

Teaching PLCS and PID Tuning Virtually using DoMore Software

Dr. Kevin R. Anderson, California State Polytechnic University, Pomona

Dr. Kevin R. Anderson is a Professor of Mechanical Engineering at California State Polytechnic University, Pomona. Dr. Anderson obtained his Ph.D. and MSME degrees from the University of Colorado Boulder and his BSME at California State Polytechnic University, Pomona. Dr. Anderson has over 100 technical publications. He was the previous director of the Control Systems Laboratory in the Mechanical Engineering Department California State Polytechnic University, Pomona focusing on PLC and PAC lab development. Dr. Anderson also holds an active Mechanical Engineering P.E. license.

Dr. Nolan Tsuchiya P.E., California State Polytechnic University, Pomona

Dr. Nolan Tsuchiya is an Assistant Professor of Mechanical Engineering at California State Polytechnic University, Pomona. Dr. Tsuchiya obtained his Ph.D. from University of California Los Angeles (UCLA) focusing on Dynamic Systems and Control. Dr. Tsuchiya teaches Feedback Control Systems, System Dynamics, and Computer Programming courses using MATLAB/SIMULINK at California State Polytechnic University, Pomona. He is currently the faculty advisor for the premier Formula and BAJA SAE collegiate racing teams. He holds an active California PE license.

Zechariah Alan Maynard, Cal Poly Pomona

Bakersfield College - AA Physics, AA Mathematics, AA-ST - Engineering Cal Poly Pomona - BS Mechanical Engineering (May, 2021)

Teaching PLCS and PID Tuning Virtually using Do-more Software

Programmable Logic Controllers (PLCs) have applications within mechanical and industrial engineering and are well adapted to a wide range of automation tasks including industrial processes in manufacturing and building automation. PLC systems are suited for environments where monitoring of several digital inputs and outputs (I/O) is required. While digital (on / off) signals comprise the majority of logic points within a PLC architecture, it is also common to require analog control of a process variable. Examples of variables requiring analog control are building temperature or fluid level of a tank wherein a 2-state digital input is insufficient. In these instances, Proportional Integral Derivative, PID controllers serve as a robust class of controllers that are easily implemented and field tuned without prior identification of the system dynamics. The combination of PLC architectures and PID control algorithms allows precise control and monitoring of large-scale industrial processes such as automobile factories and global shipment facilities.

To serve the growing need for mechanical engineers within the control systems and automation sector, many universities have begun incorporating PLC courses or learning modules within the engineering curriculum¹⁻⁸. In many cases, the PLC system interfaces with physical hardware¹⁻⁴ that is representative of an industrial application, but scaled down for the university laboratory, for instance an automated page-turning robot³. In other instances, however, researchers have leveraged simulation and web-based technologies to achieve similar learning outcomes⁹⁻¹¹. In these scenarios, the physical plant is replaced by a virtual object, but the programming tasks and recorded data remain largely unchanged. By applying virtual learning solutions to PLC courses, instructors eliminate the physical hardware bottleneck and can reach a larger student base and provide each student increased engagement with the learning materials.

In this paper, a virtual teaching module using the Do-more PLC programming software is adapted for use in a mechanical engineering undergraduate course and is the latest in an ever-evolving category of simulation solutions designed to teach PLC programming and PID control. In addition to allowing instructors to reach a larger body of students, the virtual learning environment removes the requirement of hardware interfacing while retaining the same learning outcomes. This increases the accessibility for students with disabilities, as the entire learning experience is conducted through software on a PC. The software itself is provided to students by the college of engineering, further reducing the burden on the student and allowing any student access to the learning materials.

This paper presents the use of online software to teach PLCs in a Mechanical Engineering curriculum. The software used is Automation Direct's Do-more¹² PLC Emulator Software development suite. It is chosen as the instructional vehicle herein in order to illustrate how students can learn PLC programming and implement Proportional Integral Derivative (PID) loops within simulated control system environments. The use of PID loops in industry is commonplace and the theory behind them is well understood¹³. The body of the paper is outlined as follows: first an overview of the Do-more PLC emulator is given, followed by an example of

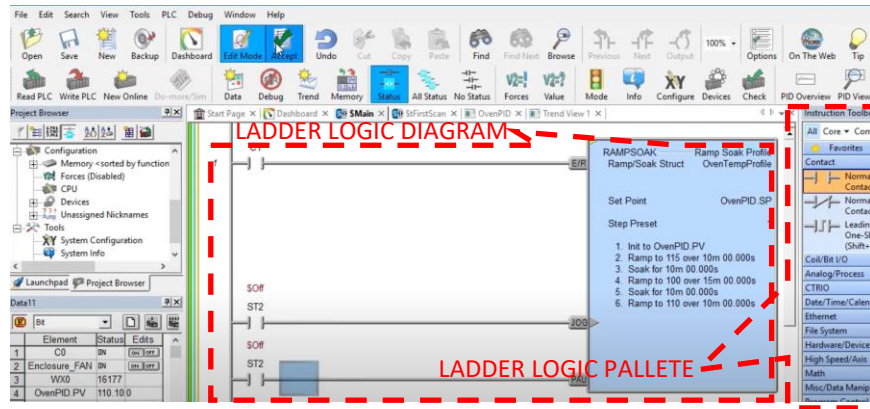


Figure 2 – Do-more PLC Ladder Logic Interface

PID Loop Example

The use of PID loop in industrial controls is proliferate including ovens and furnaces used in industrial heat treatment as well as pumps used for moving fluids, and robotic motion controllers. Figure 3 shows a typical PID loop defined using ladder logic within the Do-more PLC emulator.

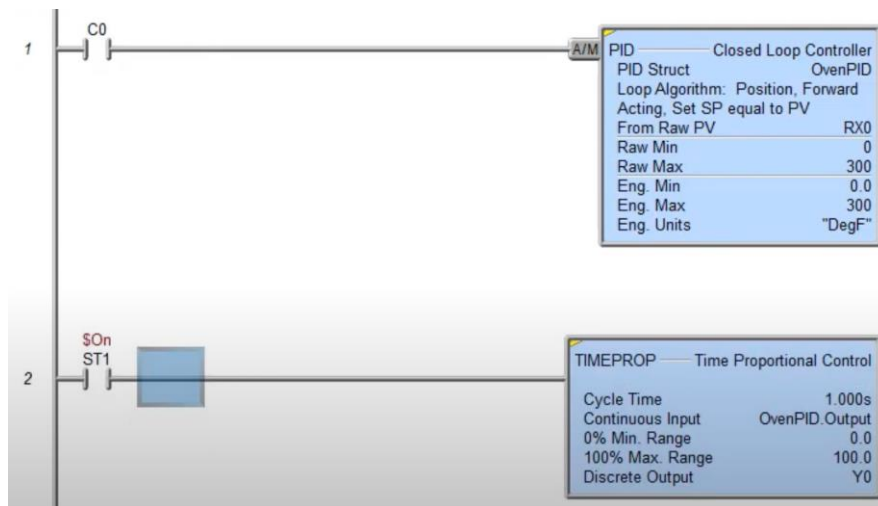


Figure 3 – PID Loop Example

In Figure 3, the PID loop is implemented on a traditional Ladder Logic diagram as an object-oriented programming structure in the upper ladder rung of Figure 3. The PID object contains the information for the set points and gain of the PID loop. The PID loop is synchronized with a continuous DO LOOP using the continuous loop object shown on the lower ladder rung of Figure 3. Figure 4 shows graphical input and output processed by the PID loop upon execution of the ladder logic.



Figure 4 – PID Input and Output

The white plot in Figure 4 is the set-point (PLC controller input), the blue plot is the PID controller output (PID conditioned signal), while the green plot is the real time error. As shown in the PID input/output dialog of Figure 4, students are able to implement a PID loop to understand the effects of signal conditioning (i.e. noise) on the performance of the controller, as well as to observe the effect of adjusting the proportional gain setting on the behavior of the output.

PID Loop Tuning

One of the most important concepts related to controls curriculum is the task of tuning a PID controller. Traditionally, in process control there are two methods of tuning: i) manual-tuning and ii) auto-tuning. Manual tuning is typically used in process control industry^{17,18}, whereas auto-tuning via Ziegler Nichols techniques is addressed in the theoretical setting¹³. The task of PID tuning entails establishing appropriate gain values for the process being controlled. This can be done manually, but most modern controllers possess auto-tuning capabilities. Manual tuning can be carried out with enough information about the process being controlled, and in some instances it may be possible to calculate optimal values of gain, reset and rate. Most PLC-based process controllers integrate the functionality of auto-tuning. In auto-tuning, the PLC learns how the process responds to a disturbance or change in set point, and calculates appropriate PID gain settings. Most PID controllers used in industry today incorporate auto-tuning functions.

Operating details vary between manufacturers, but all follow rules where the controller “learns” how the process responds to a disturbance or change in set point and calculates appropriate PID settings. The use of Do-more PLC emulator software allows the students an interactive platform to hone their skills at tuning a PID loop. Figure 5 shows the PID loop being tuned using the Do-more platform.



Figure 5 – PID Manual vs. Auto Tuning

The white plot in Figure 5 is the set-point (PLC controller input), the blue plot is the PID controller output (PID conditioned signal), while the green plot is the real time error. As shown in Figure 5, the use of the Do-more PLC simulator allows students to gain hands-on familiarity using manual vs auto-tuning and to witness the effects of implementing different tuning strategies on virtual electro-mechanical control plants. This type of experiential learning allows student to develop a “feel” for the response of typical control systems using PID loops.

Typical Outputs from using the Do-more PLC Software

In this section of the paper we present typical outputs which are obtained using the Do-more PLC emulator for Manual and Auto-tuning a temperature based PID loop, respectively. Figure 6 shows the results of manual tuning of a thermal based PID¹⁹.

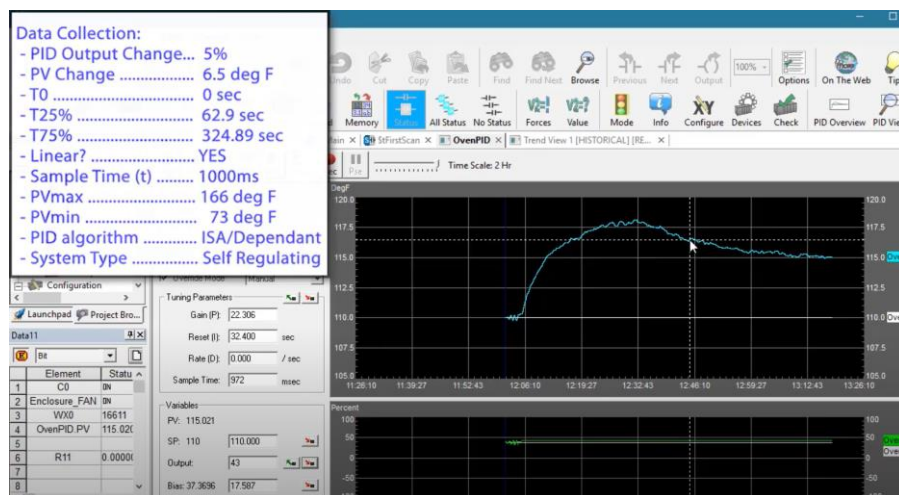


Figure 6 – Results for Manually Tuning a PID Loop Using Do-more PLC Emulator

The following formulae are used to compute the PID gains

$$k_P = (1.086/PG)(TC/DT)^{0.869} \quad (1)$$

$$k_I = TC/[0.74(TC/DT)^{0.13}] \quad (2)$$

$$k_D = 0.348TC(DT/TC)^{0.914} \quad (3)$$

where the constants (1.086, 0.869, 0.74, 0.13, 0.348 and 0.914) appearing in the above formulae¹⁹ hold for a dependent self-regulating PID algorithm as used by the Do-more PLC Emulator. Dependent denotes that changes of k_P affect k_I and k_D , while self-regulating means that response to a step input change is characterized by a change of the process variable, PV which moves to and stabilizes (or self-regulates) at a new value. In addition, Process Gain is denoted by PG , Time Constant is denoted by TC and Dead Time is denoted by DT , which are determined via the following relationships

$$PG = (PV \text{ change})/(PID \text{ output change}) \quad (4)$$

$$TC = 0.9(T75 - T25) \quad (5)$$

$$DT = (T75 - T0) - 1.4TC - t \quad (6)$$

where $T0$ denotes time the process was started, $T25$ denotes the time the PV reaches 25% of its final value and $T75$ denotes the time the PV reaches 75% of its final value. Using the parameter values listed on the inset of Figure 6 affords $PG = 1.3$, $TC = 235.79$ sec, $DT = 5$ sec, $P = 38.39$, $I = 14.98$ sec and $D = 1.42$ sec. It should be noted that the dead time of $DT = 5$ sec was scaled directly from the output of Figure 6 since the dead time here was less than 10% of the time constant¹⁹. Additionally, various tuning methods and their response outputs in the simulator are shown below for comparison. Figure 7 shows typical output for Open Loop, Auto, and Manual Tuning Exercise Using Do-more PLC Emulator²⁰.

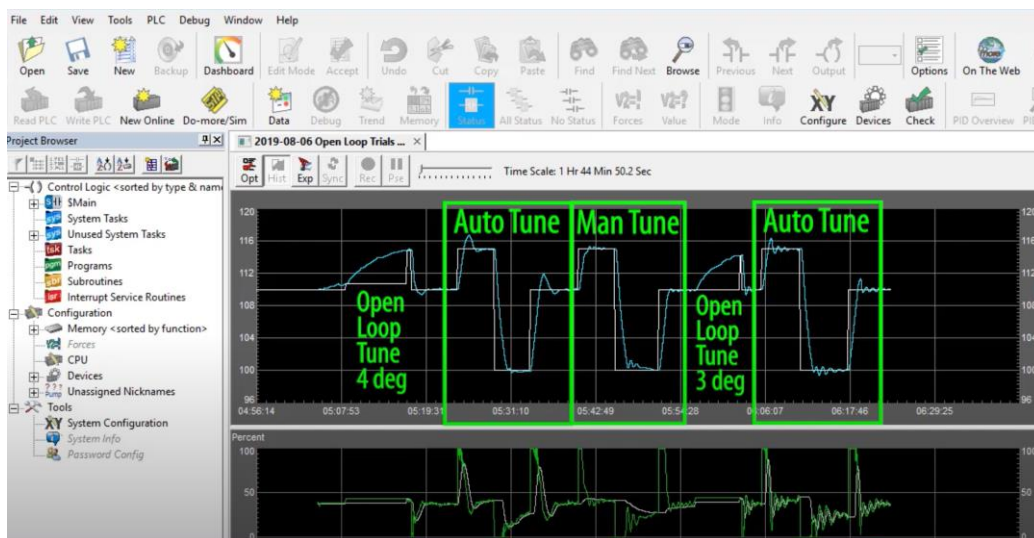


Figure 7 – Results for Open Loop, Auto, and Manual Tuning Using Do-more PLC Emulator
Planned Implementation of the Do-more PLC Emulator Software within the Mechanical Engineering Controls Curriculum

This section of the paper details how this software will be embedded into the course structure and how the software will be used in online classes. The paper presented herein is the outline of the framework for use of the Do-more PLC as was the subject of a senior project and a case study for how to realize on-line teaching resources for teaching Industrial Controls. The Do-more PLC PID tuning on-line base experiential learning presented herein is planned to be incorporated into a semester based technical elective ME 4990/L “Industrial Controls using PLCs/Lab” to be offered tentatively in Spring 2022. The ME 4990 course will be offered as a hybrid course, with in person (TBD per COVID lockdown restrictions) and on-line via CANVAS student activities used to facilitate the classroom management.

Outcomes and Objectives of Mechanical Engineering Controls Curriculum

The Mechanical Engineering curriculum contains a controls course and laboratory offered as the senior level. The outcomes and objectives encompass that the students learn the concepts of PID loop implementation in control systems, and furthermore that the students understand how the various gain setting of the PID loop affect the output of the controller. The use of the Do-more PLC emulator allows the students to simulate a virtual plant, and then experiment virtually by tuning the PID loop manually vs. using auto-tuning. One of the key components of the curriculum outcomes and objectives of the Mechanical Engineering controls curriculum is to expose the students to PID examples and real-world type of case studies. The uses of the Do-more PLC emulator development suite discussed herein is a viable option to address this outcome and objective.

Within the controls course, the Do-more PLC emulator can be used in several instances to reinforce PID control concepts. Traditionally, students learn the structure of a PID controller in the classical sense, which entails analysis of closed-loop system poles as functions of the three tuning parameters (k_P , k_I , k_D). In parallel, students are also exposed to PID tuning techniques, including the widely adopted Ziegler-Nichols algorithms, for which there are several variants. By incorporating the Do-more virtual learning tool, students can validate their analytical findings with the simulated results, resulting in positive reinforcement and an improved learning experience.

Indeed, reinforcing PID tuning concepts can certainly be achieved through traditional laboratory apparatuses, such as the Educational Control Products line of Mechatronic systems (see Figure 8). However, these turnkey systems are orders of magnitude more expensive and require physical space. With recent advances in technologies associated with distance learning, software packages like the Do-more PLC emulator can be an effective proxy for a traditional hardware system. Specifically, the traditional hardware systems require a couple of weeks of introductory assignments to acquaint students with the proper protocols and safety measures. Further, due to a limited number of available workstations, laboratory time is split between several groups of students, each of which ultimately receive less time interacting with the apparatus.

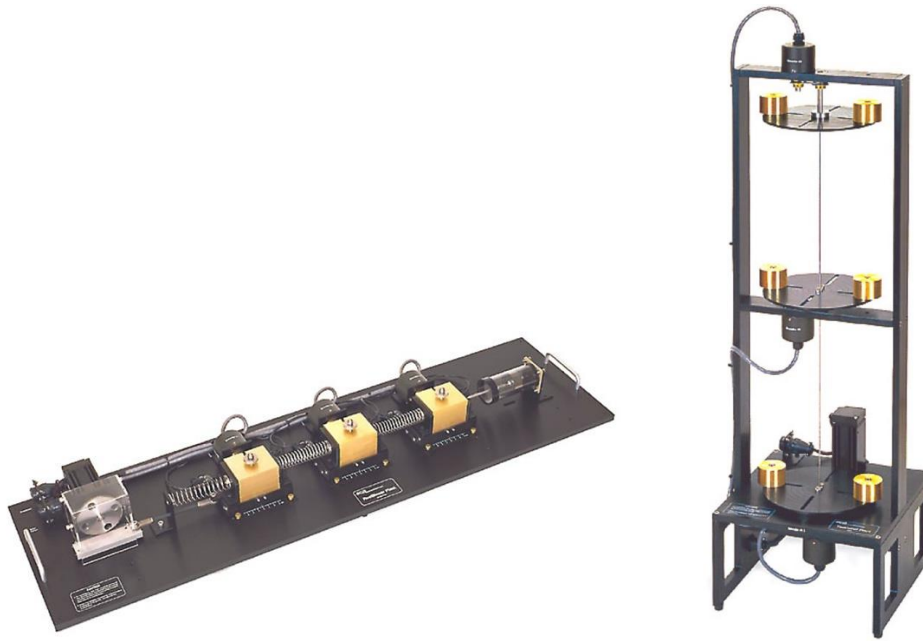


Figure 8 – Educational Control Systems (ECP) Products: Rectilinear System (Left) Torsional System (Right)

By contrast, virtual and remote laboratories like ones enabled by the Do-more PLC Emulator afford each student substantially more time to interact with the system and no time is required to ensure adequate safety training, as there are no physical components. While there is certainly some benefit to interacting with a physical system in person, the authors believe that incorporating a supplemental training mode such as a virtual PID tuning exercise using the Do-more PLC emulator can greatly enhance the student learning outcomes and the authors intend to investigate the academic merits in a subsequent study.

Concluding Remarks

This paper has described a practical implementation of a virtual teaching module using the Do-more PLC programming software, which is adapted for use in a mechanical engineering undergraduate course in Control systems. The Do-more PLC emulator is the latest in an ever-evolving category of simulation solutions designed to teach PLC programming and PID control. The virtual learning environment simplifies implementation by eliminating the hardware interfacing component, while retaining the same learning outcomes. Accessibility is improved for students with disabilities, and the software is provided to students by the college of engineering, reducing the student burden and allowing access to the learning materials while away from campus.

With this software solution, students can learn PLC programming and explore the quintessential Proportional Integral Derivative (PID) control architecture. In subsequent research, the authors intend to investigate the academic merits of the Do-more software by incorporating it into a Control Systems course and developing a course survey to assess the learning outcomes as compared to other sections of the same course without the PLC emulator.

Bibliography

1. Parker, Joey K. "Introduction to programmable logic controllers in a mechanical engineering instrumentation course." *Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition* 6:660:1 (2001).
2. Pereira, Carlos Eduardo, Suenoni Paladini, and Frederico Menine Schaf. "Control and automation engineering education: Combining physical, remote and virtual labs." *International Multi-Conference on Systems, Signals & Devices*. IEEE, 2012.
3. Özerdem, Özgür Cemal. "Design of two experimental setups for programmable logic controller (PLC) laboratory." *International Journal of Electrical Engineering Education* 53.4 (2016): 331-340.
4. Guo, Liping, and R. Pecun. "Design projects in a programmable logic controller (PLC) course in electrical engineering technology." *The technology interface journal* 10.1 (2009).
5. Saygin, C., Kahraman, F. "A Web-based programmable logic controller laboratory for manufacturing engineering education." *AMT* **24**, 590–598 (2004).
6. Lindsay, Euan and Liu, Dikai and Murray, Steve and Lowe, David. "Remote Laboratories in Engineering Education: Trends in Students' Perceptions," *18th Conference of the Australasian Association for Engineering Education*, Dec 2007.
7. C. A. Chung, "A cost-effective approach for the development of an integrated PC-PLC-robot system for industrial engineering education," in *IEEE Transactions on Education*, vol. 41, no. 4, pp. 306-310, Nov. 1998.
8. P. Pratumsuwan and W. Pongaen, "An embedded PLC development for teaching in mechatronics education," *2011 6th IEEE Conference on Industrial Electronics and Applications*, Beijing, 2011, pp. 1477-1481
9. C.-G. HABA, "Extending the Use of PLC Simulator Software in Student Laboratory Works", *Advances in Electrical and Computer Engineering*, vol. 10, pp. 84, 2010.
10. Grodotzki, Joshua, Tobias R. Ortelt, and A. Erman Tekkaya. "Remote and virtual labs for engineering education 4.0: achievements of the ELLI project at the TU Dortmund University." *Procedia Manufacturing* 26 (2018): 1349-1360.
11. Stefanovic, Miladin. "A LabVIEW-based remote laboratory experiments for control engineering education." *Computer Applications in Engineering Education* 19.3 (2011): 538-549.
12. <https://www.automationdirect.com/do-more/brx/software/simulator>
13. Franklin, Gene F., J. David Powell, Abbas Emami-Naeini, and J. David Powell. *Feedback control of dynamic systems*. Vol. 4. Upper Saddle River, NJ: Prentice Hall, 2002.
14. <https://www.famictech.com/en/Products/Automation-Studio/Educational-Edition/Technologies/PLC-Simulation>
15. DeGuglielmo, Nicolas P., Saurav MS Basnet, and Douglas E. Dow. "Introduce Ladder Logic and Programmable Logic Controller (PLC)." In *2020 Annual Conference Northeast Section (ASEE-NE)*, pp. 1-5. IEEE, 2020.
16. Rowe, Scott C., Cathleen I. Samson, and David E. Clough. "A Framework to Guide the Instruction of Industrial Programmable Logic Controllers in Undergraduate Engineering Education." *Education for Chemical Engineers* 31 (2020): 76-84.
17. Kumar, Lalit. "Comparison Between Mostly-Used Old-Fashion Method and New PITOPS Simulator for PID Controller Tuning Method." *Authorea Preprints* (2020).
18. <https://www.omega.com/en-us/resources/tuning-a-pid-controller>
19. <https://www.youtube.com/watch?v=zaX7R5BIsx8&feature=youtu.be>
20. <https://www.youtube.com/watch?v=QL6pwNPu-2k&feature=youtu.be>