Temperature Control Lab G: Nonlinear Model Predictive Control

Objective: Implement a Model Predictive Control (MPC) application for the Temperature Control lab. Use the model that you developed as part of Lab E where a hybrid model is used in Moving Horizon Estimation for parameter estimation. This document describes a possible hybrid first-principles and empirical model that was also detailed in Lab E. The following model is one possible choice of many types that can be used for this MPC exercise.

Table 1. Parameters from Estimation (***blue*** values are estimated from data)

Quantity	**Default Values or Use Values from Lab E (Preferred)**
Initial temperature (T_0)	23 °C
Ambient temperature (T_∞)	23 °C
Heater Factor (α_1)	***0.0061 W/%***
Heater Output (Q_1)	0 to 100%
Heater Factor (α_2)	***0.0031 W/%***
Heater Output (Q_2)	0 to 100%
Heat Capacity (C_p)	500 J/kg-K
Surface Area Not Between Heat Sinks (A)	1x10^{-3} m^2
Surface Area Between Heat Sinks (A_s)	2x10^{-4} m^2
Mass (m)	0.004 kg
Overall Heat Transfer Coefficient (U)	***4.05 W/m^2-K***
Emissivity (ε)	0.9
Stefan Boltzmann Constant (σ)	5.67x10^{-8} W/m^2-K^4
Empirical time constant (τ)	***15.4 sec***

Parameters α_1, α_2, U, and τ along with other parameters were updated previously in Lab E and are shown in Table 1. Other parameters can be updated such as the ambient temperature from the measured initial temperature (when cool). Four differential equations are one possible description of the temperature dynamics. The first two equations come from an energy balance for each heater.

$$mC_p \frac{dT_{h1}}{dt} = UA\,(T_\infty - T_{h1}) + \epsilon\,\sigma\,A\left(T_\infty^4 - T_{h1}^4\right) + UA_s\,(T_{h2} - T_{h1}) + \epsilon\,\sigma\,A_s\left(T_{h2}^4 - T_{h1}^4\right) + \alpha_1 Q_1$$

$$mC_p \frac{dT_{h2}}{dt} = UA\,(T_\infty - T_{h2}) + \epsilon\,\sigma\,A\left(T_\infty^4 - T_{h2}^4\right) + UA_s\,(T_{h1} - T_{h2}) + \epsilon\,\sigma\,A_s\left(T_{h1}^4 - T_{h2}^4\right) + \alpha_2 Q_2$$

Additional 2 empirical equations capture the temperature delay between the heater and the temperature sensor as:

$$\tau \frac{dT_{c1}}{dt} = -T_{c1} + T_{h1} \qquad \text{and} \qquad \tau \frac{dT_{c2}}{dt} = -T_{c2} + T_{h2}$$

where T_c is the temperature of the sensor and T_h is the temperature of the heater. The additional number 1 or 2 refers to which heater and sensor. T_{c1} is the temperature of sensor 1 and T_{c2} is the temperature of sensor 2. The heater temperatures, T_{h1} and T_{h2}, are predicted from the equations but are not directly measured.

A parameter estimation exercise (see Figure 1) is used to identify the parameters in Table 1. The four parameters α_1, α_2, U, and τ are adjusted to minimize the error between the measure and predicted temperatures.

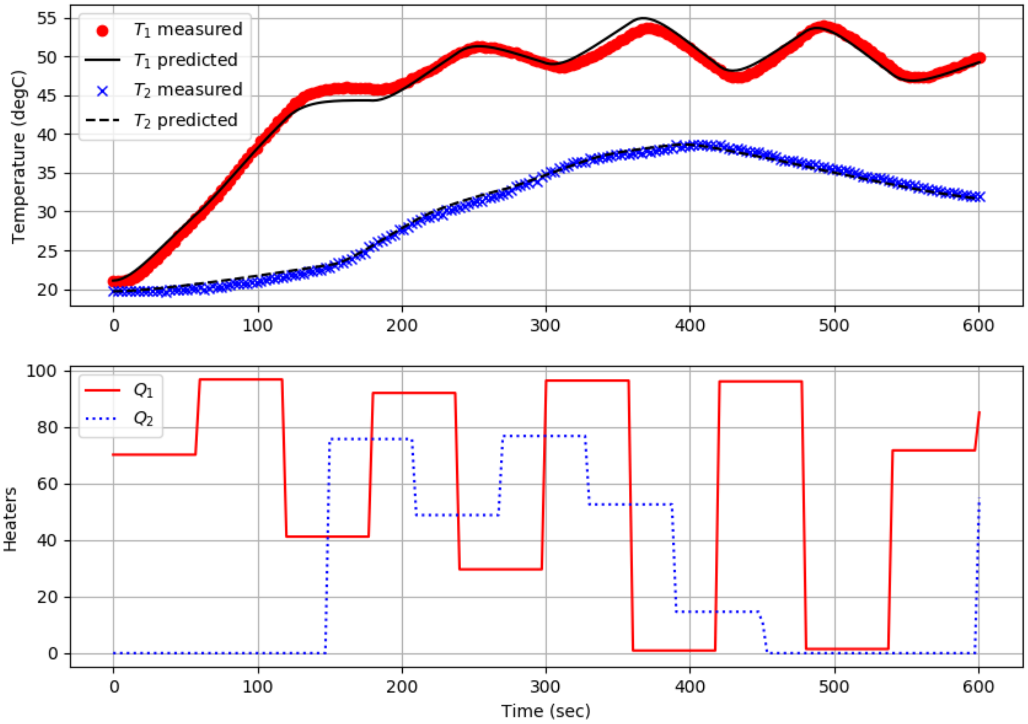

Figure 1. Parameter Estimation Results

The parameter estimation exercise can be rerun to give a more accurate model of an individual TCLab device, if desired. A more accurate ambient temperature can also be obtained by measuring the temperature with no heater output.

With this model or a custom model, implement an MPC application that follows setpoint changes for both temperatures. It is recommended to use a 10 minute test period where each setpoint changes 2-3 times during the testing. Include a plot of the setpoint values ($T_{1,sp}$ and $T_{2,sp}$), measured temperatures (T_1 and T_2), and heater values (Q_1 and Q_2). Discuss the controller performance and whether model mismatch or controller tuning can be improved.

Questions for Consideration

1. What is the difference in performance between the nonlinear MPC (this lab) and linear MPC (previous lab)? Do the nonlinear correlations significantly improve performance over the linear model? What is the difference in computational cost (solve time) for each cycle of the controller?
2. Some very complicated nonlinear models may fail to converge or intermittently require a larger number of iterations to solve. What backup measures would you suggest for an application that may intermittently fail to provide a solution?
3. One possible way to deal with nonlinear systems in control is to locally approximate the nonlinear system with a linear model. What may be the advantages or disadvantages of linearizing a nonlinear model versus solving the nonlinear model directly in a control application?

See https://github.com/APMonitor/arduino (6_Model_Predictive_Control/2nd_order_nonlinear) for source code examples related to Lab G.