Submitted By: Group 15

Harsh Kadian (2017eeb1142)

Mrityunjay (2017eeb1153)

Pradeep (2017eeb1159)

DYNAMIC CONTROL OF SYSTEMS UNDER P/PI/PID CONTROL ON SIEMENS SCADA

OBJECTIVE

- To analyze the behavior of a temperature sensor-based furnace system (first-order), and design of P, PI, and PID controller for the system.
- To discuss the possible non-linearities and deviations from the first-order behavior.

SYSTEM'S DESCRIPTION

The system consists of a slow oven with a heater whose output signal is feedback into a P/PI/PID. The cooling mechanism is achieved with the installation of a fan that is associated with the system with a Bang-Bang control.

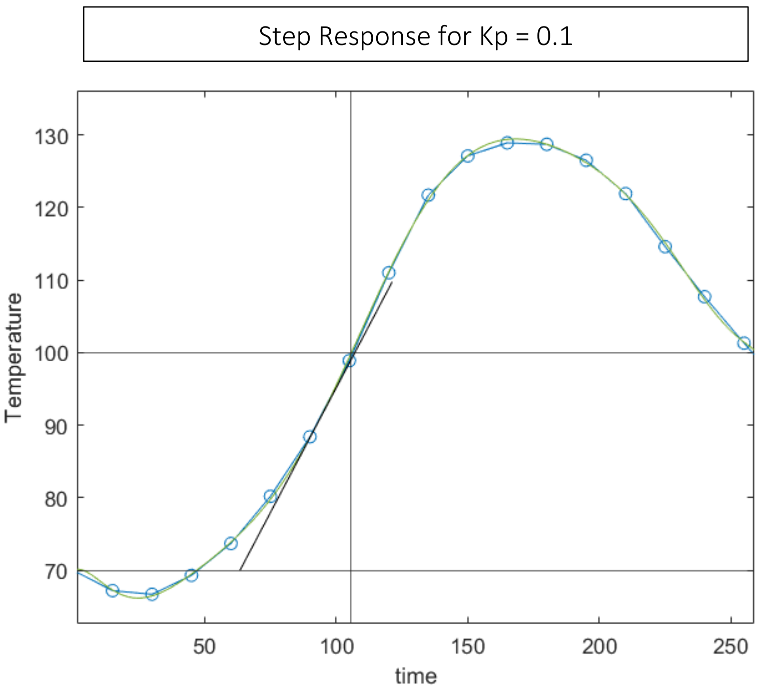
We should note that the cooling mechanism is an entirely different control achieved with Bang-Bang control as mentioned above which here translated such that as soon as the temperature of the system gets closer to set value by less than 2°C, the fan turns on automatically. Moreover, the fan turns off only when the temperature falls by the same amount from the Set Value, which sets the lower limit on the functionality and behaviour of a closed-loop system such that the absolute error from set value could not be less than 2°C.

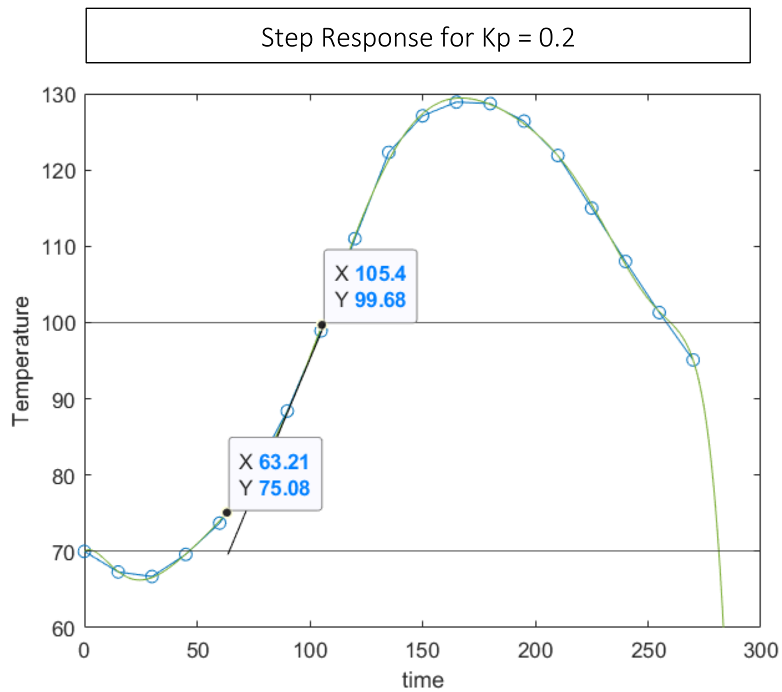
Another deviation from ideal first-order behavior lies in the interaction of the system with the surrounding, which affects the heating up as well as cooling of the system. The fan, which is installed inside which is impractical for cooling the system and its interaction with the surrounding at ambient temperature is not linear either.

To design the control system transfer function with first-order ZN rules, we are required to get the open-loop response of the system. Since this is not viable as the system is already hooked up, we change the PID such that only K_p is for the system, which also takes a minimum gain value.

STEP RESPONSE BEHAVIOR

Shown below is the behavior of the system for different values of Kp which are taken to be 0.1, 0.2 and 0.3 and tabulated are the values of Dead Time and Time constant for different values:



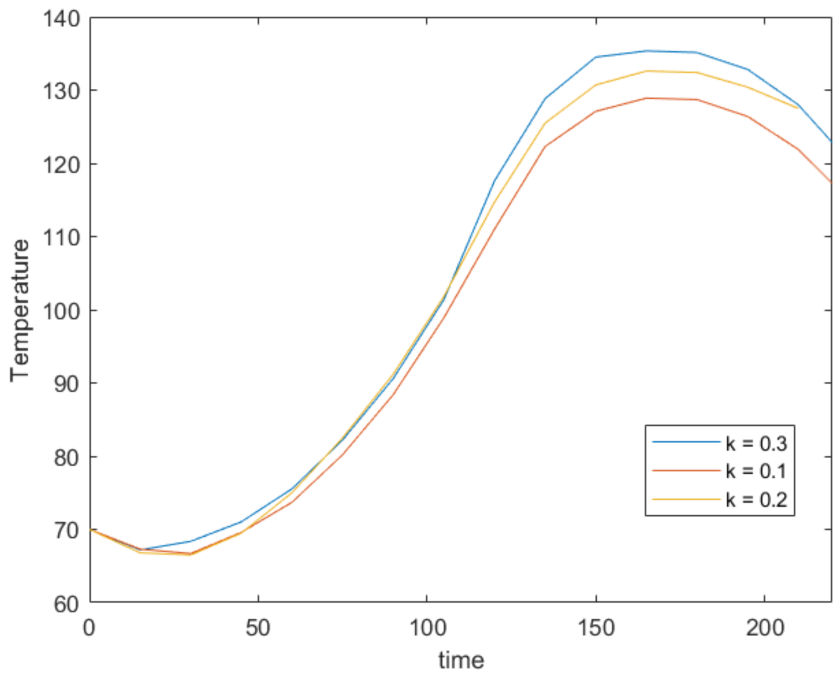


Kp	0.1	0.2	0.3	Average
Lag	61.87	63.21	62.59	62.55
Time Constant	40.13	42.19	43.01	41.77

The system showed the first-order delay and although the response is initially first order, but as soon as the fan turns on and the system starts cooling down the system is no longer first order. And therefore, oscillatory behavior is observed. Due to prolonged cooling, it takes a long time to cool down and is one of the reasons for slow behavior of the system and overshoot which is almost 90%.

Lag and Time Constant are calculated for the ZN first-order rules to design the controllers, the behavior is consistent for these small values and to predict better they are averaged out to get better precision.

Also, it was observed, which is quite apparent though, that the response becomes faster as you increase Kp which also directly affects the overshoot, one of the fallbacks of a proportional controller.



P/PI/PID Control Design

P/PI/PID controls are designed with ZN rules for fist order system, which are summarised below:

Controller	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	2L	0.5L

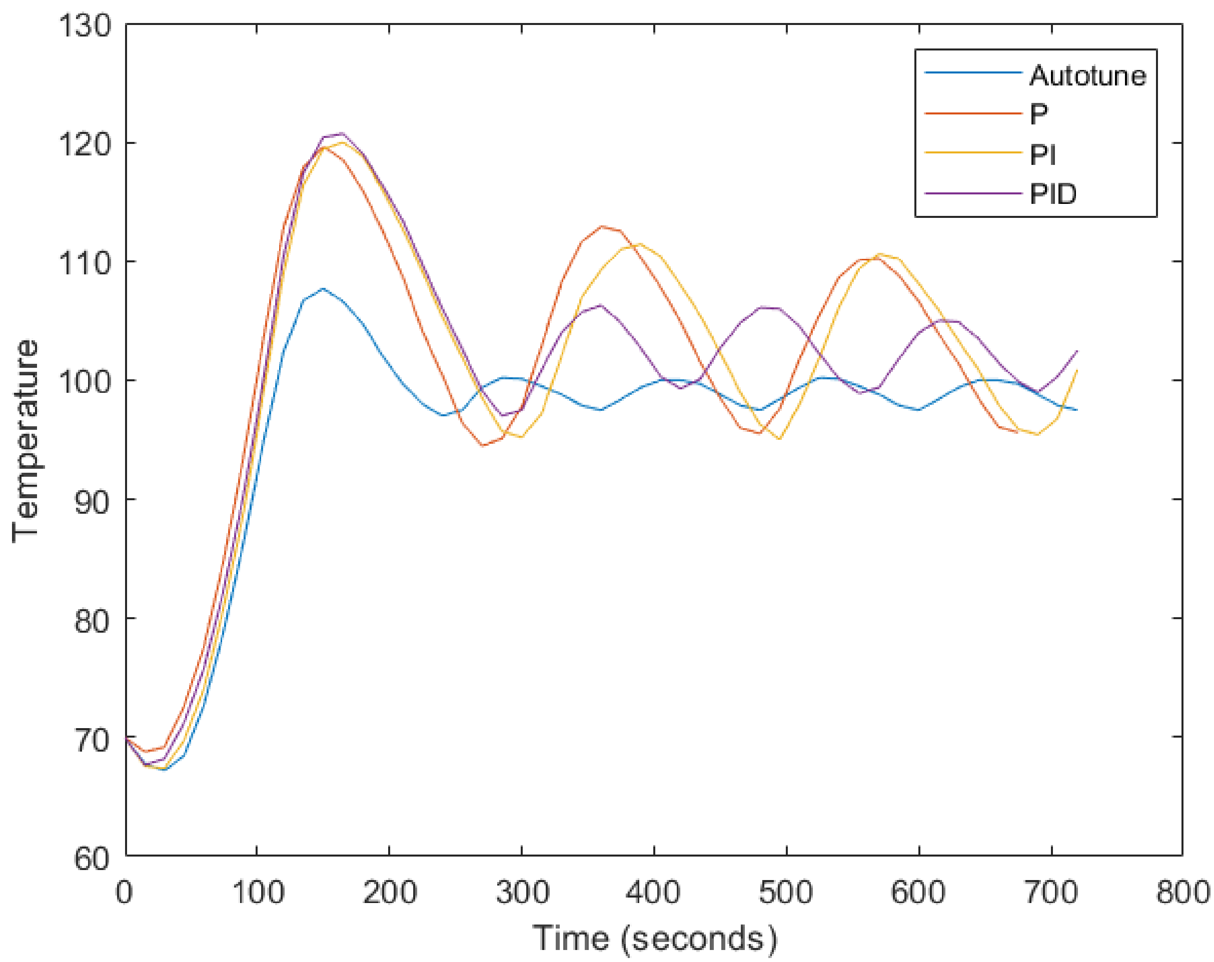
This results in the Control System with values as tabulated below along with the controller Autotuned corresponding to the operating points:

Controller	K_p	T_i	T_d
P	0.476	0	0
PI	0.428	206.4	0
PID	0.571	125.1	31.275
Autotune	17.8	154	29.5

Autotuning usually takes place during the run of the system say the system is heated from a temperature to another, this usually is some rule-based calculation similar to what an engineer would do, taking into account dynamic response behavior like overshoot, rise time, undershoot to tune the controller.

THE BEHAVIOR OF P/PI/PID AND AUTOTUNE

Shown below are the responses of the system under the application of different control systems:



As is visible, all the controllers have similar behavior initially. However, as expected of the P controller, it shows highly oscillatory behavior, whereas the PI controller which usually improves the response of the system does not prove to be of much use. On the other hand, PID, with derivative control it can reduce the oscillatory behavior as it slows down the system thus can compensate for the overshoot. With PID in place system's overshoot reduces significantly.

When Autotune is placed, it showed the best behavior with very less overshoot, and response achieves steady state pretty fast.

The dynamic performance characteristics are shown below:

	P	PI	PID	Autotune
Peak	119.6	120	120.8	107.7
Peak Time	149.8	160	155.2	144.5
Rise Time	48.5	49.7	48.5	47
Overshoot	65.33%	66.67%	69.33%	25.66%
Settling Min	94.44	95.1	97	97

Characteristics like, Rise Time, Overshoot and Peak remain the same except for the exception of Peak in Autotune where peak was shallow. Also, it is essential to note that the Settling time is not defined as the oscillations are very much pronounced (≥5% of set value). Settling Minimum decreases as the controller becomes more and more complex and is minimum for Autotuned controller.

Tradeoffs in the design are pretty much based on application, and simple controllers could also be designed for satisfactory performance whereas complex application might even require controllers designed better than autotune with much lesser overshoot, faster rise time and other requirements.