Preprints, 4th IFAC Conference on
Advances in Proportional-Integral-Derivative Control
Almería, Spain | June 12-14, 2024

Contribution to PID and PIDA Interactive Educational Tools

Katarína Žáková, **Jakub Matišák** and **Ján Šefčík**

Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Ilkovičova 3, 812 19, Bratislava, Slovakia
(e-mail: katarina.zakova@stuba.sk)

Abstract: The paper presents a simple educational tool that can be used for teaching basic concepts of PID and PIDA control. In the case of PID controller, its 3 basic forms are available: standard, parallel and series. The developed application is available via web browser and does not need any additional installation. The user can interact with the tool by choosing any non delayed linear dynamical system and by setting controller parameters. A comparison of the results from the performed simulation experiments is also available.

Keywords: PID, PIDA, computer based education, Python.

1. INTRODUCTION

Education at technical universities requires not only the acquisition of theoretical knowledge but also the development of practical skills. These can be gained by getting involved in various projects, solving practically oriented tasks and various experiments. The advantage is that it is not always necessary to carry out experiments on real devices in laboratories, but simulations and various interactive tasks can also be used for this purpose. It is encouraging that these tools are becoming available in the field of control education, too. The importance of using interactive tools in teaching control theory was already mentioned e.g. in Emami-Naeini and Diehl (1991) or later in Guzmán *et al.* (2013).

A number of such tools is developed in the Matlab/Simulink environment, which allows a relatively simple design of graphical user interfaces (GUI) for its applications. These can be run directly in Matlab. As it is evident from Johansson *et al.* (1998) such application were already developed more than 25 years ago. Further applications are described, for example, in Rossiter (2017), Rossiter (2023) or Koch *et al.* (2020).

Another approach was used in Ferrari and Visioli (2022). After development the interactive Matlab application, authors created a standalone executable file with Matlab compiler. This file can be used with only the Matlab runtime software package (a Matlab license is not required).

The approach of employing standalone subroutines to run interactive applications has been adopted also in Spain. This method has resulted in the creation of several interactive applications aimed at supporting the teaching of fundamental concepts in automatic control. (see e.g. Guzmán *et al.* (2023) or Guzmán *et al.* (2023)). These applications were not created in Matlab, but in the numerical environment Sysquake. The advantage is that the created applications can be run under all three most used operating systems (Windows, Mac, Linux).

The interesting toolbox for control design and analysis was developed in Julia language. The toolbox provides types for state-space, transfer-function, and time delay models, together with algorithms for design and analysis (Bagge Carlson *et al.*, 2021).

In this article we would like to present a different approach to providing interactive tools. Unlike most of the applications that are created, which can be used locally, we have focused on creating applications that would be delivered to users via a web page over the Internet. This approach does not require any additional installation on the user's own machine, nor does it require running exe files (standalone files), which many users may consider potentially dangerous.

A similar approach has already been taken, for example in Čech (2018). The described application is freely available via a web browser after registration. The developed tool uses noVNC HTML VNC client JavaScript library (Martin, 2022) that runs well in any modern browser including mobile browsers (iOS and Android). It is used to connect to the server where it seems (it is not mentioned in the paper) Matlab is running in the background of application.

We wanted to avoid Matlab for licensing reasons and decided to use Python in the backend of our application. It should be noted, however, that unlike some previous authors, who have already developed applications from several control topics due to years of experience in using the adopted technologies, in this article we are only trying to point out a possible way of providing additional tools.

The paper is organized as follows: the second section describes the controllers that are implemented in the developed tool; the third section describes the tool itself, and the fourth section describes the technologies used.

⋆ This work has been supported by the Slovak Grant Agency, grant KEGA 030STU-4/2021. This support is gratefully acknowledged.

© 2024 the authors. Accepted by IFAC for publication under a Creative Commons License CC-BY-NC-ND

2. CONTROLLERS

The developed interactive online tool tries to demonstrate functionality of various forms of considered PID and PIDA controllers. By doing this, we aim to improve the application's usability for students and users by eliminating the need to recalculate the algorithm into a different form.

2.1 PID controller

In the case of PID controller we consider its three forms: standard form, parallel form and series form. They are described in details in various textbooks or papers, e.g. in Åström and Hägglund (1995), Viteckova and Vitecek (2019) or Bingi *et al.* (2018).

Standard form The standard form of a PID controller is probably the most common and basic representation. It is expressed as:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right) \tag{1}$$

where $u(t)$ is the control output, $e(t)$ is the error signal, K_p is the controller gain, T_i is the integral time constant and T_d is the derivative time constant.

It can be represented by the transfer function

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \tag{2}$$

In the developed tool the derivative term was implemented as the ideal derivative filtered by the first-order system with the time constant Td/N

$$C(s) = Kp \left(1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N} s + 1} \right) \tag{3}$$

Parallel form The parallel form of a PID controller expresses the proportional, integral, and derivative terms separately

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \tag{4}$$

where K_p, K_i, and Kd are the proportional, integral, and derivative gains, respectively.

In this form, each term is written explicitly, making it easier to understand and implement each component independently.

The corresponding transfer function is

$$C(s) = K_p + K_i \frac{1}{s} + K_d s \tag{5}$$

The derivative term is often complemented by the first-order filter

$$C(s) = K_p + K_i \frac{1}{s} + K_d \frac{s}{T_f s + 1} \tag{6}$$

where T_f is filter time constant and in the presented tool it is calculated as K_d/N.

Another possibility is to use the common filter for all the terms. Then the transfer function of the controller is

$$C(s) = \frac{K_i + K_p s + K_d s^2}{s(T_f s + 1)} \tag{7}$$

The standard form can be recalculated to the parallel form by

$$K_i = \frac{K_p}{T_i}; \quad K_d = K_p T_d \tag{8}$$

Series form The transfer function of the series form can be written as follows

$$C(s) = K_c \left(\frac{1}{\tau_i s} + 1 \right) (\tau_d s + 1) \tag{9}$$

that can be rewrtitten to

$$C(s) = K_c \left[\left(\frac{\tau_d}{\tau_i} + 1 \right) + \frac{1}{\tau_i s} + \tau_d s \right] \tag{10}$$

Parameters τ_i and τ_d represents time constant of integral and derivative term and K_c is the controller gain.

In this form of PID controller the gain K_c influences all three components of the controller — proportional term, integral term, and derivative term, much like in the standard form. The main difference lies in the fact that in this case both the integral and derivative constants also influence the proportional action as it is evident from (10).

The series form of PID controller can be reformulated to to the parallel form by

$$K_p = K_c \left(\frac{\tau_d}{\tau_i} + 1 \right); \quad K_i = \frac{K_c}{\tau_i}; \quad K_d = K_c \tau_d \tag{11}$$

In the development of the interactive tool the first order filter for derivative term was used again

$$C(s) = K_c \left(\frac{1}{\tau_i s} + 1 \right) \left(\frac{\tau_d s}{\frac{\tau_d}{N} s + 1} + 1 \right) \tag{12}$$

2.2 PIDA controller

PIDA controller expands the possibilities of PID controller by adding the acceleration action. This type of controller was used already by Jung and Dorf (1996a) or Jung and Dorf (1996b). Comparison of PID and PIDA performance was done e.g. in Milanesi *et al.* (2022) and it was shown with different processes that the integrated absolute error can be significantly reduced (without lowering the robustness) for high-order processes.

The standard form of PIDA controller can be described by

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) + T_a \frac{d^2}{dt^2} e(t) \right) \tag{13}$$

The corresponding transfer function is

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s + T_a s^2 \right) \tag{14}$$

In the created tool, the derivative and acceleration components were filtered using a first-order system or a second-order system

$$C(s) = K_p \left[1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N} s + 1} + \frac{T_a^2 s^2}{\left(\frac{T_a}{\alpha} s + 1 \right)^2} \right] \tag{15}$$

Expressions T_d/N and T_a/α represent time constants of both filters.

Rise Time	Settling Time	Steady-state error	Overshoot	ISE	IAE	ITAE
0.928	5.403	0	5.957	10.882	17.370	18.159
1.303	4.010	0	1.647	13.227	21.158	24.472

Fig. 1. Online application running in web browser

3. INTERACTIVE TOOL

The developed tool is shown in Fig.1.

After selection the controller type (Fig.2) the user can define a system that should be controlled (Fig.1). It can be done using Matlab syntax that is generally known. It means that numerator and denominator are entered separately by coefficients of both polynomials that are separated by commas.

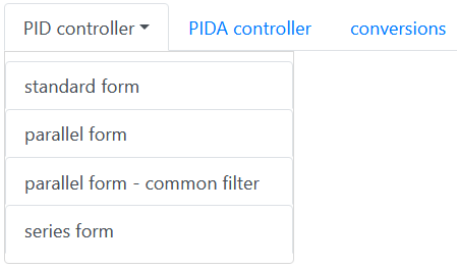

Fig. 2. Application menu

The user can set up the controller parameters and after clicking the *Simulate* button the data are computed in the application backend and visualized in the plot. Filter parameters N and α were set to the value 10 by default. However, the user is allowed to change it appropriately according to own preferences.

The performance of the controller is shown in the table below the graph and is evaluated by means of several quality criteria such as

- *rise time* - time from 10% to 90% of the steady-state value
- *settling time* - time to enter inside a error of 2%
- *overshoot* - percentage of the peak relative to steady value
- *steady-state error* - final offset or difference between the actual and desired output
- *ISE (Integral of Squared Error)* - measure of the integral of the squared error between the desired and actual outputs over time
- *IAE (Integral of Absolute Error)* - measure of the integral of the absolute value of the error between the desired and actual outputs over time
- *ITAE (Integral of Time-weighted Absolute Error)* - measure that penalizes errors that persist for a longer duration by introducing a time-weighting factor

If a new simulation is accompished, the plot in the graph is redrawn with new data.

After marking the checkbox *Hold on* the application enables to compare various settings of controller. When this checkbox is marked, each realized simulation is added into the graph, distinguished by a unique color. At the same time, corresponding quality criteria are added as a new row to the table below the graph. The color of the text corresponds to the color of the plot in the graph. It is possible to show and hide the selected simulation results

(both plot and quality criteria as well) by clicking on the plot legend.

If the user would like to see the control variable as well, it is enough to mark the *Control signal* checkbox.

The last part of the application is a conversion tool allowing to transform one form of PID regulator into another.

All formulae for mutual recalculation of individual forms of PID controllers can be found for example in Viteckova and Vitecek (2019).

4. TECHNICAL ASPECTS

The proposed solution is implemented as an interactive and user-friendly web-based application, emphasizing ease of use, minimalist design, and minimal system resource requirements.

The application's architecture is depicted in Fig.3. The web application is structured with distinct back-end and front-end components, referred to as the server and client, respectively.

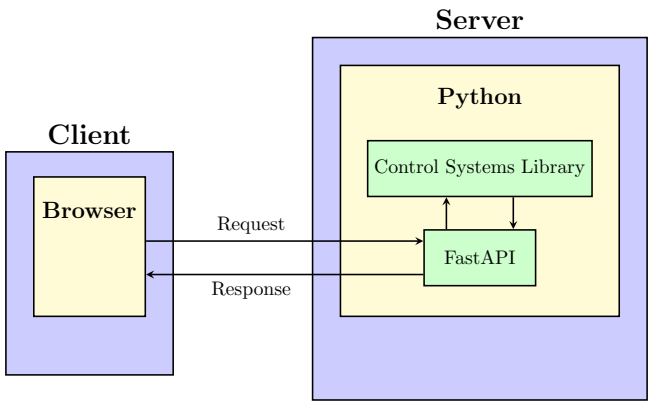

Fig. 3. Architecture of web application

The back-end is coded in Python and functions as a web service, offering a straightforward REST API. To achieve this, we employed the FastAPI micro-framework (Ramírez, 2022). The request coming from the client application includes all the parameters inputted by the user via the web browser. The computational tasks, like calculating step responses and simulating controllers, are handled using the Python Control Systems library (python-control.org, 2023). Subsequently, the back-end generates a response containing the computed data, which is then sent back to the front-end. Thanks to the FastAPI micro-framework, the computational aspect of the back-end can be utilized not only in web applications but also in other contexts such as command-line tools and scripts.

The frontend of the application was prepared in JavaScript that dynamically processed data computed in Python. The graphical plot is drawn using Plotly JavaScript library (Plotly, 2023).

5. CONCLUSION

The paper illustrated a simple control problem via interactive application built upon computations realized in Python. In the future, we aim to enhance the functionality of this application by incorporating additional features, enabling more thorough analysis of system properties and control structures.

We would also like to try to build similar application just on the client side that would enable better scalability of experiments.

Another possibility for improvement would be considering dynamical systems even with dead time delays.

REFERENCES

Åström, K. and Hägglund, T. (1995). *PID Controllers: Theory, Design, and Tuning*. ISA - The Instrumentation, Systems and Automation Society.

Bagge Carlson, F., Fält, M., Heimerson, A., and Troeng, O. (2021). Controlsystems.jl: A control toolbox in Julia. In *60th IEEE Conference on Decision and Control (CDC)*, 4847–4853. doi:10.1109/CDC45484.2021.9683403.

Bingi, K., Ibrahim, R., Karsiti, M.N., Hassan, S.M., and Harindran, V.R. (2018). A comparative study of 2DOF PID and 2DOF fractional order PID controllers on a class of unstable systems. *Archives of Control Sciences*, vol. 28(No 4), 635–682. doi:10.24425/acs.2018.125487.

Emami-Naeini, A. and Diehl, K. (1991). The role of interactive graphic tools in teaching control theory. *IFAC Proceedings Volumes*, 25(12), 53–58. doi:10.1016/S1474-6670(17)50088-9. IFAC Symposium on Advances in Control Education, Boston, MA, USA, 24-25 June 1991.

Ferrari, M. and Visioli, A. (2022). A software tool to understand the design of PIDA controllers. *IFAC-PapersOnLine*, 55(17), 249–254. doi:10.1016/j.ifacol.2022.09.287. 13th IFAC Symposium on Advances in Control Education (ACE).

Guzmán, J.L., Costa-Castelló, R., Berenguel, M., and Dormido, S. (2023). *Automatic Control with Interactive Tools*. Springer Nature.

Guzmán, J.L., Dormido, S., and Berenguel, M. (2013). Interactivity in education: An experience in the automatic control field. *Computer Applications in Engineering Education*, 21(2), 360–371.

Guzmán, J.L., Berenguel, M., Dormido, S., and Costa-Castelló, R. (2023). Using interactive tools to connect theory and practice. *IFAC-PapersOnLine*, 56(2), 9606–9611. doi:10.1016/j.ifacol.2023.10.265. 22nd IFAC World Congress.

Johansson, M., Gafvert, M., and Astrom, K. (1998). Interactive tools for education in automatic control. *IEEE Control Systems Magazine*, 18(3), 33–40. doi:10.1109/37.687617.

Jung, S. and Dorf, R. (1996a). Analytic PIDA controller design technique for a third order system. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 3, 2513–2518 vol.3. doi:10.1109/CDC.1996.573472.

Jung, S. and Dorf, R.C. (1996b). Novel analytic technique for PID and PIDA controller design. *IFAC Proceedings Volumes*, 29(1), 1146–1151. doi:10.1016/S1474-6670(17)57819-2. 13th World Congress of IFAC, San Francisco USA, 30 June - 5 July.

Koch, A., Lorenzen, M., Pauli, P., and Allgöwer, F. (2020). Facilitating learning progress in a first control course via Matlab apps. *IFAC-PapersOnLine*, 53(2), 17356–17361. doi:10.1016/j.ifacol.2020.12.2086. 21st IFAC World Congress.

Martin, J. (2022). noVNC - the open source VNC client. `https://novnc.com/info.html`.

Milanesi, M., Mirandola, E., and Visioli, A. (2022). A comparison between PID and PIDA controllers. In *IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–6. doi:10.1109/ETFA52439.2022.9921724.

Plotly (2023). Plotly JavaScript open source graphing library. `https://plotly.com/javascript/`.

python-control.org (2023). Python control systems library. `https://python-control.readthedocs.io/en/latest/`.

Ramírez, S. (2022). FastAPI. `https://fastapi.tiangolo.com/`.

Rossiter, J. (2017). Using interactive tools to create an enthusiasm for control in aerospace and chemical engineers. *IFAC-PapersOnLine*, 50(1), 9120–9125. doi:10.1016/j.ifacol.2017.08.1713. 20th IFAC World Congress.

Rossiter, J. (2023). Developing a novel MATLAB control toolbox for encouraging student engagement. In *6th Experiment International Conference (expat'23), Portugal*.

Čech, M. (2018). Web-based fractional PID controller design: www.pidlab.com. *IFAC-PapersOnLine*, 51(4), 563–568. doi:10.1016/j.ifacol.2018.06.155. 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control (PID).

Viteckova, M. and Vitecek, A. (2019). Standard, parallel and series two degree of freedom PID controllers. In *20th International Carpathian Control Conference (ICCC)*, 1–4. doi:10.1109/CarpathianCC.2019.8765932.