

## Interactive Learning Module: Basic Modelling and Identification Concepts

J.L. Guzmán\* K.J. Åström\*\* S. Dormido\*\*\* T. Hägglund\*\*  
Y. Piguet\*\*\*\* M. Berenguel\*

\* *Dep. de Lenguajes y Computación, Universidad de Almería, 04120 Almería, Spain. (Tel: +34 950 015849; e-mail:joguzman@ual.es)*

\*\* *Department of Automatic Control, Lund University, Box 118 SE-22100 Lund, Sweden (e-mail: {kja,tore}@control.lth.se)*

\*\*\* *Dep. Informática y Automática, ETSI Informática, UNED, 28040 Madrid, Spain (e-mail: sdormido@dia.uned.es)*

\*\*\*\* *Calerga Sàrl, 35, av. de la Chablière, 1004 Lausanne, Switzerland (e-mail: yves.piguet@calerga.com)*

---

**Abstract:** This paper describes an interactive tool focused on teaching and learning basic concepts about modelling and identification. Modelling and identification play fundamental roles in the field of automatic control, since it is not possible to perform a control system design without having a model describing the dynamics of the process to be controlled. The tool presented in this work is dedicated to describe the most important aspects of basic modelling centralized on first and second order systems with dead-time and zero effects. © Copyright IFAC 2008.

---

### 1. INTRODUCTION

Mathematical models are key elements in the field of automatic control. They are required for simulation as well as design in a typical control system development. Models cannot represent every aspect of reality and assumptions must be made in order to use them for control purposes. Traditionally, for control purposes the idea has been to characterize process dynamics by a few features and parameters bearing in mind to facilitate the control design procedure. From this perspective, it has been observed that many industrial processes can be modelled using first and second order system plus time delay, perhaps including zero influences.

Modelling and identification are key elements in automatic control courses, and are fundamental parts in the education of control engineers. For this reason, it is important to have tools and material oriented to the learning and teaching of modelling and identification concepts. In the literature, it is possible to find some tools related to modelling and identification, some of them oriented to identification (Wallén, 1999) and other ones dedicated to basic ideas in modelling and control design (Johansson et al., 1998). The idea of this work is to combine both purposes, modelling concepts and basic identification, into a unique interactive tool. The new tool is also provided with powerful interactive properties as it will shown in this paper. The interactive tools gives a good feel for the relations between the parameters and the response and the sensitivity of the parameters (Åström and Hägglund, 2005).

\* This work has been financed by CICYT-FEDER under Grants DPI2007-66718-C04-04 and DPI2007-61068, and by the IV PRICIT (Plan Regional de Ciencia y Tecnología de la Comunidad de Madrid), under Grant S-0505/DPI/0391.

In the process of writing the book (Åström and Hägglund, 2005) the idea emerged that it would be useful to try to develop interactive learning modules (ILMs) for modelling and control. The idea was to develop interactive learning tools which could be used for introductory control courses at universities and other schools, and for engineers in industry. The modules should be self-contained, they should be suitable both for self-study, for courses, and for demonstrations in lectures, and they should not require any additional software. In this sense, three first modules have been developed and focused on PID Control: *PID Basics*, *PID Loop Shaping*, and *PID Windup* (Guzmán et al., 2006). Therefore, in this work a new interactive tool is presented being dedicated to modelling purposes and as a new element for this collection of interactive tools.

Some years ago, the implementation of tools required a substantial effort which significantly limited their use in education. However, nowadays there exist development environments to facilitate the creation of interactive tools, such as Sysquake (Piguet, 2004), or Easy Java Simulations (Esquembre and Sánchez, 2004). Projects based on Sysquake were developed at EPFL and elsewhere, see (Piguet et al., 1999; Piguet and Longchamp, 2006; Dormido, 2003; Guzmán et al., 2005, 2006).

The interactive tool presented in this paper has been implemented in Sysquake (Piguet, 2004). The main reasons for choosing Sysquake was its power to develop interactive graphical tools and the possibility to generate executable files that can run independently and distributed without any licences.

The idea to change properties and immediately being able to see the effects of the changes is very powerful both for learning and for designing. The dynamics of the changes

provides additional information that is not available in static plots. There are many interesting issues that have to be dealt with when developing interactive tools for the automatic control field which are related to the particular graphics representations used. It is straight forward to see the effects of parameters on the graphics but not so obvious how the graphical objects should be manipulated. There are natural ways to modify pole-zero plots for example by adding poles and zeros and by dragging them. Bode plots can be manipulated by dragging the intersections of the asymptotes. However, it is less obvious how a Nyquist plot should be changed (Guzmán et al., 2006).

The interactive tools can be very helpful in education but there is a danger: *students try to learn and solve problems by manipulation without understanding*. A simple view of the idea of interactive learning is that activity promotes learning. The power of the interactive tools sometimes provokes the user's temptation to play with the different dynamic objects forgetting the real meaning of such graphical elements. Teachers must make clear to the students that interactive tool are abstractions of the theoretical concepts, and the full learning must be complemented as a mixture of both. In the same way that mathematical books are accompanied with exercises in order to understand the theoretical concepts, interactive tools should be used as support to the traditional lectures (Guzmán, 2006).

Notice that this new interactive learning module has been developed recently, and feedback from the students is not yet available. The module will be used in different control courses at Lund University, University of Almería, UNED, and EPFL. One consideration that must be kept in mind is that the tool's main feature, interactivity, cannot be easily illustrated in a written text. Nevertheless, some of the advantages of the applications are shown in the paper. The reader is cordially invited to visit the web site (at [www.calerga.com](http://www.calerga.com)) to experience the interactive features of the module.

The paper is organized as follows. In Section 2, the different models that are used in the tool are presented, as well as the selected identification algorithm. The interactive tool is presented in Section 3, and a few practical examples are given in Section 4. The paper ends with conclusions and references.

## 2. MODELLING CONCEPTS

It is well known, that most control problems are solved using PI or PID controllers being tuned using simple process models. A simple form of system identification is often made by the process engineers. Typically, they estimate gain, dead time, and time constants from open loop step responses, or relay feedback tests. The controller parameters are then obtained using simple tuning rules like the Ziegler-Nichols rules, lambda tuning, or the Amigo rules, see (Åström and Hägglund, 2005) for a comprehensive listing. For this reason, the tool presented in this paper focus on simple transfer function models that are useful for control purposes. The different kinds of available models in the tool are the following:

$$G_1(s) = \frac{K_p}{\tau s + 1} e^{-Ls} \quad (1)$$

$$G_2(s) = \frac{K_p \omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} e^{-Ls} \quad (2)$$

$$G_3(s) = \frac{K_p}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-Ls} \quad (3)$$

$$G_4(s) = \frac{(b_0 s + b_1)}{s^2 + 2\zeta \omega_n s + \omega_n^2} e^{-Ls} \quad (4)$$

$$G_5(s) = \frac{(b_0 s + b_1)}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-Ls} \quad (5)$$

where  $K_p$  represents the static gain,  $\tau$ ,  $\tau_1$ , and  $\tau_2$  are time constants,  $\zeta$  is the damping factor,  $\omega_n$  is the undamped natural frequency,  $b_0$  and  $b_1$  determine the process zero, and  $L$  is the time delay.

The tool has also been prepared for straightforward identification capabilities. The user can try to fit experimental data with some of the model structures described above by interactively modifying the associated model parameters. In order to facilitate the process engineer work, it is sometimes useful to have some automatic mechanism to perform the identification tasks. In this way, process engineers can identify the desired model automatically and also compare the results with those obtained interactively by hand.

For this reason, the tool has been provided of an automatic identification method in order to fit the experimental data with a selected model structure. The algorithm is based on minimization of the output error

$$\hat{\theta} = \arg \min_{\theta} J(\theta) = \arg \min_{\theta} \sum_t (y(t) - y_m(t, \theta))^2, \quad (6)$$

where,  $y(t)$  is the measured process output and  $y_m(t, \theta)$  is the output of the model, see Wallén (2000).

The minimization of the optimization criterion (6) has been performed using the well-known Newton-Raphson iterative method (Deuffhard, 2004). Notice that the optimization criterion (6) will typically have several local minima, since it is nonlinear in the parameters  $\theta$ .

## 3. INTERACTIVE TOOL

One of the most important features for developing Interactive Tools is the *abstraction*. In the context of this work, abstraction refers to associate theoretical concepts with graphical elements. That is, visualizing the concepts with graphical and dynamic pictures, knowing this as visualization. In order to design technical systems or simply to understand the physical laws that describe their behavior, scientists and engineers often use computers to calculate and graphically represent different magnitudes. In control engineering, these quantities include among others: time and frequency responses, poles and zeros on the complex plane, Bode, Nyquist and Nichols diagrams, phase plane, etc. Frequently these magnitudes are closely related and constitute different visions of a single reality. The understanding of these relationships is one of the keys to achieve a good learning of the basic concepts and it enables students to accurately design control systems (Dormido, 2004). The tool presented in this paper has been developed following this previous idea, as shown below.

### 3.1 Description

This section briefly describes the main features of the developed tool. The tool is focused on understanding modelling concepts and basic identification ideas using the schemes described in the previous section. As commented before, this tool has been implemented in Sysquake, a Matlab-like language with fast execution and excellent facilities for interactive graphics (Piguet, 2004). Windows, Mac, and Linux operating systems are supported. One consideration that must be kept in mind is that the tool's main feature -interactivity- cannot be easily illustrated in a written text. Nevertheless, some of the advantages of the application are shown below. The reader is cordially invited to experience the interactive features of the tool.

When developing a tool of this kind, one of the most important things that the developer has to bear in mind is the organization of the main windows and menus of the tool to facilitate the user the understanding of the control technique (Dormido, 2004).

This interactive learning module has been developed to make it possible to quickly obtain a good intuition and a good working knowledge of basic modelling concepts. As the previous modules developed (Guzmán et al., 2006), the module is composed by several parts. Two icons that can be accessed, namely, *Instructions* and *Theory*. *Instructions* gives access to a document that contains suggestions for exercises, while *Theory* provides access to relevant theory by means of the Internet. Different menus are available to load examples and choose between different configurations of the tool. The graphics can be manipulated directly by dragging points, lines, and curves or by using text-edits and sliders. Notice also that for all the graphics available in the tool, the vertical and horizontal scales can be modified using three black triangles available on the graphics ( $\blacktriangle$ ,  $\blacktriangledown$ ).

The tool is divided into two main parts, **Modelling Concepts** and **Identification**, and each part is further divided into several sections represented in Figures 1 and 2, which show two main screens of the interactive tool.

### 3.2 Modelling Concepts screen

The **Modelling Concepts** screen is shown in Figure 1. This screen is focused on understanding the main properties of the most common process transfer functions described in the previous section and that are widely used for control purposes. A symbolic representation of the model (**Modelling Transfer Function**), the associated model parameters (**Model Parameters**), and a standard pole-zero representation (**System Poles and Zeros**) are available in left-hand area of the screen. Model parameters can be modified interactively by dragging sliders or setting specific values using text-edits. In the same way, the model poles and zeros can be changed from the **System Poles and Zeros** graphic. The user can add/remove poles and zeros dragging in and dragging out the pole ( $\times$ ) and zero ( $\circ$ ) elements located at the top right side of the **System Poles and Zeros** graphic. The symbolic representation of the model shows continuously all the changes performed on the model parameters.

In the right-hand of the screen, there are different graphical elements which represent the model step response

(**System Step Response**), and the model frequency response represented in the Nyquist plot (**Nyquist**) and in the Bode diagram (**Bode Magnitude** and **Bode Phase**), respectively. All these graphics are fully interactive in such a way that the user can modify the model parameters from the different domains dragging on specific interactive elements. In this sense, it is possible to modify the static process gain using the horizontal green lines available in the **System Step Response** and **Bode Magnitude** graphics or the green point available in **Nyquist** plot. The process delay can also be modified using the vertical pink dotted line in the step response, pushing down the horizontal pink dotted line in the **Bode phase** graphic, or enlarging the **Nyquist** curve in the **Nyquist** graphic.

When the model is of first order with a dead time (FOPDT), the time constant can be interactively changed from different graphics. Its value is updated dragging on the symbol ( $\times$ ) available in the time and frequency curves. For second order models with dead time (SOPDT), the interactive elements change according to the kind of the system, considering underdamped or overdamped processes. If the process is an underdamped system the damping factor,  $\zeta$ , and the undamped natural frequency,  $w_n$ , can be modified from the different graphics. To modify the damping factor value the user can increase or decrease dragging on a red circle located at the maximum value of the curve in the step response or in the Bode magnitude diagram. In the same way, this parameter can be updated moving the red circle located at the y-axis up or down in the **Nyquist** graphic. For the undamped natural frequency, its value can be changed using the vertical red dotted lines available in the step response and Bode diagram graphics, as well as a red symbol ( $\times$ ) drawn in the **Nyquist** plot. If the model represents an overdamped process, two different time constants ( $T_1$  and  $T_2$ ) are available. Their values are represented in the different graphics expressed by the symbol ( $\times$ ) and being modified such as described for the time constant in a FOPDT. Notice that the user can interactively change between overdamped and underdamped systems.

### 3.3 Identification screen

The second screen of the tool is shown in Figure 2 and is dedicated to **Identification** purposes. This tool consists in, given the output response of a real process to a specific input signal, identifying the parameters of a process transfer function so that it captures the system dynamics. The user can choose between models with three (FOPDT), four (SOPDT), or five (SOPDT with a zero) parameters. The identification task can be performed in two different ways: by hand playing interactively with the model parameters, or automatically using the identification algorithm described in the previous section. For these purposes, this screen is divided into different parts. On the top of the Figure 2, a symbolic representation of the process transfer function and some sliders and text-edits are available with the same purposes as in the **Modelling Concepts** screen. In this same area of the screen, the type of the model can be chosen selecting the desired number of parameters.

Below these elements, there are two different graphics which represent the input signal (**Input**) and the output

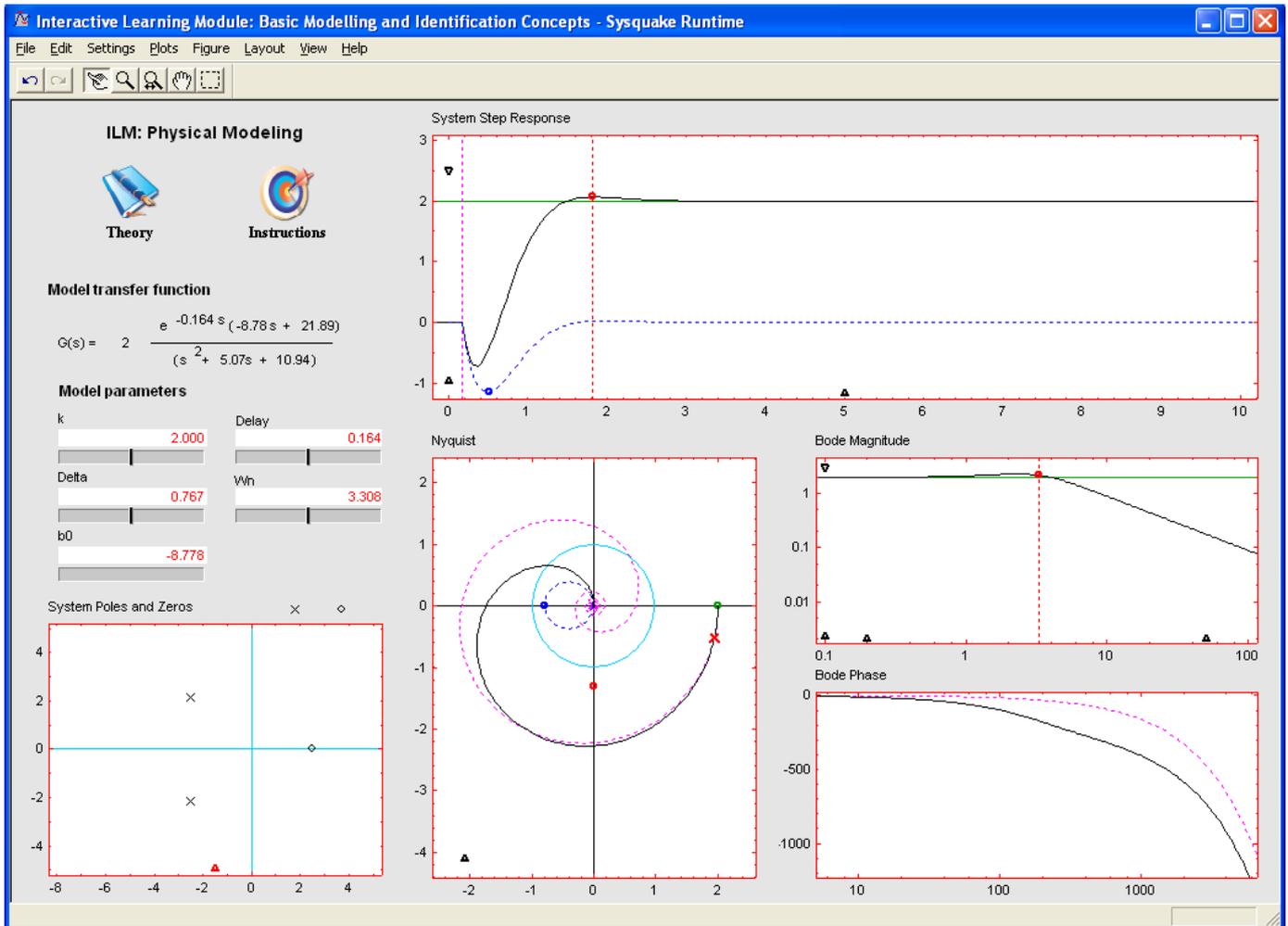


Fig. 1. Interactive tool user interface for Modelling Concepts area.

response (Output) of the system to be identified. By default, a typical step test is preloaded in the tool as shown in Figure 2. Other options to load relay test examples or data from the user are available in the **Settings** menu. Once the data is loaded, the user can try to fit the model in two different ways as commented before. The user can interactively modify the model parameters in the **Output** graphic, as described for the **Modelling Concepts** screen for the time domain, or using the sliders and text-edits. This option allows the user to manually find the adequate model for the given experimental data. Depending on the type of model selected, the user is able to modify the time constants, the damping factor, the undamped natural frequency, etc.

The model can also be fitted to the data automatically using the menu option called **Optimization fitting**. This option uses the optimization algorithm described in the previous section, which tries to obtain the adequate model for the given data considering the number of model parameters selected. On the top right corner of the **Input** graphic the mean square error between the model output and the experimental data is shown being a reference measurement to obtain the desired model.

Finally, notice that most plants in process industry are more or less nonlinear presenting different behavior across

the operating range, see for example the case in Figure 1). For this reason, if the process is normally operated within a small region, it is often reasonable to use a linearized model around the operating point (Wallén, 1999). Therefore, the tool presented in this paper allows to select the fitting area interactively in order to be able to identify different operating points. In this way, there are three black lines and two black boxes available in the **Output** graphic, which allow to delimitate the desired fitting area. Graphically, the fitting area is drawn with white background and the rest area with yellow background.

#### 4. PRACTICAL EXAMPLES

In this section, two examples will illustrate some of the features of the new tool.

##### 4.1 Nonlinear behavior

As commented above, most industrial processes present some kind of nonlinearities. It sometimes happens that when step input signal are applied to a system across the operating range, different dynamics are obtained. In these cases, it could be useful to fit different linear models for each operating point. In this way, the resulting models may

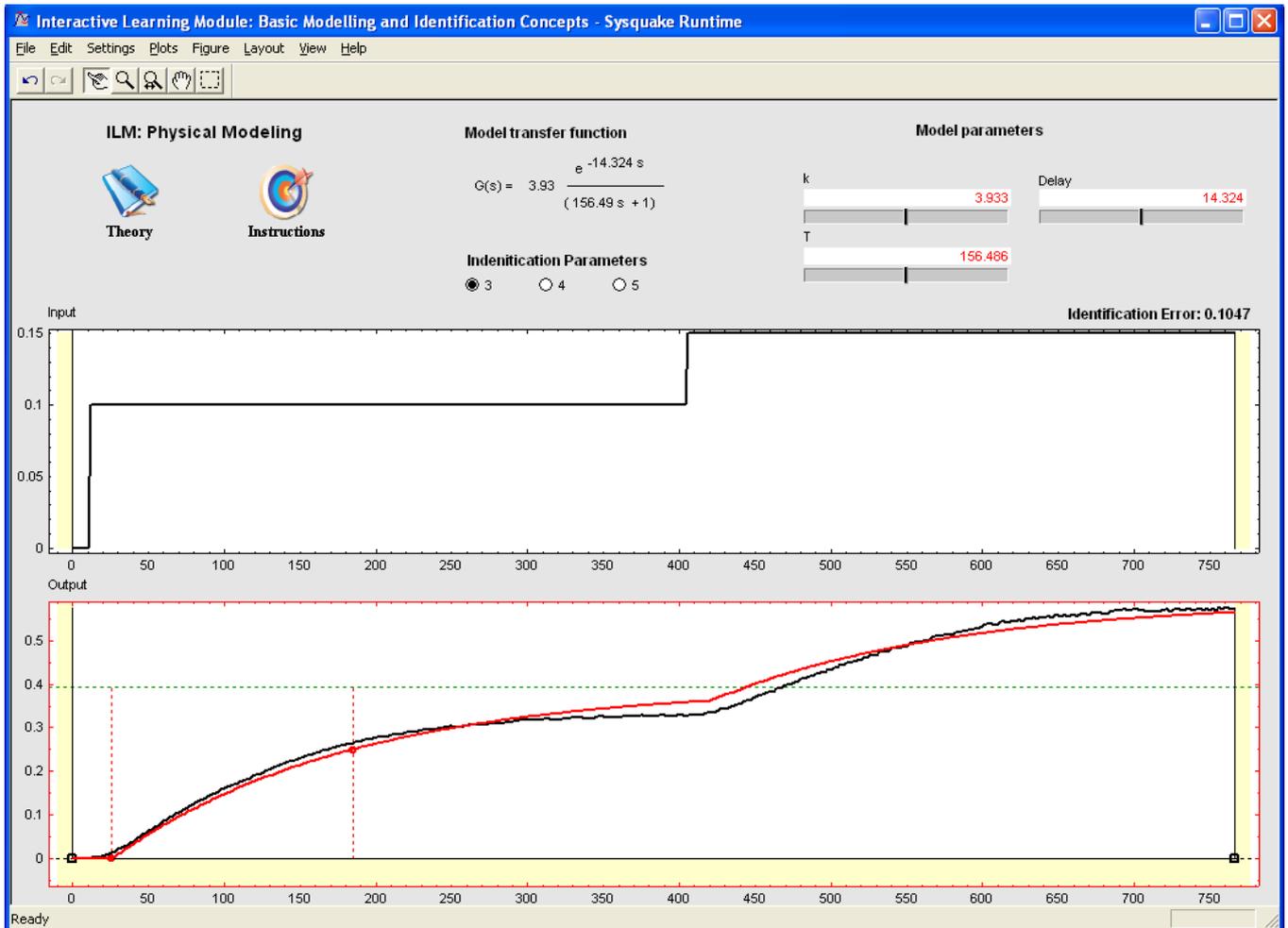


Fig. 2. Interactive tool user interface for Identification area.

be use to design, for instance, a typical gain scheduling scheme based on PID controllers.

Figure 2 shows this kind of behavior where there are two different operating points. Therefore, some exercises can be performed with the interactive tool in order to understand this phenomenon. For instance, the students can try to fix an unique model capturing approximately both operating point dynamics. This is the result obtained in Figure 2 where a FOPDT model has been fitted with  $K_p = 3.93$ ,  $\tau = 156.48$ , and  $L = 14.32$ .

After that, the students can try to identify models for each operating point obtaining very different results such as shown in Figure 3 for the first step. In this case, two different models are fitted described by  $p_1 = (K_p = 3.45, \tau = 111.55, L = 16.73)$  and  $p_2 = (K_p = 1.9, \tau = 182.05, L = 0)$  for the first and second operating point, respectively.

#### 4.2 Relay feedback

Systems identification using relay feedback is very useful for tuning PID controllers. Relay autotuning has been successful in industrial process control application for PID controllers (Åström and Hägglund, 2005; Wang et al., 1997) and for model-based controllers which need transfer function models (Hang et al., 1995).

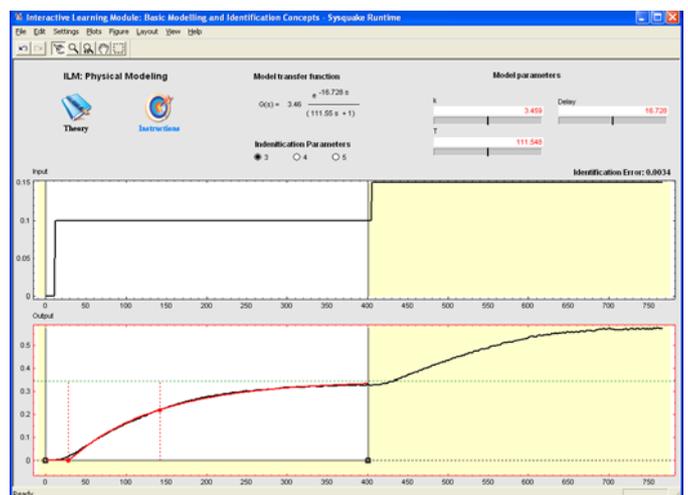


Fig. 3. Step test with nonlinear behavior.

The idea of this technique is the observation that it is possible to create an oscillation with the ultimate frequency automatically by using relay feedback. To make the experiment the system is connected in a feedback loop with a relay function. For many systems there will then be an oscillation where the control signal is a square wave and the output is close to a sinusoid.

This kind of tests can be loaded in the developed interactive tool. A relay feedback test taken from (Åström and Hägglund, 2005) is available at the Settings menu of the tool such as shown in Figure 4.

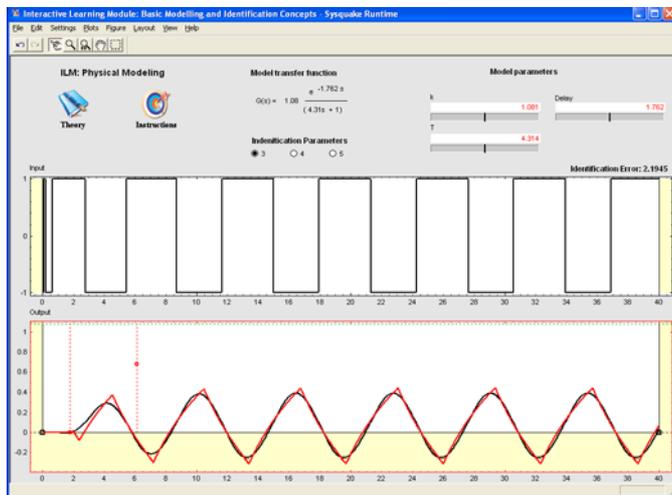


Fig. 4. Relay feedback test available from the Settings menu of the tool.

Some interesting exercises can be proposed to the students related to this kind of tests. For instance, it can be required to obtain a FOPDT for this data considering that is the most commonly used dynamic model for the industrial process. So, the students first can calculate the periods and amplitudes of limit cycles under relay feedback and use this information to model a FOPDT dynamics (Wang et al., 1997). After that, the students can use the interactive tool to validate the obtained model and playing interactively with the parameters to understand the results. The results obtained theoretically can be compared with those obtained by the optimization algorithm available in the interactive tool.

Figure 4 shows the results of using the optimization algorithm described in Section 2, where for the given experimental data a FOPDT system has been fitted with the model parameters  $K_p = 1.081$ ,  $\tau = 4.314$ , and  $L = 1.762$ .

## 5. CONCLUSION

A new interactive learning module for basic modelling and identification purposes has been described. The module belongs to the collection of interactive learning modules (Guzmán et al., 2006) that are an attempt to make figures in the book (Åström and Hägglund, 2005) interactive. The purpose is to enhance learning by exploiting the advantages of immediately seeing the effects of changes that can never be shown in static pictures. The module is implemented in Sysquake, a Matlab-like language with fast execution and excellent facilities for interactive graphics. The modules are available on the web at [www.calerga.com](http://www.calerga.com). Additional modules for design, coupled loops, and feedforward are under development.

## REFERENCES

- K.J. Åström and T. Hägglund. *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, 2005.
- P. Deuffhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Springer Series in Computational Mathematics, Berlin, 2004.
- S. Dormido. The role of interactivity in control learning. *Proceedings of the 6<sup>th</sup> IFAC Symposium on Advances in Control Education*, pages 11–22, 2003. Oulu, Finland.
- S. Dormido. Control learning: Present and future. *Annual Reviews in Control*, 28:115–136, 2004.
- F. Esquembre and J. Sánchez. *Easy Java Simulations 3.3. How to use Ejs with Matlab and Simulink*. <http://fem.um.es/Ejs/>, 2004.
- J.L. Guzmán, K.J. Åström, S. Dormido, T. Hägglund, and Y. Pigué. Interactive learning modules for PID control. *Proceedings of the 7<sup>th</sup> IFAC Symposium on Advances in Control Education ACE06*, 2006. Madrid (Spain).
- J.L. Guzmán, M. Berenguel, and S. Dormido. Interactive teaching of constrained generalized predictive control. *IEEE Control Systems Magazine*, 25(2):52–66, 2005.
- J.L. Guzmán. *Interactive Control System Design*. Phd thesis, University of Almería, Spain, 2006.
- C. C. Hang, Q. G. Wang, and L. S. Cao. Self-tuning smith predictors for processes with long dead time. *International Journal of Adapting Control Signal Process*, 9: 255–270, 1995.
- M. Johansson, M. Gäfvert, and K.J. Åström. Interactive tools for education in automatic control. *IEEE Control Systems Magazine*, 18(3):33–40, 1998.
- Y. Pigué, U. Holmberg, and R. Longchamp. Instantaneous performance visualization for graphical control design methods. *Proceedings of the 14<sup>th</sup> IFAC World Congress*, 1999. Beijing (China).
- Y. Pigué and R. Longchamp. Interactive applications in a mandatory control course. *Proceedings of the 7<sup>th</sup> IFAC Symposium on Advances in Control Education ACE06*, 2006. Madrid (Spain).
- Y. Pigué. *Sysquake 3 User Manual*. Calerga Sàrl, Lausanne, Switzerland, 2004.
- A. Wallén. A tool for rapid system identification. In *Proceedings 1999 IEEE Int. Conf. Control Applications and the Symp. Computer Aided Control Systems Design (CCA'99 & CACSD'99)*, Kohala Coast, Hawaii, August 1999.
- A. Wallén. *Tool for Autonomous Process Control*. Phd thesis, Lund University, Sweden, 2000.
- Q.-G. Wang, C.-C. Hang, and B. Zou. Low-order modeling from relay feedback. *Industrial & Engineering Chemistry Research*, 36(2):375–381, 1997.