



PID CONTROLLERS DESIGN WITHIN THE CONTEXT OF ONLINE EDUCATIONAL ACTIVITIES

Mircea DULĂU¹, Delia Bianca SASU²

¹University of Medicine, Pharmacy, Sciences and Technology “George Emil Palade” of Targu Mures
38 Gh. Marinescu st., 540139, Targu Mures, Romania

¹mircea.dulau@umfst.ro

²University of Medicine, Pharmacy, Sciences and Technology “George Emil Palade” of Targu Mures
38 Gh. Marinescu st., 540139, Targu Mures, Romania, SACPI master student

¹delia_sasu@yahoo.com

Abstract

The problem of designing conventional controllers is a major concern in both academia and industry. The paper presents the implementation of a Matlab interface that facilitates the students' understanding of the closed-loop control systems operation. There are four operating scenarios: with controllers for which the parameters are known; with controllers for which parameters are determined experimentally; with proportional controllers and additional poles/zeros introduction; with proportional controllers and serial compensator.

Keywords: PID controller, PID tuning, additional poles-zeros, compensator

1. Introduction

For a system specified by the transfer function of the process (including the transducer and the actuator), $H_P(s)$, the design problem refers to the determination of a controller, $H_R(s)$, with which the closed-loop diagram (Fig. 1) satisfies the required performances:

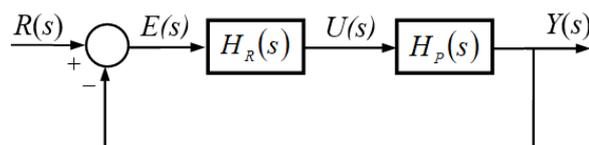


Fig. 1: General diagram of the closed-loop system

- overshoot condition:

$$\sigma \leq \sigma_{imp}, \quad (1)$$

- settling time condition:

$$t_t \leq t_{timp}, \quad (2)$$

- rise time condition:

$$t_c \leq t_{cimp}, \quad (3)$$

- steady state error condition:

$$\varepsilon \leq \varepsilon_{imp} \quad (4)$$

In order to fulfill these conditions, the dominant poles, dependent on the natural frequency, ω_n and on the damping, ξ :

$$s_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2} \quad (5)$$

must be appropriately located in the left side of the complex plane $\{\text{Re } s < 0\}$ (Fig. 2).

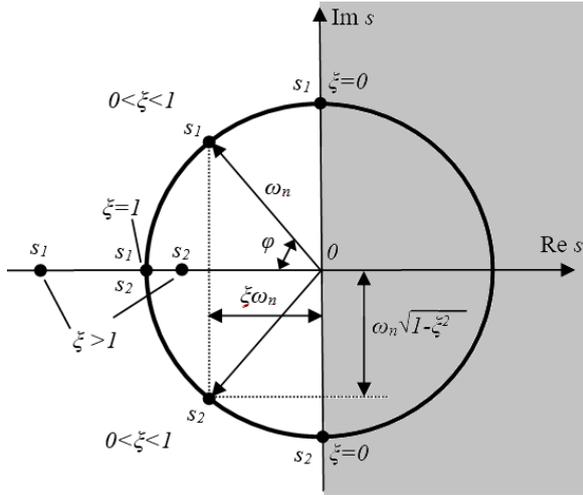


Fig. 2: Possible positions of the poles in complex plane

The major difficulty in the design procedure comes from the fact that the performance indicators are specified in the time domain (t), with the design taking place in the complex plane (s), based on the transfer functions.

The procedure can be simplified and applied to a certain class of transfer functions by translating the time domain specifications into the complex domain.

From this translation it results that the dominant poles (s) must be located in the left side of the complex plane, in the admissible domain D_{adm} defined by the inequalities (1), (2), (3), as close as possible to the border D_{adm_border} (Fig. 3).

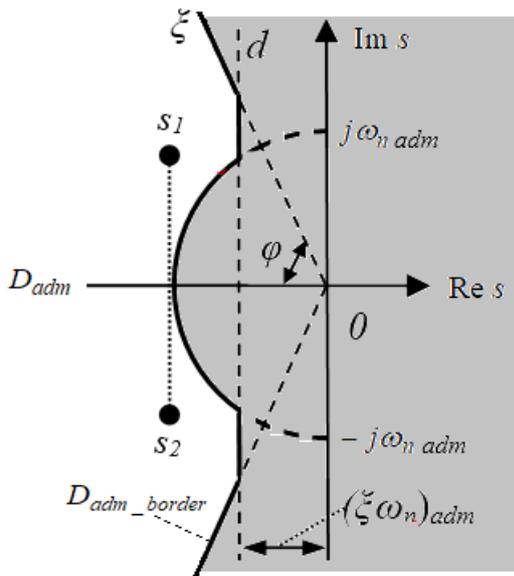


Fig. 3: Position of the D_{adm} domain in complex plane

Therefore, the problem of designing conventional controllers:

- is taught to all students in automation study programs;
- is the subject of research studies in the field, in collaboration with industry experts.

Thus, in [1] are developed: linear time invariant systems, quantitative and qualitative aspects of automated systems, design methods in the continuous and discrete field, including the roots locus method.

In [2] there are presented the fundamentals of PID controllers, as well as the tuning based on the analytical (Kessler) and experimental (Ziegler-Nichols, Kapelovici etc.) criteria. The controllers for complex structures (e.g. multivariable, cascade, feedforward etc.), including those based on integral criteria are developed for students' use in [3].

A series of simple and advanced techniques (analytical and experimental), used for choosing and tuning PID controllers are well detailed in [4].

In [5] there are presented the mathematical tools for the automated systems analysis, the design principles for single-loop control systems and for cascade and multivariable systems, as well as the techniques for frequency analysis.

Aspects related to process dynamics, advantages of feedback, design techniques for PID controllers and advanced control systems are the topics in [6].

The compensation methods for the linear systems, focused on the roots locus and on the position of poles and zeros in the complex plane (s), are identified in [7] and use the facilities offered by the Matlab software.

The utility of a PI-lead compensator in roots locus design is developed in [8], for a double integrator and a second-order system.

Paper [9] proposes an extended roots locus method for a two parameters controller, which can be extended to designing multi-parameter controllers.

According to the research [10], using a combination between the poles placement and the symmetrical optimum criterion, the values of the tuning parameters depend on the process / plant and the model of the second order system, considered as a reference.

Nowadays, the universities are directly affected by the COVID-19 pandemic, the educational activity being moved to the online environment. Thus, the teaching activities and the relationships with students need to be adapted to the new context.

A study related to continuous improvement of the student relationship management by active usage of ICT solutions in the epidemiological context (COVID-19), is developed in [11].

Matlab/Simulink software offers tools for finding the best PID controllers in a closed-loop control system [12]. Other tools contain procedures for PID tuning and design, such as: process identification, controller tuning based on specifications (gain and phase margin, disturbance rejection etc.) [13].

However, the designing of the controllers becomes a difficult task, especially in online educational activities.

The paper offers a possible solution for teaching and understanding the designing of the controllers based on roots locus (a method considered more complex than others) in the context of the online

activities.

The simulation of the open-loop and closed-loop systems behavior is facilitated by the Matlab software.

After presenting the problem of designing controllers and the significant references in the field (Section 1), Section 2 of the paper specifies the necessary steps used to design the controller based on roots locus (a method considered more complex than others).

In Section 3, the flow chart and the Matlab interface are explained.

The conclusions are provided in the final section.

2. Designing the controllers based on roots locus method

The design procedure, applicable to the general diagram of the closed-loop control system (Fig. 1) involves the following steps:

1. a control configuration with a single parameter is chosen: $H_R(s) = k_r$ (a proportional controller on forward path);
2. the closed-loop transfer function and the range of variation for the parameter k_r are determined, so that the system is stable and satisfies the condition imposed for the steady state error; if there are no such values, the design from step 1 is reconsidered with a different controller configuration;
3. the roots locus of the closed-loop system is plotted, depending on the parameter k_r ;
4. the range of values for k_r is found, based on which the roots locus is situated within the admissible domain in the left side of the complex plane; if there are no such values, the design from step 1 is reconsidered with a different controller configuration;
5. the range of values for k_r is determined, which ensures that the conditions in step 2 and step 4 are met simultaneously; if there are no such values, the design from step 1 is reconsidered with a different

controller configuration.

Alternatives to the proportional controller refer to the use of a PI, PD, PID controller or to the introduction of a compensator, $H_C(s)$, connected in series with the proportional controller, $H_R(s) = k_0$ (Fig. 4).

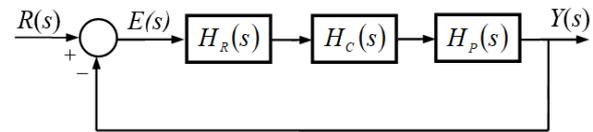


Fig. 4: Closed-loop system with compensator

The transfer function of the compensator is:

$$H_C(s) = \frac{s + g}{s + b \cdot g} \quad (6)$$

where: if $b > 1$, the network is a lead one and if $b < 1$, the network is a lag one.

3. Matlab software application

The flow chart of the application highlights four scenarios (Fig. 5):

1. the analysis of the system's behaviour with a known conventional controller (e.g. a PID controller, tuned by experimental or analytical criteria);
2. the specification of the imposed performances in stationary and transient regime, the determination of the tuning parameters of the controller, followed by the performance analysis;
3. the introduction of an additional pole / zero and the performance analysis;
4. the determination of a compensator having a lead / lag type and the performance analysis.

The Matlab interfaces, corresponding to the 3rd and 4th scenarios are presented in Fig. 6 and 7.

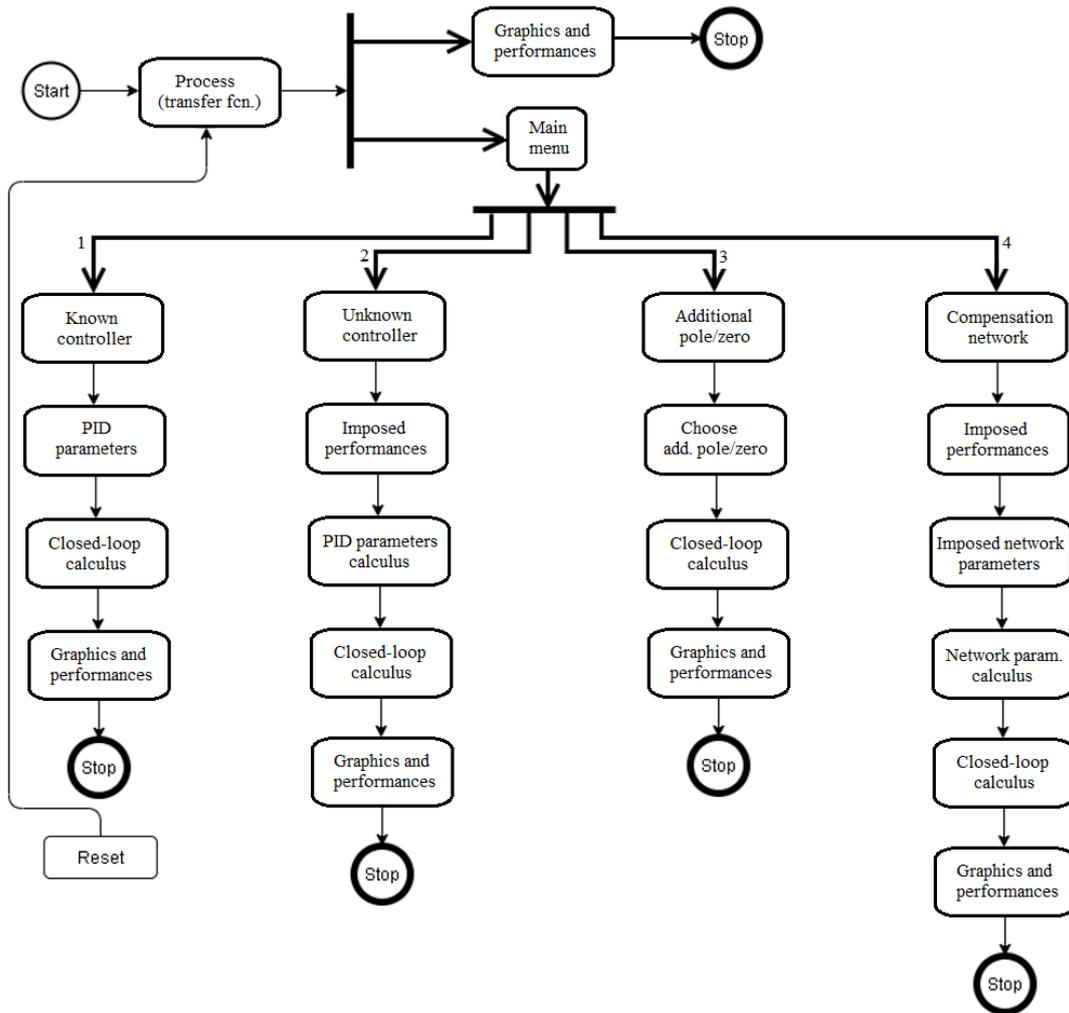


Fig. 5: Flow chart

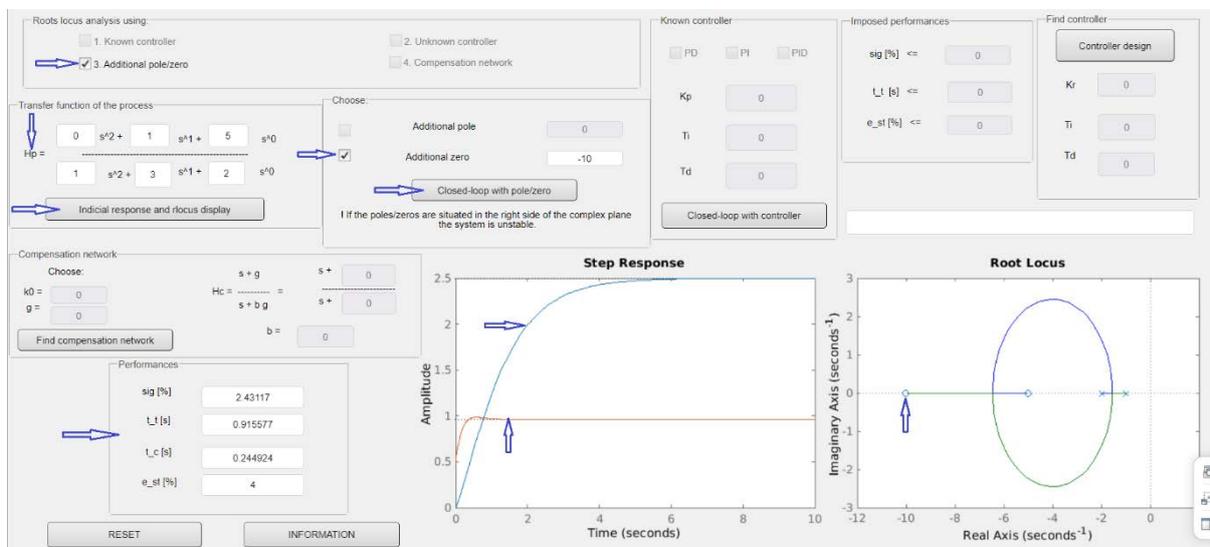


Fig. 6: Matlab interface for the 3rd scenario

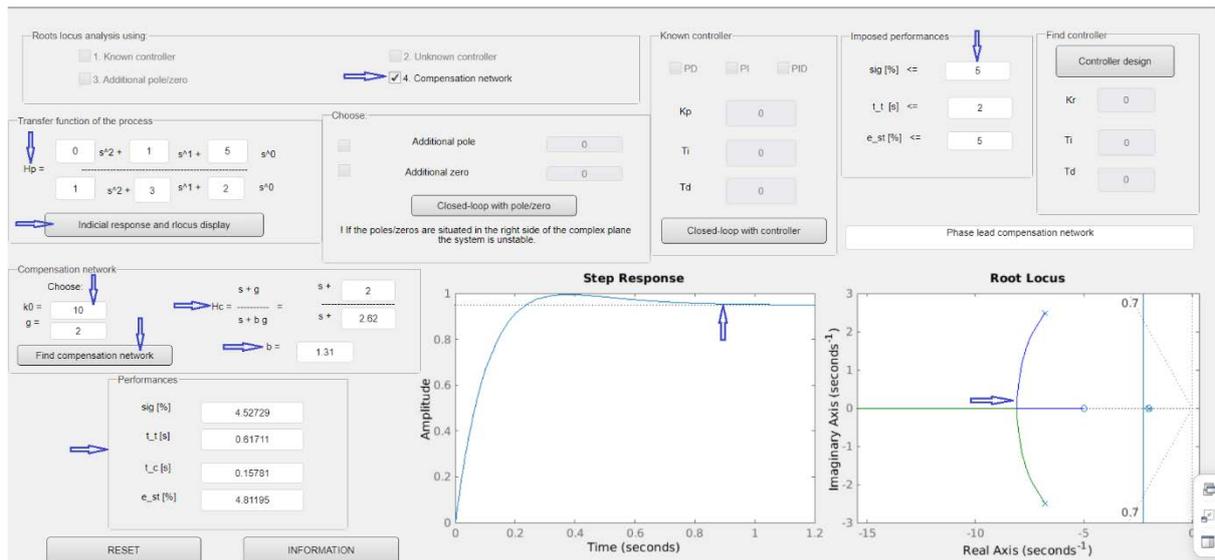


Fig. 7: Matlab interface for the 4th scenario

The Matlab interface allows an easy interaction with the student and requires specifying the transfer function of the process (max. 2nd ord. in this implementation stage), then choosing one of the mentioned scenarios.

The operations are suggested by the sequence of windows and buttons on the graphical interface.

Initially, it is possible to represent the step response of the process (without the controller), the roots locus and the performances.

According to the 3rd scenario:

- the user specifies the transfer function of the process, $H_p(s)$;
- the step response (blue line) and the roots locus are plotted;
- an additional zero is introduced (in example: $z = -10$);
- the step response (red line), the roots locus and the obtained performances are analyzed.

According to the 4th scenario:

- the user specifies the transfer function of the process, $H_p(s)$;
- the step response and the roots locus are plotted;
- the imposed performances are specified (in example: $\sigma \leq 5\%$; $t_l \leq 2 \text{ sec.}$; $e_{st} \leq 5\%$);
- the values for k_0 and g are chosen (in example: $k_0 = 10$; $g = 2$);
- the type and the transfer function of the compensator are determined (in example: $b = 1,31$; lead network);
- the step response, the roots locus and the obtained performances are analyzed.

4. Conclusions

The Matlab interface is implemented in order to facilitate the determination of conventional PID controllers for processes described by transfer functions of maximum 2nd order, without the need of

writing Matlab code.

From the online teaching/learning point of view, each student has access to the Matlab software and interface and can test it for any of the four scenarios.

Also, using an educational platform (e.g. Blackboard, Teams, Zoom, Classroom etc.), students can share the screen and discuss the performed operations with the teacher, in order to understand the mechanisms behind the four scenarios.

A successful use of the Matlab interface requires the existence of basic knowledge regarding:

- the transfer function algebra;
- the effect of feedback in a control diagram;
- the methods for choosing and tuning the PID controllers;
- the effect of the tuning parameters (k_p , T_i , T_d) on the control performances;
- the significance of the roots locus in the complex plane;
- the lead/lag compensator.

A future version of the interface aims to:

- extend the controller's design to processes with a transfer function higher than the 2nd order;
- analyze the stability according to Nyquist and Bode characteristics;
- design the numerical controllers with options for choosing the method type (e.g. from continuous PID, based on imposed number of sampling time etc.).

References

- [1] Chen, C.T. (2000), Analog and digital control system design. Transfer-function, Space-state and algebraic methods, Saunders College Publishing.
- [2] Dulău, M. (2016), Ingineria sistemelor automate. Teorie și aplicații (Control Systems Engineering. Theory and applications), "Petru Maior" University of Târgu Mureș, România.

- [3] Dulău, M. and Gligor, A. (2015), *Introducere în Ingineria sistemelor automate (Introduction to Control Systems Engineering)*, "Petru Maior" University Press, Târgu Mureș, România.
- [4] Lazăr, C., et al (2004), *Sisteme automate cu reguloare PID (Automated Systems with PID Controllers)*, MatrixRom, România.
- [5] Smith, C., Corripio, A. (1997), *Principles and Practice of Automatic Process Control*, John Wiley & Sons, Inc.
- [6] Seborg, D. et al (2011), *Process Dynamics and Control*, John Wiley & Sons, Inc.
- [7] Lu, R. (2019), *Series Lead Correction by Root Locus Method*, 2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS), Chongqing, China, pp. 604-606, doi: 10.1109/ICICAS48597.2019.00132.
- [8] Zhang, Q. and Messner, W.C. (2011), *Root locus design with Complex Proportional-Integral-Lead compensation*, Proceedings of the 2011 American Control Conference, San Francisco, CA, pp. 693-698, doi: 10.1109/ACC.2011.5991073.
- [9] Kwon, M. and Chang, H. (2016), *Extended root-locus technique applied to pole-placement for PI controller design*, 2016 UKACC 11th International Conference on Control (CONTROL), Belfast, pp. 1-6, doi: 10.1109/CONTROL.2016.7737535.
- [10] Nicolau, V. (2013), *On PID Controller Design by Combining Pole Placement Technique with Symmetrical Optimum Criterion*, *Mathematical Problems in Engineering*, vol. 2013, Article ID 316827, 8 pages, <https://doi.org/10.1155/2013/316827>.
- [11] Timus, M., et al (2020), *Student Relationship Management Optimization Using Organizational Process Automation Tools*, *Acta Marisiensis, Seria Oeconomica*, Volume 14: Issue 1, Targu Mureș, doi: 10.2478/amso-2020-0004.
- [12] Mathworks, *Matlab Control System Toolbox*, [Online]. Available: <http://matlab.mathworks.com/>.
- [13] PID Control Laboratory, [Online]. Available: <https://www.pidlab.com/>.