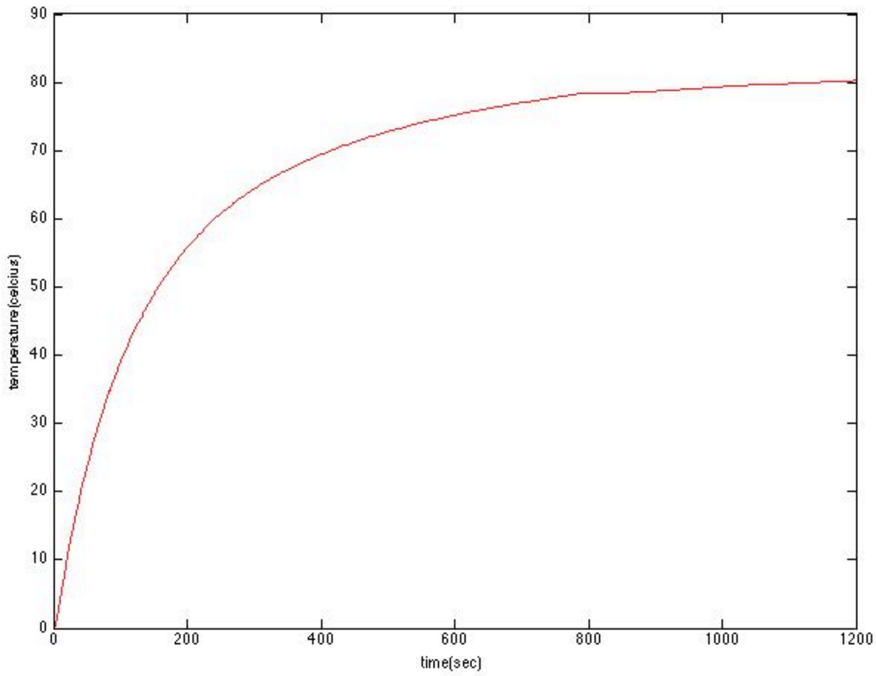
Report on Modeling and PID Tuning of Cool-down Process

Binghui Liu

I. Open loop Modeling

With two pieces of data, “cooldown_20130419_13.24.44.csv” and “cooldown_20130419_15.45.06”, a figure of output and time could be plotted.

Take “cooldown_20130419_13.24.44.csv” as an example. Below is the step response of the open loop system (open_loop_cooldown.m). Input is *u:control*, stepping from 0 to 4.31Amp.



As the figure above indicates, the system can be described as a first-order inertial system, which means transfer function of the system is :

$$Gp(s) = \frac{Ke^{-\tau s}}{Ts+1}$$

K, T and tau are the three parameters that need to be generated from the give data.

Firstly, as a commonly used method in practice developed by Ziegler-Nichols, with known data step response of a system as such, hereby the parameters can be computed as:

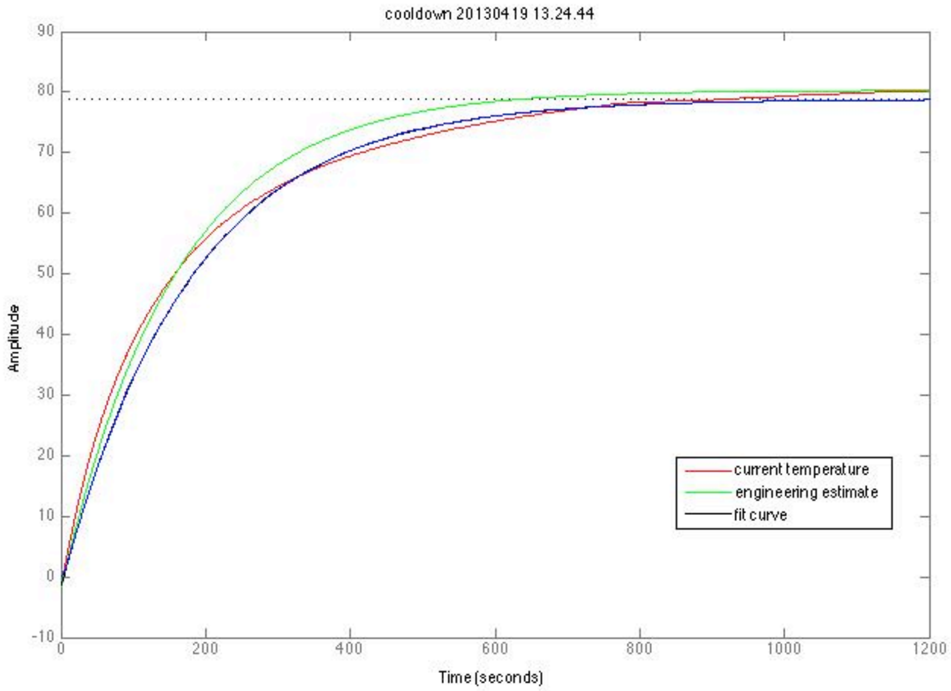
1. $K = y(\infty)/x_0$
2. $y(T) = 63.2\% \cdot y(\infty)$
3. tau is the time take before response curve is going to rise

From the information above, there is K = 80.3, T = 157.8370 and tau = 4.1840.

Secondly, to be more precise, the paramters are generated in a numeric way: with *fitcurve* function (fitcurve.m) using fminsearch() to compute K, T and tau from the data, as function of this step response can be described as :

$$y(t) = K(1 - e^{-\frac{t-\tau}{T}})U(t)\text{, with } U((t-\tau) >= 0) = 1$$

In the function, with time and output data, K, t and tau can be estimated, and the result is: K = 78.8202, T = 178.0315 and tau = - 11.2538. And to match physical



model, tau should actually be above zero, so here it is still taken as tau = 4.1840

Thirdly, plot step response of two sets of parameters on the same figure. It can be concluded that results from *fitcurve* parameters matches original curve better that Z-N parameters.

Because PID control does not require most exquisite parameters of the system model, therefore averages of two set of parameters by *fitcurve* method from two cool down data are used as the final result. So transfer function (TF) of open loop model for cool down process is:

$$Gp(s) = \frac{77.8135e^{-3.691s}}{184.8891s + 1}$$

Note that in this TF, input measurement is Ampere, and output measurement is Celsius.

II. PID Tuning and Simulation

With system model of this first-order system with delay generated as mentioned in step I, there are several ways to tune parameters of PID controller.

Since the control process is an actual digital PID control, which means it needs to be simulated in a discrete way, so Z-transform is applied here, and the discrete object for this first-order system is:

$$yout(k) = -den(2)y(k-1) + num(2)u(k-1)$$

here k is the number of sample, num and den is the factors of numerator polynomial and denominator polynomial respectively.

Three methods are used to tune PID parameters.
Controller follows:

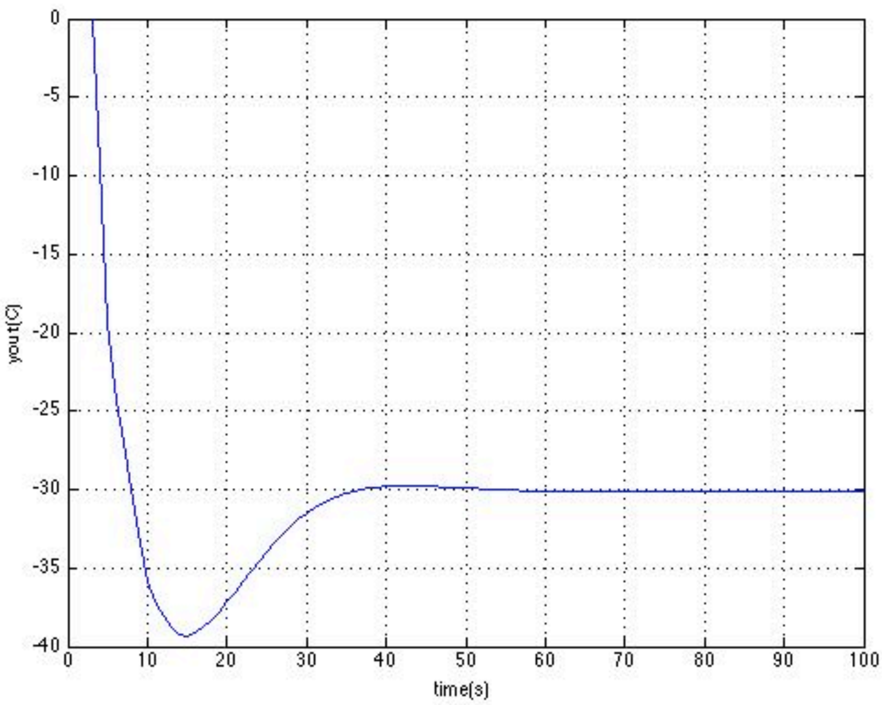
$$u(t) = Kc[1 + 1/T_I s + T_D s]e(t)$$

Setpoint is 2430 0.1 degK, as in “cooling_P100_I100_D1000.csv”, and transferred into Celsius. Output is in Celsius, and input is in Ampere. Loop is operated per second.

1. Cohen-Coon

$$\left.\begin{aligned} KcK &= 1.35(\tau/T)^{-1} + 0.27 \\ T_I/T &= [2.5(\tau/T) + 0.5(\tau/T)^2]/[1 + 0.6(\tau/T)] \\ T_D/T &= 0.37(\tau/T)[1 + 0.2(\tau/T)] \end{aligned}\right\}$$

Empirical equation as above,
Kp = 0.8725,
Ki = 0.0953,
Kd = 1.1868.
Figure of close loop is right:

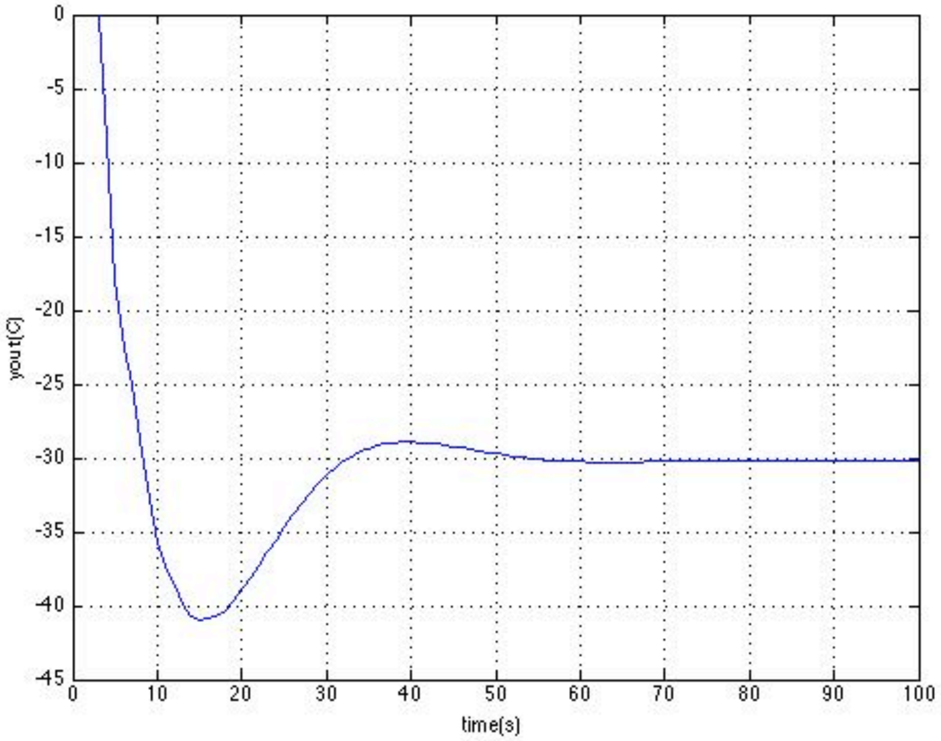


2. Ziegler-Nichols

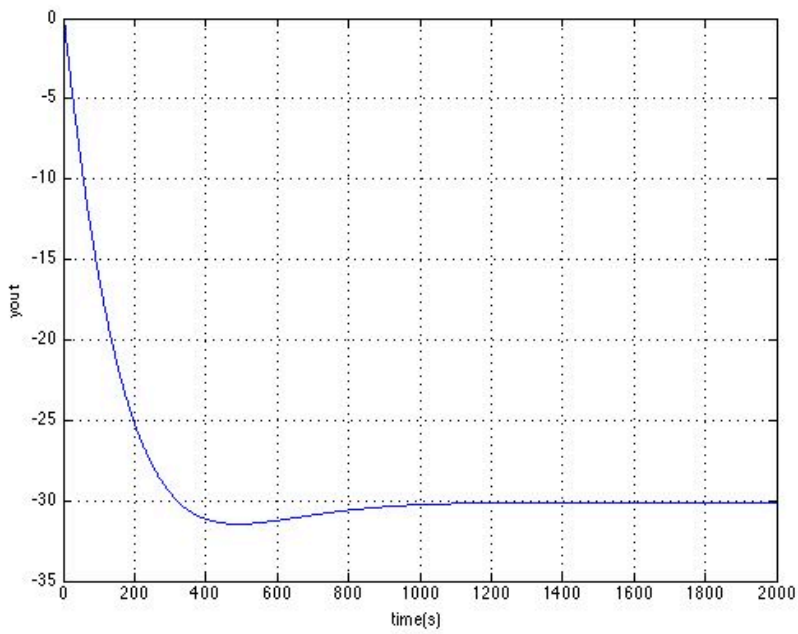
$$\begin{cases} KKc = A(\tau / T)^{-B} \\ T_I / T = C(\tau / T)^{D} \\ T_D / T = E(\tau / T)^{F} \end{cases}$$

In the empirical equation above, A = 1.200, B = 1.000, C = 2.000, D = 1.000, E = 0.500, F = 1.000

Kp = 0.7725
Ki = 0.1046
Kd = 1.4256
Figure of close loop is right:



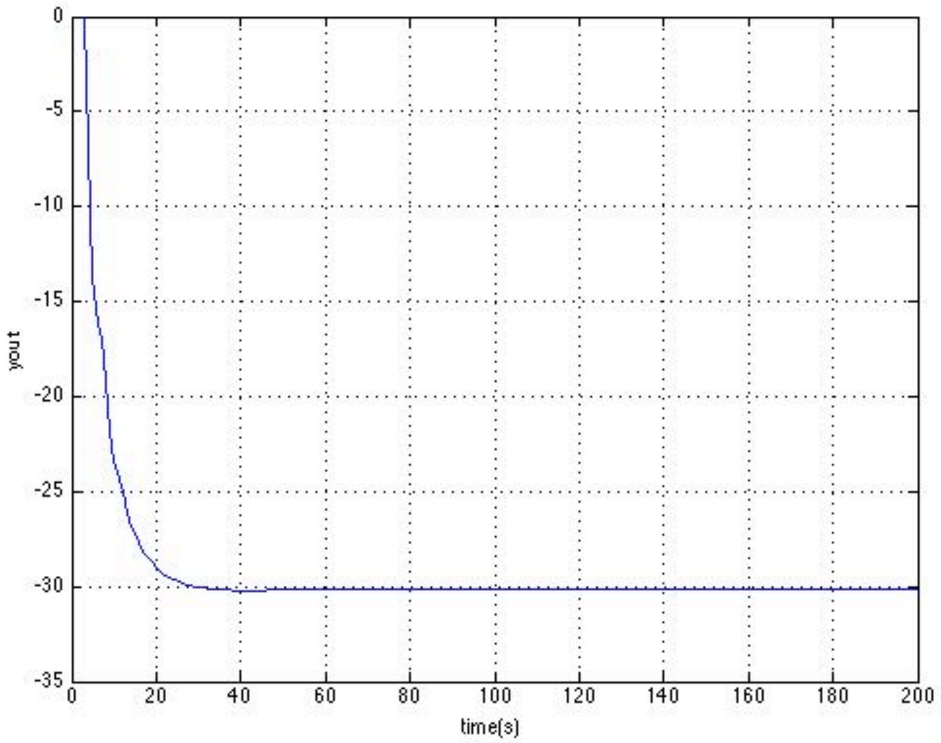
3. PID Design by Jürgen Hahn Group of Rensselaer Polytechnic Institute
Details, documentation and codes can be referred in http://homepages.rpi.edu/~hahnj/PID_Design_GUI/pid_design_gui.html . In a word, this is an automatic PID tuning software.
With open loop TF, PID parameters can be generated as:
Kp = 0.0331,
Ki = 0.0002,
Kd = 1.3817,
Figure of close loop is right:



The close loop curve of this method has a performance with longer time to stability, yet a far weaker vibration and overstrike, as the former two reach more or less -40 C, while this method has not reached -33 C.

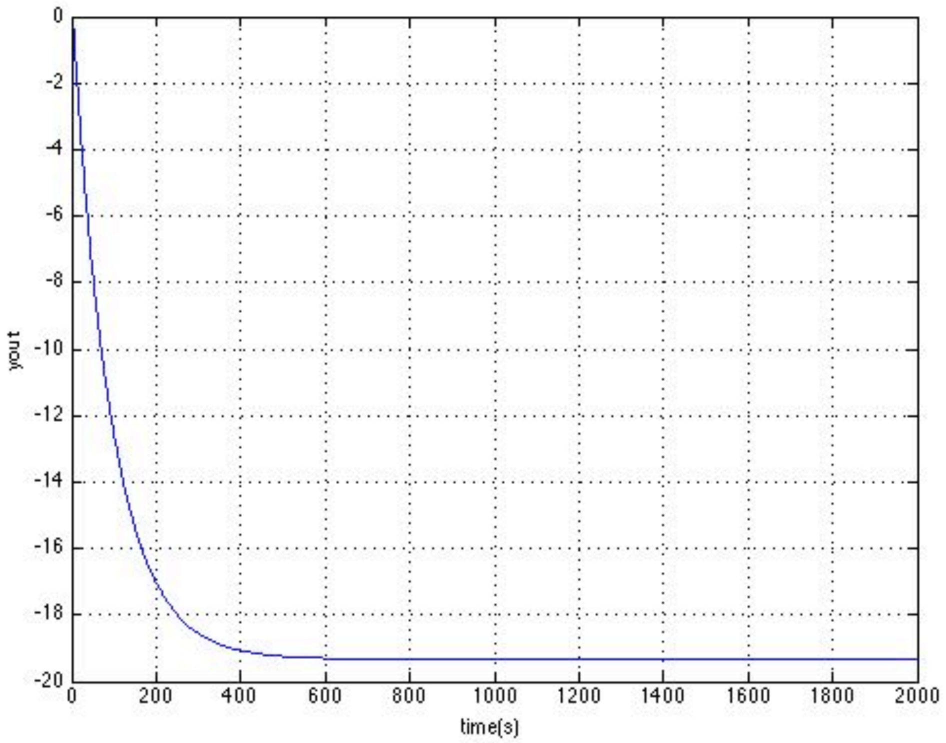
However, those figures are plotted without any restriction on *control* or *error*.
Restrictions should be:
(1) Maximum error values will be ~800 or 80 degK
(2) Clip error sum to +/- br_MaxPidControlValue, which is 6.01Amp
(3) Limit control increases in current to 1/16 of maximum.
(4) Limit absolute current to maximum for the control (60100 * 100uA = 6.01 Amps).
(5) Limit minimum current to 0.

The No.2 restriction has a drastic impact on response curve.
With parameters from C-C or Z-N method, it's easy to see that the curve barely has



vibration and falls to setpoint dramatically.
From my view, this is somehow due to a forced limitation on *error* (and *control*), that the actual effect of control is weakened.
These limitations are based on actual hardware requirement, but for simulation, which is based on ideal circumstance, they could cause different result that theoretically expected.

For example, with the same restrictions on parameters from Hahn's tuning result, which has a more ideal control effect, now has a close loop that will remain at around -19 C.



This may be due to a small Kp or Ki (much likely), which is above the realm of simple PID control tuning from simulation results.

With regards to one of the difference between this Matlab simulation and the C codes for actual control, *control* in the simulation comes in Ampere, and in the C codes it comes in 10uA, so from my understanding, the parameters used in the codes is 10^4 times of actual PID control parameters.

*All the codes and data file mentioned can be found at https://github.com/aceisScope/pidtunning