Lecture 23

Introduction to Control

CS 3630

Building on slides from Prof. Harish Ravichandar

What is a control system?

Control System:

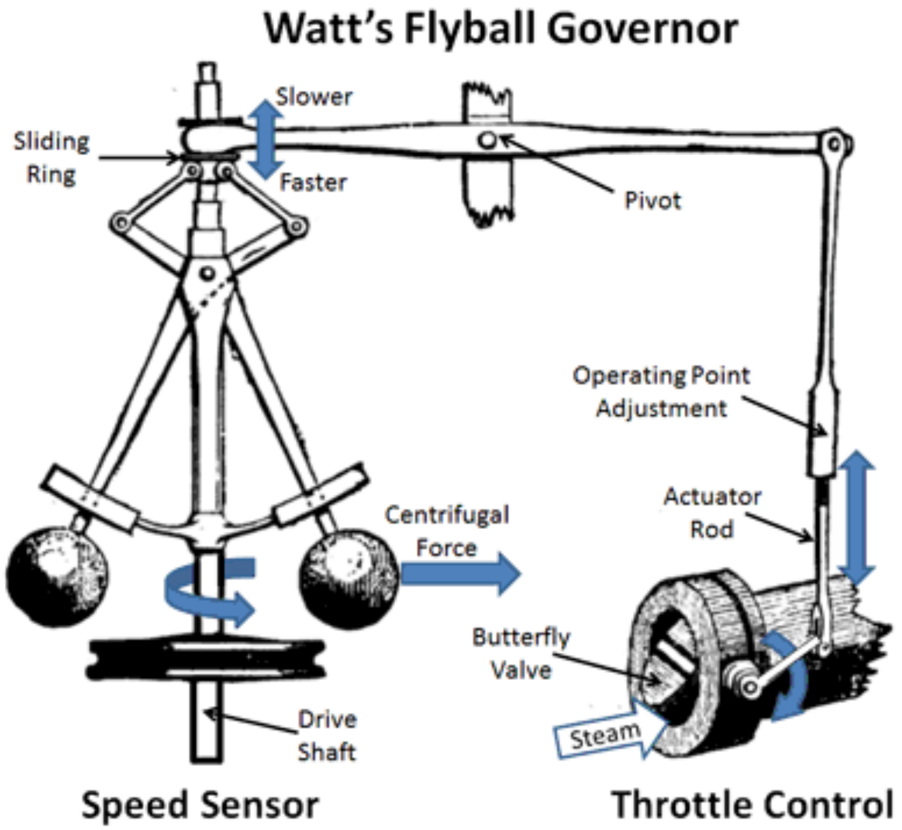
A mechanism that systematically alters the <u>future</u> states of a system.

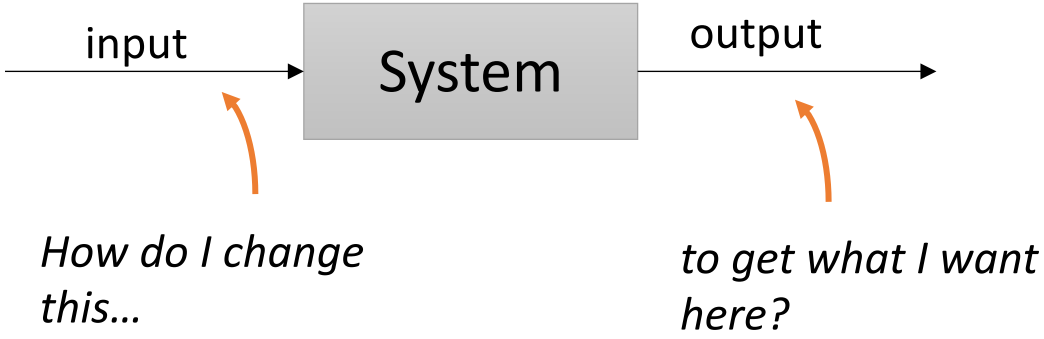
Control Theory:

A subfield of mathematics that studies strategies to control system behavior

Control systems have a rich history

- Started two thousand years ago!
  - Ancient water clock of Ktesibios in Alexandria Egypt
- First modern device in 17th century!
  - Christiaan Huygens built “governors” for windmills
- Popularized in 1788 due to steam engines
  - James Watt invented the Fly Ball Governor to regulate speed
- 19th century brought rigorous mathematical analysis to control systems
  - 1968 paper by J. C. Maxwell

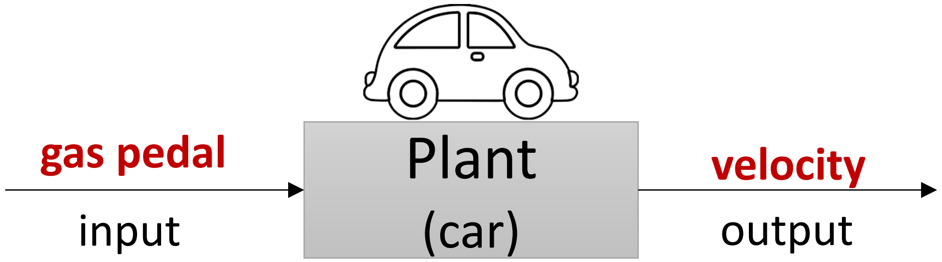




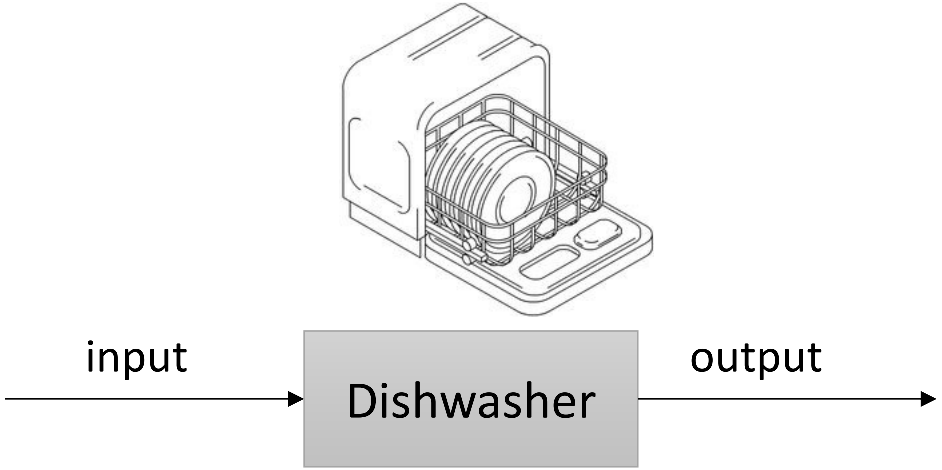
input
System
output
How do I change this…
to get what I want here?

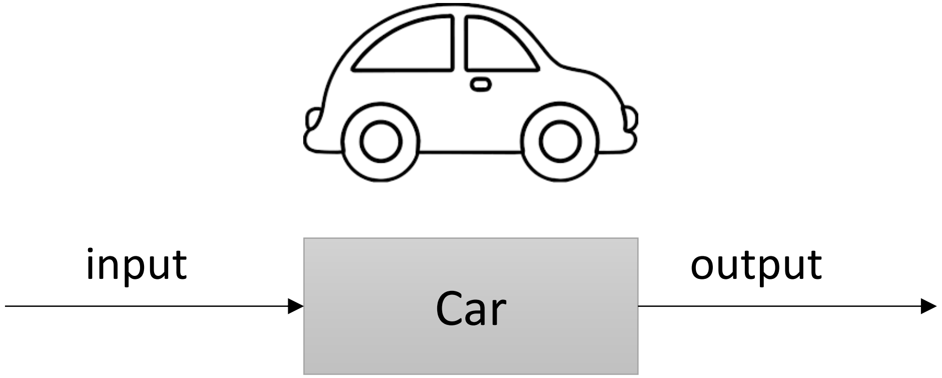
Open Loop Control System

Open Loop Control System

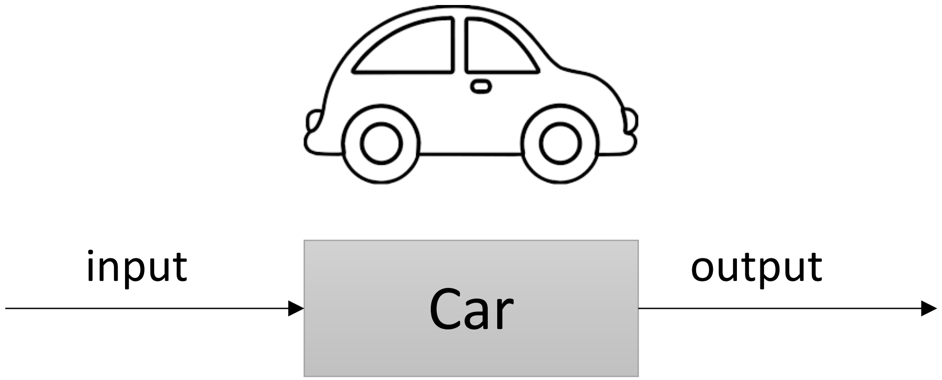


Open Loop Control Systems



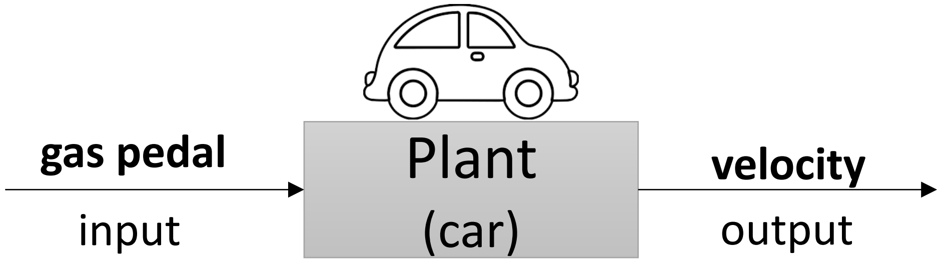


But open loop control is often insufficient

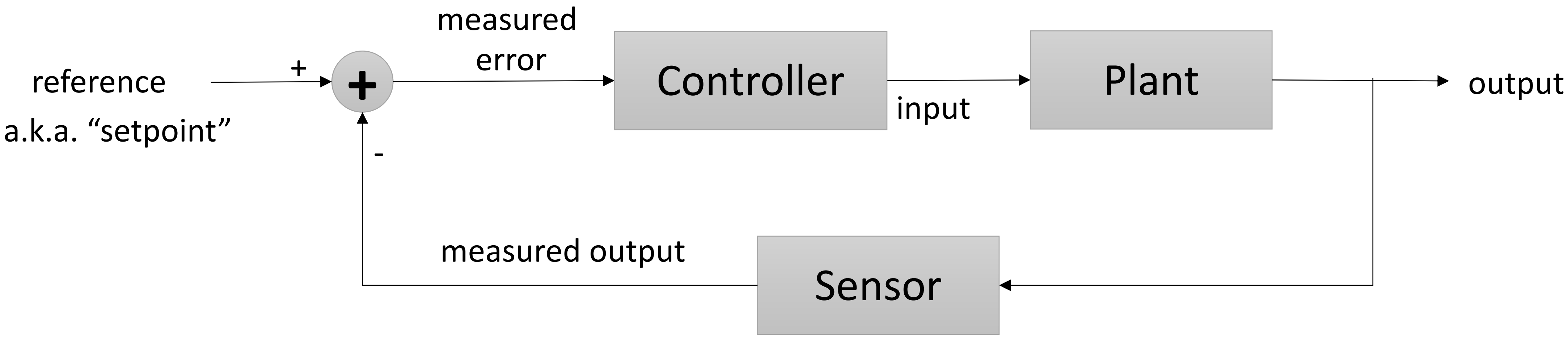


It has no feedback!

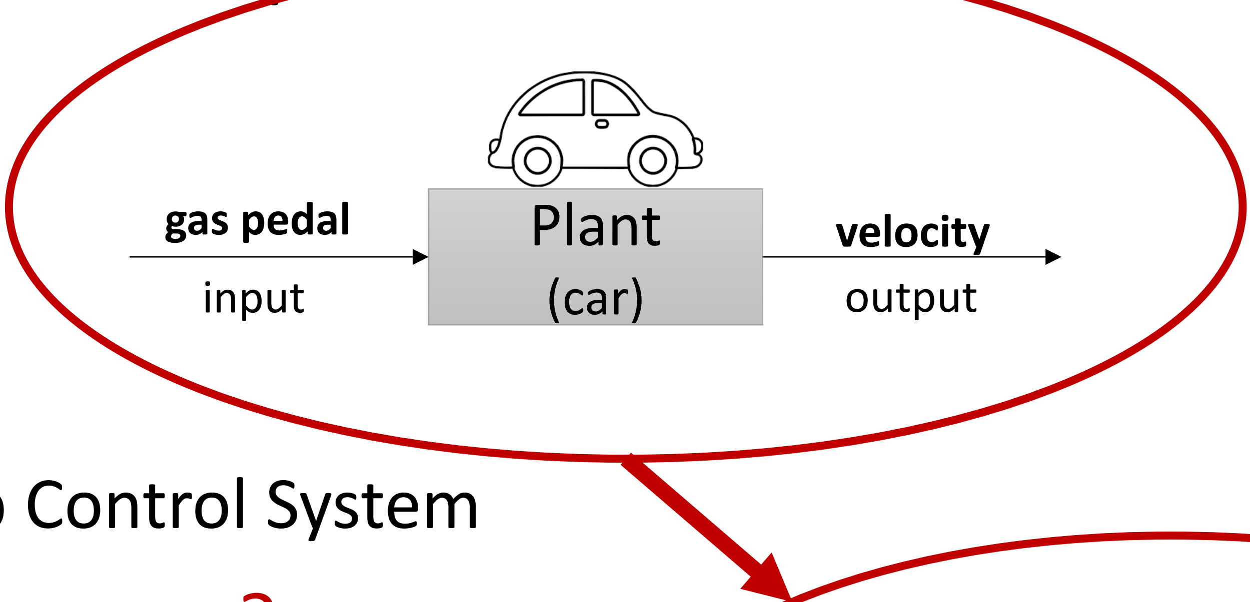
Open Loop Control System



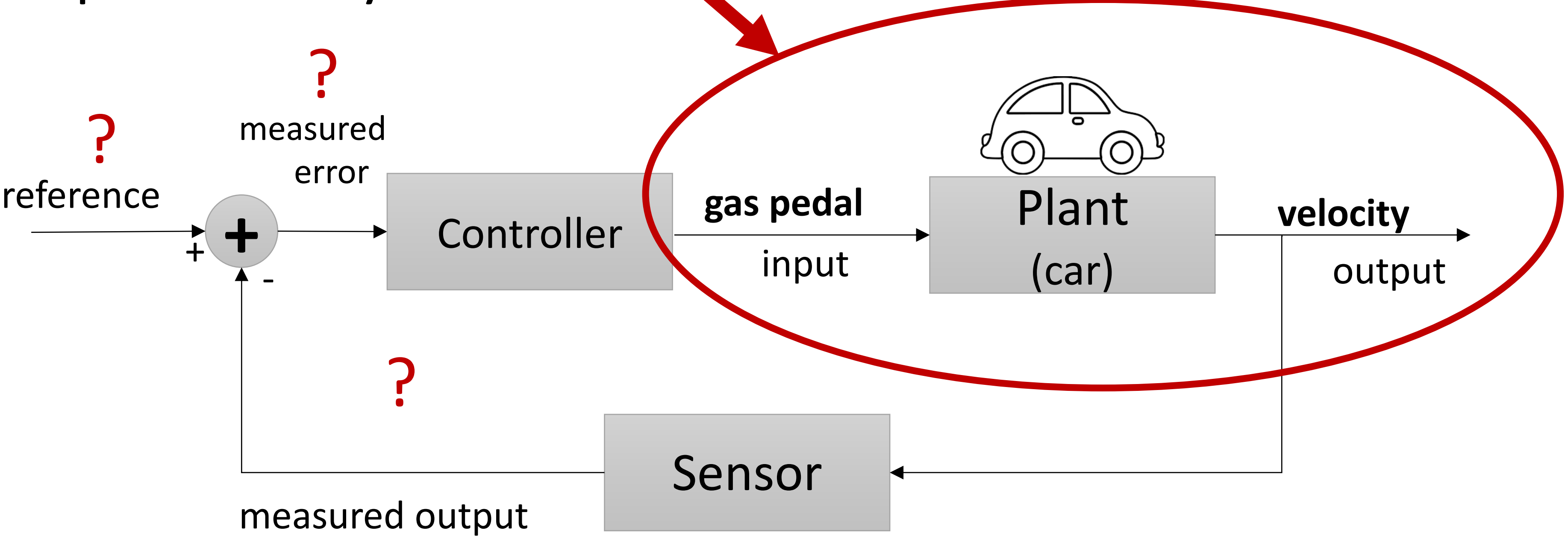
Closed Loop Control System



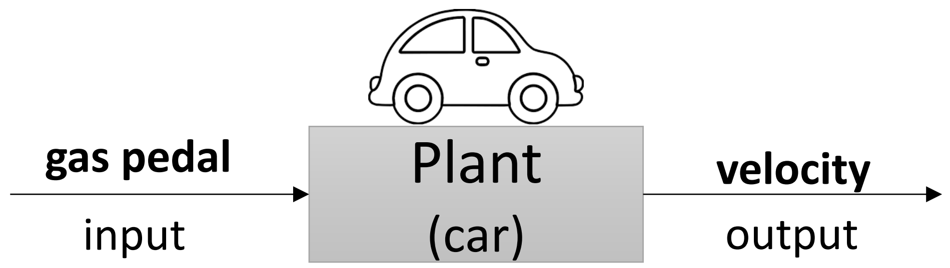
Open Loop Control System



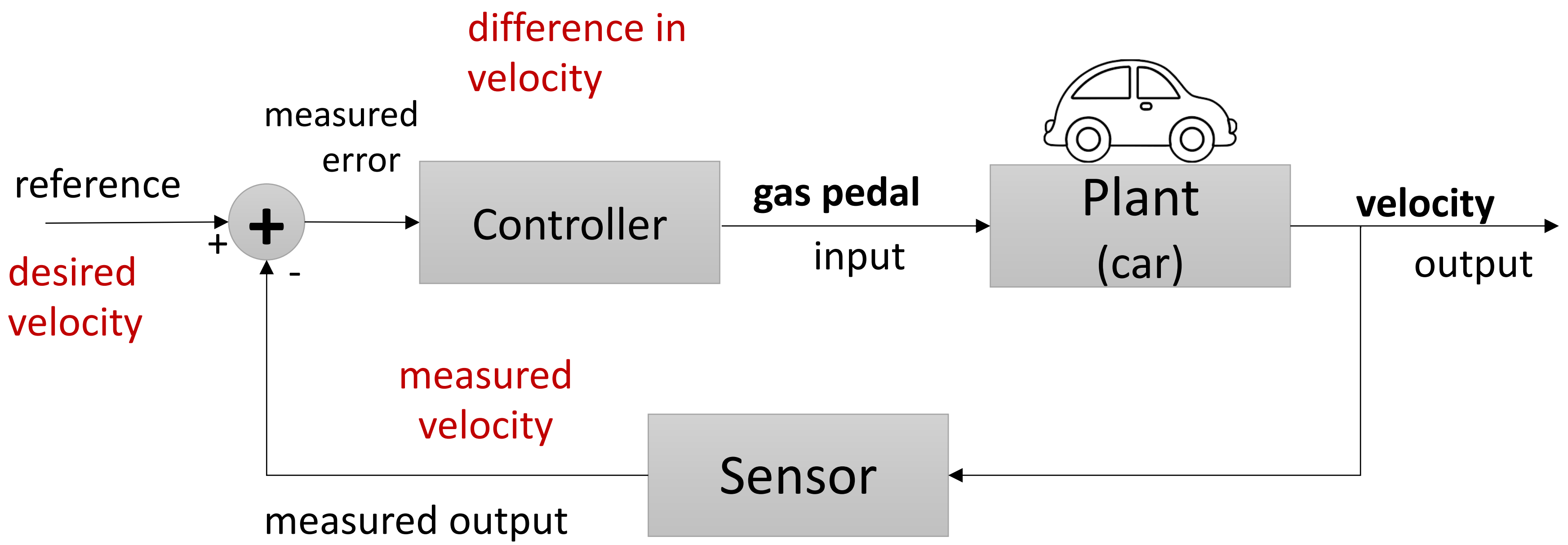
Closed Loop Control System



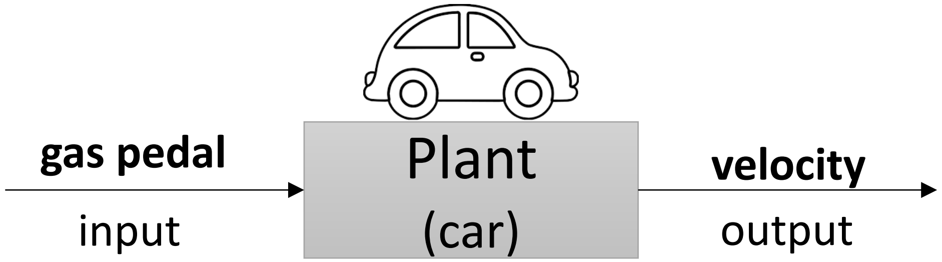
Open Loop Control System



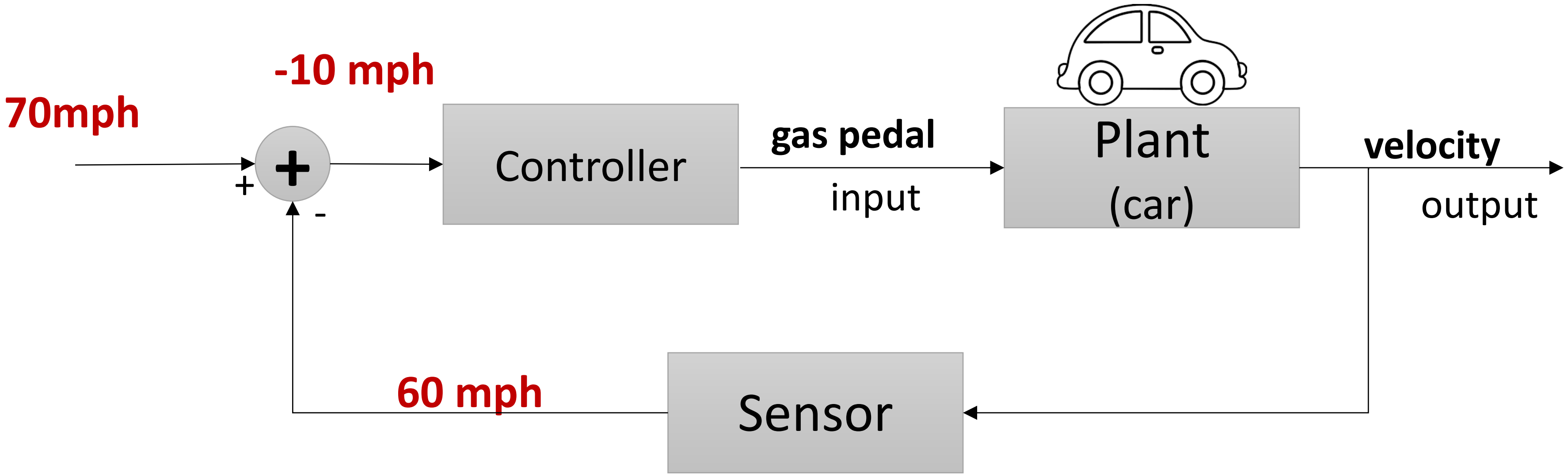
Closed Loop Control System



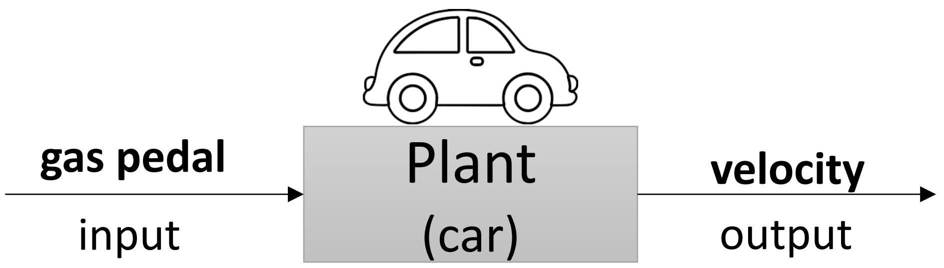
Open Loop Control System



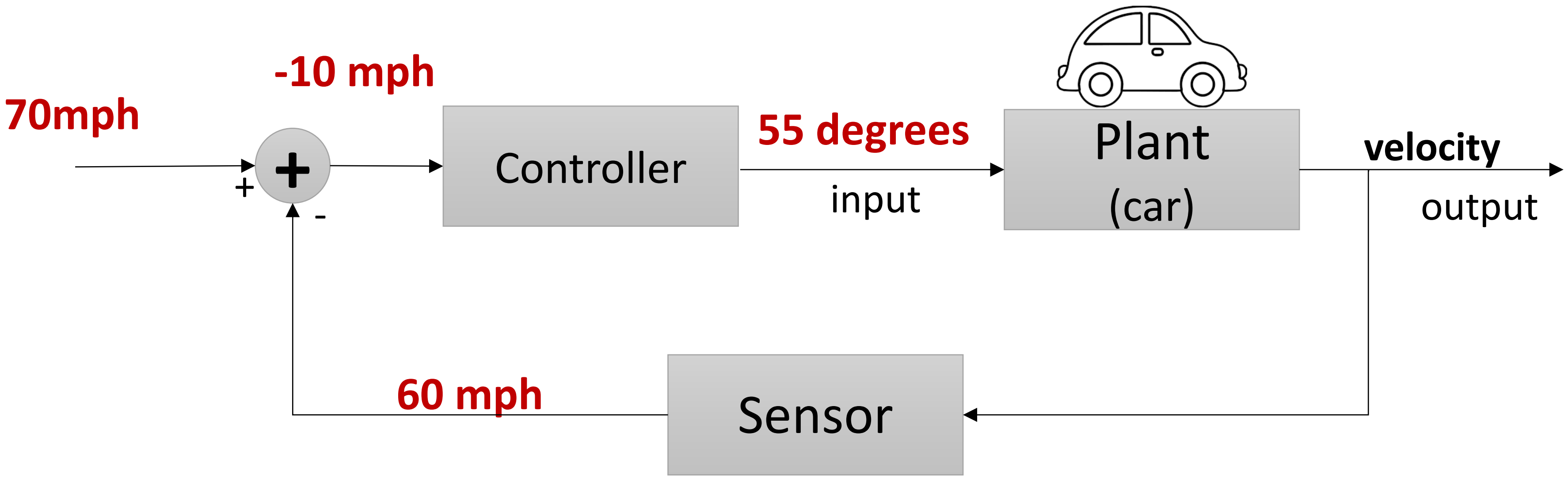
Closed Loop Control System



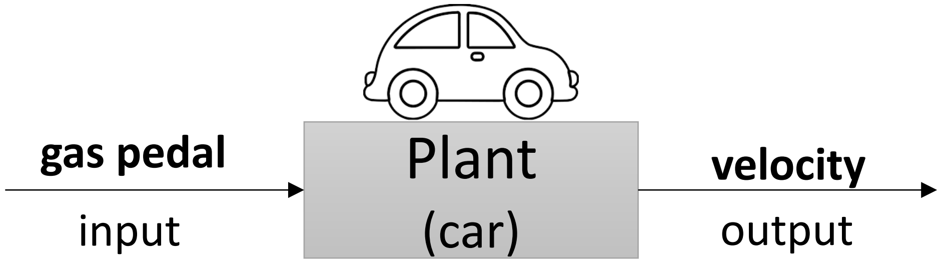
Open Loop Control System



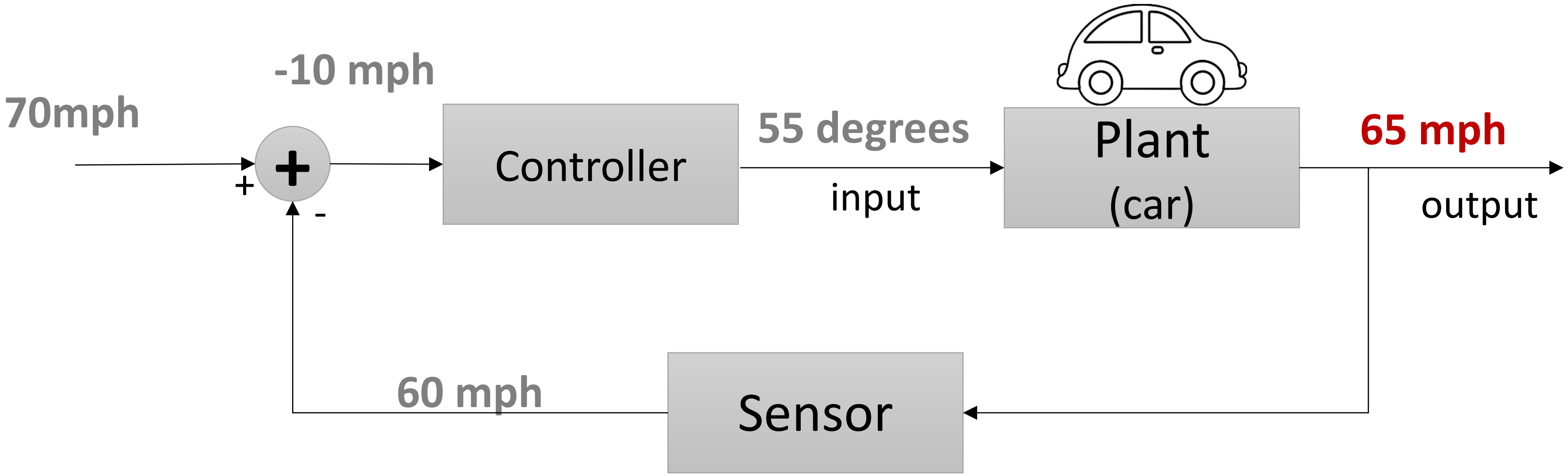
Closed Loop Control System



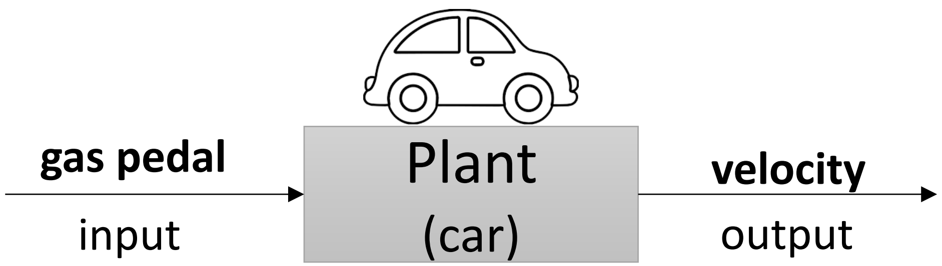
Open Loop Control System



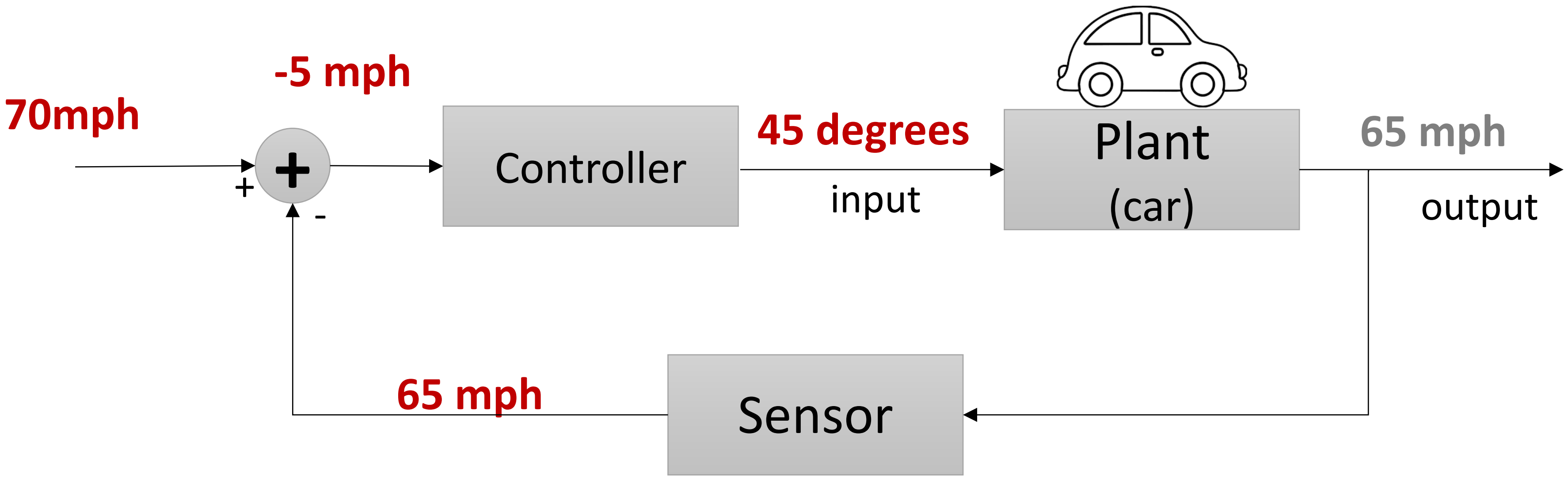
Closed Loop Control System



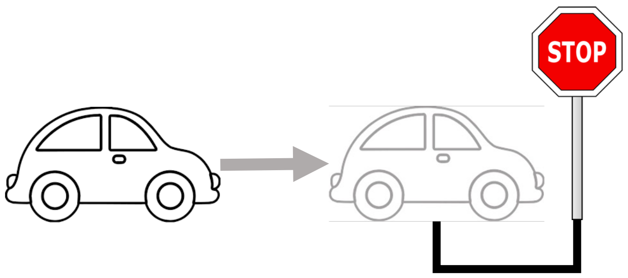
Open Loop Control System



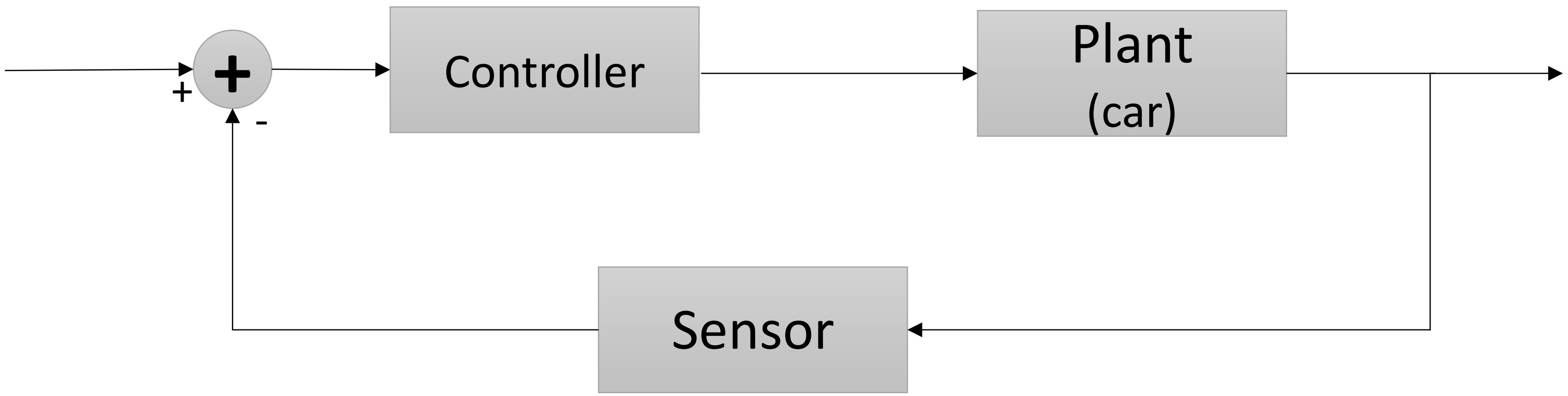
Closed Loop Control System



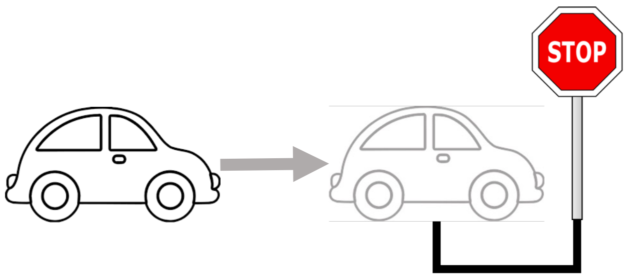
Example: *Automated Stop Sign Braking System*



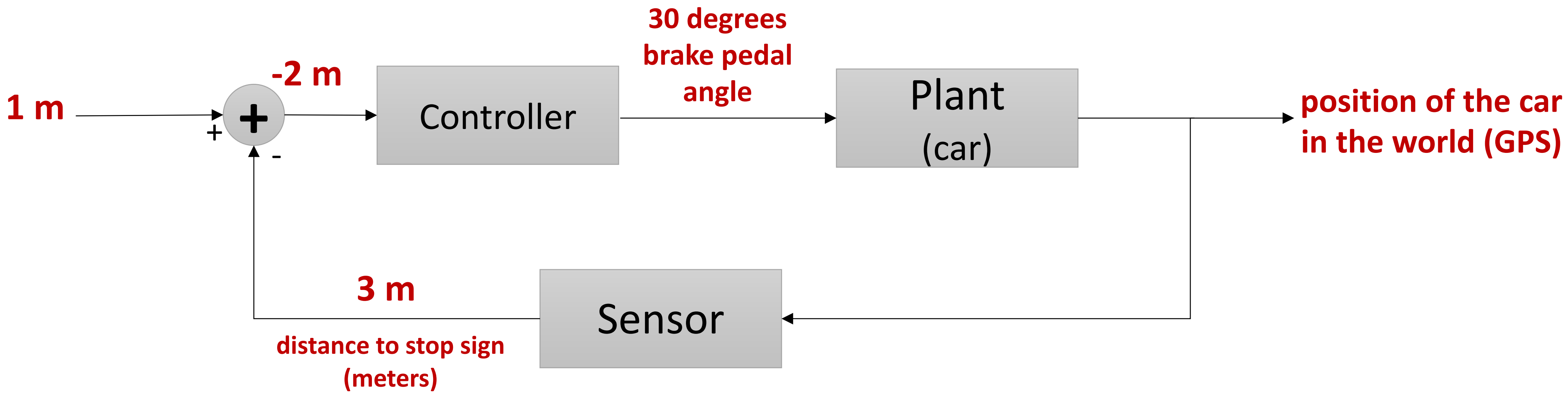
desired stopping distance of 1 meter



Example: *Automated Stop Sign Braking System*



desired stopping distance of 1 meter



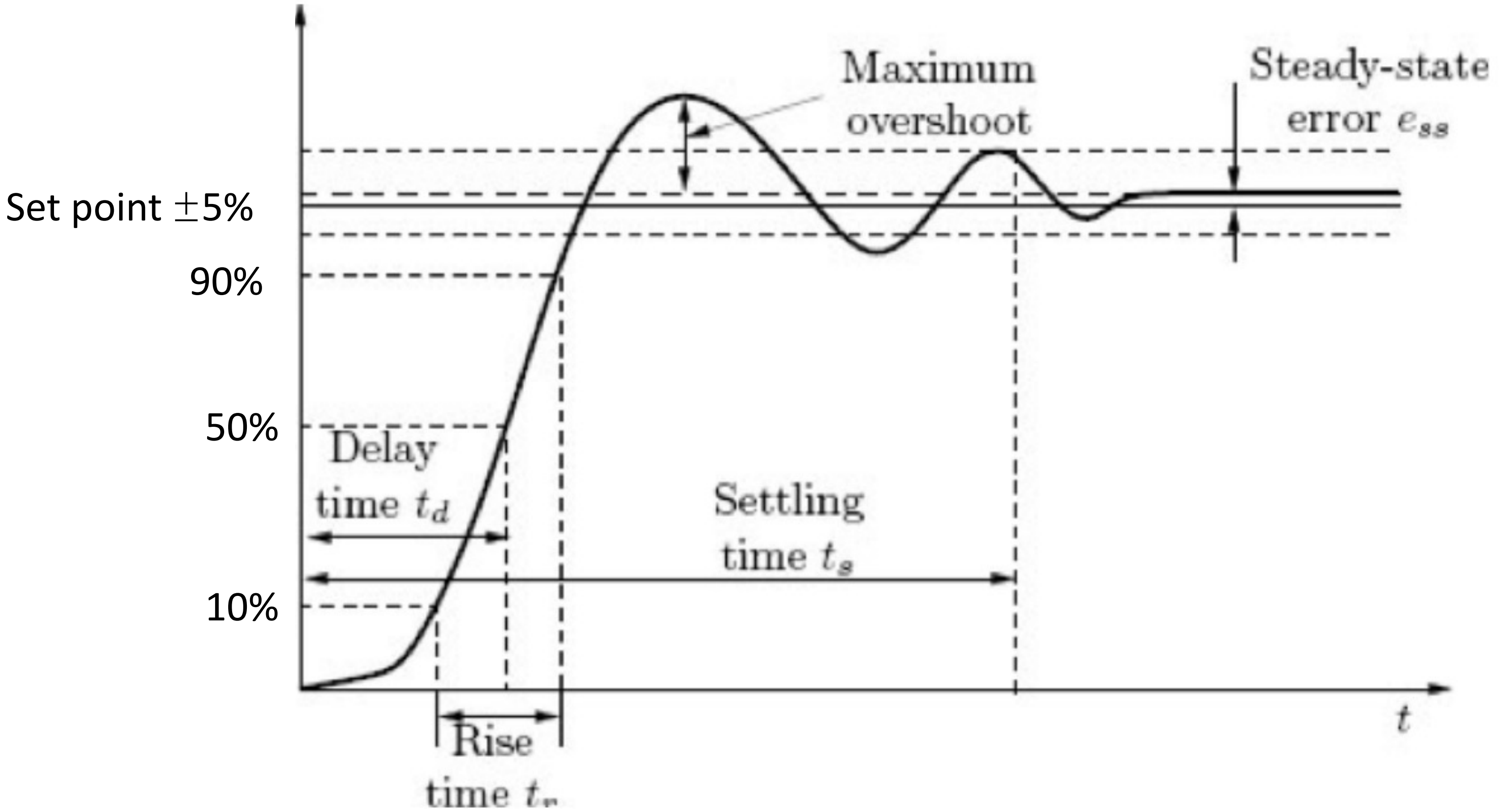
Open Loop Control: the control action from the controller is independent of the plant output (i.e., no sensing at all)

- Example: heating system running on a fixed timer

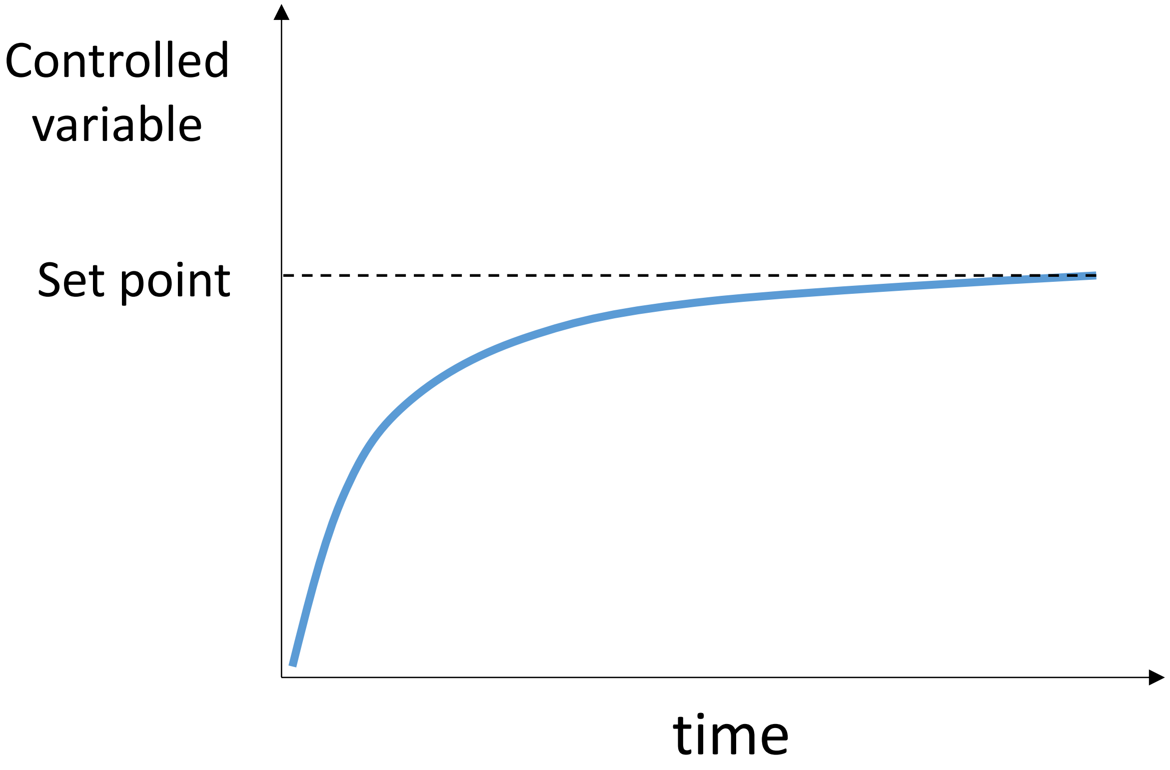
Closed Loop Control (Feedback Control): the control action from the controller is dependent on the plant output (i.e. sensing required)

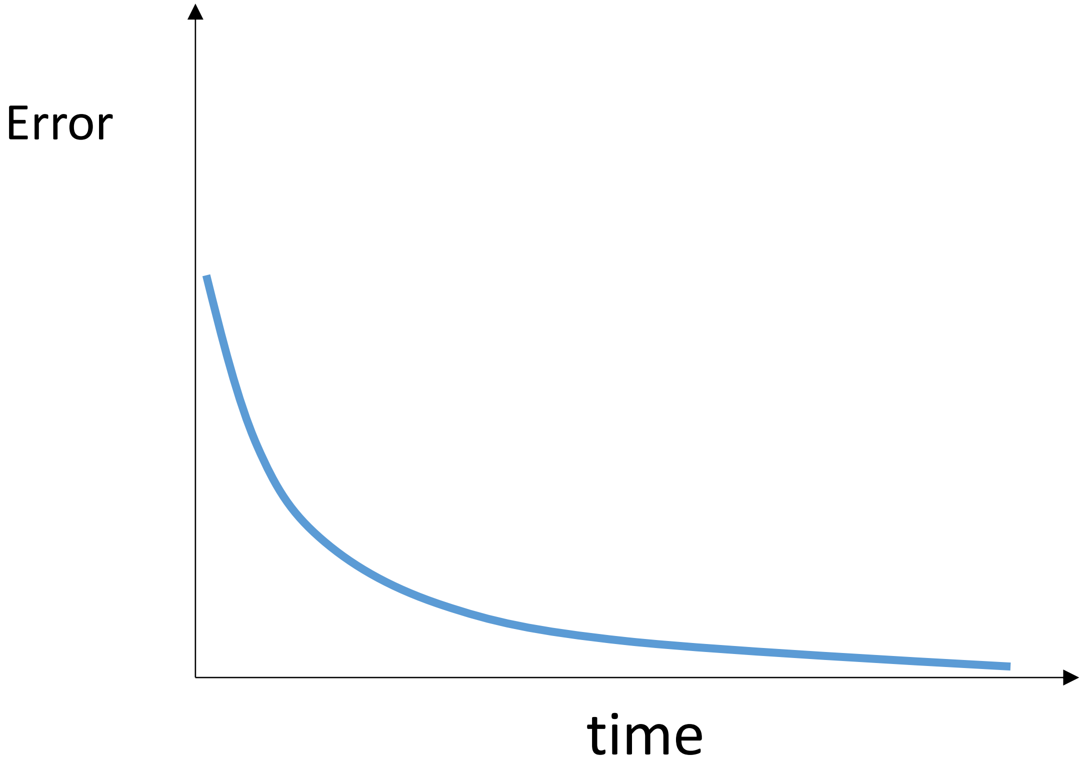
- Example: heating system trying to sense and maintain a certain temperature

Understanding the effects of a controller

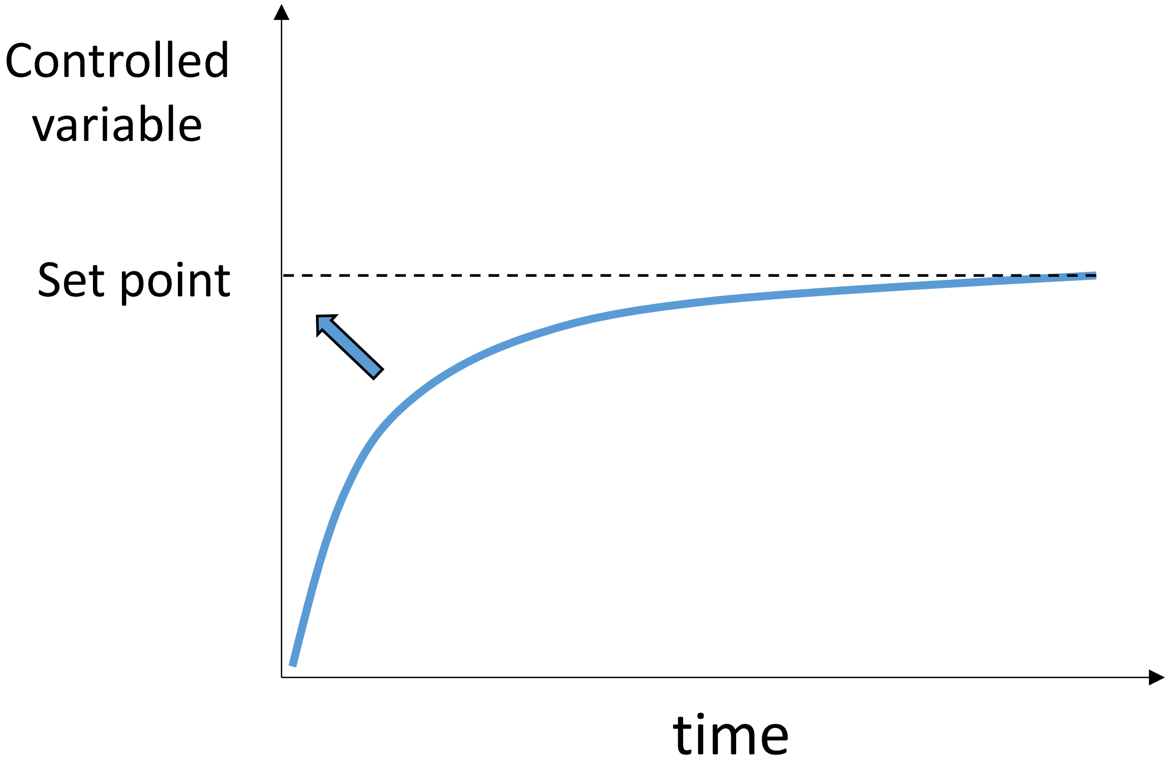


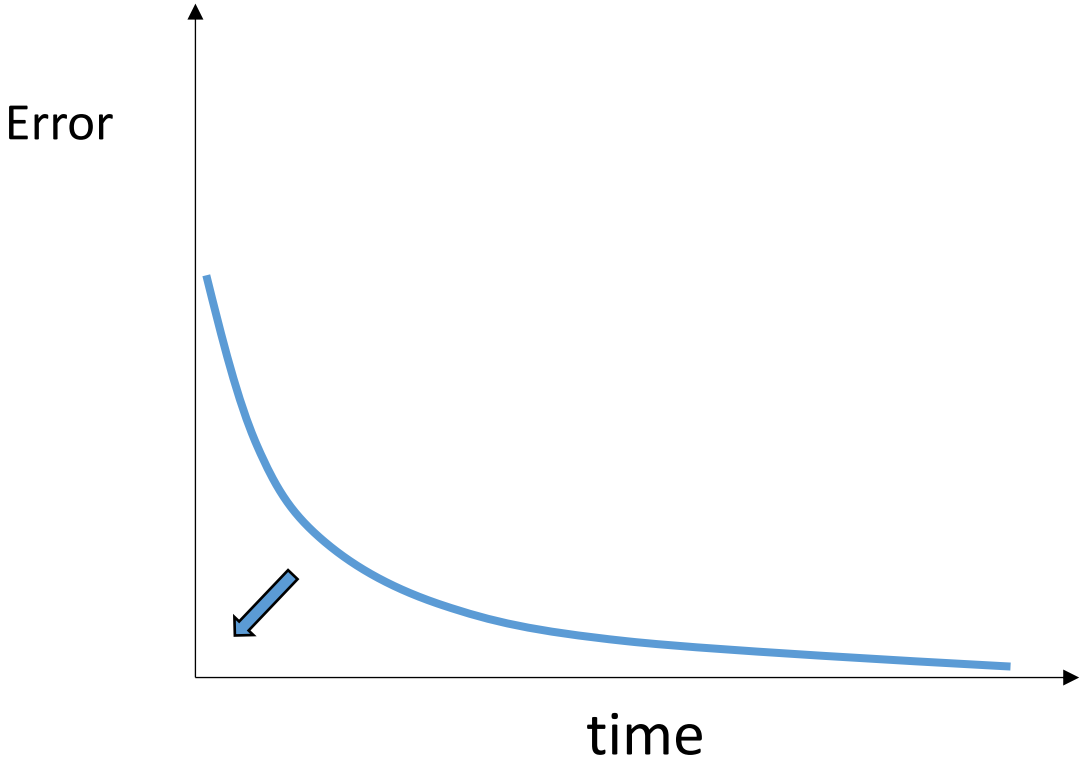
What does ideal behavior look like?



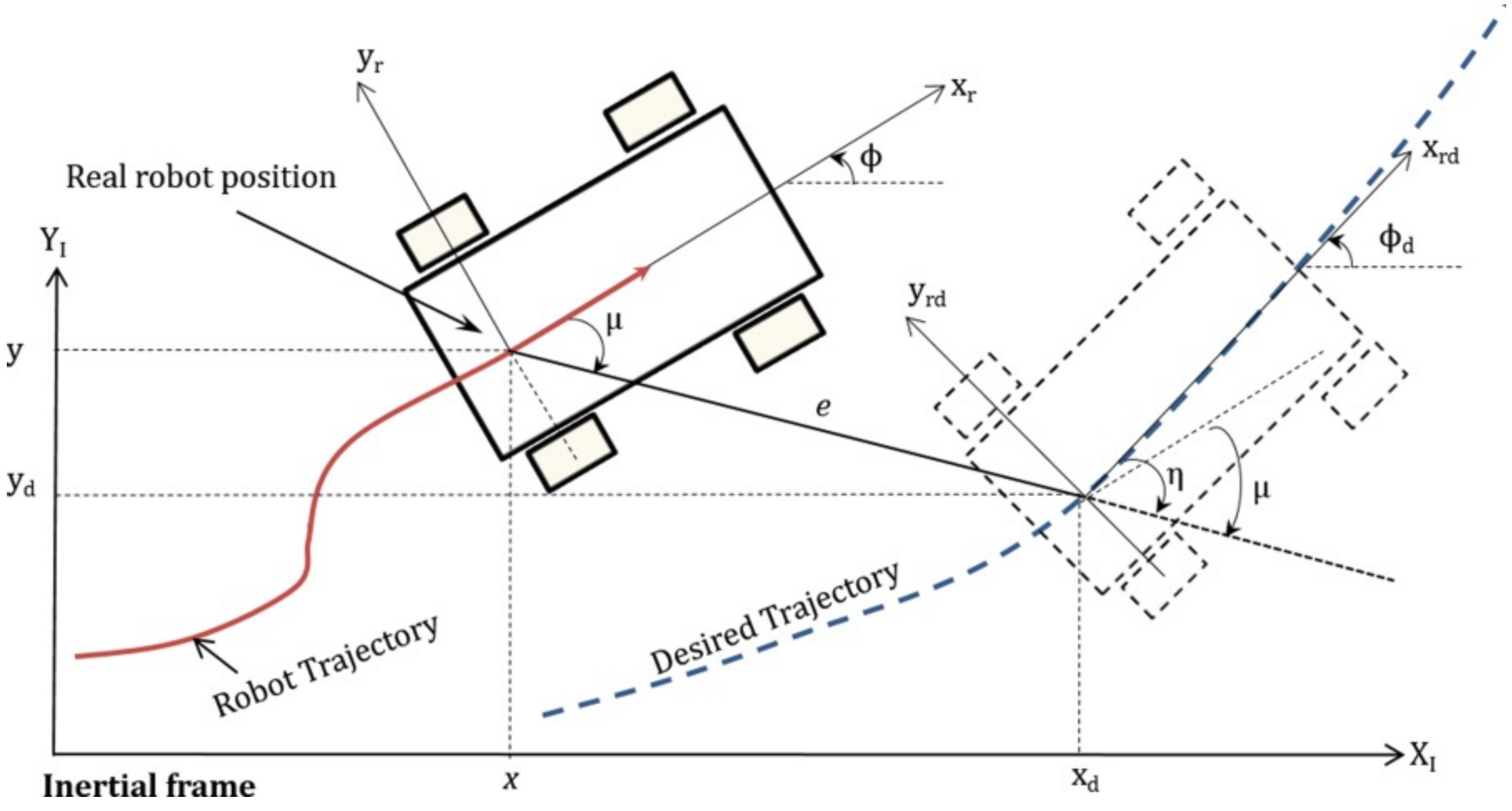


What does ideal behavior look like?

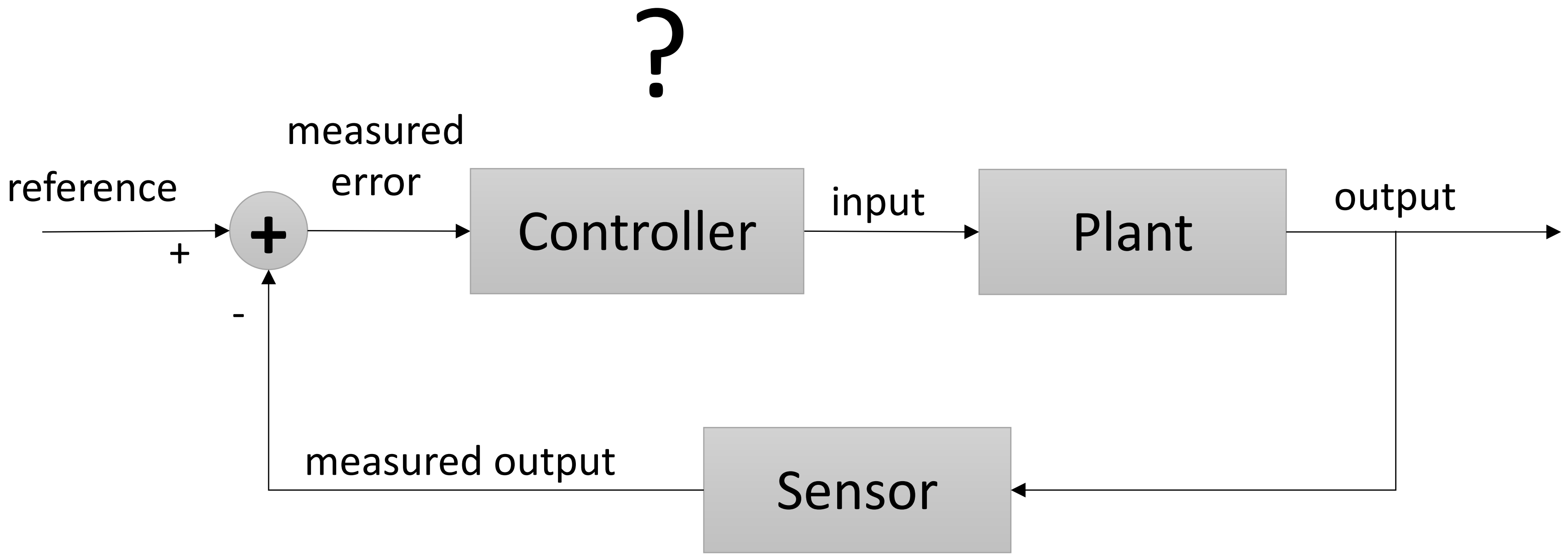




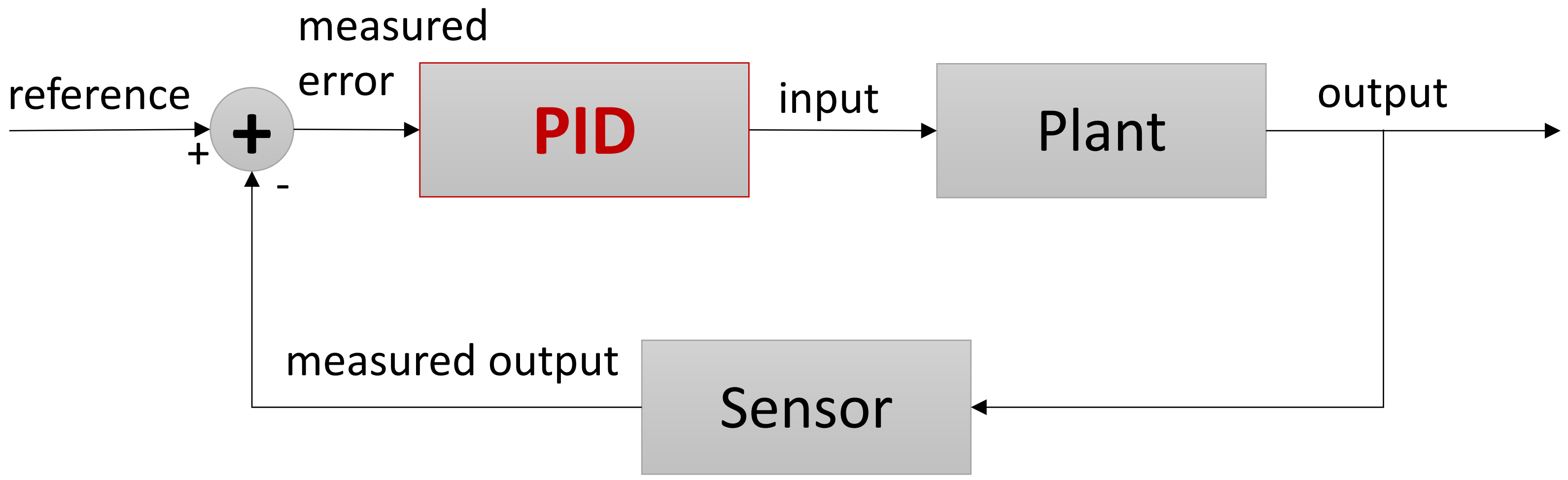
Control in robotics...



What goes into the controller?



A popular control strategy …



PID Control

Proportional

Integral

Derivative

Proportional control

Control input is set proportional to the magnitude of the error

$$y = K_p(e)$$

The proportional control gain K_p determines how quickly the system responds to error

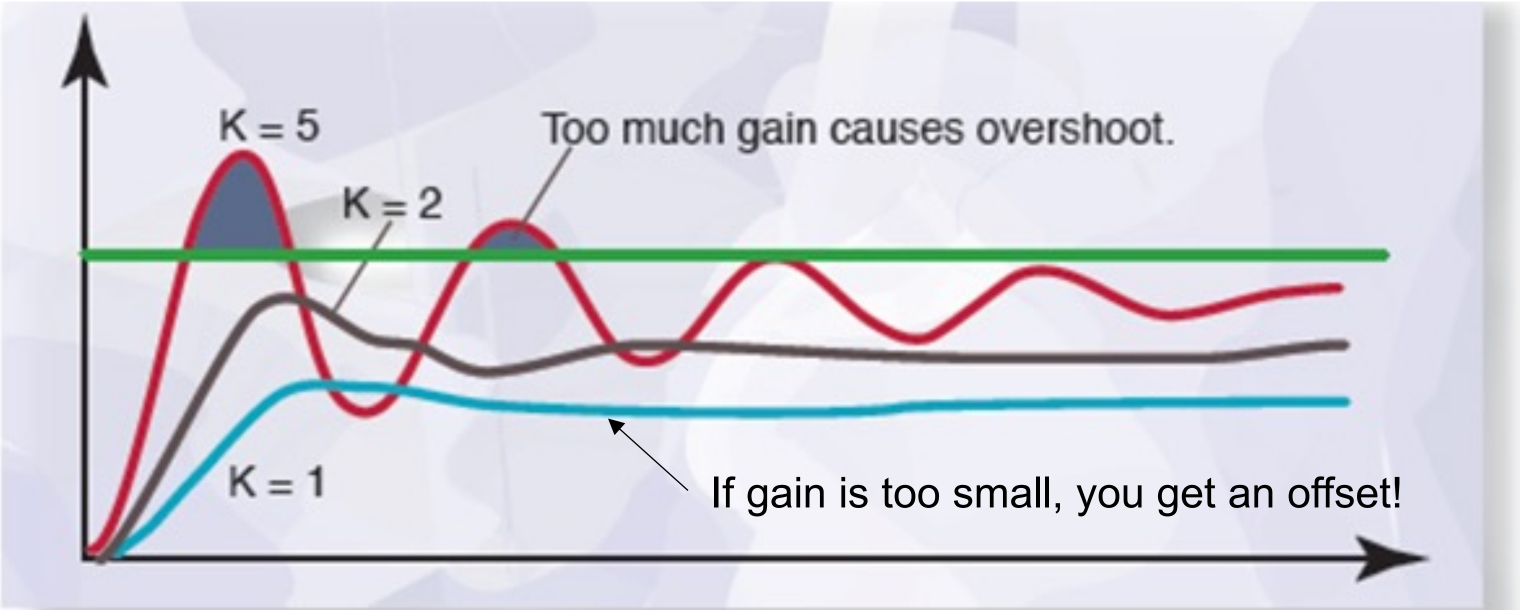
Problem with proportional control

- when error becomes too small, loop output becomes negligible.

- So, even after the proportional loop reaches steady state, error might not be zero => **steady-state error** or **offset**

- The larger the proportional gain, the smaller the steady state error — but, the larger the proportional gain, the more likely the loop is to become unstable.

Proportional control



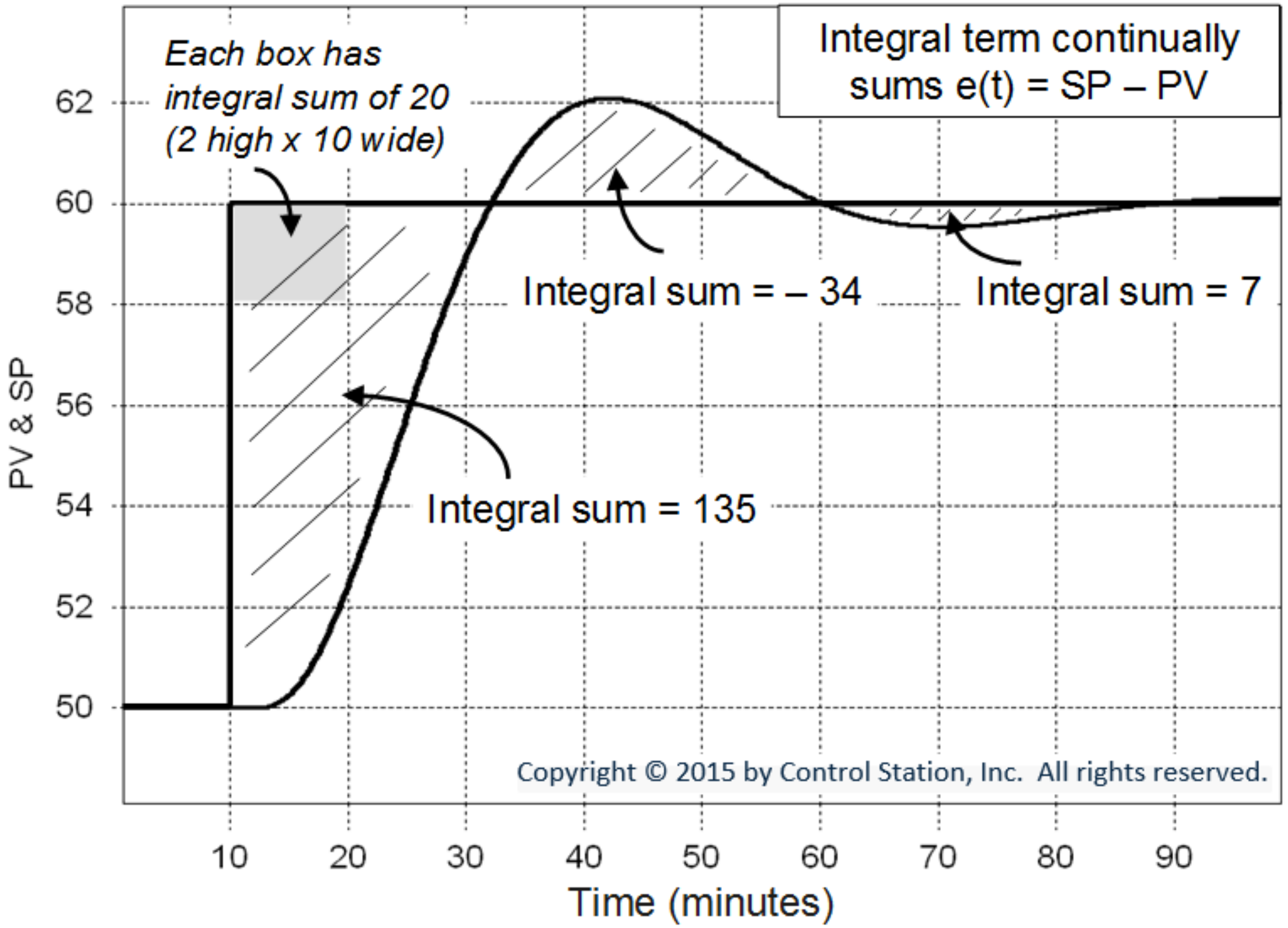
How do we get rid of the offset?

Integral Control

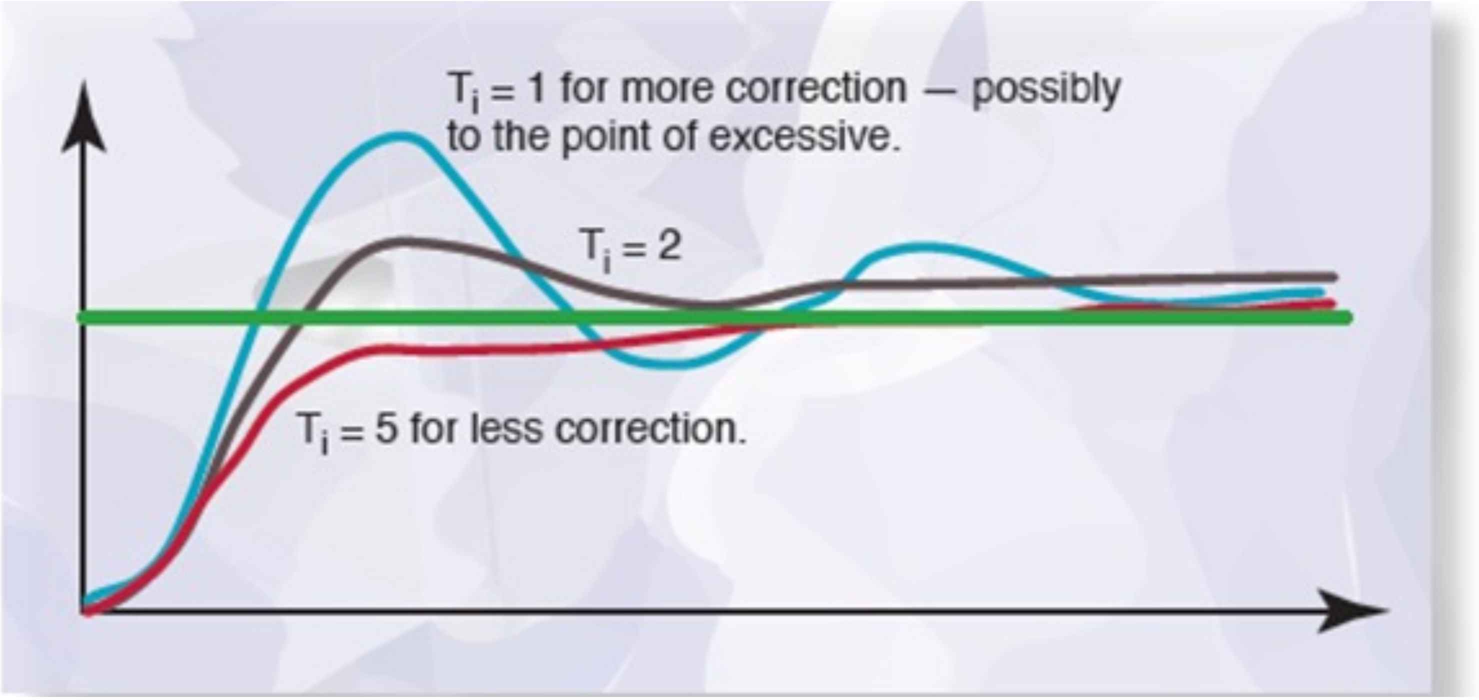
Control input is set proportional to the sum of error over time:

$$y = K_i \int e(t)dt$$

The integral control gain K_i determines how quickly the system responds to error


Each box has
integral sum of 20
(2 high x 10 wide)
Integral term continually
sums e(t) = SP – PV
Integral sum = – 34
Integral sum = 7
Integral sum = 135
PV & SP
62
60
58
56
54
52
50
10
20
30
40
50
60
70
80
90
Time (minutes)
Copyright © 2015 by Control Station, Inc. All rights reserved.

Integral Control



Even small errors (when proportional control is not effective) accumulate over time, thus allowing integral control to eliminate steady-state offset.

How to get rid of overshoot and oscillations?

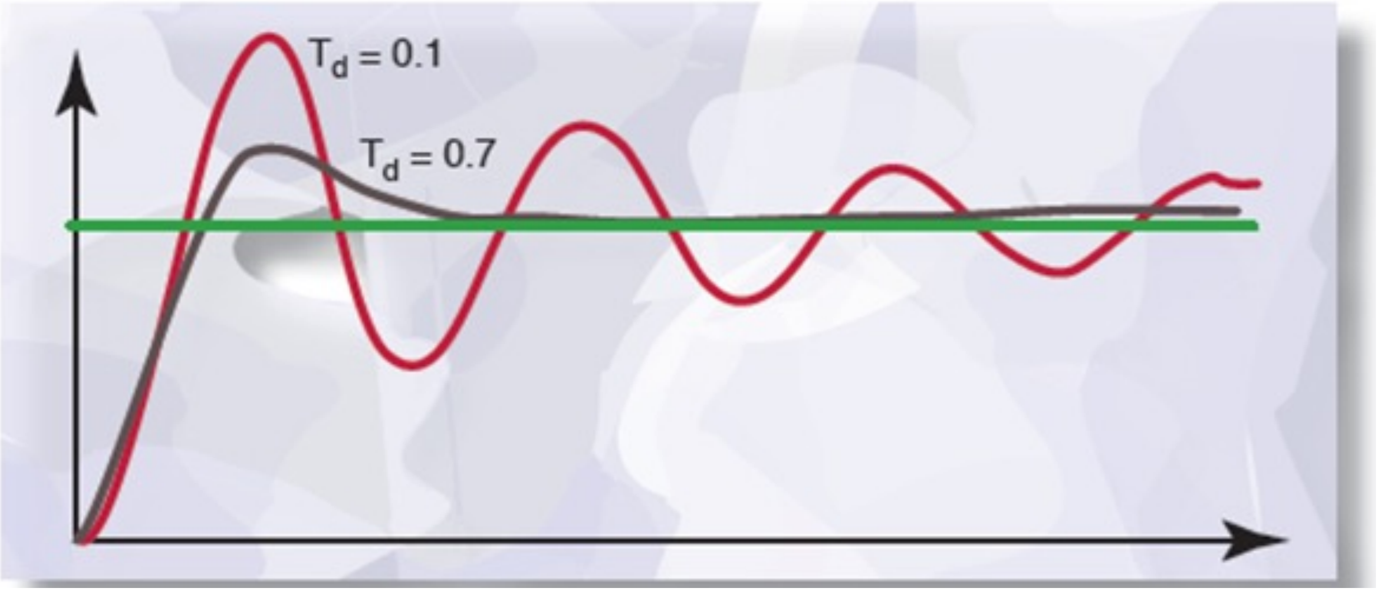
Derivative Control

Control input is set proportional to the rate of change of the error Δε

$$y = K_d \frac{de(t)}{dt}$$

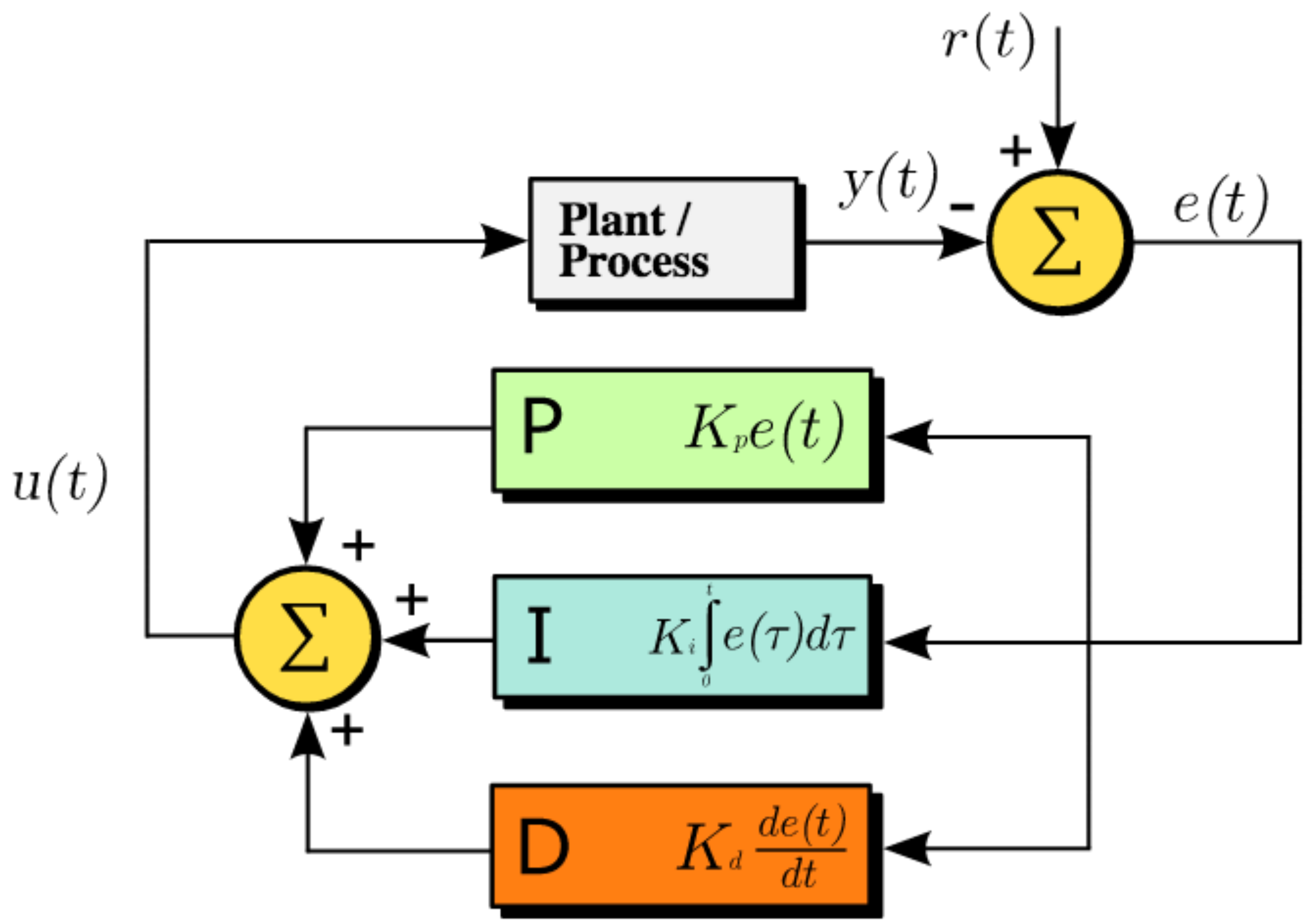
The derivative control gain K_d determines how quickly the system responds to error

Derivative Control



The more error changes, the larger the derivative factor becomes. The effect of the derivative is to counteract the overshoot caused by P and I. When the error is large, the P and the I will push the controller output. This controller response makes error change quickly, which in turn causes the derivative to more aggressively counteract the P and the I.

PID Control

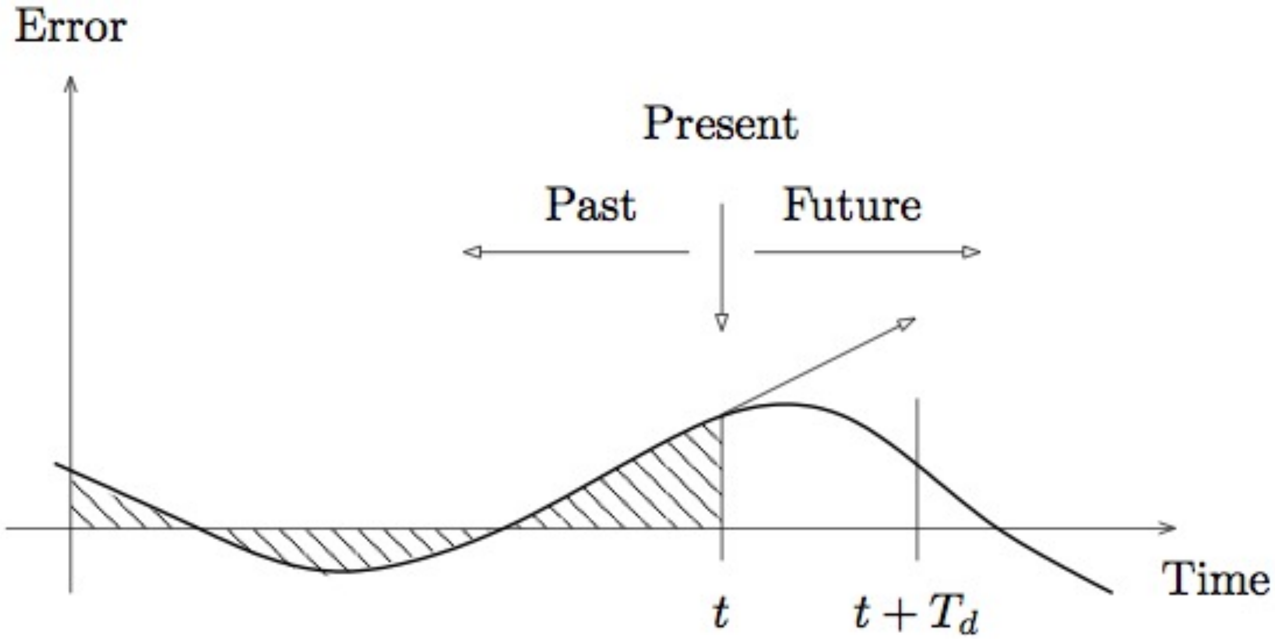


$r(t)$
+
$y(t)$
-
$e(t)$
Plant / Process
$u(t)$
P $K_p e(t)$
I $K_i \int_0^t e(\tau)d\tau$
D $K_d \frac{de(t)}{dt}$
+
+
+

PID Control

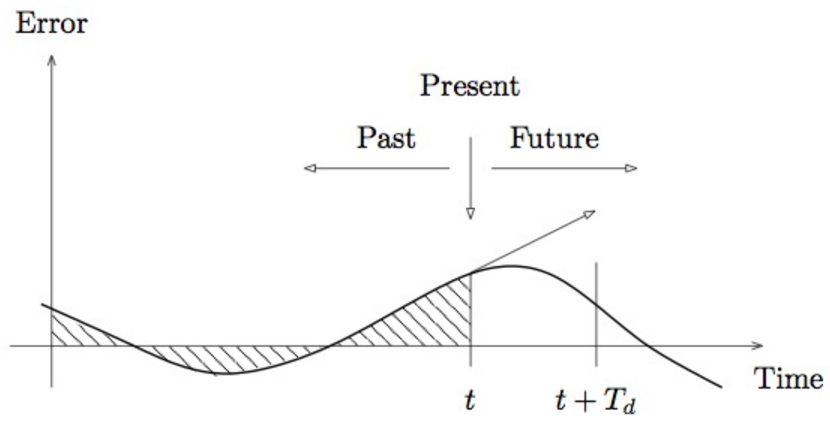
- The most widely used control approach in everything from thermostats to robots
- Any of its components can be used separately or in pairs:
  - P control, PI control, PD control…
- Tuning the controller takes some time since the three gains all interact. Many applications use just a PI controller for simplicity.
- In general, use the simplest controller that gets the job done

PID Control



PID gain values matter… a lot!

PID as three people



Imagine P, I, and D as three people with different personalities

- P likes to react proportionally only to current problems. Does not like to live in the past, nor worry about the future.
- I likes things to be perfect but is lazy. Only reacts after seeing that the problem is persistent.
- D tends to be anxious about the future and tries fix problems before they occur.

An excellent video on PID

https://www.youtube.com/watch?v=4Y7zG48uHRo

Is there a method to this madness?

Tuning PID controllers

Components are highly inter-dependent and can't be tuned independently.

- Each gain aims to change a different characteristic of the trajectory, but at the same time has a negative effect on some other behavior of the system which has to be compensated by re-tuning another gain.

CL Response	Rise Time	Overshoot	Settling TIme	Steady-State Error
KP	Decrease	Increase	Small change	Decrease
KI	Decrease	Increase	Increase	Eliminate
KD	Small change	Decrease	Decrease	No change

Tuning PID controllers

Informal tuning procedure:

1. Tune K_P to achieve the desired rise time
2. Tune K_D to achieve the desired setting time
3. Tune K_I to eliminate the steady state error
4. Repeat and fine-tune

There are (many) more formal methods that can automate this process (we will see an example soon)

You can try it yourself!

http://tommycohn.com/PID-Demo/

Ziegler–Nichols Tuning (Second) Method

1. Start with a low value for K_p
2. Increase K_p until a steady-state oscillation appears.
3. Set $\boldsymbol{K_c}$ **=** $\boldsymbol{K_p}$ and $\boldsymbol{T_c}$ **= period of the oscillation**
4. Use this gain chart:

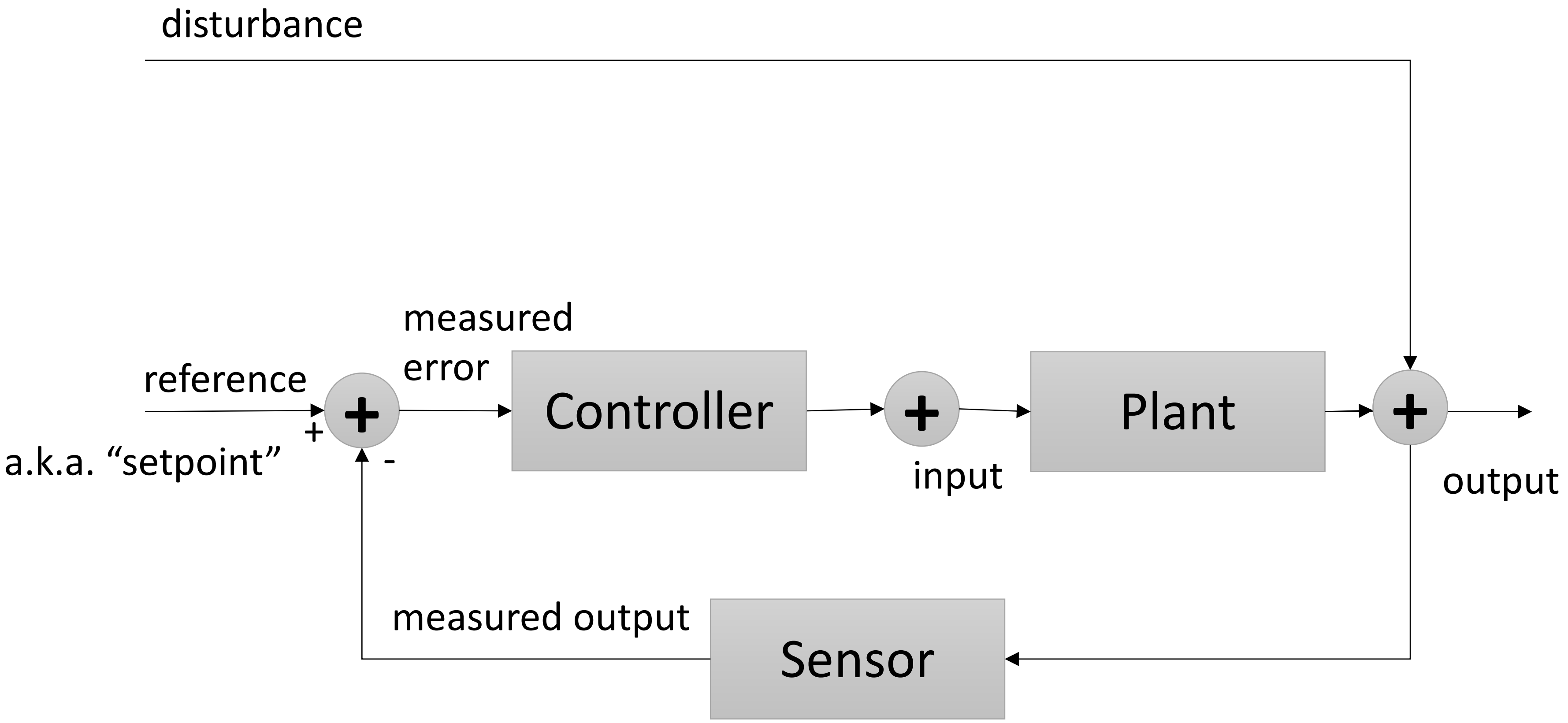
controller	K_p	$\boldsymbol{T_i}$	$\boldsymbol{T_d}$
P	$0.5K_c$	∞	0
PI	$0.45K_c$	$\frac{T_c}{1.2}$	0
PD	$0.8K_c$	∞	$\frac{T_c}{8}$
PID	$0.6K_c$	$\frac{T_c}{2}$	$\frac{T_c}{8}$

$$K_i = \frac{K_p}{T_i}$$
$$K_d = K_p T_d$$

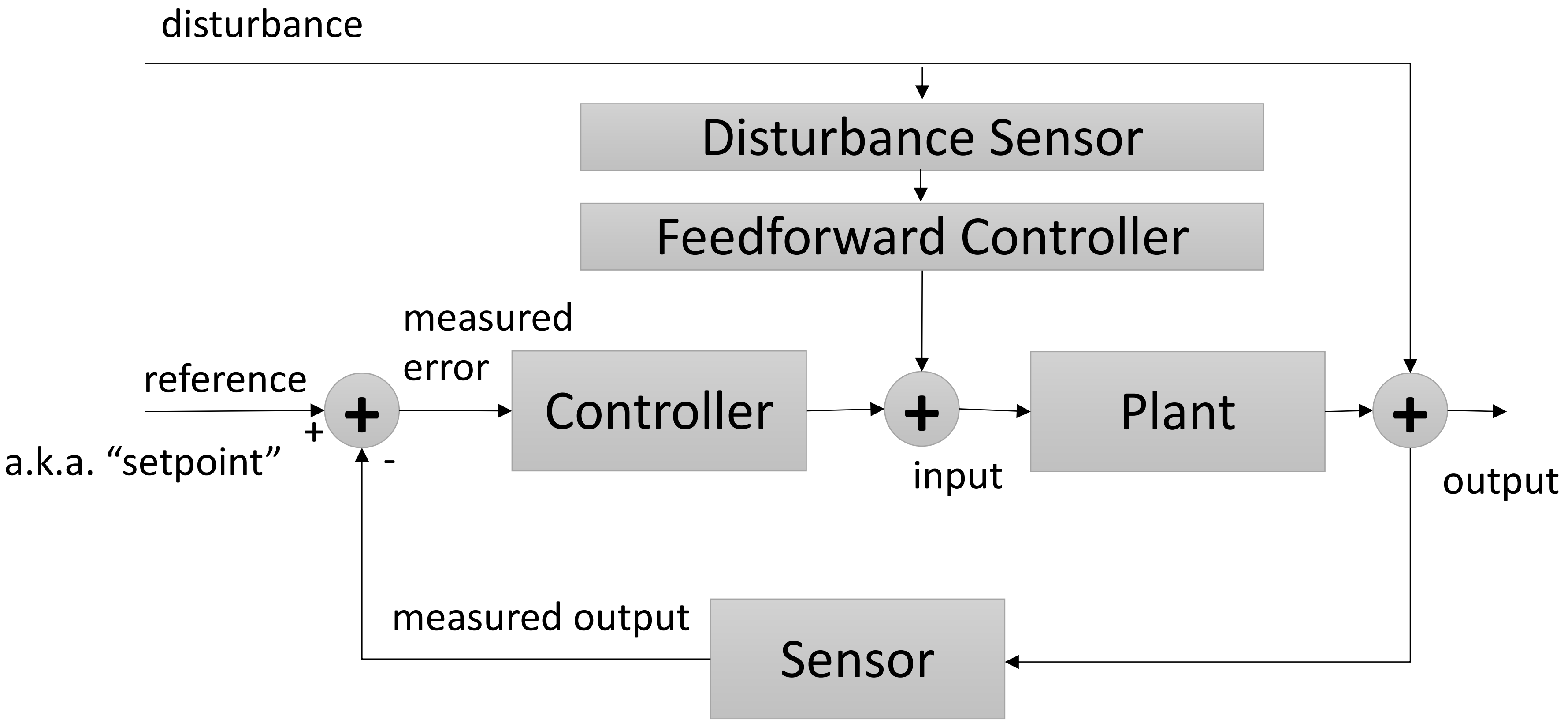
5. Fine tune

What about disturbances?

Feedforward Control System



Feedforward Control System



Open Loop Control: the control action from the controller is independent of the plant output (no sensing at all)

- Example: heating system running on a fixed timer

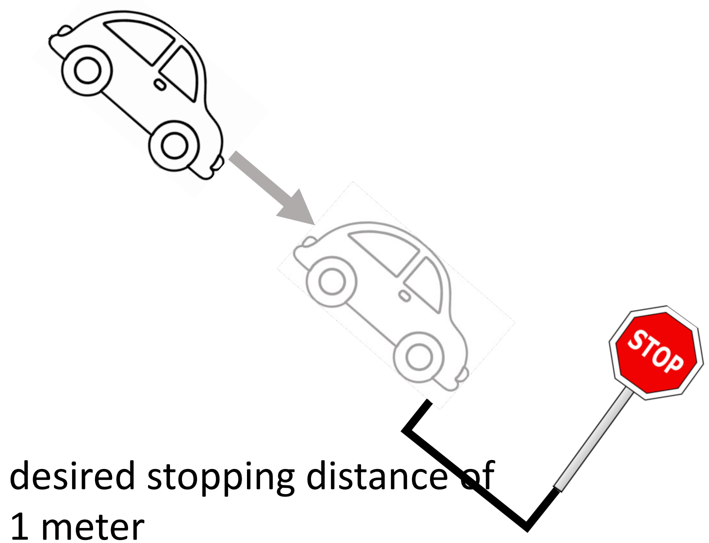
Closed Loop Control (Feedback Control): the control action from the controller is dependent on the plant output (sensing required)

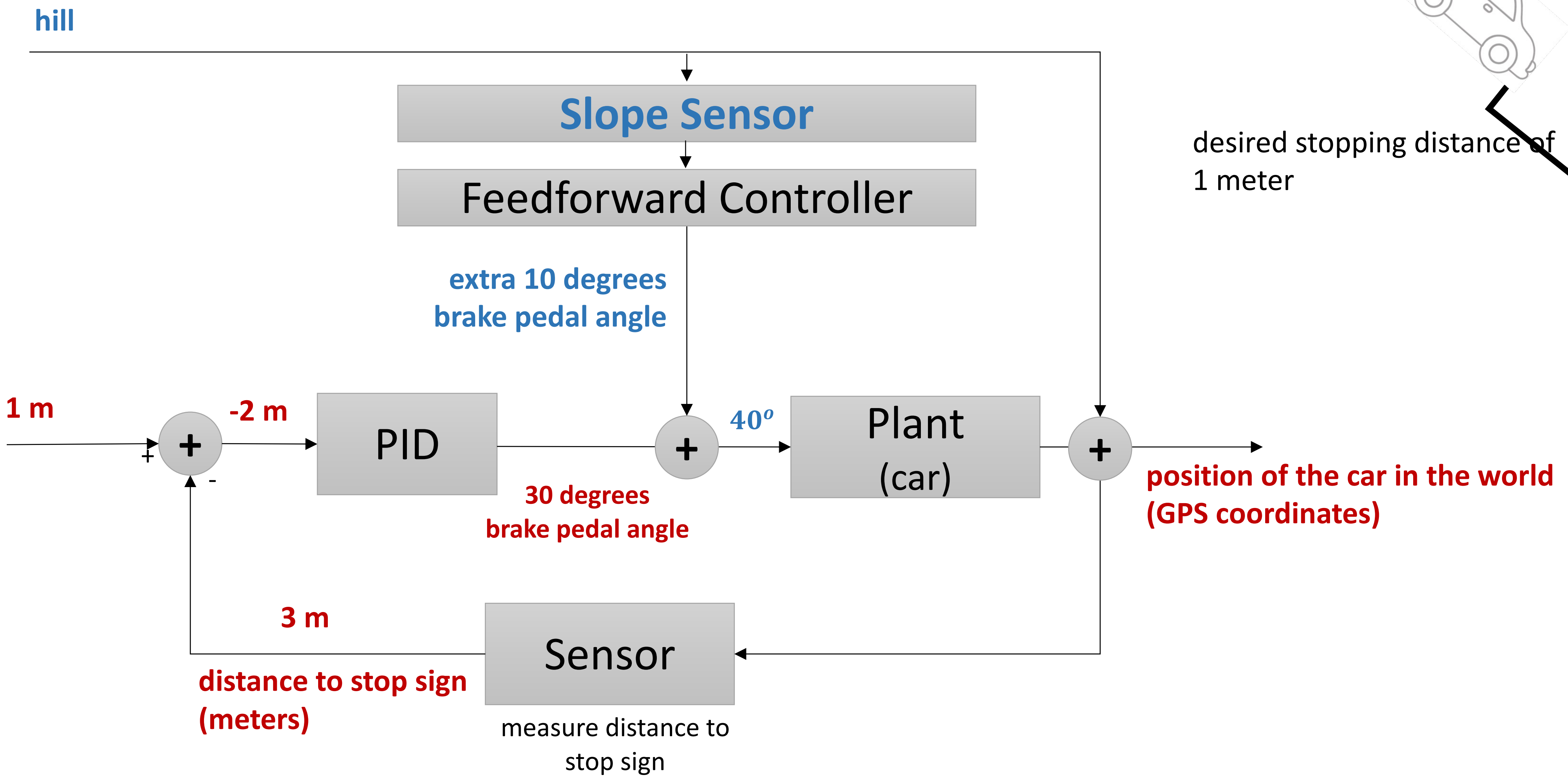
- Example: heating system trying to sense and maintain a certain temperature

Feedforward Control: the system measures an external disturbance and uses a model to predict the effect of the disturbance on the system and react to it (sensing of possible disturbances required)

- Example: sense that the door is open in the winter and turn on the heating in anticipation that the house is about to get cooler

Example: *Automated Stop Sign Braking System on a hill!*





Types of control in robotics… outside the scope of this class

- Motion control
  - Position control
  - Velocity control
- Force Control
- Hybrid Position-Force Control (e.g., writing on a board)
- Impedance control (compliance when guided)