Today’s topics

- History of Robotics
- Uses of robots
- The RCX
- ROBOLAB

- Upcoming
  - Basic control
  - Kinematics
  - Robot architectures
What is a robot?

● Definitions
  ➤ Webster: a machine that looks like a human being and performs various acts (as walking and talking) of a human being
  ➤ Robotics Institute of America: a robot is a reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks
  ➤ What's our definition

● Components of a robot system?
History

• 1921: Karel Capek’s play, Rossum’s Universal Robots
• 1942: Asimov wrote Runaround which contained the “Three Laws of Robotics”
  1. A robot must may not injure a human being or through inaction allow a human being to come to harm.
  2. A robot must obey the orders given it by human beings, except where such orders would conflict with Rule 1.
  3. A robot must protect its own existence, as long as such protection does not conflict with Rules 1-2.
• 1948: Weiner wrote “Cybernetics”
• 1961: General Motors’ puts UNIMATE online (first industrial robot)
• 1970: SRI’s Shakey: first AI mobile robot
Uses of robots

- Where and when to use robots?
  - Tasks that are dirty, dull, or dangerous
  - Where there is significant academic and industrial interest
- Ethical and liability issues
- What industries?
- What applications?
Teleoperation

- Remote control of robot manipulators
- Intermediate solution to tasks that currently could or should not be handled with autonomous robots
- Various levels of abstraction
- Issues
  - Control latency
  - Communication bandwidth
  - Operator training
Control basics

- **Some definitions:**
  - Control system: arrangement of physics components connected or related in such a manner as to form and/or act as an entire unit
  - Kinematics: the description or study of the geometry of motion
  - Dynamics: the description or study of the forces that affect the motion of objects

- **Open-loop control**
  - Compute trajectory a priori and make necessary actions to complete task

- **Closed-loop control**
  - Use sensors to provide feedback to modify the trajectory and actions
Agents and Environments

Diagram:
- Environment
- Percepts
- Actions
- Effectors
- Sensors
- Agent
Computer architecture

von Neumann model

- **Memory**: random access memory (RAM) for program instructions and data
- **ALU**: includes set of registers for performing calculations
- **Control**: responsible for fetching and decoding instructions
- **Input & output**
- **Bus**: various internal pathways
The RCX

- Hitachi H8/3297 series processor
- 3 inputs/sensors (1, 2, 3)
- 3 outputs/motors (A, B, C)
- 32k RAM/ 16k ROM
  - 12 kB free in leJOS
- Multiple threads of execution
- LCD Display & Speaker
- 3 programmable buttons
- IR send/receive
- Sensors
  - Light, touch, rotation
  - Sonar and compass avail
Working with RCX

- **RCX needs firmware on it in order to be programmable**

- **Programming process**
  1. Design program
  2. Implement on PC
  3. Download onto RCX using IR tower
     - Code converted to bytecodes that RCX understands
  4. Test
  5. Debug and go back to step 1 or 2
ROBOLAB

- **Pilot Mode**
  - graphics based
  - limited capability
  - great for kids?

- **Inventor Mode**
  - variables
  - conditionals
  - loops
  - multitasking

- **Investigator Mode**
  - Charts & graphs
  - Analyze data
Inventor
Investigator - Program Area
Investigator - Data area

[Diagram of the Investigator software interface showing a graph of light percentage over time.]
Challenges

1. Getting there
   - Using Pilot 1 - program your car to move for 1 sec
   - Measure the distance it went
   - Predict distance for n sec (Pilot 2 may be useful)
   - Run and check model

2. Touch-activated
   - Using Pilot 4 make it so your robot starts when the touch sensor is pressed and stops when it hits something
   - Can you keep your robot from running off the table with a light sensor?