Kuwait University
College of Engineering and Petroleum

جامعة الكويت
KUWAIT UNIVERSITY

ME417 CONTROL OF MECHANICAL SYSTEMS

Part I: Introduction to Feedback Control

Lecture 9: Proportional-Integral-Derivative (PID) Control via Gain Tuning

Summer 2020

Ali AlSaibie

Lecture Plan

- Objectives:
  - *Start to move beyond a simple proportional (gain) controller.*
  - *Gain an intuitive understanding on how the PID Controller works.*
- Reading:
  - *Stand-Alone Lecture*

جامعة الكويت
KUWAIT UNIVERSITY

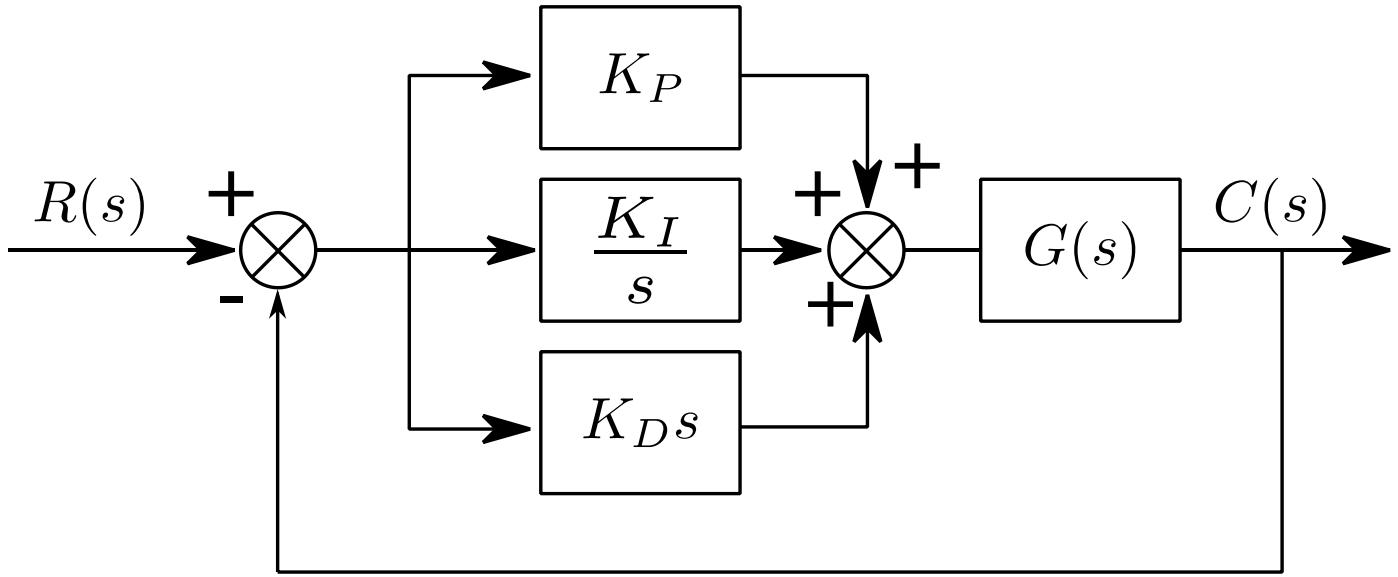
Gain Tuning?

- In this course, we focus on introducing analytical and graphical techniques to derive accurate gain values for whichever controller form we choose.
- In reality, and on many real-world applications, we have to make adjustments to those calculated gains to account for modeling errors, disturbance modeling and controller robustness.
- The analytical and graphical techniques should aid the control engineer in understanding the behavior of the controller and so, knowing what to expect when changing controller gain/s.
- In this lecture we will review the different components of a PID controller and how each of its parts affects the total response.
- In later lectures we will analyze the PID Controller through analytical and graphical methods.

جامعة الكويت
KUWAIT UNIVERSITY

Proportional-Integral-Derivative (PID) Control

- A PID controller applied in a unity feedback system is shown in the block diagram.
- The error value feeds into three parallel controllers
  1. One acts <u>proportionally</u> to the error value
  2. One acts on the <u>integral</u> of the error (on the area under the error curve)
  3. One acts on the <u>derivative</u> of the error (on the rate of change of the error)



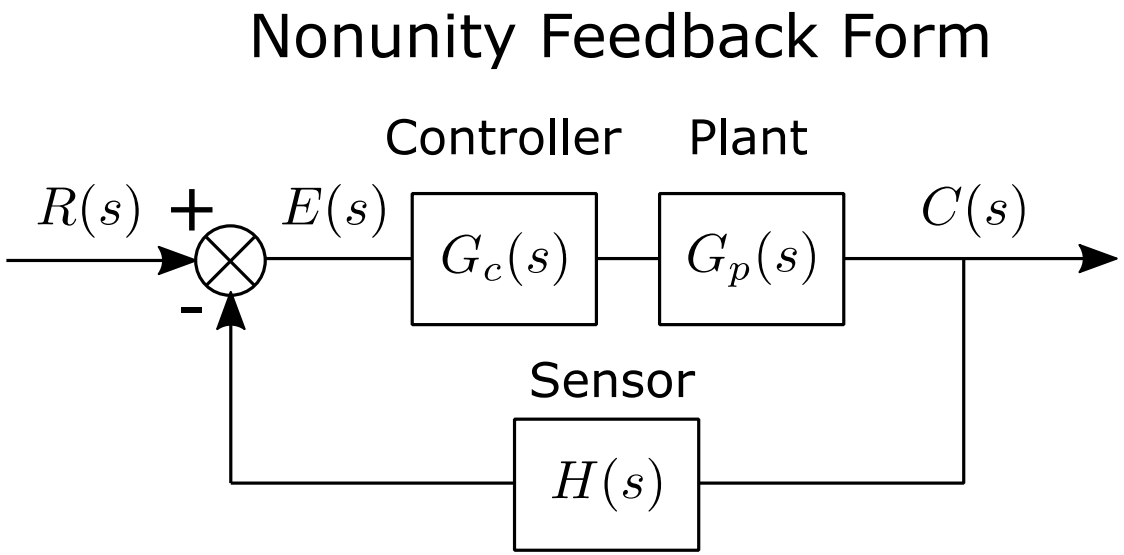
The P, PI, PD, PID Controller

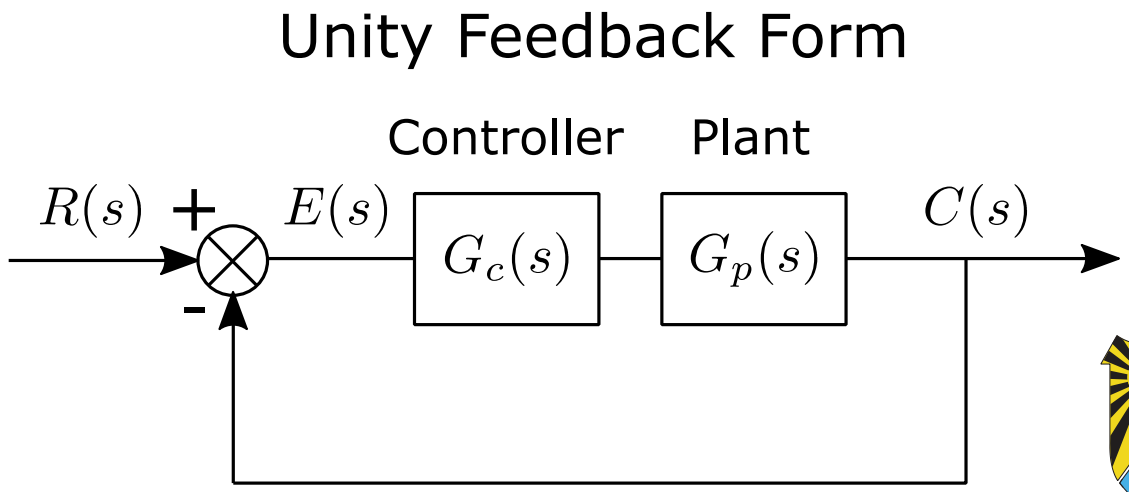
- If only the proportional controller is used, we refer to our controller as a **P Controller**, or **Proportional Controller**.
  - The proportional term is always used in PID control
- If on top of the proportional controller we just use the integral controller, we term our controller a **PI**, or Proportional-Integral, controller.
- If on top of the proportional controller we just use the derivative controller, we term our controller a **PD**, or Proportional-Derivative, controller.
- And if we use all the three components, we term our controller a **PID** Controller.
- All the preceding controllers are still technically **PID** controllers, with one or more gain is set to zero. But it helps to note exactly which components are used.

جامعة الكويت
KUWAIT UNIVERSITY

The error

- In the standard feedback form, the error term is well defined. It is the difference between the reference and the feedback signal.
- In a unity feedback system, the feedback signal is just the output of the system.
- In the time domain: $e(t) = r(t) - c(t)$, for unity feedback systems.



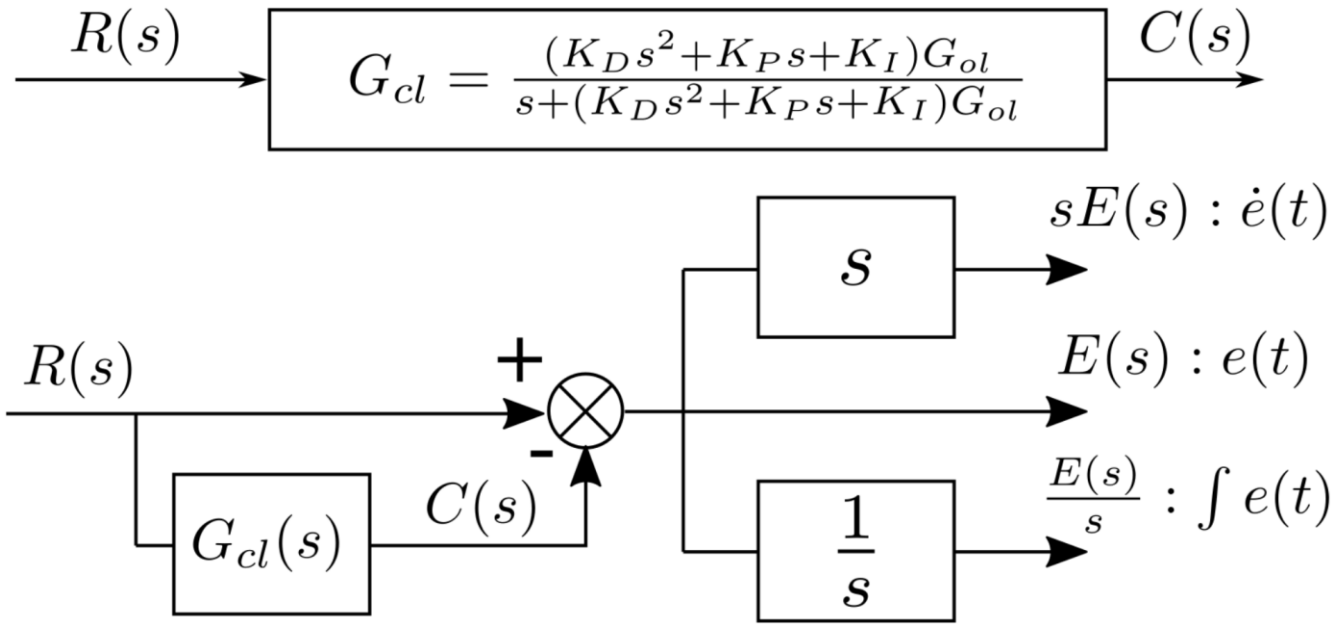


Error Dynamics

- If we have the closed loop transfer function G_{CL}, we can express the relationship between the input and the error, the integral of the error and the derivative of the error.

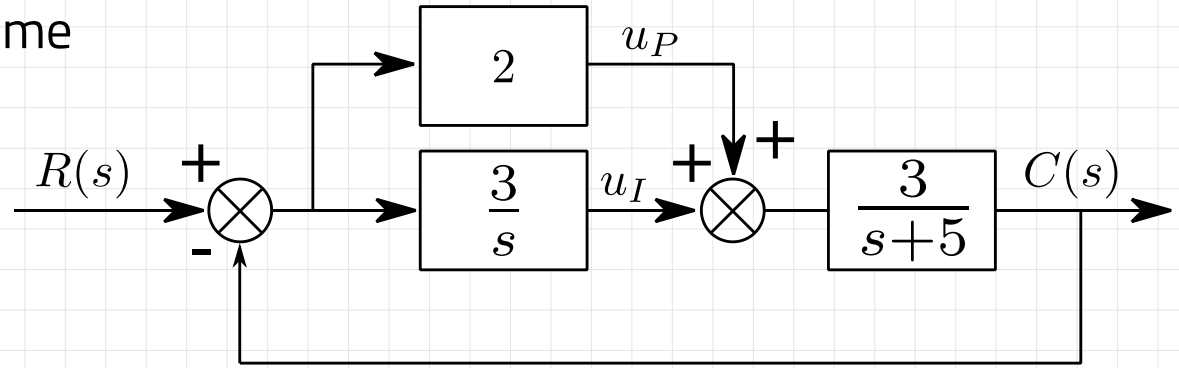
$$E(s) = (1 - G_{CL}(s))R(s)$$

- To understand how the PID controller works, we need to understand how the error behaves. Think in terms of the error dynamics/response.



Reducing the feedback control block diagram and reorienting a bit can give us a relationship between the input and the error, error integral and error derivative

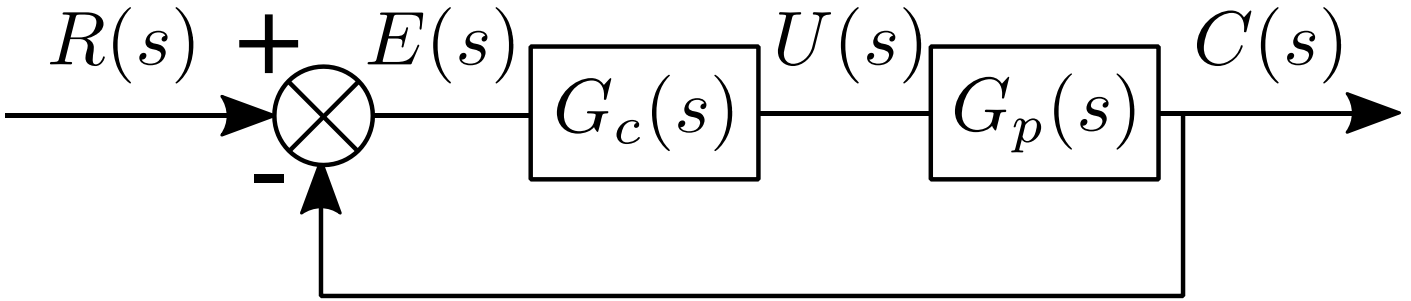
Calculate the value of the input signals $\boldsymbol{u_P}$ and $\boldsymbol{u_I}$ to the plant G for a unit-step input **r** to the control system, at times $t = 0, 2, 4s, \& \; t = \infty$, for the system shown on the figure. This time try to come up with the transfer function $\frac{E(s)}{R(s)}$ and use it

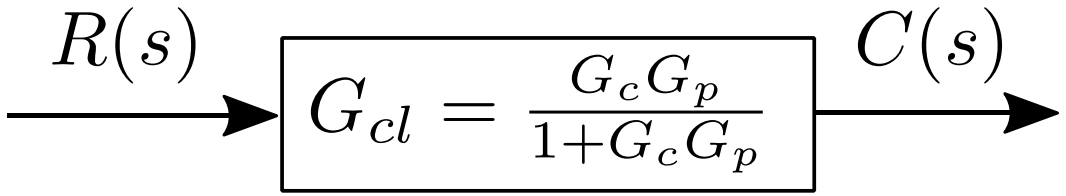


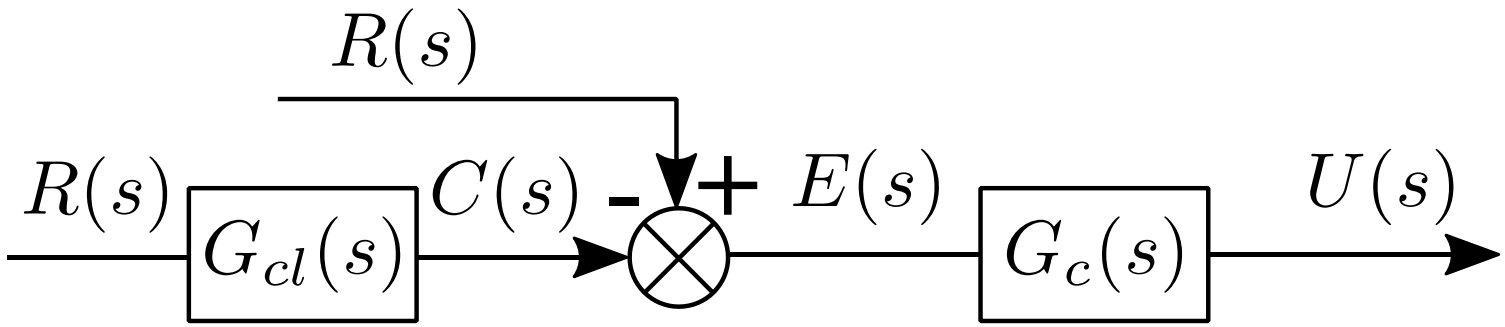
Controller Output

- When applying feedback control, the output of the controller U, is internally calculated by the control law.
- We can express $G_u(s) = \frac{U}{R}$ which allows us to simulate the controller output U given the reference input R

$$U = G_c E = G_c(R - RG_{cl})$$
$$\Rightarrow G_u(s) = G_c(1 - G_{cl})$$

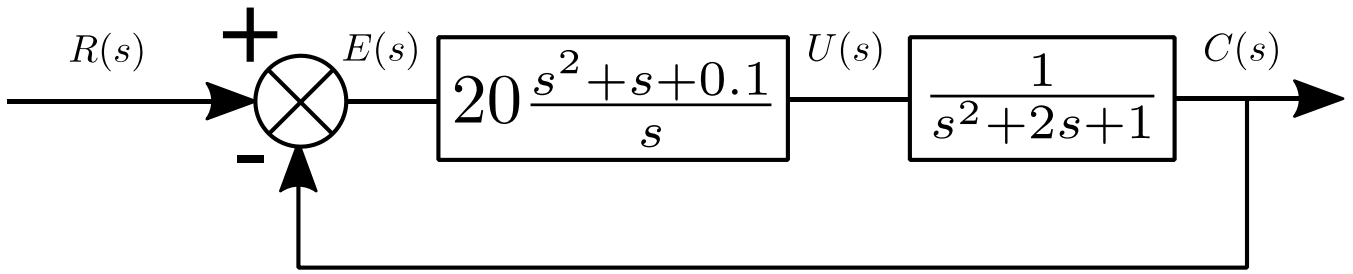


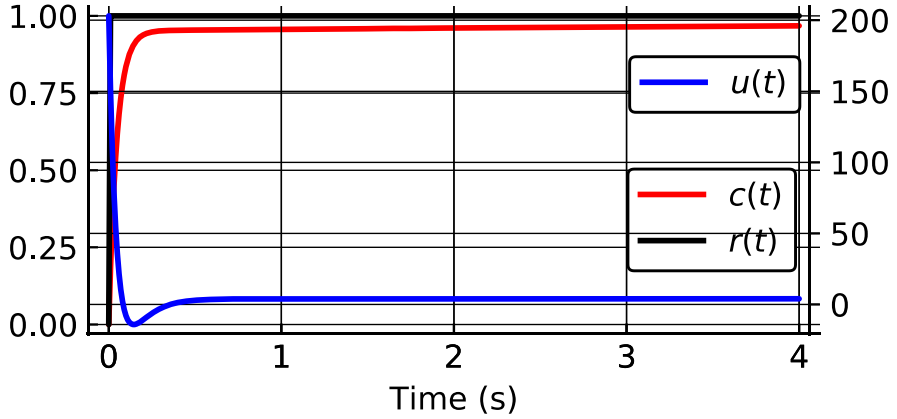




Reference Input vs Plant Input

- Consider the following system, observe the time response of :
  - The input to the control system $r(t)$, versus
  - The input to the plant $u(t)$ (Output of the controller)



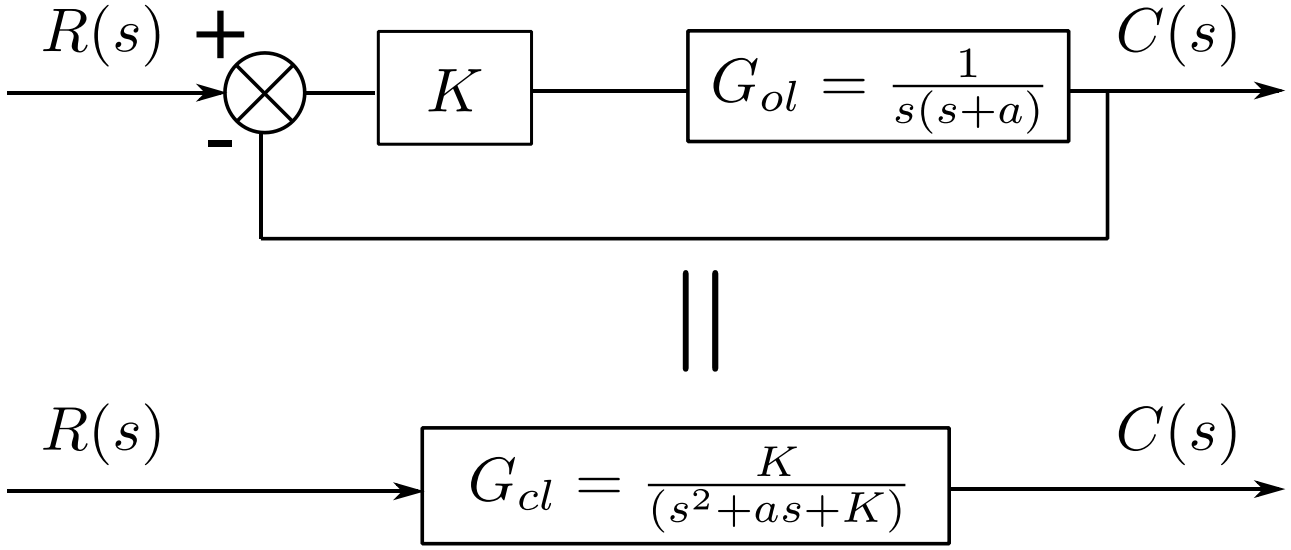


The P Controller

- The P controller is a simple gain controller that scales the error value to produce an output $u(t)$ that is applied to the system/plant

$$u_p(t) = K_p e(t)$$

- Where K_p is the **proportional gain**
- In a feedback system the proportional gain changes the dynamics of the closed-loop system by allowing the closed-loop poles to move along a defined path (locus) in the s-plane.

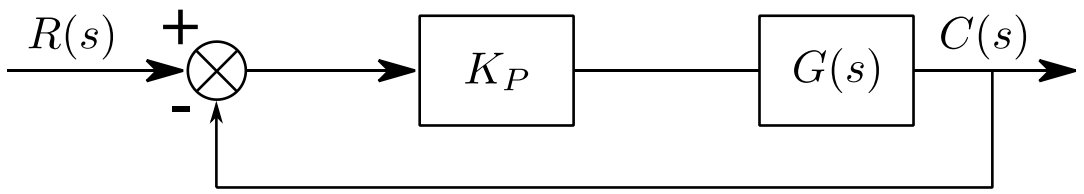


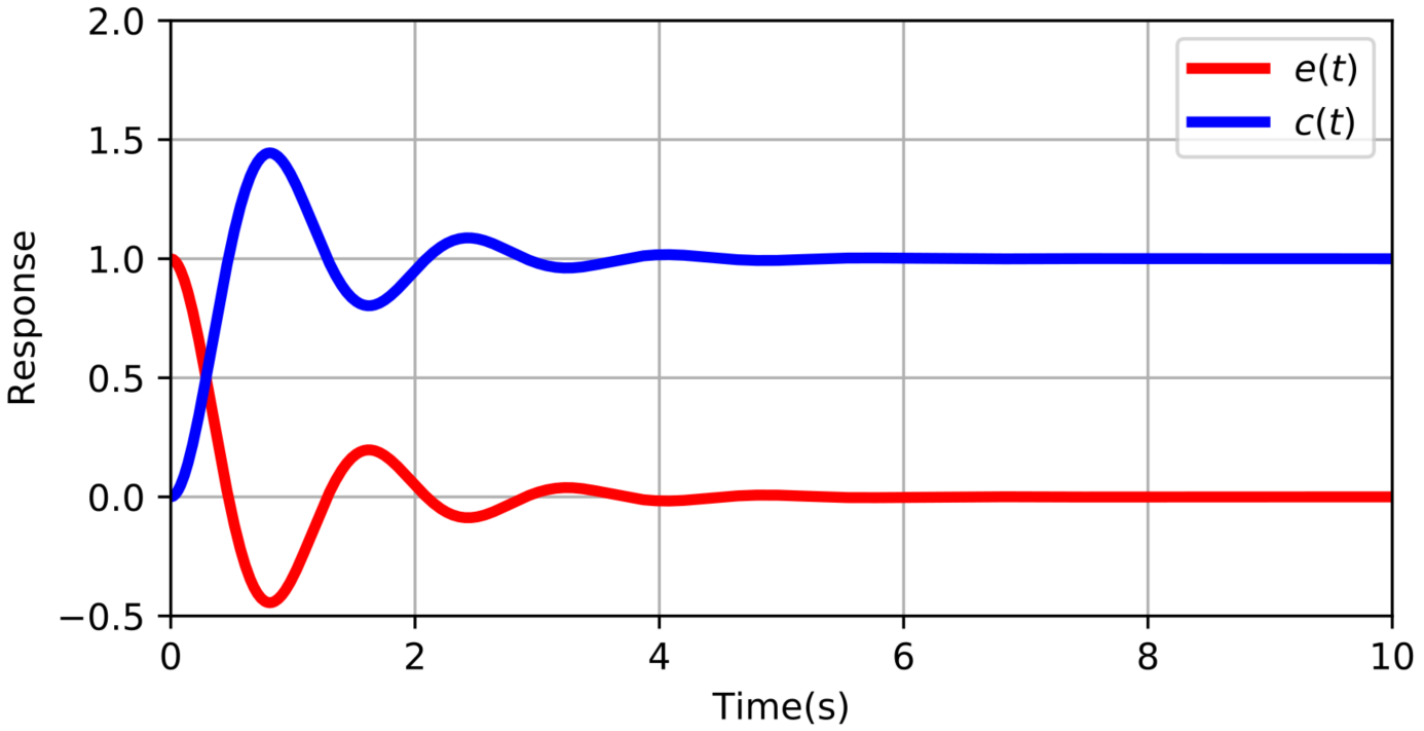
The error

- The error: $e(t) = r(t) - c(t)$, in unity feedback
- Since the proportional controller is proportional to the error, we expect the control output u_p to follow the error dynamics proportionally

Error response to a unit-step input in a feedback control system with $G_{ol} = \frac{2}{s(s+2)}$, with a **P** Controller.

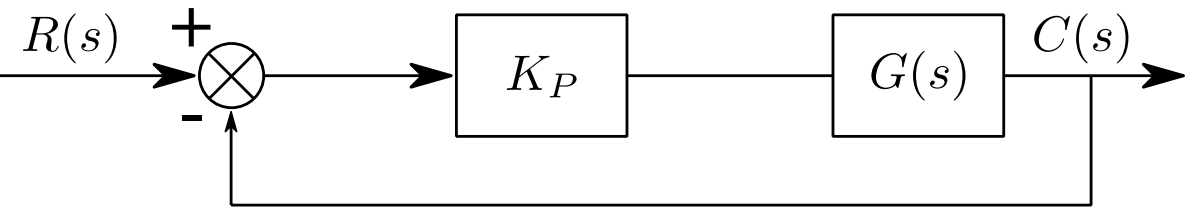
$K_p = 8$

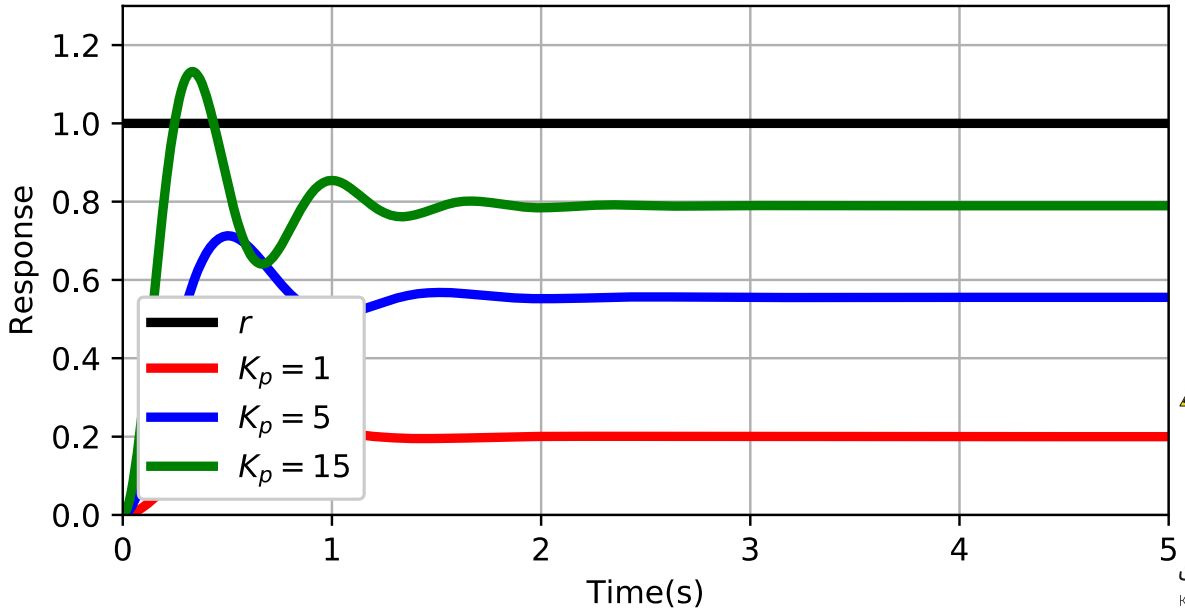




- In terms of its affect on the response, increasing the proportional gain:
  - (+) Decreases the steady-state error
  - (+) For 2nd order and higher systems, increases %OS.
  - (-) Increases overshoot
  - (-) Increases actuator effort
    - *In a real-world application, there is a limit to power/force/input to a system, we can't increase K_p without a limit.*

$$G(s) = \frac{5}{s^2+5s+20}$$

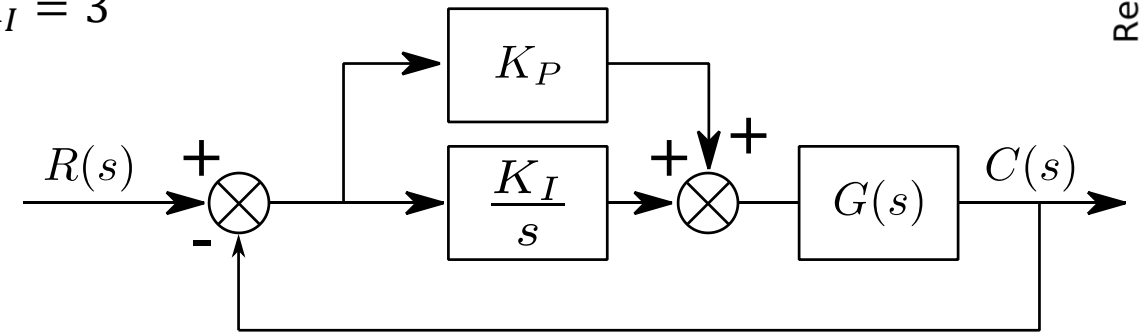


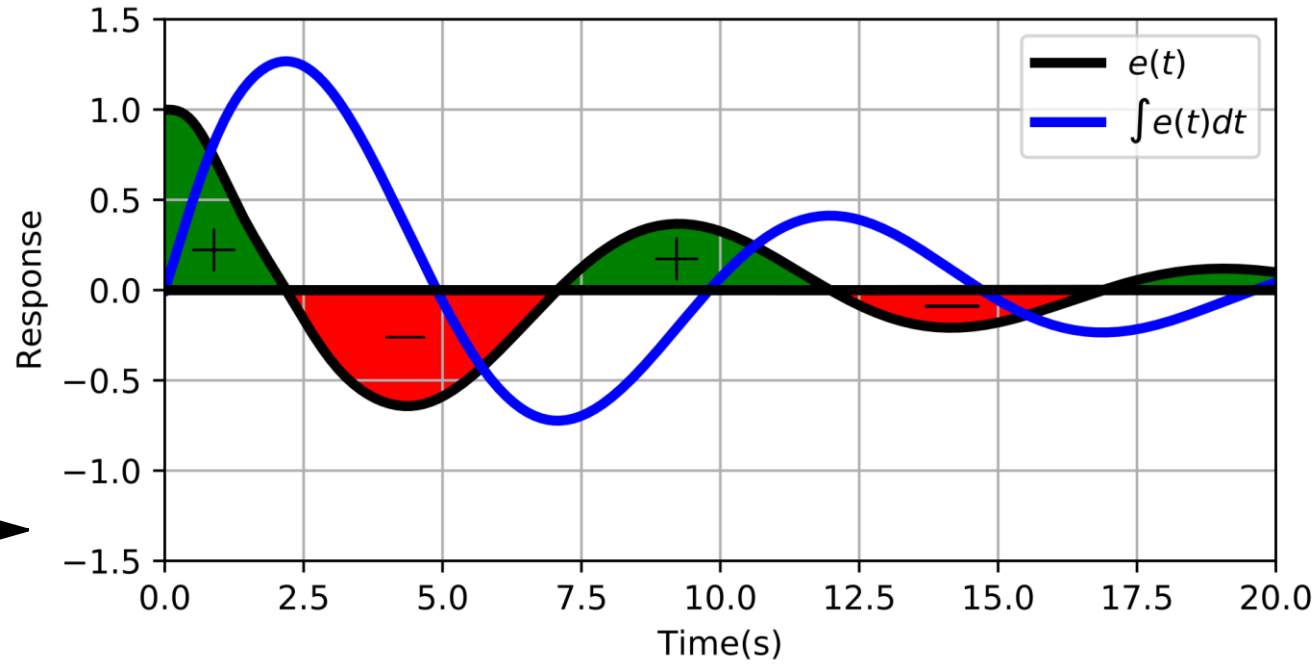


The Error Integral and Integral Control Component

- The integral of the error is the area under the error curve
- The PI Controller: $u_{PI}(t) = K_P e(t) + K_I \int e(t)dt$
- Conceptually, the integral component of the controller tries to reduce the steady-state error by balancing between the positive and negative areas.
- If Green area > Red Area -> $u_I = K_I \int e(t) > 0$, and vice versa
- If Green area = Red Area -> $u_I = 0$

Error response to a unit step input in a feedback control system with $G_{ol} = \frac{2}{s(s^2+2s+15)}$, with a **PI** Controller. $K_p = 2, K_I = 3$

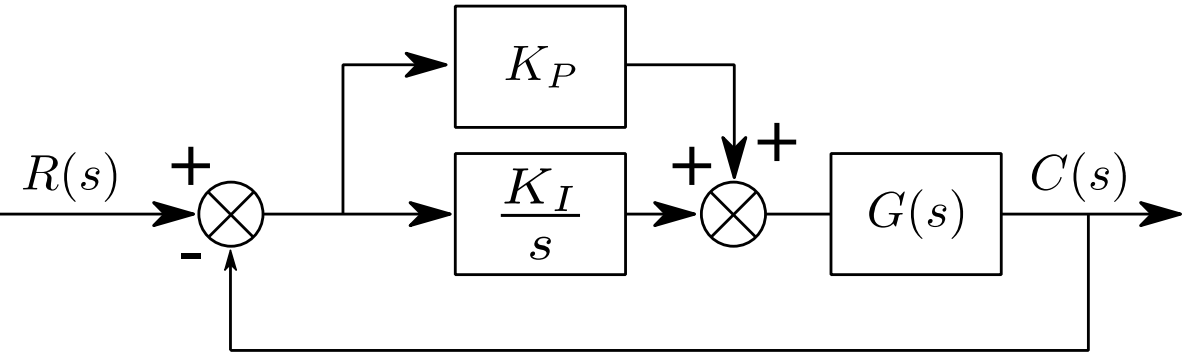


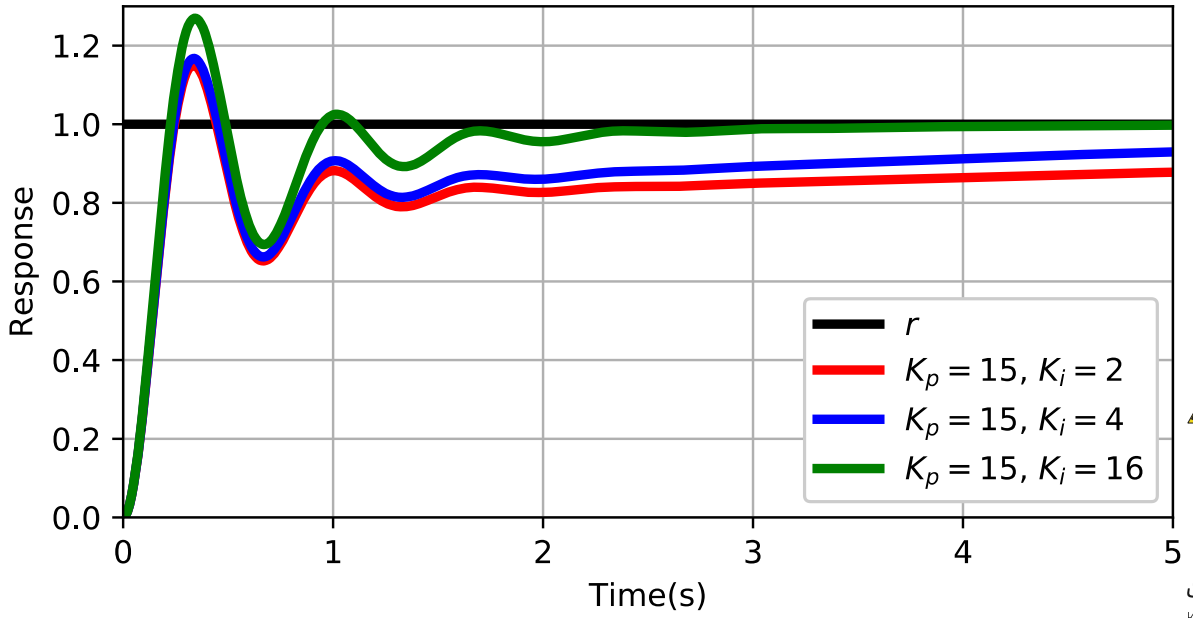


The PI Controller

- In terms of its affect on the response, increasing the integral gain:
  - (+) Reduces the steady-state error
  - (+) Reduces the need for aggressive K_p gain
  - (-) Increases oscillation
  - (-) Increases settling time T_s
  - (-) Increases overshoot

$$G(s) = \frac{5}{s^2+5s+20}$$

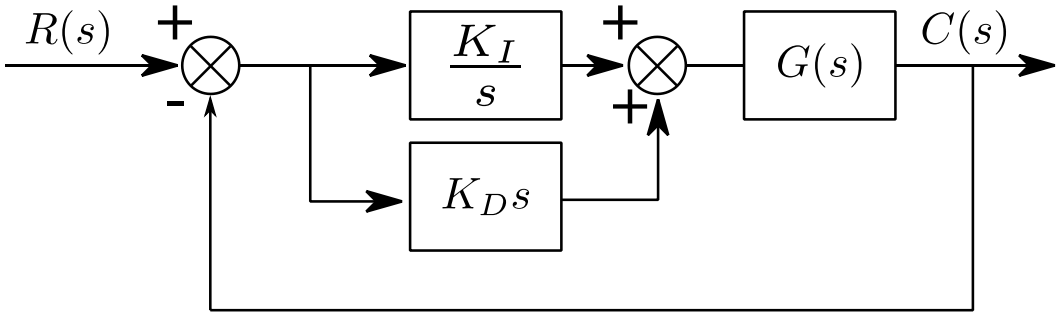


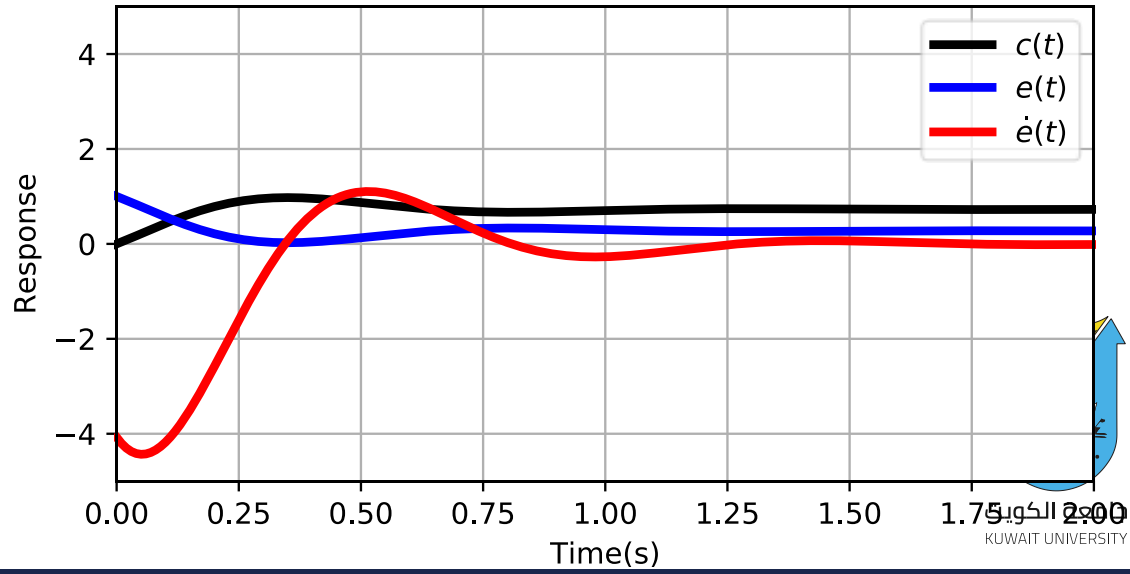


The Error Derivative and Derivative Control Component

- The error derivative $\dot{e}(t)$ responds to the rate of change of the error.
- In a way, it predicts the future by looking at the rate of change. The rate of change is an indication of how quick the desired reference is moving away from the system state (output).
  - *If you're in your car, following a friend's car, it is one thing to speed up/down to maintain a fixed gap, but if your friend suddenly speeds up or slams on the brakes, your response would be more aggressive in anticipation of a big error change.*
  - *Because you **know** the error **will** change*

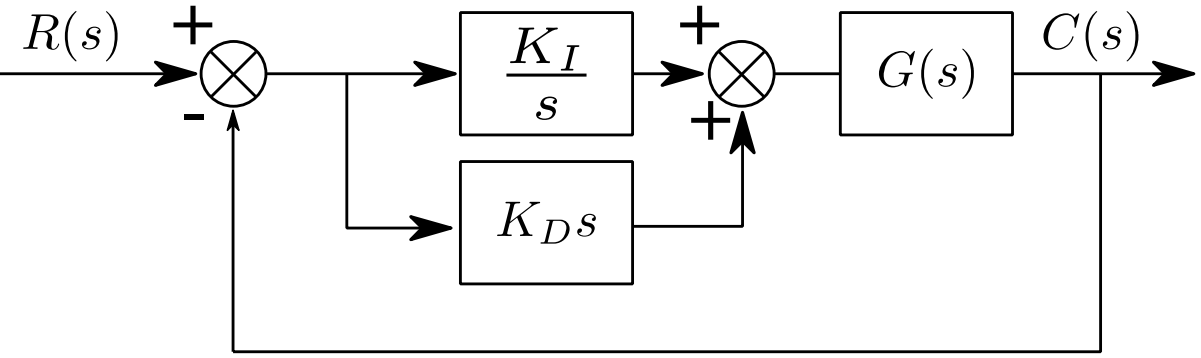
Error response to a unit step input in a feedback control system with $G_{ol} = \frac{2}{(s^2+2s+15)}$, with a **PD** Controller. $K_p = 20, K_D = 2$

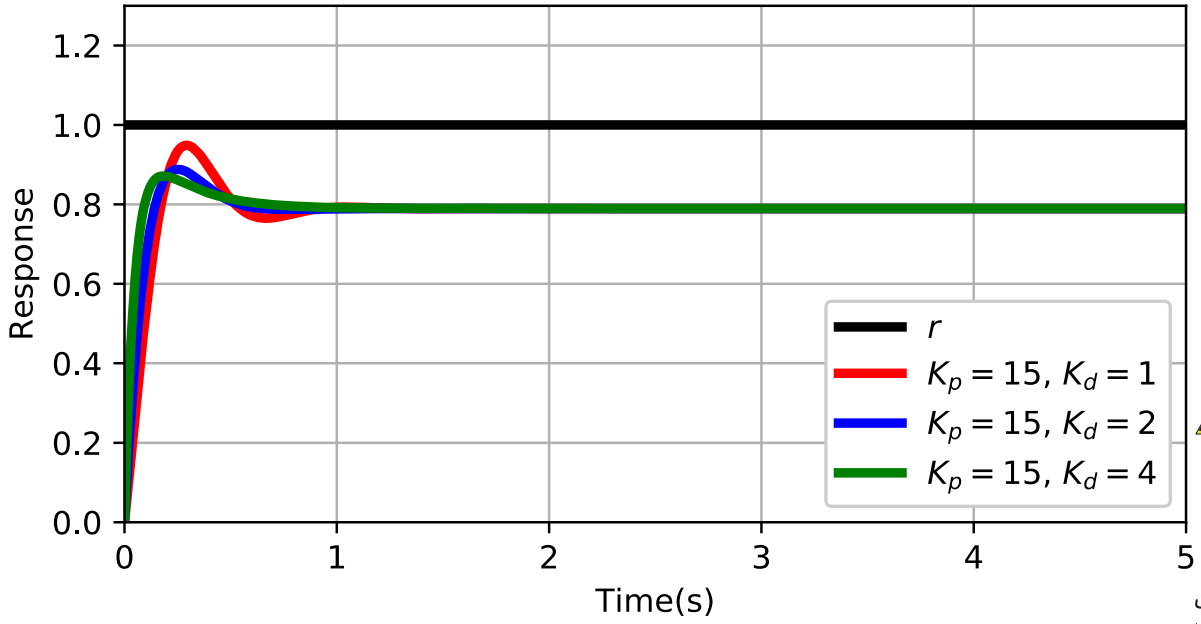




- In terms of its affect on the response, increasing the integral gain:
  - (+) Can reduce rise time T_r
  - (+) Reduces overshoot
  - (+) Reduces oscillation and settling time T_s
  - (-/+) Doesn't affect Steady-State Error

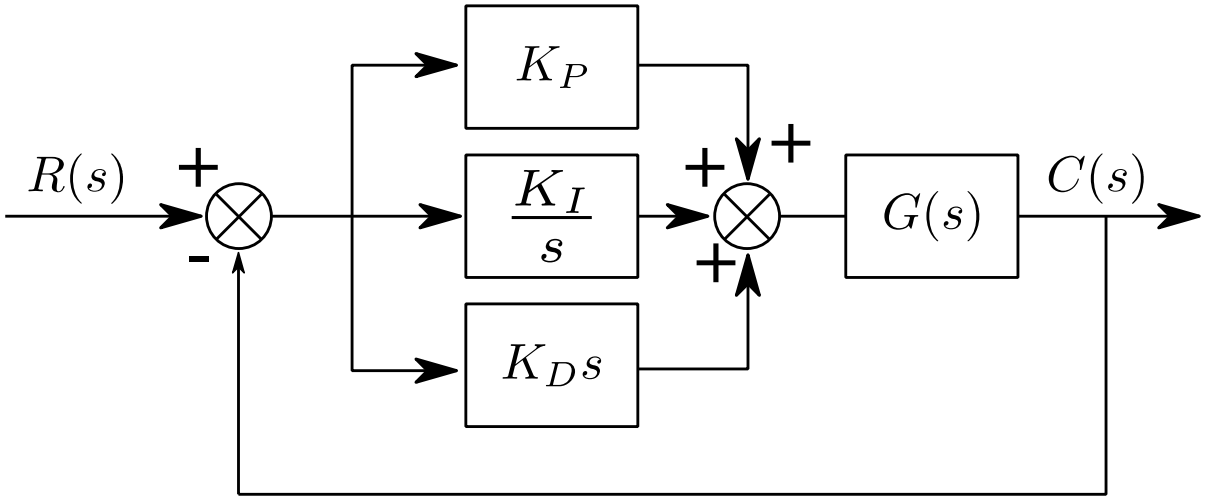
$$G(s) = \frac{5}{s^2+5s+20}$$

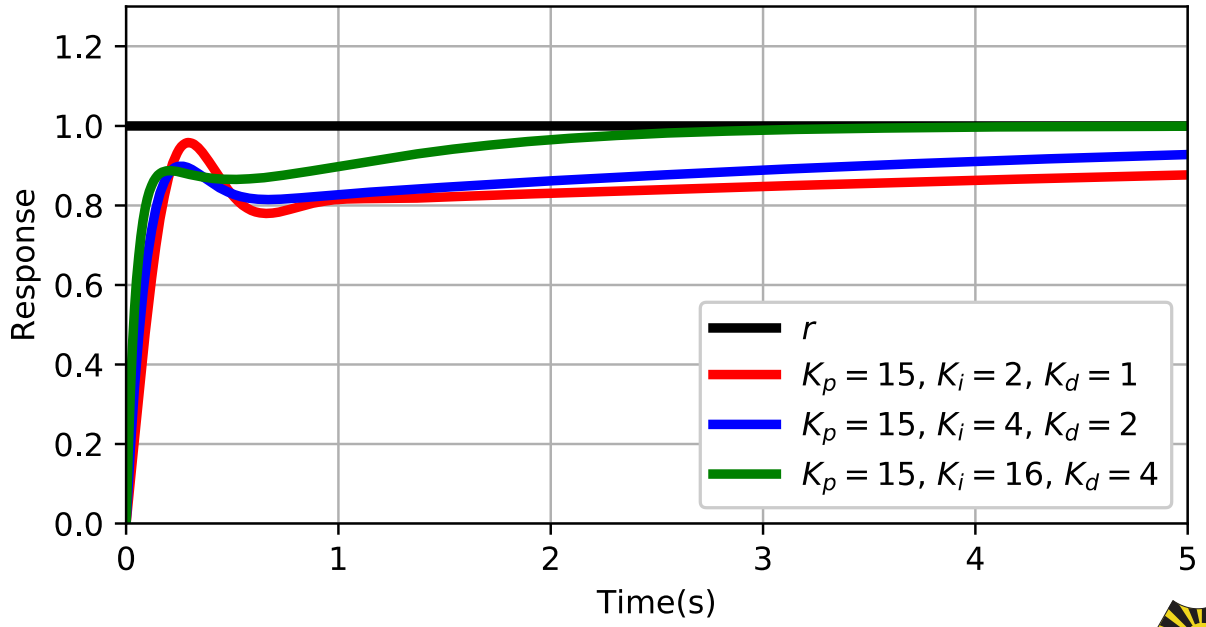




- For each component of the PID Controller there are pros and cons.
- Depending on the system we are trying to control, one or more of the components of the controller may be suitable.

$$G(s) = \frac{5}{s^2+5s+20}$$



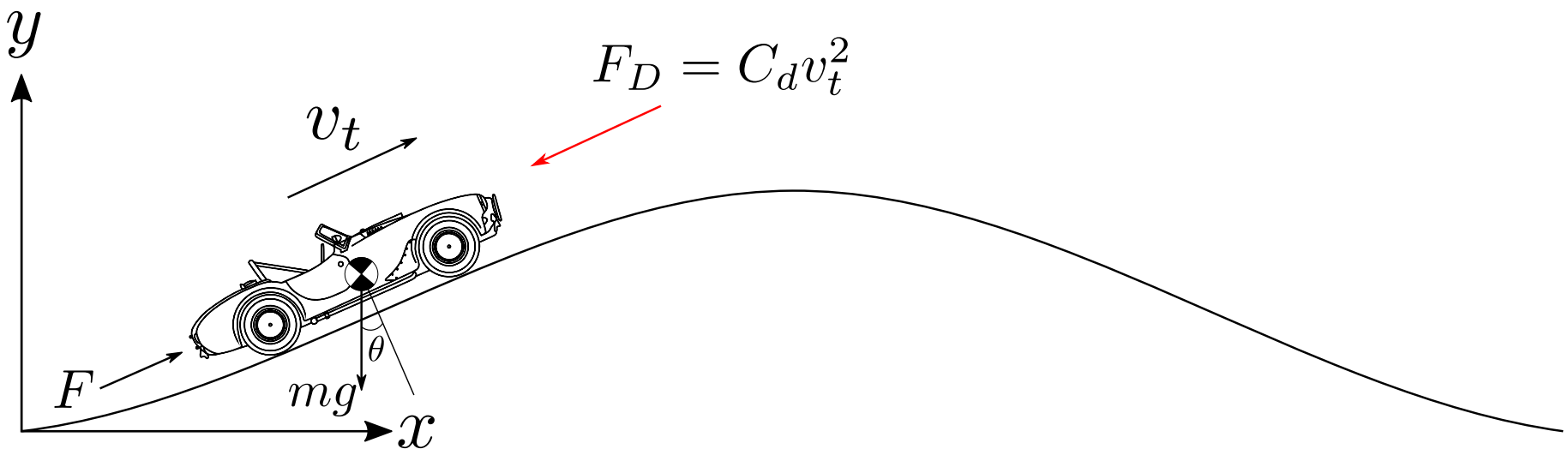


PID Tuning Methods

- There are a few methods available to tune PID controllers on real system. A popular method is called the Ziegler-Nichols Method
- First K_I, K_D are set to zero. The K_P gain is increased until the system begins to oscillate. That value of K_P is termed K_u and the oscillation time period T_u
- Then the remaining gains are set as follows.

Controller	K_P	K_I	K_D
P	$0.5K_u$	–	–
PI	$0.45K_u$	$0.54K_u/T_u$	–
PID	$0.60K_u$	$1.2K_u/T_u$	$3K_uT_u/40$

- Cruise Controller on a Car
  - A cruise control system tries to maintain the set speed of the car by increasing/decreasing engine throttle.
  - Air Drag is a function of the car speed
  - We would like to implement a PID Controller for the CCS. Let's investigate how each component of the PID Controller influences the response.
  - Refer to MATLAB Simulation.



A few notes about PID

- It is one of many forms of feedback controllers.
- It is the best candidate when controlling a system without a model available and assuming constant plant dynamics.
- When implementing the derivative component, we need to calculate the derivative. Numerically, this is a source of noise and can be unreliable.
- On real controller implementation, limited actuator ability can lead to integrator windup (The error integral grows unnecessarily). This can be managed, but it needs attention and compromise.
- PID can be used in conjunction with other controllers.
  - If a model of the system is available, a feed-forward controller can be used and the PID controller will focus on reducing the small remaining error

جامعة الكويت
KUWAIT UNIVERSITY