Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Teaching Digital Twins in Process Control Using the Temperature Control Lab

Alexander W. Dowling*, Molly Dougher, Madelynn J. Watson, Hailey G. Lynch, Zhicheng Lu, and Daniel J. Laky

University of Notre Dame, Department of Chemical and Biomolecular Engineering, Notre Dame, IN 46556, United States
* Corresponding Author: adowling@nd.edu.

ABSTRACT

Process control can be one of the most exciting and engaging chemical engineering undergraduate courses! This paper describes our experience transforming *Chemical Process Control* into *Data Analytics, Optimization, and Control* at the University of Notre Dame (second semester required course in the junior year). Our modern course is built around six hands-on experiments in which students practice data-centric modeling and analysis using the Arduino-based Temperature Control Lab (TCLab) hardware. We argue that state-space dynamic modeling and optimization are more critical for educating modern chemical engineers than topics such as frequency domain analysis and controller synthesis emphasized in many classical undergraduate control courses. All the course material is available online at https://ndcbe.github.io/controls.

Keywords: Process Control, Education, Dynamic Modelling, System Identification, Model Predictive Control, Pyomo, Process Monitoring, Process Operations, Industry 4.0

INTRODUCTION

This paper describes several innovations in the undergraduate chemical engineering required process controls course at the University of Notre Dame, including:

- Hands-on experiential learning,
- State-space modeling,
- Advanced data analytics,
- Optimization for decision-making, and
- Model predictive control (MPC) and state estimation.

Through this perspective, we frame key concepts from process control – modeling, feedback, and optimization – through the lens of digital twins, which a recent U.S. National Academies report [1] defines as follows (emphasis ours):

*"A **digital twin** is a set of **virtual information constructs** that mimics the structure, context, and behavior of a **natural**, **engineered**, or **social system** (or **system-of-systems**), is **dynamically updated** with data from its physical twin, has a **predictive capability**, and **informs decisions that realize value**. The **bidirectional interaction** between the virtual and the physical is central to the digital twin."*

Figure 1 shows the overarching themes for the course. The interplay between the physical and information realms in feedback control closely echoes key themes in the digital twin paradigm.

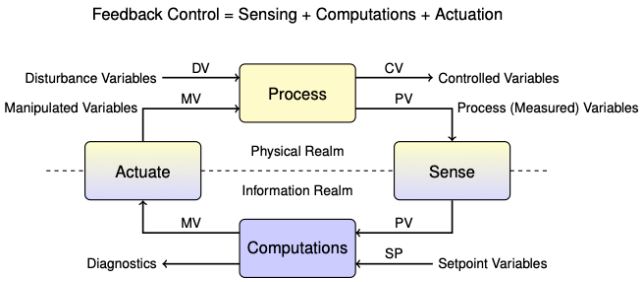

Figure 1. Feedback control, realized through integrating the physical and information realms, is the central organizing concept in the course.

Temperature Control Lab

The course is organized around the Temperature Control Lab (TCLab) [2-5], an inexpensive Arduino device (shown in **Figure 2**) that enables experiential learning. Each student purchases a TCLab device instead of a textbook, as all course material is available online at

https://ndcbe.github.io/controls. Using the TCLab, students:

- Perform step and sine wave tests;
- Identify state-space models via nonlinear regression (with uncertainty quantification);
- Use simulation to tune various classical control algorithms, e.g., on/off control, PI(D) control;
- Implement open-loop and closed-loop model predictive control;
- Learn about state estimation from an optimization perspective; and
- Compare the above-described control algorithms for following the temperature profile for tempering dark chocolate using the TCLab as the physical system

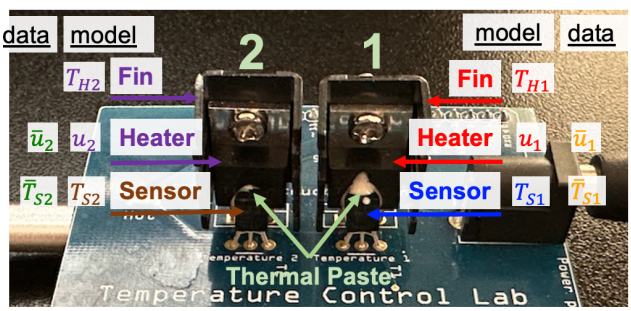

Figure 2. Photo of the TCLab hardware with two channels, each consisting of a heater, fin, and sensor. There is visible manufacturing variability in the thermal paste for each channel (and device). As such, each student has a hardware system that exhibits slightly different dynamics.

Broader Curriculum Context

At Notre Dame, chemical engineering undergraduates enroll in process controls (3 credits) in the second semester of their junior year. As prerequisites, students complete a four-semester calculus sequence, a dedicated linear algebra course, and a department-taught required course on numerical methods and applied statistics. Students concurrently take *Transport II (Heat and Mass)* with process controls. Students enroll in *Chemical Reaction Engineering* in their first semester senior year and capstone *Chemical Process Design* in their final semester. Currently, our students learn MATLAB and Python during their first year through a college-wide required course. Python programming is further taught in the sophomore-required course *Numerical and Statistical Analysis*.

State-Space Modeling and Data Analytics

Most process control courses focus on frequency domain analysis [6]. While the frequency domain perspective provides beautiful insights into certain aspects of controls (e.g., time delays and responses to periodic inputs), it requires significant time to (re)teach Laplace transformations. Moreover, students often struggle with the abstract nature of frequency domain analysis and have difficulty connecting these concepts with modeling in other core chemical engineering courses.

Compared to this classical treatment, we emphasize mathematical modeling, numerical methods, and applied statistics. The course emphasizes building state-space dynamic models from experimental data, as described below, which naturally complements the (partial) differential (algebraic) equation models taught in transport, kinetics, and thermodynamics. We (anecdotally) find that students conceptually understand state-space modeling, as it is grounded in core chemical engineering concepts, such as mass and energy balances. Moreover, we use advanced data analytics techniques such as nonlinear regression to train these physics-based state-space models from data and characterize the prediction uncertainty.

Learning Objectives and Organization

The semester learning goals are:

1. Develop mathematical models for dynamical systems from data and first principles using modern statistical methods (in Python);
2. Predict dynamical system performance using numerical methods (in Python);
3. Analyze, implement, tune, and debug feedback controllers using the hands-on laboratory (in Python);
4. Formulate and solve optimization problems for decision-making (in Python); and
5. Demonstrate mastery of at least two of the above skills in an open-ended group project.

The course content and assessments are organized into three parts, as described in the following sections. Students individually complete two homework assignments and six hands-on laboratory assignments, each with significant data analysis and interpretation components. Students take one in-class exam during the semester to assess individual mastery. In place of a final exam, students complete an open-ended group project.

PART 1: DATA-CENTRIC MODELING OF DYNAMICAL SYSTEMS (4 WEEKS)

Our assessments emphasize data analysis, computer-aided problem-solving, and open-ended discussion questions to connect theoretical concepts from lectures (75 minutes, twice each week) with hands-on activities using the TCLab. Students attend a dedicated 50-

minute laboratory session in a computer classroom once a week to start the new laboratory assignments with easy access to the teaching team when questions arise. Lectures in this part of the course emphasize simulation to understand the performance of (non)linear systems of ordinary differential equations (ODEs).

Homework 1

Homework 1 reviews key concepts from this later course, especially mathematical modeling, numerical solutions to initial value problems, nonlinear regression, uncertainty quantification (e.g., covariance and correlation matrices), uncertainty propagation, and numerical solutions to systems of nonlinear equations. See our previous work on innovations in modernizing this pre-requisite course. [7-8] Course material is available at https://ndcbe.github.io/data-and-computing.

Lab 1: Step Test

In the Lab 1 assignment, students perform a step test (**Figure 3**) using their TCLab hardware. Students then practice three methods – graphical analysis, simulation trial-and-error, and nonlinear regression – to build a first-order dynamic model for a single channel of their TCLab hardware:

$$C_p \frac{\mathrm{d}T_1}{\mathrm{dt}} = U_a(T_{amb} - T_1) + \alpha P_1 u_1 \quad (1)$$

where T_1 is the average temperature of heater/sensor one, T_{amb} is the ambient temperature, C_p is the heat capacity, U_a is the heat transfer coefficient between the sensor/heat body and the environment, α is a system calibration constant, P_1 is the maximum power setting for heater one, and u_1 is the percentage of available power for heater one. We reformulate this as a first-order linear model without a time delay using the deviation variable $T'_1 = T_1 - T_{amb}$:

$$\frac{C_p}{U_a}\frac{\mathrm{d}T'_1}{\mathrm{dt}} = -T'_1 + \frac{\alpha P_1}{U_a} u_1 \quad (2)$$

Here $\tau = C_p\, U_a^{-1}$ is the time constant and $K = \alpha P_1 U_a^{-1}$ is the gain. Through discussion questions, students reflect on the pros and cons of each model-building approach and brainstorm ways to handle noisy data.

Lab 2: Model Identification

In the Lab 2 assignment, students perform a sine wave test with their TCLab hardware. Using their step test (Lab 1) and sine test (Lab 2) data, students build a second-order dynamic model:

$$C_p^H \frac{\mathrm{d}T_{H,1}}{\mathrm{dt}} = U_a(T_{amb} - T_{H,1}) + U_b(T_{S,1} - T_{H,1}) + P_1 u_1 \quad (3)$$

$$C_p^S \frac{\mathrm{d}T_{S,1}}{\mathrm{dt}} = U_b(T_{H,1} - T_{S,1}) \quad (4)$$

where the superscripts (and subscripts) H and S denote the heater and sensor, respectively, and U_b is the heat transfer coefficient between the heater and sensor.

Throughout the Lab 2 assignment, students explore the pros and cons of step tests (**Figure 3**) and sine tests (**Figure 4**) and the two model choices (first- and second-order dynamic models). Specifically, students find that the single-state model is reasonable for the step test data, whereas the two-state model is superior for the sine test data. The two-state model incorporates natural delays in the system due to thermal inertia by considering two different temperatures (sensor and heater/fin). The sine test, which is inherently more dynamic, emphasizes the importance of (approximating) delays. Moreover, the students concurrently take *Heat and Mass Transfer*, which empowers them to connect our state-space modeling course with the broader chemical engineering curriculum and appreciate the physical differences between these two models. On a more applied note, Lab 2 shows the importance of multi-start initialization for nonlinear regression and how simultaneously regressing data from two types of experiments (sine and step tests) results in a model with lower parameter and prediction uncertainty.

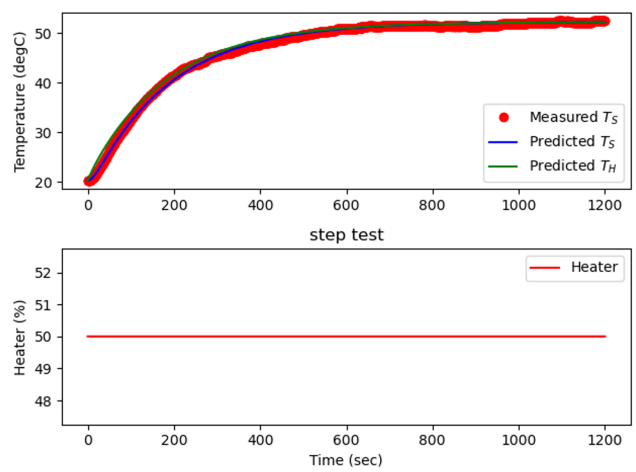

Figure 3. Parameter estimation results for the step test in Lab 1.

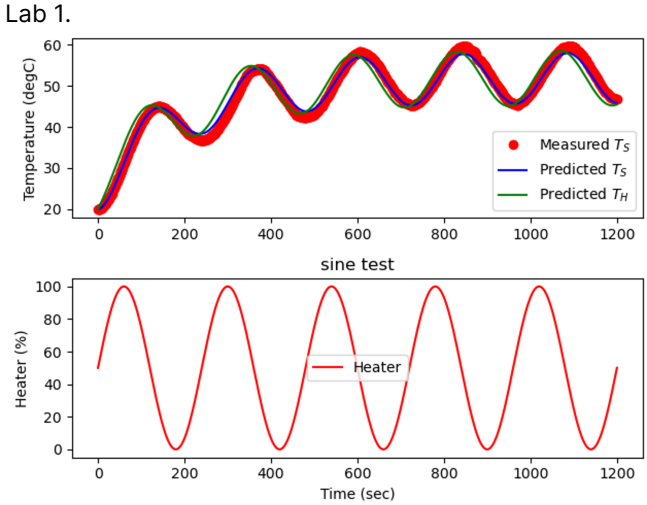

Figure 4. Parameter estimation results for the sine test in Lab 2.

PART 2: FEEDBACK CONTROL (4 WEEKS)

As a motivating example throughout the semester, students compare different control strategies to follow a piecewise linear temperature profile that mimics the process of tempering dark chocolate. This example is well-suited to the temperature range of the TCLab (from ambient to approximately 65 °C). Moreover, students can easily relate to this example and appreciate how other forms of temperature control, e.g., polymerase chain reaction (PCR) or (bio)manufacturing, are relevant to their career interests.

Lab 3: Relay (On/Off) Control

In Lab 3, students implement a simple relay (on/off) controller with a deadband (**Figure 5**). As a pre-lab exercise, students use their mathematical model from Lab 2 to tune the deadband parameter via simulation. The lab assignment focuses on implementing the controller, quantifying the controller performance, and comparing the performance to the simulation. As a final exercise in Lab 3, students use two relay controllers. One controller tracks the chocolate tempering temperature profile in channel 1. In contrast, the other controller tracks a constant temperature setpoint (30 °C) in channel 2. Through discussion questions, students contemplate potential challenges with independently controlling both channels, which interact due to their proximity.

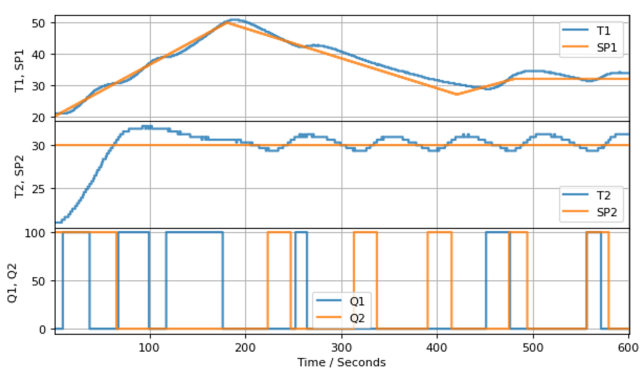

Figure 5. TCLab hardware results using relay (on/off) control in Lab 3. Top/Middle: channel 1/2 temperature data and setpoints. Bottom: controller actions.

Lab 4: PI Control

In Lab 4, students explore two proportional-integral (PI) control implementations: a "textbook" position-form PI controller and a more practical velocity-form PI controller with anti-windup reset and bumpless transfer. As a pre-lab exercise, students use their mathematical model from Lab 2 to tune the gains in both PI controllers by analyzing the eigenvalues of the closed-loop dynamical system. Students then test both controllers and quantify their performance. Thus, this lab introduces students to arguably the most ubiquitous low-level controller in (bio)molecular/chemical processes, emphasizing the power of model-based controller design and analysis. Moreover, in the complementary lectures, students learn how to use linearization to analyze the local dynamics of nonlinear systems.

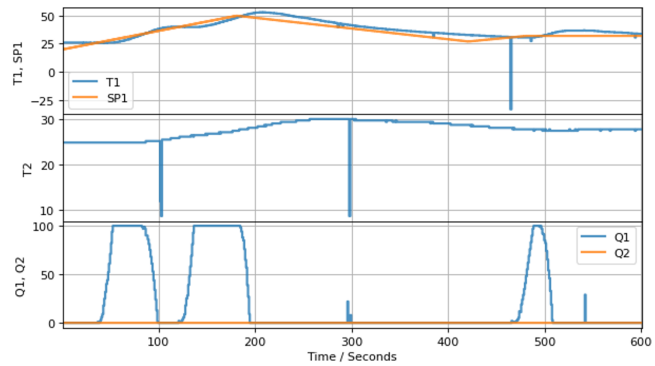

Figure 6. TCLab hardware results using PI control in Lab 4. The TCLabs occasionally drop data, which provides a good "real world" lesson: data are messy, and hardware is imperfect!

PART 3: OPTIMIZATION (4 WEEKS)

Few of our undergraduate alumni are practicing control engineers, but many (anecdotally) regularly use data analytics for decision-making in their professional careers. Thus, we emphasize optimization as a framework for data-driven decision-making. In some of our favorite lectures, we introduce students to linear programming using production planning examples, e.g., blending in a refinery. Students are amazed to see how optimization easily applies to business-relevant analyses, such as pricing the marginal value of overtime labor or additional supplies or determining the best way to change blending operations to accommodate a new product (e.g., mid-grade fuel) or environmental regulations.

Homework 2: Introduction to Pyomo

In Homework 2, students practice optimization modeling and problem-solving using Pyomo [9], an open-source algebraic modeling environment. Specifically, we focus on two examples – product blending and portfolio management under uncertainty – to appeal to diverse career interests. In the portfolio optimization example, students analyze time-series closing day prices for five funds to compute the covariance of their daily return rate. Students then use quadratic programming to calculate the Pareto optimal trade-off between expected return and volatility (risk). These examples help students appreciate the importance of modeling, data analytics, and optimization for business decisions.

Lab 5: Open-Loop Optimal Control

Equipped with an operational understanding of Pyomo, in Lab 5, students compute the open-loop, optimal time-varying heating profile $u(t)$ to track the dark chocolate tempering temperature set point. Specifically, students use Pyomo.DAE [10] to formulate and solve the dynamic optimization problem using their model from Lab 2:

$$\min_{u} \int_{t_0}^{t_f} \left(\left\| T_S^{SP}(t) - T_S(t) \right\|^2 + w \left\| T_S^{SP}(t) - T_H(t) \right\|^2 \right) dt \tag{5}$$

$$\text{s.t. } C_p^H \frac{\mathrm{d}T_H}{\mathrm{dt}} = U_a(T_{amb} - T_H) + U_b(T_S - T_H) + Pu(t) + d(t) \tag{6}$$

$$C_p^S \frac{\mathrm{d}T_S}{\mathrm{dt}} = U_b(T_H - T_S) \tag{7}$$

$$T_H(t_0) = T_{amb} \tag{8}$$

$$T_S(t_0) = T_{amb} \tag{9}$$

$$\frac{du}{dt} \leq \dot{u}_{max} \tag{10}$$

$$\frac{du}{dt} \geq \dot{u}_{min} \tag{11}$$

where the superscript SP denotes the setpoint, $d(t)$ is the disturbance (often set to zero), and $\dot{u}_{max}$ and $\dot{u}_{min}$ are the maximum and minimum rate of change of the control action, respectively.

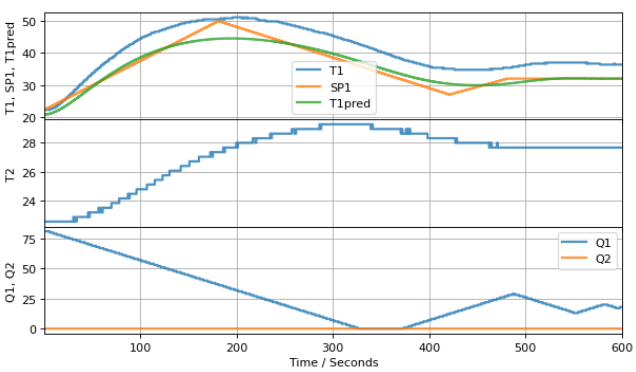

Figure 7. TCLab hardware results using open-loop optimal control in Lab 5.

In Lab 5, students execute the open-loop optimal solution with their TCLab hardware and compare their results to the model prediction (**Figure 7**). While much of the code is provided, students add a ramping constraint on the control input $u(t)$ and analyze the impact of this limitation (Equations 10-11). Through discussion questions, students explain why their open-loop optimal solutions make sense from a physical perspective.

Through examples in lectures, students learn how to use Pyomo.DAE to numerically solve boundary value problems with (partial) (differential) algebraic equations. This provides students with modeling tools for other classes, such as *Heat and Mass Transfer*, *Chemical Reaction Engineering*, and *Chemical Process Design*, in addition to our semester final project.

Lab 6: Model Predictive Control

In Lab 5, students learn that while open-loop optimization can provide a physically sensible plan, without feedback, model mismatch can cause subpar hardware performance compared to other control strategies (e.g., Lab 3, Lab 4). This motivates Lab 6, which focuses on model predictive control (MPC). We describe this to students as essentially putting the optimization formulation from Lab 5 inside of a for loop and repeating the optimization as new measurements become available.

But there is a catch! The preferred TCLab model has one measured state, $T_{S,1}(t)$, and one unmeasured state, $T_{H,1}(t)$. To initialize the optimal planning problem, we need to estimate the unmeasured state. Thus, the TCLab example motivates state estimation, which we pose as an optimization problem:

$$\min_{d(t)} \int_{t_0}^{t_f} (\|\bar{T}_s(t) - T_S(t)\|^2 + w\|d(t)\|^2) dt \tag{12}$$

$$\text{s.t. } C_p^H \frac{\mathrm{d}T_H}{\mathrm{dt}} = U_a(T_{amb} - T_H) + U_b(T_S - T_H) + Pu(t) + d(t) \tag{13}$$

$$C_p^S \frac{\mathrm{d}T_S}{\mathrm{dt}} = U_b(T_H - T_S) \tag{14}$$

$$T_H(t_0) = T_{amb} \tag{15}$$

$$T_S(t_0) = T_{amb} \tag{16}$$

Specifically, we infer the unmeasured disturbance and heater temperature $T_{H,1}(t)$ by minimizing the deviation between the measured sensor temperature $\bar{T}_s$ and the model prediction T_S.

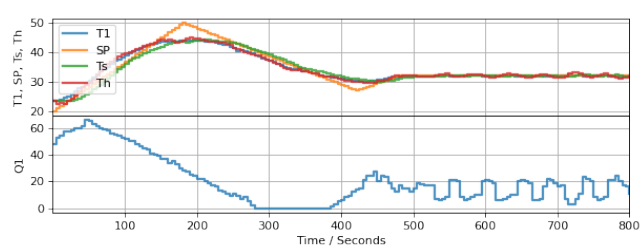

Figure 8. TCLab hardware results using MPC in Lab 6.

Lab 6 builds up these concepts by individually solving the optimal planning and state estimation problems separately. Finally, we integrate these to form a complete MPC demonstration (**Figure 8**). The data integration between the state estimation and control optimization problems is nuanced and benefits from advanced

programming concepts, e.g., Object Oriented Programming. As such, the students are provided with most of the code for Lab 6.

Thus, the Lab 6 assignment focuses on running the MPC scheme with their TCLab hardware and critically analyzing the results. Specifically, students are asked to quantitatively compare the performance of the four control strategies studied throughout the semester (Labs 3, 4, 5, and 6) on the same dark chocolate tempering problem. Students then discuss the pros and cons of each approach. Finally, the students describe a real-world application of receding horizon optimal control, state estimation, or both. Many of our students use this to share their creativity or interests. Popular answers include self-driving cars, supply chain management, and financial trading.

SEMESTER PROJECT (3 WEEKS)

In place of a final exam, teams of three or four students work on a final project incorporating at least two of the class learning objectives. Many groups are very creative, with past project topics including analysis of university sustainability projects (e.g., solar or geothermal energy), glucose modeling for diabetes management, and trajectory planning for a submarine. We use the final three class periods for short presentations. During the two preceding weeks, we hold 6- to 7-minute project consultation meetings with each team during lectures. This has the bonus of introducing students to open-ended projects, meeting preparation, time management, and note-taking skills, which are especially helpful for their capstone projects the following year.

CONCLUSIONS AND OUTLOOK

In summary, we have reorganized Process Control around six hands-on laboratory experiments, transforming the course from classical frequency domain analysis to emphasize dynamic state-space modeling, data analytics, and optimization for decision-making. The TCLab experiments enable us to distill the course into two key concepts:

1. The importance of feedback (**Figure 1**), both in classical control applications (e.g., temperature regulation) as well as many systems from daily life (e.g., cruise control).
2. The utility of predictive mathematical models for engineering [12].

We argue this modern take on process controls is better suited for today's chemical engineers, especially as the profession diversifies into new application areas and technologies. Moreover, these two key concepts capture the ethos of digital twins, which we believe will play a significant role in the future of engineering. Furthermore, we encourage students who find specific topics especially exciting, such as dynamic modeling, optimization, or statistical analysis, to pursue a technical elective in this area in their final year at Notre Dame.

We conclude by describing several opportunities for future refinement and customization. In parallel to exploring these refinements, we are exploring ways to measure student outcomes to assess the efficacy of the new course organization.

What about time delays?

One of the great strengths of frequency domain analysis is the ability to model and design control systems to mitigate time delays. Moreover, time delays are especially important in traditional chemical engineering applications (e.g., fluid flowing in a pipe) and newer frontiers (e.g., systems biology). One option is to augment the state-space model with additional states to approximate the time delay. In essence, the two-state TCLab model does this by considering the lag in heat transfer from the heater/fin to the sensor. Another option is to introduce delay differential equations.

Lost Insights from the Frequency Domain?

While we argue that the state-space perspective is more intuitive for most chemical engineering students, we also acknowledge that frequency domain analysis provides an important modeling perspective. The sine test experiment (Lab 2) starts to introduce these ideas. One opportunity is to expand this course section further to introduce Fourier analysis as a tool for analyzing periodic data. We hypothesize this data analysis perspective would be more engaging for students.

Advanced Data Science Topics

The TCLab experiments can also be extended to introduce advanced data science topics. For example, time series analysis techniques could be introduced to go beyond standard independent and identically distributed Gaussian error assumptions for systems identification and instead consider auto-correlated measurement errors. This semester, we introduced maximum likelihood estimation to provide students with a more foundational understanding of nonlinear regression and introduce a core concept in many machine learning workflows. Similarly, the course content can be extended to consider more sophisticated uncertainty quantification methods, e.g., Bayesian inference or nonlinear confidence regions. Moreover, we use the TCLab example in our hands-on workshops to introduce Pyomo.DoE, an open-source tool for model-based design of experiments [11]. We have also begun exploring the TCLab as a motivating system for hybrid models to quantify model misspecification [13].

Advanced Process Control Topics

In a Lab 3 extra credit exercise, students use their Lab 1 to 3 data to build and estimate a four-state model

that captures the interacting dynamics of the two channels in the TCLab (see **Figure 2**). This more sophisticated model motivates advanced topics such as estimability analysis and model-based design of experiments. We encourage students to presume this or other advanced process control topics as final projects. For example, students are encouraged to develop two PI controllers for one channel to follow the dark chocolate tempering profile while the other maintains a constant, elevated temperature (similar to Lab 2 with relay control). This reenforces ideas of interacting control systems. Another project idea includes model-free (e.g., Ziegler-Nichols) PID controller tuning. Some ambitious groups may even try to extend the state estimation formulation in Lab 6 for fault detection, such as determining when a hair dryer starts to blow on the TCLab.

ACKNOWLEDGEMENTS

Many course design ideas this paper presents originated with Prof. Jeffery Kantor. Jeff spent over five years conceptualizing, testing, and refining this course and creating much of the content in our online textbook. Jeff passed away in 2023. This paper is dedicated to him.

REFERENCES

1. National Academies of Sciences, Engineering, and Medicine. Foundational research gaps and future directions for digital twins. Washington, DC: The National Academies Press. (2024) https://doi.org/10.17226/26894
2. Oliveira PM and Hedengren JD. An APmonitor temperature lab PID control experiment for undergraduate students. *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* 790-797 (2019) 10.1109/ETFA.2019.8869247
3. Park J, Martin RA, Kelly JD, Hedengren JD. Benchmark temperature microcontroller for process dynamics and control. *Comput Chem Eng* 135:106736 (2020) https://doi.org/10.1016/j.compchemeng.2020.106736
4. de Moura Oliveira PB, Hedengren JD, Rossiter JA. Introducing digital controllers to undergraduate students using the TClab Arduino kit. *IFAC PapersOnLine* 53(2):17524-9 (2020) https://doi.org/10.1016/j.ifacol.2020.12.2662
5. Hedengren JD and Kantor J. Computer programming and process control take-home lab. *CACHE Newsletter* (2020)
6. Seborg DE, Edgar TF, Mellichamp DA, Doyle III FJ. Process Dynamics and Control. John Wiley & Sons (2016)
7. Boukouvala F, Dowling A, Verrett J, Ulissi Z, Zavala V. Computational notebooks in chemical engineering curricula. *Chemical Engineering Education* 54(3):143-150 (2020)
8. Dowling A. Training all chemical engineers in computing and data science. *2019 AIChE Annual Meeting* (2019) LAPSE:2019.1133
9. Bynum ML, Hackebeil GA, Hart WE, Laird CD, Nicholson BL, Siirola JD, Watson JP, Woodruff DL. Pyomo – Optimization Modeling in Python. Springer (2017)
10. Nicholson B, Siirola JD, Watson JP, Zavala VM, and Biegler LT. Pyomo.DAE: A modeling and automatic discretization framework for optimization with differential and algebraic equations. *Math Prog Comp* 10(2):187-223 (2018) https://doi.org/10.1007/s12532-017-0127-0
11. Wang J, Dowling AW. Pyomo.DOE: An open-source package for model-based design of experiments in Python. *AIChE J* 68(12):e17813 (2022) https://doi.org/10.1002/aic.17813
12. Agi, D.T., Jones, K.D., Watson, M.J., Lynch, H.G., Dougher, M., Chen, X., Carlozo, M.N., Dowling, A.W. Computational toolkits for model-based design and optimization. *Current Opinion in Chemical Engineering*, 43, 100994 (2024) https://doi.org/10.1016/j.coche.2023.100994
13. Eugene, E.A., Jones, K.D., Wang, J., Gao, X., Dowling, A.W. Learning and Optimization Under Epistemic Uncertainty with Bayesian Hybrid Models. *Computers & Chemical Engineering*, 179, 108430 (2023). https://doi.org/10.1016/j.compchemeng.2023.108430

© 2025 by the authors. Licensed to PSEcommunity.org and PSE Press. This is an open access article under the creative commons CC-BY-SA licensing terms. Credit must be given to creator and adaptations must be shared under the same terms. See https://creativecommons.org/licenses/by-sa/4.0/