



---

Faculty Publications

---

2020-04-06

## Benchmark Temperature Microcontroller for Process Dynamics and Control

Junho Park  
*Brigham Young University*

Ronald Abraham Martin  
*Brigham Young University*

Jeffrey Kelly  
*Brigham Young University*

John Hedengren  
*Brigham Young University*, [john.hedengren@byu.edu](mailto:john.hedengren@byu.edu)

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>

 Part of the Chemical Engineering Commons

### Original Publication Citation

<https://www.sciencedirect.com/science/article/abs/pii/S0098135419310129>

---

### BYU ScholarsArchive Citation

Park, Junho; Martin, Ronald Abraham; Kelly, Jeffrey; and Hedengren, John, "Benchmark Temperature Microcontroller for Process Dynamics and Control" (2020). *Faculty Publications*. 4166.  
<https://scholarsarchive.byu.edu/facpub/4166>

# Benchmark Temperature Microcontroller for Process Dynamics and Control

Junho Park<sup>a</sup>, R. Abraham Martin<sup>a</sup>, Jeffrey D. Kelly<sup>b</sup>, John D. Hedengren<sup>a,\*</sup>

<sup>a</sup>*Department of Chemical Engineering, Brigham Young University, Provo, Utah, USA*

<sup>b</sup>*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)

<sup>1</sup> **1. Introduction**

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

<sup>9</sup> There are many standard benchmark models for testing the performance  
<sup>10</sup> of estimation and control methods in chemical process control. Some of these  
<sup>11</sup> include a continuously stirred tank reactor (CSTR) with a single exothermic re-  
<sup>12</sup> action [22, 23, 24]. One of the most commonly cited models in chemical process  
<sup>13</sup> control is the Tennessee Eastman Process [25, 26]. The Tennessee Eastman Pro-  
<sup>14</sup> cess encapsulates valve characteristics, measurement noise, process nonlinearity,  
<sup>15</sup> and complex interactions between processing units for chemical manufacture.

<sup>16</sup> Besides simulation, there are standard hardware benchmarks for evaluat-  
<sup>17</sup> ing control performance such as UAV control [27], process control education  
<sup>18</sup> modules [28, 29], and quadruple tank level control [30, 31, 32]. There also  
<sup>19</sup> many studies where the authors build a unique test system or implement con-  
<sup>20</sup> trol on an industrial process [33, 34] and demonstrate various control methods.  
<sup>21</sup> However, hardware benchmarks may be difficult to reproduce or the industrial  
<sup>22</sup> process may be unavailable for independent researchers to also obtain data or  
<sup>23</sup> test methods in closed-loop.

<sup>24</sup> The purpose of this paper is to demonstrate a standard hardware benchmark  
<sup>25</sup> for control methods with a micro-controller temperature control device. This  
<sup>26</sup> Temperature Control Lab (TCLab) is used as an education module for courses  
<sup>27</sup> in process dynamics and control [35, 36]. As many have noted in assessments of  
<sup>28</sup> process control education, there is a need to give students realistic and hands-  
<sup>29</sup> on experiences with process control [37, 38, 39]. Industry desires foundational  
<sup>30</sup> and practical knowledge of control engineering concepts that are reinforced with

31 physical modules. Because the TCLab, as an educational module, has wide dis-  
32 tribution to universities and industrial practitioners (3000 units), it has potential  
33 as a standard hardware benchmark for control engineering studies. Section 2  
34 gives details of the device to enable replication of the TCLab.

35 **2. Temperature Control Lab Device**

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

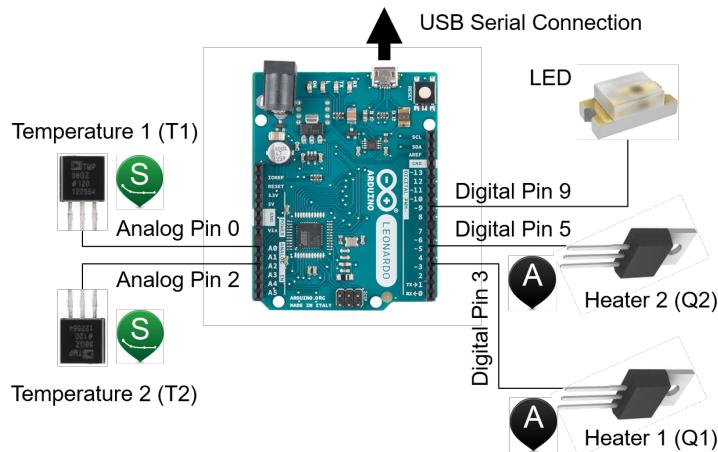
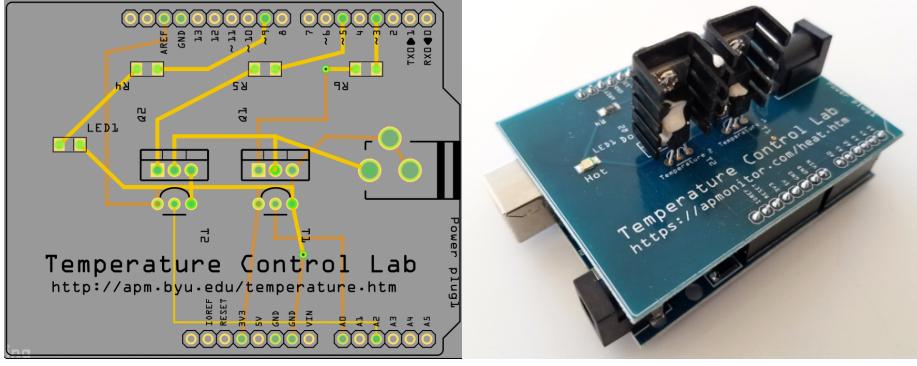


Figure 1: Temperature sensors and heater transistors with connections to an Arduino Leonardo.

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

58     As a safety and equipment protection precaution, the Arduino micro-controllers  
59    come pre-programmed to shut off the heaters if the temperature rises above  
60     $100^{\circ}C$ . The heaters are powered by a 5V 2A power supply for a maximum  
61    power output of 10 W. A 20 AWG (American Wire Gauge) power cable reduces  
62    the power dissipation compared to standard 24 AWG power cables with a barrel  
63    jack connector. A USB cable connects the Arduino to a computer for serial data  
64    communication. One TIP31C heater and one TMP36GZ sensor are connected  
65    to each other and with a thermal heat sink attached to the TIP31C transistor.  
66    The two heater units are placed in proximity to each other to transfer heat by  
67    convection and thermal radiation.

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



(a) TCLab Printed Circuit Board Layout

(b) TCLab Device

Figure 2: Temperature Control Lab Design

### 74     3. Temperature Response Models

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

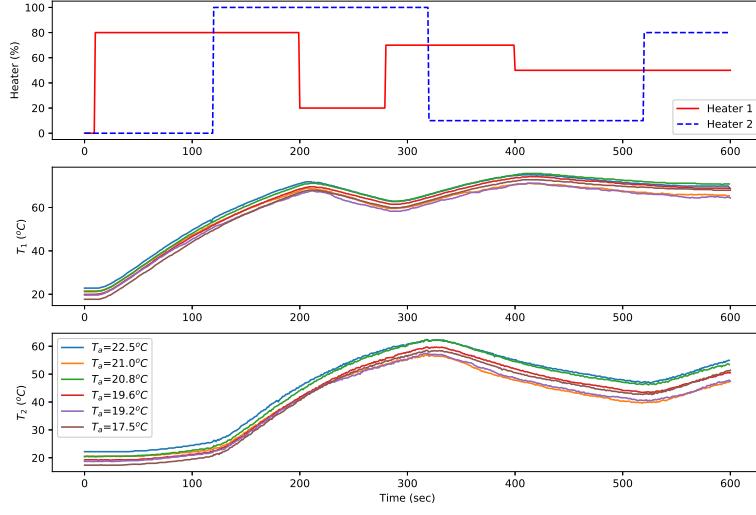


Figure 3: Variations in ambient temperature influence the temperature profiles

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  $1^{\circ}\text{C}$  equates to a  $0.928 \pm 0.033^{\circ}\text{C}$  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

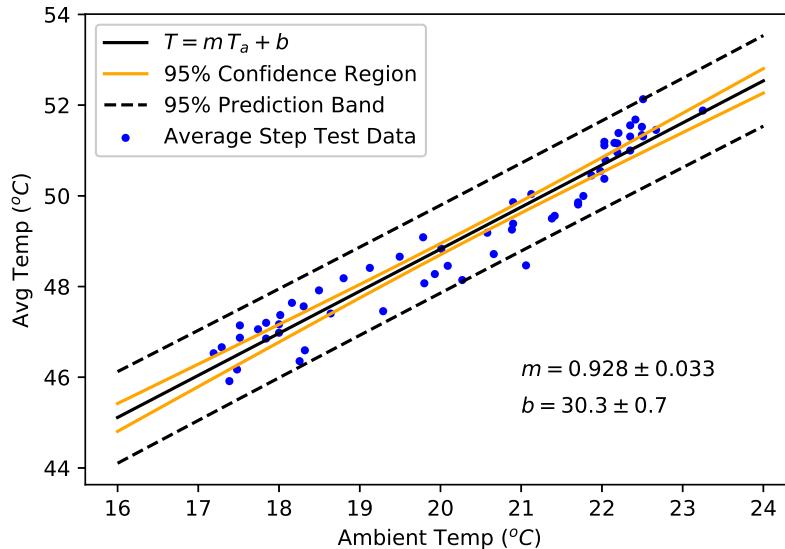


Figure 4: Correlation of ambient temperature to average temperature during 60 step tests (10 min each)

<sup>108</sup> analysis (FEA) that tracks the temperature distribution throughout the heat  
<sup>109</sup> sink and loss to the environment as shown in Figure 5.

<sup>110</sup> Details of the FEA simulation are not provided here but do provide a confirmation  
<sup>111</sup> that the temperature distribution is sufficiently uniform ( $< 3^{\circ}\text{C}$ ) for a  
<sup>112</sup> lumped parameter assumption. The lumped parameter model assumes that the  
<sup>113</sup> heaters ( $T_{H1}$  and  $T_{H2}$ ) and temperature sensors ( $T_{C1}$  and  $T_{C2}$ ) have a uniform  
<sup>114</sup> temperature. The temperature sensors ( $T_{C1}$  and  $T_{C2}$ ) have a small thermal  
<sup>115</sup> mass and surface area and temperature changes are driven by heat conduction  
<sup>116</sup> from the heaters ( $T_{H1}$  and  $T_{H2}$ ) where they are attached with thermal epoxy.  
<sup>117</sup> Parameters of the lumped parameter model are given in Table 1.

<sup>118</sup> The dynamic input power to each transistor and the temperature sensed  
<sup>119</sup> by each thermistor is developed with energy balance equations (Equations 1-4)  
<sup>120</sup> that account for convection, conduction, and thermal radiation. The amount  
<sup>121</sup> of convective heat transfer from heater 1 to heater 2 is given by  $Q_{C12} =$

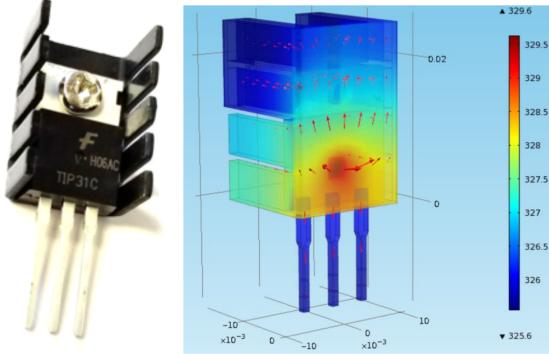


Figure 5: Finite Element Analysis of the Dynamic Temperature Response.

Table 1: Lumped Parameters from Physics-based Model

Quantity	Value
Initial temperature ( $T_0$ )	296.15 K ( $23^\circ C$ )
Ambient temperature ( $T_\infty$ )	296.15 K ( $23^\circ C$ )
Heater output ( $Q_1$ )	0 to 1 W
Heater factor ( $\alpha_1$ )	0.0131-0.0132 W/(% heater)
Heater output ( $Q_2$ )	0 to 0.75 W
Heater factor ( $\alpha_2$ )	0.0063-0.0066 W/(% heater)
Heat capacity ( $C_p$ )	500 J/kg-K
Surface Area Not Between Heaters ( $A$ )	$1.0 \times 10^{-3} m^2$ ( $10 cm^2$ )
Surface Area Between Heaters ( $A_s$ )	$2 \times 10^{-4} m^2$ ( $2 cm^2$ )
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 ( $\epsilon$ )	0.9
Stefan Boltzmann Constant ( $\sigma$ )	$5.67 \times 10^{-8} W/m^2 - K^4$
Conduction Time Constant ( $\tau_c$ )	21.1 – 23.3 sec

<sub>122</sub>  $U_s A_s (T_{H2} - T_{H1})$ . The radiative heat transfer from heater 1 to heater 2 (or

<sup>123</sup> vice versa) is given by  $Q_{R12} = \epsilon \sigma A (T_{H2}^4 - T_{H1}^4)$ .

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

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

<sup>124</sup> The dynamic temperature response of the two temperature sensors are primarily by conductive heat transfer from the heaters. The temperature sensors  
<sup>125</sup> are small in mass and surface area relative to the heaters so the heat transfer by  
<sup>126</sup> other mechanisms is ignored. The time constant  $\tau_c$  is a lumped parameter from  
<sup>127</sup> a discretized version of Fick's Law of heat transfer with  $\tau_c = m_s c_{ps} \Delta x / k_c A_{cond}$ ,  
<sup>128</sup> where  $m_s$  is the mass of the sensor,  $c_{ps}$  is the heat capacity of the sensor,  $k_c$  is  
<sup>129</sup> the thermal conductivity of the thermal epoxy, and  $\Delta x$  is the width of the thermal  
<sup>130</sup> epoxy. These parameters are combined together into one parameter  $\tau_c$  and  
<sup>131</sup> estimated from the data. The dynamic sensor temperature response expressions  
<sup>132</sup> are Equations 3 and 4.

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

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

<sup>134</sup> The test of the physics-based model is performed in two phases that includes  
<sup>135</sup> a model fitting phase followed by validation. The model fitting adjusts the pa-  
<sup>136</sup> rameters  $U$ ,  $U_s$ ,  $\alpha_1$ ,  $\alpha_2$ , and  $\tau_c$  to minimize the sum of squared error between  
<sup>137</sup> the model prediction and data as shown in Figure 6a. The model validation is a  
<sup>138</sup> simulation of the temperature profile given a different heater profile. The mea-  
<sup>139</sup> sured temperatures are not used in performing the simulation but are compared  
<sup>140</sup> afterwards to determine how well the model fitting performs on independent  
<sup>141</sup> data as shown in Figure 6b.

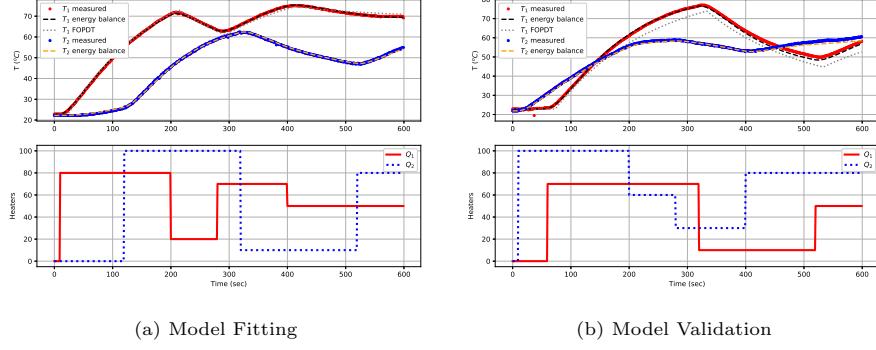


Figure 6: Dual Heater Step Response of the TCLab with Physics-based and FOPDT Model

142    *3.2. First-Order Plus Dead-time Model*

143    In addition to the physics-based model, a first-order plus dead-time (FOPDT)  
144    model is fit to step response data. An FOPDT model includes the gain ( $K_p=0.92$   
145     $^{\circ}\text{C}/\%$ ), time constant ( $\tau_p=175.2 \text{ sec}$ ), and delay time ( $\theta_p=15.6 \text{ sec}$ ). The FOPDT  
146    model is a single differential equation as shown in Equation 5.

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

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

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

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

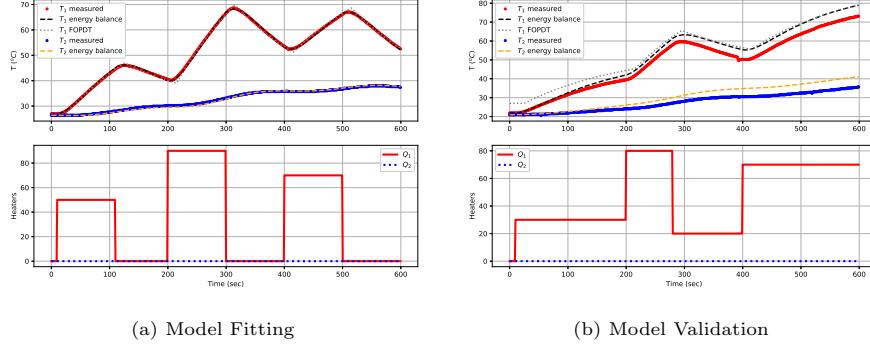


Figure 7: Single Heater Step Response of the TCLab with Physics-based and FOPDT Model

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

### 161     3.3. Linear Time Series Models

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

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

169     With  $n_\alpha = 3$  and  $n_\beta = 2$  the time series model has 5 adjustable parameters  
 170     and is shown in Equation 8. The ARX form uses prior temperature measure-  
 171     ments to predict the next temperature in the series,  $T_{C1,k+1}$ , while the OE

<sup>172</sup> form uses prior temperature predictions to predict the next temperature in the  
<sup>173</sup> sequence. The  $\gamma_1$  value is adjusted to create an unbiased model prediction.

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

<sup>174</sup> The OE identification form is used to reduce model bias. Equations 9a and  
<sup>175</sup> 9b have multiple inputs and multiple outputs for the case when  $n_\alpha = 2$  and  
<sup>176</sup>  $n_\beta = 1$ .

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

<sup>177</sup>

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

<sup>178</sup> An advantage of a linear time invariant (LTI) model such as SS, ARX, FIR,  
<sup>179</sup> or OE is that little or no physics-based information is required to obtain a  
<sup>180</sup> model prediction. When constraints are available, they are used to improve the  
<sup>181</sup> identification [41]. The model fit to the step test data is shown in Figure 8a and  
<sup>182</sup> the validation in Figure 8b.

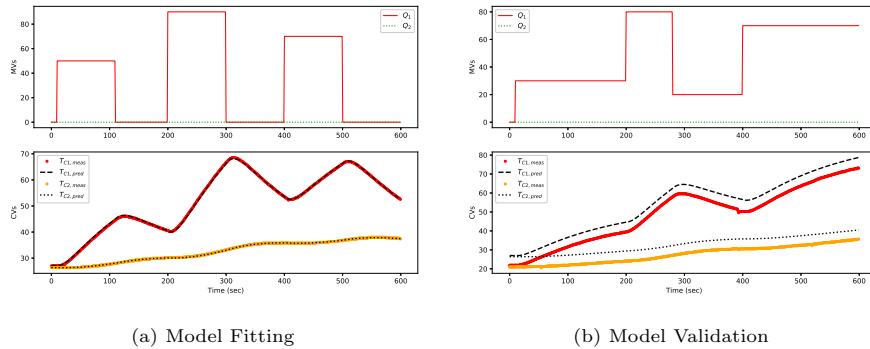


Figure 8: Single Heater Step Response of the TCLab with Linear Time Series

<sup>183</sup> There is insufficient data information to determine the  $\beta$  values associated  
<sup>184</sup> with  $Q_2$  because the value stays at zero for the duration of the test. A second

185 test is conducted where the second heater is also adjusted to get a multivariate  
 186 model from the step response data (see Figure 9).

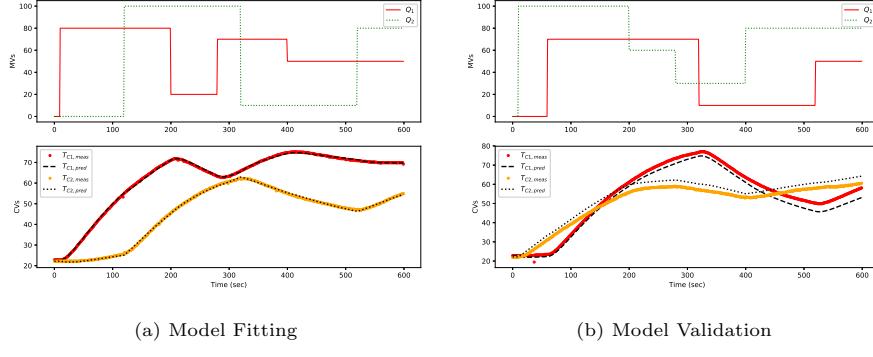


Figure 9: Dual Heater Step Response of the TCLab with Linear Time Series

### 187 3.4. Hammerstein Model with Artificial Neural Network

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

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

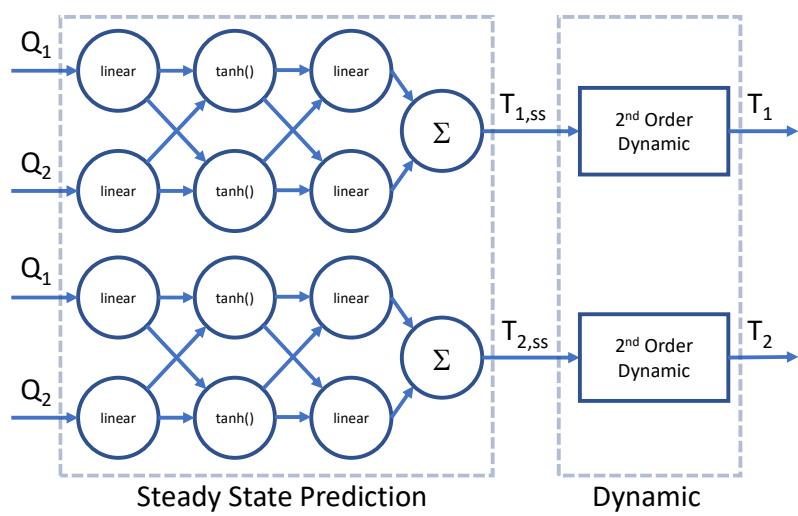


Figure 10: Architecture of the Hammerstein Model with a Steady-State Artificial Neural Network and Linear Dynamics.

204 and  $\tau_{p2}=20$  sec. The second order system approximates the time constant for  
 205 the heater and temperature sensor with heat conduction between the two.

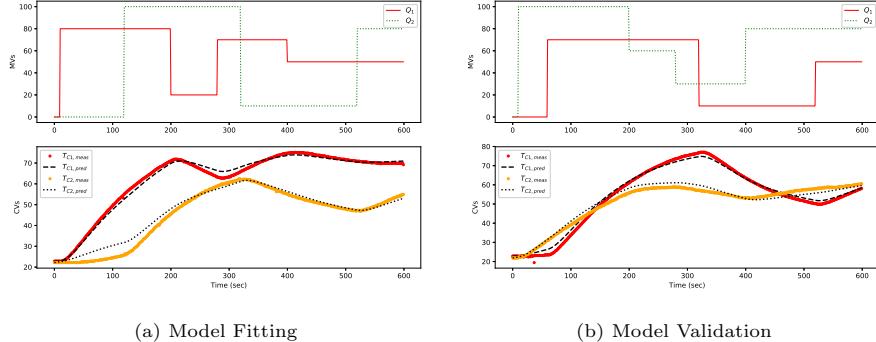


Figure 11: Hammerstein Model Fitting and Validation with 2 Heaters

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

### 213 3.5. Summary of Model Predictions with Validation

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

## 219 4. Benchmarking Closed-Loop PID Re-Tuning

220 The PID controller is a widely used basic regulatory control algorithm. PID  
 221 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

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

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

233 The objective of this closed-loop PID re-tuning is to demonstrate a TCLab  
 234 benchmark that uses historical data to optimally re-tune a PID controller. An  
 235 exhaustive search method visits all feasible combinations of the PI or PID pa-

237 rameeters to find an optimal value of the objective function without converging to  
 238 a local minimum for both output-error and input-move deviations. The method  
 239 uses simulation of the physical TCLab PID controller by: (a) re-playing back  
 240 the past or historical setpoint and load disturbances [52]; (b) allowing multi-  
 241 ple, simultaneous and probability-weighted process models to be included in  
 242 the simulations (i.e., multiple scenarios or situations each with specified proba-  
 243 bilities) for robustness; (c) including multiple and simultaneous PID controller  
 244 configuration formulations or even *ad hoc* controller designs; (d) specifying any  
 245 type of performance objective function criteria i.e., simultaneously minimize the  
 246 output-error and input-move variances, overshoot, etc. (e) adding stability rules  
 247 in the search to cut-off unstable sections of the closed-loop operating space and  
 248 (f) utilizing an indirect and constrained controller design technique [53].

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

$$OP_t = OP_{bias} + K_c \left( e_t + \frac{\Delta t}{\tau_I} \sum_1^t e_t + \tau_D \frac{PV_{t-1} - PV_t}{\Delta t} \right) \quad (10a)$$

265

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

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

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

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

289 where  $n$  is the norm  $w$  is the weighting factor for each term in the objective

290 function denoted as  $OE$  for output error and  $IM$  for input movement.

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

291 where  $UB$  is the upper bound of the input movement ( $IM$ ) which may be  
292 initially set by the centroid PID performance. Either the Archimedean or lexi-  
293 graphic form of the objective function can be used for PID controller tuning.  
294 In terms of convenience, the lexicographic form is easier to use because it re-  
295 quires one user input parameter,  $UB_{IM}$ , as opposed to the Archimedean form  
296 that requires two weighting factors on both  $OE$  and  $IM$  terms. One simplifi-  
297 cation of the Archimedean form is to reduce the weighting factors to one by  
298 dividing the objective by  $w_{OE}$ .

299 4.1. *TCLab Benchmark Validation*

300 The first step of the validation is to collect the closed-loop operation data  
301 and identify the ARX model parameters for identifying the ARX model. The  
302 setpoint is changed from ambient temperature at the initial steady-state con-  
303 dition to  $50 \text{ }^{\circ}\text{C}$ ,  $40 \text{ }^{\circ}\text{C}$ , and then to  $60 \text{ }^{\circ}\text{C}$ . The ranges of  $K_c$  and  $\tau_I$  are  
304 evaluated through the ARX model that includes the same setpoint sequences  
305 and load disturbance. The performance objective functions for each  $K_c$  and  $\tau_I$   
306 incremental combination are also calculated and stored. The  $\ell_1$ -norm objective  
307 function in the Archimedean form is chosen for the test with weighting factors  
308  $w_{OE} = 1$  and  $w_{IM} = 0.5$ . The  $K_c$  and  $\tau_I$  combination that gives a minimum  
309 value of objective function is then chosen as optimal PID tuning. The initial  $K_c$   
310 and  $\tau_I$  are from the FOPDT model in Section 3.2 and IMC aggressive tuning  
311 with  $K_c = 5.74 \frac{\%}{\text{ }^{\circ}\text{C}}$  and  $\tau_I = 175.2 \text{ sec}$ . Optimized values are  $K_c = 10.0 \frac{\%}{\text{ }^{\circ}\text{C}}$  and  
312  $\tau_I = 55.0 \text{ sec}$  as shown as the minimum value of the objective function contour  
313 map (see Figure 12).

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

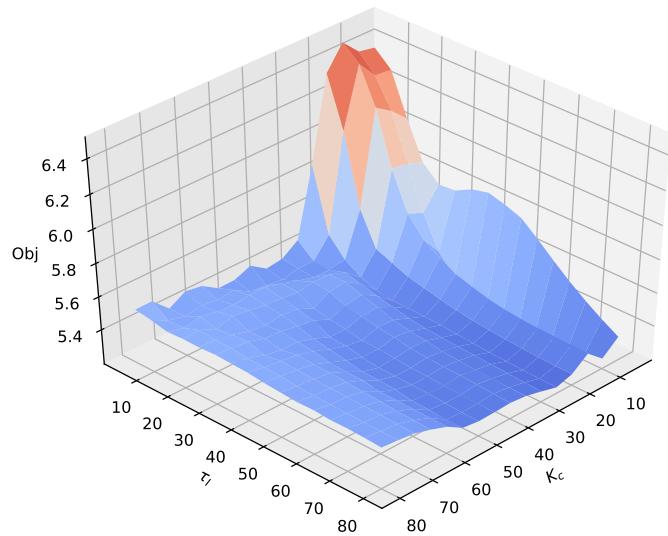


Figure 12: Average Integral Absolute Error (IAE) with  $K_c$  and  $\tau_I$  PID Parameters.

<sup>317</sup> optimized response. The validation of the optimal tuning parameter is displayed  
<sup>318</sup> as well.

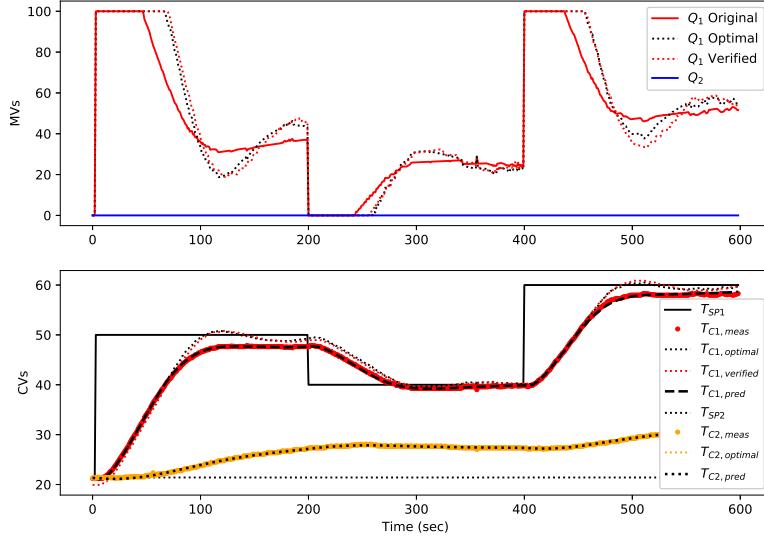


Figure 13: ARX Simulated and TCLab Validated Performance Improvement of 5.4%.

<sup>319</sup> The average IAE objective function is 6.09 with IMC tuning and 5.76 with  
<sup>320</sup> optimized parameters, an improvement of 5.4%. The PID improvement is simu-  
<sup>321</sup> lated with the ARX model and validated with closed-loop data from the Arduino  
<sup>322</sup> TCLab.

### <sup>323</sup> 5. Multivariate Control Benchmark

<sup>324</sup> Model predictive control (MPC) with the physics-based model, time series  
<sup>325</sup> linear model (ARX), and Hammerstein ANN model quantify multivariate con-  
<sup>326</sup> trol performance. Additional models in MPC or multivariate control strategies  
<sup>327</sup> are tested with the TCLab. This section shows benchmark performance with  
<sup>328</sup> three popular methods for multivariate control that range from linear to non-  
<sup>329</sup> linear and empirical to physics-based.

330 An  $\ell_1$ -norm objective function gives a target region for the temperature  
 331 range, rather than one specific target value. Equation 13 shows the  $\ell_1$ -norm  
 332 control formulation used in this work for model predictive control (MPC).

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

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

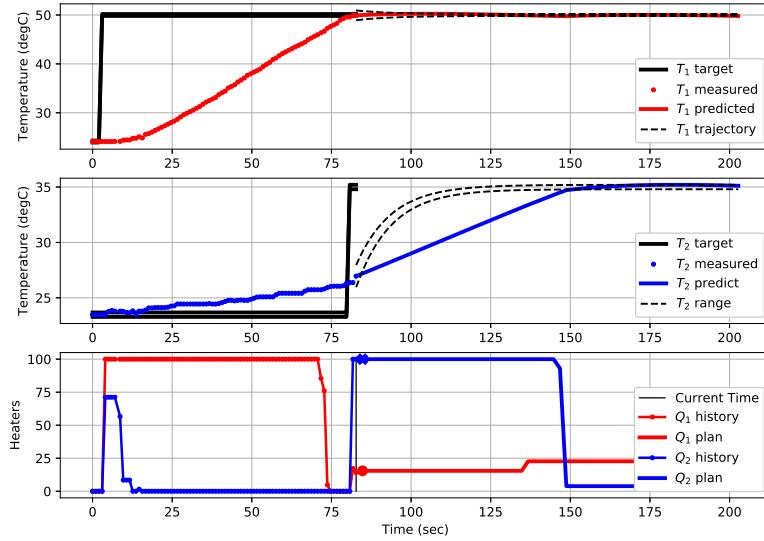


Figure 14: MPC with ARX Model at Cycle 81

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

352 **6. Benchmarking Model Predictive Control**

353 The multivariate models developed in Sections 3.1, 3.3, and 3.4 are compared  
354 in MPC. The MPC uses an  $\ell_1$ -norm objective with a temperature dead-band of  
355  $\pm 0.2^{\circ}\text{C}$  for  $T_{hi} - T_{sp}$ ,  $T_{lo} - T_{sp}$  and a first-order reference trajectory of 10 sec  
356 for setpoint changes. The move suppression factor  $c_{\Delta Q}$  is set to 0.1, the weights  
357  $w_{hi}$  and  $w_{lo}$  are set to 20.0, and the control and prediction horizon are 60 sec-  
358 onds. The linear ARX model has a cycle time of 1 second while the nonlinear  
359 physics-based and Hammerstein applications are re-computed every 2 seconds.  
360 The longer cycle time is required to enable all steps of data retrieval, model  
361 update, re-calculation of optimal move plan, retrieval of first step, and insertion  
362 into the process. Table 3 is a numeric comparison of the methods with quan-  
363 tified IAE rate ( $^{\circ}\text{C/sec}$ ) and Integral Average Move rate ( $\%/sec$ ) for the heater  
364 adjustments. Another common performance metric is a minimum variance as  
365 applied to multivariate control systems [59, 60]. Rate-based values are shown  
366 in this case because of the differing cycle times between the applications.

367 The benchmark results show that all models perform equally well in terms  
368 of the control performance ( $11.4\text{--}11.6^{\circ}\text{C/sec}$ ) as shown in Figures 15 to 17.

Table 3: Summary of Model Predictive Control Methods

<b>Model Description</b>		IAE Avg Rate (CVs)	IAE Avg Rate ( $\Delta MVs$ )
Physics-based	Lumped Parameter	11.5 $^{\circ}C/sec$	2.0 %/sec
Second Order ARX		11.6 $^{\circ}C/sec$	3.3 %/sec
Hammerstein with ANN and Linear Dynamics		11.4 $^{\circ}C/sec$	2.5 %/sec

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

376 The physics-based model has the potential to extrapolate to new operating  
 377 conditions without retuning. A physics-based MPC has the disadvantage of  
 378 relative difficulty in developing the model equations for complex systems. There  
 379 is also a potential for solver convergence problems if the physics-based model  
 380 is high nonlinear or does not have a suitable initial guess. This is not the case  
 381 for the TCLab where an approximate lumped-parameter model is an accurate  
 382 representation of the physical system. One drawback for the physics-based MPC  
 383 is that it cannot run at 1 sec cycles but does solve within a 2 second interval for  
 384 a 60 sec prediction horizon. The ARX control performance is shown in Figure  
 385 16.

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

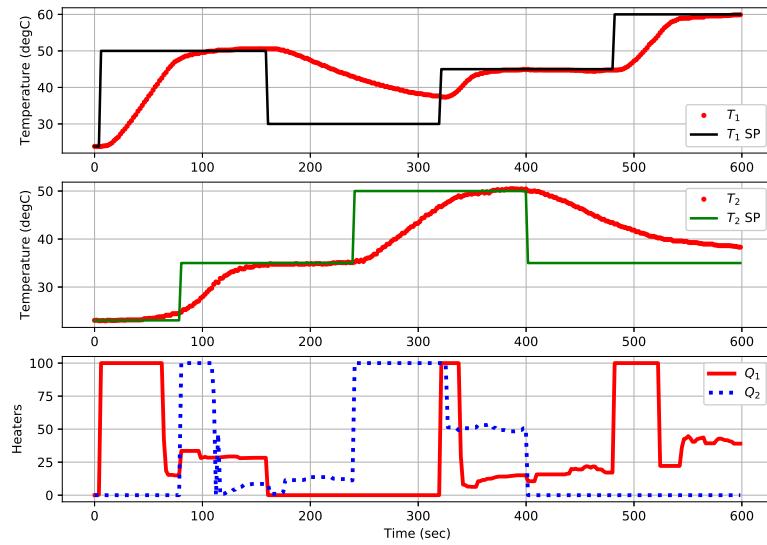


Figure 15: MPC with Physics-based Model

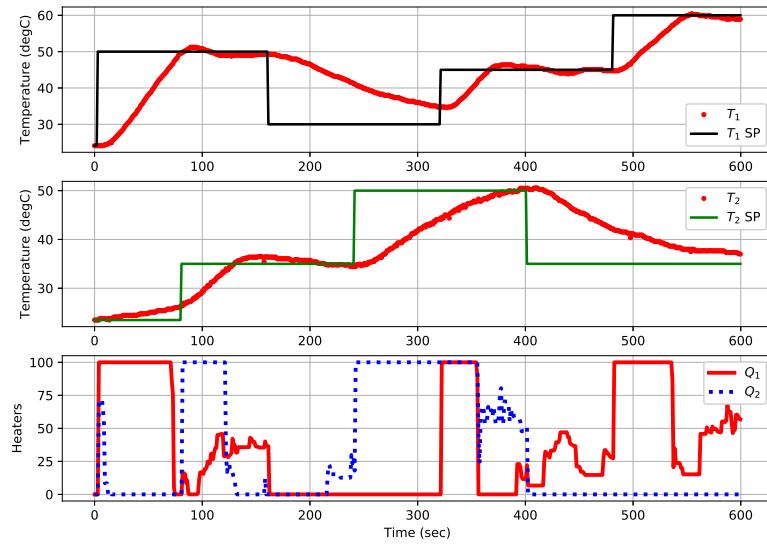


Figure 16: MPC with ARX Time Series Model

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

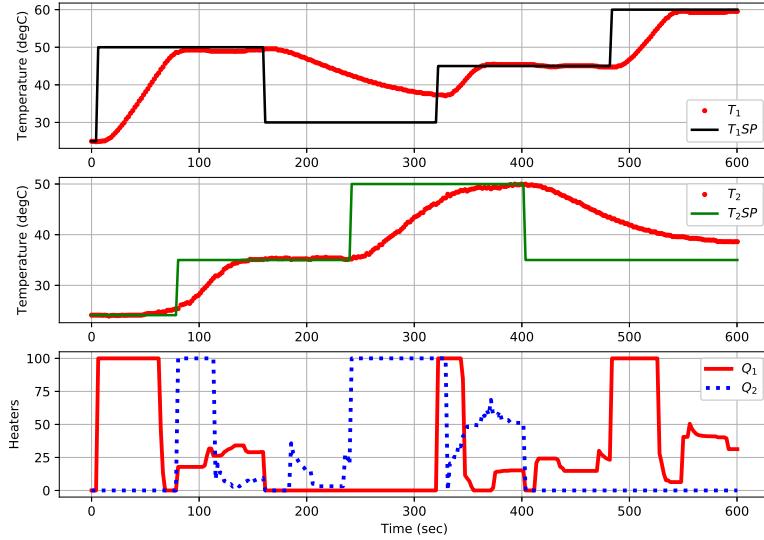


Figure 17: MPC with Hammerstein ANN Model

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

402    **7. Conclusion and Future Work**

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

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

425    **Acknowledgments**

426    This article is prepared for a Special Issue in honor of Tom Edgar's 75<sup>th</sup>  
427    birthday and to celebrate his lifetime of accomplishments and leadership in the  
428    area of Process Systems Engineering. This work is influenced by his work with  
429    energy systems, optimization, control engineering education, and advancements

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

432 **Appendix A. Software Interface to TCLab**

433 Two parts to the software interface are the firmware that runs on the Ar-  
434 duino Leonardo and the serial interface to interpret and command the TCLab.  
435 An important part of making the benchmark accessible is to create an inter-  
436 face to software (MATLAB, Simulink, and Python) where control algorithms  
437 are developed but also provide information for interfaces to other software plat-  
438 forms. There is an Arduino Support Package for MATLAB and Simulink from  
439 MathWorks that automatically loads firmware onto the Arduino when it is con-  
440 nected for the first time. The Arduino firmware for Python is an *ino* file that is  
441 augmented with additional sections to compile as *cpp* code with a gcc compiler  
442 through the Arduino IDE. The TCLab is pre-loaded with the Python interface  
443 firmware.

444

445 Listing 1: MATLAB Commands to Adjust Heaters and Display Temperatures

---

```
446 clear all
447 % include tclab.m
448 tclab;
449 disp('Turn on Heaters and LED')
450 h1(30); h2(60); led(1);
451 pause(10)
452 disp('Display Temperatures')
453 disp(T1C())
454 disp(T2C())
455 h1(0); h2(0); led(0);
```

---

457

## Adjust Heaters With Sliders

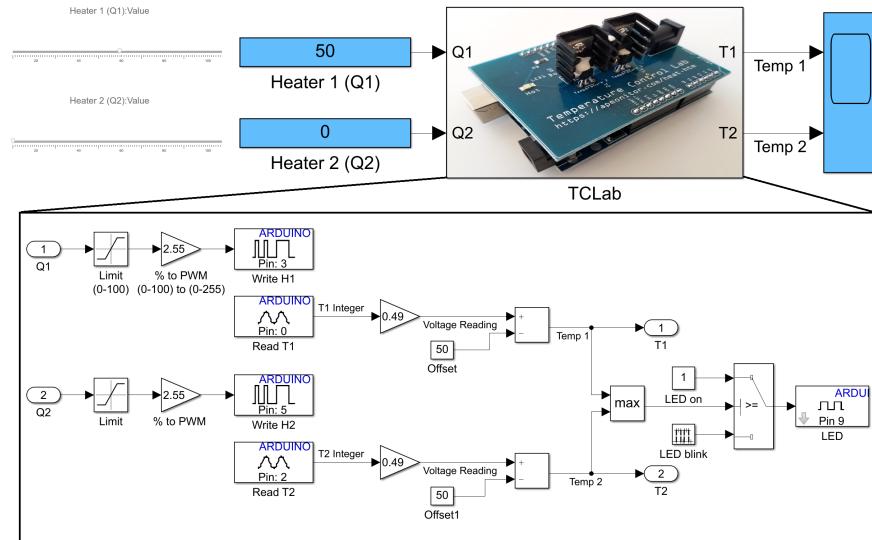


Figure A.18: Simulink Interface with Manual Sliders for Heater Levels.

Listing 2: Python Commands to Adjust Heaters and Display Temperatures

```

458
459 import tclab # pip install tclab
460
461 # Connect to Arduino
462 a = tclab.TCLab()
463 print('Turn on Heaters and LED')
464 a.Q1(30.0); a.Q2(60.0); a.LED(100)
465 time.sleep(10.0)
466 print('Display Temperatures')
467 print(a.T1)
468 print(a.T2)
469 a.close()
470

```

471    **References**

472    **References**

- 473    [1] A. F. Villaverde, D. Henriques, K. Smallbone, S. Bongard, J. Schmid,  
474       D. Cicin-Sain, A. Crombach, J. Saez-Rodriguez, K. Mauch, E. Balsa-Canto,  
475       et al., Biopredyn-bench: a suite of benchmark problems for dynamic mod-  
476       elling in systems biology, *BMC systems biology* 9 (1) (2015) 8.
- 477    [2] N. R. Lewis, J. D. Hedengren, E. L. Haseltine, Hybrid dynamic optimiza-  
478       tion methods for systems biology with efficient sensitivities, *Processes* 3 (3)  
479       (2015) 701. doi:10.3390/pr3030701.  
480       URL <http://www.mdpi.com/2227-9717/3/3/701>
- 481    [3] L. Peters, R. Arts, G. Brouwer, C. Geel, S. Cullick, R. J. Lorentzen,  
482       Y. Chen, N. Dunlop, F. C. Vossepoel, R. Xu, et al., Results of the Brugge  
483       benchmark study for flooding optimization and history matching, *SPE*  
484       *Reservoir Evaluation & Engineering* 13 (03) (2010) 391–405.
- 485    [4] V. P. Singh, A. Cavanagh, H. Hansen, B. Nazarian, M. Iding, P. S. Ringrose,  
486       et al., Reservoir modeling of CO<sub>2</sub> plume behavior calibrated against mon-  
487       itoring data from Sleipner, Norway, in: *SPE annual technical conference*  
488       and exhibition, Society of Petroleum Engineers, 2010.
- 489    [5] R. W. Rwechungura, E. Suwartadi, M. Dadashpour, J. Kleppe, B. A. Foss,  
490       et al., The Norne field case-a unique comparative case study, in: *SPE Intel-*  
491       *ligent Energy Conference and Exhibition*, Society of Petroleum Engineers,  
492       2010.
- 493    [6] J. Udy, B. Hansen, S. Maddux, D. Petersen, S. Heilner, K. Stevens,  
494       D. Lignell, J. D. Hedengren, Review of field development optimization of  
495       waterflooding, EOR, and well placement focusing on history matching and  
496       optimization algorithms, *Processes* 5 (3) (2017) 34.
- 497    [7] A. N. Eaton, L. D. Beal, S. D. Thorpe, C. B. Hubbell, J. D. Hedengren,  
498       R. Nybø, M. Aghito, Real time model identification using multi-fidelity

- 499 models in managed pressure drilling, Computers & Chemical Engineering  
500 97 (2017) 76–84.
- 501 [8] R. Asgharzadeh Shishavan, C. Hubbell, H. Perez, J. Hedengren, D. S. Pix-  
502 ton, et al., Combined rate of penetration and pressure regulation for drilling  
503 optimization using high speed telemetry, SPE Drilling & Completion Jour-  
504 nal 1 (SPE-170275-MS) (2015) 17–26.
- 505 [9] M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global  
506 optimization problems, arXiv preprint arXiv:1308.4008.
- 507 [10] R. V. Rao, V. J. Savsani, D. Vakharia, Teaching–learning-based optimiza-  
508 tion: an optimization method for continuous non-linear large scale prob-  
509 lems, Information sciences 183 (1) (2012) 1–15.
- 510 [11] S. M. Safdarnejad, J. D. Hedengren, N. R. Lewis, E. L. Hasel-  
511 tine, Initialization strategies for optimization of dynamic sys-  
512 tems, Computers & Chemical Engineering 78 (2015) 39 – 50.  
513 doi:10.1016/j.compchemeng.2015.04.016.
- 514 URL [http://www.sciencedirect.com/science/article/pii/  
515 S0098135415001179](http://www.sciencedirect.com/science/article/pii/S0098135415001179)
- 516 [12] R. Huang, Nonlinear model predictive control and dynamic real time opti-  
517 mization for large-scale processes, Ph.D. thesis, Carnegie Mellon University  
518 (12 2010).
- 519 [13] M. Hehn, R. Ritz, R. D’Andrea, Performance benchmarking of quadrotor  
520 systems using time-optimal control, Autonomous Robots 33 (1-2) (2012)  
521 69–88.
- 522 [14] W. Chen, Y. Ren, G. Zhang, L. T. Biegler, A simultaneous approach for sin-  
523 gular optimal control based on partial moving grid, AIChE Journal 65 (6).
- 524 [15] L. D. Beal, D. Petersen, D. Grimsman, S. Warnick, J. D. Hedengren, In-  
525 tegrated scheduling and control in discrete-time with dynamic parameters

- 526 and constraints, Computers & Chemical Engineering 115 (2018) 361 – 376.  
527 doi:<https://doi.org/10.1016/j.compchemeng.2018.04.010>.
- 528 URL <http://www.sciencedirect.com/science/article/pii/S0098135418303120>
- 530 [16] L. D. R. Beal, D. Petersen, G. Pila, B. Davis, S. Warnick, J. D. Hedengren,  
531 Economic benefit from progressive integration of scheduling and control for  
532 continuous chemical processes, Processes 5 (4).
- 533 [17] D. Petersen, L. D. R. Beal, D. Prestwich, S. Warnick, J. D. Hedengren,  
534 Combined noncyclic scheduling and advanced control for continuous chem-  
535 ical processes, Processes 5 (4).
- 536 [18] Y. Nie, L. T. Biegler, C. M. Villa, J. M. Wassick, Discrete Time Formulation  
537 for the Integration of Scheduling and Dynamic Optimization, Industrial &  
538 Engineering Chemistry Research 54 (16) (2015) 4303–4315. doi:[10.1021/ie502960p](https://doi.org/10.1021/ie502960p).  
539 URL <http://pubs.acs.org/doi/abs/10.1021/ie502960p>
- 540 [19] P. F. Odgaard, J. Stoustrup, M. Kinnaert, Fault tolerant control of wind  
541 turbines—a benchmark model, IFAC Proceedings Volumes 42 (8) (2009)  
542 155–160.
- 543 [20] G. M. Kopanos, C. A. Méndez, L. Puigjaner, MIP-based decomposition  
544 strategies for large-scale scheduling problems in multiproduct multistage  
545 batch plants: A benchmark scheduling problem of the pharmaceutical  
546 industry, European Journal of Operational Research 207 (2) (2010)  
547 644–655. doi:[10.1016/j.ejor.2010.06.002](https://doi.org/10.1016/j.ejor.2010.06.002).  
548 URL <http://linkinghub.elsevier.com/retrieve/pii/S037722171000408X>
- 549 [21] D. Saygin, E. Worrell, M. K. Patel, D. Gielen, Benchmarking the energy use  
550 of energy-intensive industries in industrialized and in developing countries,  
551 Energy 36 (11) (2011) 6661–6673.

- 554 [22] L. D. Beal, J. Park, D. Petersen, S. Warnick, J. D. Hedengren, Combined  
555 model predictive control and scheduling with dominant time constant com-  
556 pensation, *Computers & Chemical Engineering* 104 (2017) 271–282.
- 557 [23] M. Baldea, I. Harjunkoski, Integrated production scheduling and process  
558 control: A systematic review, *Computers & Chemical Engineering* 71  
559 (2014) 377–390. doi:10.1016/j.compchemeng.2014.09.002.
- 560 [24] J. Kelly, J. Hedengren, A steady-state detection (SSD) algorithm to detect  
561 non-stationary drifts in processes, *Journal of Process Control* 23 (3) (2013)  
562 326–331.
- 563 [25] N. L. Ricker, J. Lee, Nonlinear model predictive control of the Tennessee  
564 Eastman challenge process, *Computers & Chemical Engineering* 19 (9)  
565 (1995) 961–981.
- 566 [26] A. Bathelt, N. L. Ricker, M. Jelali, Revision of the Tennessee Eastman  
567 process model, *IFAC-PapersOnLine* 48 (8) (2015) 309–314.
- 568 [27] N. I. Vitzilaios, N. C. Tsourveloudis, Test bed for unmanned helicopters'  
569 performance evaluation and benchmarking, in: *IEEE/RSJ IROS 2008*  
570 *Workshop on Performance Evaluation and Benchmarking for Intelligent*  
571 *Robots and Systems*, Citeseer, 2008.
- 572 [28] A. Cardoso, V. Sousa, P. Gil, Demonstration of a remote control laboratory  
573 to support teaching in control engineering subjects, *IFAC-PapersOnLine*  
574 49 (6) (2016) 226–229.
- 575 [29] P. K. Singh, S. Bhanot, H. K. Mohanta, V. Bansal, Self-tuned fuzzy logic  
576 control of a ph neutralization process, in: *2015 21st International Confer-  
577 ence on Automation and Computing (ICAC)*, IEEE, 2015, pp. 1–6.
- 578 [30] I. Alvarado, D. Limon, D. M. De La Peña, J. Maestre, M. Ridao, H. Scheu,  
579 W. Marquardt, R. Negenborn, B. De Schutter, F. Valencia, A comparative  
580 analysis of distributed MPC techniques applied to the HD-MPC four-tank  
581 benchmark, *Journal of Process Control* 21 (5) (2011) 800–815.

- 582 [31] V. Kirubakaran, T. Radhakrishnan, N. Sivakumaran, Distributed multi-  
583 parametric model predictive control design for a quadruple tank process,  
584 Measurement 47 (2014) 841–854.
- 585 [32] Y. Alipouri, J. Poshtan, Optimal controller design using discrete linear  
586 model for a four tank benchmark process, ISA transactions 52 (5) (2013)  
587 644–651.
- 588 [33] B. Spivey, J. Hedengren, T. Edgar, Constrained nonlinear estimation for  
589 industrial process fouling, Industrial & Engineering Chemistry Research  
590 49 (17) (2010) 7824–7831.
- 591 [34] V. M. Zavala, L. T. Biegler, Optimization-based strategies for the operation  
592 of low-density polyethylene tubular reactors: nonlinear model predictive  
593 control, Computers & Chemical Engineering 33 (10) (2009) 1735–1746.
- 594 [35] J. Rossiter, S. Pope, B. L. Jones, J. Hedengren, Evaluation and demon-  
595 stration of take home laboratory kit, IFAC-PapersOnLine 52 (9) (2019)  
596 56–61.
- 597 [36] P. Oliveira, J. Hedengren, An APMonitor temperature lab PID control ex-  
598 periment for undergraduate students, in: 24th IEEE Conference on Emerg-  
599 ing Technologies and Factory Automation (ETFA), Zaragoza, Spain, IEEE,  
600 2019, pp. 790–797.
- 601 [37] F. G. Shinskey, Process control: as taught vs as practiced, Industrial &  
602 engineering chemistry research 41 (16) (2002) 3745–3750.
- 603 [38] T. F. Edgar, B. A. Ogunnaike, J. J. Downs, K. R. Muske, B. W. Bequette,  
604 Renovating the undergraduate process control course, Computers & chem-  
605 ical engineering 30 (10-12) (2006) 1749–1762.
- 606 [39] J. Alford, T. Edgar, Preparing chemical engineering students for industry,  
607 Chemical Engineering Progress 113 (11) (2017) 25–28.

- 608 [40] S. J. Qin, T. A. Badgwell, A survey of industrial model predictive control  
609 technology, *Control Engineering Practice* 11 (7) (2003) 733–764.
- 610 [41] J. Udy, L. Blackburn, J. D. Hedengren, M. Darby, Reduced order modeling  
611 for reservoir injection optimization and forecasting, in: Proceedings of the  
612 FOCAPO/CPC Conference, Tuscon, AZ, USA, 2017, pp. 8–12.
- 613 [42] K. J. Åström, T. Hägglund, PID controllers: theory, design, and tuning,  
614 Vol. 2, Instrument society of America Research Triangle Park, NC, 1995.
- 615 [43] B.-S. Ko, T. F. Edgar, Assessment of achievable PI control performance  
616 for linear processes with dead time, in: Proceedings of the 1998 American  
617 Control Conference. ACC (IEEE Cat. No. 98CH36207), Vol. 3, IEEE, 1998,  
618 pp. 1548–1552.
- 619 [44] S. J. Qin, Control performance monitoring—a review and assessment, *Com-*  
620 *puters & Chemical Engineering* 23 (2) (1998) 173–186.
- 621 [45] B.-S. Ko, T. F. Edgar, PID control performance assessment: The single-  
622 loop case, *AICHE Journal* 50 (6) (2004) 1211–1218.
- 623 [46] J. G. Ziegler, N. B. Nichols, Optimum settings for automatic controllers,  
624 *trans. ASME* 64 (11).
- 625 [47] K. J. Åström, T. Hägglund, Automatic tuning of simple regulators with  
626 specifications on phase and amplitude margins, *Automatica* 20 (5) (1984)  
627 645–651.
- 628 [48] N. J. Killingsworth, M. Krstic, PID tuning using extremum seeking: on-  
629 line, model-free performance optimization, *IEEE control systems magazine*  
630 26 (1) (2006) 70–79.
- 631 [49] Z.-L. Gaing, A particle swarm optimization approach for optimum design  
632 of PID controller in AVR system, *IEEE transactions on energy conversion*  
633 19 (2) (2004) 384–391.

- 634 [50] M. I. Solihin, L. F. Tack, M. L. Kean, Tuning of PID controller using  
635 particle swarm optimization (PSO), International Journal on Advanced  
636 Science, Engineering and Information Technology 1 (4) (2011) 458–461.
- 637 [51] B. Mohanty, S. Panda, P. Hota, Controller parameters tuning of differential  
638 evolution algorithm and its application to load frequency control of multi-  
639 source power system, International journal of electrical power & energy  
640 systems 54 (2014) 77–85.
- 641 [52] J. D. Kelly, Tuning digital PI controllers for minimal variance in manipu-  
642 lated input moves applied to imbalanced systems with delay, The Canadian  
643 Journal of Chemical Engineering 76 (5) (1998) 967–974.
- 644 [53] J. Park, C. Patterson, J. Kelly, J. Hedengren, Closed-loop PID re-tuning  
645 in a digital twin by re-playing past setpoint and load disturbance data, in:  
646 2019 (AIChE) Spring Meeting, New Orleans, LA, AIChE, 2019, pp. 1–6.
- 647 [54] L. Beal, D. Hill, R. Martin, J. Hedengren, Gekko optimization suite, Pro-  
648 cesses 6 (8) (2018) 106.
- 649 [55] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, F. J. Doyle III, Process  
650 dynamics and control, John Wiley & Sons, 2010.
- 651 [56] A. Voda, I. Landau, Multi-step closed loop identification and control design  
652 procedure-applications, IFAC Proceedings Volumes 27 (8) (1994) 1543–  
653 1548.
- 654 [57] E. Jahanshahi, S. Skogestad, Closed-loop model identification and PID/PI  
655 tuning for robust anti-slug control, IFAC Proceedings Volumes 46 (32)  
656 (2013) 233–240.
- 657 [58] R. Tchamna, M. Lee, Analytical design of an industrial two-term controller  
658 for optimal regulatory control of open-loop unstable processes under oper-  
659 ational constraints, ISA transactions 72 (2018) 66–76.

- 660 [59] B.-S. Ko, T. F. Edgar, Performance assessment of multivariable feedback  
661 control systems, *Automatica* 37 (6) (2001) 899–905.
- 662 [60] C. A. Harrison, S. J. Qin, Minimum variance performance map for con-  
663 strained model predictive control, *Journal of Process Control* 19 (7) (2009)  
664 1199–1204.