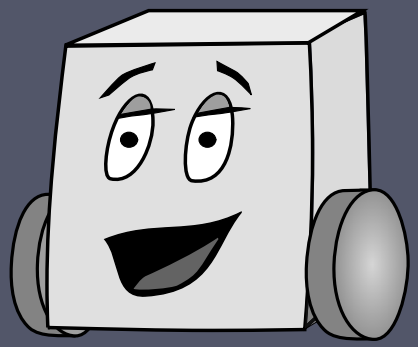
E11 Lecture 14: Feedback Control



Professor Lape

Fall 2010

Coming up in E11:

- **Wednesday:** Game kickoff and rules!
- **Monday, 11/1:** Line-following competition
- **Wednesday, 11/3**: Prof Hightower on batteries
- **Monday, 11/8**: Prof Levy on cooperative robotics
- **Wednesday, 11/10**: Thomas Valdez of JPL on fuel cells and fuel-cell-powered robots
- **Monday, 11/15**: Scrimmage
- **Wednesday, 11/17**: Prof Dodds on algorithms for robotics
- **Monday, 11/22 at 5:30 PM**: Capture the Flag Competition!

Overview

- How might you control your robot?
  - Open loop control
  - Closed loop control
    - On/Off and If-Then-Else control
    - Proportional control
- Developing control algorithms:
  - What are algorithms and how are they represented?
  - Send your robot towards the light!

Open Loop vs. Closed Loop Control

- In **open loop** control the output of the process has no effect on the input.
  - Example #1: Trying to drive your robot around a marked square based on pre-programmed timing.
  - Example #2: A heater than turns on for a set amount of time regardless of temperature.
- **Closed loop** control uses feedback (a measured value) to determine the input to the system.
  - Example #1: Driving your robot around a marked square using a line-following algorithm.
  - Example #2: A heater that turns on or off depending on measured temperature.

Block Diagrams

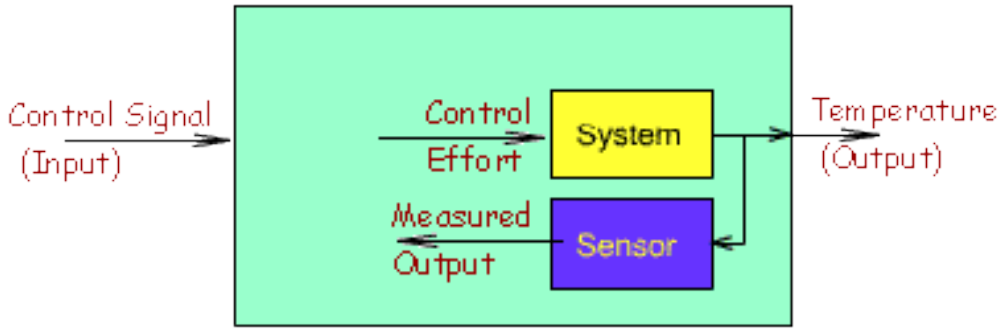
- Block diagrams are used to represent processes, sensors, controllers, and system response.

- Consider a temperature control device.
  - Overall, the device takes in an input (usually the setpoint – here, a value related to the desired temperature) that results in an output (here, the actual temperature):

http://www.facstaff.bucknell.edu/mastascu/eControlHTML/Intro/Intro1.html#WhatIsAControlSystem

Block diagrams 2

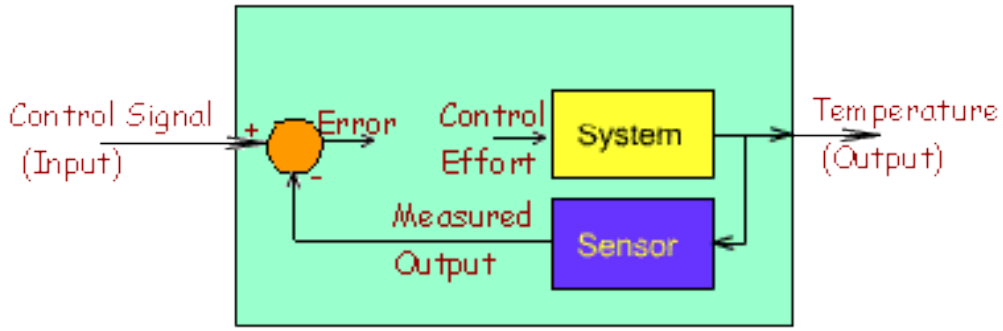
- The temperature is measured by a sensor, such as a thermocouple, which outputs a value related to the temperature (could be a voltage, for example).



http://www.facstaff.bucknell.edu/mastascu/eControlHTML/Intro/Intro1.html#WhatIsAControlSystem

Block diagrams 3

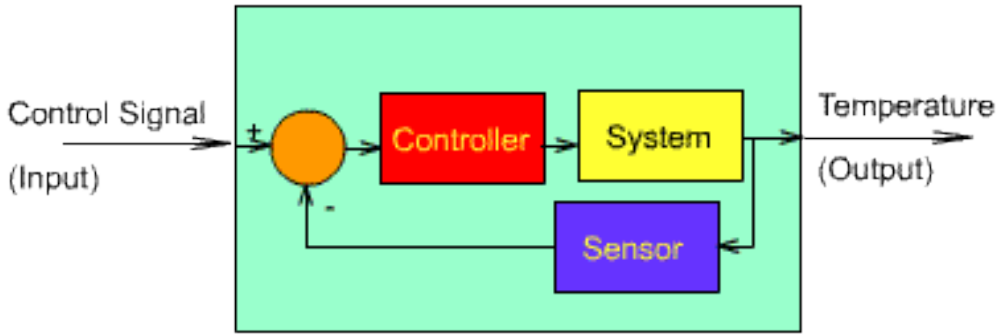
- The difference between the measured output and the setpoint is computed.
  - *Note: Rather than error, another form of comparison could be used.*



http://www.facstaff.bucknell.edu/mastascu/eControlHTML/Intro/Intro1.html#WhatIsAControlSystem

Block diagrams 4

- This difference (or results of a comparison) is fed to the controller, which determines the input to the system.



http://www.facstaff.bucknell.edu/mastascu/eControlHTML/Intro/Intro1.html#WhatIsAControlSystem

On/Off and *If-Then-Else* Closed Loop Control

- The simplest forms of feedback control check the value and executes one of two actions depending on the result.
- For **on/off control**, the controller simply turns the system on or off depending on the measured value.
  - Example: A heater turns on when the temperature drops below the setpoint and off when the temperature reaches the setpoint.
- For **if-then-else control**, the controller executes one action given a matching value and another action if the value does not match.
  - Example: Your robot proceeds forward if the reflectance sensor gives a value corresponding to a black line, but rotates to the right or left if the sensor value shows you have strayed from the line.

Proportional Control

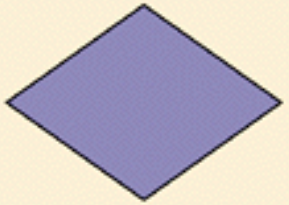
- In **proportional control**, the controller responds in a manner proportional to the error between the output and a setpoint.
  - Example #1: To control water temperature with cold and hot faucet valves, you put your hand under the faucet and open the hot (or cold) faucet valve in proportion to how much below (or above) the water is in relation to the desired temperature (setpoint).
  - Example#2: When driving, you press harder on the gas pedal when you are farther below the desired speed, then back off as you approach the desired speed.

Algorithms and Flowcharts

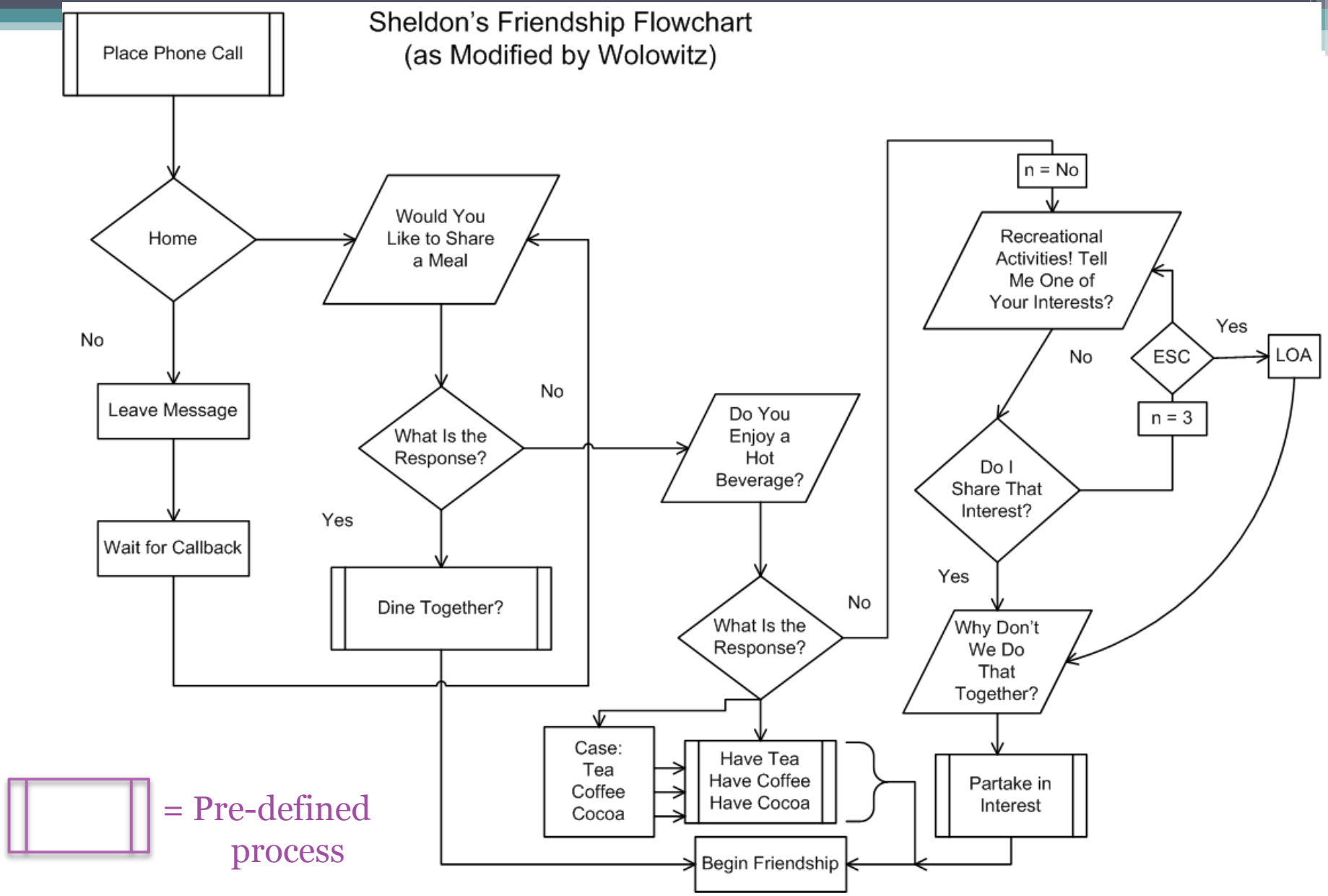
- An **algorithm** is a sequence of tasks needed to accomplish a goal.
  - An algorithm for a computational goal is not the same as the actual code used to execute the tasks.

- Algorithms are frequently represented using **flowcharts**.

Flowchart Symbols

Name	Symbol	Use in flowchart
Oval		Denotes the beginning or end of a program.
Flow line		Denotes the direction of logic flow in a program.
Parallelogram		Denotes either an input operation (e.g., INPUT) or an output operation (e.g, PRINT).
Rectangle		Denotes a process to be carried out (e.g., an addition).
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes (e.g., IF/THEN/ELSE).

http://www.wiley.com/college/busin/icmis/oakman/outline/chap05/slides/symbols.htm



http://www.nosewheelie.com/blogofthedarned/?p=52

In-class algorithm challenge

- Develop an algorithm for your robot to seek a light and head towards the brightest light (given that there may be multiple lights).
- First, optimize for each of the following using if/then/else control:
  - Speed only (maximize speed to reach the brightest light)
  - Accuracy only (maximize accuracy; this includes finding the brightest light and actually hitting the light itself, not veering off to the side)
  - Speed*accuracy (loosely, maximize the product of speed and accuracy)
- Then optimize speed*accuracy using proportional control.

Maximum Speed (If/Then/Else Control)

Maximum Accuracy (If/Then/Else Control)

Maximum Speed*Accuracy (If/Then/Else Control)

Maximum Speed*Accuracy (Proportional Control)