Research- 9/29

Summary of Previous Week

I spent the last week reading more about controls and control loop algorithms. A control loop is basically a system that regulates the behavior of another system. Often, this is done through the use of a feedback mechanism. Two control loop algorithms that I studied were the PID Control Algorithm and the Bang Bang Control Algorithm.

The Bang Bang control algorithm is simple because it essentially only oscillates between two states. The feedback mechanism feeds to the controller, which then either turns state 1 on or state 2 on depending on the feedback values.

The PID control algorithm is slightly more complicated. It gives you the option of implement proportional (where the control variable is set in proportion to the difference between the measured and desired outcomes), derivative (which considers the rate the measured outcome is changing), and the integral (which uses the average outcome in the past to detect whether the outcome is too low or too high and sets the control proportional to past errors) controls.

For simplicity, I chose to implement the Bang Bang algorithm. I implemented a simple temperature mechanism that either turns the heat on or off depending on the current temperature.

Goals for Next Week

For this coming week, I want to implement a control algorithm that better mimics a computer system.

9/29 Agenda

* Review various control algorithms
* Review Bang Bang implementation
* Discuss next implementation steps

Bang Bang Implementation

import java.util.\*;

import java.lang.\*;

import java.io.\*;

public class BangBangController

{

public static final int setpoint = 70;

public static boolean boiler\_state = false;

public static void main (String[] args) throws java.lang.Exception

{

// start with any temperature

changeBoiler(65);

}

public static void changeBoiler(int current\_temp){

if(current\_temp == setpoint){

System.out.println("Perfect temperature of 70!");

/\*System.out.println("equal");

if(boiler\_state == true){

int new\_temp = adjust\_temp(true, current\_temp);

System.out.println("true");

changeBoiler(new\_temp);

}

else if(boiler\_state == false){

int new\_temp = adjust\_temp(false, current\_temp);

changeBoiler(new\_temp);

}\*/

}else{

if( (current\_temp < setpoint) && (current\_temp > 60) ){

if(boiler\_state == true){

System.out.println("current temperature: " + current\_temp);

int moretemp = current\_temp + 1;

changeBoiler(moretemp);

}else{

System.out.println("current temperature: " + current\_temp);

int lesstemp = current\_temp -1;

changeBoiler(lesstemp);

}

}else if(current\_temp == 60){

System.out.println("reached minimum of 60");

boiler\_state = true;

int new\_temp = adjust\_temp(true, current\_temp);

changeBoiler(new\_temp);

}else if( (current\_temp > setpoint) && (current\_temp < 80)){

if(boiler\_state == true){

System.out.println("current temperature: " + current\_temp);

int moretemp = current\_temp + 1;

changeBoiler(moretemp);

}else{

System.out.println("current temperature " + current\_temp);

int lesstemp = current\_temp-1;

changeBoiler(lesstemp);

}

}else if(current\_temp == 80){

System.out.println("reached maximum of 80");

boiler\_state = false;

int new\_temp = adjust\_temp(false, current\_temp);

changeBoiler(new\_temp);

}

}

}

public static int adjust\_temp(boolean up\_down, int current\_temp){

int temp = current\_temp;

if(up\_down == true){

temp++;

}

else if(up\_down == false){

temp--;

}

return temp;

}

}