

# ProPID: An Interactive CAD Tool for Control Education<sup>1\*</sup>

MIGUEL MARTÍNEZ AND JOSÉ V. SALCEDO

*Predictive Control and Heuristic Optimization Group, Dept. of System Engineering and Control, Universidad Politécnica de Valencia, Camino de Vera 14, 46022—Valencia, Spain*

*E-mail: mmiranzo@isa.upv.es*

*The design of PID controllers using the root locus method, based on classic module and argument criteria, is currently taught in most basic academic control engineering courses, and is found in most texts on the subject. Traditionally, PID controllers are designed by applying geometrical rules to calculate PID parameters. Although this task can be computerized, it is not easy to achieve because of the special characteristics of the particular equation criteria, which are presented as a trigonometric equation that is generally solved using iterative methods. An alternative method with an equivalent result that is computationally efficient and does not require iterative calculations is implemented in Sysquake © and a new interactive CAD tool is provided: ProPID. Using this tool, students can easily become familiar with PID behaviour in a closed loop and compare the specifications obtained with those desired. This software tool also enables the student to redesign the PID to verify, as far as possible, the desired specifications. All of the design work is implemented within an interactive environment and any modification of the design parameters produces an immediate dynamic response. This key aspect enables students to learn interactively.*

**Keywords:** interactive learning; control design; PID; root locus; CAD tools; simulation languages

## INTRODUCTION

### *Root locus method*

THE ROOT LOCUS METHOD designed by Evans in 1948 is a graphic method based on a series of geometrical rules that enable us to determine the evolution of the closed loop poles of a linear process with variable proportional gains (see Fig. 1).

As is well known, the characteristic equation of the closed loop system is

$$1 + KG(s)H(s) = 0 \quad (1)$$

In this way, when gain  $K$  varies, the position of the roots of that equation in complex plane  $s$  is modified. Nevertheless, only using gain  $K$  can prove to be insufficient to make the closed loop system comply with the desired design specifications. For this reason, gain  $K$  is replaced by a controller with a more complex structure<sup>2</sup>

$$G_{PID}(s) = K \frac{(s+a)}{s} (s+b) \quad (2)$$

and it is necessary to determine gain  $K$ , as well as the position of the zeros  $s = -a$  and  $s = -b$  that make the specifications agree.

These specifications can be static or dynamic. The former can be satisfied by incorporating a pole in the origin. The latter come from the dynamic requirements that the process should ideally fulfil, for example, time constant  $\tau$ , peak time  $t_p$ , overshoot  $\delta$ , settling time  $t_s$ , etc., which can be transformed for transfer to the complex plane. These can then be turned into specifications in the complex domain such as, damping factor  $\sigma$ , damping coefficient  $\xi$ , natural frequency  $\omega_n$ , etc. These specifications do not necessarily involve a restriction of equality, but rather a restriction of inequality. When this is the case, specifications on the complex plane do not determine a specific point. Instead, the feasible specification point may be a line or a surface. From this perspective, the closed loop system will only satisfy the specifications if the root locus of the characteristic equation, with the degree of freedom incorporated by the PID controller, intersects the geometry derived from the specifications.

Since the design proposed is exclusively based on the characteristic equation, which only corresponds to the denominator of the closed loop transfer function, it is evident that the final specifications will be affected by the presence of the additional zeros of this transfer function. Since the PID's degree of freedom is restricted<sup>3</sup>, it will only be possible, at most, to fix a pair of poles from the characteristic equation, while the other poles may

\* Accepted 18 June 2008.

<sup>1</sup> This research has been partially financed by DPI2005-07835, MEC (Spain)-FEDER.

<sup>2</sup> The non consideration of any of the terms of the expression (2) leads to the established P, PI and PD controllers as particular cases.

<sup>3</sup> There is only freedom to assign  $a$  and  $b$ .

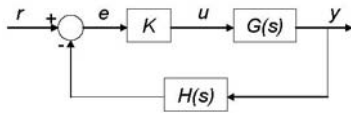


Fig. 1. Closed loop with variable gain  $K$ , process  $G(s)$ , sensor  $H(s)$ , reference  $r$ , error  $e$ , control action  $u$  and output  $y$ .

also affect the final specifications. All of these factors should be considered when establishing the real effect that the PID design has on system specifications—so the necessary amendments can be made if the result does not coincide with the designer's initial specifications.

The methodology described can be found in many Process Control and Control Engineering textbooks, some of them classics, though frequently reedited [2, 6], and some modern [4]. Nevertheless, in each studied case, this design methodology always reverts to the established geometric rules on root locus—and to the classic module and argument criteria. Although the methodology may seem easy to apply, it presents serious problems when run on a computer, since some of the plot rules and criteria require iterative trigonometric equations, where the intervention of the designer should be carefully planned to avoid results being incompatible with the original problem.

#### Alternative method

To resolve this problem it is possible to use an alternative that obviates the module and argument criteria and replaces it with pole assignment in the characteristic equation—starting with the desired specifications and solving the problem using complex number algebra that solves the equations of the real and imaginary parts of the characteristic equation.

This will have a direct result on the three parameters of the PID controller<sup>4</sup>. It is very important to notice that the result coincides with the result obtained using the classic module and argument methods.

This alternative design method begins with a selection of specifications from the complex plane<sup>5</sup> that correspond to a couple of desired dominant poles of an underdamped system. These two poles are defined as<sup>6</sup>:

$$p = -\sigma \pm \omega_p j \quad (3)$$

Once these two points are established, the controller is calculated in an *analytical way*, without using iterative calculations, specifically working out the transfer function of the controller from the characteristic equation, since the aim is to have the root locus of the characteristic equation going through

the dominant conjugated complex poles  $p$ . If  $s$  is replaced by  $p$  the result is

$$G_{PID}(p) = -\frac{1}{G(p)H(p)} \quad (4)$$

$$\operatorname{Re} G_{PID}(p) = \operatorname{Re} \left[ -\frac{1}{G(p)H(p)} \right] \quad (5)$$

$$\operatorname{Im} G_{PID}(p) = \operatorname{Im} \left[ -\frac{1}{G(p)H(p)} \right] \quad (6)$$

These two expressions set up a system of equations with a constant value to the left and a dependent value of the PID parameters ( $K$ ,  $a$  and  $b$ ) to the right. When presenting the equation system in this way, obviously, it is incompatible, but this has a meaning that is associated with the other family controllers (P, PI, PD) so that the result is sometimes associated with one of them—so reducing the number of unknowns in such a way that it becomes compatible.

As a next step, depending on the result of equations (5) and (6), an analysis will be made of which controller is more suitable and how to make the calculation: P, PI, PD or PID.

#### Educational objectives

The simplicity of the analysis and the automation of the calculations proposed by this equivalent strategy enable the root locus vision to be reconsidered from an educational point of view. This has immediate benefits, such as a reduction of student study time and the possibility of using computational tools to obtain calculations in a direct way, without requiring the intervention of the designer.

The interactive design CAD tool Sysquake, developed at the Laboratoire d'Automatique, part of the School of Engineering (STI) of the Swiss Federal Institute of Technology (EPFL) in Lausanne, Switzerland [7], is used to because of its interactive graphic capacity. The most important characteristic of Sysquake is that, during its manipulation, all the graphs are updated to reflect the change. Neither the information shown by the graphics, nor the updating of the graphs is fixed; both depend on programs written in an easy-to-learn language specifically designed for numerical computation [5]. Many other control education tools have incorporated the concepts of dynamic pictures and virtual interactive systems and can be used for this development [1]. These tools enable interactivity in the learning process.

ProPID is a CAD tool designed with Sysquake that implements an alternative method to PID design. This tool easily enables the student to become familiar with the behaviour of the designed PID on the closed loop and to compare the real specifications obtained with those desired. Finally, by using the software tool facilities, the student can also adjust the design of the PID to verify the desired specifications. Thus, the problems mentioned earlier regarding the presence of poles

<sup>4</sup> Or two, if it is a PI or a PD or one, if it is a P.

<sup>5</sup> Equality specifications correspond to points; when dealing with inequality specifications, specific points of the surface will have to be chosen.

<sup>6</sup> Represents either of the two conjugated complex poles.

and zeros additional to those assigned by the PID parameters in the characteristic equation, which affect global behaviour, can first be detected, and then corrected or improved—through a soft adjustment using the degrees of freedom the Sysquake graphic tool made available to the designer. The presented CAD tool is generic because the alternative method implemented with Sysquake can be applied to any continuous linear process  $G(s)$  (see Fig. 1) and with any continuous linear sensor  $H(s)$ . Other works [9, 12] also propose an automated design of PID controllers, and the main differences with these works are as follows.

- They use Matlab and LabVIEW, respectively, instead of Sysquake. Both programs have characteristics that are different from Sysquake, however only Sysquake was designed as a design CAD tool.
- In [9] the PIDs are designed using the ITAE criterion. This criterion is presented to students in postgraduate courses, once they understand the root locus method. For this reason, ProPID was designed to be used in undergraduate courses.
- In [12] a trainer program is designed using LabVIEW for simulating and designing PID controllers. The selection of controller parameters is made in an intuitive way after trial and error using simulations. This tool could be used by undergraduate students together with ProPID because they are complementary. However, LabVIEW is not free software, unlike ProPID. Consequently, the program developed in [12] can be used in the classroom or in the laboratory, while ProPID can be used on any computer.

In addition, ProPID exploits a key ingredient: the interactivity as a crucial educational tool in control learning [3, 8–12]. This element is included to improve the learning process in control engineering: immediate updates of graphic elements in the CAD tool provide clear feedback to the students about the changes that they are making.

This paper is organized as follows: the next section presents the ProPID tool, starting with the main features of Sysquake, describing the ProPID tool and then giving some teaching applications of the tools. The following section presents the utilities of the ProPID tool for control education. The last section presents the conclusions and discusses future work that will enable this methodology to be used in a more industrial environment.

## DESIGNING OF PID CONTROLLERS WITH SYSQUAKE: PROPID

### *Introduction to Sysquake*

Sysquake is an innovative commercial design software made by Calerga<sup>7</sup>. It is powerful and

flexible enough for understanding systems, solving problems, and designing products. Its unparalleled graphic interactivity makes it special. Sysquake can be used in many areas, including automatic control, robotics, statistics, physics, etc.

Simulation of dynamic systems can benefit greatly from Sysquake's interactivity. Parameters often have long-term effects that are difficult to deduce intuitively from the model, or from a few static simulation graphics. Manipulation with Sysquake will make apparent their role and the compromise that can be obtained. Its main features are as follows.

- Power: It includes a built-in language including more than 630 native functions, operators, and easily extensible commands. Some graphic functions are specific to dynamic systems, and others are more general for displaying any kind of data.
- Fast: What would be slow in a script written in a high-level interpreted mathematical language can be implemented very efficiently in native machine code.
- Ready to use: Sysquake comes with a rich set of applications for a broad range of areas, such as classic control, robust control, and digital filters.
- Easy to use: The graphic user interface is very simple and intuitive. Multiple-level 'undo' enables the user to experiment without fear of losing previously obtained performances.
- Easy to extend: Sysquake is based on LME, an interpreter specialized in numerical computation. Only the code necessary for displaying the figures and processing manipulations from the user is required. The management of graphics, zoom, saving, and loading data, undo, etc. is offered by Sysquake.
- Application builder: Sysquake Application Builder, provided with Sysquake, allows the creation of stand-alone applications that can be redistributed freely.
- Supported platforms:
  - PC Pentium or compatible with Windows 2000 or XP;
  - Macintosh PowerPC or Intel with Mac OS 10.3 or higher;
  - PC Pentium or compatible with Linux, X Window and GTK+ 2.

There is also a free version of Sysquake, called Sysquake LE, with some limitations that do not affect the objectives presented in the section above. The authors have been using this version and it can be downloaded from the Calerga website. Students can also use this free version ([www.calerga.com](http://www.calerga.com)).

*ProPID tool.* Since automatic control is one of the areas in which Sysquake is implemented, it seems logical to consider using this software for implementing the new methodology for designing PIDs.

ProPID<sup>8</sup> is a Sysquake application made with

<sup>7</sup> [www.calerga.com](http://www.calerga.com)

<sup>8</sup> Available at [http://personales.upv.es/~jsalcedo/sysquake/propid\\_en.sq](http://personales.upv.es/~jsalcedo/sysquake/propid_en.sq)

the free version of Sysquake (LE), which includes PID design using the method of real and imaginary parts presented in the first section. Figure 2 shows this application when started.

Four areas in the application need highlighting. The top left area presents the aspects related to the specifications, process, sensor and P, PD, PI, or PID (depending on each case) controllers:

- Type of system: this may be first or second order. If it is first order, then there is a choice between settling time ( $t_s$ ) or time constant. If it is second order, there is a choice between six pairs of options that combine specifications typical of a second order system ( $\xi$ ,  $\omega_p$ ,  $\omega_n$ ,  $\sigma$ ,  $t_s$ ,  $\delta$  and  $t_p$ ). By default a specification of the second order is selected using a settling time ( $t_s$ ) of 2 seconds and overshoot ( $\delta$ ) equal to 0.25.
- Static specification: position error, velocity error, or acceleration error. By default, the position error has a value of 0.15.
- All the specifications below can be modified by the user. By default, the values corresponding to the example presented in the ProPID tool utilities for control education section are selected.
- The integrator zero ratio can also be modified. Its default value is 6.
- The designed process, sensor, and regulator are discussed below. The process and the sensor can be modified by checking the options available in the menu settings. The introduction can be made through numerator and denominator, or through gains, zeros, and poles. By default, the process and sensor are:

$$G(s) = \frac{2}{(s + 3)(s + 1)}, \quad H(s) = 1 \quad (7)$$

The application updates the calculation of the controller whenever any of the parameters located to the upper left are modified.

The upper right area presents a figure with the resulting root locus. The position of the closed

loop poles (indicated with triangles) can be modified by selecting and dragging. This operation only modifies the gain of the controller, as observed in the expression of the controller located in the upper left area. Both the position of the zero of integrator (blue circle) and of the zero of derivator (red circle) can also be modified. These changes will enable the user to modify the controller obtained, and therefore the closed loop obtained specifications, with the aim of approaching the desired specifications—if these have not been achieved with the initial controller design. Also, the open loop poles are represented by a black x, and open loop zeros by black circles.

To check whether the desired specifications have been verified or not, the unit step responses of both closed loops (achieved specifications: discontinuous red line with red asterisks, and desired specifications: continuous blue line) have been graphically represented in the bottom left area.

Finally, the closed loop control action applied by the controller is designed when the reference applied is a unit step, and is represented in the bottom right. By using the graph, it is possible to analyse if the designed controller implements control actions that are inadmissible in the real process.

### Teaching examples

*Exact design.* This is the first type of example shown to students in practical classes. The design achieved is exact when the root locus method, or the alternative method, is applied. This occurs when the closed loop has only two poles and no zeros—and the poles are positioned using any of these methods. For example, for the process and sensor (using the ProPID tool by default):

$$G(s) = \frac{2}{(s + 3)(s + 1)}, \quad H(s) = 1 \quad (8)$$

if dynamic specifications of equality are estab-

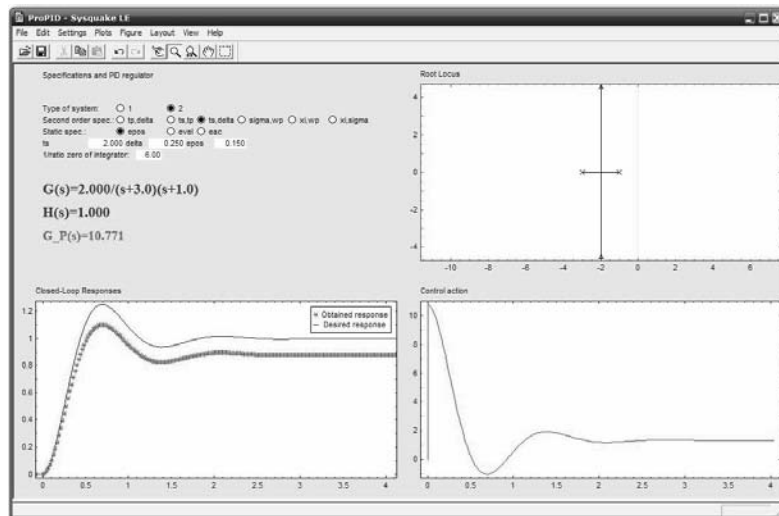


Figure 2. ProPID application developed in Sysquake LE

lished corresponding to a second order system:  $t_s = 2$  s,  $\delta = 0.25$  and a static specification of inequality is established as:  $e_p = \leq 0.15$ , then the design will be exact for the dynamic specifications—as it is only necessary to use a type P regulator that does not introduce additional poles or zeros. This can be seen in Fig. 2, which shows that the required dynamic specifications (blue curve) are the same as those obtained (red curve). Nevertheless, it is worth emphasising that both curves do not coincide as a static specification of inequality ( $e_p = \leq 0.15$ ) and this justifies the value discrepancy between both.

This conclusion can be easily reached using only reasoning based on formulae, although graphic representation provided by the ProPID tool helps support these conclusions. Nevertheless, the usefulness of ProPID is clear in the other cases where the design is not exact.

*Approximate design.* If for the above process and sensor we establish a null position error, then the controller must be type PI—as shown in as the ProPID result in Fig. 3.

In this case, there is a closed loop with three poles and a zero (see the root locus window). Only two of the poles are positioned according to the specifications and the third remains free. In addition, the zero added also influences the behaviour obtained in the closed loop—but its position is not based in the criteria associated with the desired behaviour.

In general, the PI controller does not have to satisfy the specifications. Nevertheless, using ProPID it can be seen in Fig. 3 that the desired response and the obtained response are very similar—and this enables the design to be validated. This conclusion was obtained thanks to the ProPID tool—as it automatically and interactively obtained responses in closed loop.

*Interactive design.* Nevertheless, after checking the differences between the desired and obtained responses it is often verified that the design is invalid. For example, for the process and sensor:

$$G(s) = \frac{2(s+1.5)}{(s+3)(s+2)(s+1)}, \quad H(s) = 1 \quad (9)$$

establishing dynamic and static specifications in a second order system:  $i_s = 1$  s,  $\delta = 0.1$  and  $e_p \leq 0.001$ , the ProPID tool offers the result shown in Fig. 4.

This discrepancy exists because the design only assigns one or two dominant closed loop poles, ignoring the position of the remaining poles and all of the zeros. Consequently, the classic design method based on the root locus provides PID regulators that do not necessarily satisfy the desired specifications. This is one of the justifications for designing an interactive tool such as ProPID to improve the initial design.

Following the idea mentioned previously of modifying the PID to achieve a greater similarity between the desired specifications and those obtained, the location of the PID zeros and of their gain has been modified using the previously described characteristics of interactivity provided by the represented root locus. From a practical point of view, it is the position of the derivator zero, the integrator zero, and the closed loop poles (modification of the gain) that should be experimentally adjusted. Whenever any of these parameters are modified, the root locus is updated interactively, and so is the newly obtained closed loop response. This mechanism must be tested several times before an acceptable result is obtained because, generally, the desired dynamic specifications for any process will not be obtained—because the PID has only three adjustable parameters.

In the case under study, the first step was to

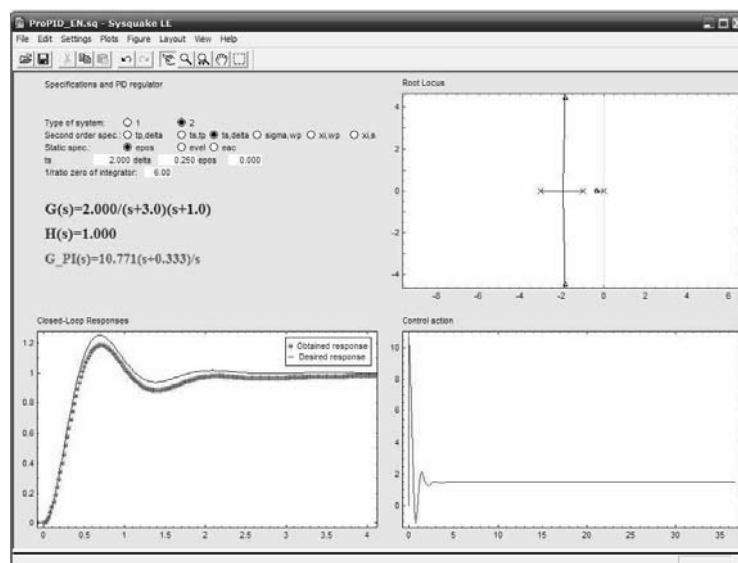


Fig. 3. Approximate design.

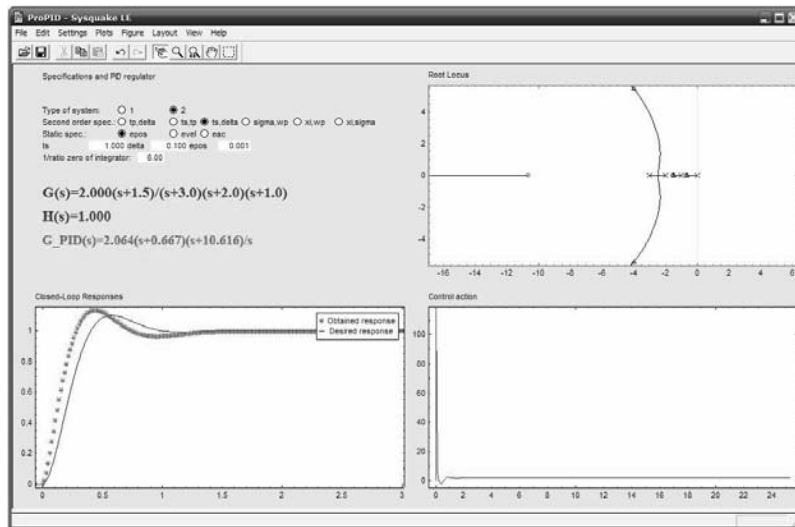


Fig. 4. Initial design.

move the derivator zero (red circle) to the right and left until it was obvious that moving it towards the right reduced the discrepancy between the specifications. Then the position of the integrator zero (blue circle) was moved to the left and the design improved. Finally, the controller gain was modified (position of the closed loop poles) until the result in Fig. 5 was obtained. The PID was specifically modified from:

$$G_{PID,1}(s) = 2.064 \frac{(s + 0.667)(s + 10.616)}{s}$$

to

$$G_{PID,2}(s) = 2.327 \frac{(s + 1.788)(s + 4.843)}{s}$$

This figure shows the great similarity between the desired specifications and those obtained after this final stage of interactive design—despite the PID having only three adjustable parameters.

This result shows that interactivity is the key to

improving the specifications obtained through the initial design based on the dominant poles, until a behaviour closer to that desired is achieved—so justifying the use of the ProPID tool.

### PROPID TOOL UTILITIES FOR CONTROL EDUCATION

Using the ProPID tool to design PID controllers in the field of process control education can significantly reduce the time required to teach how these controllers work. It can also make students' life easier during the final stage of design, when the initial controller is being improved, until specifications closer to those desired are obtained. Since solving this final stage without an interactive tool is very complex, this step was often left unexplained or unapplied in class.

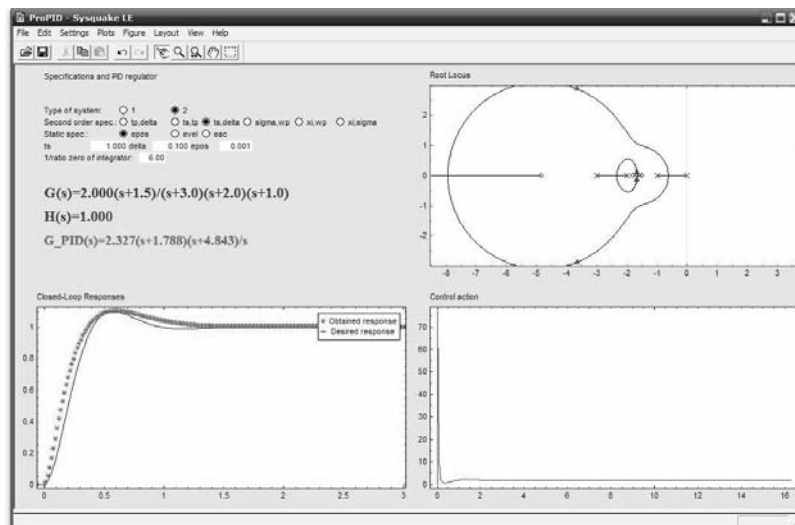


Fig. 5. Modified design.

The tool's potentialities, among others, can be summarized as follows.

- The tool can be used as a support in basic control subjects by students and teachers when comparing results obtained using classic PID design through root locus.
- It offers a new design method that can act as an alternative, or a replacement, for the classic method—if considered more suitable.
- As an iterative tool, it helps students learn about the relations existing between complex plans and time domains—thanks to its interactivity.
- It can also help students make more precise designs as compared to those of the initially selected specifications.
- It can also be used as a teaching support for explaining ideas about a specific design to students in real time.
- It is distributed free to the engineering students of the Universidad Politécnica de Valencia (Polytechnic University of Valencia).
- ProPID can be downloaded from [http://personales.upv.es/~jsalcedo/sysquake/ProPID\\_EN.sq](http://personales.upv.es/~jsalcedo/sysquake/ProPID_EN.sq).

## CONCLUSIONS AND FUTURE ACTIONS

The authors present the interactive tool ProPID, which implements an alternative strategy to root locus for designing PID controllers. This strategy provides a *direct* resolution of the characteristic equation by assigning dominant poles and by breaking it into real and imaginary parts that can be easily implemented on a computer. This new strategy is an alternative to the classic resolution

that follows the module and argument criteria and that is not particularly orientated towards computer implementation.

This tool has been implemented in Sysquake LE (free), a very interactive language that eases internal calculations and graphically enables the results of this new method of PID design to be quickly obtained. ProPID can be downloaded from: [http://personales.upv.es/~jsalcedo/sysquake/propid\\_en.sq](http://personales.upv.es/~jsalcedo/sysquake/propid_en.sq).

This tool has also proved useful for refining the design of PID controllers—so that they adjust better to the initially desired specifications. This is a task that is sometimes difficult to achieve without interactivity. This task is not easy to achieve without the help of specialised tools requiring expensive licences—something that is usually beyond a student's resources outside of laboratory practice sessions—and that tends to be a small part of the total hours of class. Nevertheless, given that ProPID is available free, a student can use it on any computer.

With regard to future areas of work, the aim in the field of education is to use this methodology to directly design discrete controllers. In the industrial field, several issues related to the controller design and actuator saturation and discrepancies between initial and final specifications have to be addressed. This tool will soon include new methodologies for PID design, including: frequency design, ITAE, Ziegler-Nichols, etc., and the ability to make comparisons between these methodologies.

*Acknowledgement*— This research has been partially financed by DPI2005-07835, MEC (Spain)-FEDER.

## REFERENCES

1. B. Wittenmark, H. Haglund and M. Johansson, Dynamic pictures and interactive learning, *IEEE Contr. Syst. Mag.*, **18**(3), 1998, pp. 26–32.
2. R. C. Dorf and R. H. Bishop, *Modern Control System*, 10th edn, Pearson Education, Upper Saddle River, NJ 07458, (2005).
3. S. Dormido, S. Dormido-Canto, R. Dormido, J. Sánchez and N. Duro, The role of interactivity in control learning, *International Journal of Engineering Education*, **21**(6), 2005, pp. 1122–1133.
4. G. C. Goodwin, S. F. Graebe and M. E. Salgado, *Control System Design*, Prentice Hall, Upper Saddle River, NJ 07458, (2001).
5. J. L. Guzmán, M. Berenguel, and S. Dormido. Interactive teaching of constrained generalized predictive control. *IEEE Contr. Syst. Mag.*, April, 2005, pp. 52–66.
6. K. Ogata, *Modern Control Engineering*, 4th edn, Prentice Hall, Upper Saddle River, NJ 07458, (2002).
7. Y. Piguet, *SysQuake: User Manual*, Calerga, Lausanne, Switzerland, (2000).
8. Ch. Salzmänn, D. Gillet and P. Huguenin, Introduction to real-time control using LabVIEW with and application of distance learning, *International Journal of Engineering Education*, **16**(2), 2000, pp. 255–272.
9. F. G. Martins, Tuning PID controllers using the ITAE criterion, *International Journal of Engineering Education*, **21**(5), 2005, pp. 867–873.
10. C. Fernández, M. A. Vicente and L. M. Jiménez, Virtual laboratories for control education: a combined methodology, *International Journal of Engineering Education*, **21**(6), 2005, pp. 1059–1067.
11. J. Sánchez, F. Esequembre, C. Martín, S. Dormido, S. Dormido-Canto, R. D. Canto, R. Pastor and A. Urquía, Easy Java simulations: an open-source tool to develop interactive virtual laboratories using MATLAB/Simulink, *International Journal of Engineering Education*, **21**(5), 2005, pp. 798–813.
12. J. P. Keller. Teaching PID and Fuzzy Controllers with LabVIEW\*, *International Journal of Engineering Education*, **16**(3), 2000, pp. 202–211.

**Miguel Martínez** is a professor in the Department of Systems Engineering and Control of the Technical University of Valencia (Spain). He has a BSc and a Ph.D. degree from the same university. He has three years of industrial experience in control engineering in cement plants. He has published several papers in conference proceedings and journals. His research interests are: applied adaptive control, predictive control, fuzzy logic, expert systems, neural networks and genetic algorithms. During part of 1994 he worked at the Department of Engineering Science of Oxford University (UK) in a new research area: genetic predictive control.

**José V. Salcedo** is a lecturer in the Department of Systems Engineering and Control of the Technical University of Valencia (Spain). He has a degree in industrial engineering and a Ph.D. from the same university. He has published several papers in conference proceedings and journals. His research interests are: applied robust control, nonlinear control, fuzzy control, and predictive control.