LabVIEW Part 2: Motor Control

October 14, 2008

www.robojackets.org

Goals for Today

- Brief review
- Open Loop Motor Control
- Feedback (encoders)
- Closed Loop Motor Control

RoboJackets

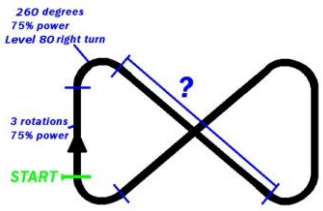
Open Loop Motor Control

- Check out
  - **NXT Toolkit>> NXT Library>> Output**
- Demo of one block (rely on help)
- Each group pick a block
  - Try it out
  - Show the class
- Review output blocks

RoboJackets

Open Loop Motor Control

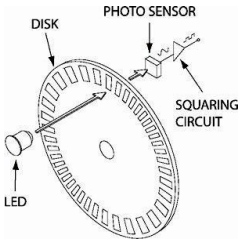
Activity!



- Teams judged on how close they get to their starting position and orientation
- Must complete 3 laps
- Coast between blocks

RoboJackets

Feedback (Encoders)



RoboJackets

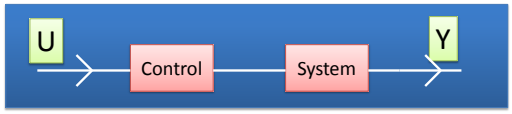
Feedback (Encoders)

- **Demo:** numerical output
- **Guided activity:** etch-a-sketch

RoboJackets

Control

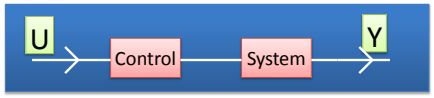
- What is a control system?



RoboJackets

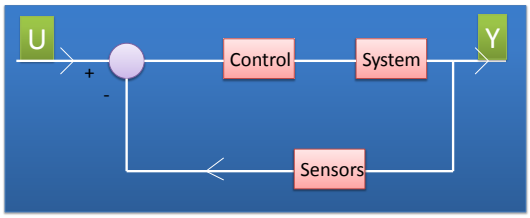
Control - Thermostat

- Input (U) = Wanted Temperature (User)
- System = AC + Room
- Control = Microcontroller to turn AC on or off
- Output (Y) = Room Temperature



RoboJackets

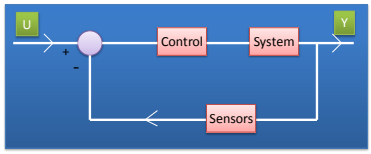
Closed Loop Control



RoboJackets

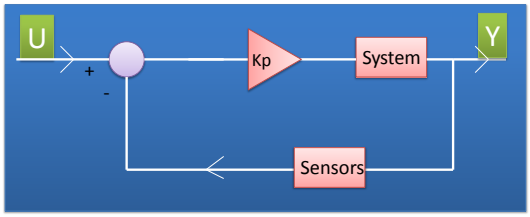
Closed Loop Control - Thermostat

- Input (U) = Desired Temperature (User)
- System = Room + AC
- Sensors = Digital Thermometer
- Control = Microcontroller to turn AC on or off
- Output (Y) = Room Temperature



RoboJackets

Proportional Controller

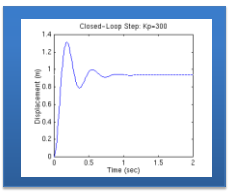


RoboJackets

Proportional Controller

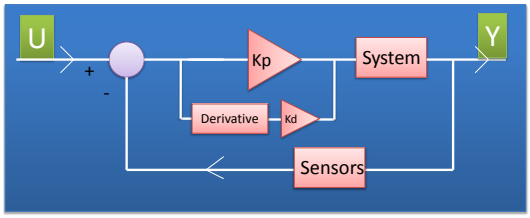
- To handle the present, the error is multiplied by a proportional constant Kp, and sent to the output.

Error = Desired Value – Actual Value
Output = Error * Kp



RoboJackets

Proportional – Derivative Controller

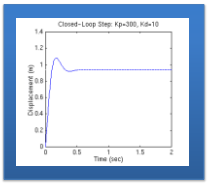


RoboJackets

Proportional – Derivative Controller

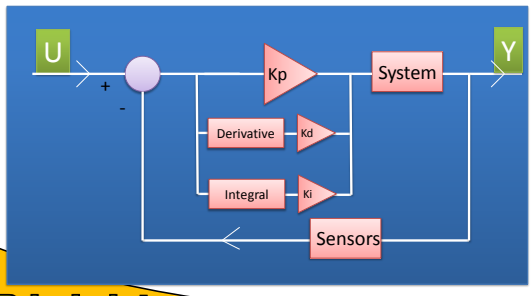
- To handle the future, the first derivative of the error (its rate of change) is calculated with respect to time, and multiplied by the constant Kd, and added to the proportional term.

Output = (Error * Kp) + ((Change in Error / Time) * Kd)



RoboJackets

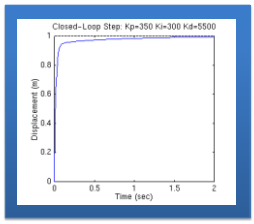
Proportional – Integral - Derivative Controller



RoboJackets

Proportional – Integral - Derivative Controller

Output = (Error * P) + (Sum of the Error * I) + ((Change in Error / Time) * D)



RoboJackets

Proportional – Integral - Derivative Controller

- **Tuning suggestion:**
  - **Start with just P control** (I = D = 0) until the system starts to oscillate, meaning it reaches the target, overshoots, reaches the target, undershoots and repeats this process.
  - **Increase I** until this oscillation stops; the control should be smoother now, but may be slow.
  - **Then increase D** until the system reaches its target at an acceptable speed (depending on the circumstances, overshoot may or may not be desirable).

RoboJackets

Proportional – Derivative Controller

Demo/Activity

RoboJackets