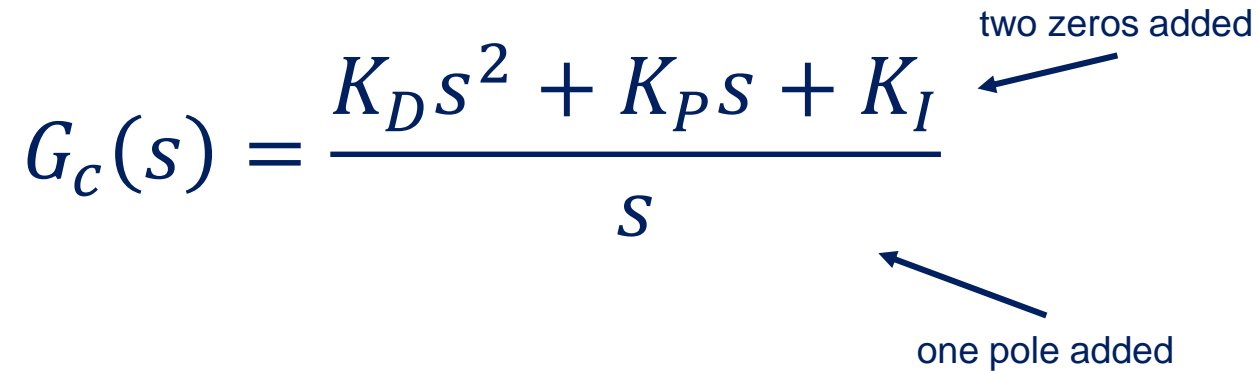
$$\begin{aligned} U(s) &= G_c(s)E(s) = \left(K_P + \frac{K_I}{s} + K_D s \right) E(s) \\ &= \left(\frac{K_D s^2 + K_P s + K_I}{s} \right) E(s) \end{aligned}$$

Proportional-Integral Derivative Controller

$$G_c(s) = \frac{K_D s^2 + K_P s + K_I}{s}$$

two zeros added

one pole added

The diagram shows the transfer function of a PID controller, G_c(s) = (K_D s^2 + K_P s + K_I) / s. An arrow points from the text 'two zeros added' to the numerator, specifically to the s^2 term. Another arrow points from the text 'one pole added' to the denominator, which is s.

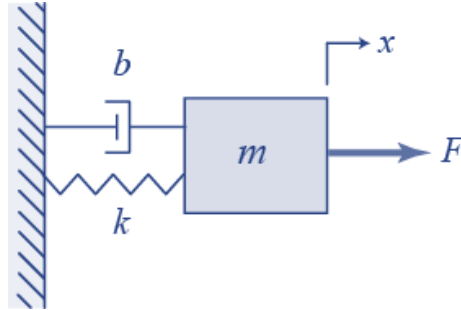
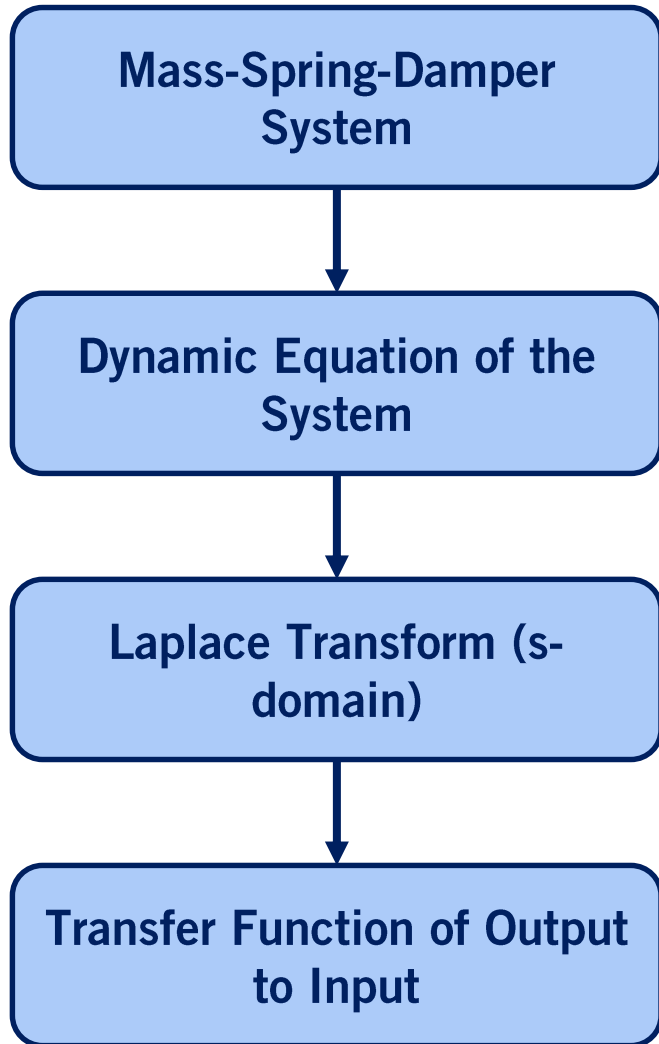
- The pole is at the origin
- The zeros can be arbitrary places on the complex plane
- PID controller design selects zero locations, by selecting P, I, and D gains
- There are several algorithms to select the PID gains (e.g. Zeigler-Nichols)

Characteristics of P, I, and D Gains

Closed Loop Response	Rise Time	Overshoot	Settling Time	Steady State Error
Increase K_P	Decrease	Increase	Small change	Decrease
Increase K_I	Decrease	Increase	Increase	Eliminate
Increase K_D	Small change	Decrease	Decrease	Small change

By properly tuning
the PID gains

Example: Second Order System



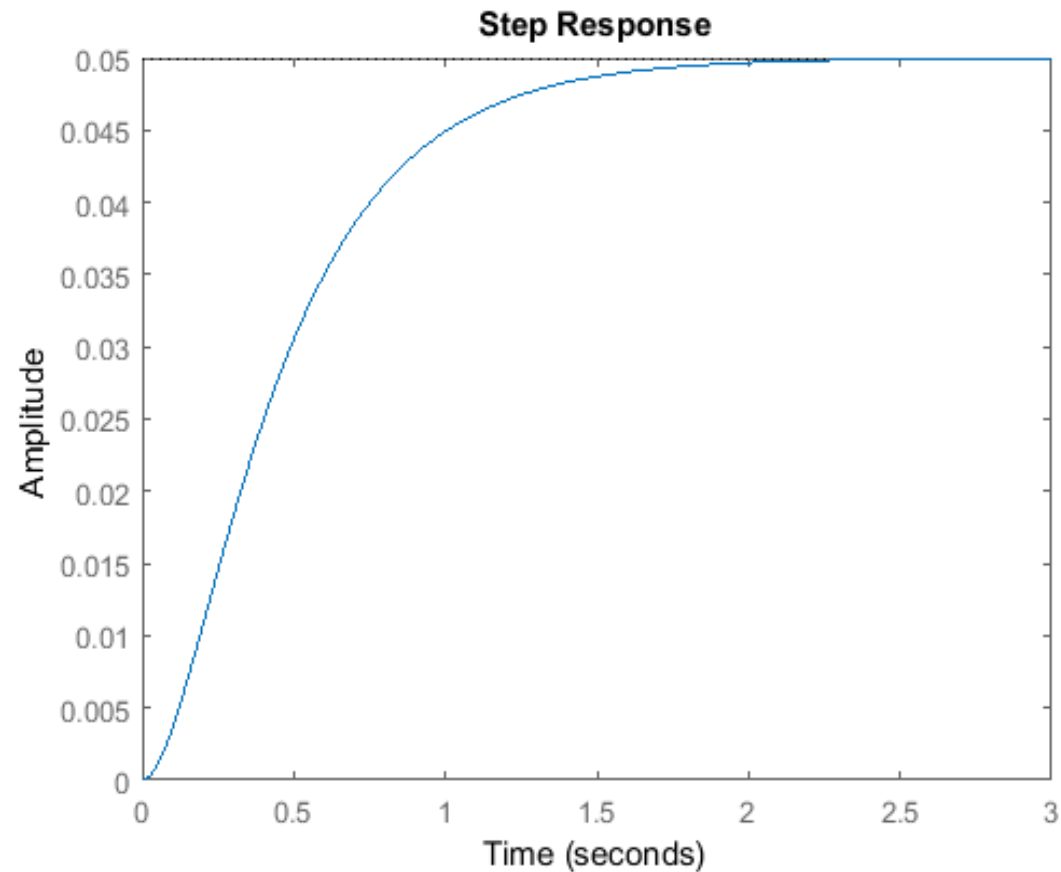
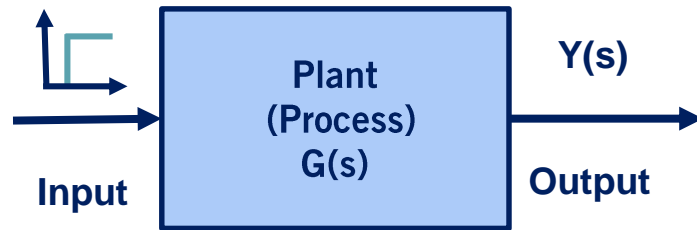
$$m\ddot{x} + b\dot{x} + kx = F, \quad x(0) = 0$$

$$ms^2X(s) + bsX(s) + kX(s) = F(s)$$

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$$

Open-Loop Step Response

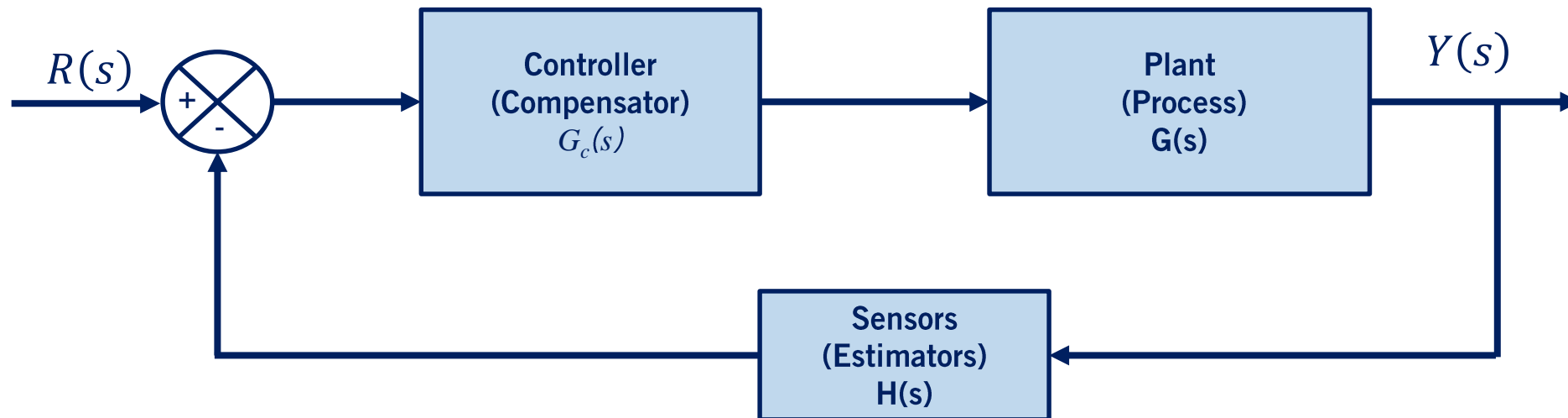
- Let $m = 1$, $b = 10$, $k = 20$, $F = 1$.



Closed-loop Response

- For the unity feedback, i.e., $H(s) = 1$, the closed loop system is given by,

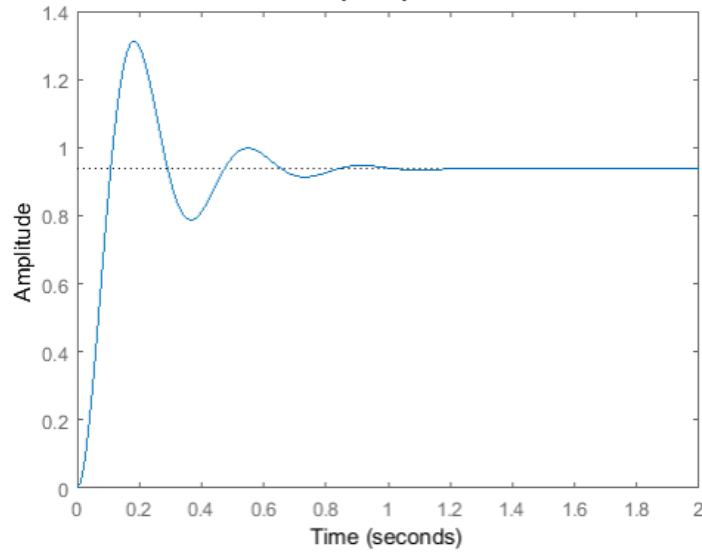
$$\frac{Y(s)}{R(s)} = \frac{G(s)G_c(s)}{1 + G(s)G_c(s)}$$



Step Response

P Controller

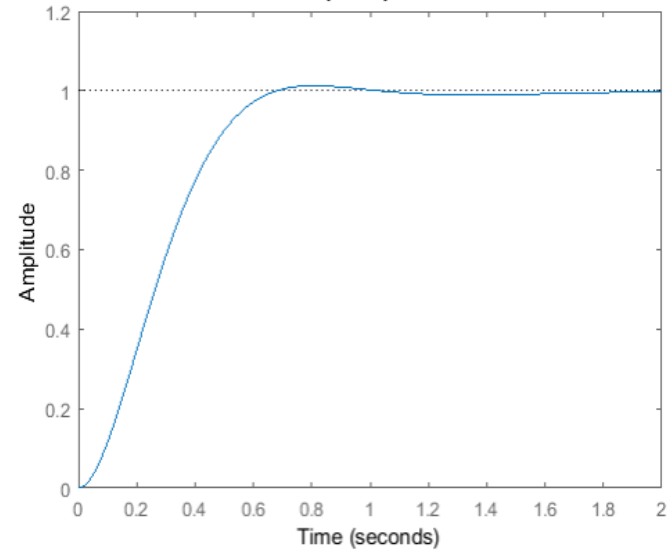
Step Response



$$G_P(s) = K_P$$

PD Controller

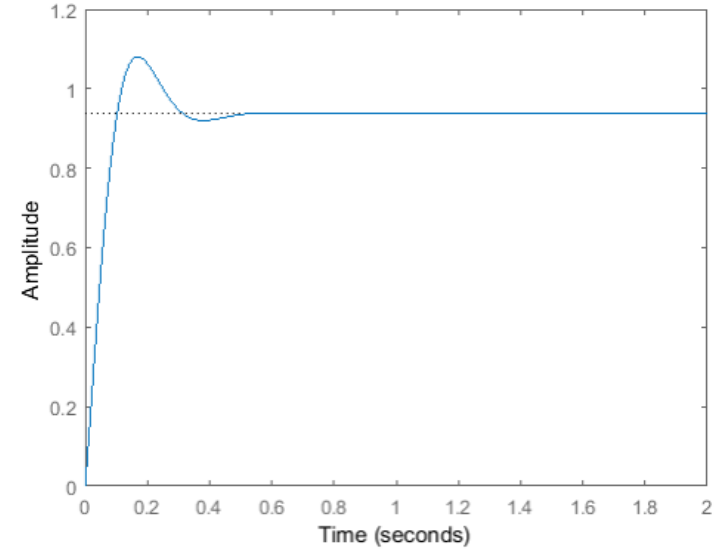
Step Response



$$G_{PD}(s) = K_P + sK_D$$

PI Controller

Step Response



$$G_{PI}(s) = K_P + \frac{K_I}{s}$$

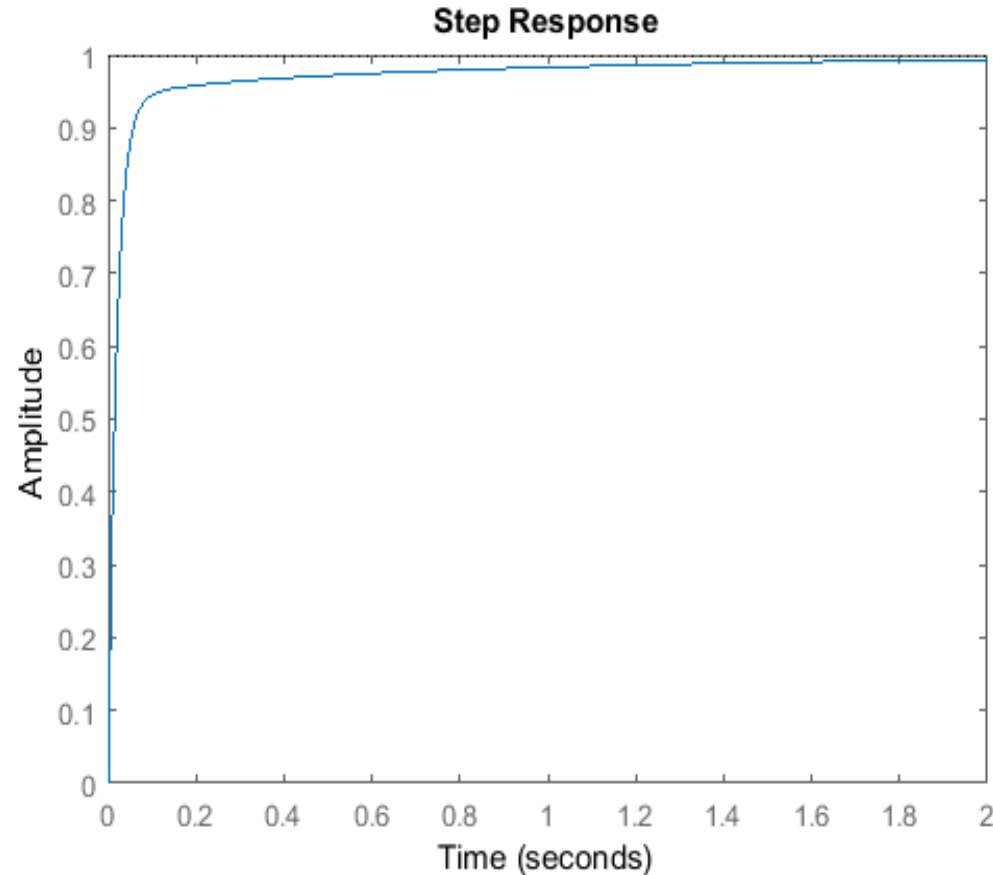
Step Response for PID Control

- PID controller is given by,

$$G_{PID}(s) = \left(K_P + K_D s + \frac{K_I}{s} \right)$$

- The closed loop system becomes,

$$G_{CL}(s) = \frac{K_D s^2 + K_P s + K_I}{s^3 + (10 + K_D)s^2 + (20 + K_P)s + K_I}$$



Summary

What we have learned from this lesson:

- Why we employ controller and its importance
- PID controller (simple but useful and applied) along with the tuning method

What is next?

- Longitudinal speed control of a vehicle using PID controller