Fırat Üniversitesi Müh. Bil. Dergisi
38(1), 245-254, 2026

Araştırma Makalesi
https://doi.org/10.35234/fumbd.1797151

The Online PID/PIDA Simulator

Gökay GONCAGÜL[1], Metin HATUN[2], Fahri VATANSEVER[3*]
[1,2,3] Department of Electrical-Electronics Engineering, Faculty of Engineering, Bursa Uludağ University, Bursa, Türkiye
[1] gokaygoncagul23@gmail.com, [2] metinh@uludag.edu.tr, [3] fahriv@uludag.edu.tr

(Geliş/Received: 04/10/2025 Kabul/Accepted: 06/11/2025)

Abstract: Nowadays, computer tools are used widely in education as well as in many other fields. Especially in engineering, it is extremely important to support complex theoretical subjects with practical applications. In cases where practical applications are limited or difficult to implement, simulations are used. In this study, a new online simulator was designed for Proportional-Integral-Derivative-Acceleration (PIDA), which is the popular controller recently. The designed simulator provides online access, supporting users in comprehending the related theoretical knowledge's and to examine the time and frequency domain system responses of the controllers individually and comparatively according to the parameters they enter. Thus, it provides both theoretical information to students and technical staff interested in this field and enables them to easily, quickly, and effectively simulate practical applications.

Keywords: PID, PIDA, controller, online simulator.

Çevrimiçi PID/PIDA Simülatörü

Öz: Günümüzde bilgisayar araçları birçok alanda olduğu gibi eğitim alanında da yaygın olarak kullanılmaktadırlar. Özellikle mühendislikte, karmaşık teorik konuların pratik uygulamalarla desteklenmesi son derece önemlidir. Pratik uygulamaların sınırlı veya zor olduğu durumlarda simülasyonlardan yararlanılmaktadır. Bu çalışmada, son zamanlarda popüler bir denetleyici olan Oransal-İntegral-Türev-İvme (PIDA) için yeni bir çevrimiçi simülatör tasarlanmıştır. Tasarlanan simülatör çevrimiçi erişim sağlayarak, kullanıcıların ilgili teorik bilgileri kavramasına ve denetleyicilerin girdikleri parametrelere göre zaman ve frekans alanı sistem tepkilerini ayrı ayrı ve karşılaştırmalı olarak incelenmesine yardımcı olmaktadır. Böylece, bu alana ilgi duyan öğrencilere ve teknik elemanlara hem teorik bilgi sunmakta hem de pratik uygulamaları kolay, hızlı ve etkili bir şekilde benzetimlerinin yapılmasını sağlamaktadır.

Anahtar kelimeler: PID, PIDA, denetleyici, çevrimiçi simülatör.

1. Introduction

Simulations play an important role in understanding and comprehending complex theoretical knowledge, systems and concepts. Interactive simulations have many advantages such as visualizing abstract concepts and making them concrete, bridging theory and practice, eliminating laboratory time constraints and providing time efficiency, reducing laboratory installation, operation and maintenance costs, eliminating physical risks in laboratories, easily realizing and repeating different application scenarios and reinforcing learning with instant feedback. To achieve these benefits, simulations that enhance conceptual understanding, encourage active learning, facilitate safe and controlled experiments, provide time efficiency, and establish a connection with the real-world, many simulators have been designed in different fields, with different features and capabilities, which can work offline and online. Thanks to these, concepts and applications such as computer-aided engineering, computer simulation, computer-aided education, computer-aided teaching, computer-aided instruction, distance education, and distance learning are becoming widespread, especially in engineering education.

In parallel with the developments in computer technology, the number of interactive teaching applications has been increasing rapidly in recent years. In this context, applications such as interactive software tools, teaching portals, websites, and virtual laboratories are being developed for various scientific fields. In literature, the number of interactive educational software tools or teaching applications has been increasing rapidly in recent years. Some of these are studies in the field of control systems that combine theory with practice, enhancing student/user motivation, comprehension, and understanding.

Proportional-Integral-Derivative (PID) controllers are among the most widely used feedback controllers in industry due to their simplicity, effectiveness, simple structure, the availability of various tuning rules, easy design

[*] Corresponding author: fahriv@uludag.edu.tr. ORCID Information: [1] 0009-0006-9352-1172, [2] 0000-0003-0279-5508, [3] 0000-0002-3885-8622

and implementation, and satisfactory performance in many applications. However, there are many industrial applications where the performance of a standard PID controller can be insufficient. In these cases, modifications and improvements can be made to the structure of these controllers by leveraging PID knowledge's to improve their performance. One of these modifications and improvements is the Proportional-Integral-Derivative-Acceleration (PIDA) controller, also called PIDD2 or PIDD^2 controller [1, 2].

There are many studies on PID controllers in the literature. However, there are few simulator studies for PIDA controllers [3-10]. The existing PIDA simulators available have limited features, and many of them work offline. The aim of this paper is to design an online PIDA simulator with more comprehensive features. An interactive simulator for many versions of PID controllers has been developed in [3]. With this simulator, single and comparative analyses of closed-loop systems in time and frequency domains can be performed. There is also interactive graphical user interface (GUI) tool in MATLAB called "pidTuner" (for PID controller) and "sisotool" (for single-input/single-output compensator) [4]. In a different study, the simulation tool for the disturbance analysis of PID with high derivative order was developed by Bisták [5]. In another study, which included a simulation tool developed using MATLAB and capable of working offline/online designed by Bisták and Huba provide a comparison of responses in PID controller with high derivative order with the first order time-delayed controlled system influenced by a noise [6]. In addition, Bistak and Huba developed a higher order-PID (HO-PID) designer software tool [7]. In addition, there are also interactive tools in the literature developed by Ferrari and Visioli for PIDA analysis [8] and by Žáková et al. for teaching PID and PIDA [9]. Finally, a software tool that performs single and comparative simulations of control systems in time and frequency domain with different types of PIDA controllers was developed in [10].

Most of the simulators or software tools in the literature work offline. If the offline software is not free and open source, relevant licensed simulator software will be needed. Online simulators, on the other hand, can run without needing any special software/platform and time or physical space limitations. The main purpose of this study is to design an interactive simulator that is accessible to everyone, has a user-friendly interface, can be used for both educational and application/practical purposes, and works online. In this study, a new web-based simulation tool was designed that analyze especially PIDA controller which also includes PID controller using the Python programming language. In user-defined systems, time and frequency domain responses and parameters are presented in a web environment both numerically (rise time, settling time, overshoot, steady error, gain margin, phase margin) and graphically (step response, pole-zero map, root locus, Bode diagram, Nyquist diagram, Nichols chart) using selected controllers with specified coefficients.

This paper is organized as follows: In Section 2, PIDA controllers are summarized and the design steps of online PID/PIDA simulator is explained. In Section 3 some applications are given and Section 4 contains conclusions.

2. Materials and Methods

2.1. PID/PIDA controllers

PID is the traditional controller widely used in industrial control applications. In some cases, PIDA controllers are used to control higher order systems. This controller, which includes the second-order derivative (acceleration) in addition to the PID controller, is used to obtain better control performance than the PID control of systems with orders higher than two. The PIDA controller adds three zeros to the loop of the controlled system, and thus, providing greater phase lead than the PID controller to control more complex systems than the second order, further improving the transient response of the system and providing greater robustness against disturbance inputs [1, 2]. The some properties, block diagrams, time and s domain equations of classical PID and PIDA controllers are summarized in Table 1.

2.2. The design steps of online PID/PIDA simulator

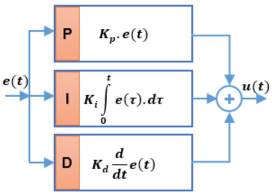
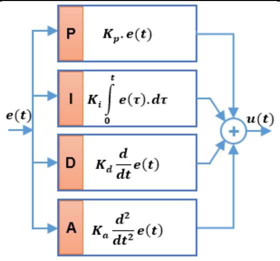
The designed simulator is implemented using the Python programming language and is built on the Flask web framework. The technologies and tools used in design of the simulator are summarized below:

- Python programming language [11]: Python is an open source and versatile programming language widely used in engineering applications. Thanks to its extensive and comprehensive library support, it offers effective solutions in many areas. The basic libraries used in the design are given in Table 2.
- HTML/CSS/JavaScript: The frontend of the simulator is developed with HTML/CSS/JavaScript. HTML, CSS and JavaScript are the basic technologies used in the design of web interfaces. HTML determines the

page structure, CSS provides design and visualization, and JavaScript enables dynamic and interactive operations on the page.

- Heroku: Heroku is a cloud-based platform that allows applications to be published quickly and easily on the internet [16].

Table 1. The PID and PIDA controllers.

Components / Structure		Proportional (**P**) + Integral (**I**) + Derivative (**D**)	Proportional (**P**) + Integral (**I**) + Derivative (**D**) + Acceleration (**A**)
Block diagram		 P $K_p.e(t)$ I $K_i\int_0^t e(\tau).d\tau$ D $K_d\frac{d}{dt}e(t)$ $e(t)$ $u(t)$	 P $K_p.e(t)$ I $K_i\int_0^t e(\tau).d\tau$ D $K_d\frac{d}{dt}e(t)$ A $K_a\frac{d^2}{dt^2}e(t)$ $e(t)$ $u(t)$
Equation	*time*	$u(t) = K_p.e(t) + K_i\int_0^t e(\tau)d\tau + K_d\frac{d}{dt}e(t)$	$u(t) = K_p.e(t) + K_i\int_0^t e(\tau)d\tau + K_d\frac{d}{dt}e(t) + K_a\frac{d^2}{dt^2}e(t)$
	s	$U(s) = \left(K_p + \frac{K_i}{s} + K_d s\right)E(s)$	$U(s) = \left(K_p + \frac{K_i}{s} + K_d s + K_a s^2\right)E(s)$
Complexity		Moderate (simpler)	Slightly more complex
Tuning	*rules*	Many	Fewer
	difficulty	3 parameters	4 parameters
Common use		General-purpose control systems	Systems requiring smoother/sensitive control

The general effect of parameters on a closed-loop system response

Parameters	*Rise time*	*Settling time*	*Overshoot*	*Steady-state error*	*Stability*
K_p	Decrease	Small effect (increase)	Increase	Decrease	Deteriorate
K_i	Decrease	Increase	Increase	Decrease (Eliminate)	Deteriorate
K_d	Small effect (decrease)	Decrease	Decrease	No effect	Improve
K_a	Small effect (decrease)	Decrease	Decrease	No effect	Improve

The designed steps of online simulator are given in Figure 1. In this study, Python was used as the main software language to perform the analysis and calculations. HTML, CSS and JavaScript technologies were used in the frontend part to create a user-friendly interface where users can interact with the online PID/PIDA simulator. Hosting the developed simulator on Heroku increases the accessibility of the application by enabling users to access it from different devices and locations. The platform's auto-scaling, security and continuous integration features support the performance and sustainability of the simulator.

CODING
WEB DEVELOPMENT
DEPLOYMENT

Figure 1. The design steps of simulator.

Table 2. The basic used libraries in Python.

Library	**Description**	**Ref.**
control	Python-control provides powerful tools for control systems.	[12]
numpy	NumPy is the basic Python library for scientific computing.	[13]
matplotlib	Matplotlib is a comprehensive library for visualization.	[14]
flask	Flask is a lightweight Web Server Gateway Interface web application framework.	[15]

2.3. Use of online PID/PIDA simulator

The designed web-based PID/PIDA simulator has been developed to enable users to access and utilize the tool from any device with an internet connection, without the need for additional software installation. Designed as a platform-independent and user-friendly environment, the simulator allows users to perform control system analysis both in the time and frequency domains, as well as to conduct comparative studies between different controller structures, notably PID and PIDA. Controllers such as P, PI, PD can also be defined by setting/entering the relevant parameters to zero. Through this simulator, users can mainly (Figure 2):

- Define transfer functions for both the system/plant ($G_p(s)$) and feedback path ($H(s)$),
- Select the controller type (PID or PIDA),
- Set control parameters/gains (K_p, K_i, K_d for PID; additionally K_a for PIDA),
- Choose and visualize various control system analysis tools, including step response, root locus, pole-zero map, Bode and Nyquist diagrams and Nichols chart,
- Perform single and comparative analyses.

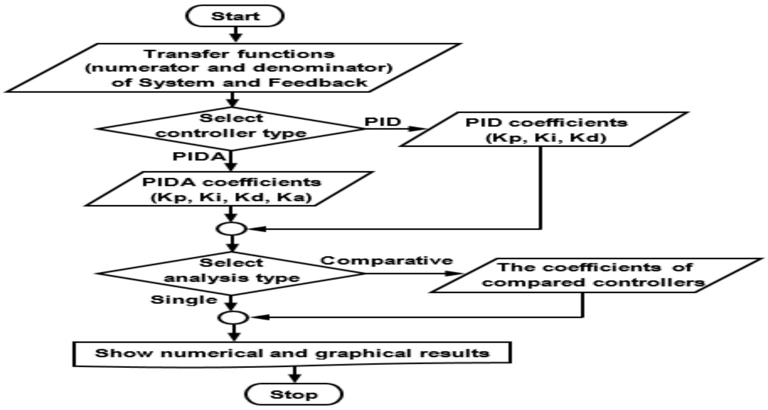
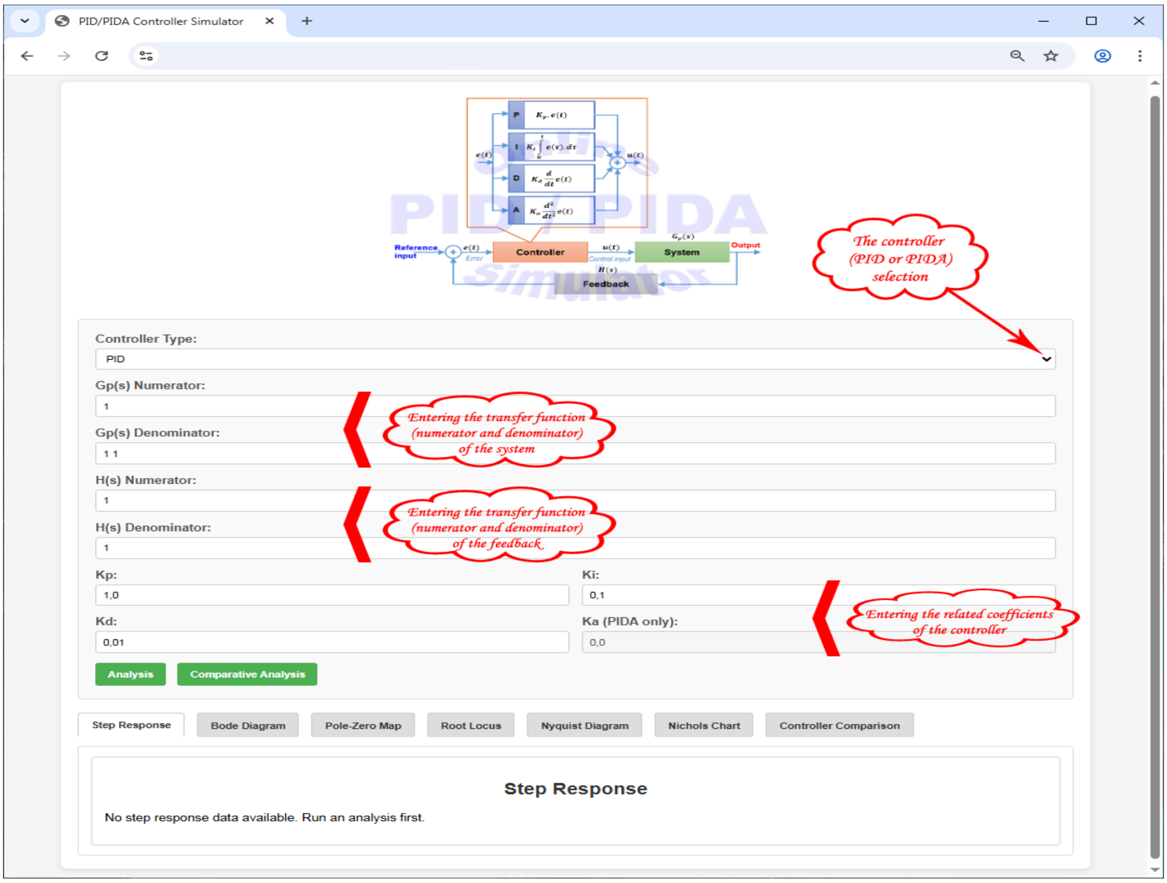

Start
Transfer functions (numerator and denominator) of System and Feedback
Select controller type
PID
PID coefficients (Kp, Ki, Kd)
PIDA
PIDA coefficients (Kp, Ki, Kd, Ka)
Select analysis type
Comparative
The coefficients of compared controllers
Single
Show numerical and graphical results
Stop

Figure 2. The flowchart for using designed simulator.


PID/PIDA Controller Simulator
PID/PIDA
Simulator
P
I
D
A
$K_p.e(t)$
$K_i\int_0^t e(\tau).d\tau$
$K_d\frac{d}{dt}e(t)$
$K_a\frac{d^2}{dt^2}e(t)$
$e(t)$
$u(t)$
Reference input
Error
Controller
Control input
System
$G_p(s)$
Output
$H(s)$
Feedback
The controller (PID or PIDA) selection
Entering the transfer function (numerator and denominator) of the system
Entering the transfer function (numerator and denominator) of the feedback
Entering the related coefficients of the controller

Controller Type: PID

Gp(s) Numerator: 1

Gp(s) Denominator: 1 1

H(s) Numerator: 1

H(s) Denominator: 1

Kp: 1,0

Ki: 0,1

Kd: 0,01

Ka (PIDA only): 0,0

Analysis
Comparative Analysis
Step Response
Bode Diagram
Pole-Zero Map
Root Locus
Nyquist Diagram
Nichols Chart
Controller Comparison

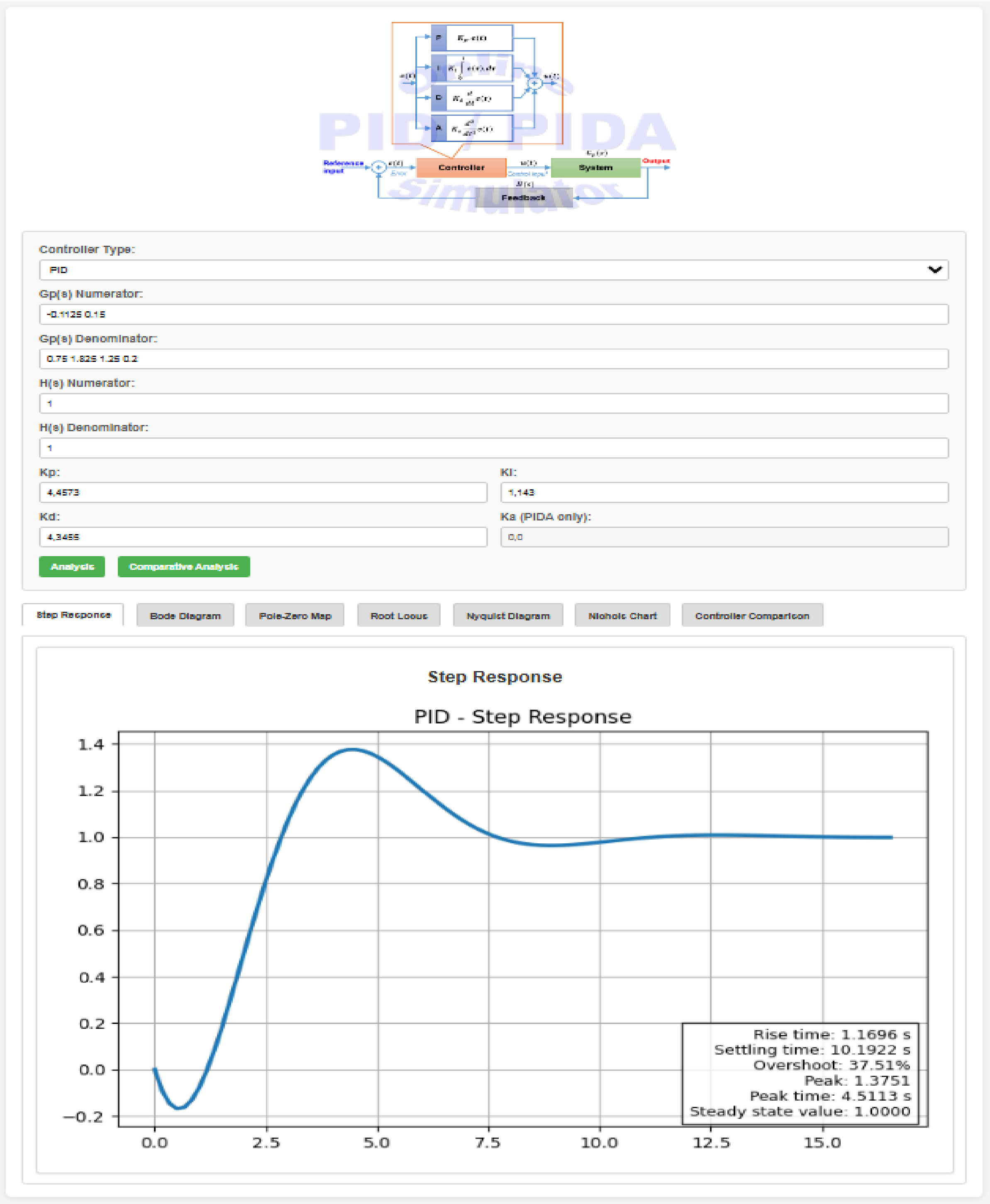
Step Response

No step response data available. Run an analysis first.

Figure 3. The main page of designed online simulator.

3. Applications with Online PID/PIDA Simulator

The main page of designed online simulator is given in Figure 3. This page includes the block diagram of the control system, controller types, inputs of the numerator and denominator coefficients of transfer functions, and single and comparative analysis selections. Additionally, this page includes tabs for selecting the desired numerical and graphical results.



Controller Type: PID

Gp(s) Numerator: -0.1125 0.15

Gp(s) Denominator: 0.75 1.825 1.25 0.2

H(s) Numerator: 1

H(s) Denominator: 1

Kp: 4,4573

Ki: 1,143

Kd: 4,3455

Ka (PIDA only): 0,0

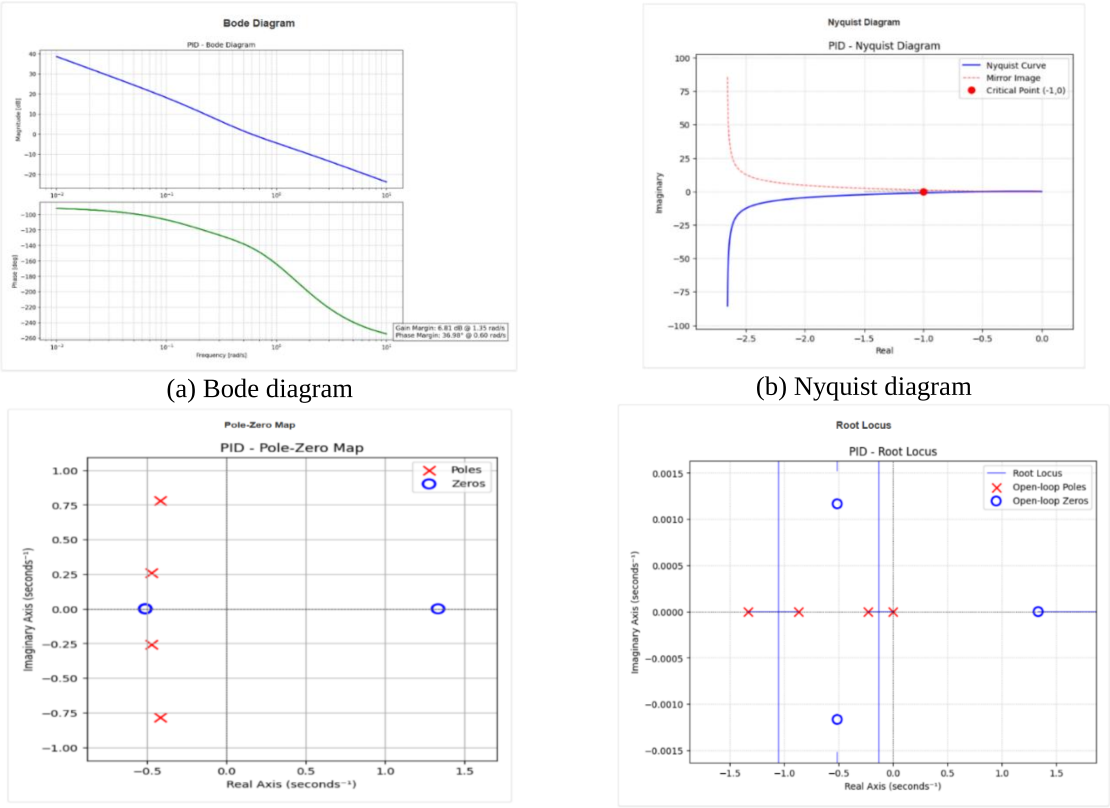
```
Rise time: 1.1696 s
Settling time: 10.1922 s
Overshoot: 37.51%
Peak: 1.3751
Peak time: 4.5113 s
Steady state value: 1.0000
```

Figure 4. The main screenshot of first application.

The first application with designed online simulator is controlling the temperature of an electric furnace with PID. The transfer function of system was given in Equation 1 [17]:

$$G(s) = \frac{-0.1125s+0.15}{0.75s^3+1.825s^2+1.25s+0.2} \tag{1}$$

The homepage screenshot and graphs containing some results are given in Figure 4-5, respectively. Figure 4 shows the step response of the system and its related performance metrics. In this way, the user can easily see the performance of the defined controller on the system. They can also quickly evaluate the effects of changing the controller coefficients. Figure 5 includes the Bode and Nyquist diagrams of the system, the root locus and the pole-zero map. These plots provide detailed information about the closed-loop system.



(a) Bode diagram

(b) Nyquist diagram

(c) Pole-zero map

(d) Root locus plot

Figure 5. Some screenshots of first application.

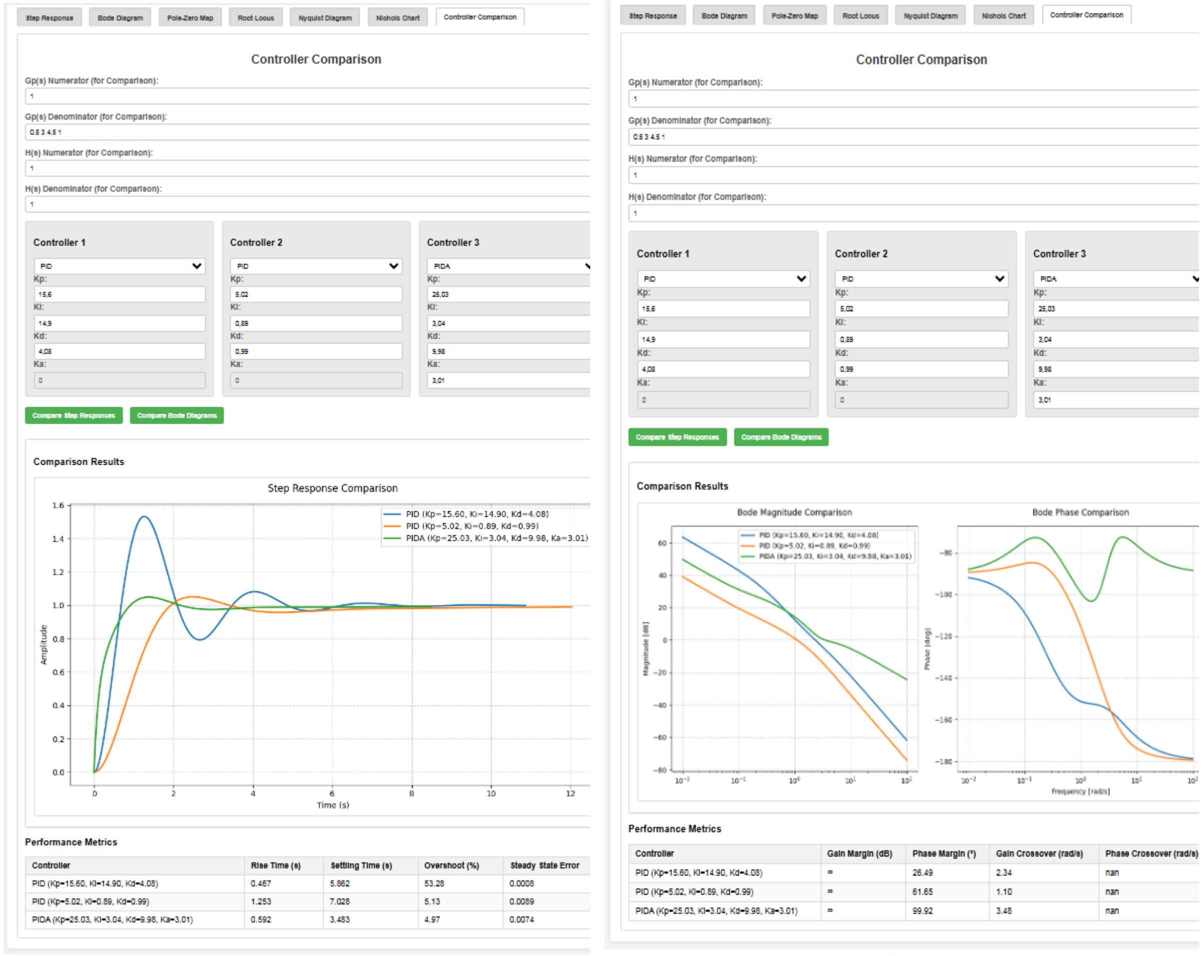
As the second application, a comparative analysis of the interacting three-tank liquid-level control system was performed [18]. The transfer function of system was given in Equation 2.

$$G(s) = \frac{1}{0.5s^3+3s^2+4.5s+1} \tag{2}$$

The transfer functions of the compared controllers were given in Equation 3.

$$\left.\begin{aligned} C_{PID_1}(s) &= 15.6 + \frac{14.9}{s} + 4.08s \\ C_{PID_2}(s) &= 5.02 + \frac{0.89}{s} + 0.99s \\ C_{PIDA}(s) &= 25.03 + \frac{3.04}{s} + 9.98s + 3.01s^2 \end{aligned}\right\} \tag{3}$$

The result screenshots are given in Figure 6. Figure 6a shows the comparative step response of the system with three controllers. Performance metrics are also included in the associated table. According to obtained results with online simulator, the superiority of the used PIDA controller to other PID controllers is clearly seen. Figure 6b shows the comparison of Bode diagrams (magnitude and phase). The frequency domain analysis results of simulator clearly show that the phase contribution of the PIDA controller is greater than the PID controllers.



Controller Comparison

Gp(s) Numerator (for Comparison): 1

Gp(s) Denominator (for Comparison): 0.5 3 4.5 1

H(s) Numerator (for Comparison): 1

H(s) Denominator (for Comparison): 1

Controller 1	Controller 2	Controller 3
PID	PID	PIDA
Kp: 15,6	Kp: 5,02	Kp: 25,03
Ki: 14,9	Ki: 0,89	Ki: 3,04
Kd: 4,08	Kd: 0,99	Kd: 9,98
Ka: 0	Ka: 0	Ka: 3,01

Comparison Results

Performance Metrics

Controller	Rise Time (s)	Settling Time (s)	Overshoot (%)	Steady State Error
PID (Kp=15.60, Ki=14.90, Kd=4.08)	0.467	5.862	53.28	0.0008
PID (Kp=5.02, Ki=0.89, Kd=0.99)	1.253	7.028	5.13	0.0089
PIDA (Kp=25.03, Ki=3.04, Kd=9.98, Ka=3.01)	0.592	3.483	4.97	0.0074

(a)

Controller Comparison

Gp(s) Numerator (for Comparison): 1

Gp(s) Denominator (for Comparison): 0.5 3 4.5 1

H(s) Numerator (for Comparison): 1

H(s) Denominator (for Comparison): 1

Controller 1	Controller 2	Controller 3
PID	PID	PIDA
Kp: 15,6	Kp: 5,02	Kp: 25,03
Ki: 14,9	Ki: 0,89	Ki: 3,04
Kd: 4,08	Kd: 0,99	Kd: 9,98
Ka: 0	Ka: 0	Ka: 3,01

Comparison Results

Performance Metrics

Controller	Gain Margin (dB)	Phase Margin (°)	Gain Crossover (rad/s)	Phase Crossover (rad/s)
PID (Kp=15.60, Ki=14.90, Kd=4.08)	∞	26.49	2.34	nan
PID (Kp=5.02, Ki=0.89, Kd=0.99)	∞	61.65	1.10	nan
PIDA (Kp=25.03, Ki=3.04, Kd=9.98, Ka=3.01)	∞	99.92	3.48	nan

(b)

Figure 6. Screenshots of the second application.

As the finally application, a comparative analysis of the induction motor control system was performed [19]. The transfer function of system was given in Equation 4.

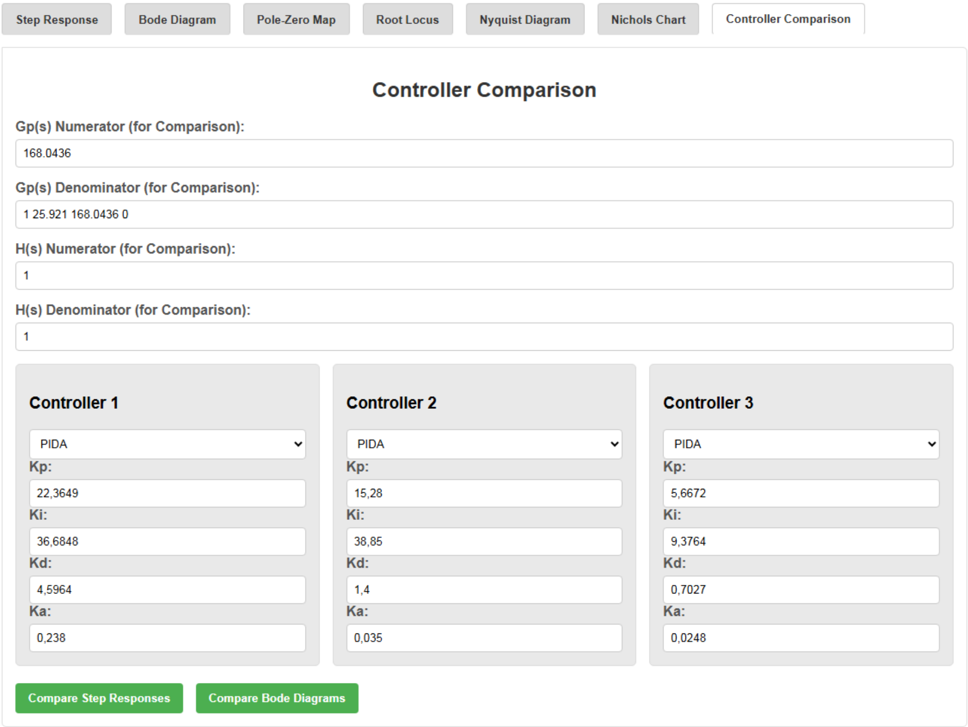
$$G(s) = \frac{168.0436}{s^3+25.921s^2+168.0436s} \tag{4}$$

The transfer functions of the compared controllers were given in Equation 5 [1, 19, 20].

$$\left.\begin{aligned} C_{Jung\&Dorf}(s) &= 22.3649 + \frac{36.6848}{s} + 4.5964s + 0.238s^2 \\ C_{CDM}(s) &= 15.28 + \frac{38.85}{s} + 1.4s + 0.035s^2 \\ C_{Kitty}(s) &= 5.6672 + \frac{9.3764}{s} + 0.7027s + 0.0248s^2 \end{aligned}\right\} \tag{5}$$

The result screenshots are given in Figure 7. Figure 7a shows the definition of the system and the entry of the coefficients of 3 different PIDA controllers obtained by the Jung & Dorf, CDM and Kitty methods. After clicking “Compare Step Responses” button, the page given in Figure 7b is opened. Among the compared PIDA controllers, it is seen that the controller designed with the Jung & Dorf method has better time domain performance parameters.

In the comparison of the Bode diagrams in Figure 7c, it is clearly seen that the phase margin of the PIDA controller designed with the Jung & Dorf method is greater than the others.



Controller Comparison

Gp(s) Numerator (for Comparison): 168.0436

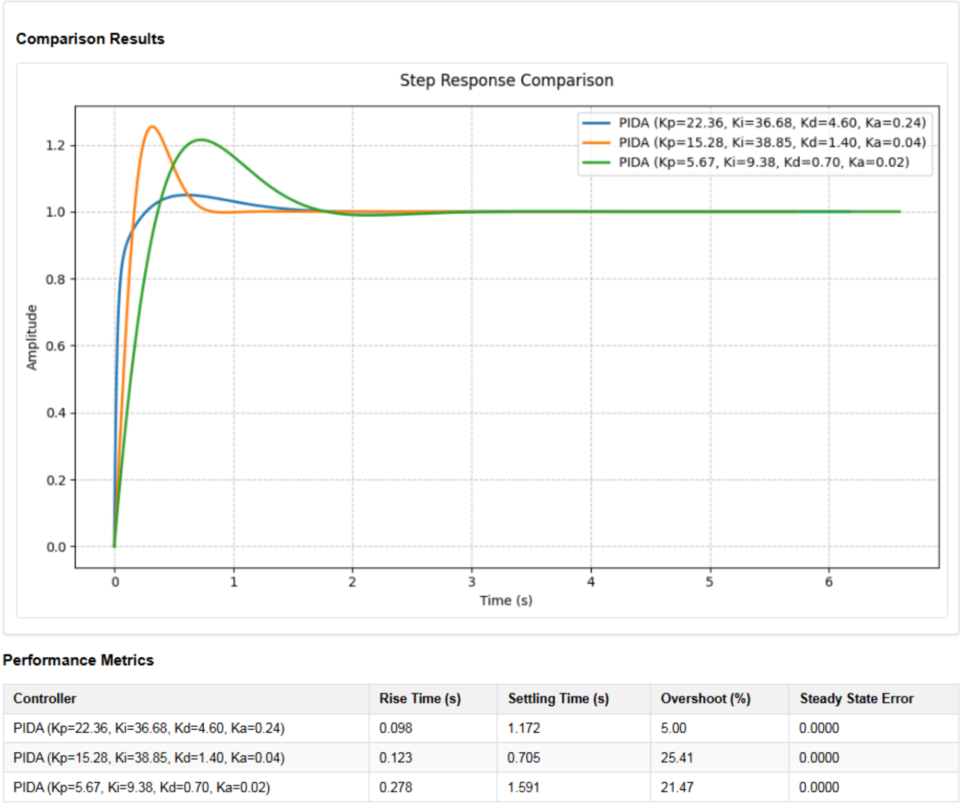
Gp(s) Denominator (for Comparison): 1 25.921 168.0436 0

H(s) Numerator (for Comparison): 1

H(s) Denominator (for Comparison): 1

Controller 1	Controller 2	Controller 3
PIDA	PIDA	PIDA
Kp: 22,3649	Kp: 15,28	Kp: 5,6672
Ki: 36,6848	Ki: 38,85	Ki: 9,3764
Kd: 4,5964	Kd: 1,4	Kd: 0,7027
Ka: 0,238	Ka: 0,035	Ka: 0,0248

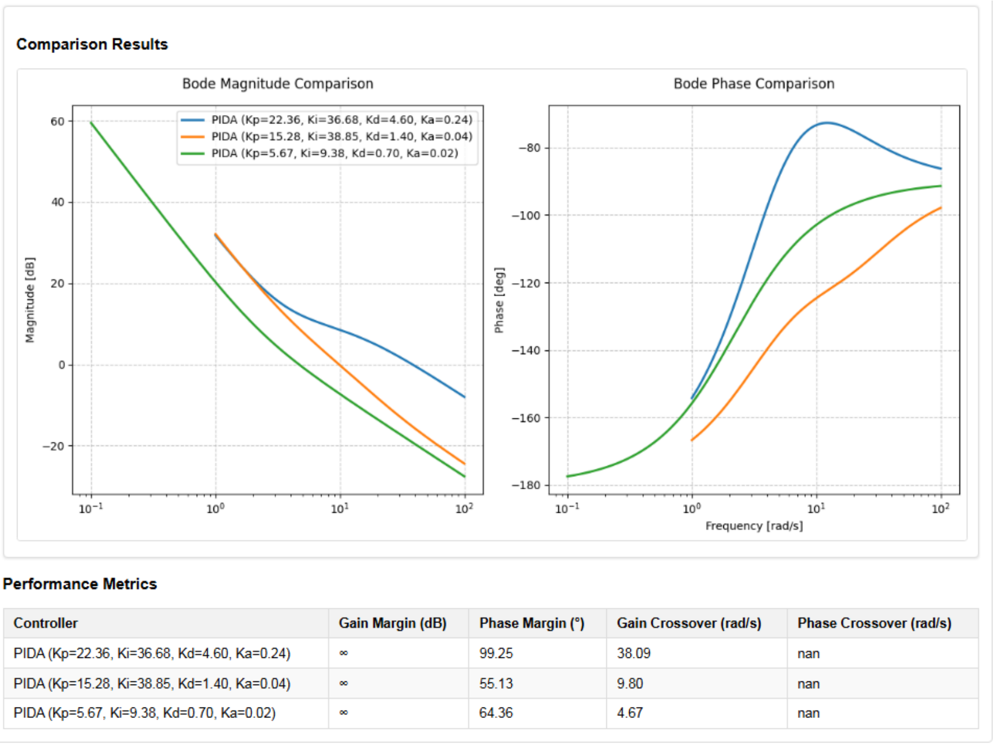
(a)



Performance Metrics

Controller	Rise Time (s)	Settling Time (s)	Overshoot (%)	Steady State Error
PIDA (Kp=22.36, Ki=36.68, Kd=4.60, Ka=0.24)	0.098	1.172	5.00	0.0000
PIDA (Kp=15.28, Ki=38.85, Kd=1.40, Ka=0.04)	0.123	0.705	25.41	0.0000
PIDA (Kp=5.67, Ki=9.38, Kd=0.70, Ka=0.02)	0.278	1.591	21.47	0.0000

(b)



Comparison Results

Performance Metrics

Controller	Gain Margin (dB)	Phase Margin (°)	Gain Crossover (rad/s)	Phase Crossover (rad/s)
PIDA (Kp=22.36, Ki=36.68, Kd=4.60, Ka=0.24)	∞	99.25	38.09	nan
PIDA (Kp=15.28, Ki=38.85, Kd=1.40, Ka=0.04)	∞	55.13	9.80	nan
PIDA (Kp=5.67, Ki=9.38, Kd=0.70, Ka=0.02)	∞	64.36	4.67	nan

(c)

Figure 7. Screenshots of the third application.

As demonstrated simulations, users can easily and quickly test system performance by changing controllers and their parameters, thus determining the most suitable controller for their systems. This also allows students to visualize the effects of controller and parameter changes on system responses, both numerically and graphically.

The basic comparison of some properties of the designed online simulator in this study with other simulators in the literature is summarized in Table 3.

Table 3. The comparison of some properties of the designed simulators.

Properties/Features			Ref. [5]	Ref. [6]	Ref. [7]	Ref. [8]	Ref. [9]	Ref. [10]	This study
Definitions	*Controlled system*		√	√	√	√	√	√	√
	Feedback system							√	√
Controller	*PID (P, PI, PD, PID)*		√	√	√	√	√	√	√
	PIDA		√	√	√	√	√	√	√
	Other type (I-PD, I-PDA, etc.)							√	
Disturbance			√	√	√			√	
Analysis type	*Single*		√	√	√	√	√	√	√
	Comparative		√	√	√	√	√	√	√
Responses	*Time domain*	*Step*	√	√	√	√	√	√	√
		Characteristic parameters				√	√	√	√
	Frequency domain	*Bode*				√		√	√
		Characteristic parameters				√		√	√
Plots	*Root locus*							√	√
	Pole-zeros map							√	√
	Nyquist diagram							√	√
	Nichols chart							√	√
Online operation (web-based)			√	√	√		√		√
Platform and/or programming language used for design			MATLAB GUI	MATLAB GUI	MATLAB HTML, CSS, JS	MATLAB App Designer	Python, JS	MATLAB App Designer	Python, HTML, CSS, JS
Basic operations (save, print etc.)					√			√	

4. Conclusion

Control systems have a wide range of applications. In parallel with technological advancements, new control algorithms and structures are constantly being developed and used in applications. Therefore, understanding and using controllers is extremely important. In this study, a web-based and user-friendly simulator with a flexible interface was developed for PID and PIDA controllers. With this online simulator, single and comparative analyses can be performed using controllers specified for user-defined systems in any environment with internet access, without time and space constraints. It facilitates both the learning and understanding of the related controllers and their use in applications with its rich numerical (performance metric values, custom values, etc.) and graphical (step response, Bode and Nyquist diagrams, root locus plot, pole-zero map) results.

References

[1] Jung S, Dorf R. Analytic PIDA controller design technique for a third order system, Proceedings of 35th IEEE Conference on Decision and Control; 1996; Kobe, Japan. pp. 2513-2518.

[2] Jung S, Dorf R. Novel analytic technique for PID and PIDA controller design. IFAC Proceedings Volumes 1996; 29(1): 1146-1151.

[3] Hatun M, Vatansever F. The design of control system simulator. Bursa Uludağ University Journal of the Faculty of Engineering 2022; 27(1): 41-56.

[4] The MathWorks Inc. 2023. MATLAB

[5] Bisták, P. Disturbance analysis virtual laboratory for PID controllers with higher derivative degrees. In: 2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA); 2018; Stary Smokovec, Slovakia. pp 69-74.

[6] Bisták P, Huba M. Analysis of higher derivative degree PID controllers via virtual laboratory. In: 2019 27th Mediterranean Conference on Control and Automation (MED); 2019; Akko, Israel. pp. 356-361.

[7] Bisták P, Huba M. Interactive software tool for design of higher derivative degree PID controllers. IFAC-PapersOnLine 2020; 53(20): 17198-17203.

[8] Ferrari M, Visioli A. A software tool to understand the design of PIDA controllers. IFAC-PapersOnLine 2022; 55(17): 249-254.

[9] Žáková K, Matišák J, Šefčík J. Contribution to PID and PIDA interactive educational tools. IFAC-PapersOnLine 2024; 58(7): 115-119.

[10] Hatun M, Vatansever F. The PIDA controller analysis simulator. Gazi University Journal of Science Part C: Design and Technology 2025; 13(2): 526-537.

[11] Python. https://www.python.org/

[12] Fuller S, Greiner B, Moore J, Murray R, van Paassen R, Yorke R. The Python Control Systems Library (python-control). In: 2021 60th IEEE Conference on Decision and Control (CDC); 2021; Austin, TX, USA. pp. 4875-4881. (Python Control Systems Library. http://python-control.org/)

[13] Harris CR, Millman KJ, van der Walt SJ et al. Array programming with NumPy. Nature 2020; 585(7825): 357-362. (NumPy. https://numpy.org/)

[14] Hunter J. Matplotlib: A 2D graphics environment. Computing in Science & Engineering 2007; 9(3): 90-95. (Matplotlib. https://matplotlib.org/)

[15] Flask. https://flask.palletsprojects.com/

[16] Heroku. https://www.heroku.com/

[17] Sinlapakun V, Assawinchaichote W. Optimized PID controller design for electric furnace temperature systems with Nelder Mead Algorithm. In: 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON); 2015; Hua Hin, Thailand. pp. 1-4.

[18] Jitwang T, Puangdownreong D. Application of cuckoo search to robust PIDA controller design for liquid-level system. International Journal of Innovative Computing, Information and Control 2020; 16(1): 189-205.

[19] Sambariya DK, Paliwal D. Comparative design and analysis of PIDA controller using Kitti's and Jung-Dorf approach for third order practical systems. Journal of Advances in Mathematics and Computer Science 2016; 16(5): 1-16.

[20] Patu P, Jongkol N., Kitti T, Noriyuki K, Shunji M. PIDA controller design by CDM. In: Proceedings of the 13th Korea Automatic Control Conference (KACC); 1998; Busan, Korea. pp. 395-400.