Robotic Agents (CMPSC 311)

Calibration

Janyl Jumadinova

September 10, 2019
Why Calibrate?

- Sensors are imperfect:
  - two sensors of the same type from the same manufacturer may produce slightly different outputs.

- The sensor is just one component in the measurement system.

- Sensors may respond differently in similar conditions due to sensor design.

- Heat, cold, humidity, etc. can affect sensors.

- Sensors can age with time.

![Graph showing sensor output vs. measured parameter]
Sensor Characteristics to Look For

- **Precision**: producing the same output for the same input.
- **Resolution**: detecting small changes in the measured parameter.
- Sensors can be calibrated for accuracy if they give consistent measurements with good resolution.

![Graph showing precision and resolution](image-url)
Sensor Calibration

Calibration
Teaching the sensor what the correct value is.

When you use EV3 Color Sensor in Light Sensor Mode (e.g., reflected light mode), you should calibrate it. You need to “teach” the sensor what is “Black” and what is “White”. White reads as 100 and Black reads as 0. Perform calibration whenever light conditions change.
Sensor Calibration

Calibration
Teaching the sensor what the correct value is.

- When you use EV3 Color Sensor in Light Sensor Mode (e.g., reflected light mode), you should calibrate it.
- You need to “teach” the sensor what is “Black” and what is “White”.
- White reads as 100 and Black reads as 0.
- Perform calibration whenever light conditions change.
A program that calibrates EV3 Color Sensors for black and white.

- Reset the existing calibration values.
- Place the robot on “black”.
- Read the data from the robot and calculate adjustment if needed for black.
- Repeat above steps for calibrating “white”.
Calibration

- Other sensor calibration steps:
  - Record the true value
  - Record the sensor value
  - Adjust the sensor value based on the true value
Sensor Errors

- **Offset error**: sensor always reports a value that is a fixed amount of the expected value (common in gyros, light sensors, accelerometers).
  - Can use LinearCalibrationFilter in leJOS to measure and correct this error.

- **Scale error**: under or over-rates the changes in the expected value.
  - Can use LinearCalibrationFilter in leJOS to measure and correct this error.

- **Sensor noise**: gives different value each time (chaotic).
  - Cannot correct for it, but can decrease noise by using averages.
Sensor Errors

- **Offset error**: sensor always reports a value that is a fixed amount of the expected value (common in gyros, light sensors, accelerometers).
  - Can use LinearCalibrationFilter in leJOS to measure and correct this error.

- **Scale error**: under or over-rates the changes in the expected value.
  - Can use LinearCalibrationFilter in leJOS to measure and correct this error.

- **Sensor noise**: gives different value each time (chaotic).
  - Cannot correct for it, but can decrease noise by using averages.
Calibration in leJOS

- leJOS supports sensor calibrating using filters

- LinearCalibrationFilter is used to calibrate sensors for offset and scale errors
  - uses linear interpolation.
  - has two modes of operation:
    - operational mode: corrects samples coming from the sensor;
    - calibration mode: calculates calibration parameters.
Calibrating non-sensors

Robotic measurements (e.g., wheels)

- Measure the wheel diameter and enter it into the program
- Run the program and check if the robot traveled the actual distance (e.g., \texttt{travel(100)})
- Adjust the diameter value (e.g., if distance traveled shorter than the actual distance, decrease the diameter)
Calibrating non-sensors

Robotic measurements (e.g., wheels)

- Next, measure the track width (distance between the center of the right and left wheels) and enter it into the program.
- Run the program and check if the robot rotated the actual distance (e.g., rotate(360))
- Adjust the track width (e.g., if over-rotated, decrease the track width)