Report on: PID Modelling of the Cryocooler temperature control system

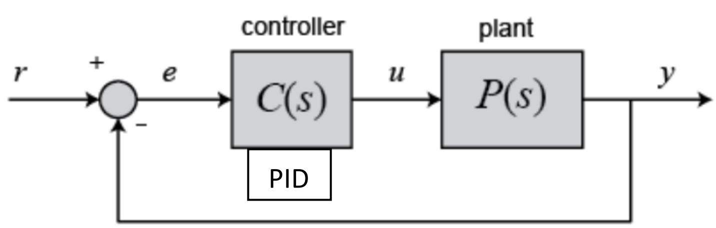
Arjun Ghosh

ABSTRACT

The Cryotel-MT Cryocooler (to be used for APSERa) from SunPower, is a Stirling Cryocooler which is interfaced with AVC system for control interface, power and communication. At the heart of the AVC Controller, for the temperature control is a second order negative feedback-based system with a PID Controller. Tuning the PID controller for optimal performance is necessary for optimum experiment conditions. Thus, we need to delve into the parameters of the PID which we can set: K_p, K_i, K_d, for optimal performance.

Modelling the Plant

The plant is considered to be a second order system, as shown below. For our case plant is the cooler, which receives a step input and performs its job, i.e. cooling. When it reaches the set temperature (pre-given by the user) the cooler is turned off. Via virtue of the experiment after the cooler is off for some time, the temperature again rises which and cooler is turned on again and this cycle repeats. This control is done by the PID controller. A PID controller is defined as by three gains, K_p, K_i, K_d; which require to be tuned.



As shown in the schematic above, we can see a cascaded structure of Controller followed by Plant. We have the following transfer functions for the PID and the second order unity negative feedback-based system as:

$$T.F._{PID} = K_p + \frac{K_i}{s} + K_d s$$

$$T.F._{overall} = \frac{w_n{}^2}{s^2 + 2\zeta w_n + w_n{}^2} \quad \text{(closed loop)}$$

The dependance of primary response parameters of the system on K_p, K_i, K_d, in most of the cases are jisted below.

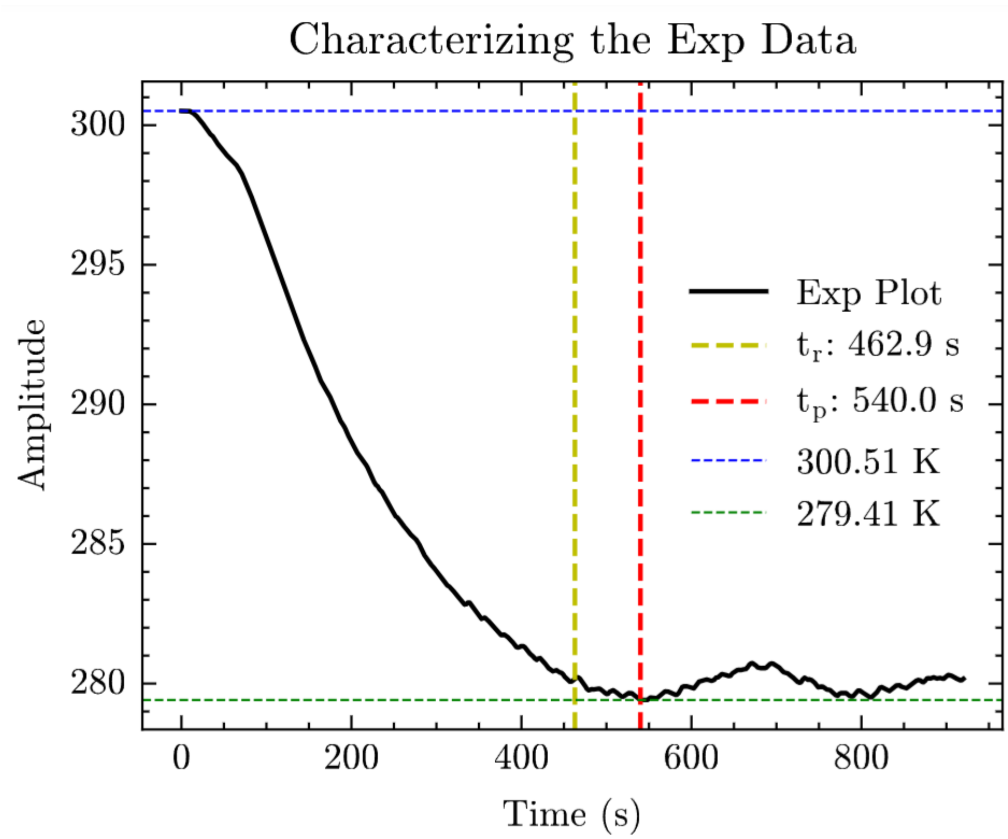
CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Decrease
Kd	Small Change	Decrease	Decrease	No Change

Characterizing the Plant Transfer Function (Exp Data)

The G(s) or the forward path gain is the product of the Transfer Function of the Plant and the PID controller.

$$G(s) = T.F._{total} = T.F._{plant} \times T.F._{PID}$$

We know the $T.F_{total}$ and the $T.F_{PID}$ from experimental data and pre-set values of K_p, K_i, K_d respectively. The cryocooler was run on 31st June, 2024 for around 25 mins. Which gave the temperature versus plot as shown below,



Analysis of the plot was done to yield the following data for the system, as shown below,

Parameter	**Value**
Peak temperature	279.41 K
Percentage overshoot	-0.211 %
Peak time	540 s
Rise time	462.9 s
Damping Factor	0.891
Natural Frequency	0.013 rad/s

Hence, we now know the $T.F_{total}$ from the above data and further the values of K_p, K_i, K_d were found to be: 5, 0.5, 0.2 respectively. Now we can extract the $T.F_{plant}$ which was obtained as shown below.

$$T.F._{plant} = \frac{0.0001641}{0.2s^3 + 5.005s^2 + 0.6141s + 0.01141}$$

Now we can multiply our new values of K_p, K_i, K_d (say for our case we multiply with 5, 0.5, 0.2 respectively, i.e. the same values as before, to observe the time response of the current set system, both of which can be set as per our convenience as well), to the above equation and we get,

$$G'(s) = \frac{0.1681s^2 + 0.02243s + 0.0006731}{0.2s^4 + 5.005s^3 + 0.6141s^2 + 0.01141s}$$

We know that, for determining closed loop transfer function with open loop transfer function G'(s) and unity feedback, it will be;

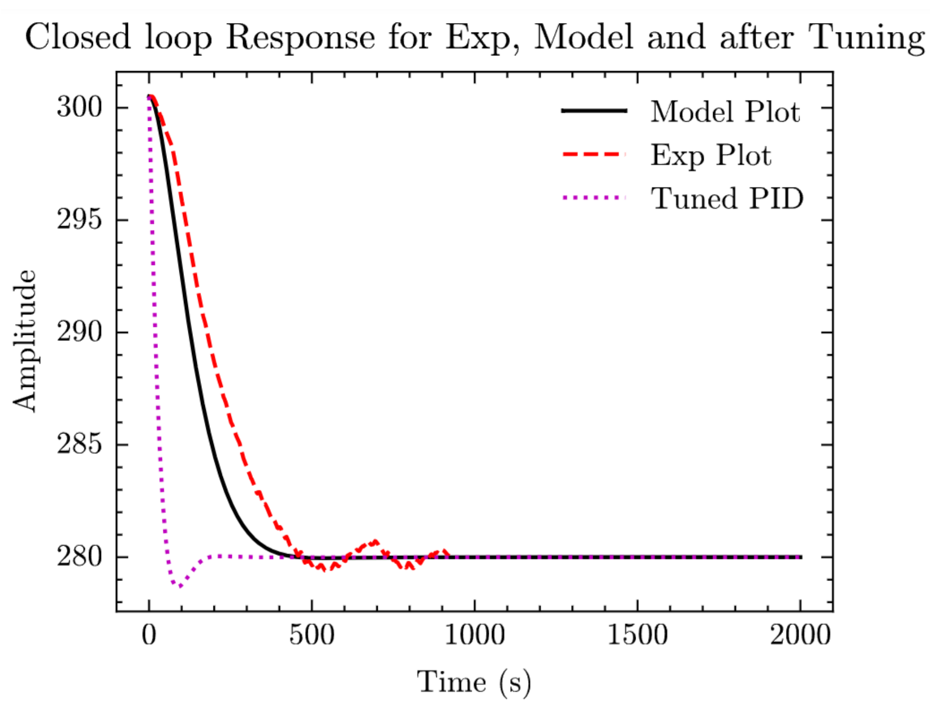
$$T.F._{overall} = \frac{G'(s)}{1+G'(s)}\ (closed\ loop)$$

Hence, we find that;

$$T.F._{overall} = \frac{0.1681s^2 + 0.02243s + 0.0006731}{0.2s^4 + 5.005s^3 + 0.7823s^2 + 0.03384s + 0.0006731}$$

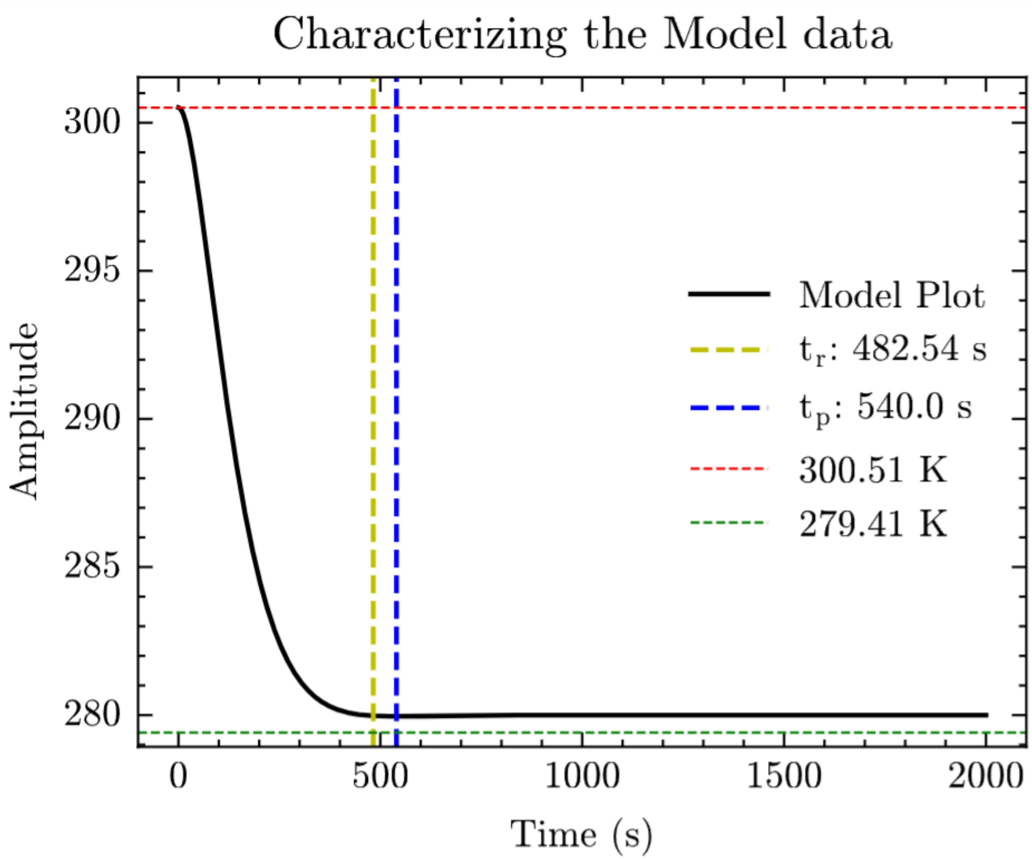
<u>Step Response of the Overall System</u>

We find the step response of the overall system in Python. The general procedure is to multiply the $TF_{overall}$ with the Step function [S(s)], then take the Inverse Laplace transform of the product to get the time response curve. There is a package in python called **control** which does all that for us. The step response plot is shown below for our problem, with label 'Model plot'. Tuning is explained later on.



The black line corresponds to the step response curve of the model we made in the last section. And the red line is the experiment time response curve we obtained previously.

The characterization of the Model plot is given below.

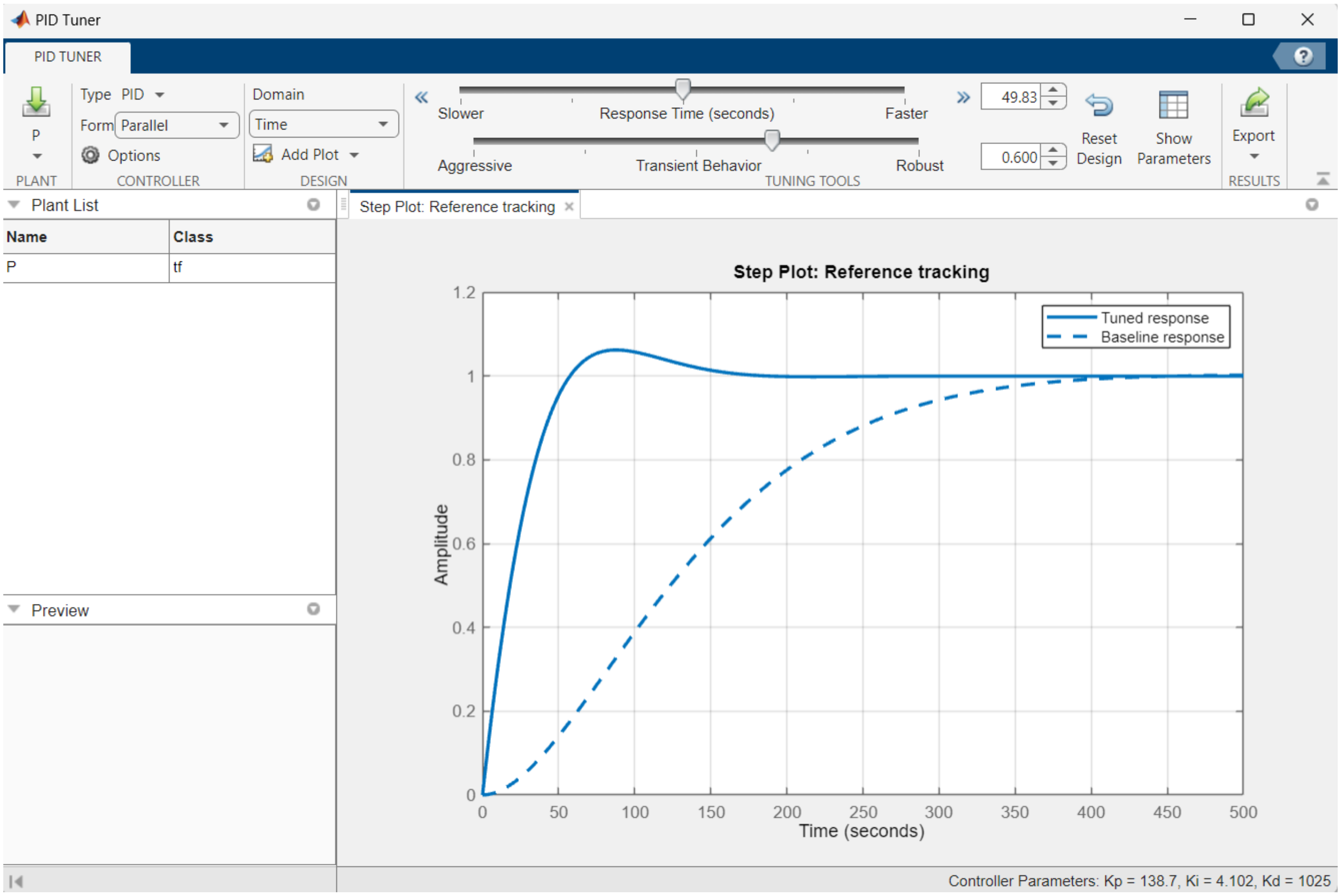


Parameter	Value
Peak temperature	279.367 K
Percentage overshoot	-0.02 %
Peak time	540 s
Rise time	482.54 s
Damping Factor	0.94
Natural Frequency	0.02 rad/s

We can clearly see that the modelling slightly doesn't match our experimental curve in terms of shape. The experimental curve clearly demonstrated a different path of rise for cooling, though they end up with the same rise time. And also has some sort of oscillations as time proceeds with some overshoot. This must be attributed to the physical load (i.e. the copper plate attached to the cold tip of the cryocooler).

Tuning of the System

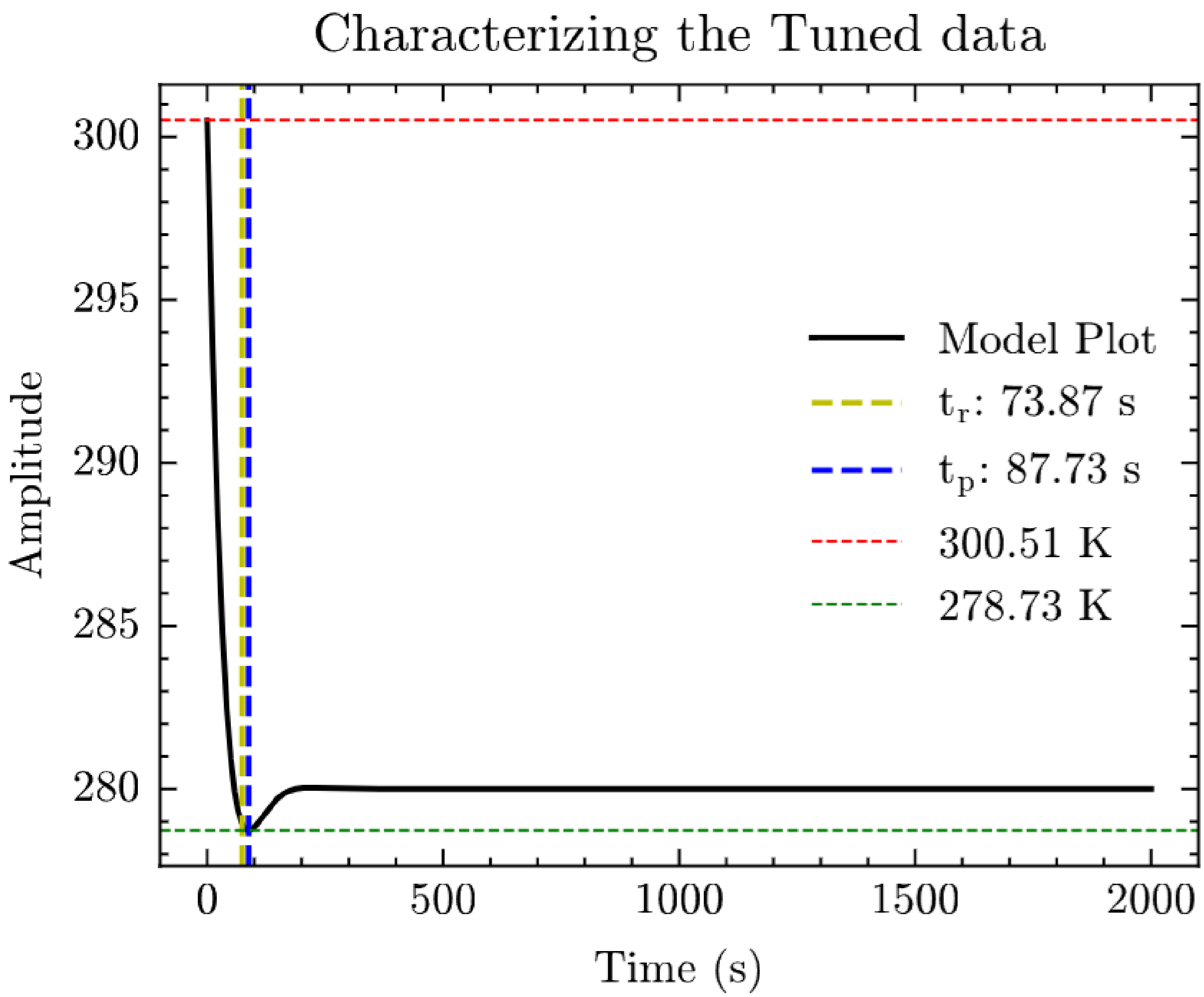
MATLAB has a command pidtuner, which finds the best PID gains for the system. On using the system, we found the values to be as: $K_p = 138.675$, $K_i = 4.1025$, $K_d = 1024.7449$.



We found the values of the tuned system as shown below.

Parameter	Value
Peak temperature	278.727 K
Percentage overshoot	-0.45 %
Peak time	87.73 s
Rise time	73.87 s
Damping Factor	0.86
Natural Frequency	0.07 rad/s

Also, the step response for the tuned system is as shown:



Conclusion:

We find a significant improvement of the tuned system over the model system after tuning. However, we need longer observation data for the system, to observe its settling time and steady state error for understanding the steady state conditions.

Also, K_d values being applied by the MATLAB are doubtful to be helpful for the high thermal inertia of the cryocooler. Trying to remove them will also be of interest in later reports.

References:

[1] Bhatt, Jiten H., Shail J. Dave, Manish M. Mehta, and Nitin Upadhyay. "Derivation of transfer function model based on miniaturized cryocooler behavior." INROADS-An International Journal of Jaipur National University 5, no. 1s (2016): 336-340.

[2] Fuller, Sawyer, Ben Greiner, Jason Moore, Richard Murray, René van Paassen, and Rory Yorke. "The python control systems library (python-control)." In 2021 60th IEEE Conference on Decision and Control (CDC), pp. 4875-4881. IEEE, 2021.