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. \(\rightarrow\) 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.
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?

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>2</td>
<td>2.5</td>
<td>2.5</td>
<td>1.25</td>
<td>1.25</td>
<td>6</td>
<td>0.5</td>
</tr>
<tr>
<td>S2</td>
<td>2.25</td>
<td>5</td>
<td>2.5</td>
<td>4</td>
<td>1.25</td>
<td>5</td>
<td>0.5</td>
</tr>
<tr>
<td>S*</td>
<td>2.25</td>
<td>5</td>
<td>2.52</td>
<td>5</td>
<td>1.25</td>
<td>5</td>
<td>0.50</td>
</tr>
</tbody>
</table>
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
- 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.

![Illustration of different types of motors]

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