EE3305 / ME3243 Robotic System Design

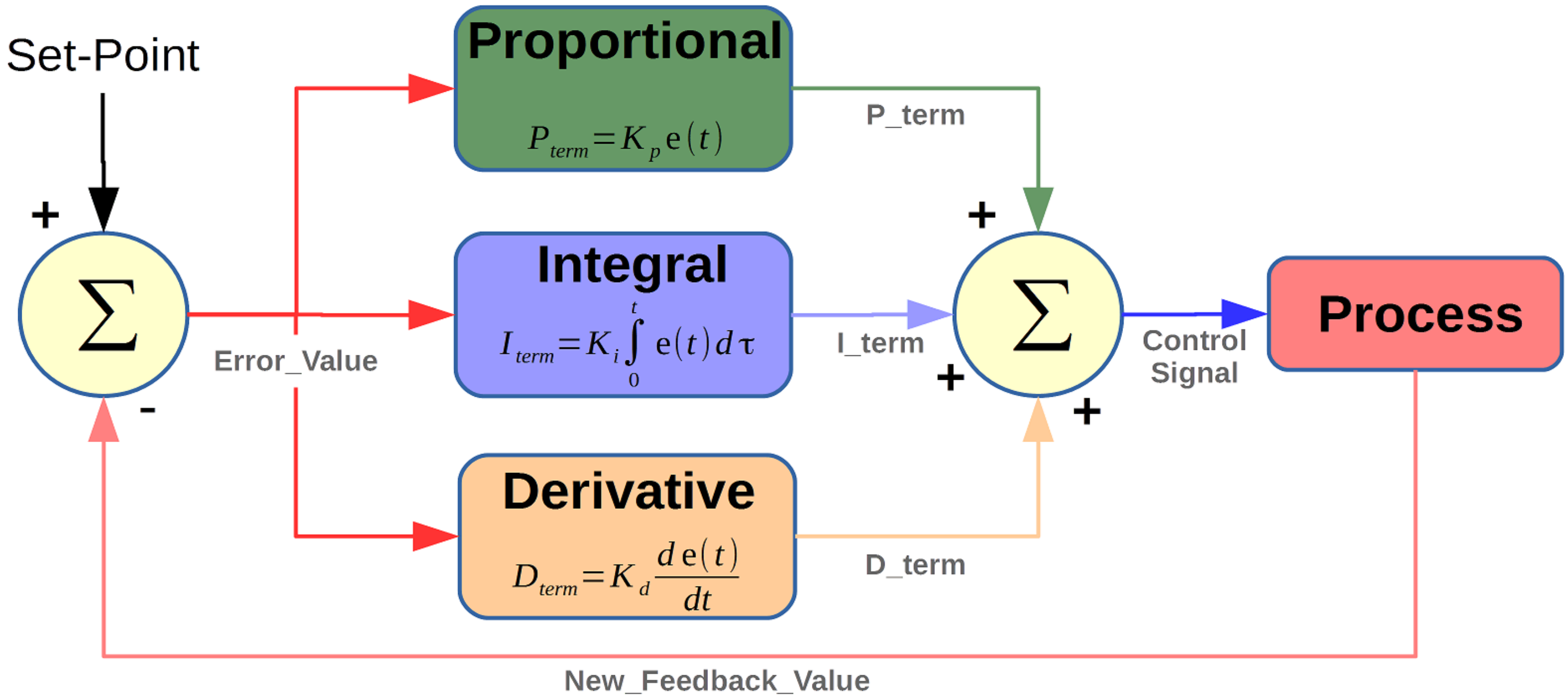
PID Control with ROS Implementation

A/Prof. Prahlad Vadakkepat

Dr. Andi Sudjana Putra

Contents

- The Need for Feedback Control
- Proportional (P), Integral (I), and Derivative (D) Control
- Computer Implementation of PID Control



What is control?

- Control is to make a dynamic process behave in a way we want.
  - A dynamic process refers to a system, system component, or phenomenon that changes or evolves over time.
  - Understanding dynamic processes is essential for modeling, predicting, and controlling systems in various domains.
  - Tools and techniques such as mathematical modeling, simulations, and control theory are often employed to analyze and manage dynamic processes to achieve specific goals or outcomes.

What is control?

Control is to make a dynamic process behave in a way we want.

To manage dynamic processes effectively, you need:

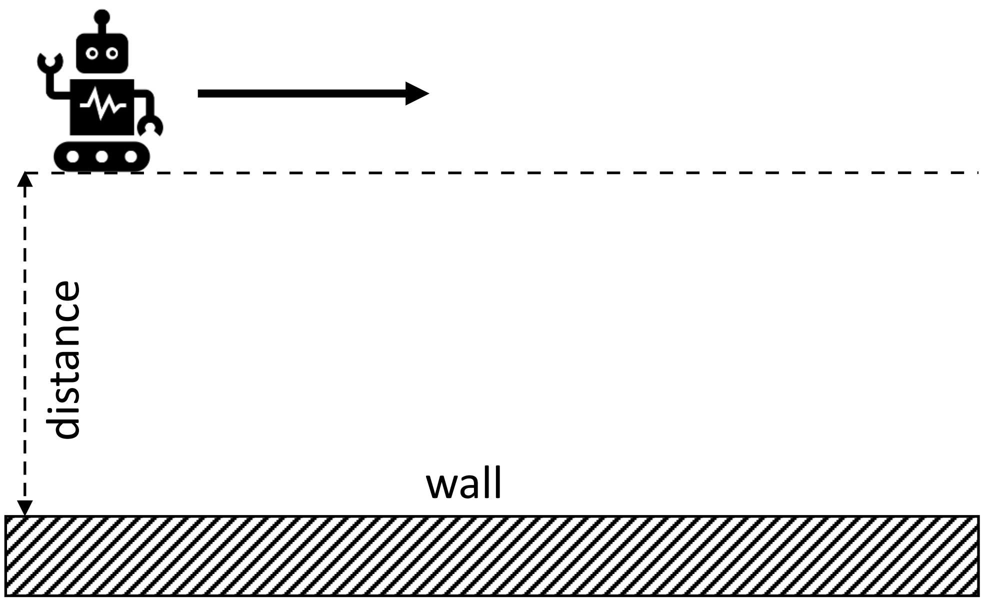
- A means to exert influence over the process.
- A method to observe and understand the process' behavior.
- A way to specify the desired behavior for the process.

The Need for Feedback Control

'You cannot control what happens to you, but you can control your attitude toward what happens to you, and in that, you will be mastering change rather than allowing it to master you' – Brian Tracy

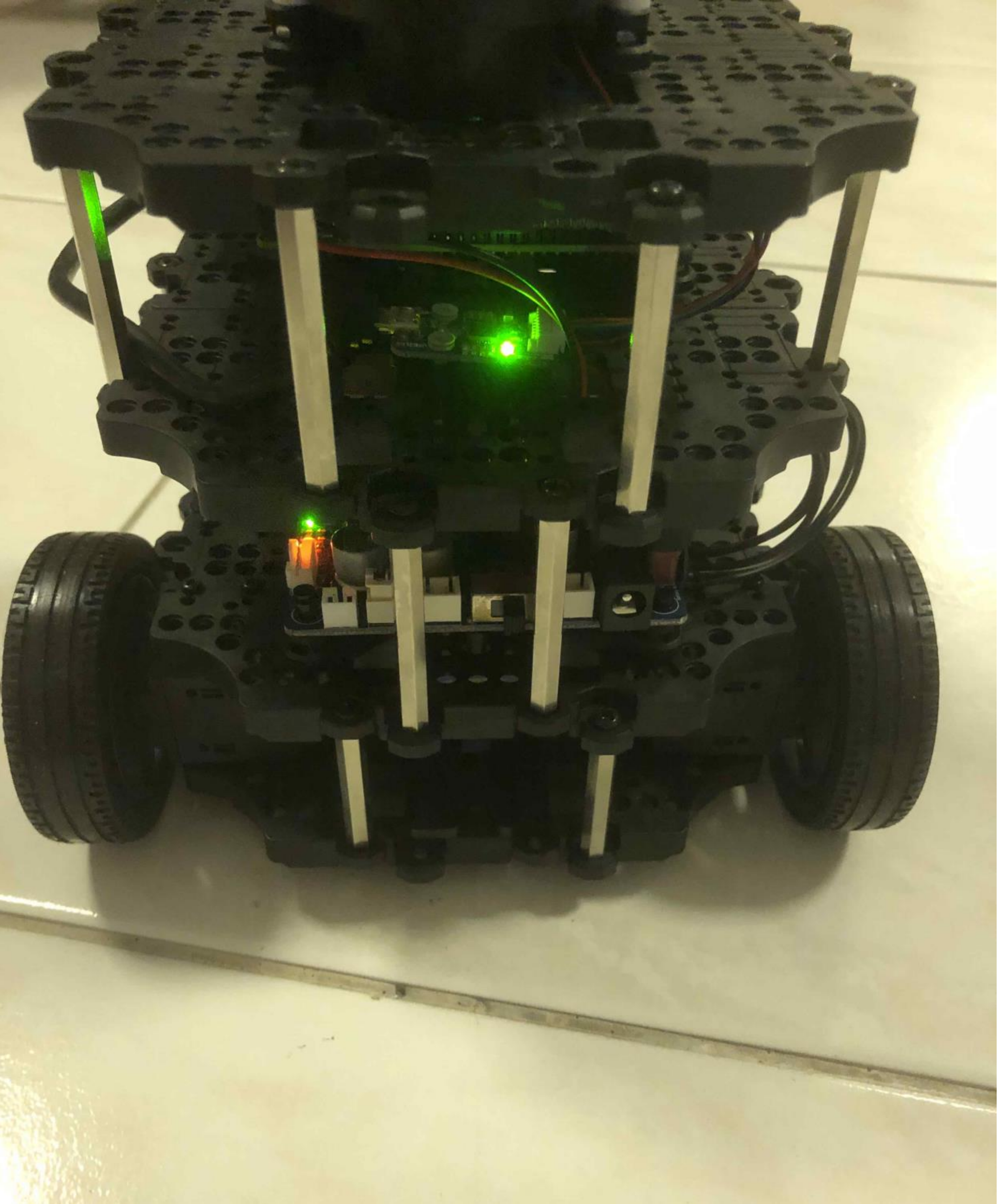
Suppose a robot is required to move along a straight wall, maintaining certain distance from the wall.

How to achieve this task?



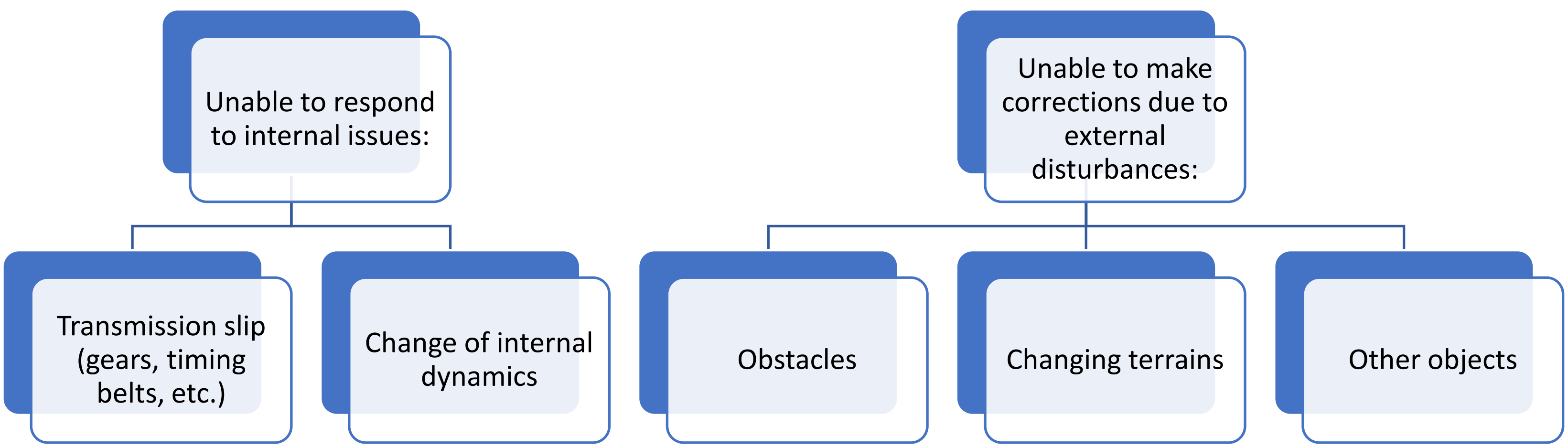
- Place the robot at the desired distance from the wall
- Orientate the robot parallel to the wall
- Run the robot.

Will it work?



- What might make open loop fail in the Turtlebot as shown?

Why can "open loop" fail?



System modeling

- System modeling is a fundamental step in control system design and operation.
- It enables a better understanding of the system's behavior, the design of effective control strategies, and the ability to optimize and adapt control systems for various applications, leading to improved performance and efficiency.

The mass-damper-spring system

The mass-damper-spring system is a simple but fundamental model that is used to describe the dynamics of a wide range of physical systems, including vehicles, buildings, bridges, and machines.

It is also used to study the behavior of human bodies and other biological systems.

The study of mass-damper-spring systems is significant because it allows us to understand and predict the behavior of these systems under different conditions.

This knowledge can be used to design and improve systems, to prevent failures, and to develop new technologies.

A tunable fuzzy logic controller for vehicle-active suspension systems
– M.V.C. Rao, V. Prahlad

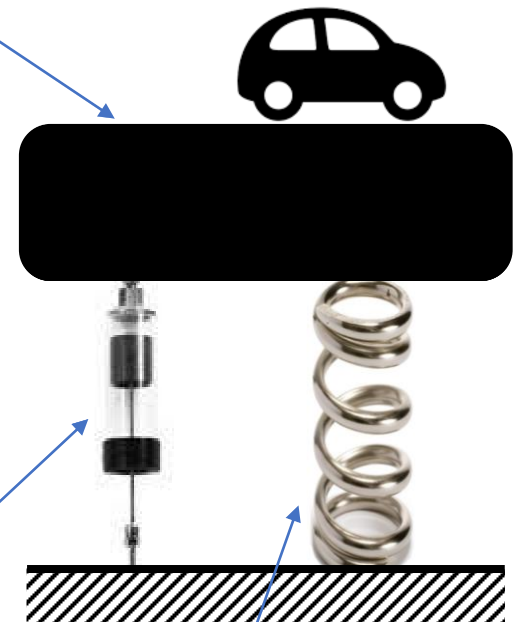
- Fuzzy Sets and Systems, Volume 85, Issue 1, 1 January 1997, Pages 11-21
- https://doi.org/10.1016/0165-0114(95)00369-X

- Keywords:
  - Fuzzy logic controller
  - Linguistic variables
  - Vehicle-active and passive suspension systems
  - Quarter-car reference model

Representing a system as a mass-damper-spring

A car jack system

can be represented by



mass of m

The car is an external force.

damper with damping ratio of c

spring with spring constant of k

mass-damper-spring

The damping ratio, mass, and spring constants are fundamental concepts in system dynamics that help describe and model the behavior of dynamic systems.

The damping ratio influences how quickly a system returns to equilibrium and whether it oscillates.

Mass represents inertia and affects the system's response to forces.

Spring constants characterize the stiffness of elastic elements in the system.

Understanding these parameters is crucial for analyzing and predicting the behavior of dynamic systems in various fields, including engineering, physics, and economics.

Damping Ratio

Damping ratio, often denoted as "ζ" (zeta), is a dimensionless parameter that describes the degree of damping or resistance to oscillation in a dynamic system.

It is a measure of how quickly a system returns to equilibrium after being subjected to an external force or disturbance.

Damping can be categorized as:

Underdamped	critically damped	overdamped.

Damping Ratio

Underdamped

- When ζ is less than 1, the system is underdamped.
- In this case, the system oscillates before settling to its final equilibrium position.
- Underdamped systems are often seen in systems where some energy is conserved.

Critically Damped

- When ζ is exactly 1, the system is critically damped.
- Critically damped systems return to equilibrium as quickly as possible without oscillation.
- They are used in applications where rapid stabilization is crucial.

Overdamped

- When ζ is greater than 1, the system is overdamped.
- Overdamped systems return to equilibrium slowly and do not oscillate.
- They are often used in situations where stability is more important than speed of response.

Mass

In system dynamics, mass represents the amount of matter in a physical system. It is a fundamental property that determines how objects respond to forces.

In dynamic systems, mass can represent the inertia of an object, and it affects how quickly or slowly a system responds to changes in applied forces.

A larger mass typically results in slower responses to forces, while a smaller mass leads to faster responses.

Spring Constants

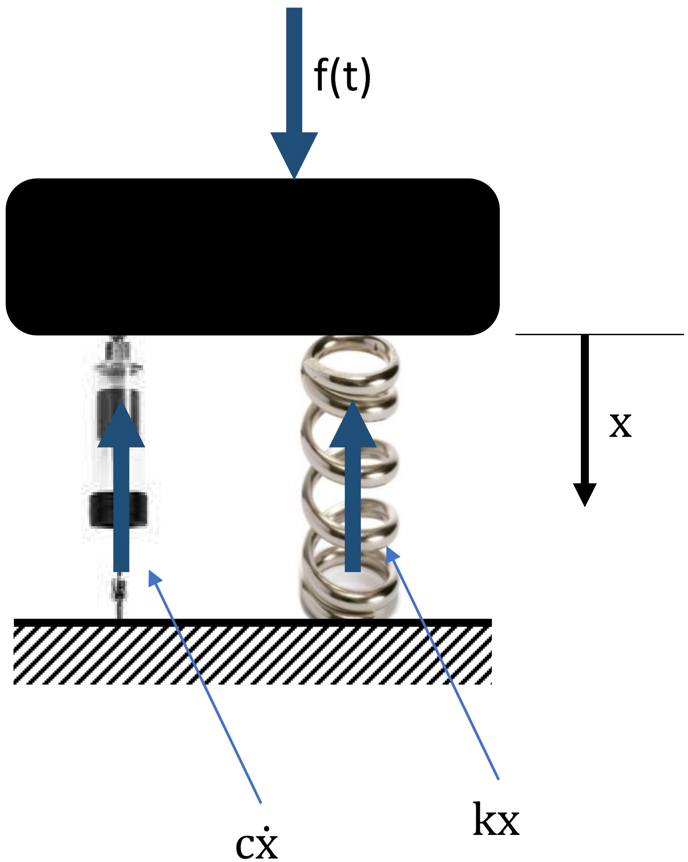
Spring constants, denoted as "k," are parameters that describe the stiffness or rigidity of springs or elastic elements in a system.

Springs are often used in dynamic systems to model the relationship between the displacement of an object and the force applied to it.

The spring constant quantifies how much force is required to displace the object a certain distance.

A higher spring constant indicates a stiffer spring, which requires more force for a given displacement.

Representing a system as a mass-damper-spring



$$\sum F = m\ddot{x}$$

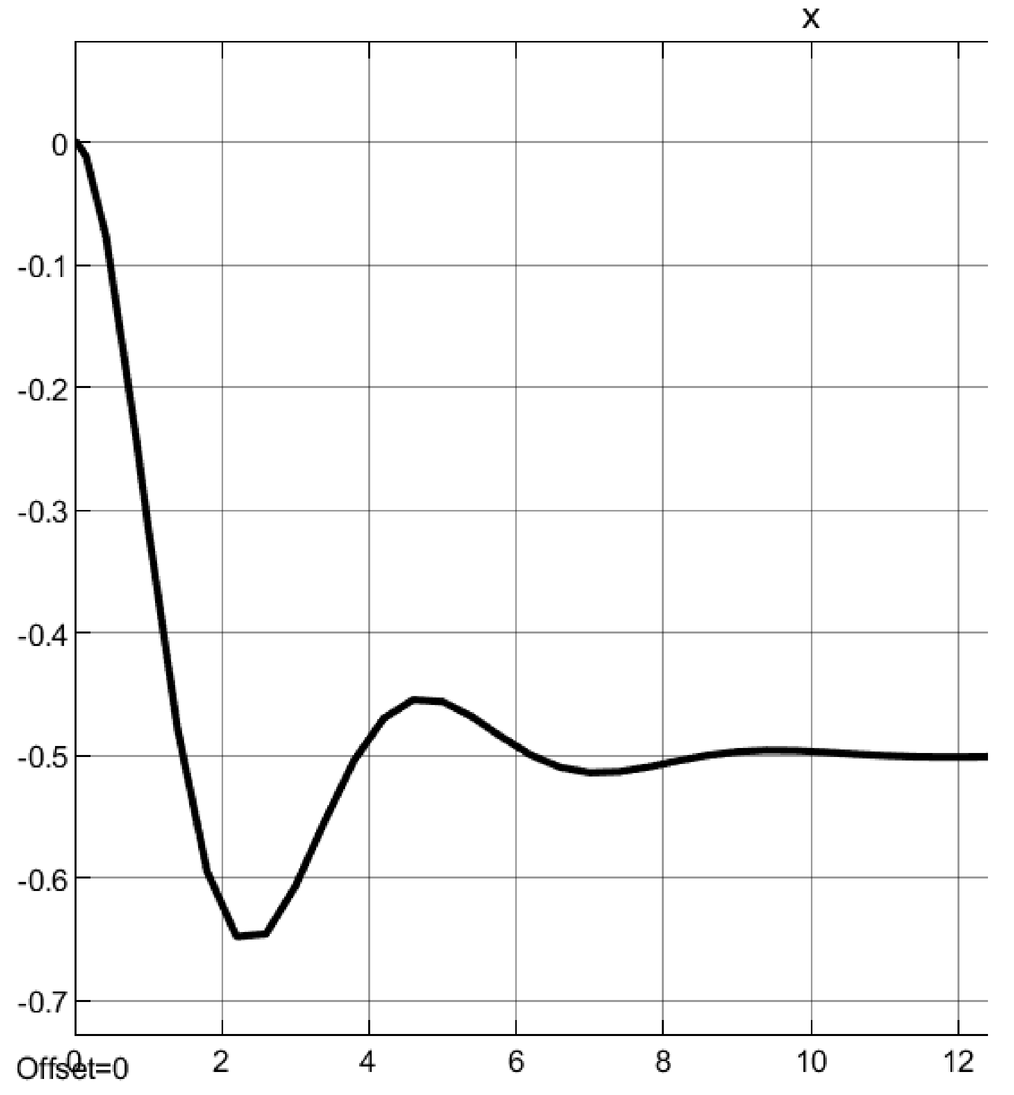
$$f(t) - c\dot{x} - kx = m\ddot{x}$$

$$\ddot{x} = \frac{1}{m}(f(t) - c\dot{x} - kx)$$

Suppose:

- The load is 10 N.
- The jack's mass is 10 kg.
- The jack's damping ratio is 10.
- The jack's spring constant is 20 N/m.

The natural response of the jack is:

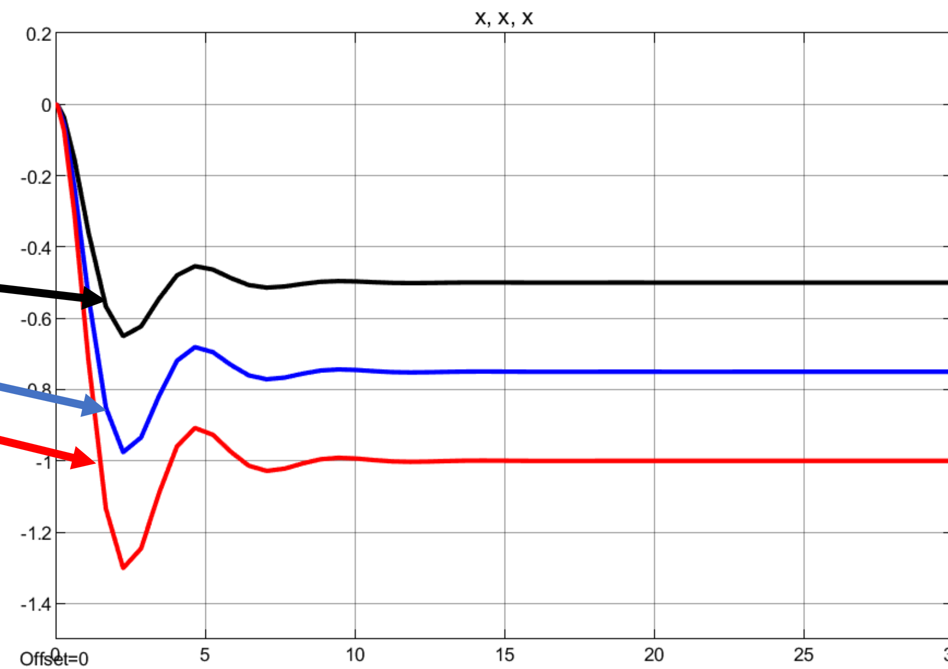


- Consider the performance of the system with varying:
  - External load
  - Spring constant
  - Damping ratio
  - Mass

Varying (external) load

Suppose:

- The load is
  - 10 N
  - 15 N
  - 20 N

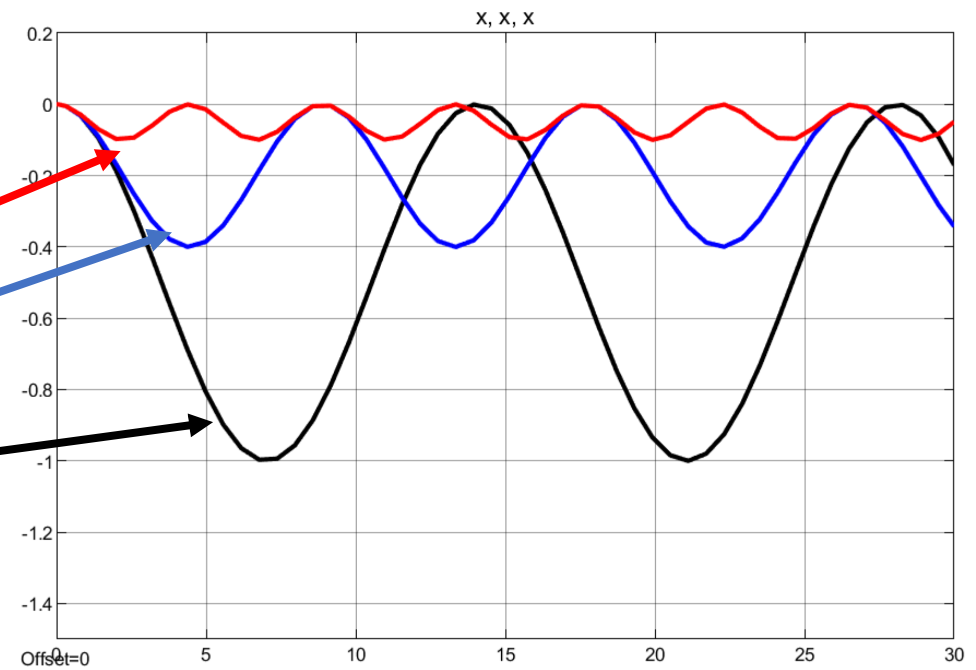


- The jack’s mass is 10 kg.
- The jack’s damping ratio is 10.
- The jack’s spring constant is 20 N/m.

Varying spring constant

Suppose:

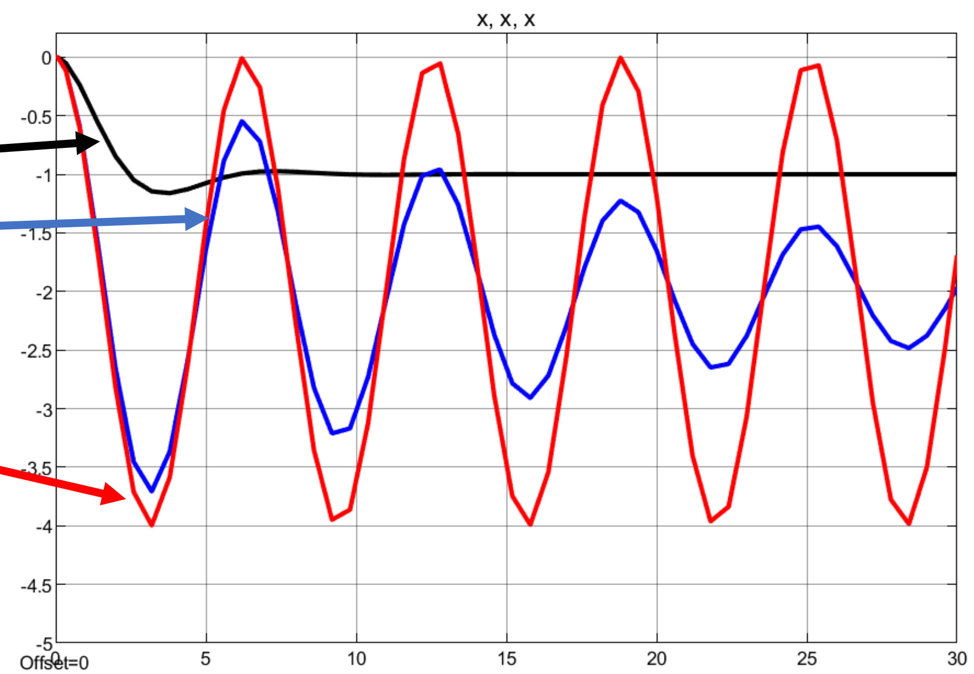
- The load is 1 N.
- The jack's mass is 10 kg.
- The jack's damping ratio is 0.
- The jack's spring constant is
    - 20 N/m
    - 5 N/m
    - 2 N/m



Varying damping ratio

Suppose:

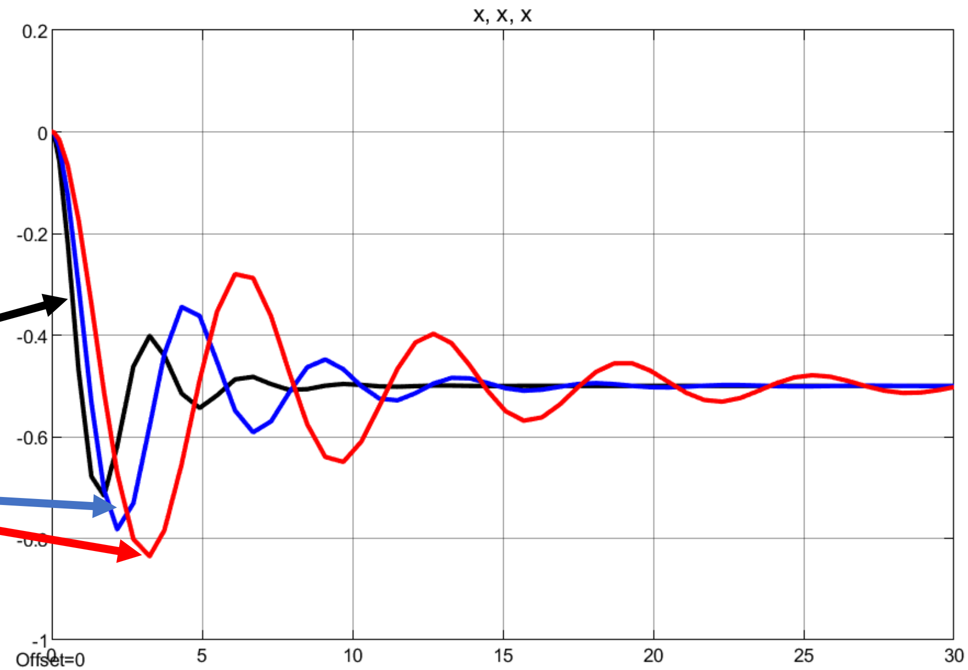
- The load is 10 N (increased from 10 N).
- The jack's mass is 10 kg.
- The jack's spring constant is 20 N/m.
- The jack's damping ratio is:
  - 10
  - 1
  - 0



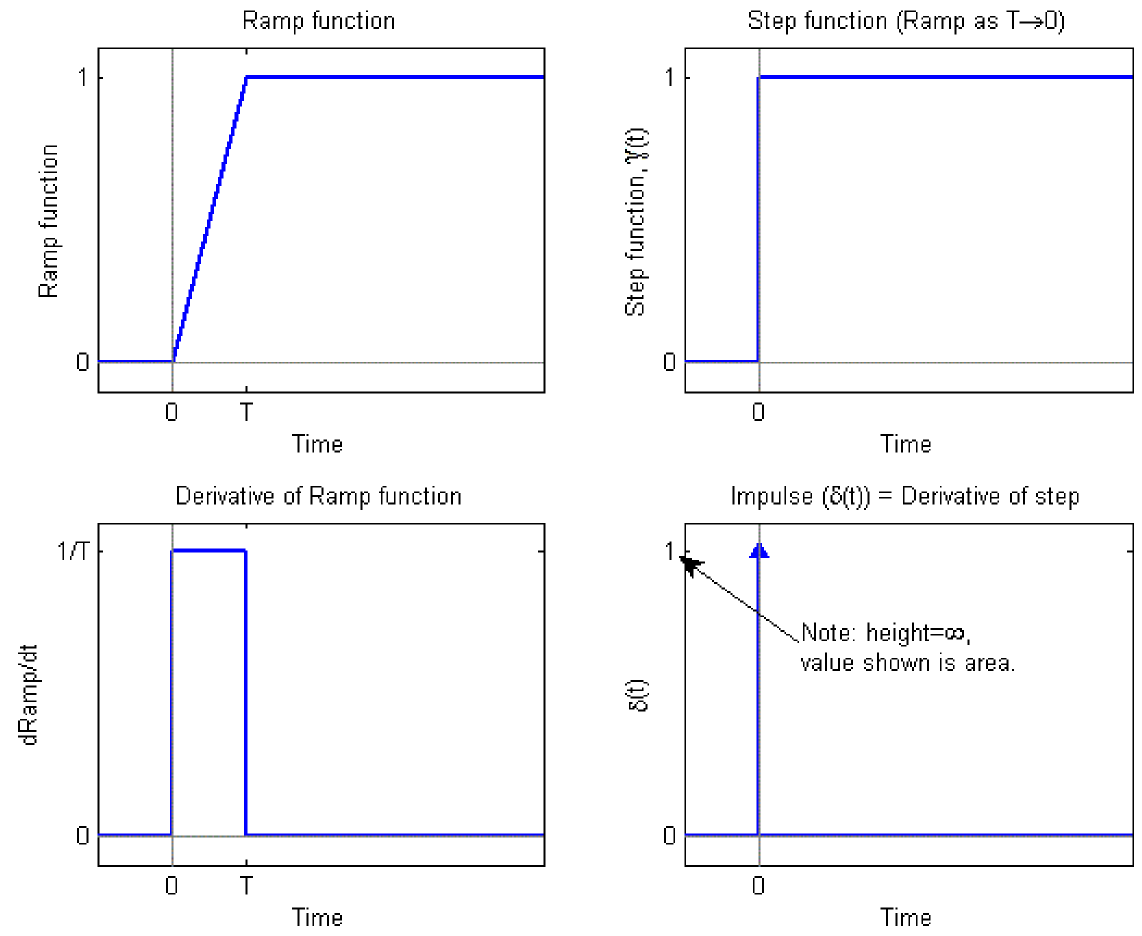
Varying mass

Suppose:

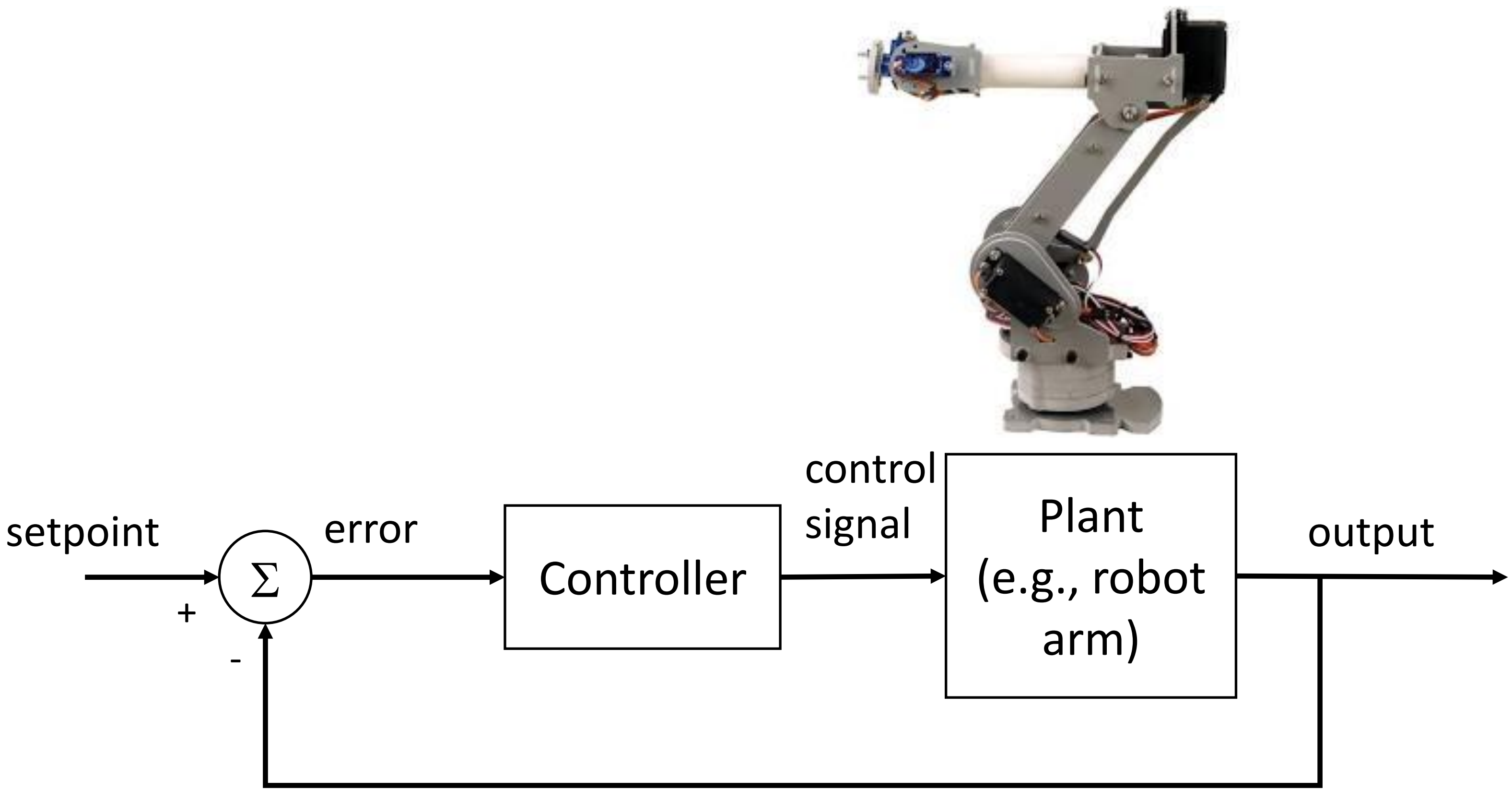
- The load is 10 N (increased from 10 N).
- The jack's damping ratio is 5.
- The jack's spring constant is 20 N/m.
- The jack's mass is
  - 5 kg
  - 10 kg
  - 20 kg



Load comes in various functions with respect to time



Feedback Control



Bang Bang Control

One of the simplest control

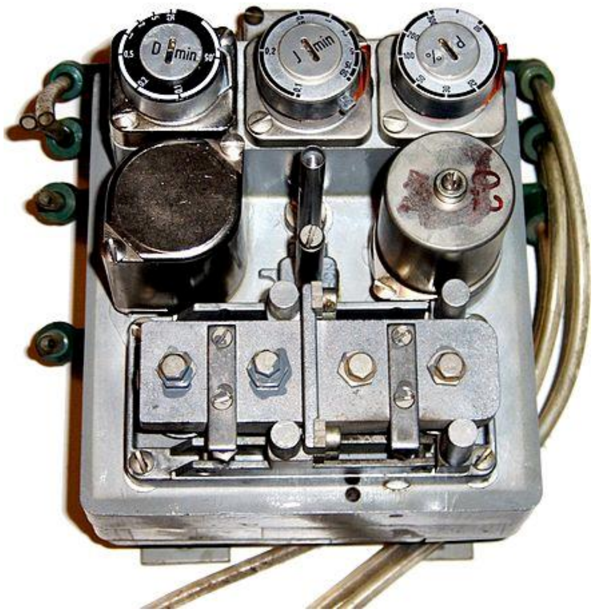
- When the position of the arm is lower than the setpoint, the motor is fully powered up.
- When the position of the arm is higher than the setpoint, the piston is fully powered down.
- This type of controller is called **Bang-bang Control**.

PID Control

'Control your own destiny or someone else will' - Jack Welch

PID Control

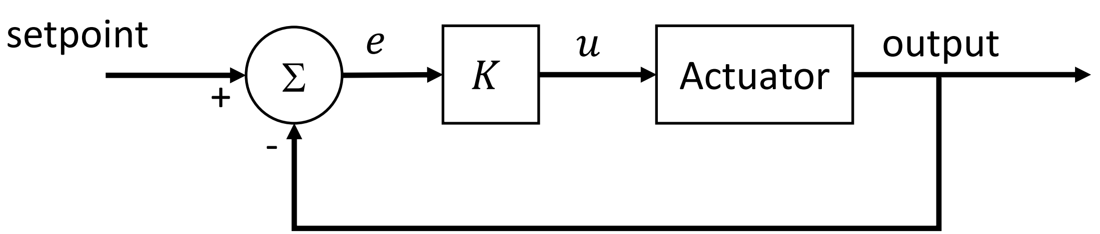
- PID control is the most common form of feedback control.
- PID control has been around since 1940s.
- Advantage:
  - Simple
  - Efficient
  - Effective



pneumatic controller using PID

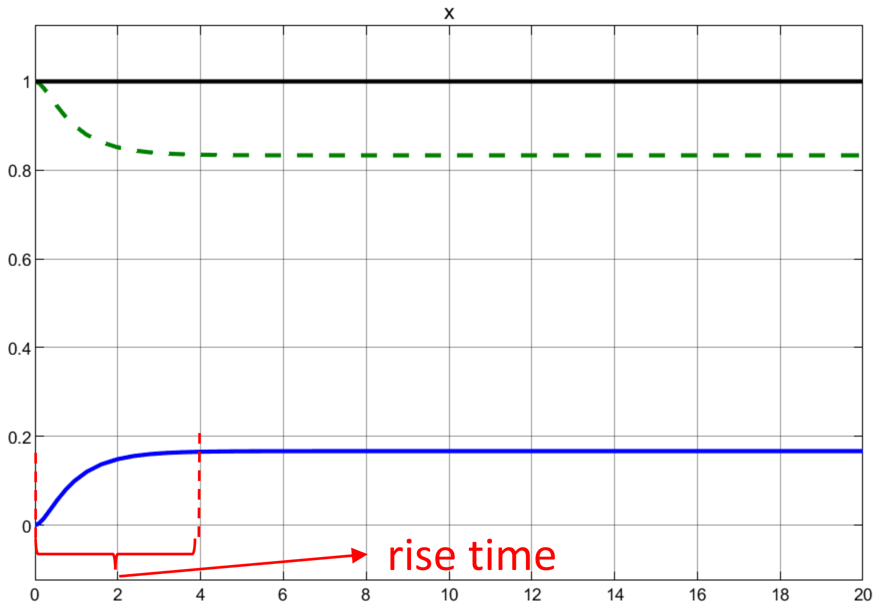
Proportional Control (P Control)

- Control signal is proportional to the error.
- The controller gain is K.
- The control signal is $u = Ke$.
- Large e, large u. Small e, small u.
- Designing P Control involves determining the controller gain K.



NUS National University of Singapore

P control, $K = 1$



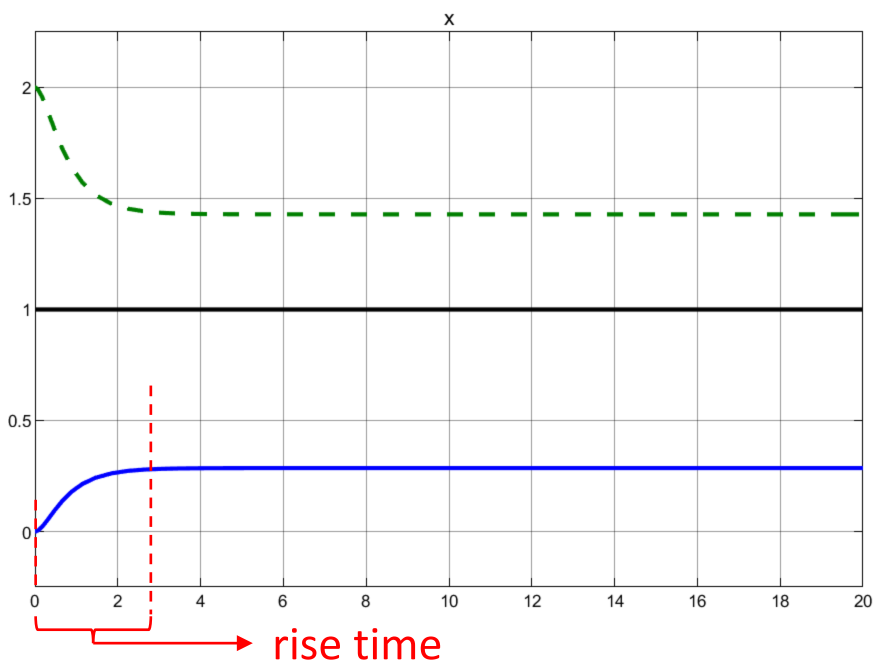
steady state error

output

setpoint

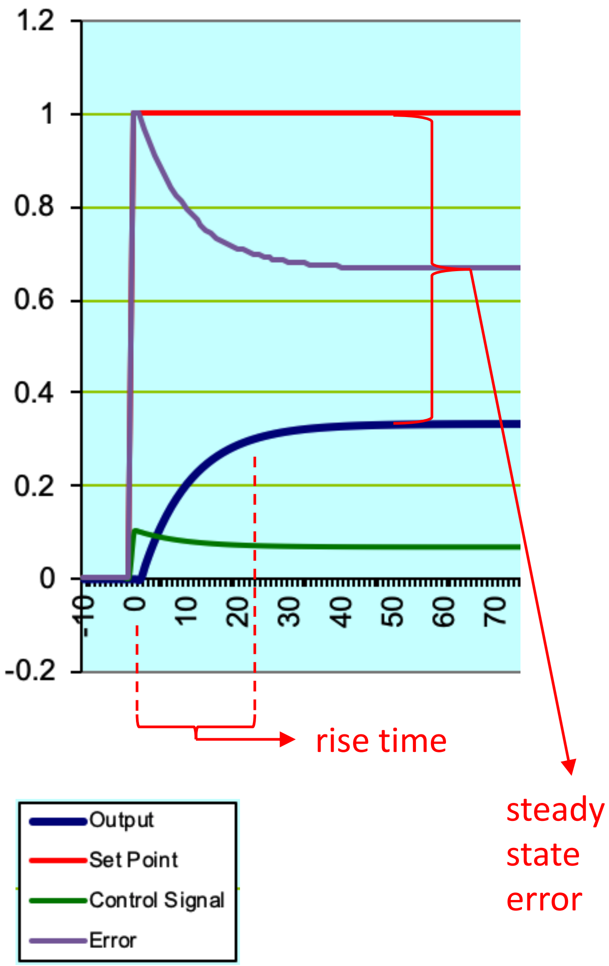
control signal

P control, $K = 2$

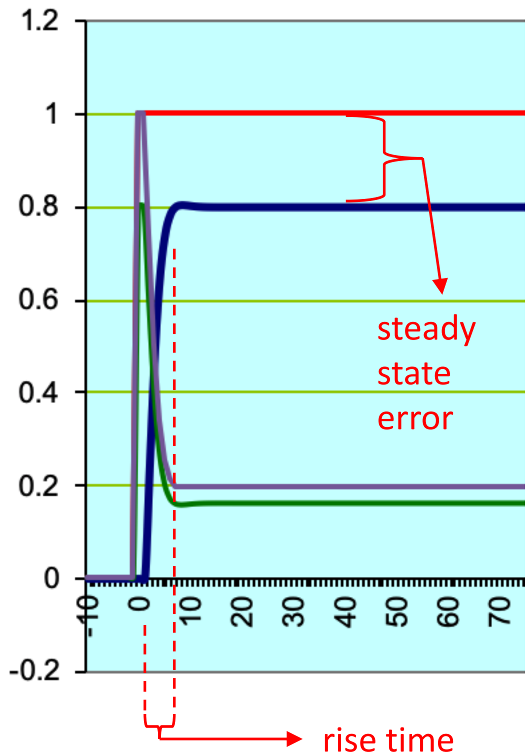


steady state error

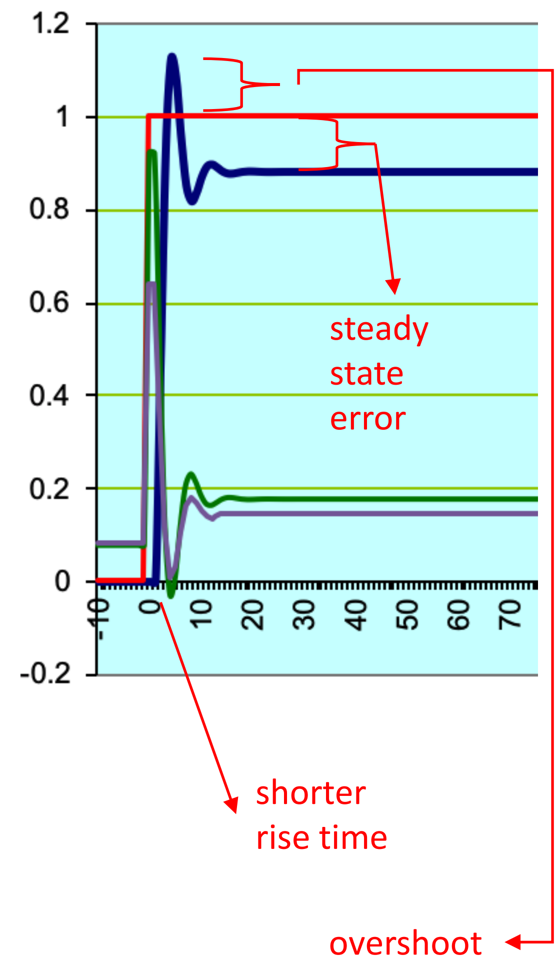
P control, $K = 0.1$



P control, $K = 0.8$



P control, $K = 1.5$



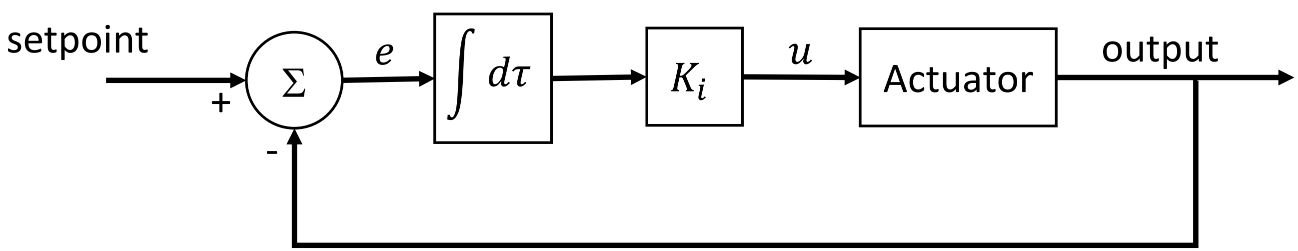
Proportional Control (P)

- Higher gain K results in:
  - Faster control action, i.e. smaller rise time
  - Smaller steady state error, although will not reach zero
  - Increase the tendency towards oscillation

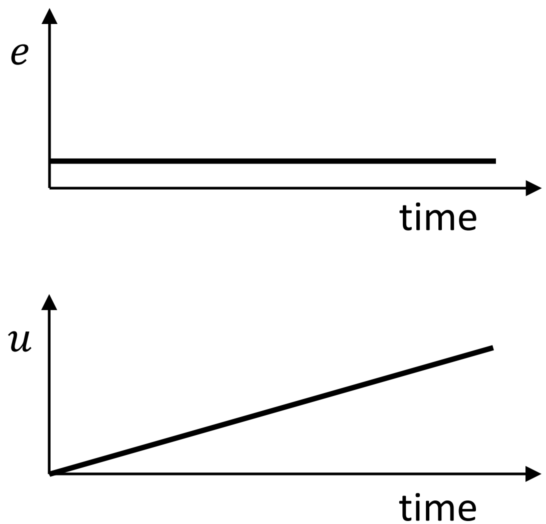
- Advantages:
  - Fast
- Disadvantages:
  - Cannot eliminate steady state error

Integral Control (I Control)

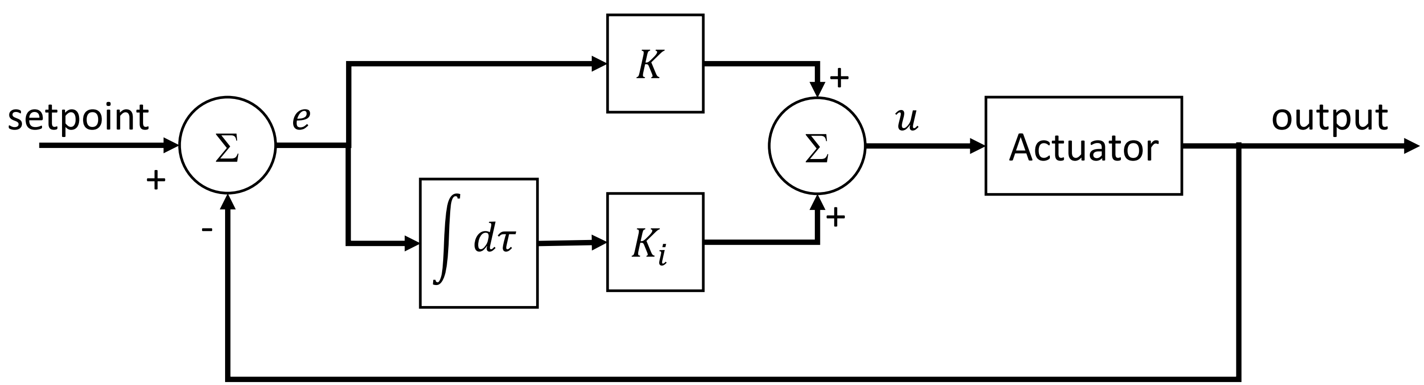
- Control signal is proportional to the cumulative error.
- The controller gain is K_i.
- The control signal is $u = K_i \int_0^t e(\tau) d\tau$.
- Even a small e will eventually result in a large u.
- Designing I Control involves determining the controller gain K_i.

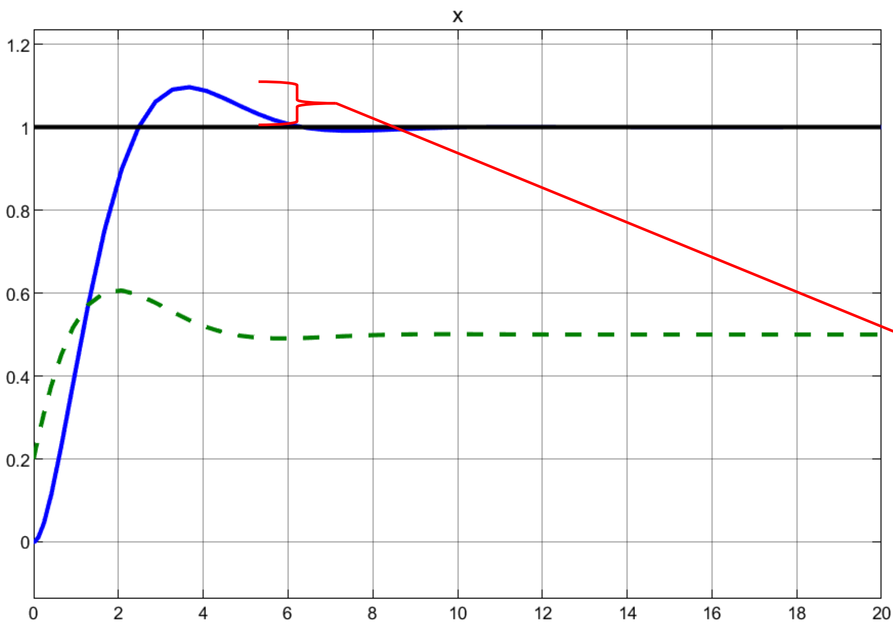


PI Control



- The chart on the left demonstrates how a small e will eventually result in a large u.
- I Control is used with P Control, otherwise it is impractical due to its slowness.

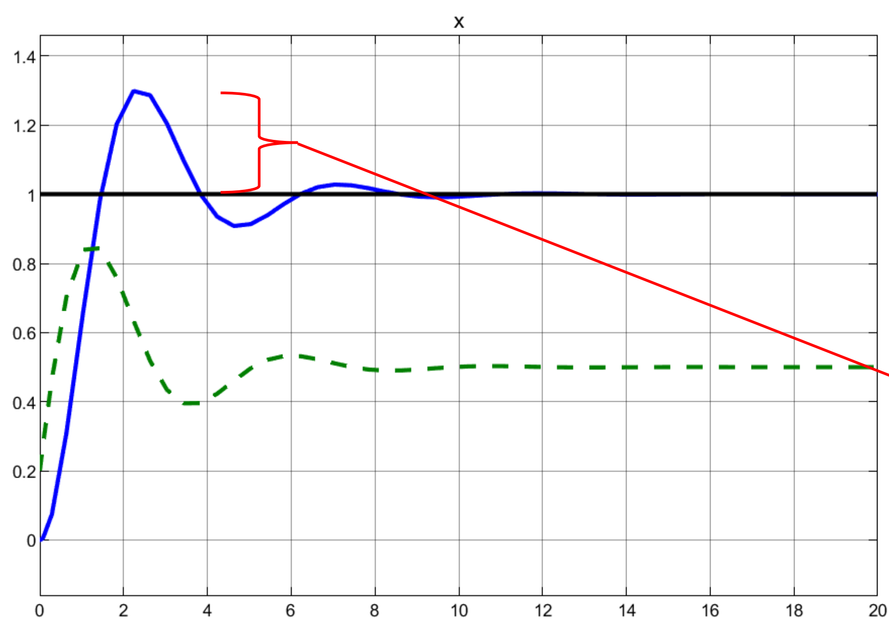




PI control,
$$K = 2, K_i = 5$$

Overshoot ≈ 10%

output
setpoint
control signal



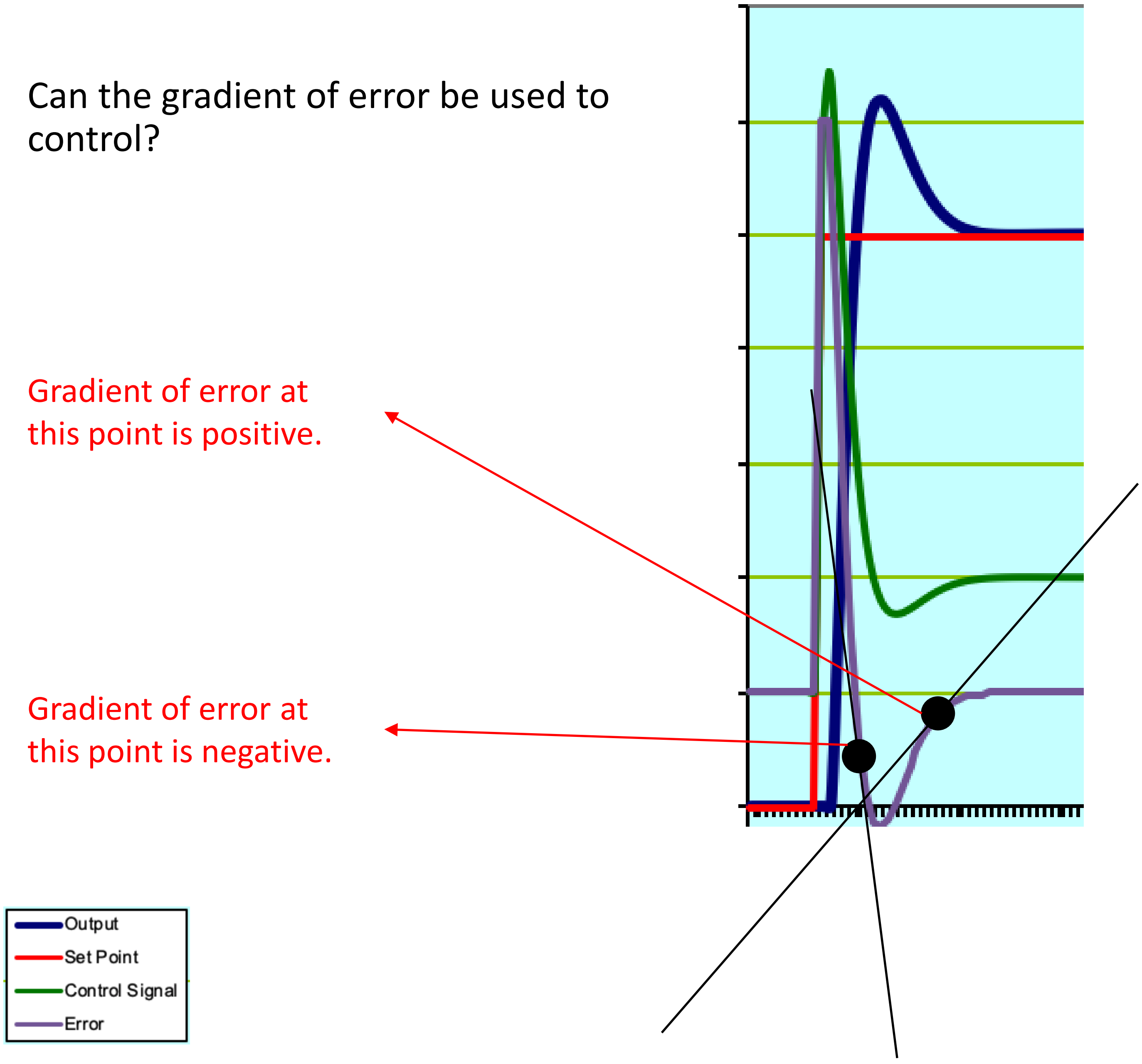
PI control,
$$K = 2, K_i = 10$$

Overshoot ≈ 30%

Can the gradient of error be used to control?

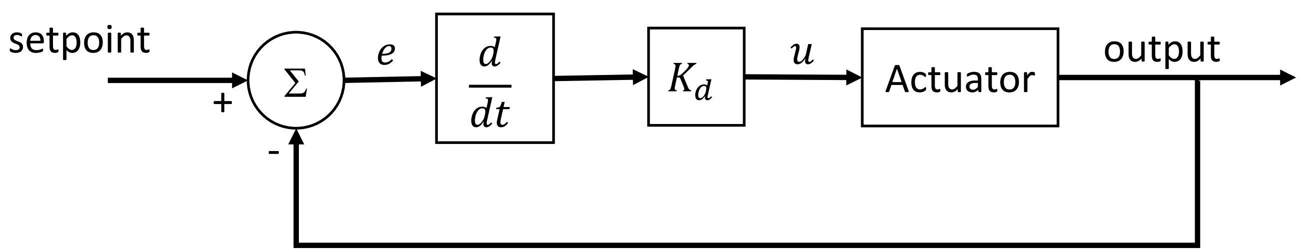
Gradient of error at this point is positive.

Gradient of error at this point is negative.



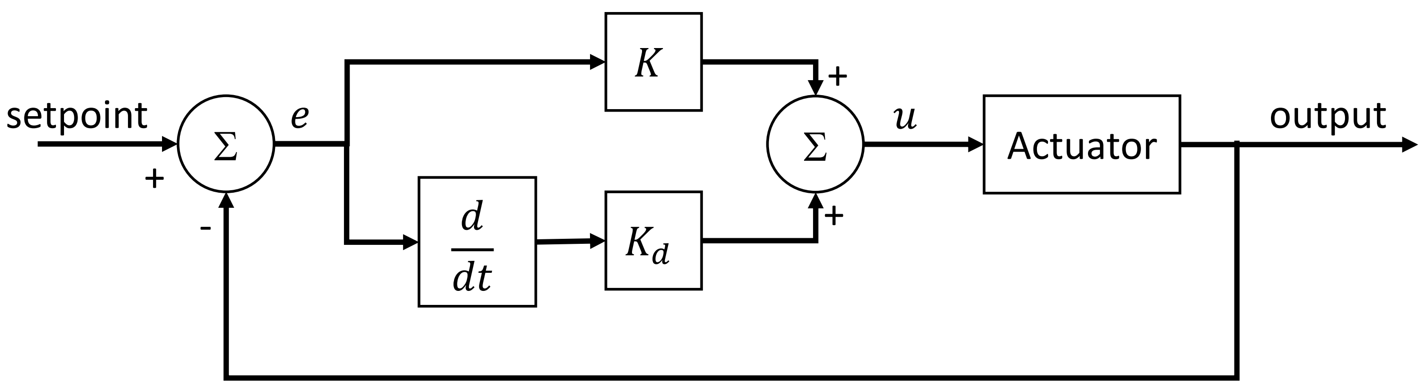
Derivative Control (D Control)

- Control signal is proportional to the change of error.
- The controller gain is K_d.
- The control signal is $u = K_d \frac{de(t)}{dt}$.
- As the output approaches the setpoint, the error gets smaller.
- Designing D Control involves determining the controller gain K_d.



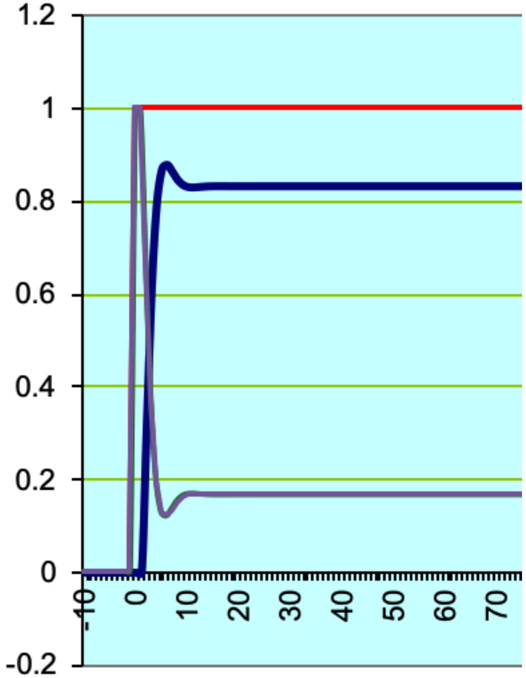
PD Control

- D Control is used with P Control, otherwise it is impractical as constant large error will not be corrected.



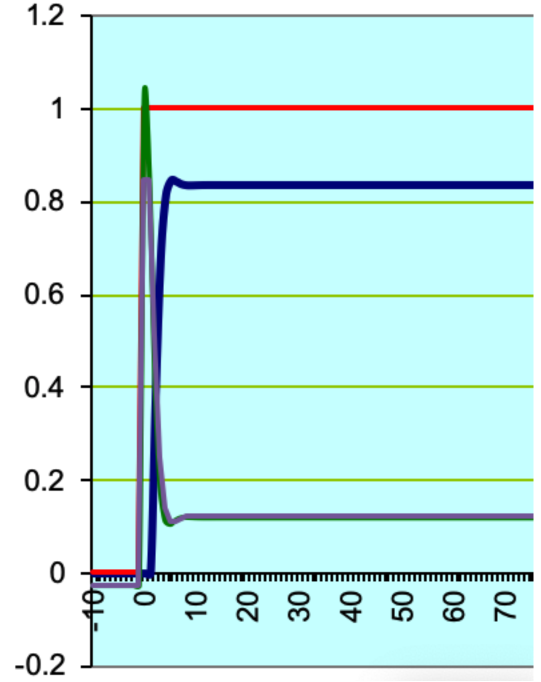
P control,

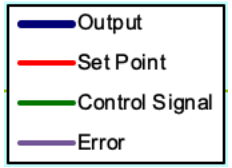
$K = 1.0$



PD control,

$K = 1.0, K_d = 0.3$





PID Control

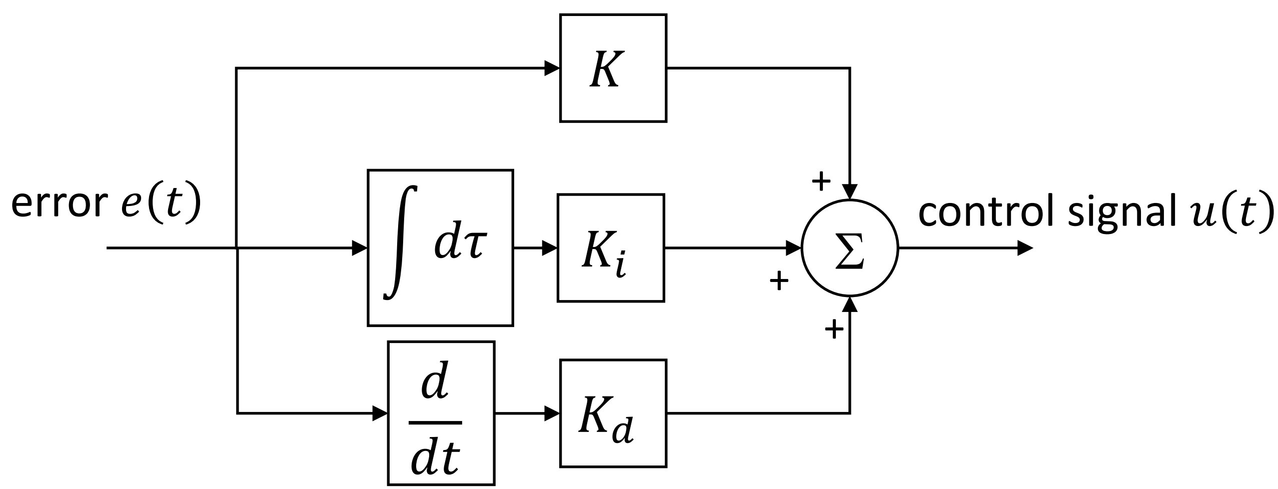
Description: $u(t) = Ke(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$

Proportional term:
K is the proportional gain.

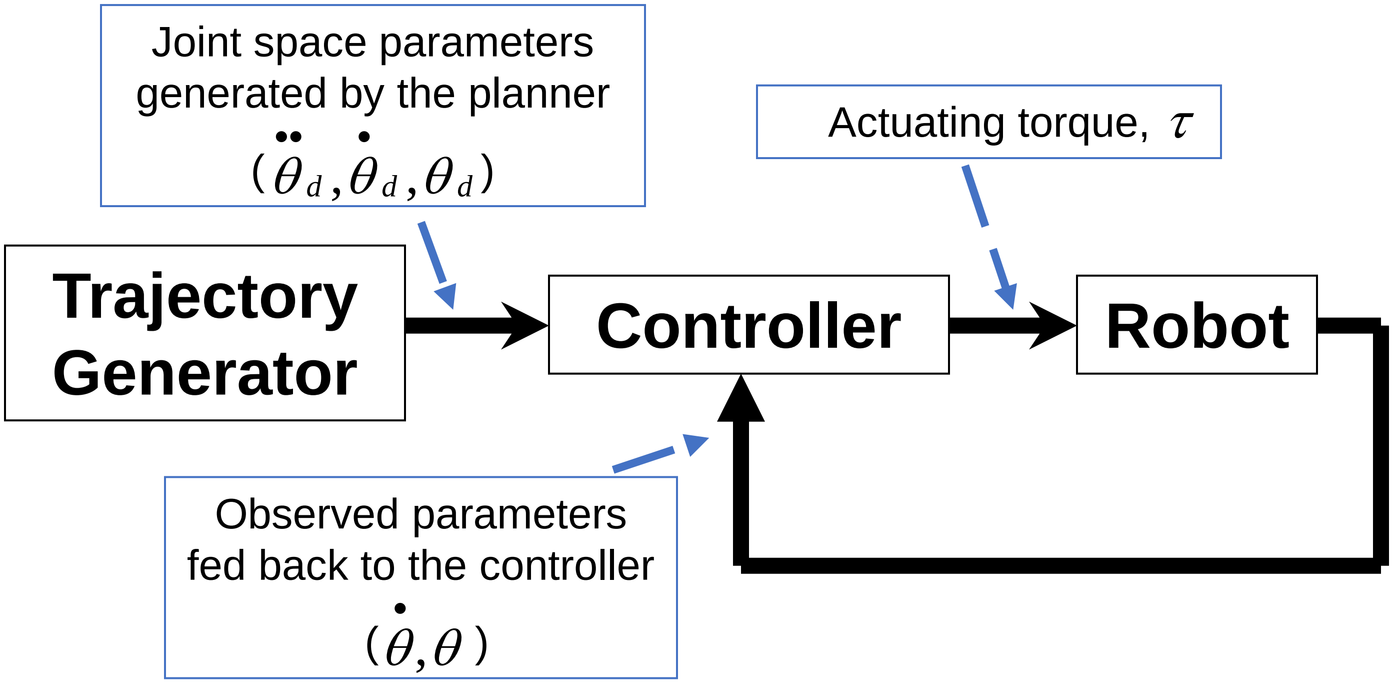
Integral term:
K_i is the integral gain.
$K_i = \frac{K}{T_i}$ where T_i is the integral time.

Derivative term:
K_d is the derivative gain.
$K_d = KT_d$ where T_d is the derivative time.

Block Diagram:



Robot Control System



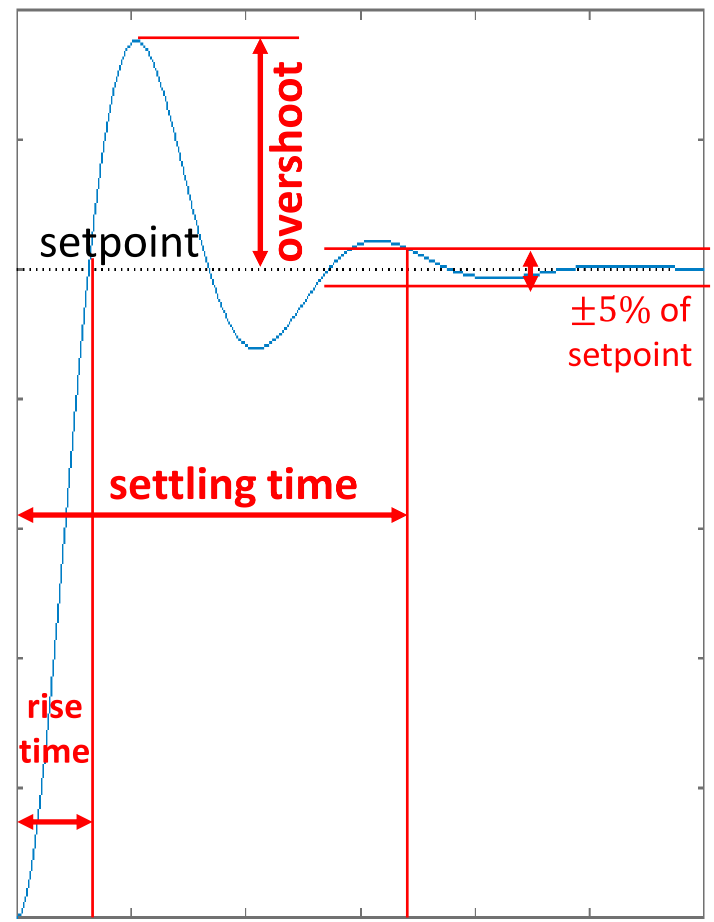
High level block diagram of a robot control system

Evaluation of Closed Loop Control

- How fast the closed loop control responds to the command or inputs
- Whether the closed loop control is stable, and how much dynamic variation it takes to make the system unstable
- How sensitive the closed loop control is to the changes in the parameters (ideally, only sensitive to input)

Effect of PID Parameters on Performance

Parameter increase	Rise time	Overshoot	Settling time	Steady-state error
K	↓	↑	Small change	↓
K_i	↓	↑	↑	Eliminate
K_d	Small change	↓	↓	Small change



Evaluation of Closed Loop Control:

- Rise time
- Overshoot
- Settling time
- Steady state error

Constraints in Implementation of PID Control

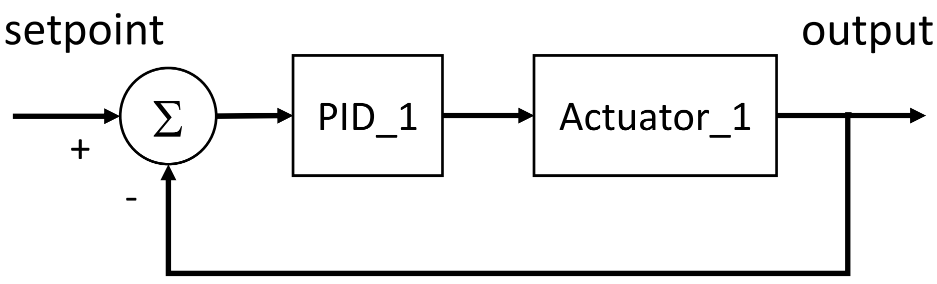
Real actuators have limitation:

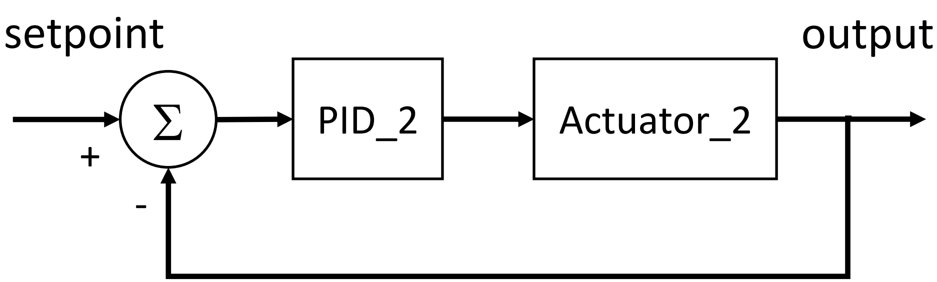
- Maximum amplitude, e.g. a motor cannot run more than a maximum speed
- Maximum rate of change, e.g. a motor cannot change its speed too fast beyond a threshold value
- Other limitation?

Dealing with actuators’ limitation:

- Reduce the performance requirements such that the controllers do not send signals beyond actuators’ limits

Parallel PID Control





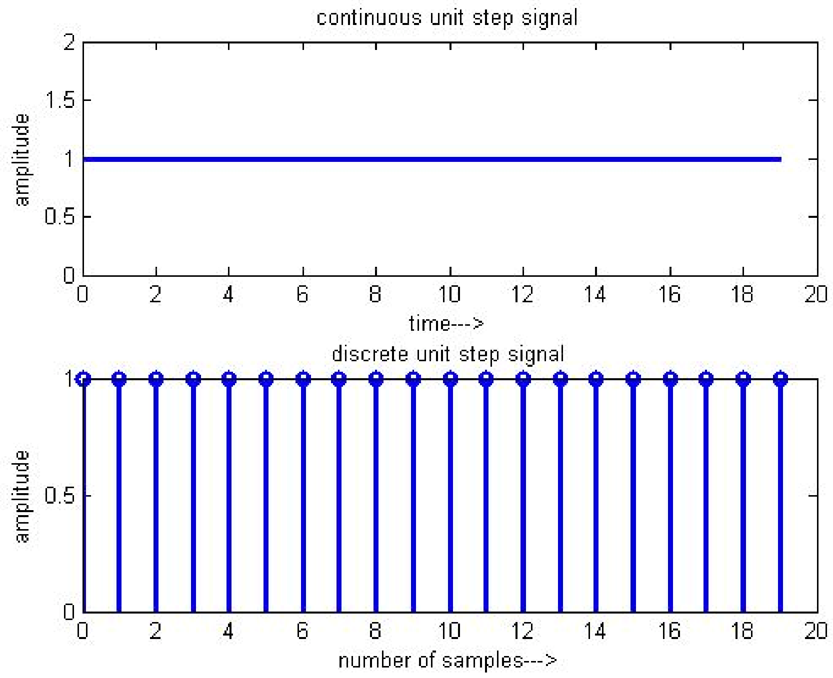
- Suitable to two (or more) decoupled systems, although they may be housed in an embodiment
- Example: a platform with independent steering and forward-backward movement

Computer Implementation of PID Control

'*What we can control is our performance and our execution, and that's what we're going to focus on*' - Bill Belichick

Computer Implementation of PID Control

- Most controllers are nowadays implemented using computers; including in robots.
- In a computer, inputs and outputs are read and set in a certain **sampling period**.



Proportional Control (Digital)

- The proportional term is: $u_p = K(y_{SP} - y)$, where y_{SP} is the setpoint.
- In discrete term: $u_p(k) = K\big(y_{SP}(k) - y(k)\big)$
- In computer code: `u_p = K * (y_sp - y)`

Integral Control (Digital)

- The integral term is: $u_i = K_i \int_0^t e(\tau)d\tau$
- In discrete term: $u_i(k) = K_i \sum_{k=0}^{t} e(k) \cdot h$, where h is the sampling period.
- In computer code:

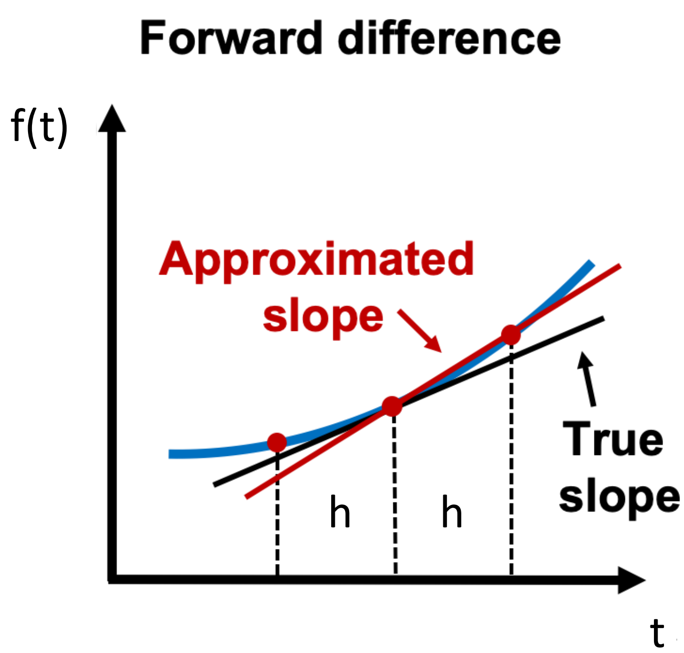
```
loop:
error = setpoint - output
cumulative_error += error * h
u_i = K_i * cumulative_error
```

Integration

- $I(t) = \int_0^t e(s)ds$
- $\frac{dI}{dt} = e$

Use the forward different approximation method:

- $\frac{I(t_{k+1}) - I(t_k)}{h} = e(t_k)$
- $I(t_{k+1}) = I(t_k) + h \cdot e(t_k)$



Derivative Control (Digital)

- The derivative term is: $u_d = K_d \frac{de}{dt}$
- In discrete term: $u_d(k) = K_d \frac{e(k)-e(k-1)}{h}$
- In computer code: `u_d = K_d * (e - e_previous) / h`

Thank you

'*You cannot always control what goes on outside. But you can always control what goes on inside*' - Wayne Dyer