CSCI 1106
Lecture 11

Introduction to Robotics
Announcements

Today's Topics

- Overview of Robotics Module
- What is Robotics?
- Anatomy of a Robot
- The Sense-Decide-Act Cycle
- Introduction to the Aseba Studio
The Robotics Module

Topics

• Overview of Robotics
• Hardware
  – Sensors
  – Actuators
• Software
  – Event Based Architecture
  – Dealing with Failure
  – Planning
  – Debugging
  – Programming Techniques
• Mindware
  – State Transition Diagrams
  – Motion model

To Do List

• Six tutorials:
  – Introducing the Thymio II
  – Modeling sensors
  – Modeling actuators
  – Modeling the real world
  – Recovering from faults
  – Programming Techniques
• Robot Olympics Project
  – Design three programs to compete in the Robot Olympics
    • Marathon, hurdles, and curling
  – Implement the programs
  – Compete in the Robot Olympics
  – Write a technical report
What is Robotics?

• From the Thrun, Burgard, and Fox “Robotics is the science of perceiving and manipulating the physical world through computer-controlled devices.”

• A robot is composed of
  – Hardware: the machine
  – Software:
    the program that controls the machine
    (Robotics middleware like ROS, SLAM, CV, etc)

• Robotics includes both aspects.
Anatomy of a Robot

• Thymio II robot
  – https://aseba.wikidot.com

• Components:
  – Sensors
  – Controller
  – Actuators
Sensors and Actuators

- Li-Po battery level
- loudspeaker
- microphone
- infrared remote control receiver
- 3-axis accelerometer
- 5 proximity sensors
  - obstacle detection
- 2 ground sensors
  - line following
- 39 LED
  - visualize sensors and interactions
- 5 capacitive touch buttons
  - activity display
  - and ON-OFF function
- pencil support
- USB connection
  - programming and recharging
- memory card slot
- hook for trailer
- 2 proximity sensors
- mechanic attachment
- 2 wheels
  - speed control
- temperature sensor
- reset button
The Sense-Decide-Act Framework

Sense → Decide → Act → Sense
Subsumption architecture

Fig. 2.1 An example of an subsumption architecture. From Maja J. Mataric, *The Robotics Primer*, MIT Press 2007
What is the Robot Doing?

- What is being sensed?
- What is being decided?
- What action(s) result?

- -> state estimation !!!!
https://www.youtube.com/watch?v=ASoCJTYgYB0

https://www.youtube.com/watch?v=6wK0Ld13US8
Programming in Aseba

• Programs are text-based

• Key Ideas:
  – Everything is done by event handlers
  – A robot is a sprite
  – The world is the stage

• Observation this is similar to game design!
Scratch vs Aseba

**Scratch**
- Variables
- Event Handler
- Conditional

**Aseba**
- Variables
  - `var name`
  - `var list[]`
- Event Handler
  - `onevent prox`
- Conditional
  - `if
  then
end`
Scratch vs Aseba

**Scratch**
- Variable/List Assignment
- Expressions
- Motion

**Aseba**
- Variable/List Assignment
- Expressions
- Motion

Example:
- Scratch: `set name to 5`
- Aseba: `name = 5`

Example:
- Scratch: `change name by 5`
- Aseba: `name += 5`

Example:
- Scratch: `insert at 1 of list`
- Aseba: `list[] =`

Example:
- Scratch: `(name + 5) * item 2 of list`
- Aseba: `((name + 5) * list[2])`

Example:
- Scratch: `move steps`
- Aseba: `motor.left.target =`
The Four Parts of an Aseba Program

• Variable declarations
  – Begin with the \textit{var} keyword

• Initialization code
  – Anything except declarations

• Event handlers
  – Begin with the \textit{onevent} keyword

• Subroutines
  – Begin with the \textit{sub} keyword
Sensors

prox.horizontal[0-4] \{0...\sim 4300\}
prox 10 Hz

button.forward \{0,1\}
button.forward pressed or released

button.left \{0,1\}
button.left pressed or released

button.center \{0,1\}
button.center pressed or released

temperature \{1/10 °C\}
temperature 1 Hz

button.backward \{0,1\}
button.backward pressed or released

prox.horizontal[5-6] \{0...\sim 4300\}
prox 10 Hz

prox.ground.delta[0-1] = reflected-ambient
prox.ground.reflected[0-1] \{0...1023\}
prox.ground.ambient[0-1] \{0...1023\}
prox 10 Hz

buttons 20 Hz

button.right \{0,1\}
button.right pressed or released

rc5.address
rc5.command
rc5 signal received

mic.threshold \{0..255\}
mic.intensity \{0..255\}
mic mic.intensity>mic.threshold

sound.record(N)
N: \{0...32767\}, record as 'rN.wav', N=-1, stop recording

acc[0-2] \{-32...32\}, 23=1g
acc 16 Hz

tap shock detected
Actuators
A Sample Program

```
var speed = 100

motor.left.target = 0
motor.right.target = 0

onevent button.forward
    motor.left.target = speed
    motor.right.target = speed

onevent button.backward
    motor.left.target = 0
    motor.right.target = 0

onevent button.left
    motor.left.target = -speed
    motor.right.target = speed

onevent button.right
    motor.left.target = speed
    motor.right.target = -speed
```

Key Idea: Actuators are controlled by setting variables that represent them.
Aseba Studio

Sensors and Actuators in Aseba

• **Key Idea:** All sensors and actuators are accessed via predefined variables, e.g.,
  
  – to control motors, assign values to motor variables
    
    ```
    motor.left.target = 100
    motor.right.target = 100
    ```
  
  – to check if an object is close, read proximity variable
    
    ```
    if prox.horizontal[2] > 1000 then
      ...
    end
    ```

• Where are all the predefined variables listed?
• When do we check variables?
When do We Check the Sensors?

• Key Idea: Sensors generate events. Event handlers check sensors. E.g.,
  – Proximity (prox) sensors generate 10 events per second
    
    ```
    onevent prox
      if prox.horizontal[2] > 1000 then
        motor.left.target = 0
        motor.right.target = 0
      else
        motor.left.target = 100
        motor.right.target = 100
      end
    ```

• Scratch and Aseba are very similar!
Event Driven Framework
(Wait) Sense (Event)-Decide-Act

Wait → Sense → Decide → Act
**Last Example**

onevent prox

```plaintext
if prox.horizontal[2] > 1000 then
    motor.left.target = 0
    motor.right.target = 0
elseif prox.horizontal[4] > 1000 then
    motor.left.target = -100
    motor.right.target = 100
elseif prox.horizontal[0] > 1000 then
    motor.left.target = 100
    motor.right.target = -100
else
    motor.left.target = 100
    motor.right.target = 100
end
```