Robot morphology (shape)

• Some possibilities
  – Humanoid
  – Trashcan robots
  – Reconfigurable
  – Shape shifting
  – Biped
  – Quadruped
  – …
Why does shape matter?

- Shape influences expectations for interaction
- Differing capabilities with differing robot shapes
- Need to select shape to fit a task
Environments for robots

• Terrestrial
• Aquatic
• Airborne
• Space
Locomotion: Wheels

- **Differential Drive**
  - Two wheels mounted on a common axis controlled by separate motors

- **Synchronous Drive**
  - Usually three wheels, all of which can be driven and steered, can be used for holonomic motion

- **Ackerman Steering**
  - Car drive: front steering wheels turn on separate arms
Locomotion: Tracks

- Usually two treads
- May have ability to change shape
Locomotion: Legs

- Varying numbers of legs: 1, 2, 4, 6, etc.
- Usually more difficult to control than wheeled robots: need to consider gait and balance
- Can traverse areas that wheels can not; argument is that the world has been engineered for people
Sensors

• Measure internal or external state of the robot
• Internal state, also called proprioception
  – Battery level
  – Wheel rotations
  – Position of gripper (e.g., open, closed)
  – Etc.
• External state:
  – Location of objects with respect to the robot
  – Color of objects
  – Location of wireless base stations
  – Etc.
Sensors

- **Active**
  - Emit energy into the environment to make observations
  - E.g., infrared, sonars
  - Don’t confuse active sensors (devices that emit energy) with active sensing (e.g., pan-tilt camera)

- **Passive**
  - Receive already existing energy from environment
  - E.g., bump sensor, video camera (without additional lighting)
Sensors

- Sensors which measure same form of energy and process it in similar ways form a modality
- Modalities refer to the raw input
  - Sound
  - Pressure
  - Temperature
  - Light
Types of Data Returned

- Range  
  (e.g., distance to objects)
- Absolute position  
  (e.g., latitude and longitude)
- Environmental  
  (e.g., temperature or color of objects)
- Inertial  
  (e.g., acceleration)
Sensors: Inertial Sensors

- **Accelerometers**
  - Measure acceleration in a single direction
- **Gyroscopes**
  - Measure change in orientation
- **Compasses**
  - Measure position relative to magnetic north
- **Inclinometers**
  - Measure orientation of the gravity vector
Sensors: Global Positioning

- Global Positioning Sensors (GPS) use satellite readings to determine location
- Differential GPS (DGPS) uses a base station to eliminate the intentional error in the satellite signals (not used much these days)
Ranging Sensors: Sonar

- Sends an acoustic pulse and listens for echo
- Can determine how far away an obstacle is based upon time of flight and speed of sound
- Can have problems with specular reflections
Ranging Sensors: Infrared

- Fast and inexpensive proximity sensors
- Emit an infrared pulse and detect the reflected signal
- Infrared can be washed out by bright light; unusable outdoors in more cases
Ranging Sensors: Laser Ranging

- Uses one of the following methods
  - Triangulation (relationship between outgoing light beam, incoming ray and its position on the film plane)
  - Time-of-flight (similar to sonar or infrared)
  - Phase-based (based on difference between phase of emitted and reflected signals)
- Low-power ranges a few meters
- High-power can range over a kilometer
- Many robot systems now using Kinect instead
Sensors: Cameras

- Images from cameras must be processed in order for the robot to use the information
  - Edge detection
  - Tracking movement
  - Finding particular colors
  - Distance of objects from the camera
- We’ll talk more about vision later in the term
Problems with Sensors

- Noisy
- Return an incomplete description of the environment
- Can not be modeled completely

- We will come back to these issues throughout the term, particularly when we talk about mapping
Constructing a Sensor Suite

Consider the following for each sensor:

1. Speed of operation
2. Cost
3. Error rate
4. Robustness
5. Computational requirements
6. Power, weight and size requirements
Attributes of a Sensor Suite

Consider for entire suite:

1. Simplicity
2. Modularity
3. Redundancy
   - A sensor might be noisy or inaccurate by itself
   - False positives / False negatives
   - Complementary sensors can be used to provide disjoint information about a percept
Fusing Sensor Data: Sensor Fission

sensor → percept → behavior → action

sensor → percept → behavior → action

sensor → percept → behavior → action

combination mechanism → action
Fusing Sensor Data:
Action-Oriented Sensor Fusion

sensor

percept

sensor

percept

fusion

percept

behavior

action
DC Motors
Motor ports 0 (left motor) and 3 (right motor)
Plugging in DC Drive Motors

- DC drive motors are the ones with two prong plugs
- The CBC has 4 drive motor ports numbered 0 & 1 on the left and 2 & 3 on the right
- When a port is powered, it has a light that glows blue for one direction and red for the other
- Plug orientation order determines motor direction, but by convention, blue is forward and red reverse
CBC Motor Test Screen
CBC Motor Commands

mav(), ao(), off()

• **mav** (*motor#*, *vel*); [mav=move at velocity]
  – *motor#* is the motor port (0-3) being used
  – *vel* is the rotational speed of the motor measured in ticks per second (-1000 to 1000)
  – the amount of rotation per tick depends on the kind of motor
  – the motor runs at the set speed until turned off or commanded otherwise

• **ao();** turns off all motor ports

• **off** (*motor#*); turns off the specified motor port
Motor Position Counter

- As a DC motor runs, the CBC keeps track of its current position in ticks
  - `get_motor_position_counter(motor#)`;
    is a library function that returns this value for `motor#`
  - `clear_motor_position_counter(motor#)`;
    is a library function that resets the `motor#` counter to 0
  - You can see the current value of the counter for a motor on the `motors..test` and `Sensor Ports` screens
CBC Sensors
CBC Sensor Ports
Analog and Digital

analog ports (0-7) and digital ports (8-15)

Sensor plug orientation
CBC Sensor Ports Screen

<table>
<thead>
<tr>
<th>Analogs</th>
<th>Digital</th>
<th>Accelerometer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0: 1022</td>
<td>8: 0</td>
<td>X: -9</td>
</tr>
<tr>
<td>1: 1021</td>
<td>9: 0</td>
<td>Y: -7</td>
</tr>
<tr>
<td>2: 1022</td>
<td>10: 0</td>
<td>Z: 58</td>
</tr>
<tr>
<td>3: 1022</td>
<td>11: 0</td>
<td>Battery</td>
</tr>
<tr>
<td>4: 1022</td>
<td>12: 0</td>
<td>V: 7.673</td>
</tr>
<tr>
<td>5: 1023</td>
<td>13: 0</td>
<td></td>
</tr>
<tr>
<td>6: 1022</td>
<td>14: 0</td>
<td></td>
</tr>
<tr>
<td>7: 1022</td>
<td>15: 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Motor</th>
<th>Power</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-17</td>
</tr>
</tbody>
</table>
CBC Sensor Ports Screen

• Go to the Sensor Ports screen
  – The Sensor Ports tab is Under the Sensors/Motors tab on the opening screen
• Plug the two sensors to be used for this activity into analog ports on the CBC
  – Plug the light sensor into the same port you chose in Activity B-a2
  – Plug the reflectance sensor into any other available analog port
• When you point the reflectance sensor towards the IR light sensor you should see a low value for its port
• If you aim the reflectance sensor at the table and move it across the table edge its value will change
digital()

- For a digital sensor such as a button sensor plugged into digital port 8, `digital(8);` will provide the current value of the sensor
  - A digital sensor’s value is 0 if off and ≠0 if on
IR Reflectance Sensors

• An IR reflectance sensor has an emitter producing an IR beam and an IR light sensor that measures the amount of IR reflected when the beam is directed at a surface

• There are two reflectance sensors available to you for the CBC

• The CBC library function `analog10` returns an amount that measures the amount of light reflected as a value between 0 and 1023

• A dark spot reflects less IR, producing a higher reading
IR Reflectance Sensor Behavior

Amount of reflected IR depends on surface texture, color, and distance to surface (higher values mean less IR indicating a dark surface or a drop off)
Sample code

• Plug a reflectance sensor in port 0 and a light sensor in port 2

```c
int main() {
    while(a_button()==0) {
        printf("reflectance: %d ", analog10(0));
        printf("light: %d\n", analog10(2));
        sleep(1);
    }
}
```
Analog and Floating Analog Sensors

- For an analog (e.g., light sensor or reflectance sensor) or a floating analog sensor (e.g., ET sensor or sonar) plugged into analog port 3, `analog10(3)`; will produce the current value of the sensor.
- An analog sensor’s value will be in the range from 0 to 1023.
- The CBC can change each analog port to be either analog (pull up resistor) or floating analog (no pull up).
  - `set_each_analog_state(0,0,0,1,0,0,0,0,0)`; sets port 3 to analog float and the rest analog.
Optical Rangefinder “ET”

- Floating analog sensor
- Connect to ports 0-7
- Access with library function `analog10(port#)`
  - You can also use `analog(port#)` for lower resolution
- Low values (0) indicate large distance
- High values indicate distance approaching ~4 inches
- Range is 4-30 inches. Result is approximately $1/d^2$. Objects closer than 4 inches will produce values indistinguishable from objects farther away
Optical Rangefinder

Position Sensing Device (PSD)

Lens

Focused IR Beam

(high value)

(low value)

(low value)
Ultrasonic Rangefinder (Sonar)

- Timed floating analog sensor.
- Connect to ports 0-7
- Access with function `analog(port#)`
  - Assumes `port#` has been set to floating!
- Returned value is distance in inches (not fractions of inches) to closest object in field of view
- Range is approximately 10-250 inches
- **Important:** When first powered up, make sure there is nothing within 24 inches (60cm) so sonar can self calibrate.
Ultrasonic Sensors

- Puts out a short burst of high frequency sound
- Listens for the echo
- Speed of sound is ~300mm/ms
- \textbf{analog}(\textit{port#}) measures how long it takes to hear the echo, divides it by two, and then multiplies it by the speed of sound
- The sonar field of view is an approximately 30° cone
- We’ll use this sensor in Lab 5
Sample code

- Plug an ET sensor in port 1 and a sonar sensor in port 3

```c
int main() {
    set_each_analog_state(0,1,0,1,0,0,0,0);
    while(a_button()==0) {
        printf("ET: %d ",analog10(1));
        printf("Sonar: %d inches\n",analog(3));
        sleep(1);
    }
}
```
Sensor and Motor Manual

• For further detail about sensors, consult the Sensor and Motor Manual on the course web site (also distributed in last week’s lab)
Lab 2 (Thursday)

• Backing up and turning after hitting an obstacle with the bumper (touch sensors)
• Line following (IR reflectance sensors)

• Remember that your robot car needs to be ready to go at the start of the lab period!