Verina Abdelmesih

4/12/2023

Powder Coating Oven

Control System Analysis

Section 1

Introduction

The goal of this project is to create a powder coating oven. The powder coating oven will be used to coat things for the rocket club and the Baja club. This analysis will discuss the control system for the powder coating oven. In industry control systems are typically used to operate heating devices, fans/blowers, vents, lights, thermocouples, and emergency buttons. This control system will attempt to operate all of the listed devices using a PID (proportional integral derivative) controller. PID is generally preferred over other systems such as Arduino because it combines the average of each type of control while including quick response times. This analysis will go over the semantics of the PID controller along with the potential set up that will be used.

Method

PID is also known as proportional integral derivative control system or a three-term controller. PIDs consist of a control loop mechanism that employs feedback along with other applications required for continuous modulated control. Figure 1 below shows a block diagram of the PID controller feedback loop.

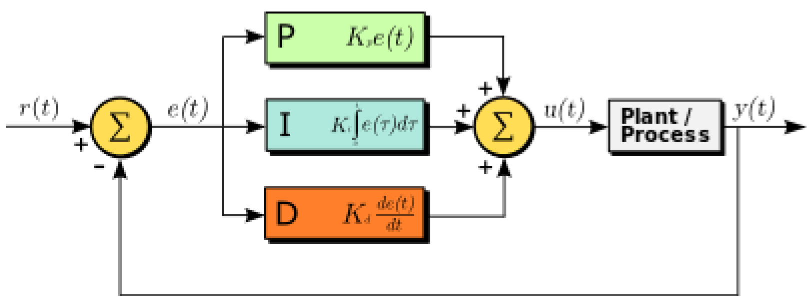

Figure 1: PID Block Diagram [1]

Figure 1 shows the how the controller uses three control terms of proportional, integral, and derivative to provide an accurate output. The generic equation used is

$$SP = r(t) \qquad \text{Equation 1}$$

Where SP is the measured value and r(t) is the desired value. In order to calculate the error value (e(t)) equation 2 is used.

$$e(t) = r(t) - y(t) \qquad \text{Equation 2}$$

Where y(t) is the process variable. Equation two will calculate the continuous error values and apply corrections based upon the PID terms. The equation for the overall control function is written as

$$C = K_p + \frac{K_i}{s} + \frac{K_d}{T_f s+1} \quad \text{Equation 3}$$

Where C is continuous time, K_p is proportional gain, K_i is integral gain, K_d is derivative gain, s is the setpoint or input value, and T_f is first order derivative filter time constant. Although PID systems can use both continuous and discrete time, continuous time is required for this set up due to the thermocouples continuously sending input values to the PID system. In order to see if the PID system would best fit the oven a model of the system was created in Matlab. Equation 3 was used to create the Matlab were s was changed to B and K which are inputs 1 and 2 that are provided by the thermocouple. Figure 3 below shows the Matlab value inputs required for the PID system.

```
tic                  % Timer Function:start timer to calculate CPU time

%Inputs
desired = 350;       % desired temperature (degrees Fahrenheit)
feed1 = 1;           % can be replaced with damping coefficient B or (B/Mass)
feed2 = 1;           % can be replaced with spring coefficient K or (K/Mass)
B = feed1;
K = feed2;
Kp = 1;               % proportional term Kp
Ki = 0.01;            % Integral term Ki
Kd = 0.01;            % derivative term Kd
dt = 0.01;            % sampling time
Time = 10;            % total simulation time in seconds
n = round(Time/dt);   % number of samples
```

Figure 3: Matlab Input Values [2]

As shown in Figure 3 the inputs are required in order to run the PID system. The input labeled as desired represents the setpoint value or in the case of the oven the temperature that the oven needs to reach.

The Matlab begins by utilizing the tic function which is a timer function that calculates the amount of time the function takes to reach the desired temperature. The code then inputs the listed values shown in Figure 3 required for the PID equation. The code then continues to assign arrays to better optimize the processing time of the PID. The arrays are shown in Figure 4 below.

```
% pre-assign all the arrays to optimize simulation time
Prop(1:n+1) = 0;
Der(1:n+1) = 0;
Int(1:n+1) = 0;
I(1:n+1) = 0;
PID(1:n+1) = 0;
FeedBack(1:n+1) = 0;
Output(1:n+1) = 0;
Error(1:n+1) = 0;
state1(1:n+1) = 0;
STATE1(1:n+1) = 0;
state2(1:n+1) = 0;
STATE2(1:n+1) = 0;
```

Figure 4: Pre-assigned Arrays for Optimization [2]

After pre-assigning the arrays, a for loop is used to find the error entering the PID controller system. The for loop is used to find the error of the proportional term, the derivative of the error, the integration of the error, and the sum of the integration of the error. These values are then used to calculate the three PID terms. Moving forward these values are used to calculate the sum of the first integration and the output after the first integrator. The first integrator value is used to calculate the second integration and the loop will continue until the output value reaches the desired value. Figure 5 below shows the for loop that is described above.

```
for i = 1:n
    Error(i+1) = desired - FeedBack(i); % error entering the PID controller

    Prop(i+1) = Error(i+1);% error of proportional term
    Der(i+1)  = (Error(i+1) - Error(i))/dt; % derivative of the error
    Int(i+1)  = (Error(i+1) + Error(i))*dt/2; % integration of the error
    I(i+1)    = sum(Int); % the sum of the integration of the error

    PID(i+1)  = Kp*Prop(i) + Ki*I(i+1)+ Kd*Der(i); % the three PID terms

    STATE1(i+1) = sum(PID); % sum PID term to calculate the first integration
    state2(i+1) = (STATE1(i+1) + STATE1(i))*dt/2; % output after the first integrator
    STATE2(i+1) = sum(state2); % sum output of first integrator to calculate the second integration
    Output(i+1) = (STATE2(i+1) + STATE2(i))*dt/2; % output of the system after the second integrator
    FeedBack(i+1) = state2(i+1)*feed1 + Output(i+1)*feed2;
end

tsim = toc % simulation time

% plot results
T = 0:dt:Time;
Reference = desired*ones(1,i+1);
plot(T,Reference,'r',T,Output,'b')
xlabel('Time (sec)')
ylabel('Temperature (F)')
title('PID Simulation')
legend('Desired','Simulated')
```

Figure 5: For Loop and Plot Code [2]

Results

For the simulation the desired output was set to 350 degrees Fahrenheit. Figure 6 below shows the results of the 350 degrees Fahrenheit output.

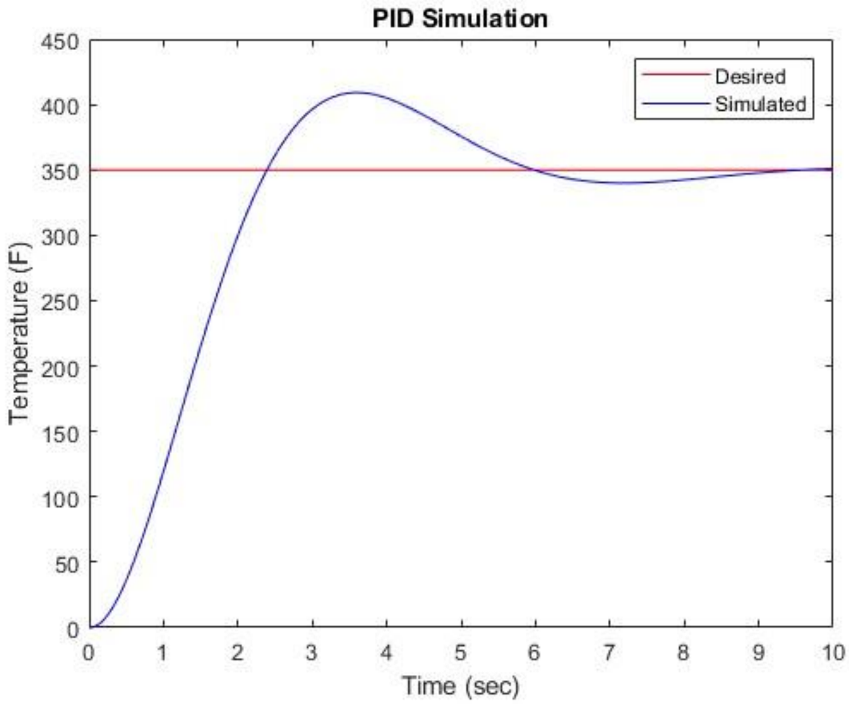

Figure 6: Set point of 350 degrees F

For the simulation shown in Figure 7 below the desired output was set to be 200 degrees Fahrenheit.

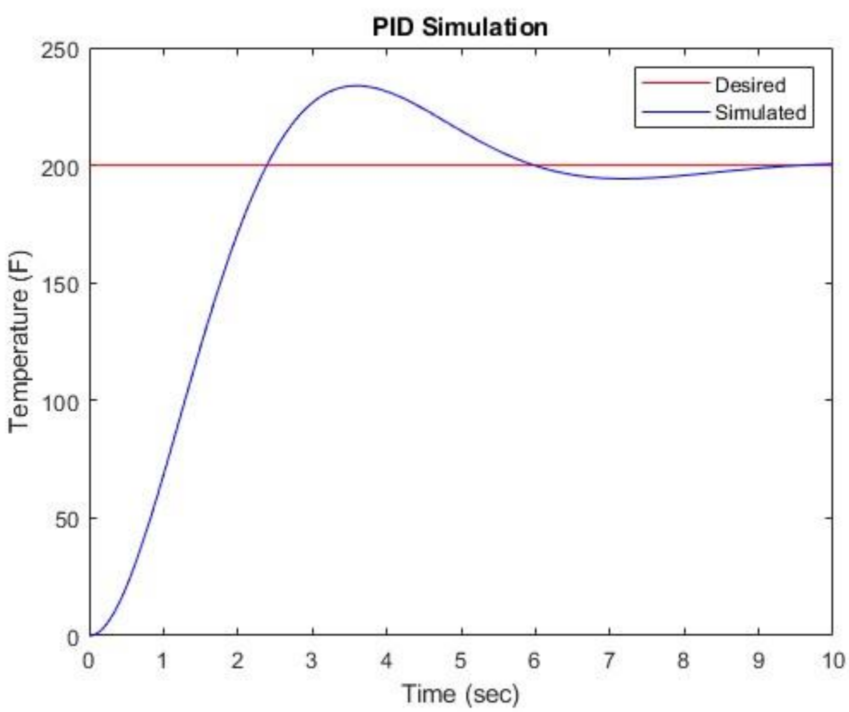

Figure 7: Set point of 200 degrees F

Discussion

Figures 6 and 7 shown above demonstrate how the PID system works in an ideal environment. It shows how the system will continue to read inputs provided by feedback 1 and 2 until the feedback values match the input values. Also shown in Figures 6 and 7 the desired temperature was able to be reached in approximately 10 seconds which will vary when being used for the oven. The PID for the oven will be receiving feedback every 2 to 3 seconds, however the time in which the feedback matches the desired temperature is contingent on the heater being used along with the volume of the oven. The temperature will also be contingent on if the vent system is activated. The vent system will be active if the feedback values are found to be higher than the desired value. The vent will open to allow the oven to cool down to meet the desired value. Once the feedback value either decreases or meets the desired value the vent will close until the feedback value increases and becomes greater than the desired value again.

Conclusion

Given the results shown in Figures 6 and 7 PID is an ideal choice for use as a control system. It is not only used on most ovens made in industry, but it's programming also allows for the system to receive constant feedback while calculating the errors in order to achieve the most accurate readings. The PID system is also capable of controlling multiple devices in one given moment. For this project the PID will be controlling 4 thermocouples, a torpedo heater, vent, 4 lights, and an emergency off switch button.

References

[1] "Pid Controller," *Wikipedia*, 01-Apr-2023. [Online]. Available: https://en.wikipedia.org/wiki/PID_controller. [Accessed: 30-Apr-2023].

[2] "Matlab based PID controller," *MathWorks*. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/35163-matlab-based-pid-controller. [Accessed: 30-Apr-2023].

[3] Libretexts, "9.2: P, I, D, PI, PD, and PID Control," *Engineering LibreTexts*, 11-Mar-2023. [Online]. Available: https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls_(Woolf)/09%3A_Proportional-Integral-Derivative_(PID)_Control/9.02%3A_P%2C_I%2C_D%2C_PI%2C_PD%2C_and_PID_control#:~:text=PID%2Dcontrol%20is%20most%20commonly,additionally%20using%20the%20I%2Dcontrol. [Accessed: 30-Apr-2023].

[4] "Sys," *PID controller in parallel form - MATLAB*. [Online]. Available: https://www.mathworks.com/help/control/ref/pid.html. [Accessed: 30-Apr-2023].