Week 6: PID II

IEEE Micromouse Decal

What our mouse needs to do

- This week: TRAVEL STRAIGHT
- Also this week: WALL FOLLOWING

Problem:

- Motors move at slightly different speeds
- CROOKED PATHS

Solution:

Feedback system!

Brief Feedback Example: Homeostasis

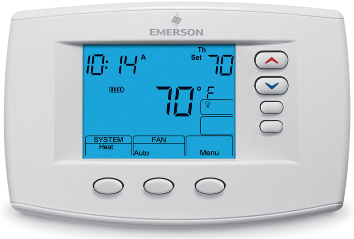
Is this a negative or positive feedback loop?
How can we decide whether to sweat or to shiver using a mathematical model?

Control

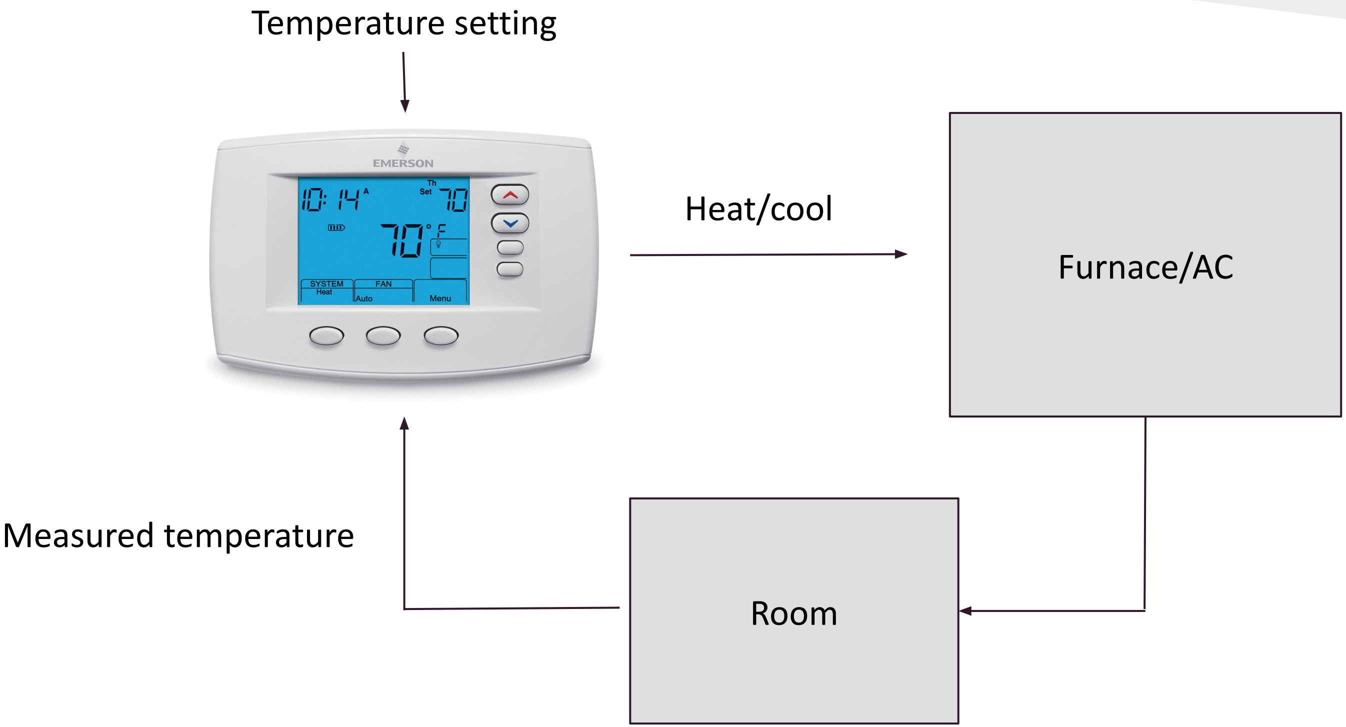
In control theory, a **controller** is a device, historically using mechanical, hydraulic, pneumatic or electronic techniques often in combination, but more recently in the form of a microprocessor or computer, which monitors and physically alters the operating conditions of a given dynamical system.

–Wikipedia

Making an output value match a desired value



Example - Thermostat



Example - Thermostat

```
if temp < setting
    turn on heat
else if temp > setting
    turn on AC
else
    do nothing
```

Temperature is too low

Do something that increases the temperature

Temperature is too high

Do something that decreases the temperature

Example - Thermostat

```
if (setting - temp) > 0
    turn on heat
else if (setting - temp) < 0
    turn on AC
else
    do nothing

error = (setting - temp)
```

Definitions

System - converts inputs to outputs

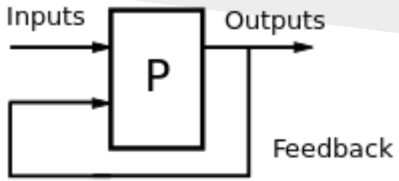
Input variable - variable you change

Output variable - variable you observe

Setpoint - desired value of output variable

Error - (setpoint - output variable)

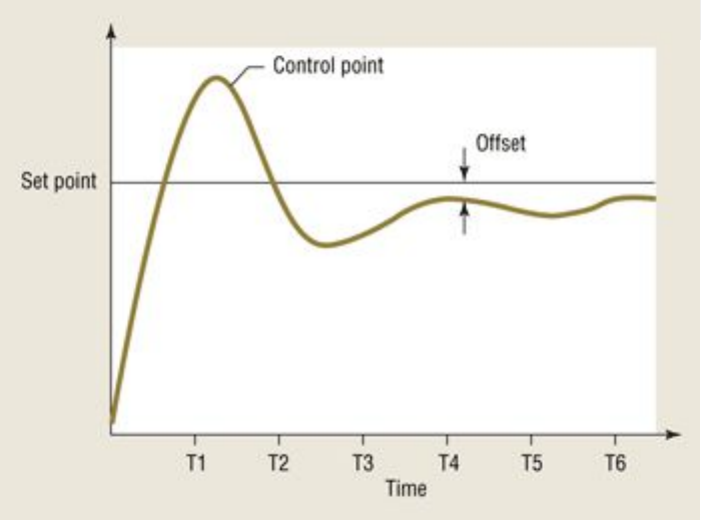
Feedback control - set input based on output



(Wikipedia)

Control goals

- Reach the setpoint quickly
- Stay at the setpoint
- Don’t oscillate



Control in Micromouse

- System is the robot
- Lots of variables to control

Input variable	**Output variable**	**Setpoint**
Steering	Angle turned (left velocity - right velocity)	0
Steering	Distance from walls	Centered between walls
Motor PWM	Distance traveled	*n* maze cells

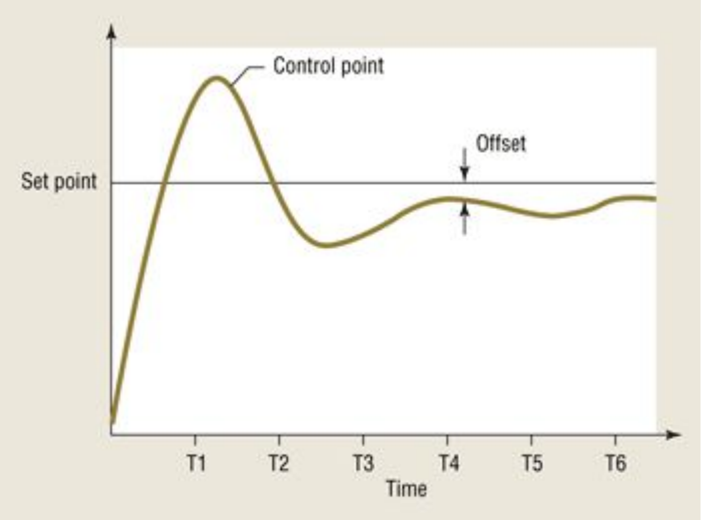
Proportional control

- **<u>Desired</u>**: Left velocity = Right velocity
  - Left velocity = constant + correction
  - Right velocity = constant + correction
- Correction term:
  - $u_p(t) = K_p(l\_vel - r\_vel) = K_p e(t)$
- Output is only <u>proportional</u> to <u>current</u> error
- Good start
- Sometimes gets “stuck”

Proportional control problems

Thermostat example

- Temperature doesn't respond immediately
- Overshoot



PI control

- Errors can accumulate over time! Get rid of the error.
- Sum the error over time, and correct for it.
- Ex: Sleep Debt accumulates.
    - Let **<u>Target = 9 hours</u>**

DAY	SLEEP AMT	ERROR
Monday	1 hr	--8 hr
Tuesday	9 hr	0 hr
Wednesday	8 hr	--1 hr
Thursday	10 hr	+1 hr

I'm getting good sleep!!

No, you need to make up for those 8 hours.

PI control

Desired: left_velocity - right_velocity = 0 (Going straight)

Time	Left_velocity - right_velocity (m/s)	ERROR
0.1s	0.02	-0.02
0.2s	0.03	-0.03
0.3s	0.02	-0.02
0.4s	0.03	-0.03
SUM OF ERRORS		-0.1 m/s

CORRECTION: increase right_velocity so that momentarily,
left_velocity - right_velocity < 0.
This makes Sum of errors closer to 0.

PI control

- $u(t) = K_p e(t) + K_i \int e(\tau)\, d\tau$
- Output also takes accumulated error into account
- Eliminates steady-state error
- Watch out for runaway integral term
    - Integral clamping

PID control

- $u(t) = K_p e(t) + K_i \int e(\tau)\, d\tau + K_d(e(t)-e(t-1))$
- Output takes both accumulated error and rate of change of error into account
- Derivative term
  - Adds damping
  - Prevents overshoot

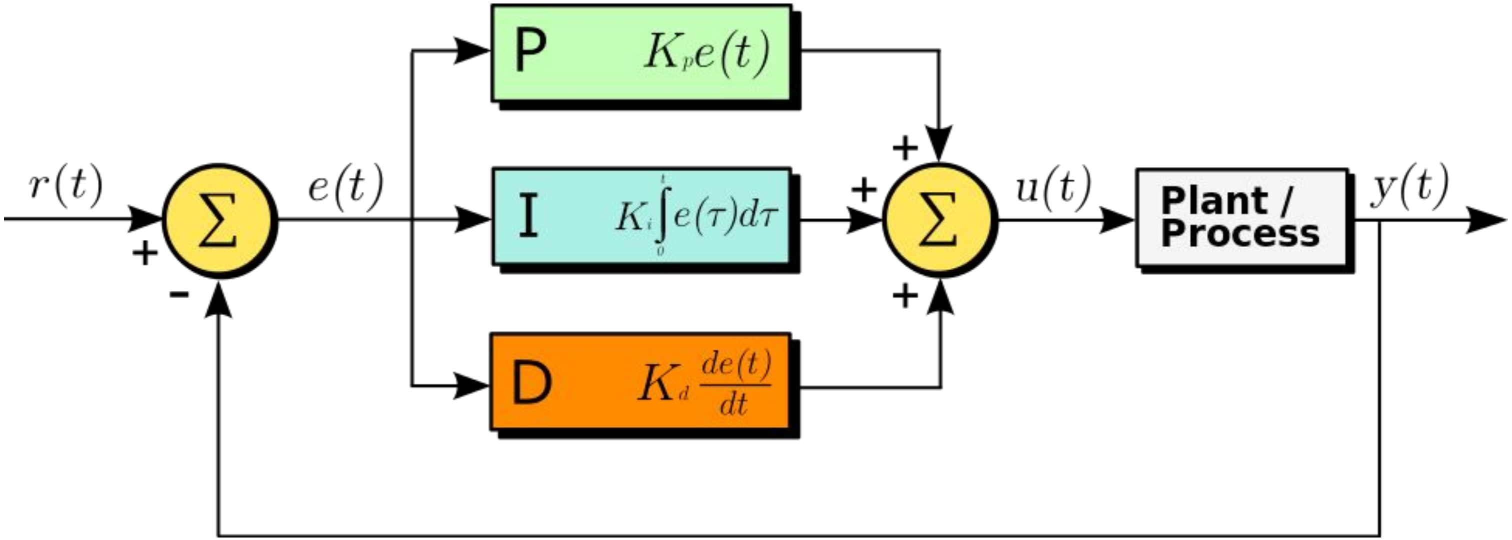
PID control

- Proportional measured the error: (left_velocity - right_velocity)

- Derivative measures error RATE: (e[t] - e[t-1])

- **Derivative Correction = K_d (Error Rate) = K_d (e[t] - e[t-1])**

PID control



(Wikipedia)

PID tuning

Choosing the parameters $\{K_p, K_i, K_d\}$

- Start with Kp
- Try different values
- More mechanical methods exist (MATLAB, Python)
- Try it out: https://sites.google.com/site/fpgaandco/pid

Summary

- e = error = (left_velocity - right_velocity)
- We want e = 0
- Proportional: PRESENT correction
- Integral: PAST correction
- Derivative: FUTURE correction

$$\mathrm{u} = \underbrace{K_p e}_{\substack{\text{Proportional}\\ \text{Term}}} + \underbrace{K_i \int_0^t e\,dt}_{\substack{\text{Integral}\\ \text{Term}}} + \underbrace{K_d \frac{d}{dt} e}_{\substack{\text{Differential}\\ \text{Term}}}$$