Advances in Engineering Software 45 (2012) 115–123

Contents lists available at SciVerse ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

An interactive software tool for system identification

J.L. Guzmán [a], D.E. Rivera [b], S. Dormido [c,*], M. Berenguel [a]

[a] Dep. de Lenguajes y Computación, Universidad de Almería, 04120 Almería, Spain
[b] School for Engineering of Matter, Transport, and Energy, Arizona State University, Tempe, AZ 85287-6106, USA
[c] Dep. Informática y Automática, UNED, C/Juan del Rosal, 16, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 27 July 2010
Received in revised form 21 July 2011
Accepted 31 August 2011
Available online 17 October 2011

Keywords:
System identification education
Interactivity
Experimental design
Prediction-error estimation
Model validation
Engineers training

ABSTRACT

The paper describes the conceptual basis, main features and functionality of an interactive software tool developed in support of system identification education and discovery. This Interactive Tool for System Identification Education (*ITSIE*) has been developed using Sysquake, a Matlab-like language with fast execution and excellent facilities for interactive graphics, and is deliverd as a stand-alone executable that is readily accessible to students and engineers. *ITSIE* provides two distinct functional modes that are very useful from an educational and industrial point of view. The simulation mode enables the user to evaluate the main stages of system identification, from input signal design through model validation, simultaneously and interactively in one screen on a user-specified dynamical system. The real data mode allows the user to load experimental data obtained externally and identify suitable models in an interactive fashion. The interactive tool enables students and engineers in industry to discover a myriad of fundamental system identification concepts with a much lower learning curve than existing methods.

1. Introduction

System identification deals with the problem of building dynamical models of systems from experimental data, and is a key component in control engineering practice [1]. Consequently, system identification education forms an essential part of any comprehensive control engineering curriculum, and as such requires flexible and simple-to-use software tools. There are many powerful software tools available for system identification [2–4], but these present several disadvantages when viewed from a primarily educational point view. Normally, these tools do not evaluate all stages of the system identification process (experimental design, model structure selection, parameter estimation, and validation) in an integrated fashion. Furthermore, available tools provide substantial amounts of information in many different screens, which can be quite confusing for users. Finally, system identification is naturally performed in an iterative manner, that is, it involves a refining process where subsequent stages need to be recalculated step by step when a parameter or specification is modified. Failure to accomplish these iterations in a manner transparent for the user diminishes any educational benefits since students lose the connection with theoretical ideas and become less motivated. Thus, a new generation of software tools addressing these concerns are needed in support of advancing system identification education.

Advances in information technologies have resulted in novel instructional methods that increase student motivation and improve educational outcomes. The automatic control field is a clear example where these advances have provided powerful software tools for engineer training [5–8]. Interactive software tools have been proven as particularly useful techniques with high impact on control education [9–12]. Interactive tools provide a real-time connection between decisions made during the design phase and results obtained in the analysis phase of any control-related project. Because system identification is a field rich in visual content that can be represented intuitively and geometrically [3], it naturally lends itself to interactivity. A novel interactive software tool for system identification has been incrementally developed in [13,14] based on these ideas leading to the Interactive Tool for System Identification Education (*ITSIE*) presented in this paper. It includes all stages of system identification in the same screen, with the different stages connected interactively in such a way that a modification in one stage is automatically visualized in the remaining stages.

The main objective of this paper is thus to explore this vast untapped potential of interactivity in the identification field. The tool draws from the experience of the authors in teaching system identification courses in both short and semester-long formats, and to diverse audiences.

The work described in [13,14] represents the initial efforts to develop an interactive software tool for identification. This paper presents an updated version of the tool, providing features such as enabling the instructor to configure his/her own simulated

* Corresponding author.
E-mail addresses: joguzman@ual.es (J.L. Guzmán), daniel.rivera@asu.edu (D.E. Rivera), sdormido@dia.uned.es (S. Dormido), beren@ual.es (M. Berenguel).

0965-9978/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2011.09.013

example that could be shared with students, loading external data in order to identify models from real experiments, reports generation, among others. The tool consists of a graphical interface depicting the various stages of system identification. The paper emphasis is on describing the tool that examines the integrated effect of experimental design and model structure selection on prediction-error estimation. Both pseudo-random binary sequence (PRBS) and minimum crest factor multisine inputs are applied for AutoRegressive EXogenous (ARX), AutoRegressive Moving Average eXogenous (ARMAX), Output Error (OE), Box–Jenkins (BJ), and State Space (SS) estimation of this system [15]. Experimental duration, estimation and cross-validation data sets, input signal bandwidth and magnitude, and model structure are evaluated under varying signal-to-noise ratios, with all results computed and displayed interactively to the user. The interactive tool is coded in Sysquake, a Matlab-like language with fast execution and excellent facilities for interactive graphics [16]. Executable files for the modules that do not require the Sysquake software to operate are in the public domain and available for the main operating systems (http://aer.ual.es/ITSIE/).

The paper is organized as follows: a brief description on the theoretical background behind the tool is presented in Section 2. A summary of the tool's functionality is presented in Section 3. A series of illustrative examples are presented in Section 4, while Section 5 is devoted to discussing some pedagogical experiences with the tool. The paper concludes with a brief discussion.

2. Theoretical background

In *ITSIE*, the plant to be identified consists of a discrete-time system sampled at a value specified by the user (default value $T = 1$ sec) and subject to noise and disturbances according to

$$y(t) = p^*(q)\ (u(t) + n_1(t)) + n_2(t). \tag{1}$$

In (1), $y(t)$ is the measured output signal and $u(t)$ is the input signal that is designed by the user, $p^*(q)$ is the zero-order-hold-equivalent transfer function for $p(s)$, where q is the forward-shift operator. The system is subject to two stationary white noise sources (n_1 and n_2) introduced at different locations in the plant. n_1 allows evaluating the effects of autocorrelated disturbances in the data, while n_2 introduces white noise directly to the output signal.

A comprehensive system identification procedure consists of experimental design and execution, data preprocessing, model structure selection and parameter estimation, and model validation; these are standard designations for the stages of system identification, as shown in classical texts such as [1]. The following functionality is emphasized in the tool:

- *Experimental design and execution.* The success of any identification methodology hinges on the availability of an informative input/output data set obtained from a sensibly designed identification experiment. In *ITSIE*, deterministic, periodic signals relying on pseudo-random binary sequence (PRBS) and multisine inputs are considered. A PRBS is a binary signal generated by using shift register modulo 2 addition. One cycle of a PRBS sequence is determined by the number of registers n_r and the switching time T_{sw}. The signal repeats itself after N_sT_{sw} units of time, where $N_s = 2^{n_r} - 1$. The power spectral density for a PRBS signal is given by

$$\Phi_u(\omega) = \frac{a_{\mathrm{mag}}^2 (N_s+1) T_{sw}}{N_s} \left[\frac{\sin\left(\frac{\omega T_{sw}}{2}\right)}{\frac{\omega T_{sw}}{2}} \right]^2, \tag{2}$$

where a_{mag} is the magnitude of the PRBS signal.
Multisine signals are deterministic, periodic signals, represented in the single input case by the equation

$$u(k) = \lambda \sum_{i=1}^{n_s} \sqrt{2\alpha_i}\ cos(\omega_i kT + \Phi_i) \tag{3}$$
$$\omega_i = 2\pi i / N_s T, \quad n_s \leqslant N_s/2$$

The power spectrum of the multisine input

$$\Phi_u(\omega_i) = \left(\frac{\lambda^2 \alpha_i}{2} N_s \right) \quad i = 1, \ldots, n_s \tag{4}$$

is directly specified through the selection of the scaling factor λ, the Fourier coefficients α_i, the number of harmonics n_s, and the signal length N_s. In (3), Φ_i represents the phase angles. In multisine inputs, the choice of phase angles Φ_i does not influence the power spectrum, but it does strongly influence plant-friendly metrics such as crest factor [17]. Both the work of [18], who derives a closed-form formula $\Phi_i = 2\pi \sum_{j=1}^{i} j\alpha_i$ to select the phases in (3) and the successive p-norm approach by [19] are implemented in *ITSIE*.
Both direct parameter specification and applying time constant-based guidelines for input design are evaluated in the tool [20]. In practice, little is known about the process dynamics at the start of identification testing, and plant operating restrictions will discourage excessively long or very intrusive identification experiments. A guideline that provides a suitable estimate of the frequency band over which excitation is required is [20,21]

$$\frac{1}{\beta_s \tau_{\mathrm{dom}}^H} \leqslant \omega \leqslant \frac{\alpha_s}{\tau_{\mathrm{dom}}^L}, \tag{5}$$

where τ_{dom}^H and τ_{dom}^L are high and low estimates of the dominant time constant, and β_s is an integer factor representing the settling time of the process. For example, $\beta_s = 3$; specifies the low frequency bound using the 95% settling time ($T_{95\%}$) of the process. α_s, meanwhile, is a factor representing the closed-loop speed of response, written as a multiple of the open-loop response time.
Eq. (5) is used in *ITSIE* to specify design variables in both PRBS and multisine inputs. Expressions for specifying T_{sw} and n_r based on (5) are developed in [22]:

$$T_{sw} \leqslant \frac{2.8\tau_{\mathrm{dom}}^L}{\alpha_s}, \quad N_s = 2^{n_r} - 1 \geqslant \frac{2\pi\beta_s\tau_{\mathrm{dom}}^H}{T_{sw}} \tag{6}$$

n_r and N_s are integer values, while T_{sw} is an integer multiple of the sampling time T. Similarly, Eq. (5) can also be used to specify design variables in multisine inputs, using guidelines found in [21]

$$N_s \geqslant \frac{2\pi\beta_s\tau_{dom}^H}{T}, \qquad n_s \geqslant \frac{N_s T \alpha_s}{2\pi\tau_{dom}^L} \tag{7}$$

In both cases increasing α_s and β_s will widen the frequency band of emphasis in the input signal and increase the resolution of the input signal spectrum. To reduce model variance it is beneficial to apply the highest input signal amplitudes a_{mag} or λ that operations will allow, and implement the greatest number of input cycles m possible. In practice, decisions regarding the magnitude of the input signal, spectral content, and experimental test duration are dictated by physical limitations, economics, and safety considerations [1].

- *Data preprocessing. ITSIE* data preprocessing supports mean subtraction, differencing, and subtraction of baseline values; mean detrending is applied by default.
- *Model structure selection and parameter estimation. ITSIE* examines the general family of prediction-error models which corresponds to

$$A(q)y(t) = \frac{B(q)}{F(q)} u(t - nk) + \frac{C(q)}{D(q)} e(t) \tag{8}$$

or

$$y(t) = \tilde{p}(q)u(t) + \tilde{p}_e(q)e(t) \tag{9}$$

where

$$A(q) = 1 + a_1q^{-1} + \cdots + a_{n_a}q^{-n_a}$$
$$B(q) = b_1 + b_2q^{-1} + \cdots + b_{n_b}q^{-n_b+1}$$
$$C(q) = 1 + c_1q^{-1} + \cdots + c_{n_c}q^{-n_c}$$
$$D(q) = 1 + d_1q^{-1} + \cdots + d_{n_d}q^{-n_d}$$
$$F(q) = 1 + f_1q^{-1} + \cdots + f_{n_f}q^{-n_f}$$

and *nk* is the dead time in samples. The five most popular PEM models shown in Table 1 are evaluated in *ITSIE*, with FIR belonging as a subset of ARX models. The tool also includes PEM estimation of state-space (SS) models. Although these models are not analyzed in detail in this paper for space reason, they can be easily used in the tool to be compared with other available methods.

As noted in [1], PEM estimation involves either linear and nonlinear regression, depending on the model structure being evaluated.

$$\arg\min_{\tilde{p},\tilde{p}_e} \frac{1}{N}\sum_{i=1}^{N} e^2(i) = \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\left[y - \varphi^T(t|\theta)\theta\right]^2 \quad (10)$$

where N represents the number of data, θ is a vector including the model parameters to be identified and $\varphi^T(t|\theta)$ is the model output for a given combination of the model parameters θ.

The use of Parseval's Theorem enables a frequency-domain analysis of bias effects in PEM estimation that allows deep insights into the selection of design variables for these techniques. As the number of observations $N \to \infty$, the least-squares estimation problem denoted by (10) can be written as:

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} e^2(t) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_e(\omega)d\omega \quad (11)$$

where $\Phi_e(\omega)$, the prediction-error power spectrum is

$$\Phi_e(\omega) = \frac{1}{|\tilde{p}_e(e^{j\omega})|^2}\Big(|p^*(e^{j\omega}) - \tilde{p}(e^{j\omega})|^2\Phi_u(\omega) + |p^*(e^{j\omega})|^2\sigma_{n_1}^2 + \sigma_{n_2}^2\Big) \quad (12)$$

where p^* is the frequency domain estimates of the true plant, which are constructed by taking the discrete Fourier transform on the filtered time-domain data.

Eq. (12) helps explain systematic bias effects in identification, which can be readily explored in *ITSIE*. This includes issues relating to the spectral content in the input signal, bias that is introduced (or removed) by the choice of model structure (particularly the noise model), and the associated multi-objective optimization problem resulting from varying magnitudes of the noise variances $\sigma_{n_1}^2$ and $\sigma_{n_2}^2$.

- *Model validation.* In *ITSIE*, model validation consists principally of classical methods of simulation, cross-validation, residual analysis on the prediction errors, and step responses. To enhance its educational value, in the simulation mode the step response of the true plant is presented alongside that generated by the estimated models. The percent of the output variance explained by each model on the cross-validation data set is also reported.

Table 1
Prediction-error model structures evaluated in *ITSIE*.

Method	$\tilde{p}(q)$	$\tilde{p}_e(q)$	A	B	C	D	F
ARX	$\frac{B(q)}{A(q)}q^{-nk}$	$\frac{1}{A(q)}$	$A(q)$	$B(q)$	$C(q) = 1$	$D(q) = 1$	$F(q) = 1$
ARMAX	$\frac{B(q)}{A(q)}q^{-nk}$	$\frac{C(q)}{A(q)}$	$A(q)$	$B(q)$	$C(q)$	$D(q) = 1$	$F(q) = 1$
FIR	$B(q)q^{-nk}$	1	$A(q) = 1$	$B(q)$	$C(q) = 1$	$D(q) = 1$	$F(q) = 1$
Box–Jenkins	$\frac{B(q)}{F(q)}q^{-nk}$	$\frac{C(q)}{D(q)}$	$A(q) = 1$	$B(q)$	$C(q)$	$D(q)$	$F(q)$
Output error	$\frac{B(q)}{F(q)}q^{-nk}$	1	$A(q) = 1$	$B(q)$	$C(q) = 1$	$D(q) = 1$	$F(q)$

Leveraging the interplay between the various stages of the identification problem is readily supported in *ITSIE*. One example is ARX estimation, where model structure selection can be accomplished without substantial user intervention through the sensible use of cross-validation. Because ARX parameter estimation consists of solving a linear least squares problem, a large number of model structures defined by ranges for n_a, n_b and n_k can be evaluated without incurring significant computational burden. The model order that minimizes the loss function over a cross-validation data set can be obtained without iteration.

3. Interactive tool description

This section briefly describes the functionality of the developed tool, which highlights the theoretical concepts described in the previous section. The tool is freely available through http://aer.ual.es/ITSIE/ and does not require a Sysquake license in order to execute [13].

Sysquake is a Matlab-like programming environment with excellent facilities for interactive graphics. The implementation of tools in Sysquake is divided in three main different programming components: calculation functions, graphical functions (graphics and events), and dependence functions. The calculation functions incorporate the mathematical calculations to be done when any parameter is modified in the tool. The interactive graphical part in Sysquake is composed of graphs and events. Graphs are dedicated only to show the mathematical results and display the interactive elements, such as, curves, lines, sliders, text-boxes, and so on, where an interactive ID is assigned to each graphical element. Then, a set of events is implemented to capture any interactive change on the graphical area of the tool (by using the interactive ID of the graphical elements) and to provide a new stage of the parameters according to that change. Finally, the dependence functions are coded to make the connection between the events and the calculation functions. Therefore, each time an event occurs, the new stage of parameters is used by the dependence functions to detect all the elements affected by this change. Then, the new stage of all affected parameters is used by the calculation functions to calculate the new mathematical results, and this new results are shown by the graphical part. Hence, everything is interconnected and thus providing powerful interactive capabilities.

One consideration that must be kept in mind is that the tool's main feature – interactivity – cannot be easily illustrated with written text. Nonetheless, some of the features and advantages of the application are shown below. The reader is cordially invited to download the tool and personally experience its interactive features.

When developing a tool of this kind, one of the most important considerations that the developer needs to keep in mind is the organization of the main windows and menus to facilitate to the user an understanding of the identification technique [6,11]. The main screen has been organized from an engineering point view, in such a way that the most important stages of the system identification process are displayed in the screen at the same time. Thus, the user can easily work according to a typical engineering procedure by changing parameters of the different system identification stages and analyze how these changes are affecting to the rest of the phases. The tool has two different modes, a simulation mode (depicted in Figs. 1 and 2) and a real data mode, represented in Fig. 4. The ensuing subsections briefly describe the main features of these modes.

3.1. Simulation mode

In this mode, a user-specified simulated process is evaluated. The graphical distribution has been performed according to the

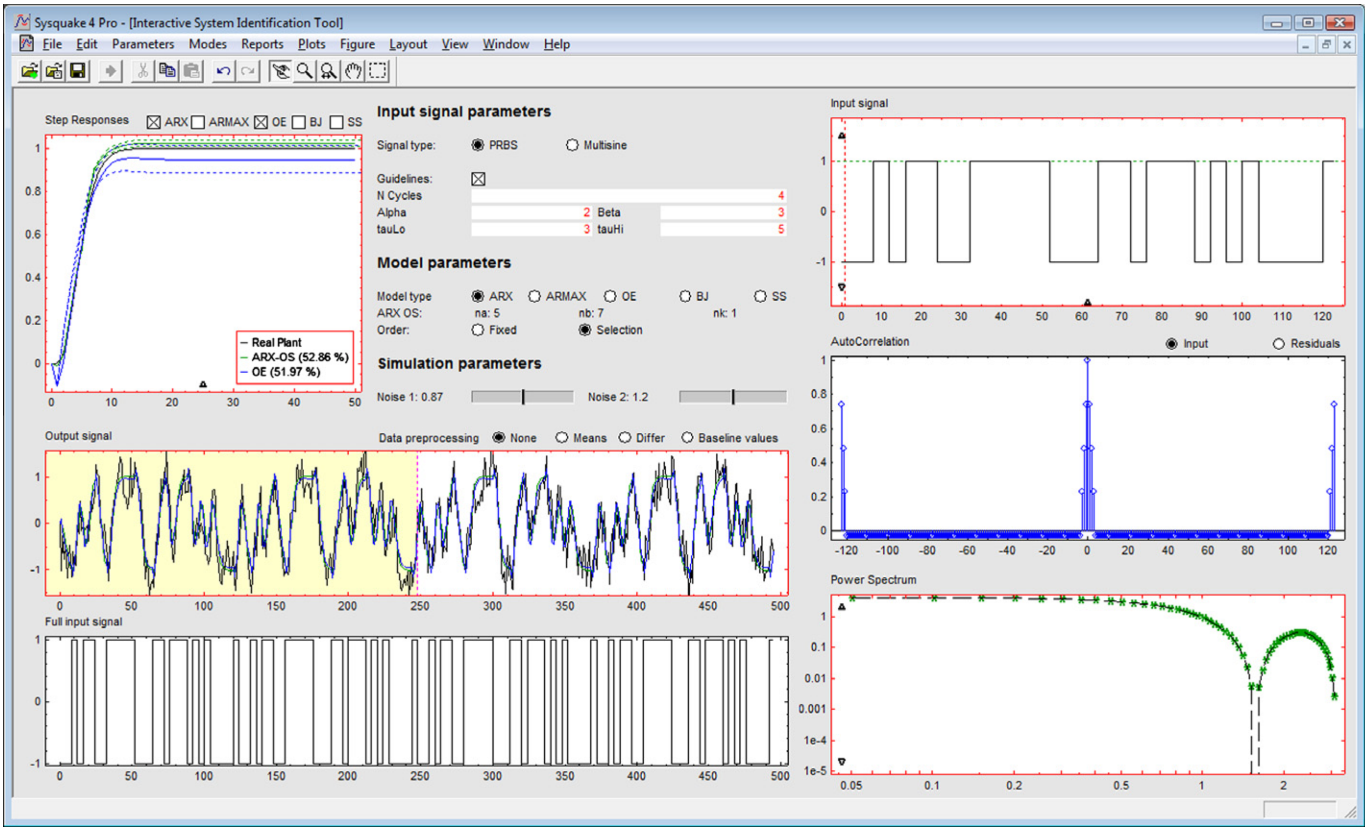

Fig. 1. *ITSIE* interactive tool user interface demonstrating four cycles of a PRBS input applied to a simulated fifth-order system. The time-constant guidelines from Section 2 are used to define input parameters. An OE-[2 2 1] model is compared with an ARX-[5 7 1] model obtained from exhaustive order selection on a cross-validation data set.

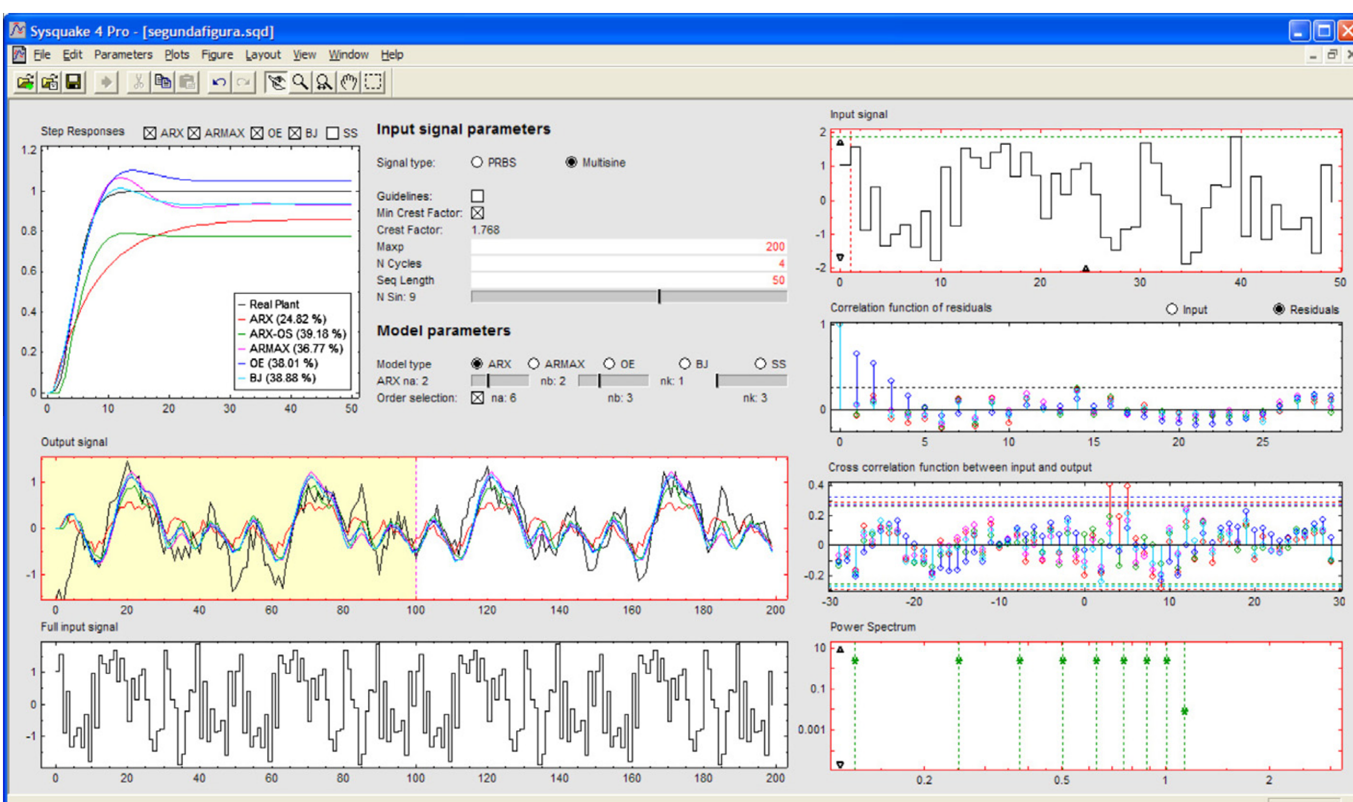

Fig. 2. *ITSIE* interactive tool user interface depicting four cycles of a minimum crest factor multisine input with phases per [19] from directly specified signal parameters. Model estimates for ARX, ARMAX, OE, and Box–Jenkins estimation are shown along with residual analysis of the prediction errors.

most important steps in a system identification process, described as follows (see Fig. 1):

- *Plant definition and simulation parameters*. The central part of the tool in this mode has a section called Simulation parameters, which allows interactively modifying the noise sources of the simulated process. Two sliders are available. The first one, Noise 1, allows modifying the noise source $n_1(t)$ in Eq. (1) and the second one, Noise 2, is used to change the noise source $n_2(t)$ in the same equation. On the other hand, other simulation parameters, such as sampling time, order selection limits, confidence intervals and baseline values are available from an entry at the Parameters menu. Notice that the sampling time can be also modified from the Input signal graphic by dragging on the red vertical line. Furthermore, the simulated process can be configured from the Modes → Simulation menu (moreover different models can be used for the simulated process and for the n_1 filter, although by default both are the same, as shown in Eq. (1)), that also includes a couple of examples: a fifth-order system and a fluidized-bed calciner plant. The process model configuration can also be loaded and stored from files.
- *Input design*. A parameter definition section and three interactive graphics characterize the input design stage. The parameter definition section is called Input signal parameters, being located at the top of the middle section of the tool. The three graphics are located at the right-hand side of the tool, namely, Input signal, Autocorrelation, and dPower Spectrum, representing one cycle of the input signal, the input signal autocorrelation, and the input signal power spectrum, respectively (see Figs. 1 and 2). From the Input signal parameters area, the user can choose the type of the input signal (PRBS as shown in Fig. 1 or multisine as shown in Fig. 2) and whether to use the checkbox called Guidelines to decide between specifying the input signal directly or following the guidelines mentioned in Section 2. When the user does not select the guidelines, that is, the Guidelines checkbox is not active, the input signal parameters can be interactively modified using specific sliders or dragging on the graphics. For instance, if the PRBS is selected without activating the Guidelines checkbox (such as shown in Fig. 3), a text edit and two sliders appear to modify the number of cycles (N Cycles), the number of registers (N Reg), and the switching time (Tsw), respectively. At the same time, from the Input signal graphic, it is possible to modify the switching time dragging on the magenta vertical line, the signal amplitude using the green horizontal line, and the number of cycles dragging on the small black triangle located on the x-axis. Furthermore, the number of registers and the switching time can be changed from the Power Spectrum graphic using the green vertical lines. The user can rely on these interactive features to understand the influence of input signal parameters from different points-of-view.
- *Model structure selection and parameter estimation*. On top of the Step responses graphic, located on the upper left-hand side of the tool, there is a set of checkboxes allowing to activate the different model structures, namely, ARX, ARMAX, OE, BJ, and SS. Once a model structure is selected, the estimation and validation results for that model are shown in corresponding parts of the tool. Below the Input signal parameters section there is an area called Model parameters showing parameters to modify the orders of the different model structures. Several radio buttons are available to choose between the different model structures. Once a model structure is selected, different sliders appear making it possible to modify the associated orders interactively. For instance, if an ARX model is chosen, sliders representing n_a, n_b, and n_k are shown. Furthermore, for the case of the ARX model structure, a checkbox is displayed to activate the automatic order selection mode using cross-validation, as described in Section 2. By default, the parameter ranges are set to $n_a = 1 \ldots 10$, $n_b = 1 \ldots 10$ and $n_k = 1 \ldots 10$, but these limits can be changed from the Parameters menu (order selection limits).
  Once an input signal has been configured, the final input with all the desired cycles is shown in a graphic called Full input signal, which is located at the lower-left corner of the tool. This full input signal is applied to the simulated plant with noise in order to obtain the simulated "real data" (shown in black in the Output signal graphic), which is used as real process data in the estimation and validation process. In the Output signal graphic, an interactive magenta vertical line defines the estimation and validation data sets. The area shown in yellow (to the left of the vertical line) specifies the estimation data, whereas the white area represents the validation data (on the right side of the vertical line). Therefore, when a model structure is selected, this estimation data is used to estimate the model parameters and the validation data to test the resulting model. Then, for each selected model structure, the full input signal is applied to the obtained model, and the results are shown in the Output signal graphic together with the original data (by default a fifth-order system is used, that can be changed using the Modes → Configuration → Model configuration option). Different colors are used to distinguish between signals, black for the original data (by default the fifth-order system), red for ARX, green for ARX with order selection, magenta for ARMAX, blue for OE, cyan for BJ, and orange for SS. These colors are consistently used in different parts of the tool to refer to the model results. Clicking on the Output signal graphic will generate a fresh realization of the noise sequences n_1 and n_2, enabling the user to interactively experience variability in the estimates resulting from the stochastic nature of the disturbance.
- *Model validation*. As mentioned in the previous bullet, the magenta-colored vertical line of the Output signal graphic is interactively used to define the estimation and validation data sets. The validation data is used for cross-validation purposes. Model validation results are displayed in other three different graphics: Step responses, Correlation function of residuals, and Cross-correlation function between input and output. For all these graphics, the same color distribution noted before is used to represent the results of each model. The Step responses graphic, which is located at the upper left-hand side of the tool, shows the step responses for the each resulting model and a legend representing its goodness of fit in %. Confidence intervals can be also shown in this graphic activating this option from the Parameters menu. On the other hand, Correlation function of residuals and Cross-correlation function between input and output graphics, located between the Input signal and Power spectrum graphics, describe the auto- and cross-correlation between the input signal and the prediction-error for each model. By default, the input Autocorrelation graphic is shown instead of these two graphics. In order to switch between the input autocorrelation and residual analysis, two radio buttons are shown below the Input signal graphic that enable this commutation.

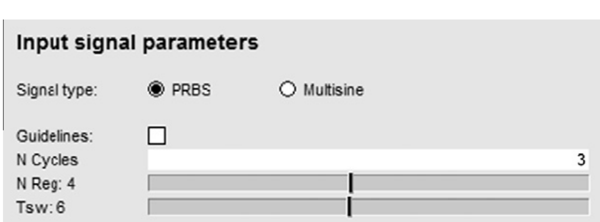

Fig. 3. *ITSIE* Input signal parameters example of PRBS signal.

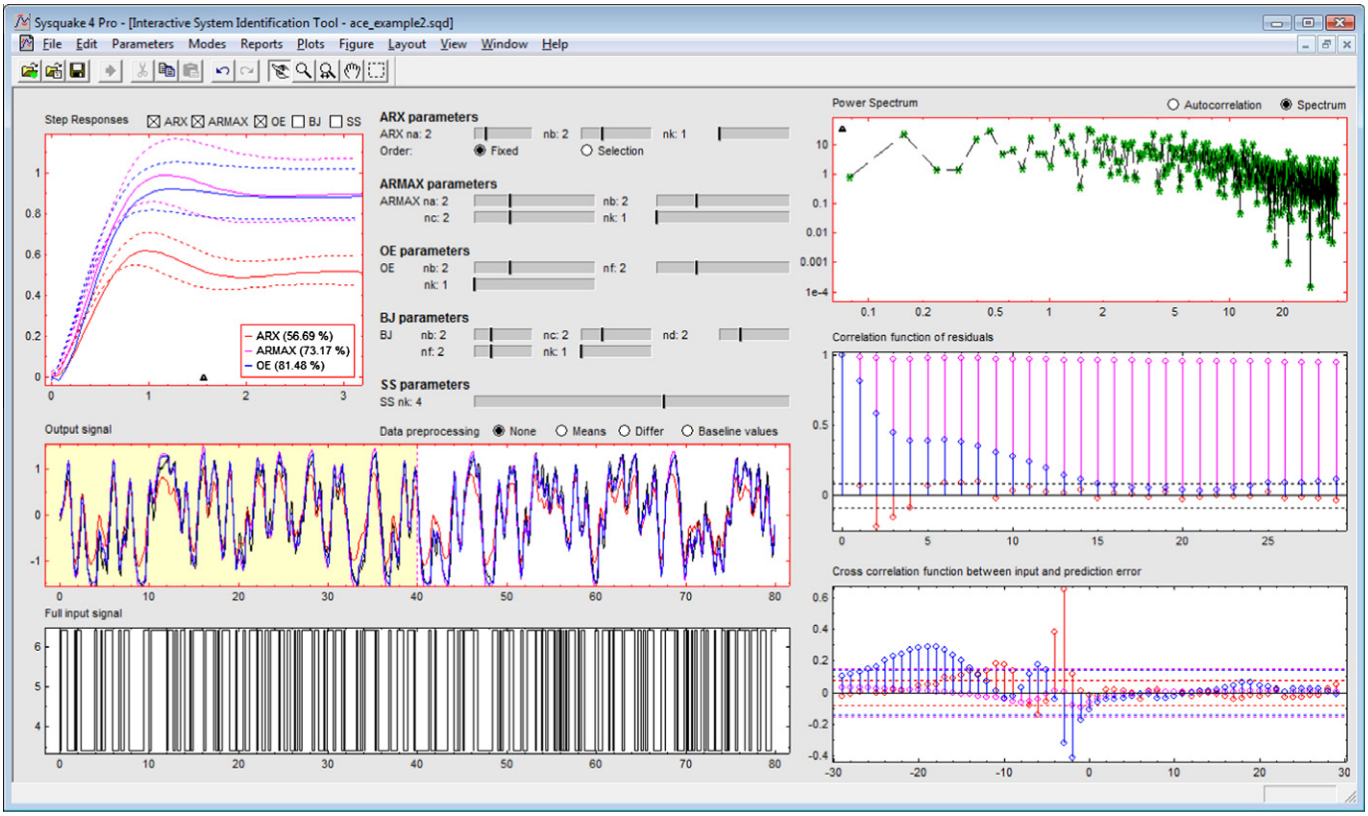

Fig. 4. Real mode of the *ITSIE* interactive tool evaluating external data corresponding to the system identification toobox's "hairdryer" data set in Matlab. Model estimates for ARX, ARMAX and OE estimation are shown along with residual analysis of the prediction errors.

3.2. Real data mode

This mode allows to load real data from the Mode → Real data menu. The real data can be loaded in ASCII and Matlab formats.[1] For ASCII format, the data must be organized in columns with the following order: time, output, and input signals. If the Matlab format is used, the file must contain three variables called "t", "y", and "u" for the time, the output, and the input, respectively. When real data is loaded, the tool screen is changed such as shown in Fig. 4. As it can be observed, those areas in the simulation mode dedicated to input design and plant definition and simulation parameters are changed. The *Model Structure Selection*, *Parameter Estimation* and *Model Validation* areas are exactly the same than in the simulation mode, but now working with real data loaded from file. In this mode, all the model parameters are always shown simultaneously on the right side of the Step responses graphic.

3.3. Additional options for education and training

The tool has been complemented with some additional options to be used for educational and training purposes. For instance, the teacher can define his/her own process model for the simulation mode using the Mode → Simulation menu, such as mentioned above. Once the the process model is defined, the teacher can export the model into a file and share it with the students. Notice that the model is hidden for the students. On the other hand, students and teachers can obtain detailed reports of the results from the Reports menu. The reports include information about the resulting identified models, e.g., goodness of fit, model structure, model parameters, and transfer functions in Matlab format.

[1] For the Matlab format, the data must be compatible with Matlab version 4. Use -V4 option with "save" Matlab command.

The interactive tool provides the user with multiple degrees of freedom for understanding the theoretical concepts and impact of choices made in the different steps of the system identification process. The main advantage with respect to other existing software tools is that the most important stages of system identification are shown simultaneously in one screen (input design, model structure selection, parameter estimation, and validation), and that the interactive features of the tool allow the user to understand and experience the relationships between these different stages, the meaning and effects of the associated parameters, and the bidirectional interpretation between parameter modifications from numerical (using sliders) and graphical (using interactive elements on the graphics) points of view.

In the following section, some of the many activities that can be shown by using this interactive tool are summarized, and an illustrative example is presented that highlights its usefulness for conveying a comprehensive picture of system identification.

4. An illustrative example

It has been noted that there are large numbers of possible scenarios with meaningful educational value that can be illustrated by the *ITSIE* tool. The list below is by no means exhaustive, but representative of some valuable concepts:

1. The importance of selecting cross-validation data, and how it impacts parameter estimation, particularly the effectiveness of automated order selection in ARX estimation.
2. A comparison between two different input signal types (i.e., PRBS versus multisines) and the usefulness of crest factor minimization for achieving "plant-friendliness" [17].
3. Understanding the issue of persistent excitation, as displayed in the interrelationship between input design and model order

selection. This is particularly useful when using the multisine input signal, given that the user can directly specify the number of nonzero harmonics in this signal.

4. The importance of taking advantage of *a priori* knowledge in input design. The time-constant guidelines presented in Section 2 can be thoroughly evaluated and appreciated.
5. The relative merits of various validation criteria, and the need to determine model adequacy based on more than one criterion. Correlation analysis on the residuals may indicate model adequacy, but simulation results on the model may show otherwise. The opposite can also be true: correlation analysis may indicate that there is still a need to refine on model structure; however, the model may still describe a large percentage of the output variance in the validation data and closely match the plant step response.

Figs. 5–7 illustrate the progression of an interesting educational example with the tool. The system considered is the simulated fifth-order system that is default in the tool, represented according to the transfer function:

$$p(s) = \frac{1}{(s+1)^5}, \quad T = 1\ \mathrm{s} \tag{13}$$

The main purpose of the example is to compare the application of Output Error and ARX modeling to this system under varying experimental conditions. The starting point for the example is illustrated in Fig. 5. Under conditions of significant noise, a relatively short experiment relying on two cycles of an arbitrarily specified PRBS signal with magnitude 1, shift registers $n_r = 4$ and switching time $T_{sw} = 3$ leads to erroneous models for both OE-[2 2 1] and ARX-[2 2 1] structures. Validation criteria that indicates the inadequacy of these models (in the absence of knowledge of the true system as provided in the step response) include the poor fits to both the estimation and validation data (both visually and in terms of variance captured by the model) and the wide discrepancy in the model step responses. Curiously, for this data set correlation analysis of the residuals for both model estimates falls within the standard error bounds, incorrectly implying model adequacy. Because of the short duration of the tests, the standard error bounds (determined by $\pm 1/\sqrt{N}$, where N is the length of the data set) are high, indicating to users that correlation analysis may be unreliable for short data sets under these experimental conditions.

Users are asked to determine what changes could be made to the identification procedure under these noise conditions without increasing input magnitude. One approach is to improve the design of the PRBS signal by relying on the time constant guidelines presented in Section 2. Fig. 6 illustrates the use of the guidelines in Section 2 for $\alpha_s = 2$, $\beta_s = 3$ and a dominant time constant range $(3 = \tau^L_{\text{dom}} \leqslant \tau_{\text{dom}} \leqslant \tau^H_{\text{dom}} = 5)$. The results of these guidelines are a recommendation to increase the number of shift registers and switching time to $n_r = 5$ and $T_{sw} = 4$, respectively, resulting in an increase in the PRBS cycle length N_s from 45 to 124. The results of this more sensibly designed signal improve the OE results substantially; however, the ARX-[2 2 1] model estimate remains inadequate. An explanation of this result is provided by the bias relationship shown in Eq. (12). Because ARX model estimation involves a trade-off between the fit to the noise model and the fit to the transfer function, the results under very noisy conditions with a restricted complexity structure will be prone to significant bias. Because the OE structure displays a fixed, unity noise model, it is able to achieve a consistent plant transfer function estimate regardless of the noise when the input and noise signals are uncorrelated; this property of the estimator can also be explained using Eq. (12). The fact that the ARX-[2 2 1] model bias is a systematic effect can be readily observed interactively by users by increasing

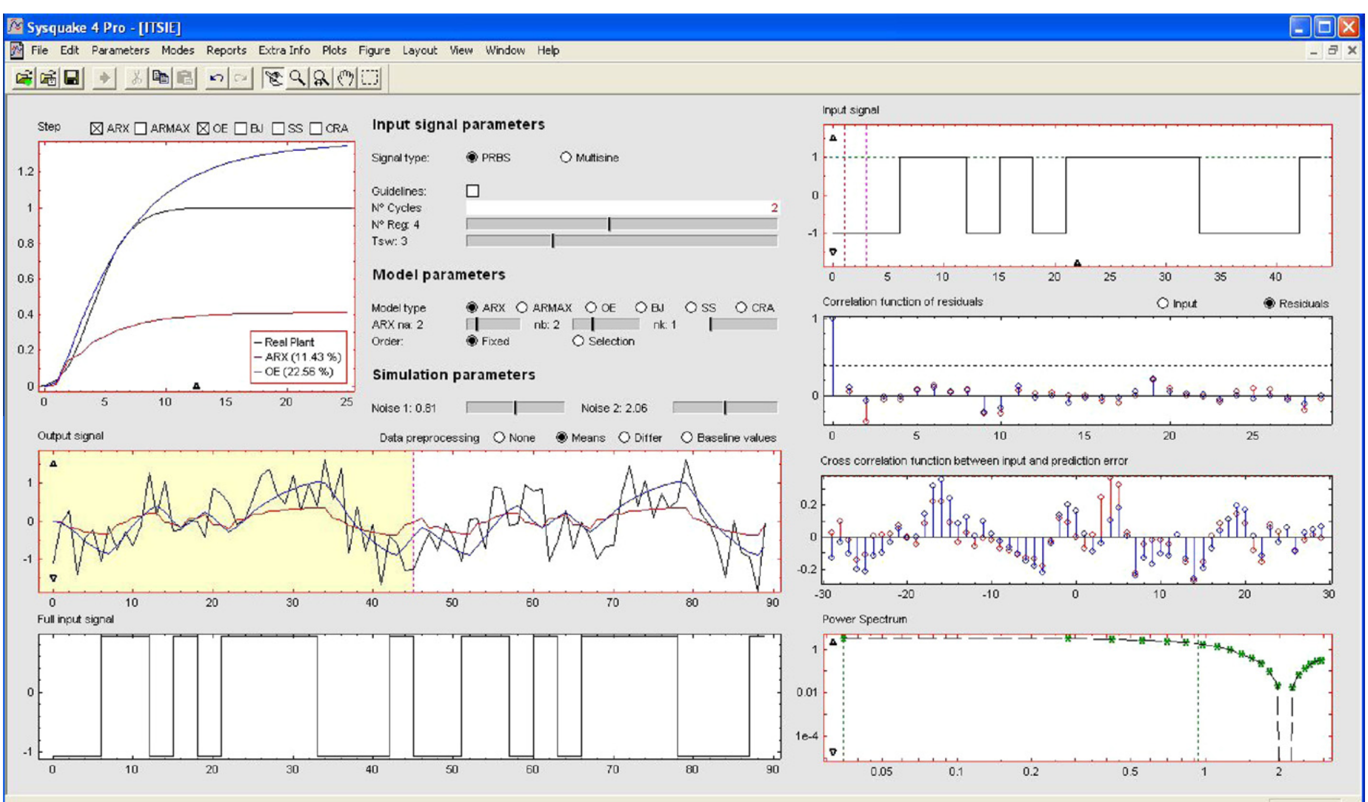

Fig. 5. *ITSIE* interactive tool user interface demonstrating two cycles of a PRBS input applied to a simulated fifth-order system. The input signal is designed arbitrarily. An OE-[2 2 1] model is compared with an ARX-[2 2 1] model for a short, noisy dataset.

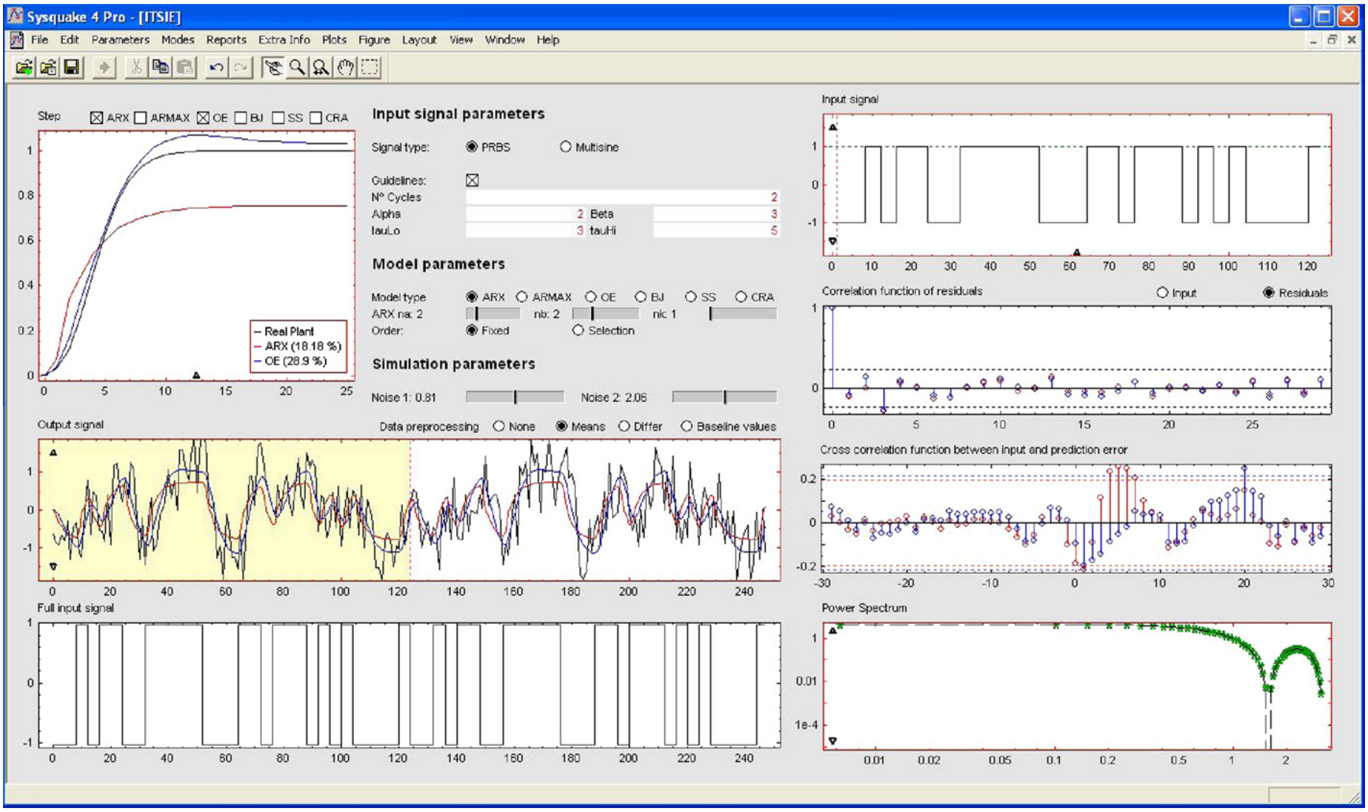

Fig. 6. *ITSIE* interactive tool user interface demonstrating two cycles of a PRBS input applied to a simulated fifth-order system. The time-constant guidelines from Section 2 are used to define input parameters. An OE-[2 2 1] model is compared with an ARX-[2 2 1] model.

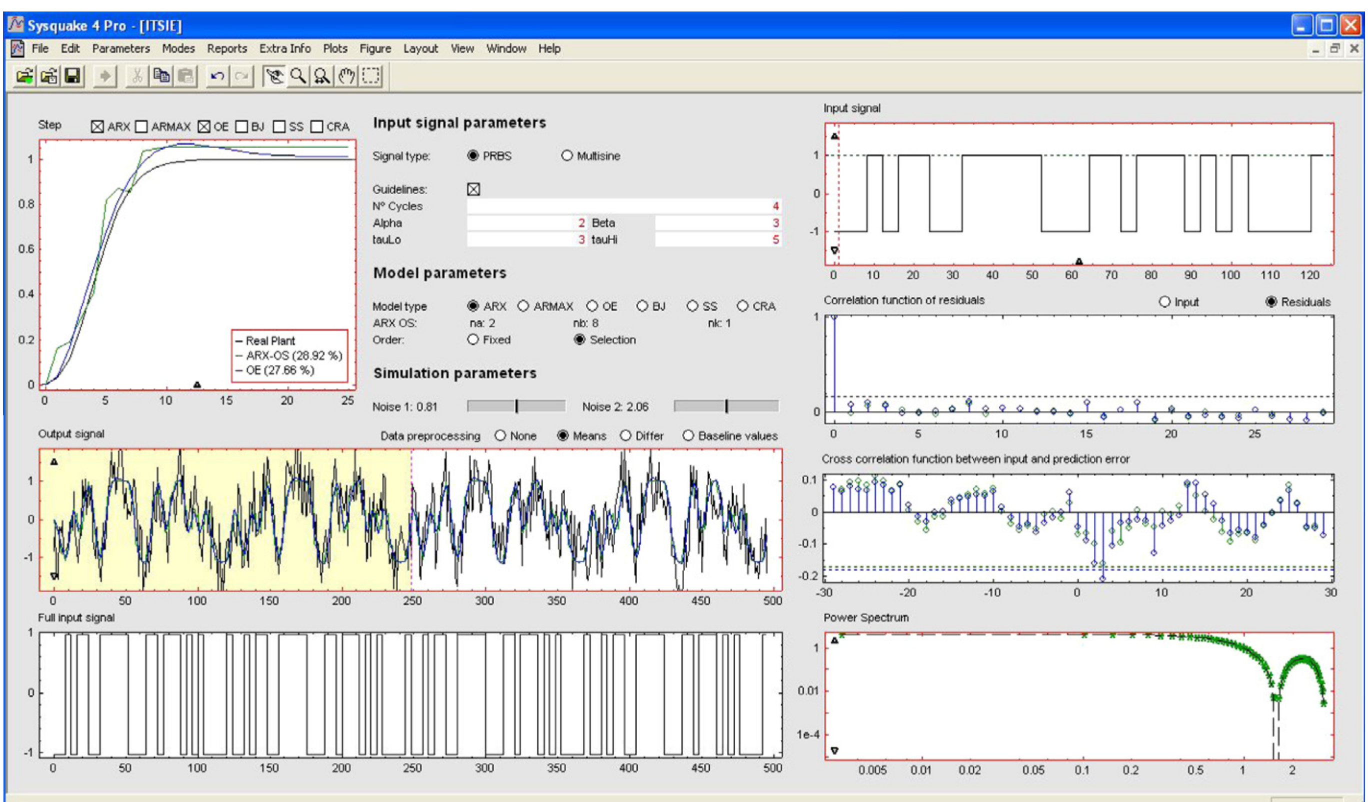

Fig. 7. *ITSIE* interactive tool user interface demonstrating four cycles of a PRBS input applied to a simulated fifth-order system. The time-constant guidelines from Section 2 are used to define input parameters. An OE-[2 2 1] model is compared with an ARX-[2 8 1] model obtained from exhaustive order selection on a cross-validation data set.

the number of PRBS input signal cycles substantially and witnessing that the model estimate and goodness-of-fit does not improve.

A final challenge can be presented to users: how can the results from ARX model estimation be improved to result in a satisfactory transfer function estimate? From consistency theory it is necessary to increase the order of the ARX structure, thus allowing the estimation procedure enough "room" to accommodate both the fit to transfer function and the fit to noise. The ARX order selection procedure described in Section 2 is ideal for this purpose, and requires the availability of good validation data. The price to be paid, however, is that a high-order model needs to be estimated, and therefore the variance of the estimated ARX model parameters is more pronounced than in the OE case. Additional input cycles and/or increasing the input magnitude, can reduce the variance. Fig. 7 shows a final result involving four cycles of the previously designed PRBS signal. By using order selection on a validation data set, an ARX-[2 8 1] model structure fits the plant well, matching the Output Error results. For subsequent tasks such as control design the ARX-[2 8 1] model represents a good precursor model, and can be further refined through model reduction or other means.

5. Pedagogical experiences

ITSIE has been applied in diverse educational settings that include both academic and industrial audiences. The first official use of *ITSIE* in the classroom was as part of a system identification short course taught at the University of Almería in September, 2008. It has also been used as part of ChE 494–598: introduction to system identification, a combined undergraduate-graduate level course taught at Arizona State University during the 2009 and 2011 spring semesters. Student response has been largely positive, and these experiences have provided input for further refinement and organization of the tool.

ITSIE is used to support the theoretical discussion in our classes, and it is the first tool used in the courses by the students. However, it is not the exclusive tool used in these courses, as students are also required to work and in some cases even code their own algorithms in Matlab. The student feedback before and after using *ITSIE* has been evaluated, and we have observed an impressive reduction in the learning curve thanks to the use of the interactive tool. This fact has been also observed by the authors' in other experiences with interactive tools used in the automatic control field, where significant improvements in learning outcomes were achieved by using interactive modules as support theoretical instruction [9–12].

6. Summary and conclusions

This paper has presented an interactive tool to perform the main stages of the system identification process. The tool provides different functionality modes that make it possible to use its capabilities for students and engineers with a small learning curve. The tool is for free and it can be downloaded from http://aer.ual.es/ITSIE/.

This paper has focused on the beginning of what we envision as a comprehensive family of novel interactive tools for system identification. Future tools will examine the interplay between input design, data prefiltering, and model structure on control-relevance, as well as tradeoffs in closed-loop identification and issues in multivariable system identification.

We invite the reader to download and access the tool, as only a hands-on evaluation can truly provide an appreciation for the benefits of an integrated solution to the identification problem, and the power of interactivity.

Acknowledgments

This work was supported by the Spanish CICYT and FEDER funds under Grants DIP2007-61068 and DPI2010-21589-C05-04, and by the Spanish Ministry of Education under the Project PHB2009-0008.

References

[1] Ljung L. System identification: theory for the user. 2nd ed. New Jersey: Prentice-Hall; 1999.
[2] Garnier H, Mensler M. CONTSID toolbox: a matlab toolbox for continuous-time system identification. In: Proceedings of the 12th IFAC symposium on system identification, Santa Barbara, USA; 2000.
[3] Ljung L. Educational aspects of identification software user interfaces. In: Proceedings of the 13th IFAC symposium on system identification, Rotterdam, The Netherlands; 2003. p. 1590–4.
[4] Ljung L. Version 6 of the system identification toolbox. In: Proceedings of the 13th IFAC symposium on system identification, Rotterdam, The Netherlands; 2003. p. 989–94.
[5] Nassirharand A, Karimib H, Dadfarnia M. A new software tool for synthesis of linear PID controllers. Adv Eng Software 2003;34:551–7.
[6] Dormido S. Control learning: present and future. Annu Rev Control 2004;28:115–36.
[7] Casini M, Prattichizzo D, Vicino A. The automatic control telelab: a web-based technology for distance learning. IEEE Control Syst Mag 2004;24:36–44.
[8] Nassirharand A. A new software tool for design of linear compensators. Adv Eng Software 2008;39:132–6.
[9] Dormido S, Dormido-Canto S, Dormido R, Sánchez J, Duro N. The role of interactivity in control learning. Int J Eng Educat 2005;21:1122–33.
[10] Guzmán JL, Berenguel M, Dormido S. Interactive teaching of constrained generalized predictive control. IEEE Control Syst Mag 2005;25:52–66. <http://aer.ual.es/siso-gpcit/>.
[11] Guzmán JL. Interactive control system design. PhD thesis, University of Almería, Spain; 2006.
[12] Guzmán J, Åström K, Dormido S, Hägglund T, Berenguel M, Piguet Y. Interactive learning modules for PID control. IEEE Control Syst Mag 2008;28:118–34.
[13] Guzmán JL, Rivera D, Dormido S, Berenguel M. ITSIE: an interactive software tool for system identification education. In: Proceedings of the 15th IFAC symposium on system identification, St. Malo, France; 2009. <http://aer.ual.es/ITSIE/>.
[14] Guzmán JL, Rivera D, Dormido S, Berenguel M. Teaching system identification through interactivity. In: Proceedings of the 8th IFAC symposium on advances in control education (ACE09), Kumamoto, Japan; 2009. <http://aer.ual.es/ITSIE/>.
[15] Braun M, Ortiz-Mojica R, Rivera D. Design of minimum crest factor multisinusoidal signals for plant-friendly identification of nonlinear process systems. Control Eng Pract 2002;3:301–13.
[16] Piguet Y. SysQuake 3 user manual, Calerga S'arl, Lausanne (Switzerland); 2004.
[17] Rivera D, Lee H, Braun M, Mittelmann H. Plant-friendly system identification: a challenge for the process industries. In: 13th IFAC symposium on system identification, Rotterdam, Netherlands; 2003. p. 917–22.
[18] Schroeder M. Synthesis of low-peak-factor signals and binary sequences with low autocorrelation. IEEE Trans Inform Theory 1970;IT-16:85–9.
[19] Guillaume P, Schoukens J, Pintelon R, Kollár I. Crest-factor minimization using nonlinear Chebyshev approximation methods. IEEE Trans Instrum Measur 1991;40:982–9.
[20] Rivera D, Gaikwad S. Systematic techniques for determining modelling requirements for SISO and MIMO feedback control. J Process Control 1995;5:213–24.
[21] Rivera D, Chen X, Bayard D. Experimental design for robust process control using schroeder-phased input signals. In: Proceedings of American control conference, San Francisco, CA; 1993. p. 895–9.
[22] Rivera D. Monitoring tools for PRBS testing in closed-loop system identification. AIChE annual meeting, Miami, FL; 1992. p. 1–24.