Carnegie Mellon University Qatar

Introduction to Embedded Control

October 9, 2022 & October 11, 2022

By Eduardo Feo-Flushing

1

Agenda

- What is a control system, and why is it important
- Open-loop control and its advantages and limitations
- Closed-loop control: basic concepts and definitions
  - Bang-Bang Controller
  - Proportional Controller
  - PID Controller
- How to tune a PID controller

2

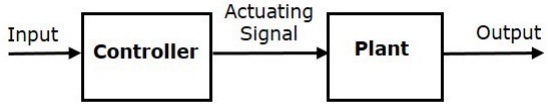
What's a control system?

3

3

Control System

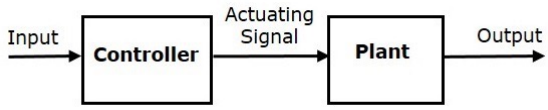
- A collection of
  - Mechanical Devices
  - Electrical Devices
- That can be commanded and directed to control a physical system or device
  - Regulate the behavior of another system
  - Obtain a desired behavior/output from another system



4

4

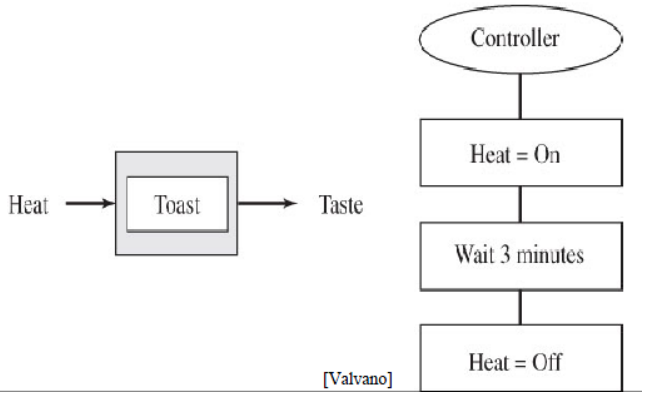
Open-Loop Control



5

5

Open-Loop Control



6

6

Open-Loop Control

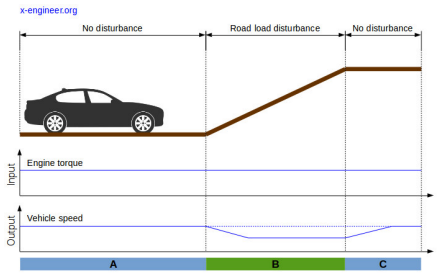
- Other real-life examples?

7

Open-Loop Control

8

Open-Loop Control



9

9

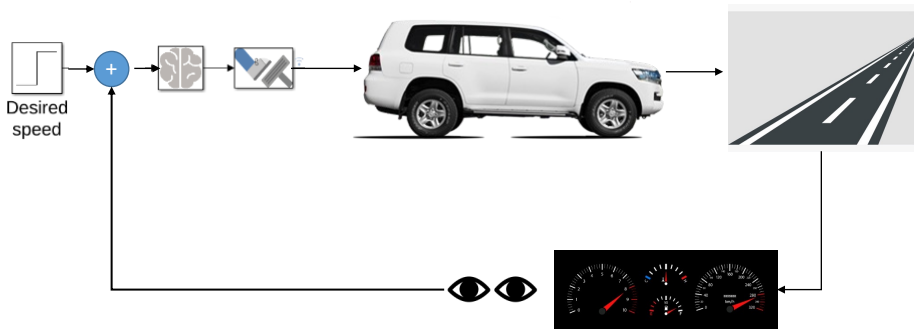
Open-Loop Control

- Does not have state estimators – no sensors
- Does not have any feedback mechanism (open loop)
  - Issues with output noise, disturbances
- Uses heuristics to achieve the desired state
  - Error prone

10

10

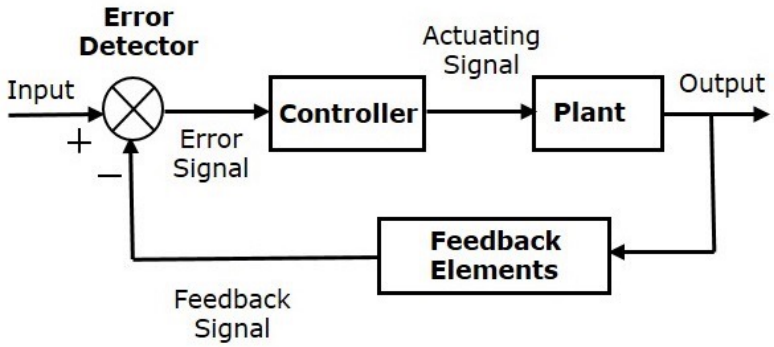
Closed-Loop Control



11

11

Closed-Loop



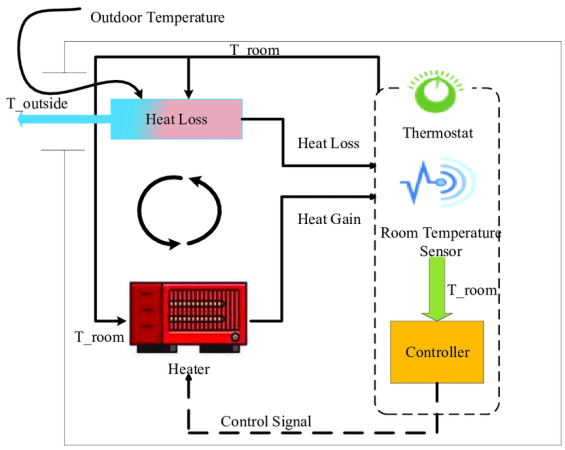
12

Closed-Loop: Feedback

- Achieve the desired state
- Counteract disturbances
- Model uncertainty/noise
- Stabilize unstable systems

13

Home heater with thermostat



14

Basic Idea

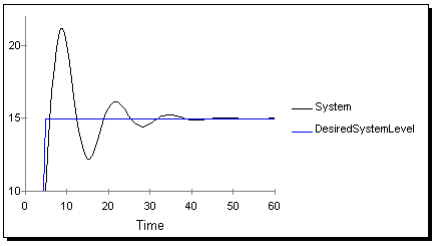
The basic idea of closed-loop control:

- 1. Read sensors
- 2. Figure out the difference between what the sensors say and what you want
- 3. Command actuators to change the system How?
- 4. Go back to step 1 (i.e., this is a control *loop*)

15

Effectiveness of a Control System

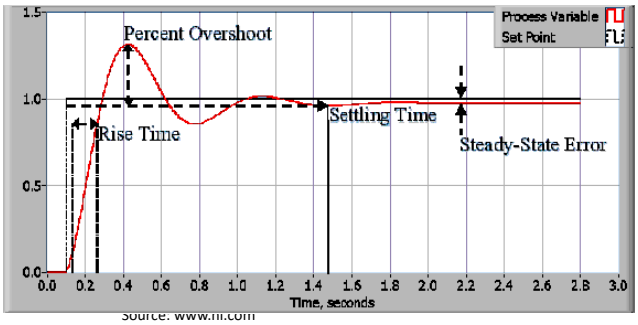
- Get to the desired state as quickly as possible and stay in that state.
- We want to system to:
  - Reach Steady-State – minimize the average error e(t)
  - Minimize transient response – reach steady state fast
  - Be stable in steady state – minimize oscillations around the steady state



16

Definitions

- Rise Time: Time taken to go from 10% to 90% of the steady-state
- Percent Overshoot: Amount by which system state overshoots the target
- Settling Time: Time required to reach steady state (±ϵ of target)
- Steady State Error: difference between system state and target

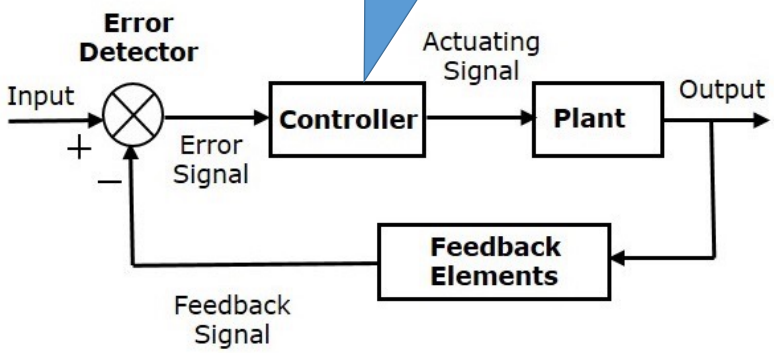


Source: www.ni.com

17

Closed-Loop

How to convert the error signal into a command?



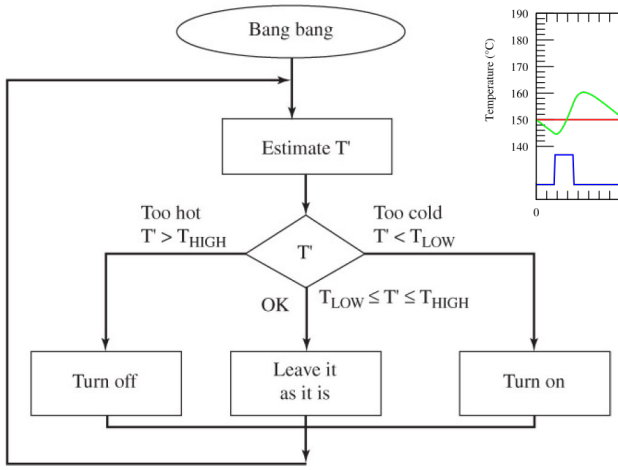
18

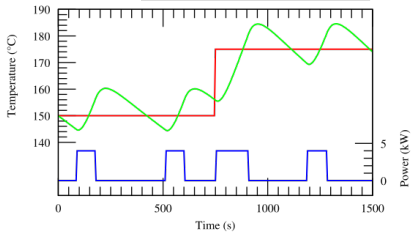
Bang-bang

- Simplest closed-loop control
- Control strategy:
  - If sensed value < desired value -> do action a
  - If sensed value > desired value -> do action b
- Equal? Use a dead band

19

Bang-Bang example: Heater





20

Bang-Bang Control

- Advantages:
  - Simple
  - Effective when desired state is constant
- Disadvantages:
  - Always uses maximum control energy
  - Not very precise: oscillations

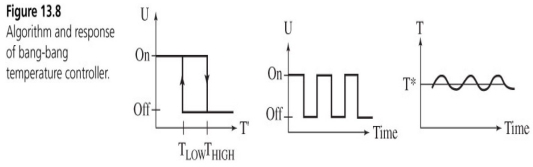


Figure 13.8
Algorithm and response of bang-bang temperature controller.

21

Proportional Control

- Make the control signal proportional to the magnitude of the error in the current state
- Examples:

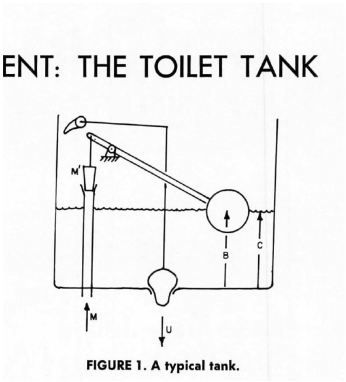
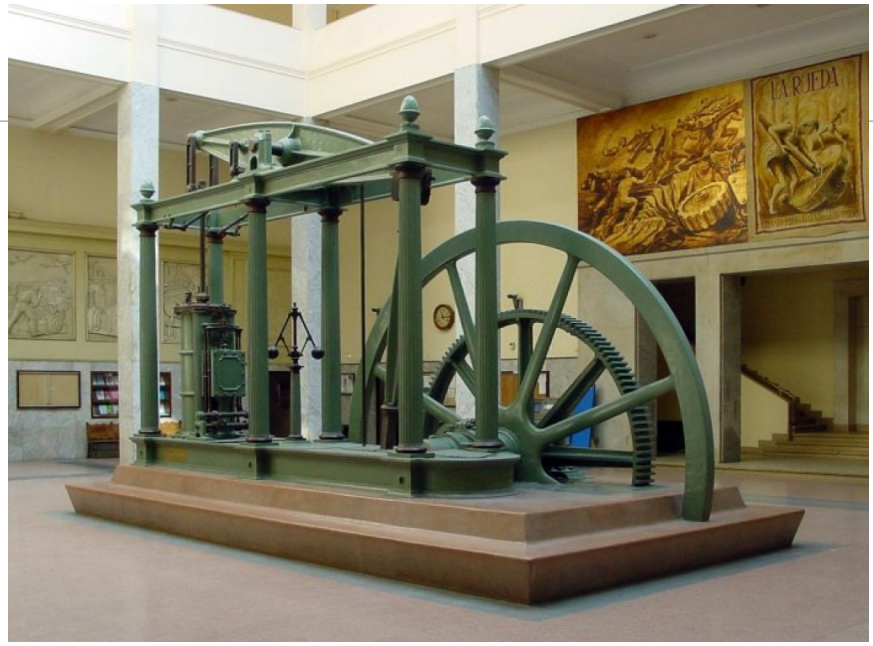


FIGURE 1. A typical tank.

22

22

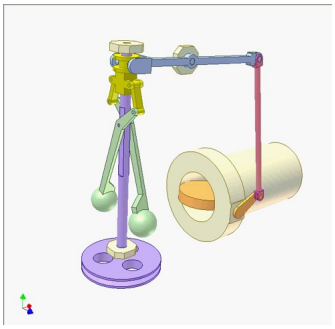


23

23

Proportional Control

- The Centrifugal Governor maintains the mean speed of the engine by regulating the flow of working fluid when there are variations in the load.
- Introduced by Watts and Boulton in 1788
- Uses the principle of Proportional Control



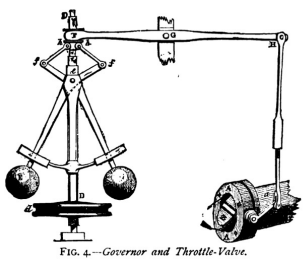


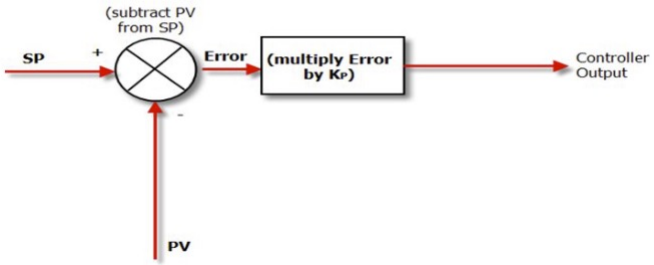
FIG. 4.—*Governor and Throttle-Valve.*

24

24

Proportional Control

- The controller multiplies the error by the "proportional gain" Kp to get the controller output
- Kp must be "tuned". Usually, it is fixed.

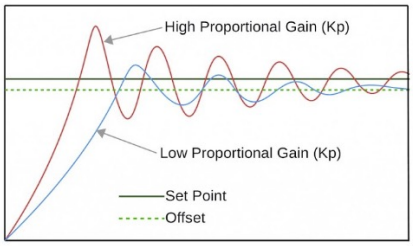


25

25

Proportional Control

- Large values for Kp: approaches SP faster
- Can lead to overshoot and even instability
- Steady-state offset
- Usually not good when:
  - Too much delay in plant response
  - Quick response is needed



$$P = K_p e(t)$$

26

26

Can we do better?

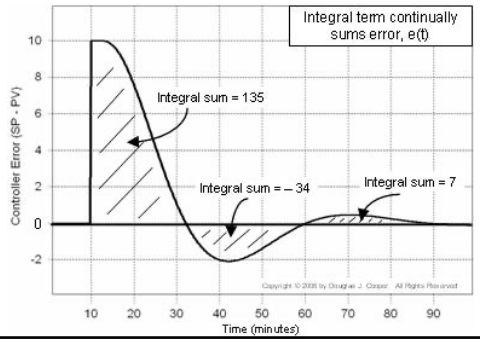
- Combine Proportional Control with other types of control strategies

27

27

Integral Control

- A factor in the output proportional to the **duration** and **magnitude** of the Error Signal.
- The more prolonged the error and the greater the amount, the larger the integral output

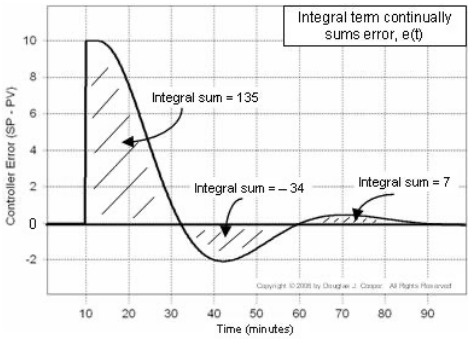


28

28

Integral Control

- The controller multiplies the sum of the errors by the "integral gain" Ki and **adds it** to the controller output



$$I = K_i \int_0^t e(\tau)d\tau$$

29

Proportional-Integral (PI) Control

$$K_p e(t) + K_i \int_0^t e(\tau)d\tau$$

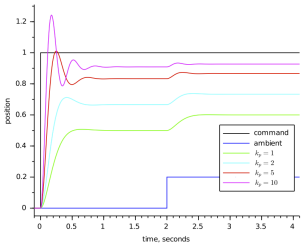


Figure 11: Heater with proportional control.

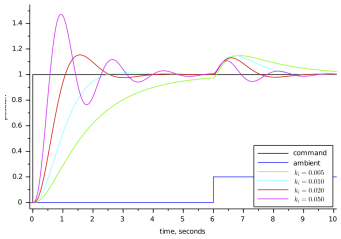


Figure 13: Heater with integral-only control.

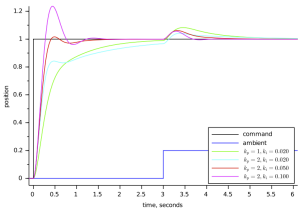


Figure 15: Heater with proportional-integral control.

30

Derivative Control

- Increase response time
  - Greater control when the error is increasing
  - Gentler control when the error is decreasing
- Proportional to the rate of change of the error
- Accounts for "future behavior" or "trend"
- Attempts to reduce overshoot

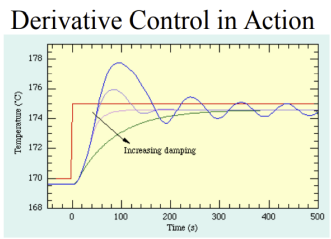
$$D = K_d \frac{d}{dt} e(t)$$

31

31

Derivative Control

- Rarely used alone
- Its damping effect allows the system to have a higher value of Kp and/or Ki.
- Very powerful but "problematic"
- Sensible to sampling irregularities, noise, high-frequency oscillations

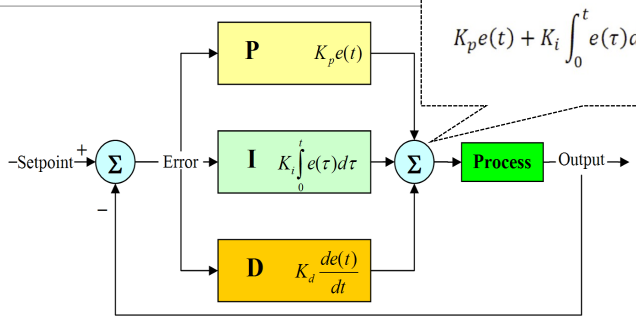


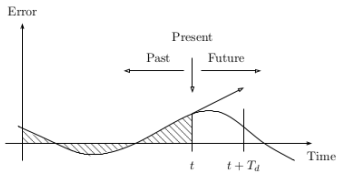
32

32

Putting all together: PID

$$K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t)$$





33

33

Implementing a PID Controller

34

34

PID: Implementation

```
error_prior = 0
integral_prior = 0
KP = //Some value you need to come up (see tuning section below)
KI = //Some value you need to come up (see tuning section below)
KD = //Some value you need to come up (see tuning section below)

desired_value = //  User setting
actual_value = // Updated elsewhere, e.g., IR handler


while(1) {
    error = desired_value - actual_value;
    integral = integral_prior + error * iteration_time;
    derivative = (error - error_prior) / iteration_time;
    output = KP*error + KI*integral + KD*derivative;
    // output should be sent to the plant here, or from another handler (e.g., timer)
    error_prior = error;
    integral_prior = integral;
    sleep(iteration_time);
}
```

35

35

PID: Implementation

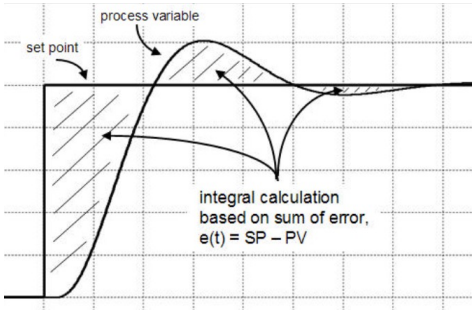
- Sampling rate:
  - between 1/10th and 1/100th of the desired settling time
- Sampling irregularities
- Noise
- Fixed-point arithmetic

36

36

PID: Implementation

- Integral Windup: Error accumulates during rise
  - Conditional Integration
  - Reset when error crosses 0

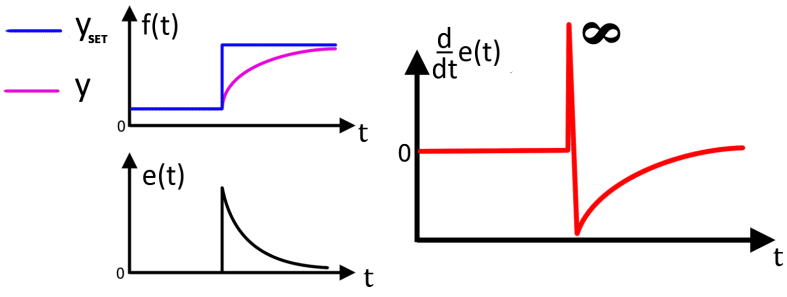


37

37

PID: Implementation

- Derivative Kick: any change in Setpoint causes an instantaneous change in error.



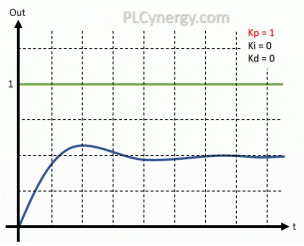
38

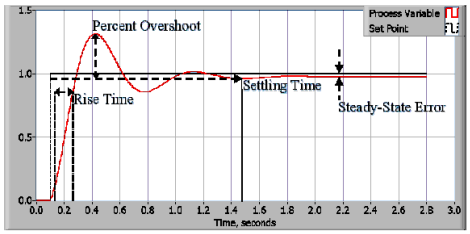
38

Tuning PID

Response	Rise Time	Overshoot	Settling Time	S-S Error
K_P	Decrease	Increase	NT	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	NT	Decrease	Decrease	NT

NT: No definite trend. Minor change.



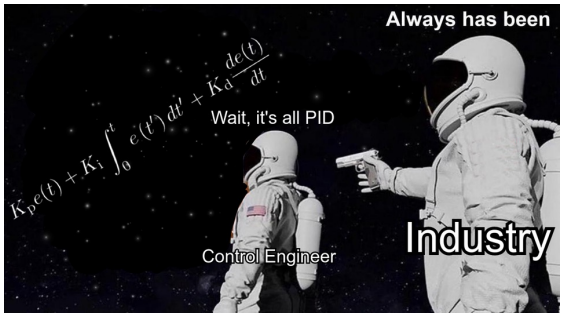


39

39

Tuning PID

- "Ninety percent of the closed-loop controller applications in the world do very well indeed with a PID controller that is only tuned fairly well, so you're in luck" Tim Wescott



40

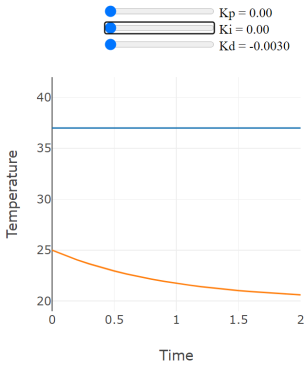
Tuning PID

- Try only Proportional
  - If it works, you are lucky!
- If It oscillates
  - Set a minimum Kp
  - We start with Derivative: Kd=100*Kp
  - Adjust derivative until the system is on the verge of oscillation
  - Adjust Proportional: Kp = (1/100)*Kp
  - Adjust Integral

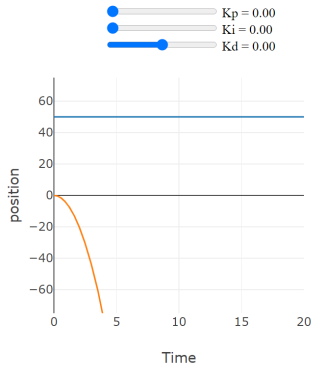
41

Tuning PID: Practice

bit.ly/3EymTWo



bit.ly/3RVbW42



Source: https://thomasfermi.github.io/Algorithms-for-Automated-Driving/Control/PID.html

42

Fuzzy Control

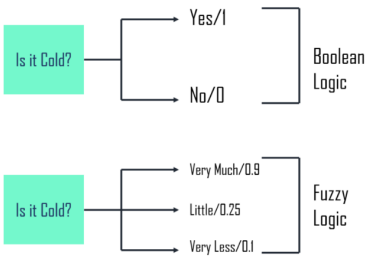
when PID is not enough

43

43

Fuzzy Control

- PID tuning can be complicated
- Modeling and simulating complex real-world systems is difficult
- Fuzzy does **not** mean "it works, but we don't know how!"
- Involves logic that cannot be expressed as **true** or **false** but as somewhat true/false



44

Fuzzy Control

- First presented by Lotfi Zadeh in 1960s
- Rule-based automatic controllers using natural language
- Static nonlinear mapping between its inputs and outputs (like PIDs?)
- Still popular and effective

45

Example: Fan Speed Controller

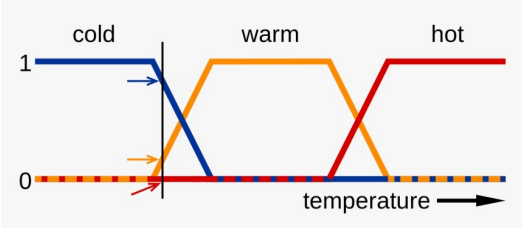
- **Temperature** is the input, and **Fan Speed** is the output.

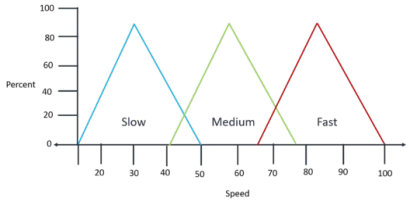
46

46

Example: Fan Speed Controller

- **Temperature** is the input, and **Fan Speed** is the output.





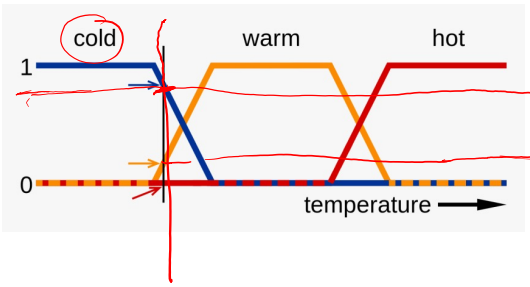
47

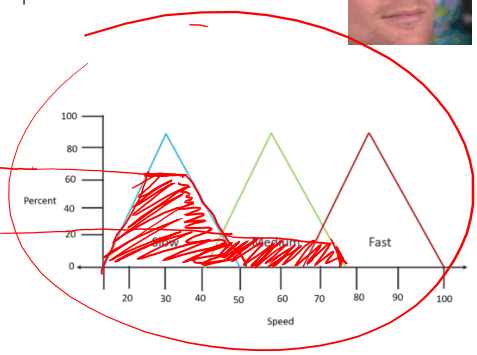
47

Example: Fan Speed Controller

- Rules:
  - If **Hot**, then set Fast Speed
  - If **Warm**, then set Medium Speed
  - If **Cold**, then Slow set Speed

It's not too much cold and a bit warm; what should the fan speed be?





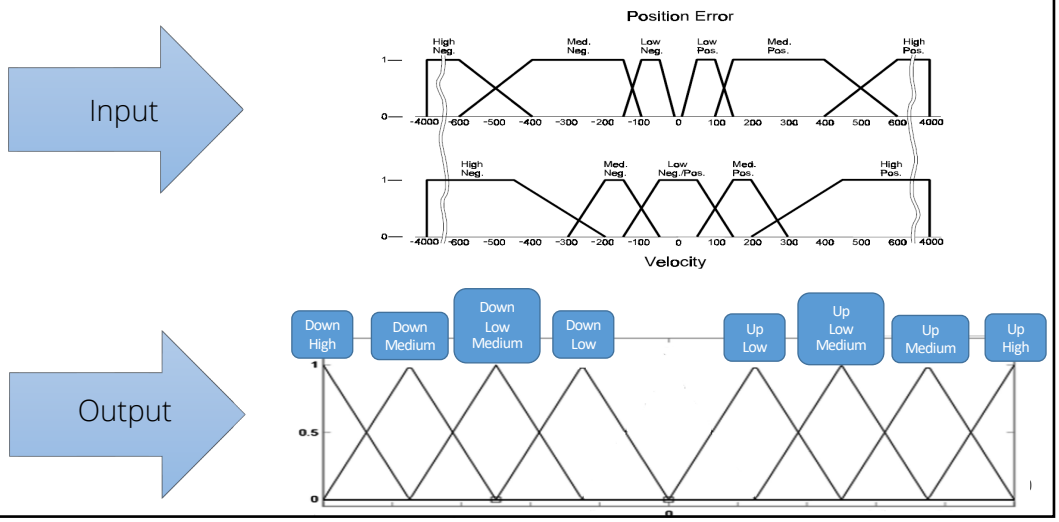
48

48

Knowledge into Rules

- General form:
  - IF premise THEN consequent
- Premise can be composed of several terms
- Tabular representation

Beam ball example



Beam ball example

- 2 inputs:
  - Position error
  - Velocity
- 6 levels:
  - 36 combinations
  - Symmetric
- 9 output levels

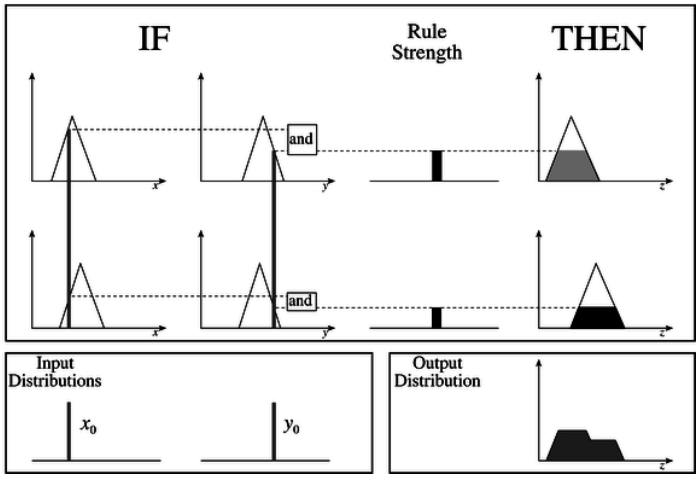
INPUT CONDITION			CONSEQUENCE (required output level)	
Index	Position Error	Velocity	Required Move for the Driven End of the Track	Index of Output
1	High +	High +	Down Low Medium	3
2	High +	Med. +	Down Low Medium	3
3	High +	Low +	Down High	1
4	High +	Low -	Down High	1
5	High +	Med. -	Down High	1
6	High +	High -	Down High	1
7	Med. +	High +	Up High	8
8	Med. +	Med. +	Up Medium	7
9	Med. +	Low +	Down Low Medium	3
10	Med. +	Low -	Down Low Medium	3
11	Med. +	Med. -	Down High	1
12	Med. +	High -	Down High	1
13	Low +	High +	Up High	8
14	Low +	Med. +	Up Low Medium	6
15	Low +	Low +	Down Low	4
16	Low +	Low -	Down Low	4
17	Low +	Med. -	Down Medium	2
18	Low +	High -	Down High	1

IF position error is "Low+" AND velocity is "Med-" THEN beam position is "Down Medium"

51

51

Rules: Multiple inputs

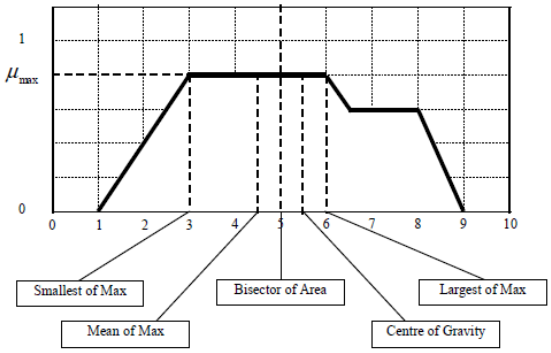


52

52

Defuzzification: other methods

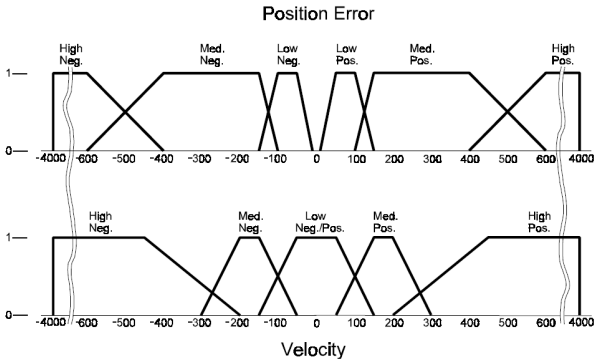
- Mean-of-Max, Center of Gravity, …



53

53

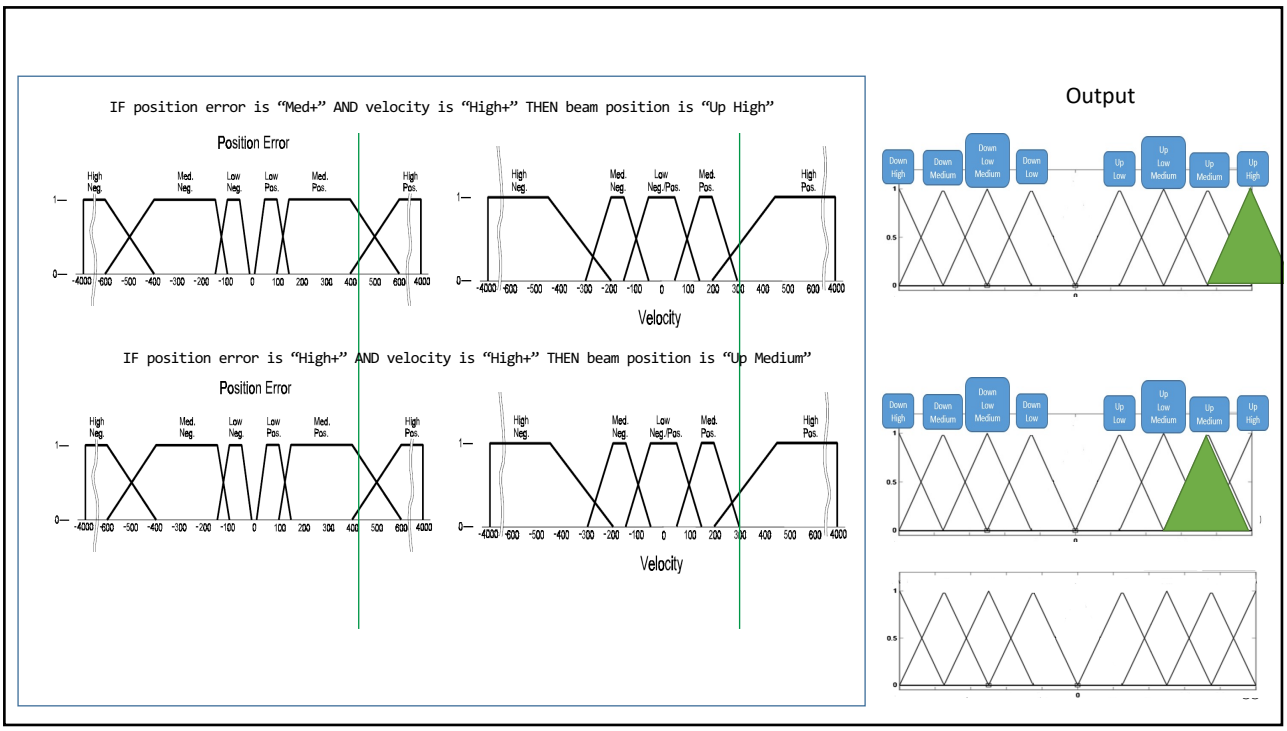
Beam ball example



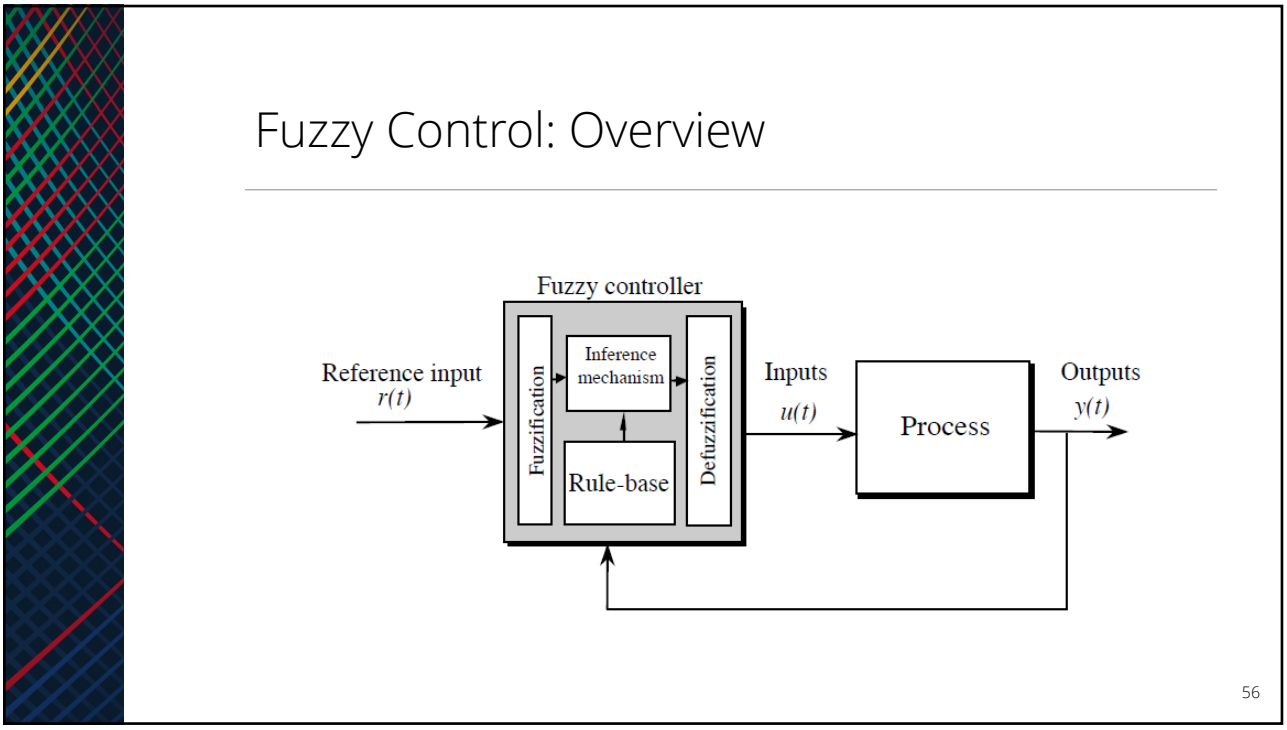
IF position error is “Med+” AND velocity is “High+” THEN beam position is “Up High”
IF position error is “High+” AND velocity is “High+” THEN beam position is “Up Medium”

54

54


IF position error is "Med+" AND velocity is "High+" THEN beam position is "Up High"
Position Error
Velocity
IF position error is "High+" AND velocity is "High+" THEN beam position is "Up Medium"
Position Error
Velocity
Output

55


Fuzzy Control: Overview
Fuzzy controller
Reference input
r(t)
Fuzzification
Inference mechanism
Rule-base
Defuzzification
Inputs
u(t)
Process
Outputs
y(t)
56

56

Fuzzy Control: Summary

1. Preprocessor calculates relevant input called 'crisp inputs'
2. Fuzzification will convert crisp input into fuzzy membership set
3. Fuzzy logic converts input fuzzy set to output fuzzy set
4. Defuzzification converts fuzzy outputs to crisp outputs
5. Actuator system changes the physical plant based on these outputs

57

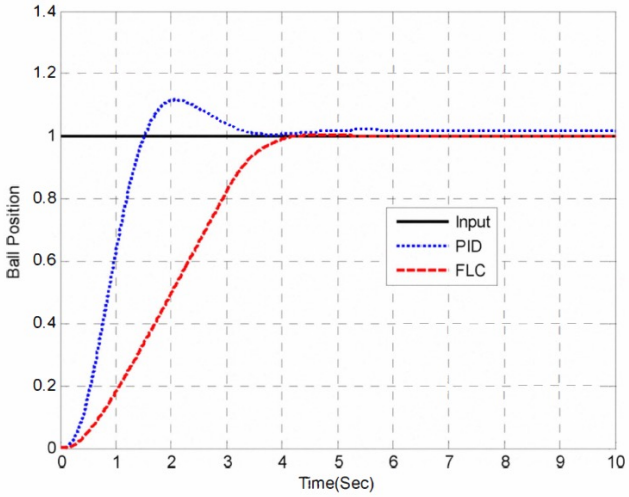
57

Feedback Control: Recommendations

- Try PID first
- Start with a simple fuzzy controller
- Do not add inputs, write complex rules, or use complex membership functions until you need them
- Tuning using scaling gains
  - Input' = k*Input
  - Output'=k*Output

58

58



59

59

The Magic of Feedback Control

Some examples

60

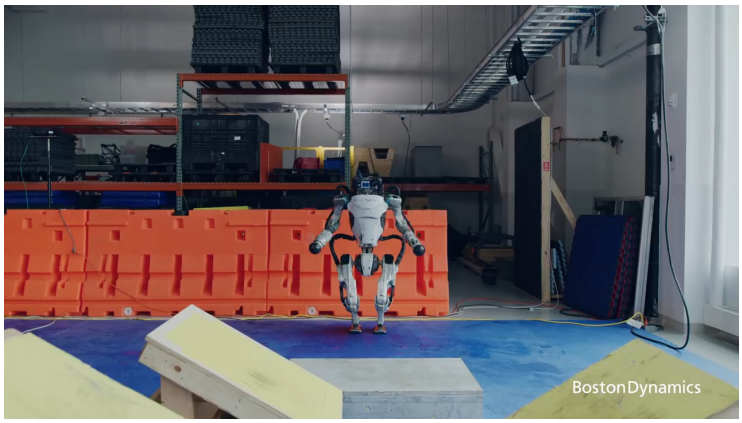
60

Some cool examples

Quadcoper Pole Acrobatics: Brescianini, Hehn, D'Andrea

61

Some cool examples



Atlas does parkour. Boston Dynamics

62

Some cool examples

Ball Balancing PID System. www.ba-bot.com

63