Benchmark Temperature Microcontroller for Process Dynamics and Control

Junho Park[a], R. Abraham Martin[a], Jeffrey D. Kelly[b], John D. Hedengren[a,*]

[a] *Department of Chemical Engineering, Brigham Young University, Provo, Utah, USA*
[b] *Industrial Algorithms, 15 St. Andrews Road, Toronto, ON, Canada, M1P 4C3*

Abstract

Standard benchmarks are important repositories to establish comparisons between competing model and control methods, especially when a new method is proposed. This paper presents details of an Arduino micro-controller temperature control lab as a benchmark for modeling and control methods. As opposed to simulation studies, a physical benchmark considers real process characteristics such as the requirement to meet a cycle time, discrete sampling intervals, communication overhead with the process, and model mismatch. An example case study of the benchmark is quantifying an optimization approach for a PID controller with 5.4% improved performance. A multivariate example shows the quantified performance improvement by using model predictive control with a physics-based model, an autoregressive time series model, and a Hammerstein model with an artificial neural network to capture the static nonlinearity. These results demonstrate the potential of a hardware benchmark for transient modeling and regulatory or advanced control methods.

Keywords: benchmark, dynamics, PID tuning, model predictive control, microcontroller

*corresponding author
Email address: `john_hedengren@byu.edu` (John D. Hedengren)

1. Introduction

Benchmark problems are standard repositories in many scientific disciplines such as systems biology [1, 2], reservoir modeling [3, 4, 5, 6], drilling [7, 8], optimization [9, 10], dynamic optimization [11, 12], singular optimal control [13, 14], combined scheduling and control [15, 16, 17, 18], and others [19, 20, 21]. The benchmark problems serve as a consistent measure of innovations that are proposed to increase profitability or improve some aspect of control or optimization performance.

There are many standard benchmark models for testing the performance of estimation and control methods in chemical process control. Some of these include a continuously stirred tank reactor (CSTR) with a single exothermic reaction [22, 23, 24]. One of the most commonly cited models in chemical process control is the Tennessee Eastman Process [25, 26]. The Tennessee Eastman Process encapsulates valve characteristics, measurement noise, process nonlinearity, and complex interactions between processing units for chemical manufacture.

Besides simulation, there are standard hardware benchmarks for evaluating control performance such as UAV control [27], process control education modules [28, 29], and quadruple tank level control [30, 31, 32]. There also many studies where the authors build a unique test system or implement control on an industrial process [33, 34] and demonstrate various control methods. However, hardware benchmarks may be difficult to reproduce or the industrial process may be unavailable for independent researchers to also obtain data or test methods in closed-loop.

The purpose of this paper is to demonstrate a standard hardware benchmark for control methods with a micro-controller temperature control device. This Temperature Control Lab (TCLab) is used as an education module for courses in process dynamics and control [35, 36]. As many have noted in assessments of process control education, there is a need to give students realistic and hands-on experiences with process control [37, 38, 39]. Industry desires foundational and practical knowledge of control engineering concepts that are reinforced with physical modules. Because the TCLab, as an educational module, has wide distribution to universities and industrial practitioners (3000 units), it has potential as a standard hardware benchmark for control engineering studies. Section 2 gives details of the device to enable replication of the TCLab.

2. Temperature Control Lab Device

The TCLab is printed circuit board (PCB) shield that connects to an Arduino micro-controller. The TCLab shield has two transistors as heaters and two thermistor temperature sensors as shown in Figure 1. A step response of the heater (0-100%) has a temperature response with an approximate dominant time constant (τ) of $2.9\ min$ and a gain of $0.9\,\frac{^oC}{\%heater}$. The process exhibits second order dynamics and the two adjacent heaters create a compact multivariate control system. The Arduino micro-controller is an Arduino Uno or Arduino Leonardo that includes a 10-bit Analog to Digital Converter (ADC) to measure voltage of the temperature sensors in 1024 (2^{10}) discrete analog levels and Pulse Width Modulation (PWM) with 256 (2^8) levels to change the output to the heaters and LED.

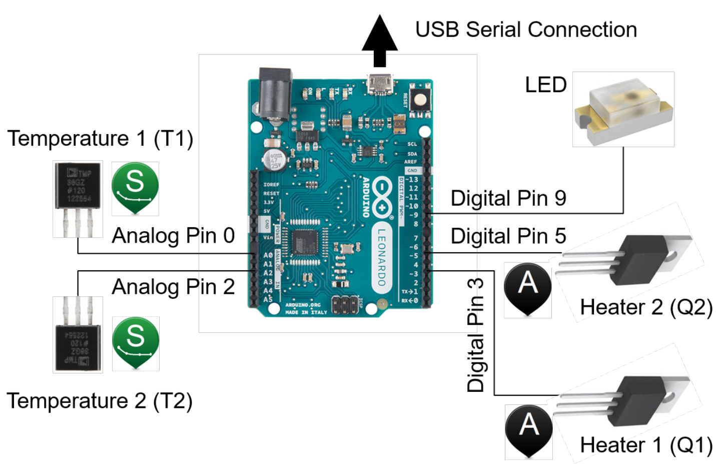
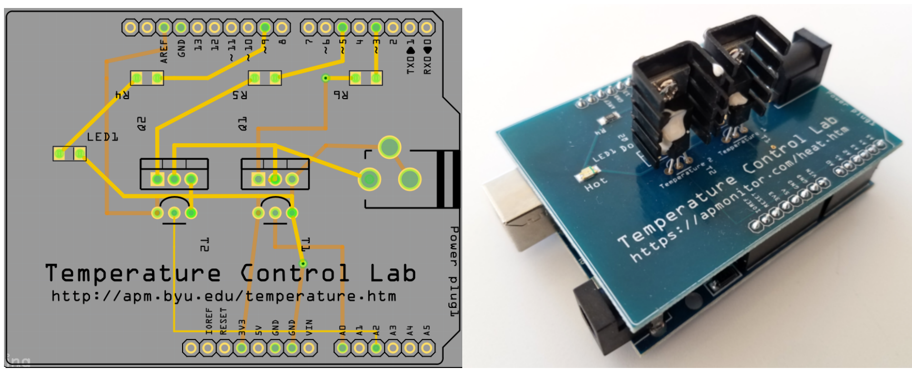


Figure 1: Temperature sensors and heater transistors with connections to an Arduino Leonardo.

The transistor heaters are TIP31C NPN Bipolar Junction Transistors (BJTs) in a TO-220 package. These transistors are commonly used in audio, power, and switching applications but not commonly as heaters. During the development of the TCLab, the initial design was to include a MOSFET transistor (low power loss switch) with a power resistor as the heating element. Instead, the BJT TIP31C is able to act as both the switch and the heater, thereby simplifying the design and reducing the cost of the hardware. The two temperature sensors on the TCLab are standard TMP36GZ thermistors with an output voltage (mV) that is linearly proportional to temperature ($T^oC = 0.1\,mV - 50$) and no requirement for calibration. Typical sensor accuracy is $\pm 1^oC$ at room temperature (25^oC) and $\pm 2^oC$ over the -40^oC to 150^oC operating range.

As a safety and equipment protection precaution, the Arduino micro-controllers come pre-programmed to shut off the heaters if the temperature rises above 100^oC. The heaters are powered by a 5V 2A power supply for a maximum power output of 10 W. A 20 AWG (American Wire Gauge) power cable reduces the power dissipation compared to standard 24 AWG power cables with a barrel jack connector. A USB cable connects the Arduino to a computer for serial data communication. One TIP31C heater and one TMP36GZ sensor are connected to each other and with a thermal heat sink attached to the TIP31C transistor. The two heater units are placed in proximity to each other to transfer heat by convection and thermal radiation.

Software interfaces to TCLab in Python, MATLAB, and Simulink are described in Appendix A. The software adjusts the two heater levels between 0 and 100% and the LED brightness between 0 and 100% using PWM with 2^8 discrete levels. The PWM rapidly fluctuates between on and off to give nearly continuous values $0, 0.392, 0.784, \ldots, 99.61, 100$ for actuation of the heaters and LED.



(a) TCLab Printed Circuit Board Layout (b) TCLab Device

Figure 2: Temperature Control Lab Design

3. Temperature Response Models

This section summarizes four simulation models that describe the dynamic response of the heaters to temperature changes. The four are a lumped parameter energy balance (Section 3.1), a first-order plus dead-time (FOPDT) model (Section 3.2), a higher-order autoregressive exogenous input (ARX) model (Section 3.3), and an artificial neural network (ANN) steady state and linear dynamic Hammerstein model (Section 3.4). Section 3.5 compares all of the models on open-loop step test data both for Single Input Single Output (SISO) and Multiple Input Multiple Output (MIMO) modes. Multivariate, model-based control relies on an accurate simulation of the process. The models described in this section are not an exhaustive list of physics-based and empirical representations. Each TCLab device is slightly different so the model parameters are uniquely identified. One of the principal differences is the ambient temperature where the test occurs. Other potential disturbances include the power supply output, air currents (e.g. nearby computer fan), and others. Figure 3 shows variability due to ambient temperature differences for six tests that use the same heater profile. With $\pm 2.5^oC$ ambient temperature difference, there is a similiar spread in the heater temperature response although the trends are not parallel and completely predictable, especially for heater 2 temperature.

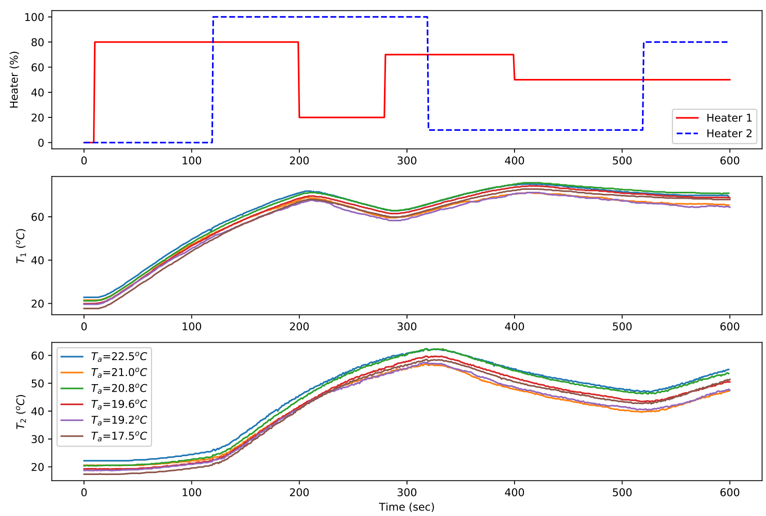


Figure 3: Variations in ambient temperature influence the temperature profiles

Along with measurement noise, the stochastic nature of the data is a feature of the lab that portrays performance on a physical system. Reporting, plotting, or controlling the starting (ambient) temperature is an important requirement of the benchmark as shown in Figure 4.

According to the slope of the regression, an ambient temperature increase of 1^oC equates to a 0.928 ± 0.033^oC rise in average temperature of the step tests. One possible explanation for the slope less than unity is the radiative heat transfer that has a quadratic dependence on absolute temperature and would lose heat at a higher rate at elevated conditions. The main conclusion from this result is that ambient temperature has a reproducible effect on the outcome of benchmark tests and should be reported and controlled for repeatable results.

3.1. Physics-based Model

A lumped parameter model with convection, conduction, and thermal radiation describes the second-order temperature response to heater changes. The lumped parameter model is a simplification of a more rigorous finite element

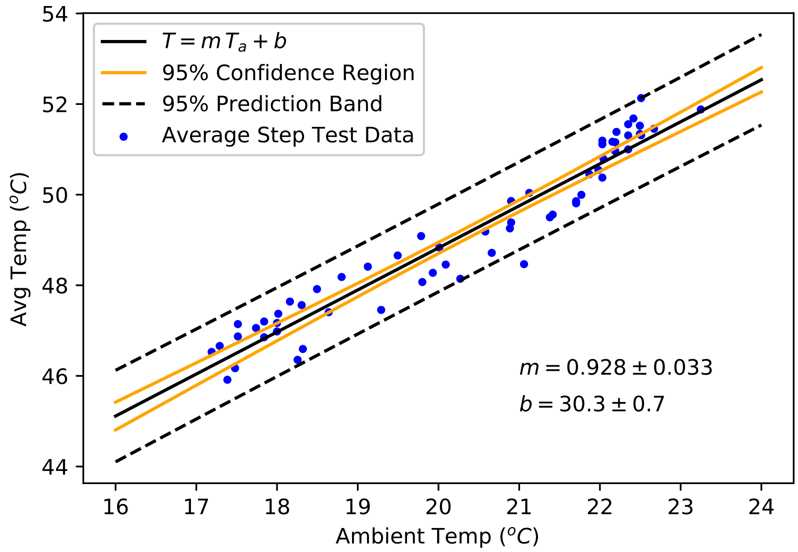


Figure 4: Correlation of ambient temperature to average temperature during 60 step tests (10 min each)

analysis (FEA) that tracks the temperature distribution throughout the heat sink and loss to the environment as shown in Figure 5.

Details of the FEA simulation are not provided here but do provide a confirmation that the temperature distribution is sufficiently uniform ($< 3^oC$) for a lumped parameter assumption. The lumped parameter model assumes that the heaters (T_{H1} and T_{H2}) and temperature sensors (T_{C1} and T_{C2}) have a uniform temperature. The temperature sensors (T_{C1} and T_{C2}) have a small thermal mass and surface area and temperature changes are driven by heat conduction from the heaters (T_{H1} and T_{H2}) where they are attached with thermal epoxy. Parameters of the lumped parameter model are given in Table 1.

The dynamic input power to each transistor and the temperature sensed by each thermistor is developed with energy balance equations (Equations 1-4) that account for convection, conduction, and thermal radiation. The amount of convective heat transfer from heater 1 to heater 2 is given by $Q_{C12} =$

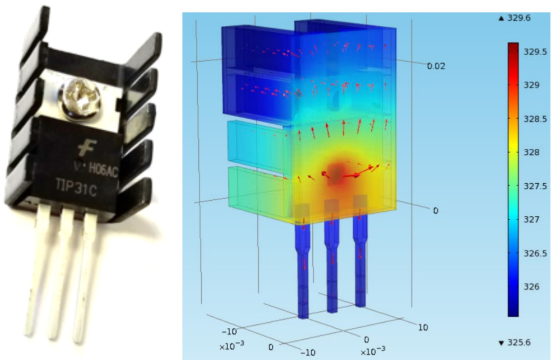


Figure 5: Finite Element Analysis of the Dynamic Temperature Response.

Table 1: Lumped Parameters from Physics-based Model

Quantity	Value
Initial temperature (T_0)	296.15 K (23^oC)
Ambient temperature (T_∞)	296.15 K (23^oC)
Heater output (Q_1)	0 to 1 W
Heater factor (α_1)	0.0131-0.0132 W/(% heater)
Heater output (Q_2)	0 to 0.75 W
Heater factor (α_2)	0.0063-0.0066 W/(% heater)
Heat capacity (C_p)	500 J/kg-K
Surface Area Not Between Heaters (A)	1.0x10^{-3} m^2 (10 cm^2)
Surface Area Between Heaters (A_s)	2x10^{-4} m^2 (2 cm^2)
Mass (m)	0.004 kg (4 gm)
Heat Transfer Coefficient (U)	4.4-4.6 W/m^2-K
Heat Transfer Coefficient Between Heaters (U_s)	23.6-24.4 W/m^2-K
Emissivity (ϵ)	0.9
Stefan Boltzmann Constant (σ)	5.67x10^{-8} W/m^2-K^4
Conduction Time Constant (τ_c)	$21.1-23.3\,sec$

$U_s\,A_s\,(T_{H2}-T_{H1})$. The radiative heat transfer from heater 1 to heater 2 (or vice versa) is given by $Q_{R12} = \epsilon\,\sigma\,A\,\left(T_{H2}^4 - T_{H1}^4\right)$.

$$m\,c_p \frac{dT_{H1}}{dt} = U\,A\,\left(T_\infty - T_{H1}\right) + \epsilon\,\sigma\,A\,\left(T_\infty^4 - T_{H1}^4\right) + Q_{C12} + Q_{R12} + \alpha_1 Q_1 \quad (1)$$

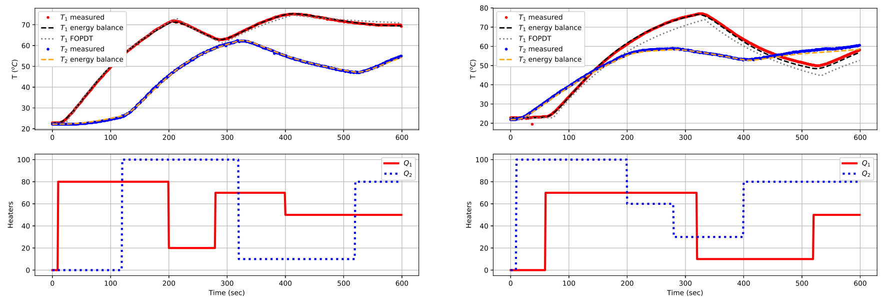
$$m\,c_p \frac{dT_{H2}}{dt} = U\,A\,\left(T_\infty - T_{H2}\right) + \epsilon\,\sigma\,A\,\left(T_\infty^4 - T_{H2}^4\right) - Q_{C12} - Q_{R12} + \alpha_2 Q_2 \quad (2)$$

The dynamic temperature response of the two temperature sensors are primarily by conductive heat transfer from the heaters. The temperature sensors are small in mass and surface area relative to the heaters so the heat transfer by other mechanisms is ignored. The time constant τ_c is a lumped parameter from a discretized version of Fick's Law of heat transfer with $\tau_c = {}^{m_s\,c_{ps}\,\Delta x}\!/_{k_c A_{cond}}$, where m_s is the mass of the sensor, c_{ps} is the heat capacity of the sensor, k_c is the thermal conductivity of the thermal epoxy, and Δx is the width of the thermal epoxy. These parameters are combined together into one parameter τ_c and estimated from the data. The dynamic sensor temperature response expressions are Equations 3 and 4.

$$\tau_c \frac{dT_{C1}}{dt} = T_{H1} - T_{C1} \quad (3)$$

$$\tau_c \frac{dT_{C2}}{dt} = T_{H2} - T_{C2} \quad (4)$$

The test of the physics-based model is performed in two phases that includes a model fitting phase followed by validation. The model fitting adjusts the parameters U, U_s, α_1, α_2, and τ_c to minimize the sum of squared error between the model prediction and data as shown in Figure 6a. The model validation is a simulation of the temperature profile given a different heater profile. The measured temperatures are not used in performing the simulation but are compared afterwards to determine how well the model fitting performs on independent data as shown in Figure 6b.



(a) Model Fitting

(b) Model Validation

Figure 6: Dual Heater Step Response of the TCLab with Physics-based and FOPDT Model

3.2. First-Order Plus Dead-time Model

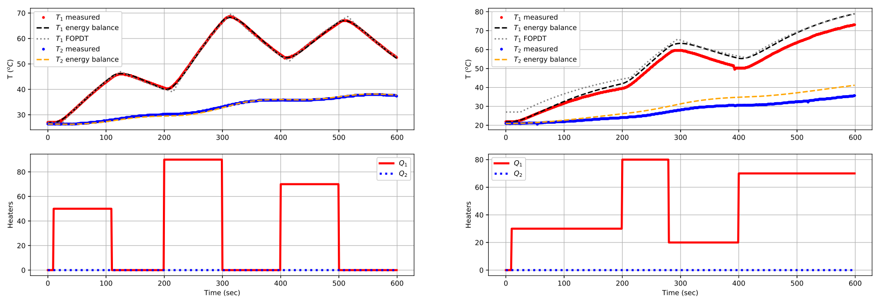
In addition to the physics-based model, a first-order plus dead-time (FOPDT) model is fit to step response data. An FOPDT model includes the gain (K_p=0.92 $^oC/\%$), time constant (τ_p=175.2 sec), and delay time (θ_p=15.6 sec). The FOPDT model is a single differential equation as shown in Equation 5.

$$\tau_p \frac{dT_{C1}}{dt} = -T_{C1} + K_p\, Q_1\,(t - \theta_p) \tag{5}$$

The discrete solution to the FOPDT equation is Equation 6 when there is a zero-order hold for the heaters between sampling intervals (Δt) between time interval j and $j-1$.

$$T_{C1,j} = e^{\frac{-\Delta\, t}{\tau_p}}\,(T_{C1,j-1} - T_{C1,0}) + \left(1 - e^{\frac{-\Delta\, t}{\tau_p}}\right)\, K_p\,\left(Q_{1,j-\theta_p-1} - Q_{1,0}\right) + T_{C1,0} \tag{6}$$

The FOPDT model is used in this example for obtaining initial tuning parameters to a Proportional-Integral-Derivative (PID) controller for an optimization-based tuning approach as detailed in Section 4. Heater 1 (Q_1) is adjusted with variable step sizes and heater 2 (Q_2) remains off to generate step response data for the FOPDT. The results of the temperature data and model fit is shown in Figure 7a and Figure 7b for validation with a different heater profile.



(a) Model Fitting (b) Model Validation

Figure 7: Single Heater Step Response of the TCLab with Physics-based and FOPDT Model

The physics-based model has a lower average absolute error while the FOPDT model has a higher error because a first order model is fit to a higher order response. The physics-based model fits the temperature response better when the heater is adjusted because of the second-order model and nonlinear radiative heat transfer term.

3.3. Linear Time Series Models

Auto-Regressive eXogenous input (ARX) time series models are a linear representation of a dynamic system in discrete time. The ARX, Output Error (OE), Finite Impulse Response (FIR), State Space (SS), and other forms are common in industrial multivariate identification and control [40]. Equation 7 is an ARX time series model with a single heater input and single temperature output with k index for the time step, i index for prediction horizon step, and adjustable parameters α, β, and γ.

$$T_{C1,k+1} = \sum_{i=1}^{n_\alpha} \alpha_i T_{C1,k-i+1} + \sum_{i=1}^{n_\beta} \beta_i Q_{1,k-i+1} + \gamma \tag{7}$$

With $n_\alpha = 3$ and $n_\beta = 2$ the time series model has 5 adjustable parameters and is shown in Equation 8. The ARX form uses prior temperature measurements to predict the next temperature in the series, $T_{C1,k+1}$, while the OE form uses prior temperature predictions to predict the next temperature in the sequence. The γ_1 value is adjusted to create an unbiased model prediction.

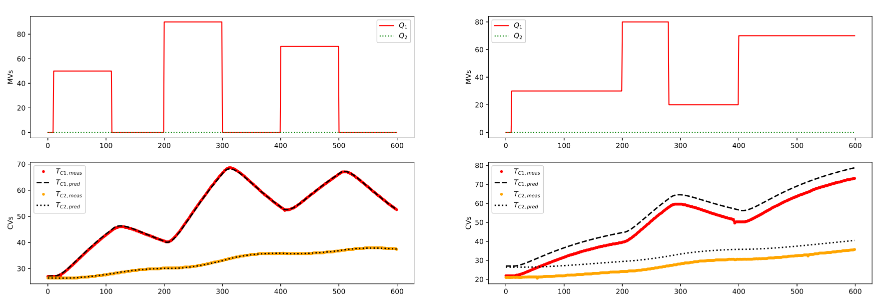
$$T_{C1,k+1} = \alpha_1\, T_{C1,k} + \alpha_2\, T_{C1,k-1} + \alpha_3\, T_{C1,k-2} + \beta_1\, Q_{1,k} + \beta_2\, Q_{1,k-1} + \gamma_1 \tag{8}$$

The OE identification form is used to reduce model bias. Equations 9a and 9b have multiple inputs and multiple outputs for the case when $n_\alpha = 2$ and $n_\beta = 1$.

$$T_{C1,k+1} = \alpha_{1,1}\, T_{C1,k} + \alpha_{2,1}\, T_{C1,k-1} + \beta_{1,1}\, Q_{1,k} + \beta_{1,2}\, Q_{2,k} + \gamma_1 \tag{9a}$$

$$T_{C2,k+1} = \alpha_{1,2}\, T_{C2,k} + \alpha_{2,2}\, T_{C2,k-1} + \beta_{2,1}\, Q_{1,k} + \beta_{2,2}\, Q_{2,k} + \gamma_2 \tag{9b}$$

An advantage of a linear time invariant (LTI) model such as SS, ARX, FIR, or OE is that little or no physics-based information is required to obtain a model prediction. When constraints are available, they are used to improve the identification [41]. The model fit to the step test data is shown in Figure 8a and the validation in Figure 8b.

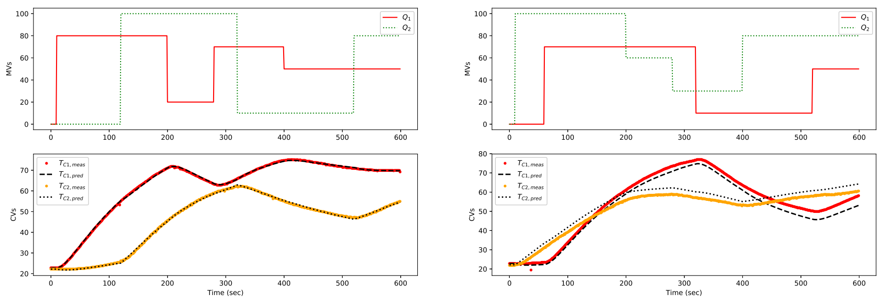


(a) Model Fitting

(b) Model Validation

Figure 8: Single Heater Step Response of the TCLab with Linear Time Series

There is insufficient data information to determine the β values associated with Q_2 because the value stays at zero for the duration of the test. A second test is conducted where the second heater is also adjusted to get a multivariate model from the step response data (see Figure 9).



(a) Model Fitting (b) Model Validation

Figure 9: Dual Heater Step Response of the TCLab with Linear Time Series

3.4. Hammerstein Model with Artificial Neural Network

A final modeling approach is a Hammerstein Model with an Artificial Neural Network (ANN) to predict the steady-state relationship between the heaters and temperatures and a linear dynamic block that translates the steady-state prediction into a dynamic prediction. The ANN is not trained directly on the dynamic data because a Recurrent Neural Network or Convolutional Neural Network is better suited for this type of predictive model and this is the topic of future work. A diagram of the model is shown in Figure 10.

The parameter weights, represented by arrows connecting each of the nodes, are adjusted to minimize a sum of squared error with 70 steady-state data points. The steady-state data points are obtained by setting random heater values between 0 and 80% for 5 min, recording the temperatures, and then adjusting the heater values to random levels for another data point. Although the system does not fully reach steady-state (2 τ or 95% of change), it is judged to be sufficiently close to fit the steady-state correlation. The linear dynamic part is approximated as a second-order dynamic relationship between the steady-state temperature outputs of the ANN and the dynamic response with τ_{p1}=140 *sec*

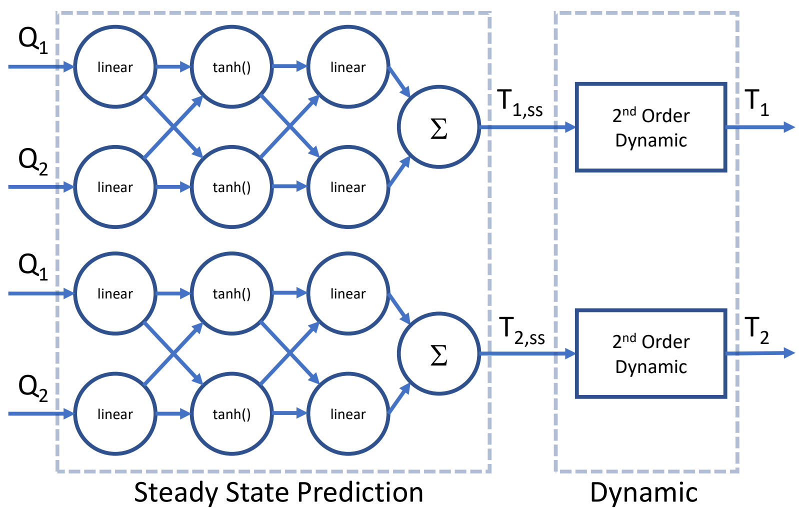
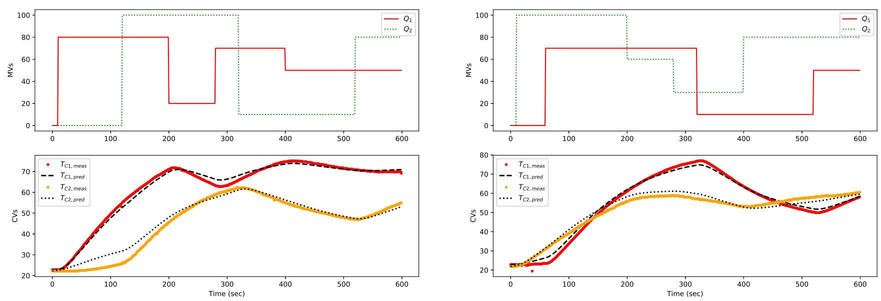


Figure 10: Architecture of the Hammerstein Model with a Steady-State Artificial Neural Network and Linear Dynamics.

and τ_{p2}=20 *sec*. The second order system approximates the time constant for the heater and temperature sensor with heat conduction between the two.



(a) Model Fitting

(b) Model Validation

Figure 11: Hammerstein Model Fitting and Validation with 2 Heaters

The fitting data is shown in Figure 11a and validation is shown in Figure 11b. Because the steady-state data is a different data set than the dynamic fitting data set, there is some offset between the predictions and data. There are many ANN forms and a future case study could investigate the use of convolutional or recurrent neural networks such as a network with LSTM (Long Short-Term Memory) nodes to combine the dynamic and steady-state predictions into one model.

3.5. Summary of Model Predictions with Validation

For model-based controllers, the choice of model depends on many factors such as computation speed, ability to extrapolate outside the training region, degree of nonlinearity, and others. Table 2 summarizes the model fit to data with the model regression and validation tests as an average sum of absolute error.

4. Benchmarking Closed-Loop PID Re-Tuning

The PID controller is a widely used basic regulatory control algorithm. PID control is important in chemical engineering processes as it plays a critical role

Table 2: Summary of Regression and Validation for Single Heater (SISO) and Dual Heater (MIMO) Tests

	Model Description	**Training**	**Validation**
SISO	Physics-based Lumped Parameter	0.20 ^oC	3.32 ^oC
SISO	First-order Plus Dead-time	0.41 ^oC	5.11 ^oC
SISO	Second Order ARX	0.18 ^oC	5.16 ^oC
SISO	Hammerstein with ANN and Linear Dynamics	3.83 ^oC	1.66 ^oC
MIMO	Physics-based Lumped Parameter	0.23 ^oC	0.70 ^oC
MIMO	Second Order ARX	0.26 ^oC	2.66 ^oC
MIMO	Hammerstein with ANN and Linear Dynamics	1.57 ^oC	1.55 ^oC

as a base regulatory layer foundation for advanced process control and optimization systems. PID performance varies greatly on the parameters obtained from tuning rules or heuristics [42, 43]. Control performance metrics such as minimum variance control are common assessments of performance [44, 45]. Methods such as Zeigler-Nichols closed-loop tuning requires sustained oscillation data to obtain an ultimate gain (K_u) and ultimate period (P_u) [46]. To avoid driving a process to the limitation of the stability region to obtain the sustained oscillation data, a relay method is introduced [47]. Tuning rules are a valuable starting point for further manual tuning but may not be optimized. Optimization-based PID tuning is another option with prior work in extremum seeking [48] algorithms, particle swarm [49, 50], and meta-heuristics such as genetic algorithms [51].

The objective of this closed-loop PID re-tuning is to demonstrate a TCLab benchmark that uses historical data to optimally re-tune a PID controller. An exhaustive search method visits all feasible combinations of the PI or PID parameters to find an optimal value of the objective function without converging to a local minimum for both output-error and input-move deviations. The method uses simulation of the physical TCLab PID controller by: (a) re-playing back the past or historical setpoint and load disturbances [52]; (b) allowing multiple, simultaneous and probability-weighted process models to be included in the simulations (i.e., multiple scenarios or situations each with specified probabilities) for robustness; (c) including multiple and simultaneous PID controller configuration formulations or even *ad hoc* controller designs; (d) specifying any type of performance objective function criteria i.e., simultaneously minimize the output-error and input-move variances, overshoot, etc. (e) adding stability rules in the search to cut-off unstable sections of the closed-loop operating space and (f) utilizing an indirect and constrained controller design technique [53].

The exhaustive search method is tested with the TCLab as a benchmark for closed-loop control performance. The TCLab produces the closed-loop operating data with IMC PID parameters and a selected setpoint change sequence. A deterministic parametric process model is then identified using an ARX structured model using the GEKKO dynamic optimization suite [54], estimating coefficients using a least-squares prediction-error objective function. Then, the exhaustive search method evaluates the range or domain of the different P, I, and/or D parameters. The best search objective function found provides the P, I, and/or D. The PID controller is then run again with the temperature control lab using the re-tuned PID parameters and the data recorded. There are many derivations of PID formula rooted in the original continuous equation [42]. For implementing PID controllers in modern digital control platforms such as a DCS (Distributed Control Systems) or PLC (Programmable Logic Controllers), two popular discrete forms are widely used in industry. One is the positional form (Equation 10a) and the other is the velocity form (Equation 10b), which are exchangeable.

$$OP_t = OP_{bias} + K_c \left(e_t + \frac{\Delta t}{\tau_I} \sum_{1}^{t} e_t + \tau_D \frac{PV_{t-1} - PV_t}{\Delta t} \right) \tag{10a}$$

$$OP_t = OP_{t-1} + K_c \left((e_t - e_{t-1}) + \frac{\Delta t}{\tau_I} e_t + \frac{\tau_D}{\Delta t} \left(PV_t - 2\,PV_{t-1} + PV_{t-2} \right) \right) \tag{10b}$$

where the output error $e_t = SP_t - PV_t$. Whereas the positional form calculates the controller output position (OP), the velocity form calculates the change in controller output ($\Delta OP = OP_t - OP_{t-1}$). Although the positional form is more straightforward to understand as the P, I, and D terms are directly translated from the original continuous form, the velocity form has several advantages from the convenience perspective such as no additional logic is required for anti-reset windup [55]. The positional form PI controller is used in this study while a prior study [53] used a PID controller in velocity form. In both cases, an ARX model is identified from closed-loop data. ARX and Box-Jenkins models have proven consistency in closed-loop identification [56, 57]. The potential PID tunings are re-played with the same past setpoint and load disturbance as in the process data (y_t) with $z_t = y_t - x_t$ where, x_t represents the ARX model output for time-step t. The load disturbance (z_t) is super-imposed on the ARX simulated process output during the search for optimal tuning parameters.

Two different types of objective functions are considered for PID tuning. The objective functions are a variation of the PID control performance index known as average IAE (Integral Absolute Error). The objective function consists of the output-error (OE) term, and the input movement (IM) term. The optimization solution of output error combined with input movement (or, rate of change) has been analytically derived and investigated in [58] and is the simplest form of move suppression. These multi-objective functions can be express in two different ways. One is Archimedean and the other is the lexicographic form (or goal programming) as shown in Equations 11 and 12, respectively.

$$\min_{K_c, \tau_I, \tau_D} J = \frac{\sum_{i=1}^{t} \left(w_{OE} \left\| SP_i - x_i \right\|_n + w_{IM} \left\| OP_i - OP_{i-1} \right\|_n \right)}{t} \tag{11}$$

where n is the norm w is the weighting factor for each term in the objective function denoted as OE for output error and IM for input movement.

$$\min_{K_c,\tau_I,\tau_D} J = \frac{\sum_{i=1}^{t} \left(\|SP_i - x_i\|_n\right) \text{Subject to} \|OP_i - OP_{i-1}\|_n \leq UB_{IM}}{t} \tag{12}$$

where UB is the upper bound of the input movement (IM) which may be initially set by the centroid PID performance. Either the Archimedean or lexicographic form of the objective function can be used for PID controller tuning. In terms of convenience, the lexicographic form is easier to use because it requires one user input parameter, UB_{IM}, as opposed to the Archimedean form that requires two weighting factors on both OE and IM terms. One simplification of the Archimedean form is to reduce the weighting factors to one by dividing the objective by w_{OE}.

4.1. TCLab Benchmark Validation

The first step of the validation is to collect the closed-loop operation data and identify the ARX model parameters for identifying the ARX model. The setpoint is changed from ambient temperature at the initial steady-state condition to 50 ^oC, 40 ^oC, and then to 60 ^oC. The ranges of K_c and τ_I are evaluated through the ARX model that includes the same setpoint sequences and load disturbance. The performance objective functions for each $K_c$and τ_I incremental combination are also calculated and stored. The ℓ_1-norm objective function in the Archimedean form is chosen for the test with weighting factors $w_{OE} = 1$ and $w_{IM} = 0.5$. The K_c and τ_I combination that gives a minimum value of objective function is then chosen as optimal PID tuning. The initial K_c and τ_I are from the FOPDT model in Section 3.2 and IMC aggressive tuning with $K_c = 5.74 \frac{\%}{^oC}$ and $\tau_I = 175.2\, sec$. Optimized values are $K_c = 10.0 \frac{\%}{^oC}$ and $\tau_I = 55.0\, sec$ as shown as the minimum value of the objective function contour map (see Figure 12).

The objective function surface is not smooth because of the load disturbances that are replayed with every PID parameter combination. Figure 13 shows the measured temperature and ARX model response for both the original and

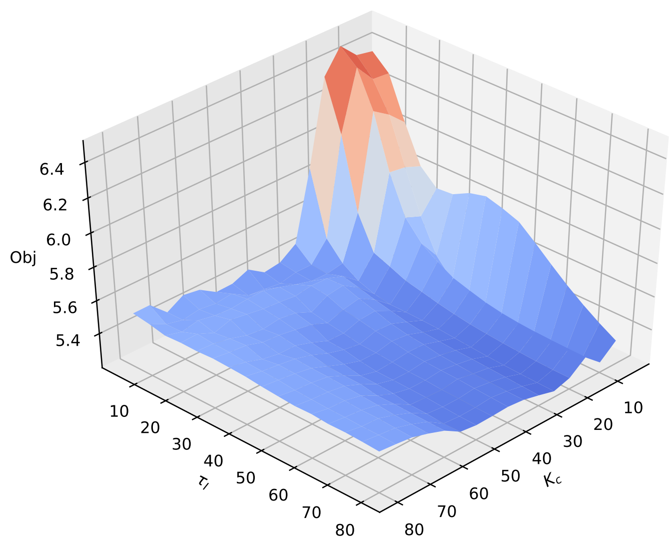


Figure 12: Average Integral Absolute Error (IAE) with K_c and τ_I PID Parameters.

optimized response. The validation of the optimal tuning parameter is displayed as well.

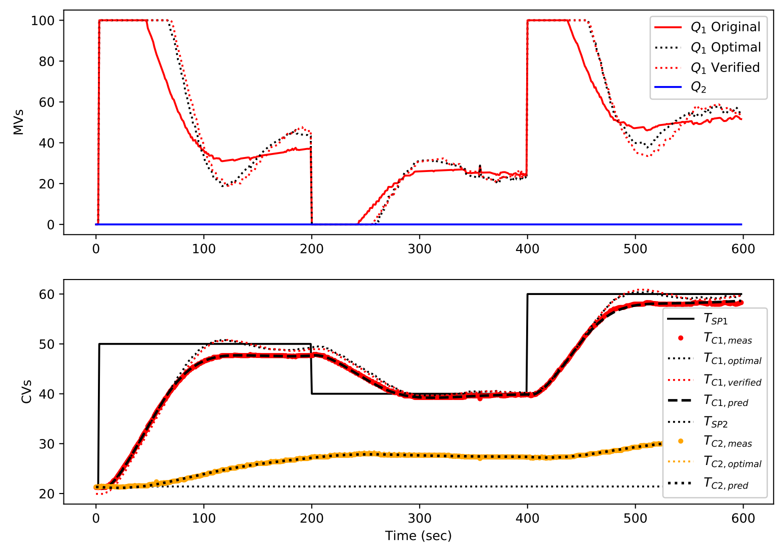


Figure 13: ARX Simulated and TCLab Validated Performance Improvement of 5.4%.

The average IAE objective function is 6.09 with IMC tuning and 5.76 with optimized parameters, an improvement of 5.4%. The PID improvement is simulated with the ARX model and validated with closed-loop data from the Arduino TCLab.

5. Multivariate Control Benchmark

Model predictive control (MPC) with the physics-based model, time series linear model (ARX), and Hammerstein ANN model quantify multivariate control performance. Additional models in MPC or multivariate control strategies are tested with the TCLab. This section shows benchmark performance with three popular methods for multivariate control that range from linear to nonlinear and empirical to physics-based.

An ℓ_1-norm objective function gives a target region for the temperature range, rather than one specific target value. Equation 13 shows the ℓ_1-norm control formulation used in this work for model predictive control (MPC).

$$
\begin{aligned}
\min_{x,CV,MV} \quad & J = w_{hi}^T e_{hi} + w_{lo}^T e_{lo} + \Delta Q^T c_{\Delta Q} \\
\text{s.t.} \quad & 0 = f\left(\tfrac{dT}{dt}, T, Q\right) \\
& e_{hi} \geq T - T_{hi} \\
& e_{lo} \geq T_{lo} - T
\end{aligned}
\tag{13}
$$

where J is the objective function, T is the temperature, Q is the heater, w_{lo} and w_{hi} are penalty matrices for solutions outside the target temperature region. Slack variables e_{lo} and e_{hi} are the error of the dead-band low and high limits, respectively. Parameter $c_{\Delta Q}$ is a move suppression factor. The function f is an open-equation set of model equations that include T, Q, and time derivatives of T. The demand targets T_{lo} and T_{hi} define lower and upper target limits for temperature as shown in Figure 14.

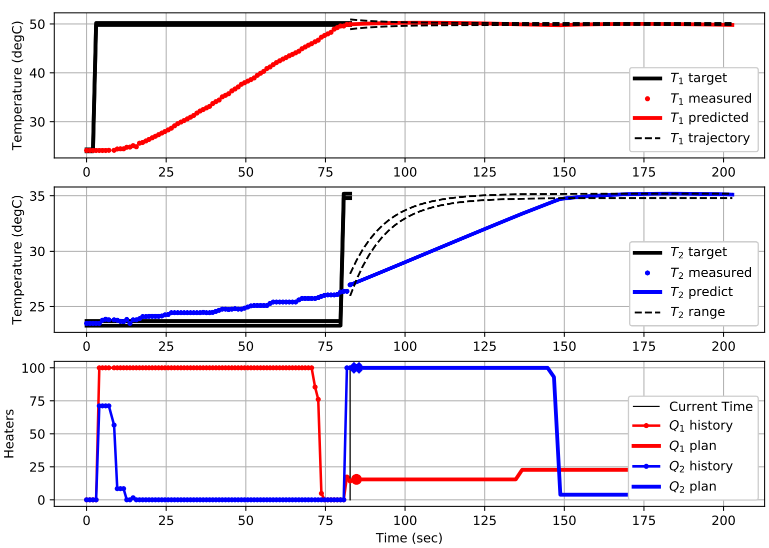


Figure 14: MPC with ARX Model at Cycle 81

At cycle 81, temperature 1 has just reached the target temperature setpoint of 50 ^oC after heater 1 is ramped down from 100% to 0% at 10-15 sec prior to reaching the setpoint. The model predictive controller anticipates the continued rise in temperature and turns the heater off for a period of 5 seconds before returning to a baseline heater value to maintain the 50 ^oC setpoint. The model also anticipates the increase in temperature 2 due to the setpoint change to 35 ^oC at cycle 80. The reference trajectory with time constant τ=10 sec gives a guide for the fastest that the temperature should approach the new setpoint. The setpoint has a $\pm$0.2 ^oC range with a $\pm$1.0 ^oC larger opening at the beginning for less MV movement for near-term adjustments. The underlying ARX time-series model coordinates the MV movements to meet both setpoints considering multivariate effects.

6. Benchmarking Model Predictive Control

The multivariate models developed in Sections 3.1, 3.3, and 3.4 are compared in MPC. The MPC uses an ℓ_1-norm objective with a temperature dead-band of $\pm$0.2 ^oC for $T_{hi} - T_{sp}$, $T_{lo} - T_{sp}$ and a first-order reference trajectory of 10 sec for setpoint changes. The move suppression factor $c_{\Delta Q}$ is set to 0.1, the weights w_{hi} and w_{lo} are set to 20.0, and the control and prediction horizon are 60 seconds. The linear ARX model has a cycle time of 1 second while the nonlinear physics-based and Hammerstein applications are re-computed every 2 seconds. The longer cycle time is required to enable all steps of data retrieval, model update, re-calculation of optimal move plan, retrieval of first step, and insertion into the process. Table 3 is a numeric comparison of the methods with quantified IAE rate ($^oC/sec$) and Integral Average Move rate ($\%/sec$) for the heater adjustments. Another common performance metric is a minimum variance as applied to multivariate control systems [59, 60]. Rate-based values are shown in this case because of the differing cycle times between the applications.

The benchmark results show that all models perform equally well in terms of the control performance (11.4-11.6 ^oC/sec) as shown in Figures 15 to 17.

Table 3: Summary of Model Predictive Control Methods

Model Description	IAE Avg Rate (CVs)	IAE Avg Rate (ΔMVs)
Physics-based Lumped Parameter	11.5 $^oC/sec$	2.0 $\%/sec$
Second Order ARX	11.6 $^oC/sec$	3.3 $\%/sec$
Hammerstein with ANN and Linear Dynamics	11.4 $^oC/sec$	2.5 $\%/sec$

In all cases, T_1 is not able to reach the setpoint of 30^oC between 160-320 sec because of insufficient cooling rate when Q_1 is off. The ARX model has the highest MV movement (3.3 %/sec) and the physics-based model has the lowest MV movement even with rapid fluctuations on Q_2 during the first setpoint change at $t = 105$ sec. The values for MPC are more than the PID control performance metric because there are two CVs and two MVs that accumulate error approximately twice as fast and with more frequent setpoint changes.

The physics-based model has the potential to extrapolate to new operating conditions without retuning. A physics-based MPC has the disadvantage of relative difficulty in developing the model equations for complex systems. There is also a potential for solver convergence problems if the physics-based model is high nonlinear or does not have a suitable initial guess. This is not the case for the TCLab where an approximate lumped-parameter model is an accurate representation of the physical system. One drawback for the physics-based MPC is that it cannot run at 1 sec cycles but does solve within a 2 second interval for a 60 sec prediction horizon. The ARX control performance is shown in Figure 16.

The ARX MPC has the fastest cycle time (1 sec versus 2 sec) so that it can respond more quickly to disturbances or setpoint changes. Because it is a linear model, the cycle time can be faster (up to 5 Hz) due to reduced computing time. The disadvantage of the ARX MPC is that it is a linear representation of the slightly nonlinear TCLab. This requires re-adjustment of the move plan and

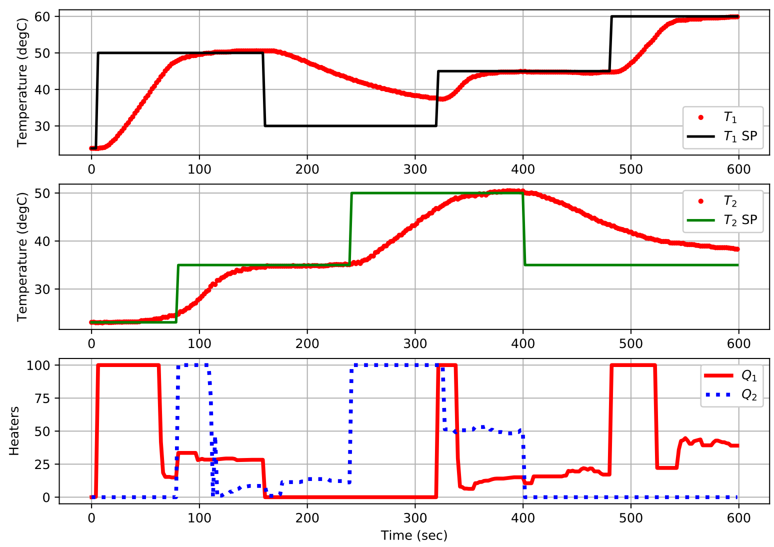


Figure 15: MPC with Physics-based Model

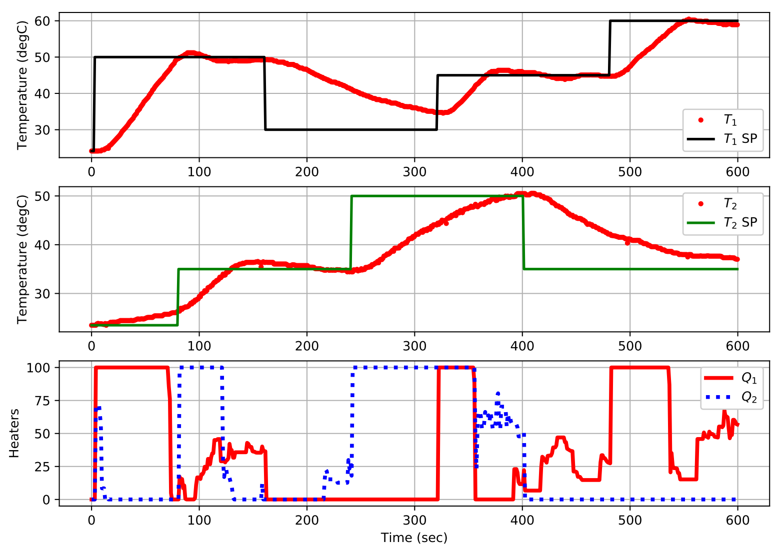


Figure 16: MPC with ARX Time Series Model

increased cycling due to model mismatch. The ARX MPC has slight overshoot due to the underestimation of process gain that leads to overly aggressive MV movement as shown in Figure 17.

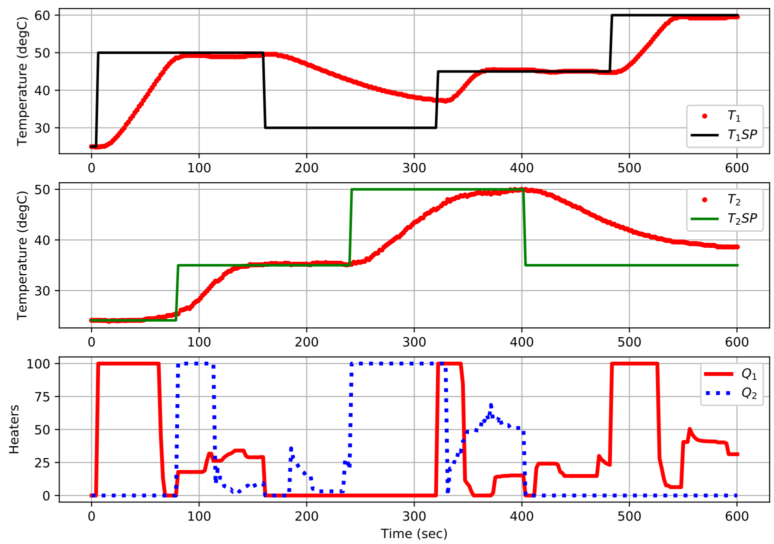


Figure 17: MPC with Hammerstein ANN Model

The Hammerstein MPC has the potential to excel in situations where the process is highly nonlinear and there is not a suitable physics-based representation of the process. Like the physics-based MPC, it requires a slower 2 sec cycle time to meet the real-time constraint. Unlike the physics-based MPC, it is not expected to perform well when used outside of the training domain. To facilitate the comparison, a repository of source code and Arduino firmware `https://github.com/APMonitor/arduino` is available with all the examples from this paper.

7. Conclusion and Future Work

The benchmark studies included in this paper are a sampling of common modeling and control methods that are quantified with the TCLab shield and an Arduino microcontroller. The temperature response is modeled with four approaches: physics-based, FOPDT, ARX, and Hammerstein ANN with linear dynamics. Separate data sets are used for training and validation. The objective of the modeling is to create automatic controllers with PID and MPC. A PID optimal tuning case study uses an exhaustive search as a straightforward method for closed-loop retuning to improve performance by 5.4%. The optimal PID parameters are selected by replaying past setpoint and load disturbances where the residuals of estimation are considered as the unmeasured load disturbances. A second study is the application of the three multivariate models in MPC with varying degrees of nonlinearity and physics-based foundation.

This study presents a sample of potential modeling and control applications that are quantified with the TCLab hardware benchmark. There are additional potential applications for evaluating methods in estimation, data reconciliation, machine learning, classification, fault detection, anomaly detection, disturbance identification and rejection, integration of control and scheduling, mixed integer systems, stability analysis, explicit MPC, and others. Because each TCLab device is slightly different, benchmark evaluations are performed on the same device and with similar ambient conditions. The TCLab is an accessible hardware platform for benchmarking models and closed-loop performance with real data.

Acknowledgments

This article is prepared for a Special Issue in honor of Tom Edgar's 75^{th} birthday and to celebrate his lifetime of accomplishments and leadership in the area of Process Systems Engineering. This work is influenced by his work with energy systems, optimization, control engineering education, and advancements

in model-based control among many other areas of contribution. We are grateful for his contributions and continued service to the community.

Appendix A. Software Interface to TCLab

Two parts to the software interface are the firmware that runs on the Arduino Leonardo and the serial interface to interpret and command the TCLab. An important part of making the benchmark accessible is to create an interface to software (MATLAB, Simulink, and Python) where control algorithms are developed but also provide information for interfaces to other software platforms. There is an Arduino Support Package for MATLAB and Simulink from MathWorks that automatically loads firmware onto the Arduino when it is connected for the first time. The Arduino firmware for Python is an *ino* file that is augmented with additional sections to compile as *cpp* code with a gcc compiler through the Arduino IDE. The TCLab is pre-loaded with the Python interface firmware.

Listing 1: MATLAB Commands to Adjust Heaters and Display Temperatures

```
clear all
% include tclab.m
tclab;
disp('Turn on Heaters and LED')
h1(30);  h2(60); led(1);
pause(10)
disp('Display Temperatures')
disp(T1C())
disp(T2C())
h1(0);  h2(0); led(0);
```

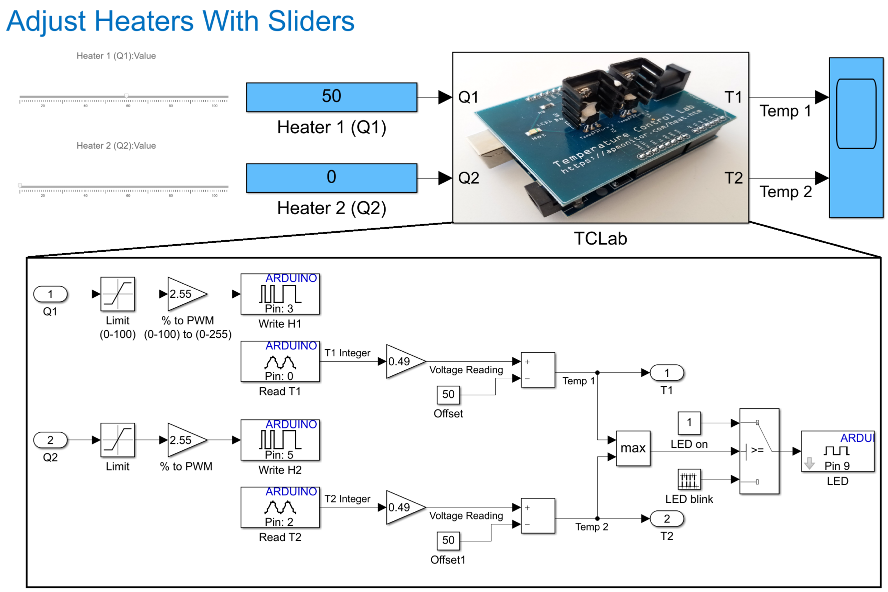


Figure A.18: Simulink Interface with Manual Sliders for Heater Levels.

Listing 2: Python Commands to Adjust Heaters and Display Temperatures

```python
import tclab  # pip install tclab
import time
# Connect to Arduino
a = tclab.TCLab()
print('Turn on Heaters and LED')
a.Q1(30.0); a.Q2(60.0); a.LED(100)
time.sleep(10.0)
print('Display Temperatures')
print(a.T1)
print(a.T2)
a.close()
```

References

References

[1] A. F. Villaverde, D. Henriques, K. Smallbone, S. Bongard, J. Schmid, D. Cicin-Sain, A. Crombach, J. Saez-Rodriguez, K. Mauch, E. Balsa-Canto, et al., Biopredyn-bench: a suite of benchmark problems for dynamic modelling in systems biology, BMC systems biology 9 (1) (2015) 8.

[2] N. R. Lewis, J. D. Hedengren, E. L. Haseltine, Hybrid dynamic optimization methods for systems biology with efficient sensitivities, Processes 3 (3) (2015) 701. `doi:10.3390/pr3030701`.
URL `http://www.mdpi.com/2227-9717/3/3/701`

[3] L. Peters, R. Arts, G. Brouwer, C. Geel, S. Cullick, R. J. Lorentzen, Y. Chen, N. Dunlop, F. C. Vossepoel, R. Xu, et al., Results of the Brugge benchmark study for flooding optimization and history matching, SPE Reservoir Evaluation & Engineering 13 (03) (2010) 391–405.

[4] V. P. Singh, A. Cavanagh, H. Hansen, B. Nazarian, M. Iding, P. S. Ringrose, et al., Reservoir modeling of CO2 plume behavior calibrated against monitoring data from Sleipner, Norway, in: SPE annual technical conference and exhibition, Society of Petroleum Engineers, 2010.

[5] R. W. Rwechungura, E. Suwartadi, M. Dadashpour, J. Kleppe, B. A. Foss, et al., The Norne field case-a unique comparative case study, in: SPE Intelligent Energy Conference and Exhibition, Society of Petroleum Engineers, 2010.

[6] J. Udy, B. Hansen, S. Maddux, D. Petersen, S. Heilner, K. Stevens, D. Lignell, J. D. Hedengren, Review of field development optimization of waterflooding, EOR, and well placement focusing on history matching and optimization algorithms, Processes 5 (3) (2017) 34.

[7] A. N. Eaton, L. D. Beal, S. D. Thorpe, C. B. Hubbell, J. D. Hedengren, R. Nybø, M. Aghito, Real time model identification using multi-fidelity

models in managed pressure drilling, Computers & Chemical Engineering 97 (2017) 76–84.

[8] R. Asgharzadeh Shishavan, C. Hubbell, H. Perez, J. Hedengren, D. S. Pixton, et al., Combined rate of penetration and pressure regulation for drilling optimization using high speed telemetry, SPE Drilling & Completion Journal 1 (SPE-170275-MS) (2015) 17–26.

[9] M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global optimization problems, arXiv preprint arXiv:1308.4008.

[10] R. V. Rao, V. J. Savsani, D. Vakharia, Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems, Information sciences 183 (1) (2012) 1–15.

[11] S. M. Safdarnejad, J. D. Hedengren, N. R. Lewis, E. L. Haseltine, Initialization strategies for optimization of dynamic systems, Computers & Chemical Engineering 78 (2015) 39 – 50. `doi:10.1016/j.compchemeng.2015.04.016`.
URL `http://www.sciencedirect.com/science/article/pii/S0098135415001179`

[12] R. Huang, Nonlinear model predictive control and dynamic real time optimization for large-scale processes, Ph.D. thesis, Carnegie Mellon University (12 2010).

[13] M. Hehn, R. Ritz, R. D'Andrea, Performance benchmarking of quadrotor systems using time-optimal control, Autonomous Robots 33 (1-2) (2012) 69–88.

[14] W. Chen, Y. Ren, G. Zhang, L. T. Biegler, A simultaneous approach for singular optimal control based on partial moving grid, AIChE Journal 65 (6).

[15] L. D. Beal, D. Petersen, D. Grimsman, S. Warnick, J. D. Hedengren, Integrated scheduling and control in discrete-time with dynamic parameters

and constraints, Computers & Chemical Engineering 115 (2018) 361 – 376. doi:https://doi.org/10.1016/j.compchemeng.2018.04.010.
URL http://www.sciencedirect.com/science/article/pii/S0098135418303120

[16] L. D. R. Beal, D. Petersen, G. Pila, B. Davis, S. Warnick, J. D. Hedengren, Economic benefit from progressive integration of scheduling and control for continuous chemical processes, Processes 5 (4).

[17] D. Petersen, L. D. R. Beal, D. Prestwich, S. Warnick, J. D. Hedengren, Combined noncyclic scheduling and advanced control for continuous chemical processes, Processes 5 (4).

[18] Y. Nie, L. T. Biegler, C. M. Villa, J. M. Wassick, Discrete Time Formulation for the Integration of Scheduling and Dynamic Optimization, Industrial & Engineering Chemistry Research 54 (16) (2015) 4303–4315. doi:10.1021/ie502960p.
URL http://pubs.acs.org/doi/abs/10.1021/ie502960p

[19] P. F. Odgaard, J. Stoustrup, M. Kinnaert, Fault tolerant control of wind turbines–a benchmark model, IFAC Proceedings Volumes 42 (8) (2009) 155–160.

[20] G. M. Kopanos, C. A. Méndez, L. Puigjaner, MIP-based decomposition strategies for large-scale scheduling problems in multiproduct multistage batch plants: A benchmark scheduling problem of the pharmaceutical industry, European Journal of Operational Research 207 (2) (2010) 644–655. doi:10.1016/j.ejor.2010.06.002.
URL http://linkinghub.elsevier.com/retrieve/pii/S037722171000408X

[21] D. Saygin, E. Worrell, M. K. Patel, D. Gielen, Benchmarking the energy use of energy-intensive industries in industrialized and in developing countries, Energy 36 (11) (2011) 6661–6673.

[22] L. D. Beal, J. Park, D. Petersen, S. Warnick, J. D. Hedengren, Combined model predictive control and scheduling with dominant time constant compensation, Computers & Chemical Engineering 104 (2017) 271–282.

[23] M. Baldea, I. Harjunkoski, Integrated production scheduling and process control: A systematic review, Computers & Chemical Engineering 71 (2014) 377–390. doi:10.1016/j.compchemeng.2014.09.002.

[24] J. Kelly, J. Hedengren, A steady-state detection (SSD) algorithm to detect non-stationary drifts in processes, Journal of Process Control 23 (3) (2013) 326–331.

[25] N. L. Ricker, J. Lee, Nonlinear model predictive control of the Tennessee Eastman challenge process, Computers & Chemical Engineering 19 (9) (1995) 961–981.

[26] A. Bathelt, N. L. Ricker, M. Jelali, Revision of the Tennessee Eastman process model, IFAC-PapersOnLine 48 (8) (2015) 309–314.

[27] N. I. Vitzilaios, N. C. Tsourveloudis, Test bed for unmanned helicopters' performance evaluation and benchmarking, in: IEEE/RSJ IROS 2008 Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems, Citeseer, 2008.

[28] A. Cardoso, V. Sousa, P. Gil, Demonstration of a remote control laboratory to support teaching in control engineering subjects, IFAC-PapersOnLine 49 (6) (2016) 226–229.

[29] P. K. Singh, S. Bhanot, H. K. Mohanta, V. Bansal, Self-tuned fuzzy logic control of a ph neutralization process, in: 2015 21st International Conference on Automation and Computing (ICAC), IEEE, 2015, pp. 1–6.

[30] I. Alvarado, D. Limon, D. M. De La Peña, J. Maestre, M. Ridao, H. Scheu, W. Marquardt, R. Negenborn, B. De Schutter, F. Valencia, A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark, Journal of Process Control 21 (5) (2011) 800–815.

[31] V. Kirubakaran, T. Radhakrishnan, N. Sivakumaran, Distributed multi-parametric model predictive control design for a quadruple tank process, Measurement 47 (2014) 841–854.

[32] Y. Alipouri, J. Poshtan, Optimal controller design using discrete linear model for a four tank benchmark process, ISA transactions 52 (5) (2013) 644–651.

[33] B. Spivey, J. Hedengren, T. Edgar, Constrained nonlinear estimation for industrial process fouling, Industrial & Engineering Chemistry Research 49 (17) (2010) 7824–7831.

[34] V. M. Zavala, L. T. Biegler, Optimization-based strategies for the operation of low-density polyethylene tubular reactors: nonlinear model predictive control, Computers & Chemical Engineering 33 (10) (2009) 1735–1746.

[35] J. Rossiter, S. Pope, B. L. Jones, J. Hedengren, Evaluation and demonstration of take home laboratory kit, IFAC-PapersOnLine 52 (9) (2019) 56–61.

[36] P. Oliveira, J. Hedengren, An APMonitor temperature lab PID control experiment for undergraduate students, in: 24th IEEE Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, IEEE, 2019, pp. 790–797.

[37] F. G. Shinskey, Process control: as taught vs as practiced, Industrial & engineering chemistry research 41 (16) (2002) 3745–3750.

[38] T. F. Edgar, B. A. Ogunnaike, J. J. Downs, K. R. Muske, B. W. Bequette, Renovating the undergraduate process control course, Computers & chemical engineering 30 (10-12) (2006) 1749–1762.

[39] J. Alford, T. Edgar, Preparing chemical engineering students for industry, Chemical Engineering Progress 113 (11) (2017) 25–28.

[40] S. J. Qin, T. A. Badgwell, A survey of industrial model predictive control technology, Control Engineering Practice 11 (7) (2003) 733–764.

[41] J. Udy, L. Blackburn, J. D. Hedengren, M. Darby, Reduced order modeling for reservoir injection optimization and forecasting, in: Proceedings of the FOCAPO/CPC Conference, Tuscon, AZ, USA, 2017, pp. 8–12.

[42] K. J. Åström, T. Hägglund, PID controllers: theory, design, and tuning, Vol. 2, Instrument society of America Research Triangle Park, NC, 1995.

[43] B.-S. Ko, T. F. Edgar, Assessment of achievable PI control performance for linear processes with dead time, in: Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207), Vol. 3, IEEE, 1998, pp. 1548–1552.

[44] S. J. Qin, Control performance monitoring—a review and assessment, Computers & Chemical Engineering 23 (2) (1998) 173–186.

[45] B.-S. Ko, T. F. Edgar, PID control performance assessment: The single-loop case, AIChE Journal 50 (6) (2004) 1211–1218.

[46] J. G. Ziegler, N. B. Nichols, Optimum settings for automatic controllers, trans. ASME 64 (11).

[47] K. J. Åström, T. Hägglund, Automatic tuning of simple regulators with specifications on phase and amplitude margins, Automatica 20 (5) (1984) 645–651.

[48] N. J. Killingsworth, M. Krstic, PID tuning using extremum seeking: online, model-free performance optimization, IEEE control systems magazine 26 (1) (2006) 70–79.

[49] Z.-L. Gaing, A particle swarm optimization approach for optimum design of PID controller in AVR system, IEEE transactions on energy conversion 19 (2) (2004) 384–391.

[50] M. I. Solihin, L. F. Tack, M. L. Kean, Tuning of PID controller using particle swarm optimization (PSO), International Journal on Advanced Science, Engineering and Information Technology 1 (4) (2011) 458–461.

[51] B. Mohanty, S. Panda, P. Hota, Controller parameters tuning of differential evolution algorithm and its application to load frequency control of multi-source power system, International journal of electrical power & energy systems 54 (2014) 77–85.

[52] J. D. Kelly, Tuning digital PI controllers for minimal variance in manipulated input moves applied to imbalanced systems with delay, The Canadian Journal of Chemical Engineering 76 (5) (1998) 967–974.

[53] J. Park, C. Patterson, J. Kelly, J. Hedengren, Closed-loop PID re-tuning in a digital twin by re-playing past setpoint and load disturbance data, in: 2019 (AIChE) Spring Meeting, New Orleans, LA, AIChE, 2019, pp. 1–6.

[54] L. Beal, D. Hill, R. Martin, J. Hedengren, Gekko optimization suite, Processes 6 (8) (2018) 106.

[55] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, F. J. Doyle III, Process dynamics and control, John Wiley & Sons, 2010.

[56] A. Voda, I. Landau, Multi-step closed loop identification and control design procedure-applications, IFAC Proceedings Volumes 27 (8) (1994) 1543–1548.

[57] E. Jahanshahi, S. Skogestad, Closed-loop model identification and PID/PI tuning for robust anti-slug control, IFAC Proceedings Volumes 46 (32) (2013) 233–240.

[58] R. Tchamna, M. Lee, Analytical design of an industrial two-term controller for optimal regulatory control of open-loop unstable processes under operational constraints, ISA transactions 72 (2018) 66–76.

[59] B.-S. Ko, T. F. Edgar, Performance assessment of multivariable feedback control systems, Automatica 37 (6) (2001) 899–905.

[60] C. A. Harrison, S. J. Qin, Minimum variance performance map for constrained model predictive control, Journal of Process Control 19 (7) (2009) 1199–1204.