Lab 1
Controls

Amanda Sutherland
February 18, 2015

System Construction

I built my system on a breadboard, as in figure 1 below, and for the sake of speed and simplicity I did not take the time to make the wires beautiful. Apologies if it is messy! For the connection between the thermistor and the transistor I kept them from shorting by wrapping each component in electrical tape. I then shrink-wrapped them together to hold them together well.

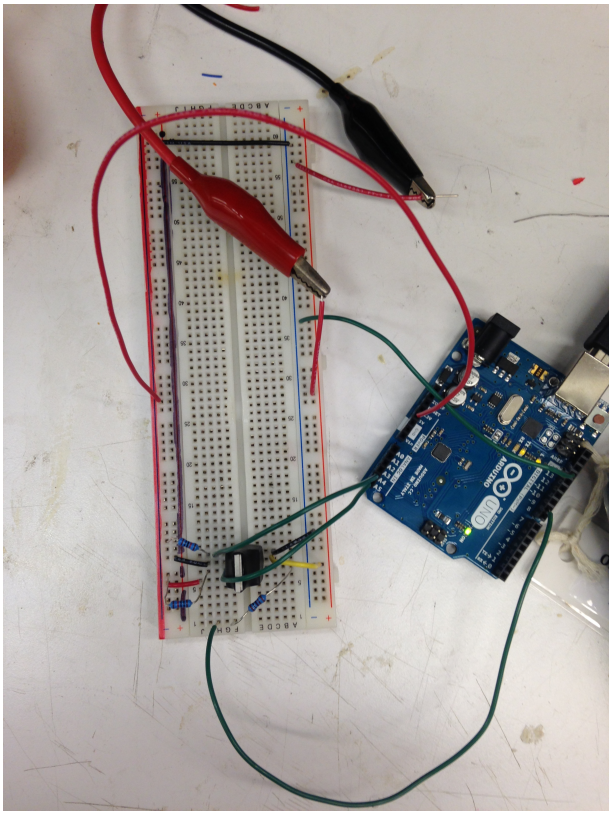

Figure 1: My system.

Feedback loop

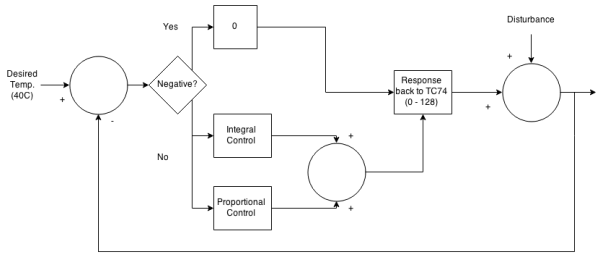

Figure 2: System Diagram. The possible output to the TC74 is limited to 128 because after testing my system with the 1 k resistor that was what the system could handle without railing the power supply.

System Schematic

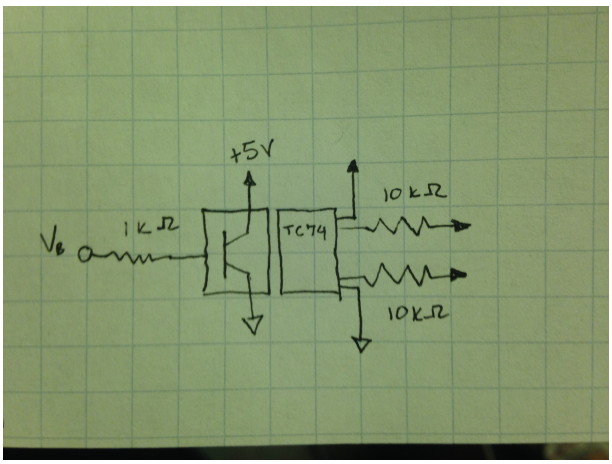

Figure 3: System Diagram

Description of Control Law

I started off intending to use proportional and integral control. This system would have used proportional control to find the error between the intended temperature and the actual temperature, then take that error away from the actual temperature to reach the intended 40 degrees. This worked reasonably well but stayed at a constant amount from the intended temperature, so I then

added integral control to reduce this error and get the temperature closer to the intended value.

This worked reasonably, but I realized that through the way I was implementing this I had essentially created a hysteresis control system; I had added limits to the max temperature possible which prevented the integral control from having much of an effect. Thus, I had a system such that I raised the temperature until it reached 40, then turned it off until it came below that value again. Up until 40 degrees it used proportional and integral control. Although simple, this worked very well.

System Response

To fully test my system. I subjected it to four distinct tests.
1. Leaving it at room temperature and having it go to goal.
2. Heating it with a hair-drier, leaving hair-drier on for 10 seconds after data collection starts, then letting it go to goal.
3. Blew on it twice for ten seconds each, once at 30 and once at 2.
4. Touched it with ice twice , once at 30 for 10 secs, once at 2 minutes for 20 seconds.

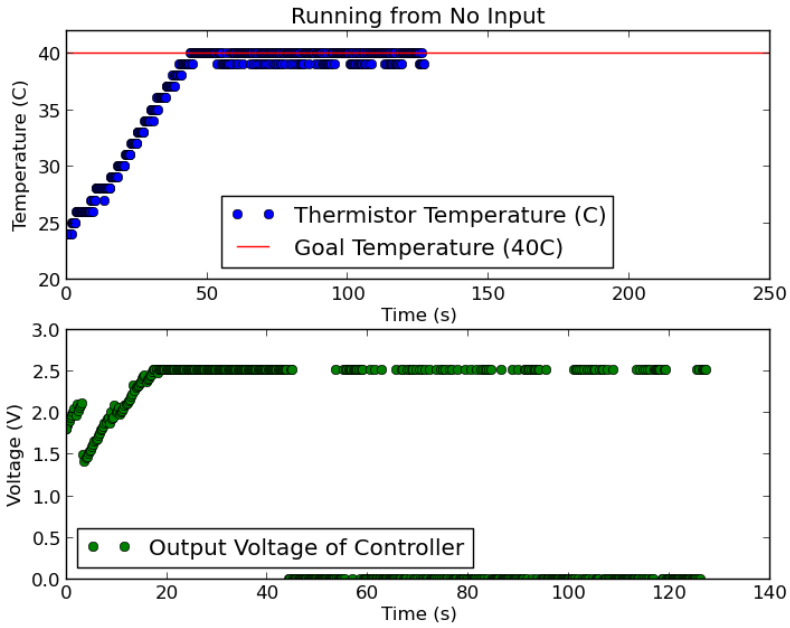

Figure 4: Here I simply recorded data starting when the system was at room temperature. As expected, the temperature rises fairly quickly (40 seconds), and stays between 40 and 39 after that point. The output voltage of the controller shows the voltage occasionally going to zero from 128, which shows the temperature had gone to 40 and, as per my control loop, had been set to zero.

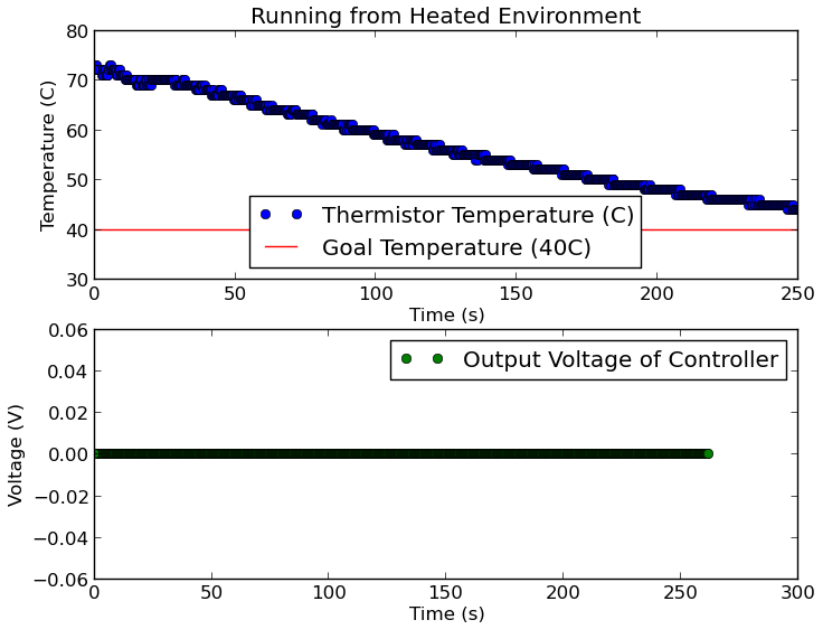

Figure 5: I heated the system with a hair drier before recording data, then kept applying heat for 10 seconds after beginning recording. As the plot shows, the temperature was slowly decreasing and thus the output voltage was zero, as my control loop was constantly in the "no response necessary" branch. My setup is show in the figure below.

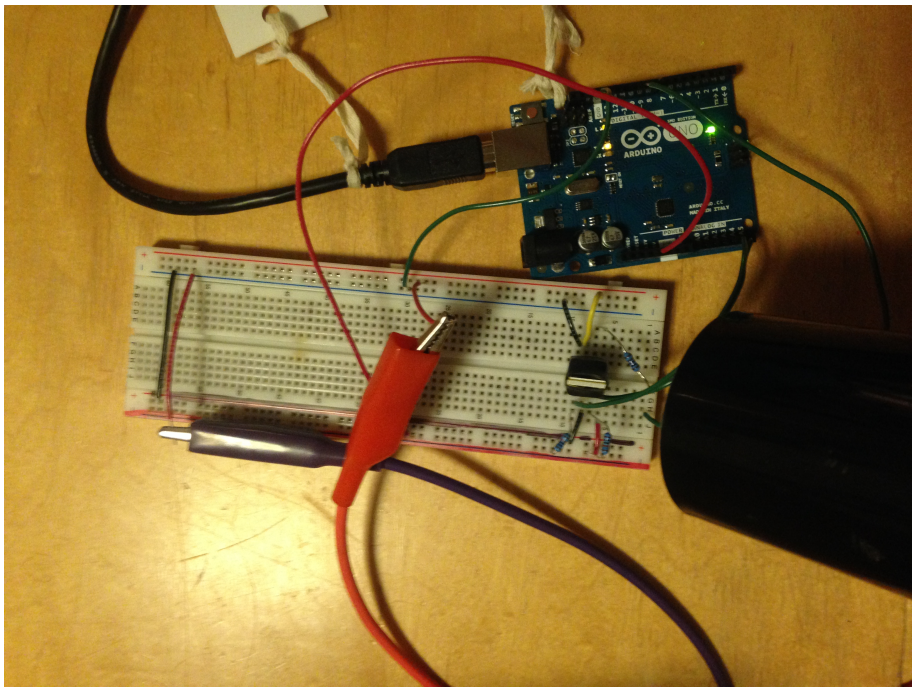

Figure 6: I applied a hair drier to my circuit to heat it up. It raised it to 70 degrees!

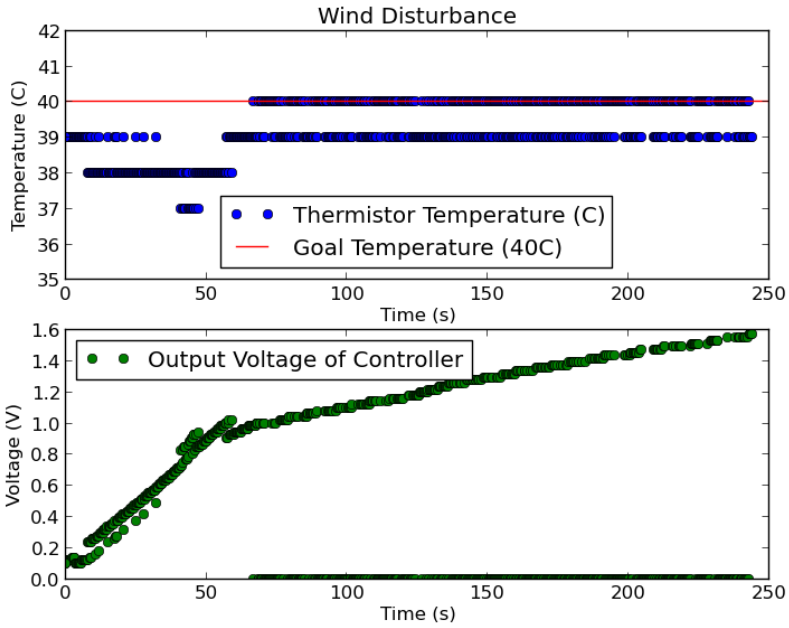

Figure 7: Here the circuit took time to reach it's optimal voltage, and once it did it was not affected by me blowing on it. It showed the effects of wind at 30 seconds, but not at 2 minutes. I question my data for the two minute marker, as it should have affected it equally.

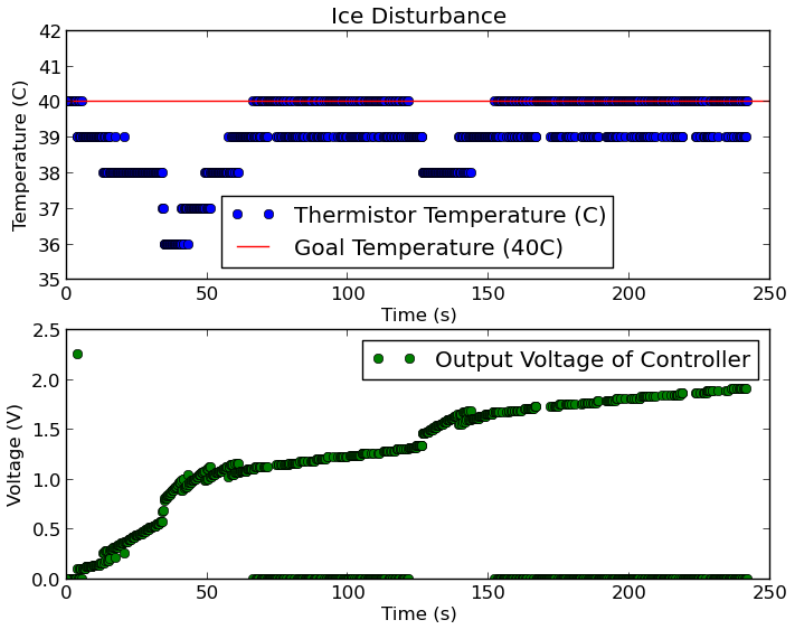

Figure 8: The ice was applied at 30 seconds for 10 seconds, and at 120 seconds for 20 seconds. The system responds fairly quickly, although for the first ice application it took longer and I do not know why this would make sense. I can only assume I applied the ice for longer than I intended somehow, as the rest of the system responded as I expected. My method of testing is shown below.

Figure 9: I used our lovely weather conditions to their best advantage and applied snow to the circuit. Everything was nice and chilly, and since it was snow I could get it to surround the transistor/thermister pairing.

Works Cited

To understand and control the I^2C buss I extensively used a tutorial found at http://exploringarduino.com/content/ch8/. Jeremy Blum gives the arduino code for working with the TC74 and gives circuitry for it as well.

Appendix

All of my code can be found on my github at
https://github.com/AmandaSutherland/Controls$_{O}linSpring$2015.

Here is the Arduino Code used for my control loop.

```
//include wire library
#include <Wire.h>

// These constants won't change.  They're used to give names
```

```
// to the pins used:
const int analogOutPin = 9; // Analog output pin that the LED is attached to
int outputValue;
//set the address of the temp sensor
int temp_address = 72;
int kP = 5 ;
float kI = 0.1;
int P;
int I ;
long IntegralSum ;

void setup()
{
  Serial.begin(9600);
  Wire.begin();
}

void loop()
{
    Wire.beginTransmission(temp_address);  //start talking
    Wire.write((uint8_t) 0);
//Ask for register zero
    Wire.endTransmission();                          //Complete transmission
    Wire.requestFrom(temp_address, 1);     //request 1 byte
    while(Wire.available() == 0);                     //wait for response
    int c = Wire.read();
//get the temp

    // proportiona8l control/
    int Error = 40 - c;

    IntegralSum += Error;
    P = Error * kP;
    I = IntegralSum * kI;
    outputValue = P+I;
    if (c >= 40){  // don't heat transistor if temp is above 40
      outputValue = 0 ;
    }
    else if (outputValue > 128 ){
      outputValue = 128;
    }

     // change the analog out value:
    analogWrite(analogOutPin, outputValue);

    //convert from celcius to farenheight
```

```
    int f = round(c*(9.0/5.0) +32.0);

    //print the results
    Serial.print(c);
//    Serial.print(" C, ");
//    Serial.print(f);
//    Serial.print(" F, ");
    Serial.print(",");
    Serial.println(outputValue);
//    Serial.println(" outputValue");

    //delay, then do it again
    delay(250);

}
```