ENGI 128

INTRODUCTION TO ENGINEERING SYSTEMS

Lecture 10:
Feedback Control

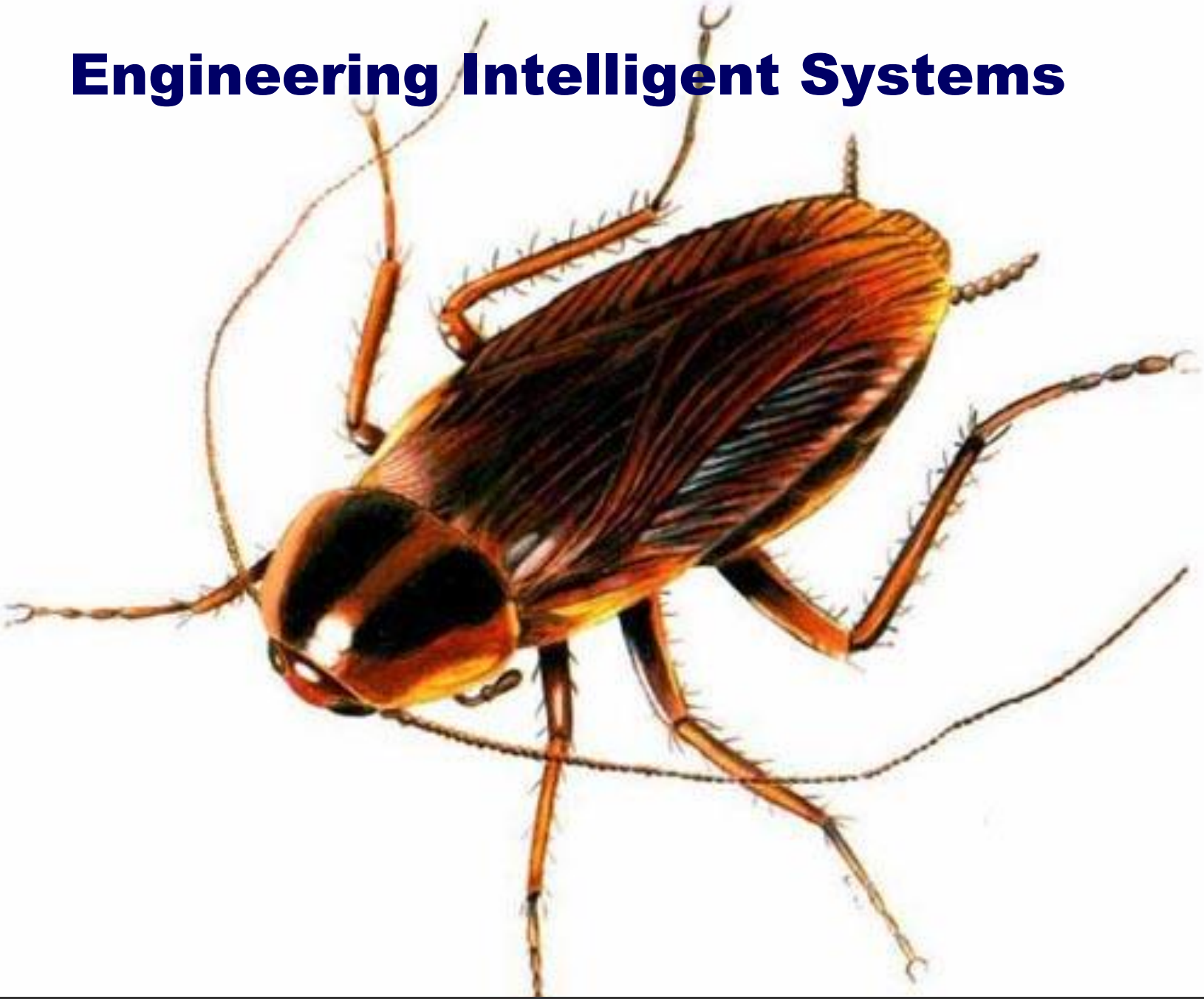
“Understand Your Technical World”

Disturbance Rejection: Cockroach Engineering

Disturbance Rejection: Artificial System

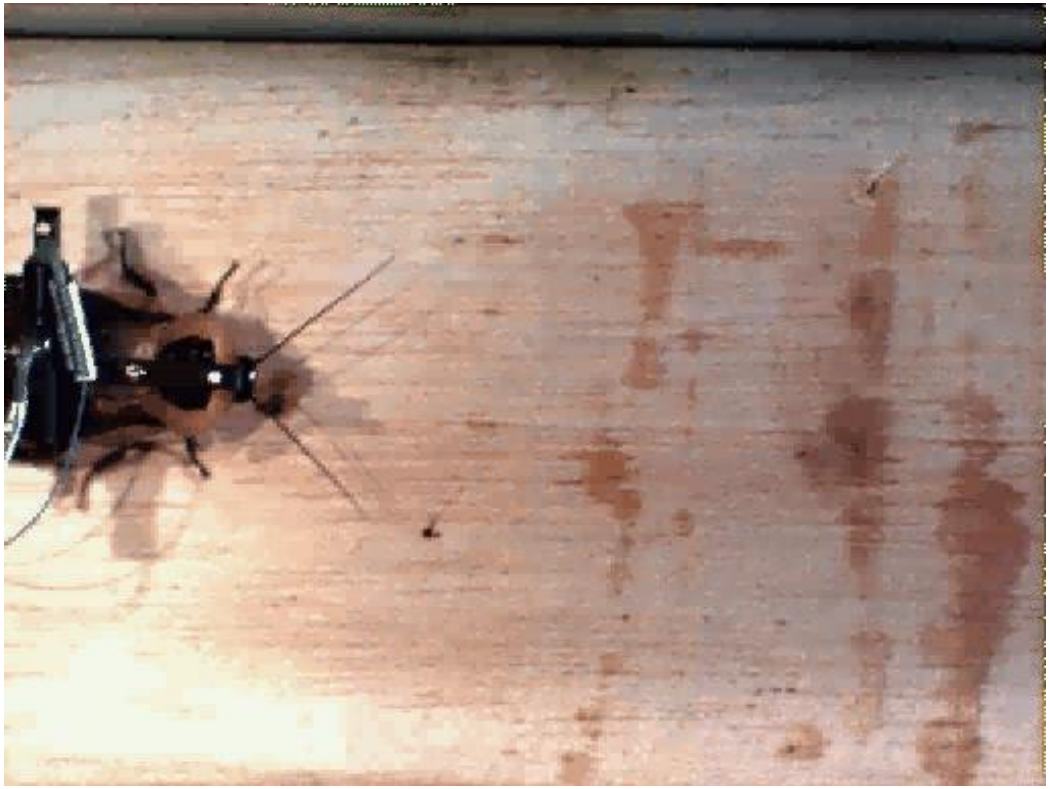
"Big Dog", www.bostondynamics.com

Engineering Intelligent Systems



Roeder KD (1948) Organization of the ascending giant fibre system in the cockroach.

Disturbance Rejection: Natural System



(Recovery in ~27ms!)

Jindrich, D. L. and Full, R. J. (2002). Dynamic stabilization of rapid hexapedal locomotion. Journal of Experimental Biology. 205,2803-2823. (Video and picture courtesy Devin Jindrich)

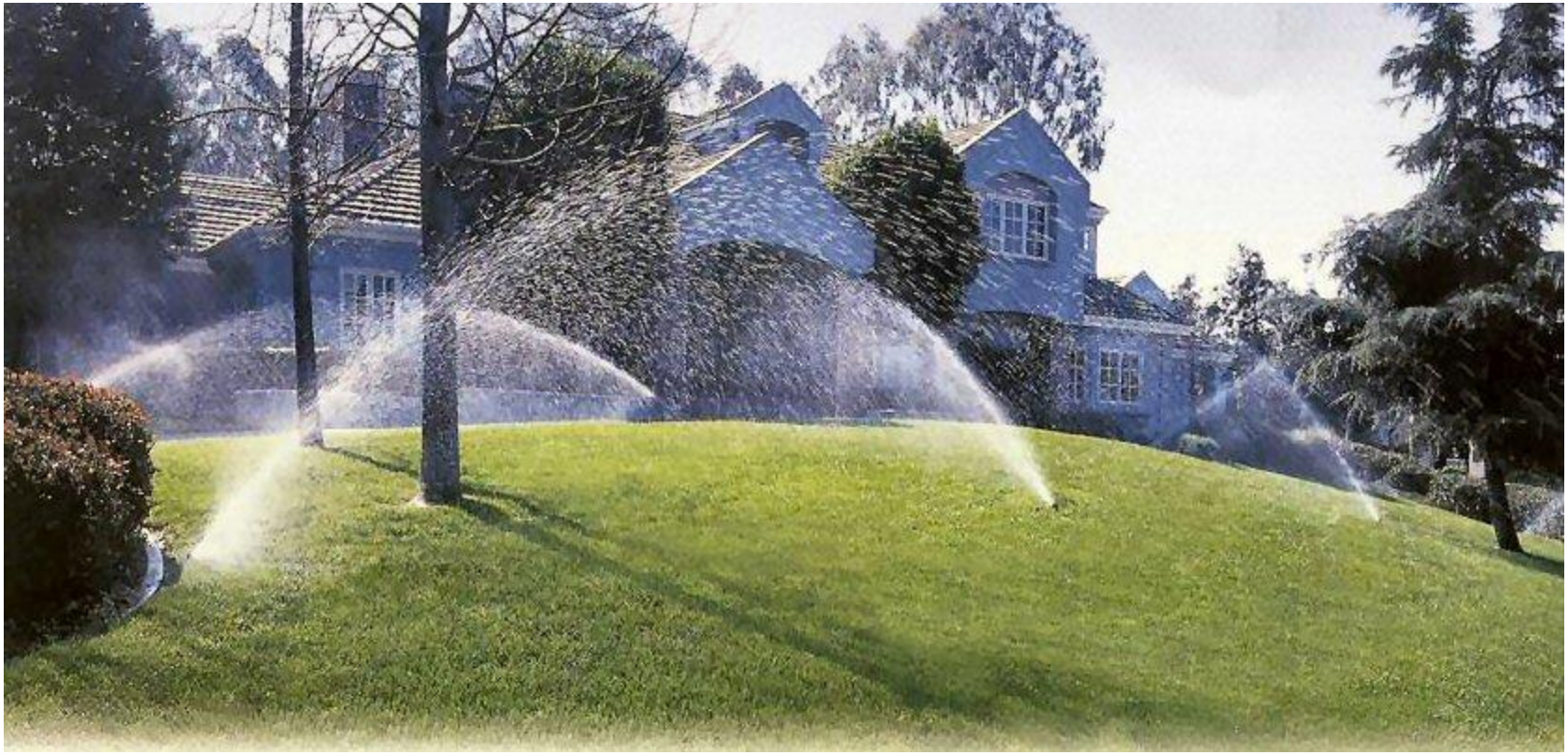
Controlling Engineering Systems

Open Loop vs. Closed Loop

Feedback vs. Feedforward

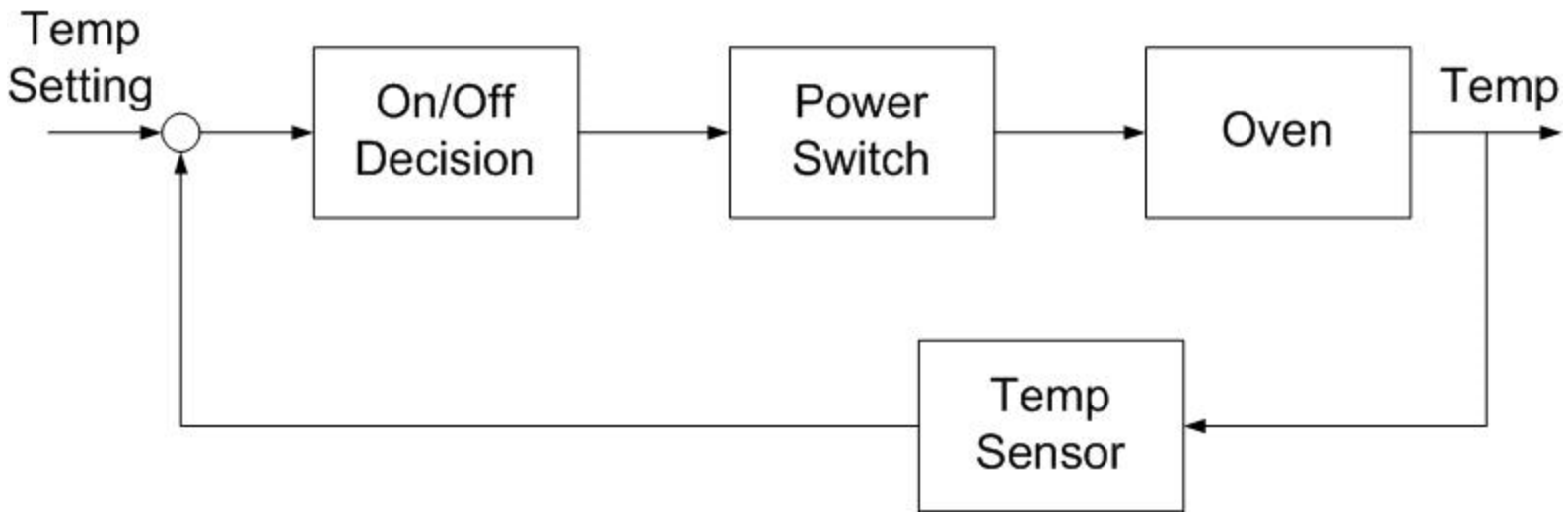
Types of Control: open loop

- Open Loop Control - sprinkler system



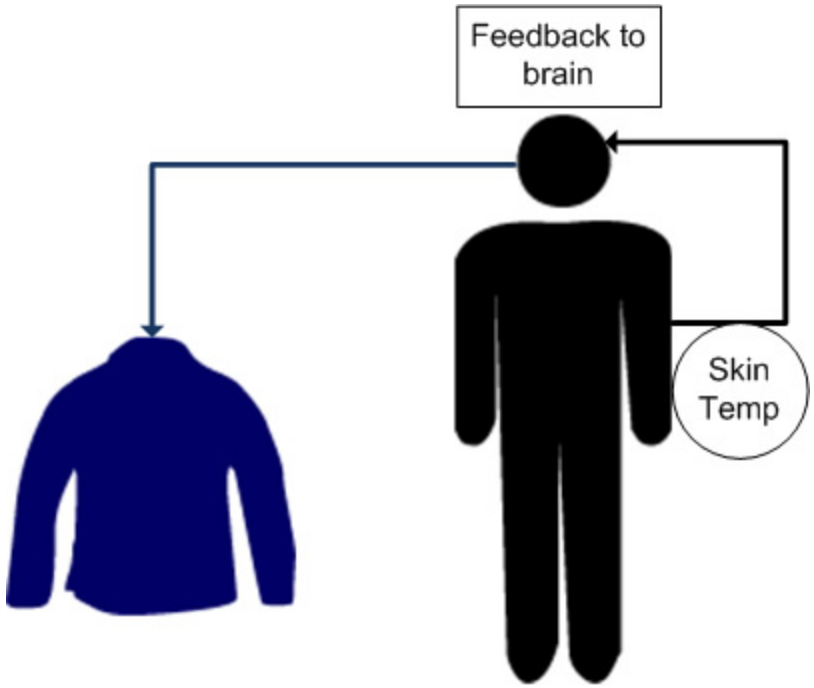
Types of Control: feedback

- Feedback control - Oven



Types of Control: feedback

- Winter growing up in Ohio



Types of Control: feedforward

- Feedforward control –
  the egg toss

Why use feedback control

- or better, why do you need a control system at all?
- consider ovens, A/C units, airplanes, manufacturing, pumping stations, etc
- What are we controlling?
  - some physical quantity (constant)
  - a dynamic behavior (a function of time)
- We need to 'tell' the system how we want it to behave

How do we ‘tell’ a system how to behave?

- turn a dial
- type a number

- But we need to know how well we are doing!
  - sensor data (temperature, pressure, speed, position, light, etc)

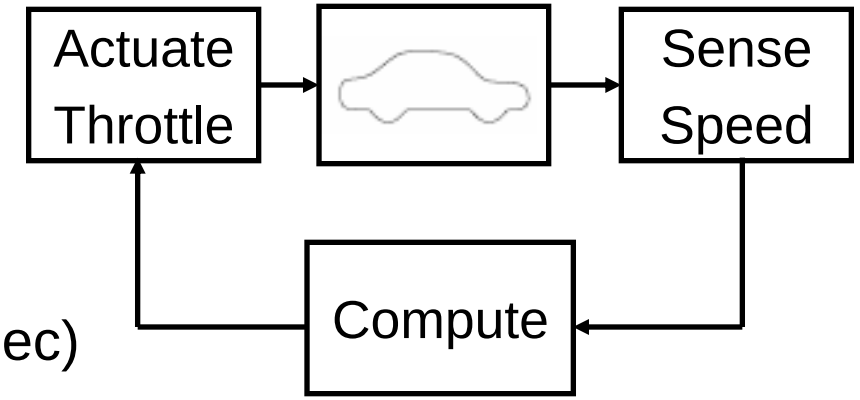
- OK, I know what I *want* the system to do. and I can monitor what the system is *actually* doing. Now what?

Control

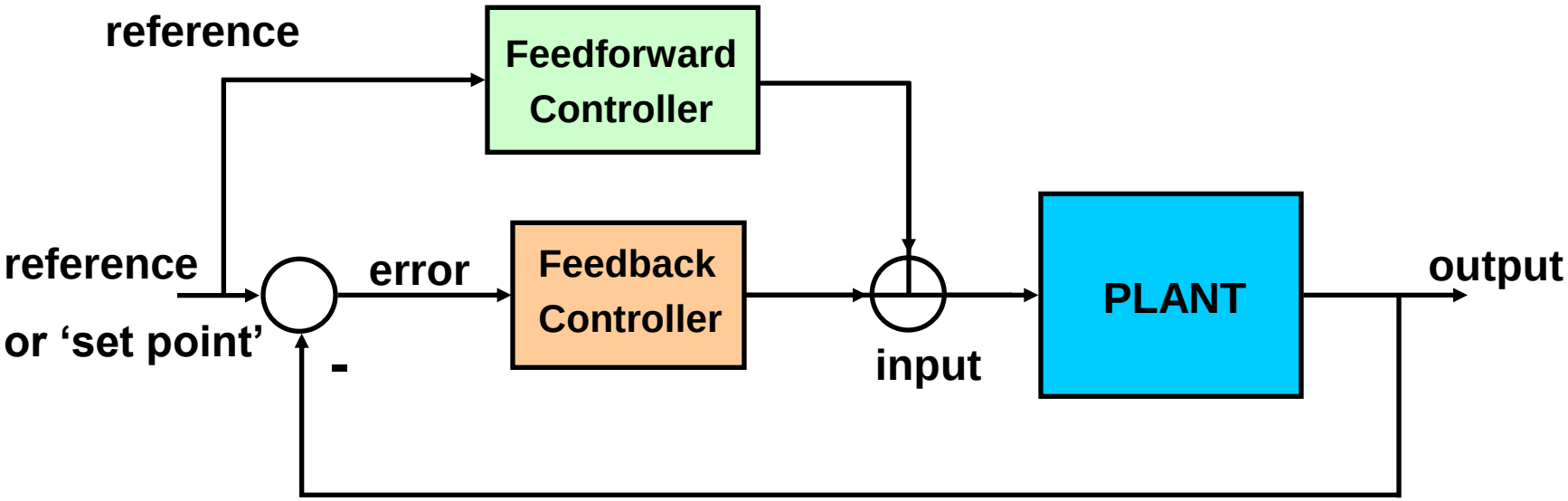
Control **= Sensing + Computation + Actuation**

In Feedback "Loop"

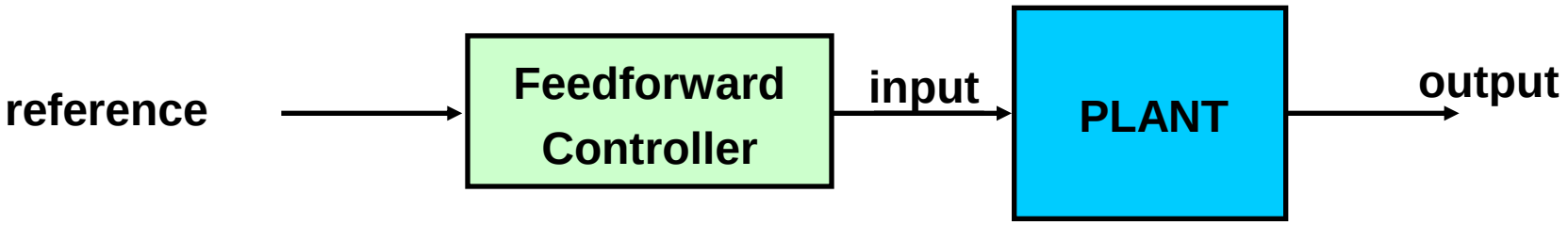
- Ensure stability
  - System maintains desired operating point (hold steady speed)
- Improve performance
  - System responds rapidly to changes (accelerate to 6 m/sec)
- Guarantee robustness
  - System tolerates perturbations in dynamics (mass, drag, etc)



Sample control systems



Feedforward control



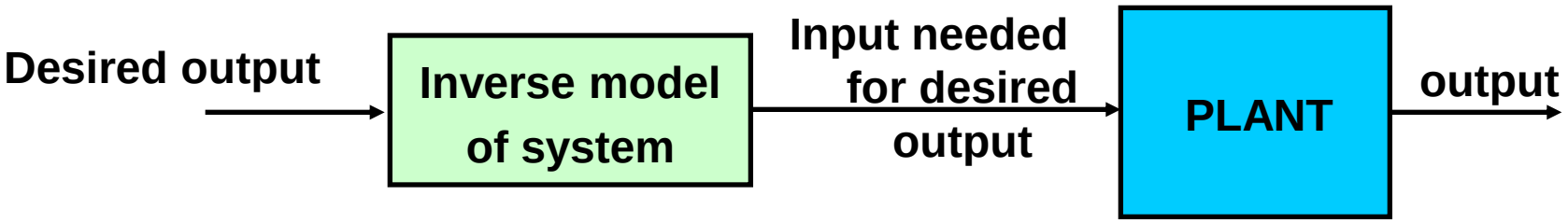
Control element responds to change in command or measured disturbance in a pre-defined way

Based on prediction of plant behavior (requires model)

Can react before error actually occurs

- Overcome sluggish dynamics and delays
- Does not jeopardize stability

One implementation of feedforward



Model-based prediction of input

Ideally consists of exact inverse model of the plant

Can compensate for known plant dynamics, delays (before you get errors)

No sensors needed

System response must be predictable

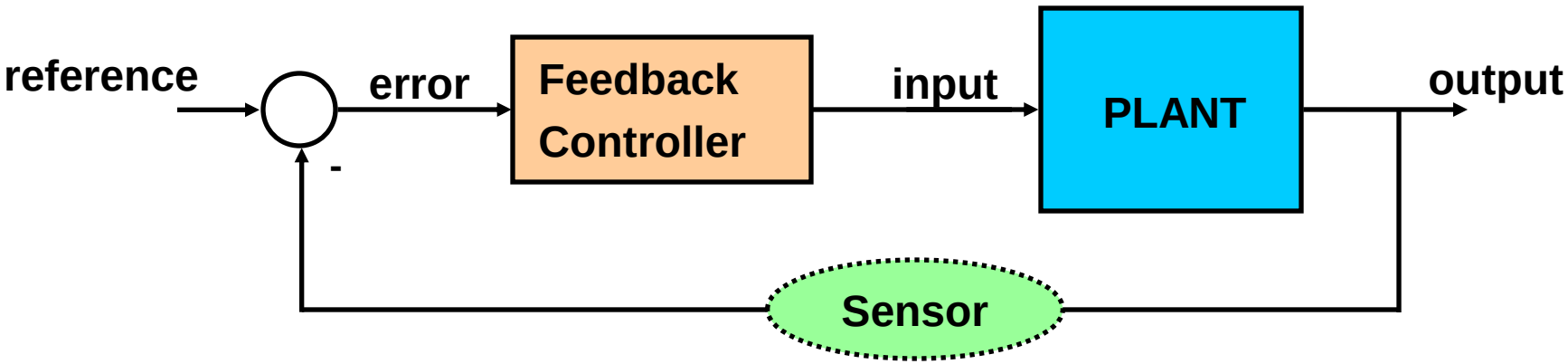
Limitations of feedforward control

Effects of disturbance or command input must be predictable

May not generalize to other conditions

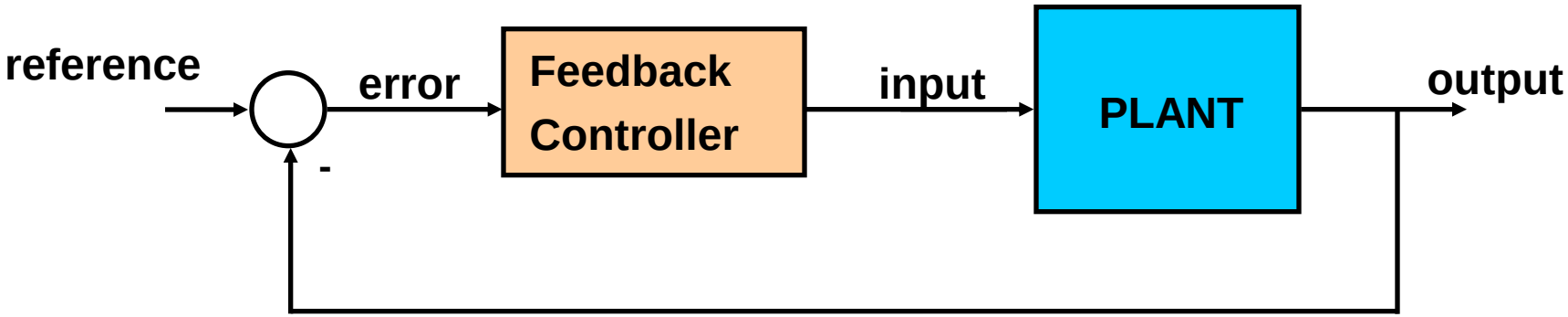
Will not be accurate if the system changes

Feedback



Plant	System to be controlled
Reference	Desired value of output (also ‘set point’)
Controller	Computes compensatory command to the plant based on error
Sensor	(implied)

Features of feedback

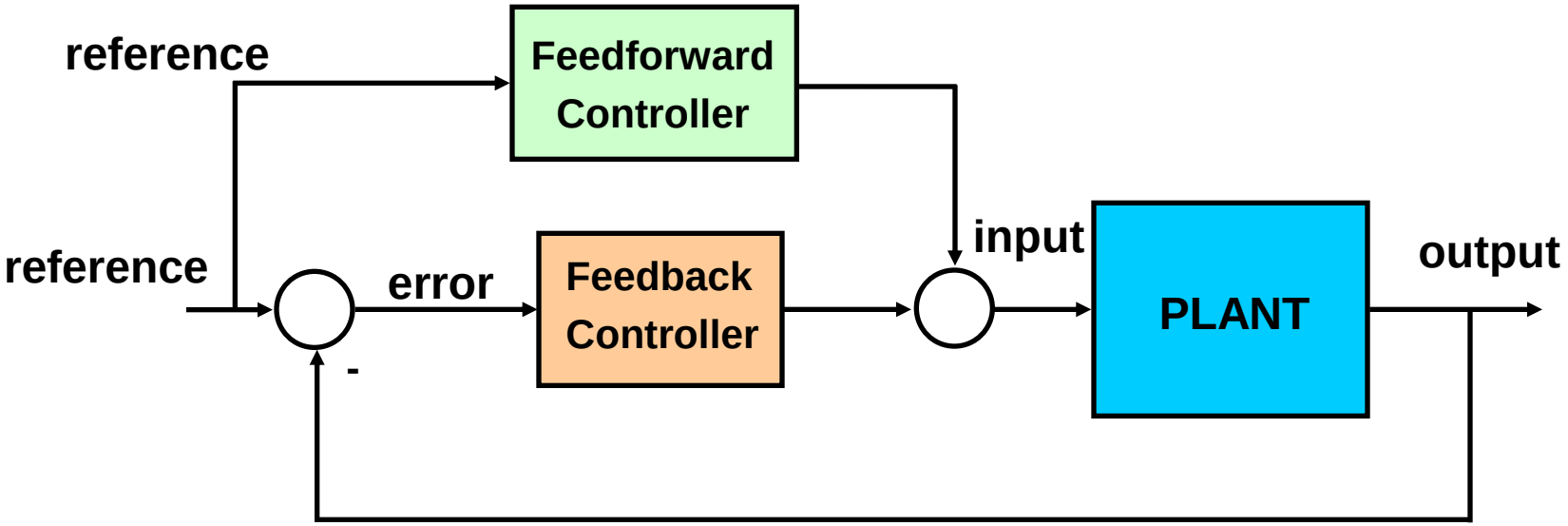


- Reactive / Error-driven
- Automatically compensates for disturbances (controller acts on error)
- Automatically follows change in desired state (set point can change)
- Can improve undesirable properties of system/plant
- Can be very simple

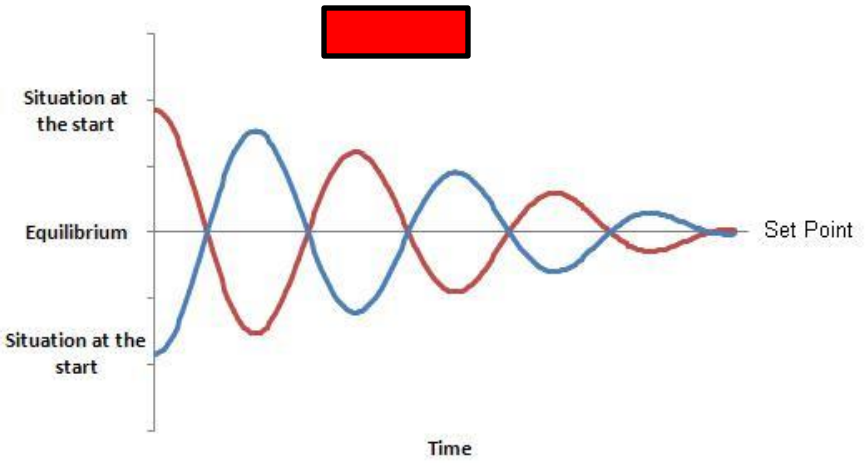
Combining feedback and feedforward

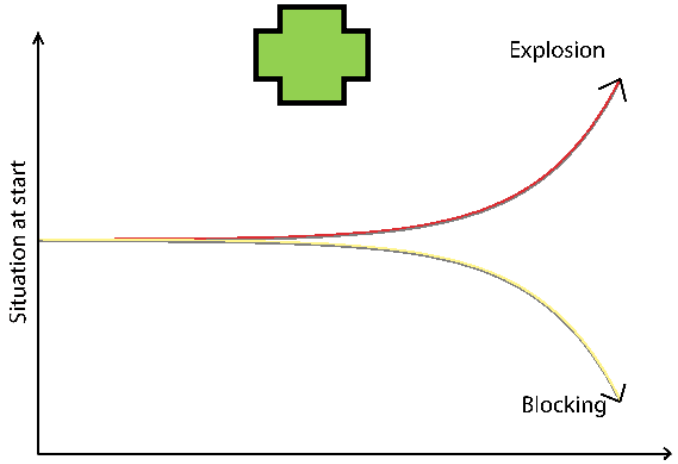
Feedforward and feedback are often used together

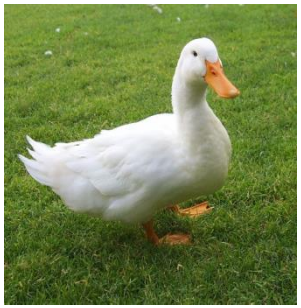
- Feedforward component provides rapid response
- Feedback component fills in the rest of the response accurately, compensating for errors in the model



Negative vs. Positive feedback



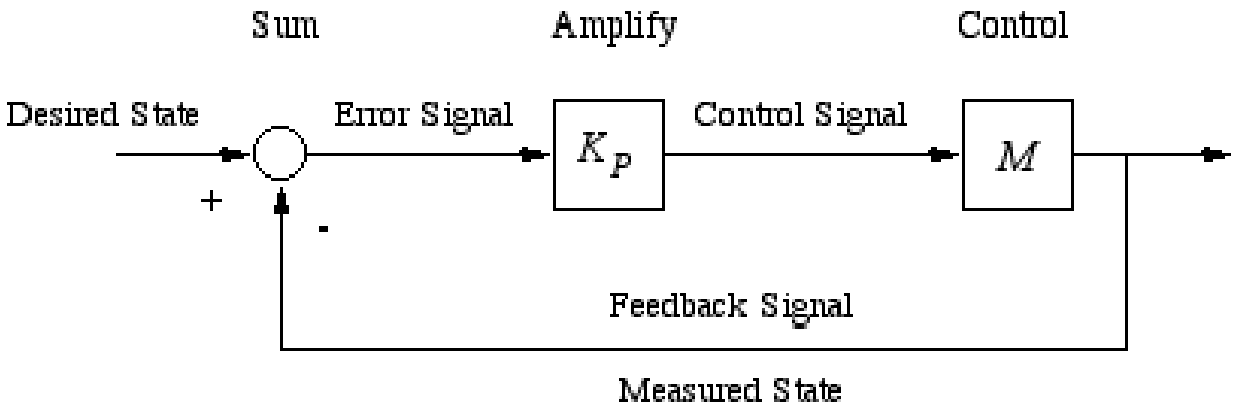




[whiteboard – PID control]

The Simplest feedback controller

- A proportional (P) controller
  - In a proportional controller, the control action is proportional to the error, and we can represent the controller as a gain, K_p.
- Represent with block diagram:

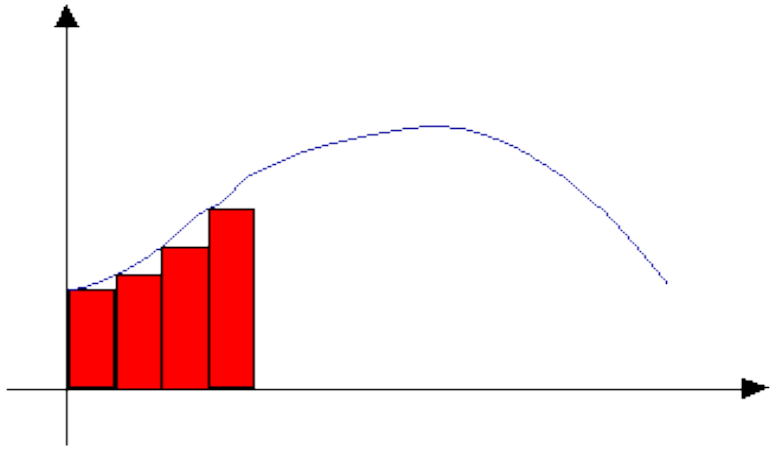


Limitations of P-control

- There are many times when you want the output of a system to be equal to the input value.
  - The proportional controller amplifies the error and applies a control effort to the system that is proportional to the error.
  - P-controller must have some error in order to provide control output
- If you want better error performance, you might want to consider using an integral controller
  - In integral control, the control effort is proportional to the integral of the error
- So what?

Integration

- An integral is really the area under a curve.
  - Let's assume that the independent variable is time, t.
  - Then as time goes on the area accumulates.



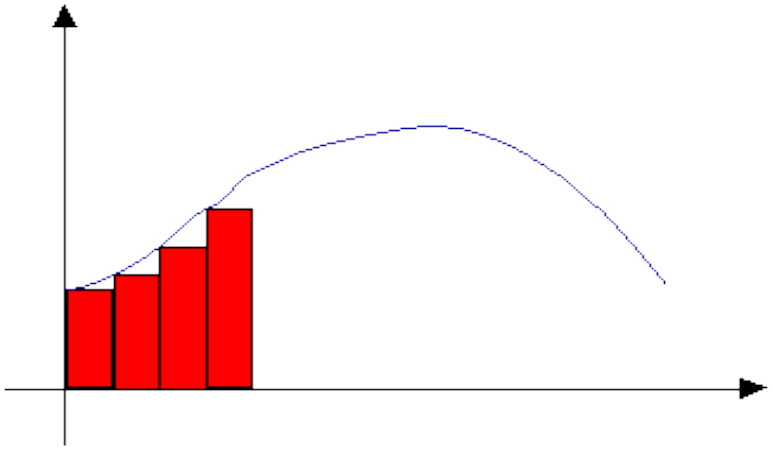
- In math courses when they talk about integration, they picture it as the limit of a process of taking small incremental areas - shown below - and letting the interval, T, shrink to zero.
- In digital integration, that visualization process is important.

Summary of Integrator behavior

- If output level matches the desired level, the error is 0
- Because error is 0, the integrator output doesn’t change
- Because the integrator output doesn't change, if the rest of the system is at steady state nothing else changes.
- This sounds too good to be true. What could possibly go wrong?

Integral control in a digital system

- Often implemented in code in some programming language like C (or Python!).
- To implement integral control you use an approximation to the integral.



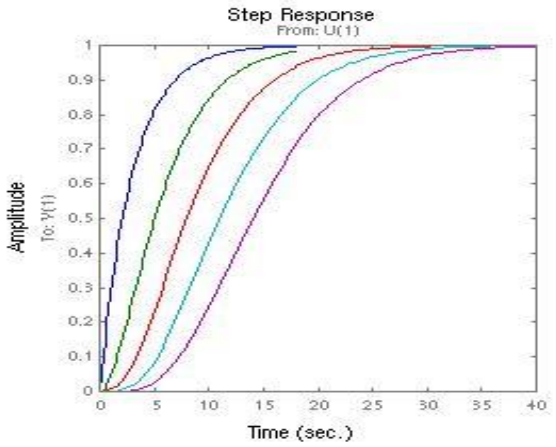
- The integral computation is updated by adding an area equal to the latest measurement multiplied by the sampling period between measurements (the width of the rectangle).

Pseudo-code for integral controller

- ErrorInt = 0 /reset the integral approximation/
- $MeasuredOutput_n$ = MeasureVolts(instrument) /Measure the output/
- $Error_n = DesiredOutput - MeasuredOutput_n$ /Compute Error/
- $ErrorInt_n = ErrorInt_{n-1} + \Delta T*(Error_n)$ /Integrate Error/
- $VoltsOut_n = IntegralGain*ErrorInt_n$ /Compute output voltage/
- OutputVoltage(VoltsOut) /Output the control signal/

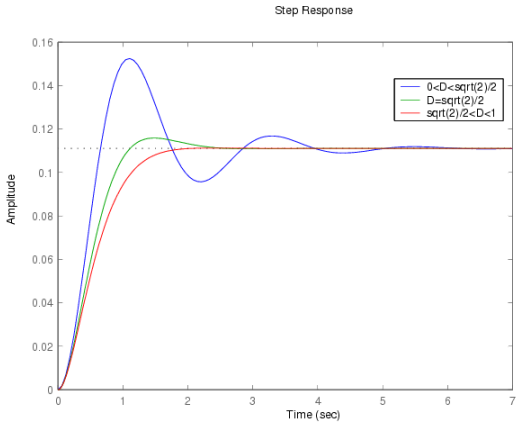
- This code assumes that you have a function - MeasureVolts - that will measure voltage using an instrument connected to the computer, and can output a voltage with a function - OutputVoltage - that uses another instrument connected to the computer.

Performance, Stability, Robustness



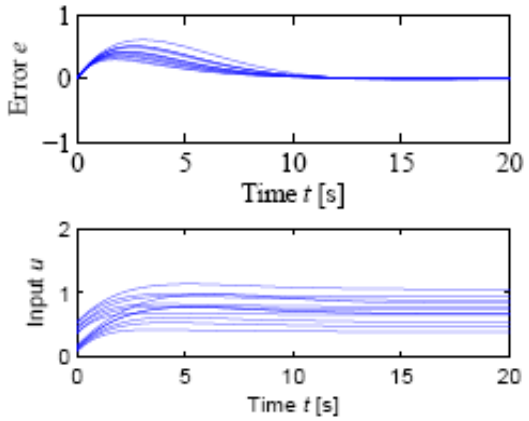
Improve performance

(e.g., response time of a first order system)



Stabilize a response

(e.g., reduce oscillatory effects)



Ensure robustness

(e.g., minimize change in output response for variations in input)

Two main principles of feedback

Robustness to uncertainty through feedback

- Allows high performance in the presence of uncertainty
- Accurate sensing to compare actual to desired, correction through computation and actuation

Design of dynamics through feedback

- Allows the dynamics (behavior) of the system to be modified
- Interconnection gives *closed loop* that modifies natural behavior
- Leverage capability to enhance performance or affect stability

Limitations of Feedback

Reactionary solution that relies on existence and observance of error

Disturbances applied to system will generate errors

Response will be delayed (disturbance rejection)

Trade-offs exist between performance and stability

Effects of delay in the feedback path lead to instabilities

Summary of closed-loop feedback control

Reactive controller based on error between desired and actual states

Automatically compensates for external disturbances and follows changes in command

Significant impact on overall system response

Used extensively in both natural and artificial systems

Limitations:

- Error must be present before actions taken
- Tradeoff between performance and stability