

EXPLORE ROBOTICS – CISC 1003



ROBOT CONTROL

CISC1003

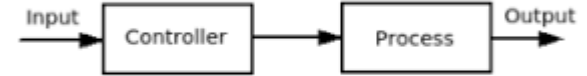
Robot Control

- So far we talked about the robot bodies
 - Sensors, effectors, etc.
- But what about their brains?
 - How do robots make decisions?

Robot Control

- Control Theory:
 - How do you actually control systems?
 - Decoupled from robotics
 - Relevant to distributed systems in general
 - How is it relevant to robots?
 - How are algorithms designed to allow robots control?

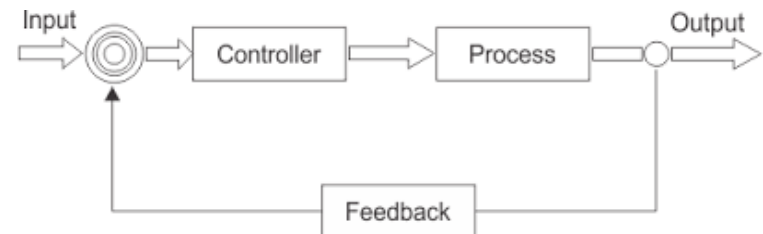
Open Loop Control



- Sends commands to make a robot perform some movement
 - without attempting to check if it is doing things properly.
- For example: a rover on Mars being told by a human operator to go forward 1 meter.
 - If the wheels get dirt in them or hit a rock the robot won't move straight.

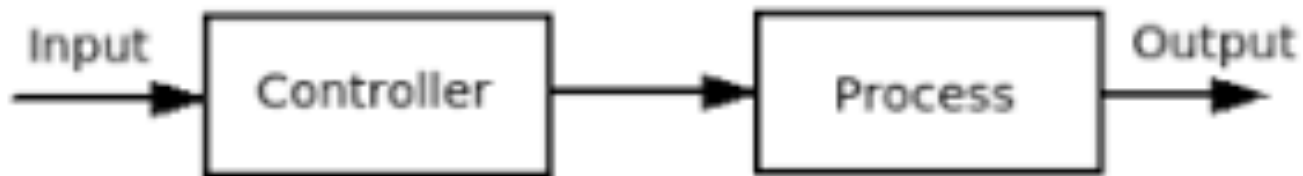
Feedback or Closed Loop Control

- Feedback control:
 - A means of getting a system (a robot) to achieve and maintain a desired state
 - State is usually called the set point
 - Achieved by continuously comparing its current state with its desired state.
- Feedback refers to the information sent back
 - literally “fed back,” into the system’s controller.



Control Mechanisms

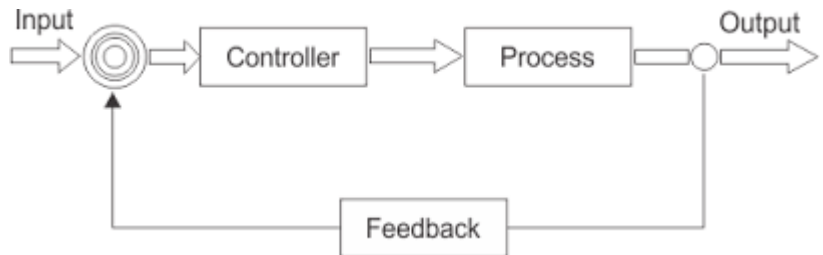
- Open Loop Control
 - Very few automatic controls or feedback system



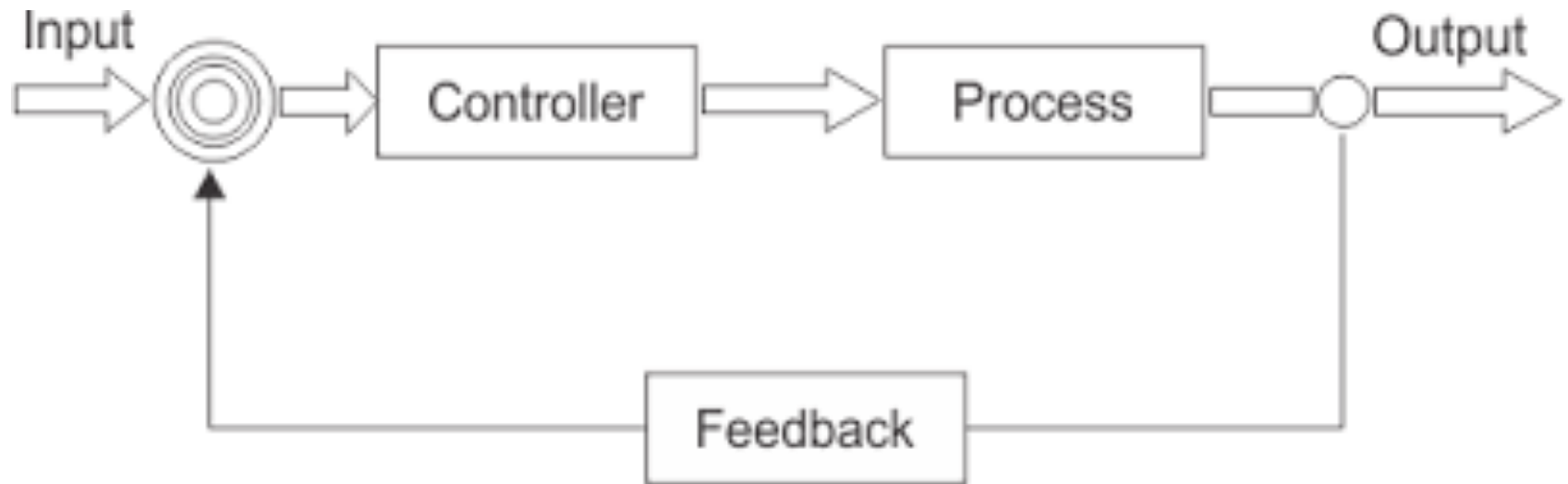
Open Loop Control System

Control Mechanisms

- Closed Loop Control
 - Has one or more feedback loops
 - Provides more accurate control of the process
 - Monitoring its output and “feeding” some of it back
 - compare the actual output with the desired output
 - Reduce the error
 - If required, bring the output of the system back to the original or desired response



Closed Loop Control



Goals



- **Goal state:** the *desired state* of the system
 - where the system wants to be.
- In AI, there are two types of goals:
 - ***Maintenance*** and ***achievement*** goals

Goals



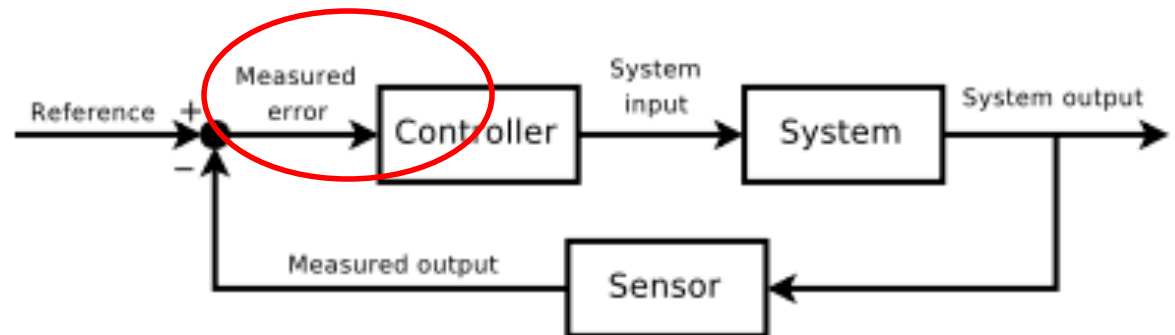
- ***Maintenance goals:***
 - Needed to reach a goal state
 - This will require ongoing active work on the part of the system.
 - Example: keeping a biped robot balanced
 - This is a maintenance goal.

Goals

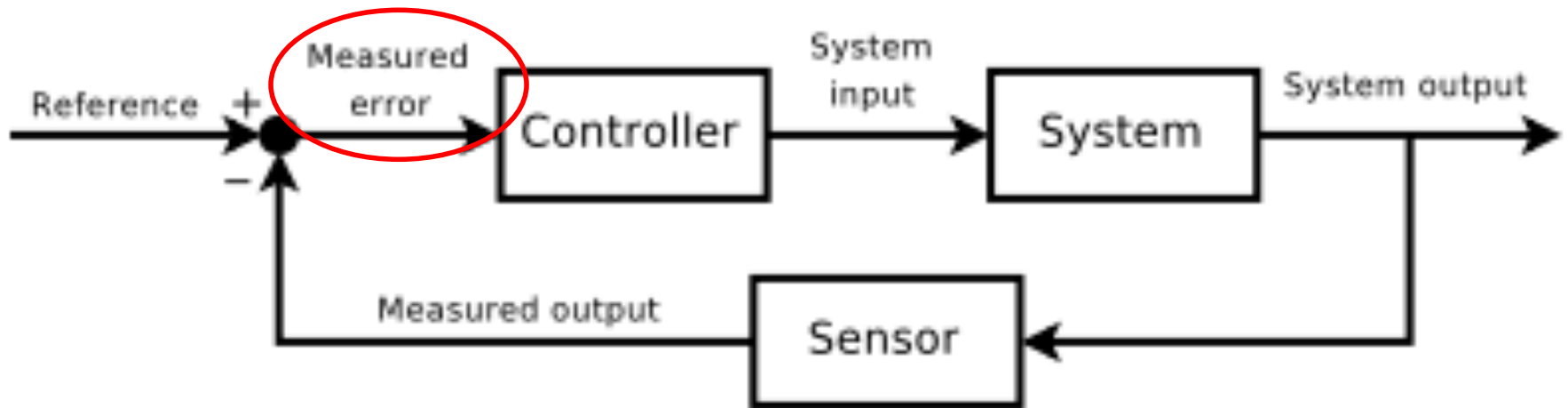
- ***Achievement goals***: states the system tries to reach
 - such as a particular location, perhaps the end of a maze
 - Once the system is there, it is done

Errors

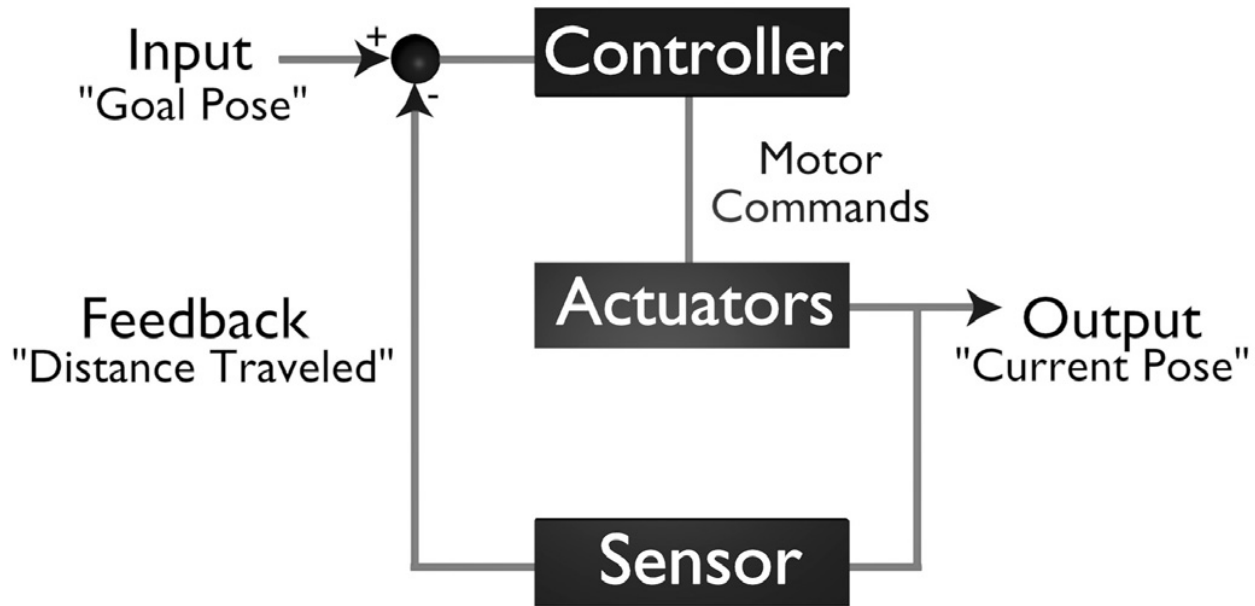
- *Error*: The difference between the current state and the goal state of a system.
 - The control system is designed to minimize error.
- Feedback control calculates the error in order to help the robot reach the goal.
 - When the error is zero (or small enough), the goal state is reached.



Errors

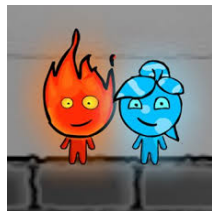


Feedback System



Feedback Example

- Real-world example of error and feedback:
 - The “hot and cold” game:
 - Player has to find or guess some hidden object (goal)
 - Participants help by saying things like “You’re getting warmer, hotter, colder, freezing” etc. (feedback)



Feedback Example

- Imagine an overly simplified version of the same game
 - Users tell you only “You are there, you win!” or “Nope, you are not there.”
 - In that case, what they are telling you is only if the error is zero or non-zero (if you are at the goal state or not).
 - This is not very much information
 - It does not help you figure out which way to go in order to get closer to the goal, to minimize the error.

Feedback Example (cont.)

- In the normal version of the game, you are being given the ***direction of the error***
 - By being told “hot” or “cold”
 - Helps minimize the error and getting close to the goal.

Feedback Example (cont.)

- When the system knows how far off it is from the goal, it knows the ***magnitude of error***
 - The distance to the goal state.
 - In the “hot and cold” game:
 - Gradations of input used to indicate the distance from (or closeness to) the goal object.
 - freezing, chilled, cool, warm, and so on

Feedback Control Robot - Example

- How would you write a controller for a wall-following robot using feedback control?



Feedback Control Robot – Example

- The first step is to consider the goal of the task.
- In wall-following, the goal state is a particular distance, or range of distances, from a wall
 - Is this a maintenance or achievement goal?
 - This is a maintenance goal, since wall-following involves keeping that distance over time.



Feedback Control Robot – Example

- What is the error in case of wall-following?
 - The difference between the desired distance from the wall and the actual distance at any time.
- Whenever the robot is at the desired distance (or range of distances), it is in the goal state.
 - Otherwise, it is not.

Feedback Control Robot – Example

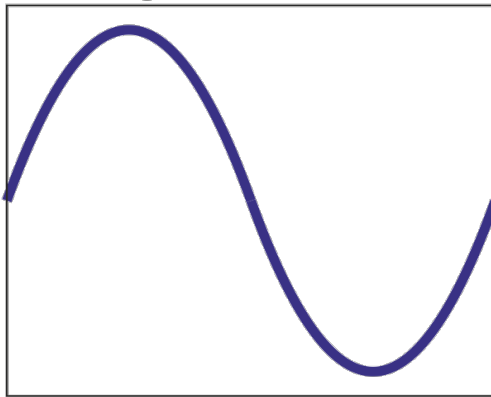
- What sensor(s) would you use for a wall-following robot?
 - what information would they provide?

Feedback Control Robot – Example

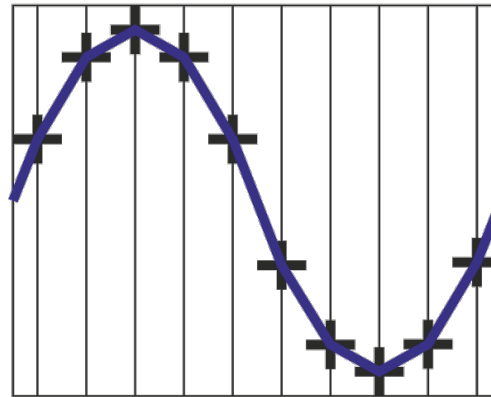
- Sampling Rate:
 - The rate with which new distance-to-wall is sensed and computed
 - Controls the interval at which sensor events are sent to your application
 - Sensors will have a maximum sampling rate
 - Part of their specifications

Sampling Rate

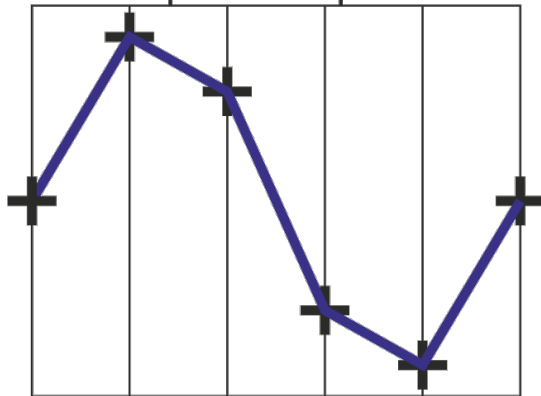
Original Waveform



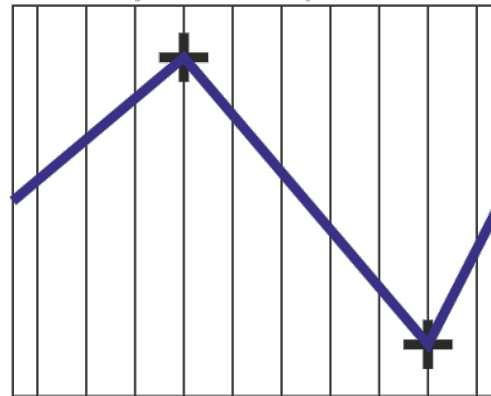
Sampled at 10 points



Sampled at 6 points



Sampled at 2 points



Feedback Control Robot – Example

- Whatever sensor is used, assume that it provides the information to compute *distance-to-wall*.
- Consider the following controller:
 - If distance-to-wall is in the desired range,
 - keep moving forward.
 - If distance-to-wall is larger than desired,
 - turn toward the wall,
 - If distance-to-wall is smaller than desired,
 - turn away from the wall

Feedback Control Robot – Example

- Given the previous controller algorithm, what will robot do?
 - The robot's behavior will keep it moving and wiggle back and forth as it moves along.
- How much switching back and forth will it do?
 - That depends on two parameters:
 - How often the error is computed.
 - How much of a correction (turn) is made each time.

Feedback Control Robot – Example

- Consider the following controller:
 - If distance-to-wall is exactly as desired,
 - keep going.
 - If distance-to-wall is larger than desired,
 - turn by 90 degrees toward the wall,
 - else
 - turn by 90 degrees away from the wall.

Feedback Control Robot – Example

- Given this algorithm, what will robot do?
 - It oscillates a great deal
 - Rarely if ever reaches the desired distance before getting too close to or too far from the wall.
 - In general, the behavior of any simple feedback system oscillates around the desired state.
 - Therefore, the robot oscillates around the desired distance from the wall
 - most of the time it is either too close or too far away.

How can we decrease this oscillation?

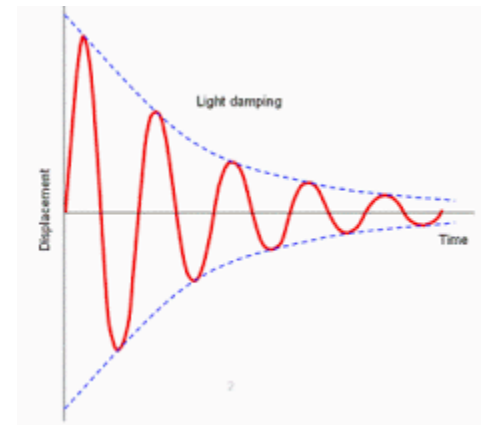
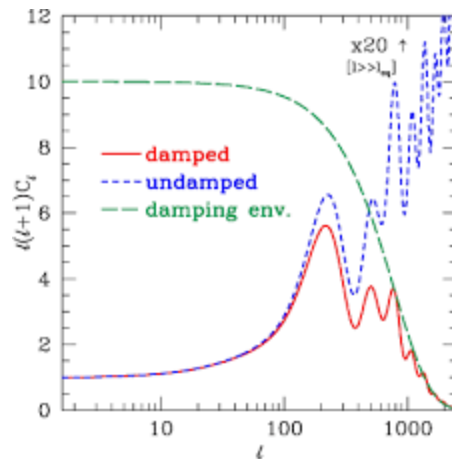
- There are a few things we can do:
 - Compute the error often, so the robot can turn often rather than rarely.
 - Adjust the turning angle so the robot turns by small rather than large angles.
 - Find the optimal range of distances that defines the robot's goal.

Feedback Control Robot

- Consider the following controller:
 - If distance-to-wall is exactly as desired,
 - keep going.
 - If distance-to-wall is larger than desired,
 - turn by 45 degrees toward the wall,
 - else
 - turn by 45 degrees away from the wall

Decreasing Oscillations (cont.)

- *Damping* refers to the process of systematically decreasing oscillations.
 - A system is properly *damped* if it does not oscillate out of control.



Decreasing Oscillations (cont.)

- Motor response to speed commands plays a key part in control, wear and tear on the gears.
 - The faster the response, the better the control



Decreasing Oscillations (cont.)

- *Actuator uncertainty* makes it impossible for a robot to know the exact outcome of an action ahead of time
 - similar to human actions and responses
 - even for simple actions
 - such as “Go forward three feet.”



