Hovering a ping-pong ball: A demonstration setup for teaching PID control

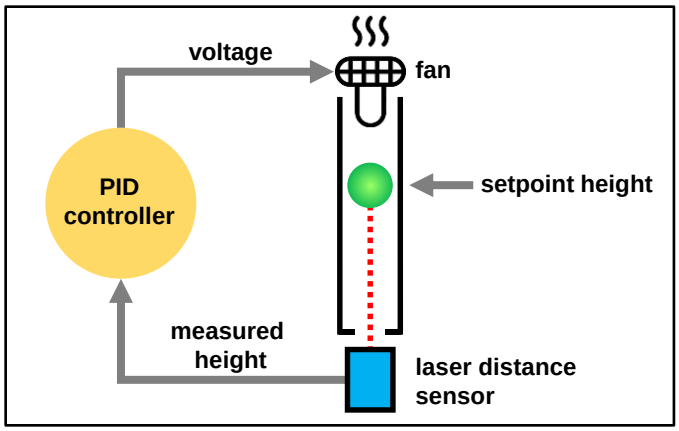
Christoph G. Salzmann,* Sophia M. Vecchi Marsh, Jinjie Li, Luca Slater

University College London, Department of Chemistry, 20 Gordon Street, London WC1H 0AJ, UK.

ABSTRACT

Proportional-Integral-Derivative (PID) controllers are essential in ensuring the stability and efficiency of numerous scientific, industrial and medical processes. However, teaching the principles of PID control can be challenging, especially when the introduction focusses on the underlying mathematical framework. To address this, we developed the PingPongPID - a visually engaging and interactive demonstration instrument designed to make the concepts of PID control more accessible to students. The PingPongPID features a colored ping-pong ball suspended within a transparent plastic tube by a fan whose voltage is controlled by a microcontroller running a PID algorithm. The ball's height is measured in real-time by a laser distance sensor and the system continuously adjusts the fan voltage to maintain a set target height. The PingPongPID also connects to a computer allowing real-time data logging and visualization through a custom-built Python graphical user interface (GUI). The process of obtaining the three PID gain constants using both the trial-and-error and Ziegler-Nichols methods can be illustrated very nicely with the PingPongPID. In summary, the PingPongPID serves as a powerful educational tool allowing students to explore both the conceptual and practical aspects of PID control before delving into its mathematical foundations. We provide a complete assembly and user guide for the PingPongPID as well as the codes for the microcontroller and the Python GUI.

GRAPHICAL ABSTRACT



KEYWORDS

General Public, Upper-Division Undergraduate, Graduate Education / Research, Chemical Engineering, Demonstrations, Laboratory Instruction, Hands-On Learning / Manipulatives, Inquiry-Based / Discovery Learning, Problem Solving / Decision Making, Laboratory Computing / Interfacing, Laboratory Equipment / Apparatus

INTRODUCTION

Being able to control environmental variables such as temperature, pressure, speed or position is essential for ensuring the stability and reliability of a wide range of applications.[1] Effective control mechanisms respond quickly to environmental changes or disturbances, preventing processes from becoming unstable, energy-wasting or even hazardous. Automatic control therefore greatly reduces the need for human interventions, making it an important component of the Internet of Things.[2]

The vast majority of control applications utilize a feedback-based algorithm known as the proportional-integral-derivative (PID) controller.[1] For instance, without PID control, drones could not maintain stable flight[3] and the navigation of large ships would be very difficult.[4] In medicine, PID controllers are used to regulate arterial blood pressure, muscle relaxation, the concentration of anesthetics and blood-glucose levels.[5] In the field of chemistry, PID controllers are widely used in applications such as pH regulation,[6] maintaining constant water levels in lab-scale tanks,[7] ensuring stable mixing conditions,[8] calorimetry,[9] achieving consistent chemical adsorption processes[10] as well as controlling temperature[11-13] and humidity.[14] Automatic process control is also critically important in the chemical industry.[15, 16]

To support education in this area, a spreadsheet-based tool has been developed for simulating and controlling chemical processes.[17] Electronics kits have also been designed to teach the controlling of chemical processes.[18]

Despite the importance of PID control in chemistry, and related scientific and technological fields, teaching its fundamental principles can be challenging, especially when the introduction begins with the underlying mathematical concepts. At our chemistry department, PID control is taught as part of a lecture course on "Designing Scientific Instruments" at Master's level. To make the concepts of PID control more accessible, we recently developed a simple demonstration setup. The idea was to showcase the capabilities of PID control to the students first during the lectures before explaining the mathematical framework. An ideal demonstration setup for PID control should display quick changes and be visually engaging.

DESIGN OF THE PINGPONGPID INSTRUMENT

To illustrate the principles of PID control, our idea was to hover a colored ping-pong ball in a transparent plastic tube at a setpoint height from the bottom. A photo of the complete PingPongPID instrument is shown in Figure 1. The (1) ping-pong ball can be moved up and down within a (2) one-meter long transparent acrylic tube by adjusting the voltage of a (3) fan from a hairdryer at the top. The voltage for the fan is supplied by a (4) benchtop power supply through (5) power cables. The height of the ping-pong ball is measured with a (6) laser distance sensor at the bottom of the tube. A (7) controller box is connected to the (8) distance sensor and the (9) power supply. Within the controller box, a microcontroller runs the PID algorithm (Arduino PID library by Brett Beauregard). Based on the measured height of the ping-pong ball, $h(t)$, the PID controller regulates the voltage of the fan, $V(t)$, automatically so that the ball moves to the setpoint height. For convenience, $h(t)$ and the setpoint height, $h_{SP}(t)$, are shown on the LED displays at the front of the controller box.

The controller box also provides a (10) USB port for connecting to a (11) computer for controlling the instrument and recording data. This can be done either with line commands in the Arduino IDE or, alternatively, with a Python Graphical User Interface (GUI) that we have written. Our GUI can also plot the height data in real time which is visually engaging if shown on a large projected screen during

the lectures. A complete assembly and user guide together with a list of components and required tools for building a PingPongPID is given in the Supporting Information. We also provide links for downloading the C++ code for the microcontroller and the Python GUI.

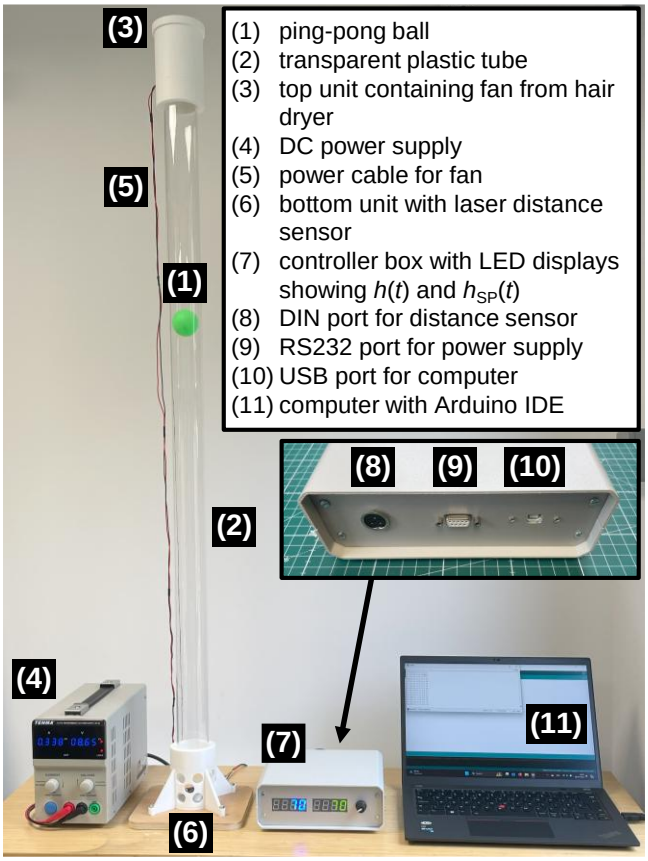


Figure 1. Photo of the fully assembled PingPongPID instrument. The various components are labeled and described in the legend. The inset image shows a back view of the controller box.

Before using the automatic PID control during the lecture, we found it insightful to first ask a student to achieve a certain setpoint height by manually controlling the voltage of the fan. It becomes apparent very quickly that this is a difficult if not impossible task and it would certainly not be practical to maintain the setpoint height for extended periods of time.

Having illustrated the need for PID control, the PingPongPID instrument was then started with automatic PID control. This can be done with the GUI or the Arduino IDE, or simply by pushing the knob at the front of the controller box. The colored ping-pong ball then rises to the setpoint height, which is 80 cm by default, within about three seconds and remains stable at this height. The recorded height data is shown in Figure 2(a). Figure 2(b) shows the required voltage of the fan as calculated by

the PID algorithm. Following a short period of operating at the maximal voltage, V_{max}, the voltage settles at a lower value, which is needed to maintain the height of the ping-pong ball. It can be seen that small corrections are necessary over time to achieve this. We also defined a minimal voltage, V_{min}, which is the voltage required for the fan to start turning. For demonstration purposes, $h_{SP}(t)$ can be changed by turning the knob at the front of the controller box. This illustrates that $h_{SP}(t)$ does not need to remain constant but that it can change as a function of time.

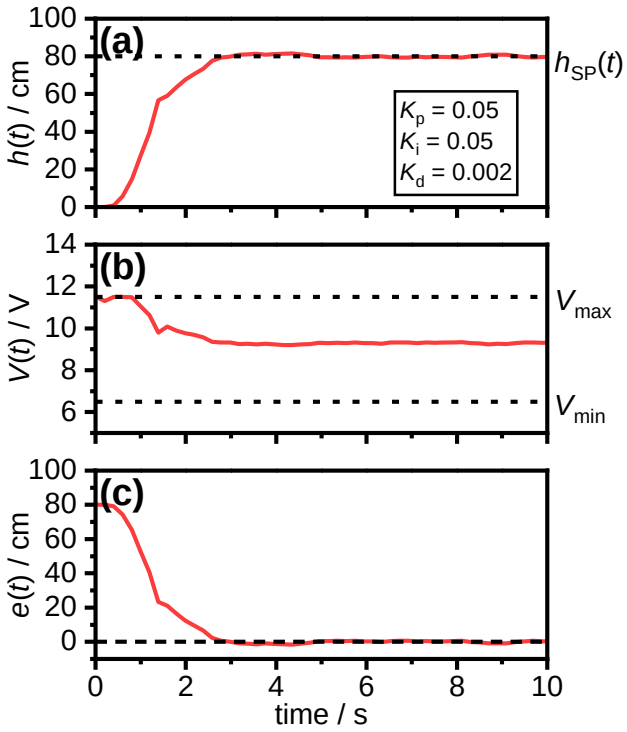


Figure 2. Operating the PingPongPID instrument with optimal gain constants. The PID algorithm uses the (a) measured height, $h(t)$, to calculate the (b) voltage of the fan, $V(t)$, required to raise the ping-pong ball to the setpoint height, $h_{SP}(t)$. In doing so, the (c) error function, $e(t)$, is brought as close as possible to zero. The setpoint height of 80 cm is indicate by a dashed horizontal line in (a). The dashed lines in (b) indicate the minimal and maximal voltages V_{min} and V_{max} respectively.

An important quantity in PID control is the error function, $e(t)$, which is calculated by subtracting $h(t)$ from $h_{SP}(t)$:

$$e(t) = h_{SP}(t) - h(t) \quad (1)$$

It turns out that the PID algorithm actually does not need to know $h(t)$ and $h_{SP}(t)$. Its sole purpose is to bring $e(t)$ as close as possible to zero, which means that $h(t)$ equals $h_{SP}(t)$. This process is shown in Figure 2(c). Following the rapid rise of the ping-pong ball, there is a small "overshoot" where $e(t)$ takes

slightly negative values. The ultimate aim of the PID control is to achieve the quickest possible rise while avoiding overshoots and oscillations around $h_{SP}(t)$ as much as possible.

At this stage is insightful to simulate an environmental disturbance by manually restricting the airflow into the tube. The PID algorithm will try its hardest to counteract the disturbance by increasing the voltage. It is also interesting at this point to remove the fan from the top of the acrylic tube. In a desperate attempt to keep $e(t)$ at zero, the PID controller will apply V_{max} to the fan. When the fan is placed back at the top of the tube, $V(t)$ will settle between V_{max} and V_{min} once the setpoint height is restored. It is important to highlight the dangers of a broken feedback loop. For example, removing the temperature sensor from a furnace may result in the furnace operating at full heating power with dangerous consequences.

PRINCIPLES OF PID CONTROL

Having illustrated the importance and usefulness of PID control, and how it operates under optimized conditions, it now makes sense to introduce the mathematical background. Figure 3 shows a schematic of the PID control process.

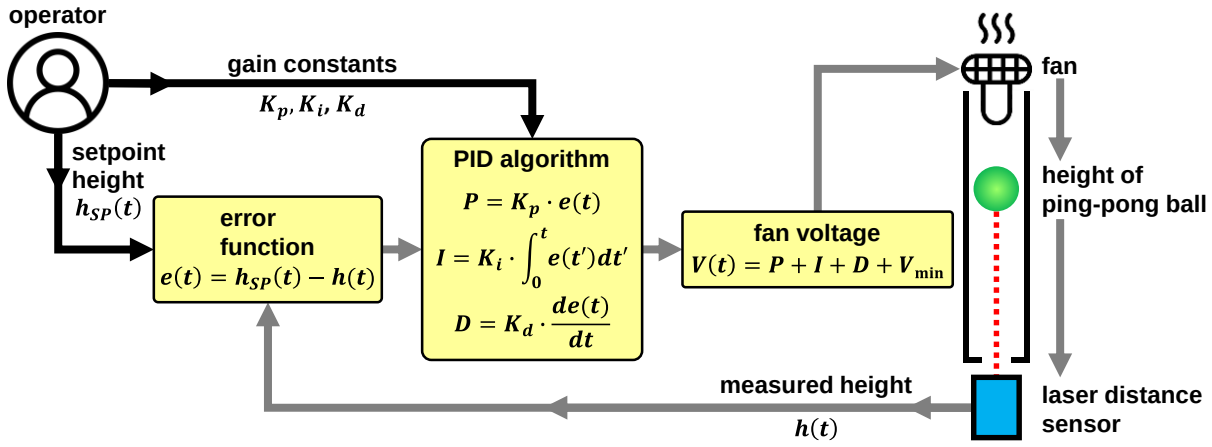


Figure 3. Schematic illustration of the PID control process for hovering a ping-pong ball at a user-defined setpoint height, $h_{SP}(t)$. The ping-pong ball can move up and down a transparent plastic tube with the help of a fan from a hairdryer. The error function, $e(t)$, is the difference between $h_{SP}(t)$ and the measured height of the ping-pong ball, $h(t)$. Using the PID gain constants, K_p, K_i and K_d, provided by the operator, the PID algorithm minimizes $e(t)$ by calculating the voltage needed to operate the fan, $V(t)$. In the optimal case, $e(t)$ approaches zero very quickly, which means that $h(t)$ and $h_{SP}(t)$ are identical. The user inputs are shown as black arrows whereas the automatic PID control process follows the closed loop indicated by the gray arrows.

The PID algorithm requires the current value of $e(t)$ as input and it will calculate $V(t)$ from the sum of the P (proportional), I (integral) and D (derivative) terms as well as V_{min} (if required by the application):

$$V(t) = P + I + D + V_{\mathrm{min}} \tag{2}$$

The P, I and D terms all depend on $e(t)$ in one way or another and contain gain constants called K_{p}, K_{i} and K_{d}. The gain constants determine how much weight is put on each of the three terms and therefore how much they contribute to $V(t)$:

$$P = K_{\mathrm{p}} \cdot e(t) \tag{3}$$

$$I = K_{\mathrm{i}} \cdot \int_0^t e(t')dt' \tag{4}$$

$$D = K_{\mathrm{d}} \cdot \frac{de(t)}{dt} \tag{5}$$

The way how each of the three terms contributes to hover the ping-pong ball and how the optimal values for the three gain constants are determined can be demonstrated nicely using the PingPongPID instrument in a next step.

TRIAL-AND-ERROR METHOD FOR TUNING THE PID GAIN CONSTANTS

To begin with, all three gain constants are set to zero. This means that the sum of the P, I and D terms is zero and hence $V(t) = V_{\mathrm{min}}$. In our case, V_{min} is set to 6.5 V which is the voltage at which the fan starts turning and producing airflow within the tube. Defining V_{min} is done simply for practical purposes since voltages between zero and V_{min} do not produce any airflow. In Figure 4, $h_{\mathrm{SP}}(t)$ is set to 80 cm. With all three gain constants set to zero, $V(t) = V_{\mathrm{min}}$ and $h(t)$ stays at zero.

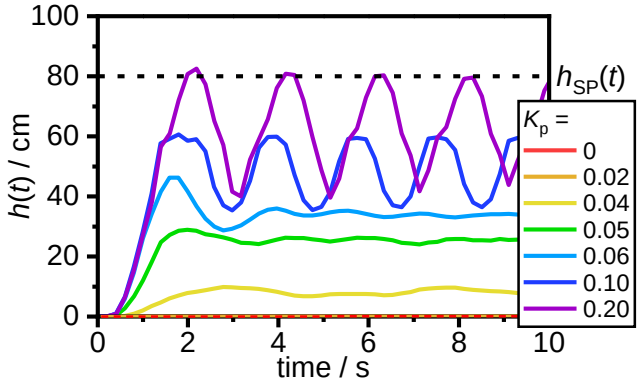


Figure 4. Exploring the effects of increasing the K_{p} gain constant while keeping $K_{\mathrm{i}} = 0$ and $K_{\mathrm{d}} = 0$. The setpoint height of 80 cm is indicate by a dashed horizontal line.

According to the trial-and-error method for tuning the gain constants, the value of K_p is optimized first while keeping K_i and K_d set at zero. This means that only the *P*-term is 'switched on' by increasing K_p. The contribution of the *P*-term to $V(t)$ is proportional to the current value of $e(t)$ with K_p as the constant of proportionality (eq. 3). In other words, the larger $e(t)$ and K_p, the larger the *P*-term. When the ping-pong ball reaches $h_{SP}(t)$, the *P*-term will be zero. Upon increasing K_p from zero to 0.02, the fan starts to turn. However, the airflow is not sufficient to achieve a lift-off of the ping-pong ball. A rising ping-pong ball is achieved with $K_p = 0.04$, yet $h(t)$ settles far below $h_{SP}(t)$. This type of steady-state error is typical for *P*-control where the voltage decreases to V_{min} as the setpoint height is approached. Increasing K_p further leads to oscillations which still fade at 0.06 but become undamped or 'critical' at $K_p = 0.10$. According to the trial-and-error method, the K_p value needs to be identified at which these critical oscillations first appear. This value of K_p is then divided by two, which is 0.05 in our case, and carried forward. This value of K_p achieves eliminating some of the error, yet a constant steady-state error is still observed.

As a side note, it should be mentioned that K_p may take negative values if there is an indirect proportionality between the controlled variable and the output of the PID algorithm. For example, for our IceBox instrument, increasing the voltage of a Peltier element reduces the temperature of a sample stage.[13]

Unlike the *P*-term, which considers the *present* error, the *I* term considers the *past* error. This is achieved mathematically by integrating over $e(t)$ from the start of the controlling experiment ($t = 0$) to the current time t (eq. 4). Please note that t' is simply a 'dummy' variable in eq. 4 required to carry out the integration. The *I*-term will keep moving more and more away from zero when $e(t)$ is not zero. As can be seen in Figure 2(c), $e(t)$ has positives values during the initial rise of the ping-pong ball. This means that the *I*-term will contribute an increasingly more positive contribution to $V(t)$ until $h_{SP}(t)$ is reached. How strongly the *I*-term contributes towards $V(t)$ is determined by the K_i gain constant. In Figure 5, it can be seen that increasing K_i from zero to 0.01 rises the ping-pong ball slowly up to the setpoint height. Increasing K_i further makes this process happen faster. Yet, above $K_i = 0.05$, overshoots are observed as the system becomes 'impatient' and ramps up the voltage too quickly. At $K_i = 0.20$, critical oscillations are observed as the system essentially switches straight from V_{max} to V_{min}

upon overshooting and from V_{min} to V_{max} upon undershooting. K_i = 0.05 appears to be a reasonable value with a relatively quick rise time to $h_{SP}(t)$ and a minimal overshoot. The fact that this value of K_i is the same as K_p determined earlier is a coincidence and not typically the case.

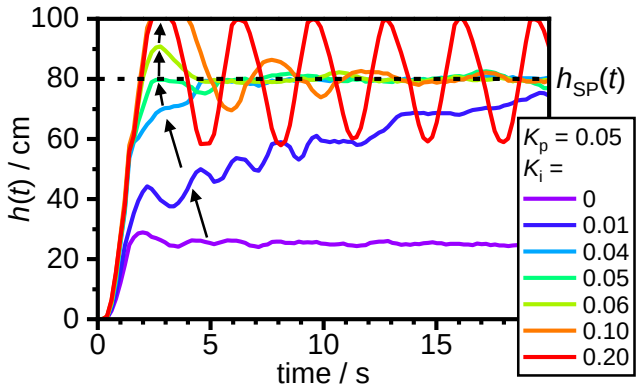


Figure 5. Bringing the K_i gain constant into the mix while keeping K_p at 0.05. The setpoint height of 80 cm is indicate by a dashed horizontal line.

The effect of the *I*-term can also be illustrated by first removing fan from the top of the acrylic tube and then starting the PingPongPID. The longer the fan stays away from the tube, the larger the *I*-term will grow. Upon placing the fan back, a massive overshoot will be observed and the ping-pong ball will only approach the setpoint height from above very slowly as the integration over the now negative values of $e(t)$ reduce the overall value of the integral.

Many control applications will work well using only PI control as in our case. Yet, in particular if the overshoots need to be avoided, it can be useful to introduce the *D*-term, which contains the K_d gain constant and the derivative of $e(t)$ with respect to time (eq. 5). If $h(t)$ and hence $e(t)$ change quickly, overshoots and undershoots are likely to happen. In a sense, the *D*-term tries to predict the *future* of the system. As can be seen in Figure 2(c), the slope of $e(t)$ is negative as $h(t)$ approaches $h_{SP}(t)$ upon startup. Increasing the value of the K_d gain constant therefore has a reducing effect on $V(t)$ thereby decreasing the rise time. This can be seen in Figure 6 when K_d is changed from zero to 0.002. Increasing K_d further, progressively slows down the rise time and again leads to oscillations as the system eventually overreacts. For the PingPongPID, good control would be obtained without the *D*-term and hence K_d = 0. However, it seems as if K_d = 0.002 stabilizes the height of the ping-pong ball at $h_{SP}(t)$ to some extent. So, the slightly decreased rise time is probably a price worth paying for the additional stability.

Particular care should be taken when choosing K_d if the recorded data is noisy. Noise components are by definition associated with fast erratic changes which can result in overall chaotic outputs from the PID algorithm because of the *D*-term. On the other hand, it should be noted that some systems will work well with only *PD* control.

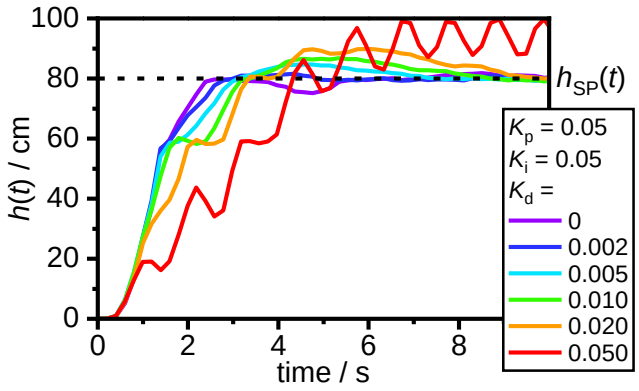


Figure 6. Effects of increasing the K_d gain constant with K_p = 0.05 and K_i = 0.05. The setpoint height of 80 cm is indicate by a dashed horizontal line.

In summary, the trial-and-error method illustrates the roles of the *P*, *I* and *D*-terms nicely and how the gain constants act to put different weights on the three terms: K_p is responsible for reducing the rise time, K_i eliminates steady-state error and K_d minimizes overshoots. While it is often straight forward to find reasonable values for the three gain constants with the trial-and-error method, seeking perfection may take a long time and require a lot of effort.

Having found a good set of gain constants, different setpoint heights can now be chosen. Figure 7(a) shows how $h(t)$ follows the selected setpoint heights of 90, 20, 40, 60, 80 and finally 30 cm. $V(t)$ required to realize these changes are shown in Figure 7(b). Every positive change in $h_{sp}(t)$ requires a positive voltage spike whereas decreases in $h_{sp}(t)$ go along with negative voltage peaks. Overall, the voltage required to stabilize each of the setpoint heights is actually independent of the actual heights themselves. This can be explained by the fact that a certain voltage will create the same airflow throughout the entire tube. However, for other controlling applications such a scenario may not necessarily always the case. For example, a furnace will require higher voltages to maintain higher temperatures. Nevertheless, the PID algorithm would be capable of dealing with both scenarios.

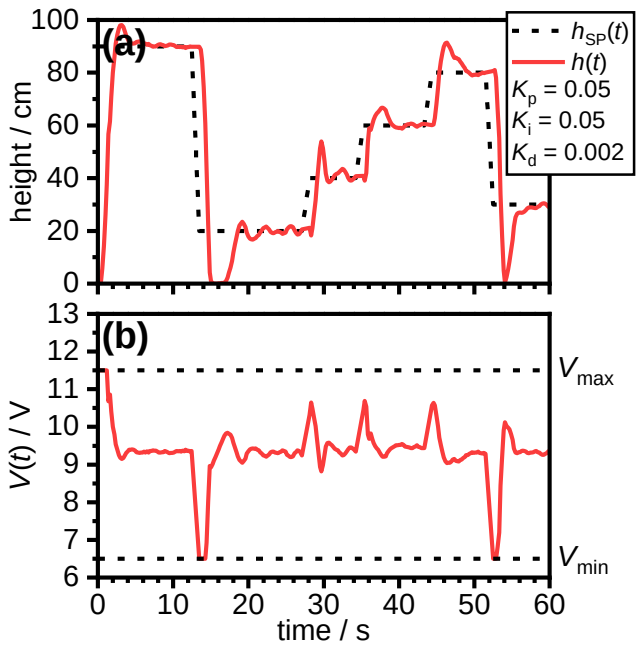


Figure 7. Changing the setpoint height with optimized gain parameters. The measured, $h(t)$, and setpoint height, $h_{sp}(t)$, are shown in (a) whereas (b) shows the corresponding voltage, $V(t)$, calculated by the PID algorithm.

In addition to changing setpoints, it is possible to define even more complex setpoint functions. For example, the data shown in Figure 8(a) was recorded using a sine wave as $h_{sp}(t)$ that oscillates between 10 and 90 cm with a period of 15 seconds. It can be seen that $h(t)$ follows $h_{sp}(t)$ very closely with just a little bit of a time delay. Our power supply requires a relay to be turned on above 10 V to provide the additional output power. This manifests as small steps around 60 cm, one of which is highlighted with an asterisk in Figure 8(a). Figure 8(b) shows the $V(t)$ required to follow the sine setpoint function. It is interesting to note that $V(t)$ also resembles a sine function, yet with a phase shift as the maximum voltage is required during the steepest rises of the setpoint function and the minimal voltage during the sharpest falls.

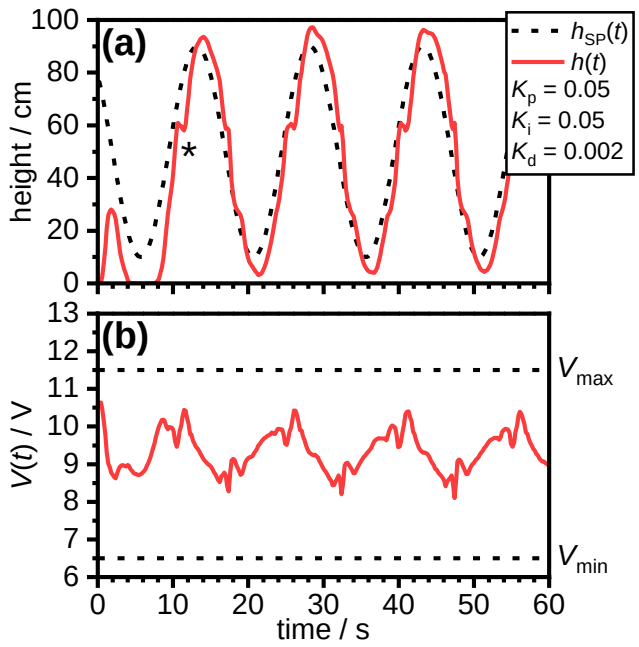


Figure 8. Using a sine function with a period of 15 seconds as the setpoint height function with optimized gain parameters. The measured, $h(t)$, and setpoint height, $h_{sp}(t)$, are shown in (a) whereas (b) shows the voltage, $V(t)$, calculated by the PID algorithm. The asterisk in (a) highlights a small step due to a relay in the power supply which repeats periodically and can also be seen in the $V(t)$ data in panel (b).

If the period of the sine wave was reduced further and further then at some point the lag in the system would prevent the ping-pong ball from being able to follow the setpoint function. To realize short periods, V_{max} could be increased to enable faster airflows. Furthermore, as an upgrade to the instrument, an additional fan could be used at the bottom of the tube to actively 'pull' the ping-pong ball down.

ZIEGLER-NICHOLS METHOD FOR DETERMINING GAIN CONSTANTS

A disadvantage of the trial-and-error method is that all three gain constants need to be optimized. The Ziegler-Nichols method provides a more straight-forward approach. Mathematically, it is designed to provide an initial overshoot whose amplitude reduces to a quarter for the second oscillation around the setpoint.[19] The advantage of the Ziegler-Nichols method is that only the effects of the K_p gain constant need to be explored while keeping K_i and K_d at zero. Specifically, K_p is increased until the critical oscillations are first observed. Using smaller increments than previously for the trial-and-error method, we found that this is observed at $K_p = 0.086$ as shown in Figure 9(a). This gain constant is called the ultimate gain, K_u. In a next step, the time period of the critical oscillations, T_u, is determined as 1.870 seconds as indicated by the vertical dashed lines in Figure 9(a).

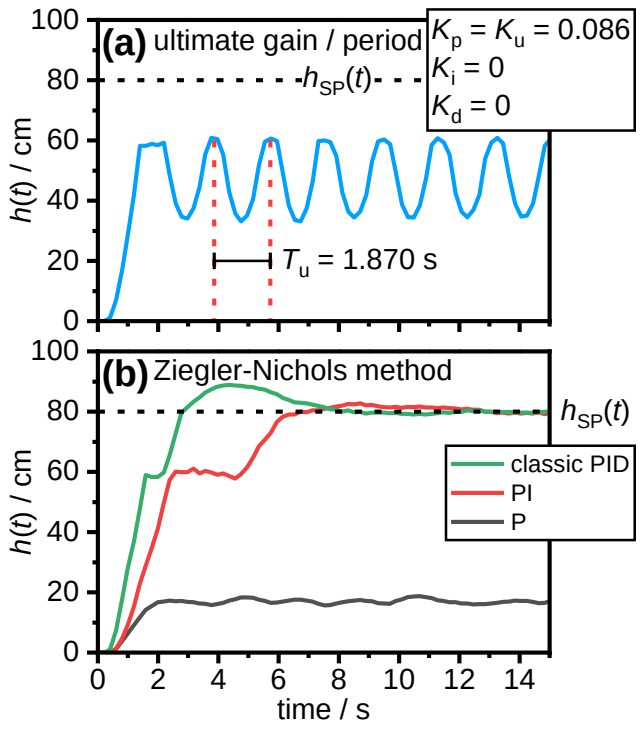


Figure 9. Determination of the gain constants using the Ziegler-Nichols method. (a) Critical oscillations are first observed for $K_p = K_u = 0.086$. The time period of the critical oscillations, T_u, is indicated by two dashed vertical lines. (b) Resulting height data using the gain constants calculated from K_u and T_u according to Table 1 and using a setpoint height of 80 cm.

The equations for calculating the gain constants from K_u and T_u according to Ziegler Nichols are shown in Table 1.[19] For PID control, this gives $K_p = 0.0516$, $K_i = 0.055$ and $K_d = 0.012$. Incidentally, the first two are quite similar as determined using the trial-and-error method. However, K_d is somewhat larger with a value of 0.012. As shown in Figure 9(b), this means that the initial rise is slowed down which in turn then leads to an overshoot due to the accumulating effect of the *I* term. Nevertheless, the Ziegler Nichols method predicts reasonable *PID* gain constants that lead to stable controlling conditions. The gain constants as calculated for Ziegler-Nichols *PI* control would also lead to stable control conditions for our system. However, *P* control alone would not work for the PingPongPID.

Table 1. Calculation of the gain constants according to the Ziegler-Nichols method using the ultimate gain, K_u, and ultimate period T_u.

control type	K_p	K_i	K_d
PID	$K_u/2$	$2K_p/T_u$	$K_pT_u/8$
PI	$K_u/2.2$	$1.2K_p/T_u$	-
P	$0.60K_u$	-	-

CONCLUSIONS

Using simple materials and electronic components, we built a versatile, visually appealing and, in our experience, portable demonstration setup for teaching PID control. Using the PingPongPID setup, it is possible to demonstrate the need for PID control at the beginning of a teaching session and to show how well it works with optimized gain constants. An introduction to the mathematical background can then follow as well as finding optimal values for the gain constants using the trial-and-error and Ziegler-Nichols methods. To conclude the teaching session, the students could be encouraged to design or perhaps even build their own instruments using PID control. Using PID control for minimizing energy consumption or preventing unnecessary CO_2 emissions could be very timely projects.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available on the ACS Publications website at DOI:

10.1021/acs.jchemed.XXXXXXX.

Assembly & user guide for the PingPongPID instrument (PDF)

AUTHOR INFORMATION

Corresponding Author

*E-mail: c.salzmann@ucl.ac.uk

ACKNOWLEDGMENTS

We thank the UCL Department of Chemistry for a contribution towards the costs for materials and electronic components.

REFERENCES

(1) Borase, R. P.; Maghade, D. K.; Sondkar, S. Y.; Pawar, S. N. A review of PID control, tuning methods and applications. *Int. J. Dyn. Control* **2021**, *9*, 818-827.

(2) Lynn, T.; Endo, P. T.; Ribeiro, A. M. N. C.; Barbosa, G. B. N.; Rosati, P. The Internet of Things: Definitions, Key Concepts, and Reference Architectures. In *The Cloud-to-Thing Continuum: Opportunities and Challenges in Cloud, Fog and Edge Computing*, Lynn, T., Mooney, J. G., Lee, B., Endo, P. T. Eds.; Springer International Publishing, 2020; pp 1-22.

(3) Lopez-Sanchez, I.; Moreno-Valenzuela, J. PID control of quadrotor UAVs: A survey. *Annu. Rev. Control* **2023**, *56*, 100900.

(4) Minorsky, N. DIRECTIONAL STABILITY OF AUTOMATICALLY STEERED BODIES. *J. Am. Soc. Nav. Eng.* **1922**, *34*, 280-309.

(5) Ramprasad, Y.; Rangaiah, G. P.; Lakshminarayanan, S. Robust PID Controller for Blood Glucose Regulation in Type I Diabetics. *Ind. Eng. Chem. Res.* **2004**, *43*, 8257-8268.

(6) Jacobs, O. L. R.; Hewkin, P. F.; While, C. Online computer control of pH in an industrial process. *IEE Proc.* **1980**; Vol. 127D, pp 161-168.

(7) Gawthrop, P. Self-tuning PID controllers: Algorithms and implementation. *IEEE Trans. Autom. Contr.* **1986**, *31*, 201-209.

(8) Tunjung, D.; Prajitno, P.; Handoko, D. Temperature and water level control system in water thermal mixing process using adaptive fuzzy PID controller. *J. Phys. Conf. Ser.* **2021**, *1816*, 012032.

(9) Li, J.; Parkes, M. A.; Salzmann, C. G. Extensive Screening and Performance Testing of Nucleating Agents for the Sodium Acetate Trihydrate Phase-Change Material. *Crystal. Growth Des.* **2024**, *24*, 8292-8300.

(10) Unguresan, M. L.; Muresan, V.; Abrudean, M.; Clitan, I.; Colosi, T.; Moga, D. PID Control of a Chemical Absorption Process. In *2015 20th International Conference on Control Systems and Computer Science*, 27-29 May 2015, **2015**; pp 104-111. DOI: 10.1109/CSCS.2015.119.

(11) Yamamoto, T.; Shah, S. L. Design and experimental evaluation of a multivariable self-tuning PID controller. *IEE Proc. Control Theroy Appl.* **2004**; Vol. 151, pp 645-652.

(12) Jamil, A. A.; Tu, W. F.; Ali, S. W.; Terriche, Y.; Guerrero, J. M. Fractional-Order PID Controllers for Temperature Control: A Review. *Energies*, **2022**; Vol. 15.

(13) Carlotta E. A. Blamey-Beccaria; Michael A. Parkes; Jinjie Li; Salzmann, C. G. Ice-nucleation properties of aluminum surfaces. *ChemRxiv* **2025**, 10.26434/chemrxiv-22025-26432hb26437h.

(14) Matamoros, M.; Gómez-Blanco, J. C.; Sánchez, Á. J.; Mancha, E.; Marcos, A. C.; Carrasco-Amador, J. P.; Pagador, J. B. Temperature and Humidity PID Controller for a Bioprinter Atmospheric Enclosure System. *Micromachines*, **2020**; Vol. 11.

(15) Juneja, P. K.; Sunori, S. K.; Sharma, A.; Sharma, A.; Pathak, H.; Joshi, V.; Bhasin, P. A Review on Control System Applications in Industrial Processes. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1022*, 012010.

(16) Bequette, B. W.; Ogunnaike, B. A. Chemical process control education and practice. *IEEE Control Syst.* **2001**, *21*, 10-17.

(17) Nachtigalova, I.; Finkeova, J.; Krbcova, Z.; Souskova, H. A spreadsheet-based tool for education of chemical process simulation and control fundamentals. *Comput. Appl. Eng. Educ.* **2020**, *28*, 923-937.

(18) Prabhu, G. R. D.; Urban, P. L. Elevating Chemistry Research with a Modern Electronics Toolkit. *Chem. Rev.* **2020**, *120*, 9482-9553.

(19) Ziegler, J. G.; Nichols, N. B. Optimum settings for automatic controllers. *Trans. Am. Soc. Mech. Eng.* **1942**, *64*, 759-769.