CSCI 1106
Lecture 4

Using Sensors and Actuators

Announcements

- This Friday, Jan 18, there will be a guest lecture
- The first quiz will be the following Friday, Jan 25 in class
- Could the following students please see me after class:
  Spencer Guthro, Christopher Kelloway, Hamid Hooshmandi
- Lab report
- Today's Topics
  - Noisy environment and changing world
  - Using Sensors (Polling, Sampling, Debouncing)
  - Actuators

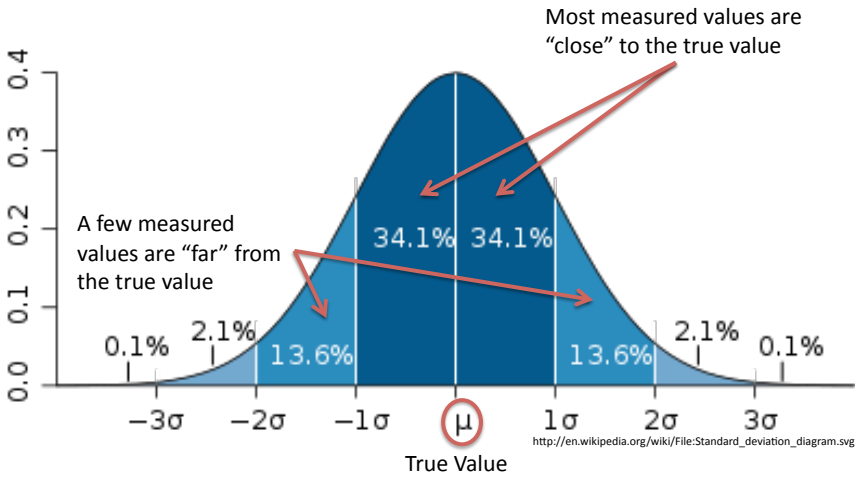
Noisy environment and changing world

- We talked about noisy sensors and how to describe them. → random variable & probability theory
- The changing sensor response due to the changing environment, specifically our uncertainty about it, can also be expressed with probability theory

Probability theory

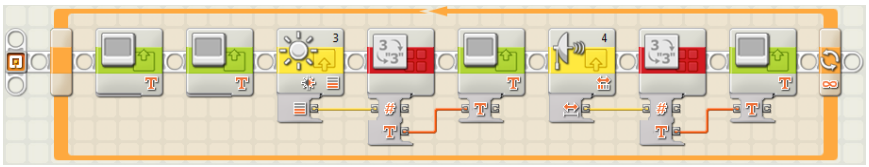
- Probability theory is the mathematical formalism to describe uncertainty (how to calculate with random variables)
- What is a random variable (as opposed to a regular variable)?
- All we can say about a random variable is encapsulated in a probability density function.

Normal Distribution



Using Sensors

- A program *polls* the sensor to get its current value
- It is the program's responsibility to *interpret* the value, i.e.,
  - Translate the value into a meaningful decision



Using Sensors (cont)

- A sensor will not inform you when a property has changed
- The program must poll the sensor repeatedly to detect change
- Usually a program polls until a *threshold* is reached

Thresholds

- We are typically not interested in what the value of a measurement is
- We are typically interested
  - when that value changes, or
  - when that value reaches a specific threshold
- A *threshold* is a fixed constant such that an event is triggered when a measurement from a sensor returns a value that is above (or below) the constant

Examples of Thresholds

- Object too close: if distance < threshold, trigger event
- Loud sound: if sound level > threshold, trigger event
- Light/Dark threshold:
  - If light level > threshold, a light surface is detected
  - if light level < threshold, a dark surface is detected
- But … How often should we poll?

Polling Frequency

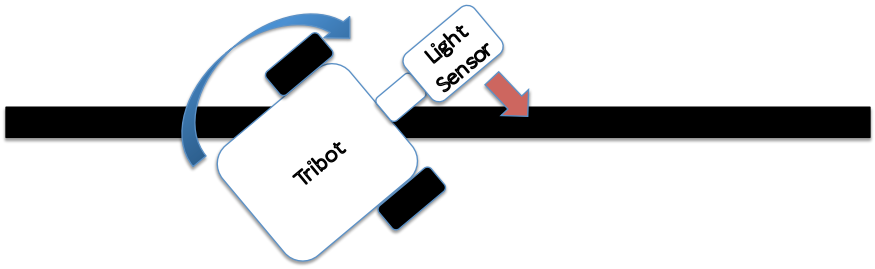
- Polling frequency depends on
  - The *response time* of the sensor
  - *The frequency of change* in the environment
- Response time dictates the maximum useful polling rate
- Frequency of change dictates the minimum rate needed to ensure that no events are missed

Polling Frequency vs Response Time

- Observation: There is no point in polling the sensor quickly if its response time is slow
  - Are we there yet? How about now? Now? Now?
- Polling the sensor too quickly does not hurt, but wastes CPU resources
- The Lego sensors have a fast response time (mostly)
- When the response time is slow, our programs need to take this into account

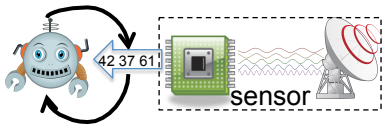
When Response Time Matters

- In Follow-The-Line
  - The angular velocity of the light sensor is quite fast
  - This could cause the sensor to move over the black line too quickly to pick it up
  - What happens if this occurs?
- How do we ensure that we don't miss the line?

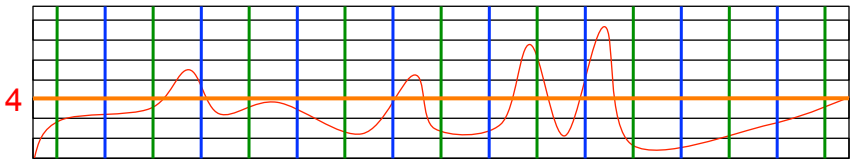


Sampling

- Sensors must be polled (sampled) for values
- The *sampling rate* is the frequency of the polls
- A higher rate means we are
  - Less likely to miss a change in inputs
  - Using more CPU time to poll the sensor
- If the rate is too high, there is no time to do anything else



Example: When is the Signal above 4?



S1	2	2.5	2.5	1.25	1.25	6	0.5	1	2.5
S2	2.25	5	2.5	4	1.25	5	0.5	2	
S*	2	2.25	2.5	5	2.5	2.5	1.25	4	1.25	1.25	6	5	0.5	0.5	1	2	2.5

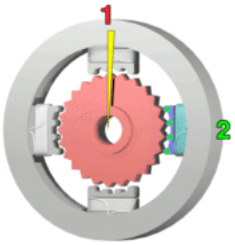
Actuators

- Actuators allow the robot to affect the world
- Actuators include:
  - Motors
  - Solenoids
  - Hydraulic mechanisms
  - Lasers
- Actuators are characterized by their parameters and tolerances:
  - Torque, force, and pressure, speed, power
  - Accuracy and precision
  - http://www.philohome.com/nxtmotor/nxtmotor.htm
- Parameters are fed into actuators as digital values
- Actuators use the parameters to control the behaviour, e.g.,
  - Turn a motor for 90 degrees at 75% power.

DC motor

Servo motor

Stepper motor



Using Actuators

- Actuators are typically used in two ways:
  - Synchronous use
    - E.g., move forward 1 rotation
  - Asynchronous use
    - E.g., start moving forward
- Synchronous use:
  - Start operation
  - Wait until the operation completes
  - Continue program
- Asynchronous use:
  - Start operation
  - Continue program
  - Use sensors or poll actuator to determine operation completion

Motors

- Motors create rotation
- Motors have following parameters:
  - *Direction*: forward or backward rotation
  - *Duration*: measured in
    - number of complete rotations
    - degrees (360 degrees = 1 rotation)
    - time (in seconds)
    - unlimited (rotate until stopped)
  - Power: (0 slow) ... (100 full speed)
    - more power implies more forward momentum
    - more power implies less accuracy and precision
  - Brake: use brake or coast after action
- Motors have rotation sensors (speed and distance)
  - Reports the duration of rotation

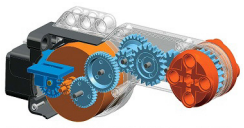


Illustration of the motor's inner workings
http://www.syraweb.org