Control Engineering

Experiment 6: Dynamics of thermal systems under P/PI/PID control

GROUP 1B:

Manan Dangwal (2021EEB1186)
Pragya Gupta (2021EEB1192)

Objectives

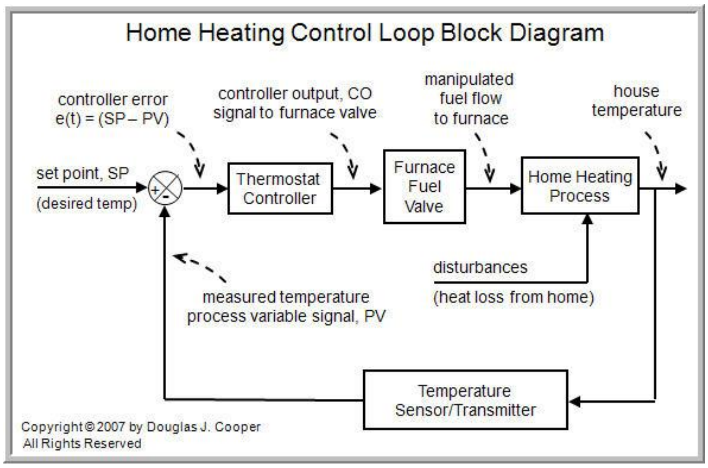
- Calculating the time constant of the first order thermal system.
- To design P, PI and PID controllers for the system using Ziegler-Nichols Tuning Methods.
- Comparison of the controller with the system's autotuned controller response.

Experimental Setup

Thermal Setup: Our thermal system consists of a heater, an RTD *(Resistance Temperature Detector)* sensor for temperature measurement and a fan for cooling.

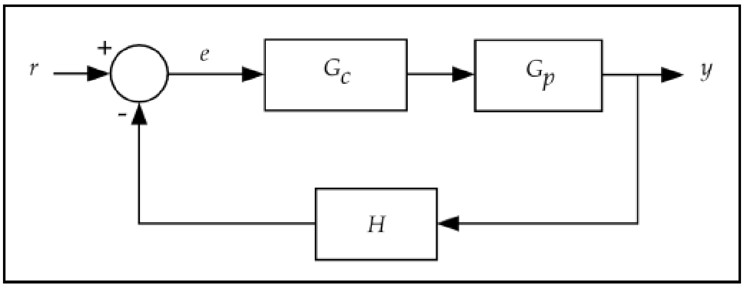
Feedback Controller: The feedback to the system works on the heater, and not the fan. The temperature sensed by the RTD is fed back to the controller. The controller can be configured either manually aur set to auto-tuning.

Software: The Controller is connected to Siemens **SIMATIC V15 SCADA3** via **RS 485**n interface (**MODBUS-RTU** protocol) with the software running in real time on an appropriate host computer.



Home Heating control is an interesting and practical use case where a thermal system controller needs to be employed. Ref: https://controlguru.com/the-components-of-a-control-loop/

Theory



Ref: https://ledin.com/control-systems-basics/

G_c is the controller transfer function, G_p is the plant transfer function and H is the feedback loop gain

The open loop system has first order dynamics. Mathematically,

$$G_{OL} = \frac{K}{\tau s+1}$$

Our system is in a closed loop with a cascaded controller, which gives the closed loop transfer function as follows,

$$G_{CL} = \frac{Kp.K}{\tau s+(1+Kp.K)}$$

Where K_p is the controller transfer function. To perform Ziegler-Nichols Open Loop tuning on this transfer function, we need to mathematically manipulate it into an *open loop* form. Consider,

$$G_{CL}=\frac{\frac{Kp.K}{1+Kp.K}}{s.\left(\frac{\tau}{1+Kp.K}\right)+1}$$

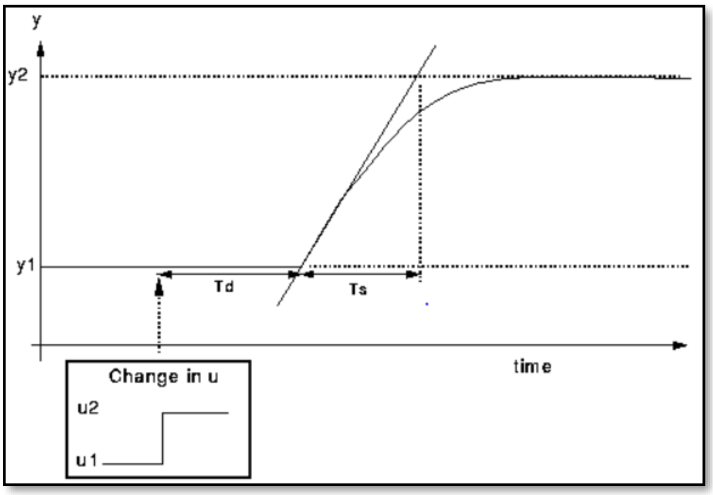
With an effective DC gain $K' = \frac{Kp.K}{1+Kp.K}$ and an effective time constant $\tau' = \frac{\tau}{1+Kp.K}$.

For K = 1 (which implies an open loop system DC gain of 1), the transfer function reduces to

$$G_{CL}=\frac{\frac{Kp}{1+Kp}}{s.\left(\frac{\tau}{1+Kp}\right)+1}$$

Zeigler-Nichols Open Loop Tuning Method

We can now derive the PID expressions using this method. For a step response, the following algorithm is to be followed to get the desired PID parameters.



Draw a tangent at the inflection point of the curve.

1. The intersection with the X axis gives the Lag Time (T_d).
2. The time constant (T_s) can be found by 63% of the maximum value.
3. The SS gain can be found by g_{ss} = (y2-y1)/(x2-x1).

The constants are given as follows: -

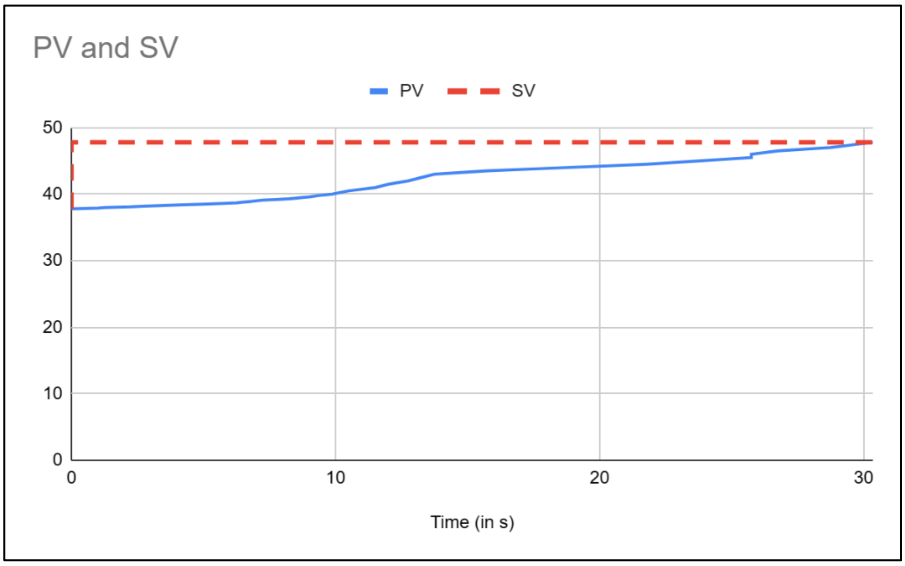
P controller: $K_p = \frac{Ts}{gss.Td}$

PI controller: $K_p = \frac{0.9.Ts}{gss.Td}$ $T_i = \frac{Td}{0.3}$

PID controller: $K_p = \frac{1.2Ts}{gss.Td}$ $T_i = 2T_s$ $T_d = 0.5T_s$

Observations

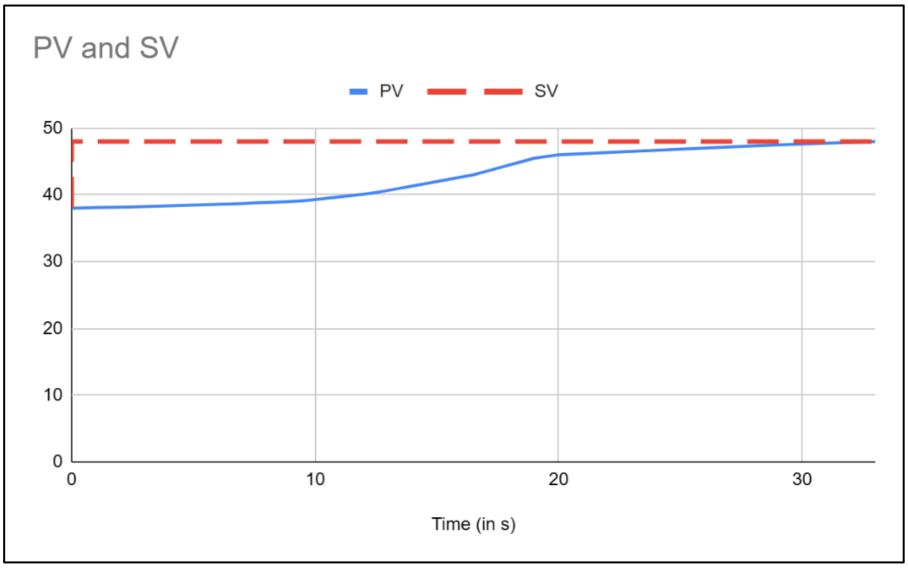
Test 1: K_p = 10, SV = 47.8°C (step size = 10°C)



Studying the plot, we can see the inflection point is at around 13.5s. Drawing a tangent at that point and intersecting it with the X-axis yields a **lag time of 8.55s.**

Time constant = 19.35s

Test 2: K_p = 5, SV = 48°C (step size = 10°C)

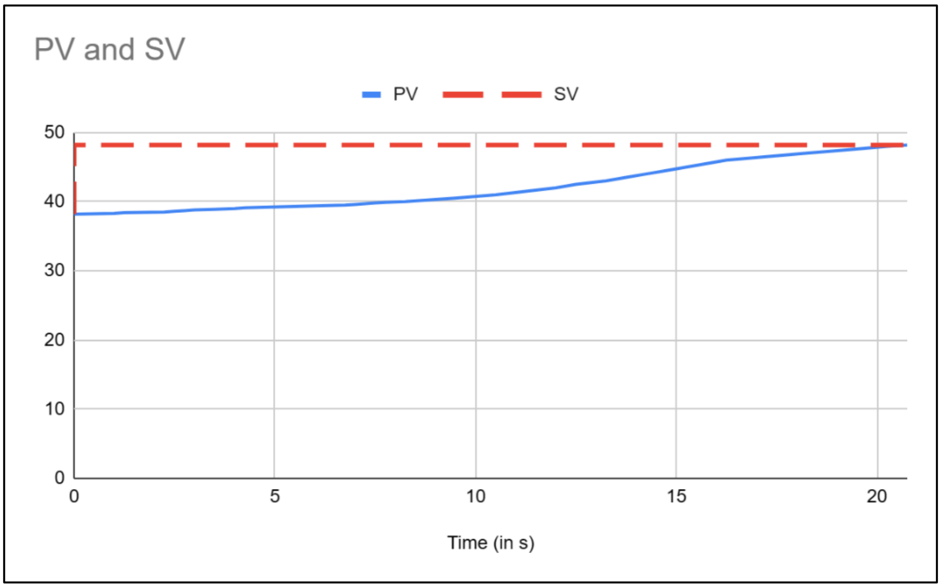


PV refers to the Process Variable, i.e., the actual temperature sensed by the RTD. SV is the Set Variable, which is the reference signal. As the plot shows, SV is a step of magnitude 10° and PV is the corresponding step response of the heater.

Studying the plot, we can see the inflection point is at around 19s. Drawing a tangent at that point and intersecting it with the X-axis yields a **lag time of 4s.**

Time constant = 17.82s

Test 3: K_p = 1, SV = 48.2°C (step size = 10°C)



Studying the plot, we can see the inflection point is at around 15.5s. Drawing a tangent at that point and intersecting it with the X-axis yields a **lag time of 8.45s.**

Time constant = 14.75s

Results

1. The Lag Time (T_d) of the system was calculated to be **7s**.
2. The Time Constants for 3 different K_p (10, 5 and 1) came out to be **17.82s, 19.35s** and **14.75s**. Thus, average time constant is **17.31s**.
3. The steady state gain was **1**.

Thus, for the 3 controllers, we get the following results –

P controller: K_p = 2.47

PI controller: K_p = 2.23, T_i = 23.33

PID controller: K_p = 2.96, T_i = 34.62, T_d = 8.66

Analysis of the controllers

As per the results we can determine the OLTF to be,

$$G_{OL} = \frac{1}{17.31s+1}$$

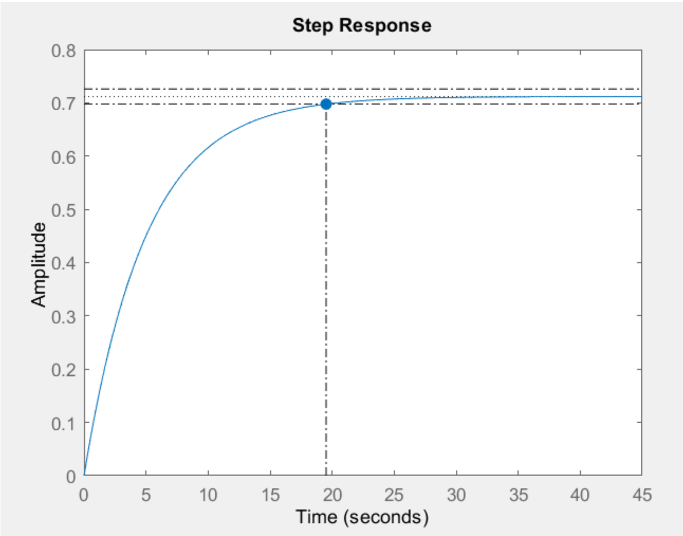
We now use MATLAB to study how the ZN-tuned controllers will work on the system.

P controller

With $K_p = 2.47$, we find G_{CL} as

```
Gcl =

       2.47
  --------------
  17.31 s + 3.47
```



The steady state error was observed to be 0.288 and the settling time was observed as 19.5s. To remove the steady state error, we need an integral component in the controller.

PI controller

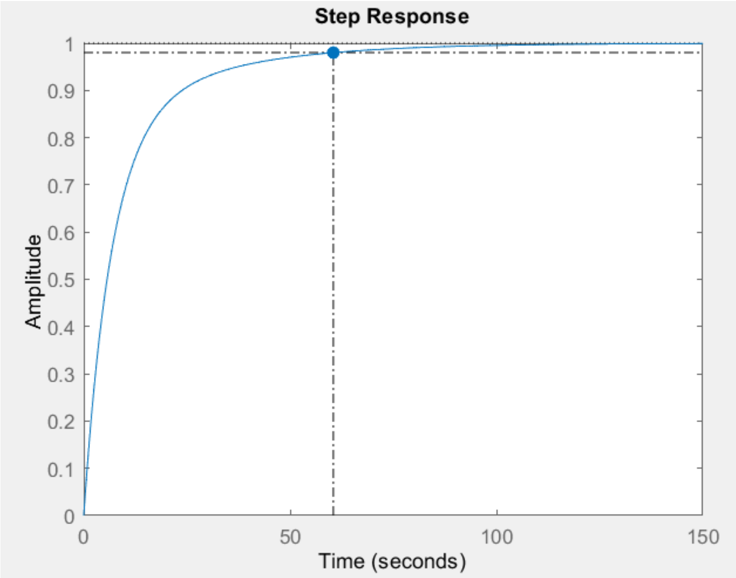
With $K_p = 2.23$ and Ti = 23.33, we find the PI controller transfer function Kpi and G_{CL} as,

```
Kpi =

  52.03 s + 2.23
  --------------
     23.33 s

Gcl =

        52.03 s + 2.23
  --------------------------
  403.8 s^2 + 75.36 s + 2.23
```



The settling time was increased to 60.5s, however the steady state error was driven down to 0. Interestingly, no peak overshoot was observed.

PID controller

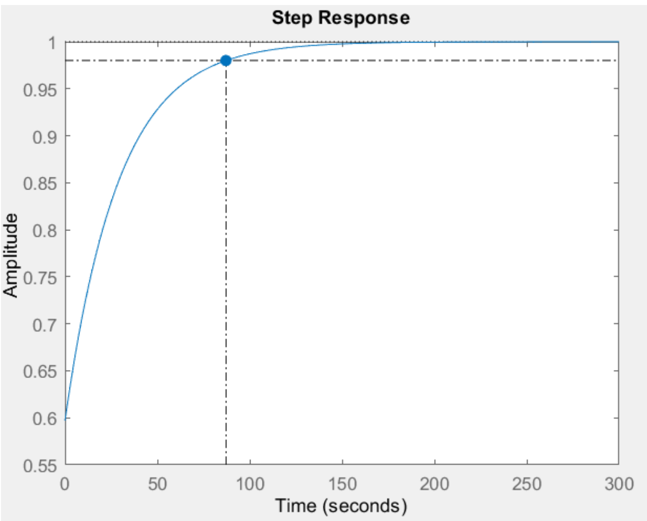
With $K_p = 2.96$, Ti = 34.62 and $T_d = 8.66$, we find the PID controller transfer function Kpid and G_{CL} as,

```
Kpid =

  887.4 s^2 + 102.5 s + 2.96
  --------------------------
          34.62 s

Gcl =

  887.4 s^2 + 102.5 s + 2.96
  --------------------------
  1487 s^2 + 137.1 s + 2.96
```



The settling time in this case was 87.1s, with the cascaded system again exhibiting first order dynamics with zero steady state error and no peak overshoot.

Autotuned PID controller

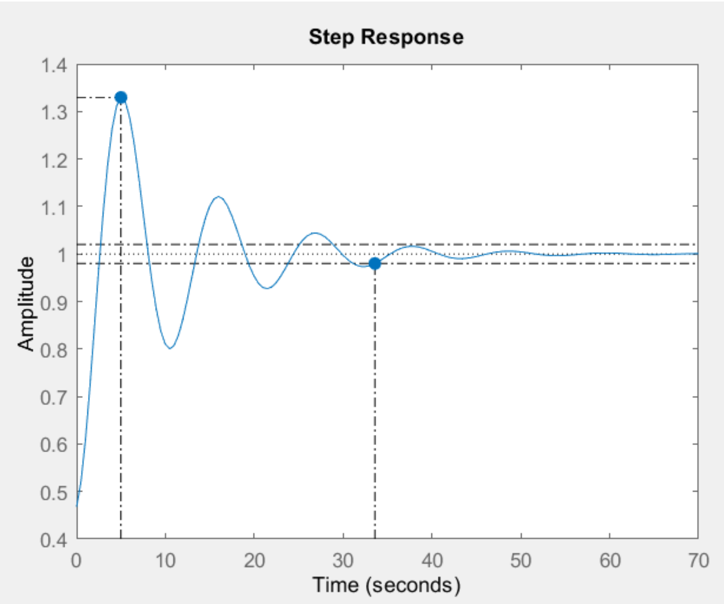
$K_p = 5$, $K_I = 11.1$, $K_D = 15.2$

The CLTF is,

```
Gcl =

  15.2 s^2 + 5 s + 11.1
  ----------------------
  32.51 s^2 + 6 s + 11.1
```

And the step response is,



The settling time is 33.6s, a significant reduction from the PI and PID controllers designed. However, the autotuned controller introduces oscillatory dynamics in the system, with a peak overshoot of 33%. This is a decent trade-off, considering the fact that the autotuned controller drives down the error to zero at a faster speed.

Note: It is possible that an even better controller exists for this thermal system. It may be possible that variations in parameters like feedback loop gain or the introduction of more complex controllers (with more poles and zeroes) might give better results. However, we need not study them, since the system on which we performed the experimentation is inherently a closed loop one, and there is no way for us to open the loops and change the parameters. For example, the gain in the error sensed by the RTD is coded into the software itself, and it makes no practical sense for us to theoretically simulate the variation in this parameter.

Comparative Tabulation

Controller	Settling Time	Steady State Error	Remarks
P	19.5	0.288	Best settling time, however a significant steady state error
PI	60.5	0	No steady state error at the cost of a higher settling time
PID	87.1	0	Higher settling time with no improvements – not recommended
PID (Autotuned)	33.6	0	Introduction of second order dynamics with a 33% overshoot, but with a smaller settling time and no steady state error.

Conclusion

We successfully calculated the time constant for a first order system and designed P, PI and PID controllers for it using Zeigler – Nichols Open Loop Tuning Method. Then, we used MATLAB to compare the step responses and their characteristics of various controllers and the auto-tuned PID controller. The autotuned controller performs better than the PI and PID controllers designed, as far as the settling times were concerned. It also works better than a simple P controller by driving down the error to zero (at the expense of a higher settling time).

Our project finds out that the autotuned controller, while introducing second order dynamics, works best for the given thermal system than all the other controllers tested.

Finally, some non-idealities and irregularities were observed during the project, which can be explained by the effect of the cooling fan (which is not a first order system and has not been accounted for) and inaccuracies in measurements.