International Workshop on Advanced Control
IWAC2014
November, 3rd-4th 2014, Guelma, Algeria.

RIM: An Education Platform for Recursive Identification Methods

Aissa REBAI
Process Control Laboratory
Department of Control
National Polytechnic School of Algiers- Algeria
aissa.rebai@g.enp.edu.dz

Kamel GUESMI
CReSTIC,
Reims University,
France
kamel.guesmi@univ-reims.fr

Boualem HEMICI
Process Control Laboratory
Department of Control
National Polytechnic School of Algiers- Algeria
bhemici@yahoo.fr

***Abstract* - This paper presents an education platform for Recursive Identification Methods (RIM). The aim is to help on theoretical results validation and to carry out comparison between identification methods without passing by algorithms programming. Furthermore, this platform can be used to study the effect of identification parameters on model validity and results accuracy. An application example is given to validate the platform and to show their performances and capabilities.**

Index Terms* - *Education platform, Matlab GUI, System identification, Recursive methods, Adaptation gain.

I. INTRODUCTION

Description of physical systems behaviour can be done using mathematical models. These expressions can be established using analytical development and physical laws when it's possible. In the opposite case, the system behaviour can be approximated by the mathematical model giving the closest description to the physical system behaviour. We talk, in this case, about an identification process [1], [2].

In this context, there are several methods of identification and the most used are those called recursive methods. Indeed, they are very used because of their high performances and capabilities to approximate a large class of systems. To simplify the use of these techniques, several toolboxes and Guide User Interfaces GUI was implemented in Matlab environment [3]-[5]. However, these tools are not, in most cases, easy to use and need a minimum of information on Matlab environment and manipulations. Furthermore, these tools are relatively limited to specific methods and do not allow the comparison between the main techniques of identification. In this paper, we present RIM which is an identification platform that includes the main recursive methods. This platform is developed using Matlab GUI [6] and presents the advantage to be easy to use by beginners and students to validate the theoretical results without any need to learn about Matlab programming environment and manipulations.

The paper is organized as follows. In Section II, the main recursive identification methods are presented. Description of the RIM GUI is given in Section III and an application example is discussed in Section IV and the conclusion is given in the last section.

II. RECURSIVE IDENTIFICATION TECHNIQUES

The recursive identification techniques are divided into two categories; the first one is based on the whitening of the prediction error. The second is based on the uncorrelation of observation vector and prediction error [7], [8]. In this section, we give a survey on the most known techniques of these two categories.

For the first category, the four most known methods are the RLS, ELS, GLS and OEEPM.

A. Recursive Least Squares (RLS)

The selected model of this technique is ARX given by

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + e(t) \tag{1}$$

with

$$A(q^{-1}) = 1 + \sum_{i=1}^{n_A} a_i q^{-i}$$

$$B(q^{-1}) = \sum_{i=1}^{n_B} b_i q^{-i}$$

and $y(t)$, $u(t)$ are respectively the output and the input signals, $e(t)$ is a white noise with zero mean value and constant variance and d is pure time delay.

Hence, model (1) can be written as:

$$y(t+1) = \theta^T \varphi(t) + e(t+1) \tag{2}$$

where

$$\theta^T = \left[a_1, \dots, a_{n_A}, b_1, \dots, b_{n_B}\right]$$

$$\varphi^T = \left[-y(t), \dots, -y(t+1-n_A), u(t-d), \dots \right.$$
$$\left. \dots, u(t+1-d-n_B)\right]$$

Table I. Effect of $\lambda_1(t)$ and $\lambda_2(t)$

Adaptation gain	$\lambda_1(t)$ **and** $\lambda_2(t)$ **values**
Decreasing Gain	$\lambda_1(t)=1$ $\lambda_2(t)=1$
Constant Forgetting Factor	$\lambda_1(t)=\lambda_1, 0.95 \le \lambda_1 \le 0.99$ $\lambda_2(t)=1$
Variable Forgetting Factor	$\lambda_1(t)=\lambda_0\lambda_1(t-1)+1-\lambda_0$, $0.95 \le \lambda_0 \le 0.99$ $\lambda_2(t)=1$
Constant Gain	$\lambda_1(t)=1$ $\lambda_2(t)=0$

If $\hat{\theta}$ is the estimate of θ, then the estimated output $\hat{y}$ is given by:

$$\hat{y}(t+1)=\hat{\theta}^T(t)\varphi(t) \tag{3}$$

with

$$\hat{\theta}^T=\left[\hat{a}_1,...,\hat{a}_{n_A},\hat{b}_1,...,\hat{b}_{n_B}\right]$$

Minimizing the criterion defined by:

$$\min_{\hat{\theta}(t)} J(t)=\frac{1}{t}\sum_{i=1}^{t}\left[y(i)-\hat{\theta}^T(t)\varphi(i-1)\right]^2 \tag{4}$$

The estimated parameters vector θ is given by [8]:

$$\hat{\theta}(t+1)=\hat{\theta}(t)+F(t+1)\varphi(t)\varepsilon(t+1) \tag{5}$$

where

$$F(t+1)=F(t)-\frac{F(t)\varphi(t)\varphi^T(t)F(t)}{1+\varphi^T F(t)\varphi(t)} \tag{6}$$

and

$\theta(t)$: estimated parameters vector

$F(t)$: adaptation gain

$\varepsilon(t)$: prediction error

$\varphi(t)$: observation vector

The formula of the adaptation gain is generalized by introducing factors $\lambda_1(t)$, $\lambda_2(t)$ [9] and it is given by:

$$F(t+1)=\frac{1}{\lambda_1(t)}\left[F(t)-\frac{F(t)\varphi(t)\varphi^T(t)F(t)}{\frac{\lambda_1(t)}{\lambda_2(t)}+\varphi^T F(t)\varphi(t)}\right] \tag{7}$$

The effects of $\lambda_1(t)$ and $\lambda_2(t)$ on the adaptation gain are summarized in Table I.

The equations (5) and (7) are called parametric adaptation algorithm (PAA) and they are used for all recursive identification methods.

B. Extended Least Squares (ELS)

In this method, the model is ARMAX given by:

$$A(q^{-1})y(t)=q^{-d}B(q^{-1})u(t)+C(q^{-1})e(t) \tag{8}$$

with

$$\theta^T=\left[a_1,...,a_{n_A},b_1,...,b_{n_B},c_1,...,c_{n_C}\right]$$

$$\varphi^T=\left[-y(t),....,-y(t+1-n_A),u(t-d),...,u(t+1-d-n_B),e(t),....,e(t+1-n_C)\right]$$

We replace *e(t)* by *ε(t)* in the observation vector because *e(t)* is not observable. Finally, the PAA can be applied to estimate the parameters with:

$$\hat{\theta}^T=\left[\hat{a}_1,...,\hat{a}_{n_A},\hat{b}_1,...,\hat{b}_{n_B},\hat{c}_1,...,\hat{c}_{n_C}\right]$$

$$\varphi^T=\left[-y(t),....,-y(t+1-n_A),u(t-d),...,u(t+1-d-n_B),\varepsilon(t),....,\varepsilon(t+1-n_C)\right]$$

C. Generalized Least Squares (GLS)

This method is adapted for ARARX models as follows:

$$A(q^{-1})y(t)=q^{-d}B(q^{-1})u(t)+\frac{e(t)}{C(q^{-1})} \tag{9}$$

The parametric adaptation algorithm (PAA) is applied with:

$$\hat{\theta}^T=\left[\hat{a}_1,...,\hat{a}_{n_A},\hat{b}_1,...,\hat{b}_{n_B},\hat{c}_1,...,\hat{c}_{n_C}\right]$$

$$\varphi^T=\left[-y(t),....,-y(t+1-n_A),u(t-d),...,u(t+1-d-n_B),\alpha(t),....,\alpha(t+1-n_C)\right]$$

The estimated value of $\alpha(t)$ is given by:

$$\alpha(t)=\hat{A}(q^{-1})y(t)-q^{-d}\hat{B}(q^{-1})u(t) \tag{10}$$

D. Output Error with Extended Prediction Model (OEEPM)

The algorithm of this one is the same as the ELS with:

$$\hat{\theta}^T=\left[\hat{a}_1,...,\hat{a}_{n_A},\hat{b}_1,...,\hat{b}_{n_B},\hat{h}_1,...,\hat{h}_{n_C}\right]$$

$$\varphi^T=\left[-y(t),....,-y(t+1-n_A),u(t-d),...,u(t+1-d-n_B),\varepsilon(t),....,\varepsilon(t+1-n_C)\right]$$

$$\hat{c}_i=\hat{h}_i+\hat{a}_i$$

For the second category, the four main methods are the IVAM, OEFC, OEFO and OEAFO.

E. Instrumental Variable with Auxiliary Model (IVAM)

For this technique we have

$$\hat{\theta}^T=\left[\hat{a}_1,...,\hat{a}_{n_A},\hat{b}_1,...,\hat{b}_{n_B}\right]$$

$$\varphi^T=\left[-y_{IV}(t),....,-y_{IV}(t+1-n_A),u(t-d),......,u(t+1-d-n_B)\right]$$

where the prediction auxiliary model is defined by:

$$y_{IV}(t+1)=-\sum_{i=1}^{n_A}\hat{a}_i y_{IV}(t+1-i)+\sum_{i=1}^{n_B}\hat{b}_i u(t+1-d-i) \tag{11}$$

F. Output Error with Fixed Compensator (OEFC)

The disturbed output *y(t)* is replaced by the prediction $\hat{y}(t)$ in the RLS predictor and the PAA is used with:

$$\hat{\theta}^T = \left[\hat{a}_1,...,\hat{a}_{n_A},\hat{b}_1,...,\hat{b}_{n_B}\right]$$

$$\varphi^T = \left[-\hat{y}(t),....,-\hat{y}(t+1-n_A),u(t-d),...\right.$$
$$\left. ...,u(t+1-d-n_B)\right]$$

G. Output Error with Filtered Observations (OEFO)

This method is based on the same principle of the previous one with a filtered observation vector obtained by a stable filter $A_0(q^{-1})$. Thus,

$$\varepsilon(t+1) = y(t+1) - \hat{\theta}^T\varphi(t)$$

$$F(t+1) = F(t) - \frac{F(t)\varphi_f(t)\varphi_f^{\ T}(t)F(t)}{1+\varphi_f^{\ T}F(t)\varphi_f(t)}$$

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F(t+1)\varphi_f(t)\varepsilon(t+1)$$

with

$$\hat{\theta}^T = \left[\hat{a}_1,...,\hat{a}_{n_A},\hat{b}_1,...,\hat{b}_{n_B}\right]$$

$$\varphi^T = \left[-\hat{y}(t),....,-\hat{y}(t+1-n_A),u(t-d),...\right.$$
$$\left. ...,u(t+1-d-n_B)\right]$$

$$\varphi_f^T(t) = \frac{1}{A_0(q^{-1})}\varphi^T(t)$$

H. Output Error with Adaptive Filtered Observations (OEAFO)

In this method the observation vector is filtered by an estimated filter $\hat{A}(q^{-1})$. Hence,

$$\hat{\theta}^T = \left[\hat{a}_1,...,\hat{a}_{n_A},\hat{b}_1,...,\hat{b}_{n_B}\right]$$

$$\varphi^T = \left[-\hat{y}(t),....,-\hat{y}(t+1-n_A),u(t-d),...\right.$$
$$\left. ...,u(t+1-d-n_B)\right]$$

$$\varphi_f^T(t) = \frac{1}{\hat{A}(q^{-1})}\varphi^T(t)$$

For the validation the various identified models, the estimations of the normalized autocorrelations *RN(i)* are calculated. *RN(i)* is defined for the 1st category of identification methods as:

$$RN(i) = \frac{R(i)}{R(0)}; \qquad i = 1,2,....,\max(n_A, n_B + d) \qquad (12)$$

with

$$R(i) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)\varepsilon(t-i); \qquad R(0) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t) \qquad (13)$$

For the 2nd category we have:

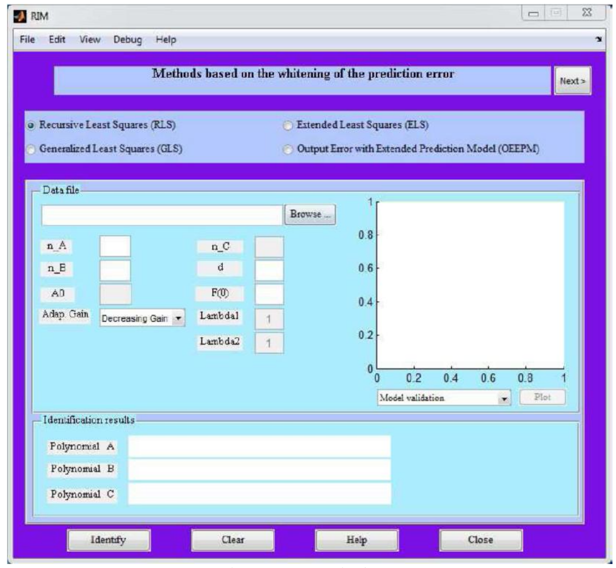


Fig. 1: RIM window

$$RN(i) = \frac{R(i)}{\sqrt{\left[\frac{1}{N}\sum_{t=1}^{N}\hat{y}^2(t)\right]\left[\frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t)\right]}} \qquad (14)$$

with

$$R(i) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)\hat{y}(t-i); \; i = 1,2,....,\max(n_A, n_B + d) \; (15)$$

with

ε(t) : output prediction error
y(t): system output
$\hat{y}(t)$: model output
N: data number

The identified model is valid if $|RN(i)| \le \frac{2.17}{\sqrt{N}}$. In practice, the used criterion is $|RN(i)| \le 0.15$ [8].

III. Description of the RIM platform

The proposed RIM GUI is developed using tools available in Matlab/GUI and all algorithms described in section II are programmed using Matlab environment in such way that the user do not need Matlab or any other programming language experience. To start the RIM platform, you have to enter into the MATLAB command line:

RIM

The RIM window appears on the screen as shown in Fig. 1. In this platform, the user feels free to use classical menu or buttons. Indeed, in the menu, we set the links of different buttons. Furthermore, RIM allows user to save or print the identification process results in several formats such as **.m,*.txt* and **.doc* using the option *Save as...* and *Print* in the menu. RIM allows also plotting the variables (identification error, error

Table II: Identification parameters

Parameter	Value
n_A	*2*
n_B	*2*
n_C	*1*
d	*0*
$F(0)$	1×10^5
A_0	*[1 0.3]*
λ_0	*0.97*
λ_1	*0.97*

Table III: Influence of adaptation gain on model validity

Adaptation gain	Model validity
Decreasing Gain	valid
Constant Forgetting Factor	not valid
Variable Forgetting Factor	valid
Constant Gain	not valid

autocorrelation, etc) in external figures. Furthermore, the RIM shows a message indicating if the model is valid or not based on the values of the error autocorrelation.

The user can also jump from identification category to the other by pressing the button *Next* or *Previous*. For all buttons, we set accelerators such as *CTRL+S* for the option *Save as...*, *CTRL+P* for *Print*, ... etc.

The use of RIM can be summarized on three steps and they are the main steps of any identification process. The first step is the generation of I/O file or data file. The second is the choice of the techniques to be used for system identification, and the last one is the selection of the identification parameters such as the polynomials orders n_A, n_B and n_C, the initial adaptation gain *F(0)*and the forgetting factor $\lambda_1(t)$.

IV. Application example

In order to illustrate the application and the performance of the RIM GUI, this section discusses the identification of a piezoelectric actuator which is a nonlinear system.

The data file (I/O file) used for the identification corresponds to the piezoelectric actuator APA-120ML excited by a sinusoidal signal of frequency *50Hz:* $u(t) = 68.5\sin(2\pi\times50t + 0.44) + 61.5$. The piezoelectric actuators are generally modeled by a second order system. Therefore the polynomials orders of the model are selected as follows: $n_A = 2, n_B = 2, n_C = 1$ and $d = 0$.

The identification parameters used in this paper are summarized in Table II. The identification results using the RLS method are shown in Fig. 2. From Fig. 2, the values of $|RN(i)|$ are all less than *0.15* which means that this model is valid and the system can be represented by the following ARX model:

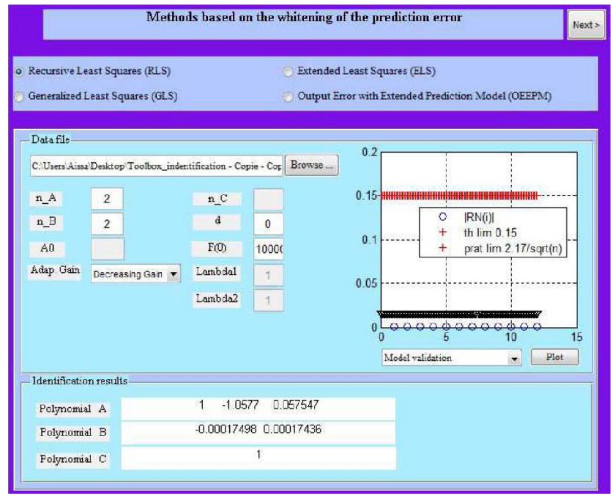


Fig. 2: Identification results using the RLS method

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) + b_1 u(t-1) + b_2 u(t-2) + e(t)$$

with

$$a_1 = -1.0577$$
$$a_2 = 0.0575$$
$$b_1 = -15.49\times10^{-5}$$
$$b_2 = 17.43\times10^{-5}$$

Based on RIM, more information can be illustrated (see Fig. 3a-3c) to conclude about the model validity.

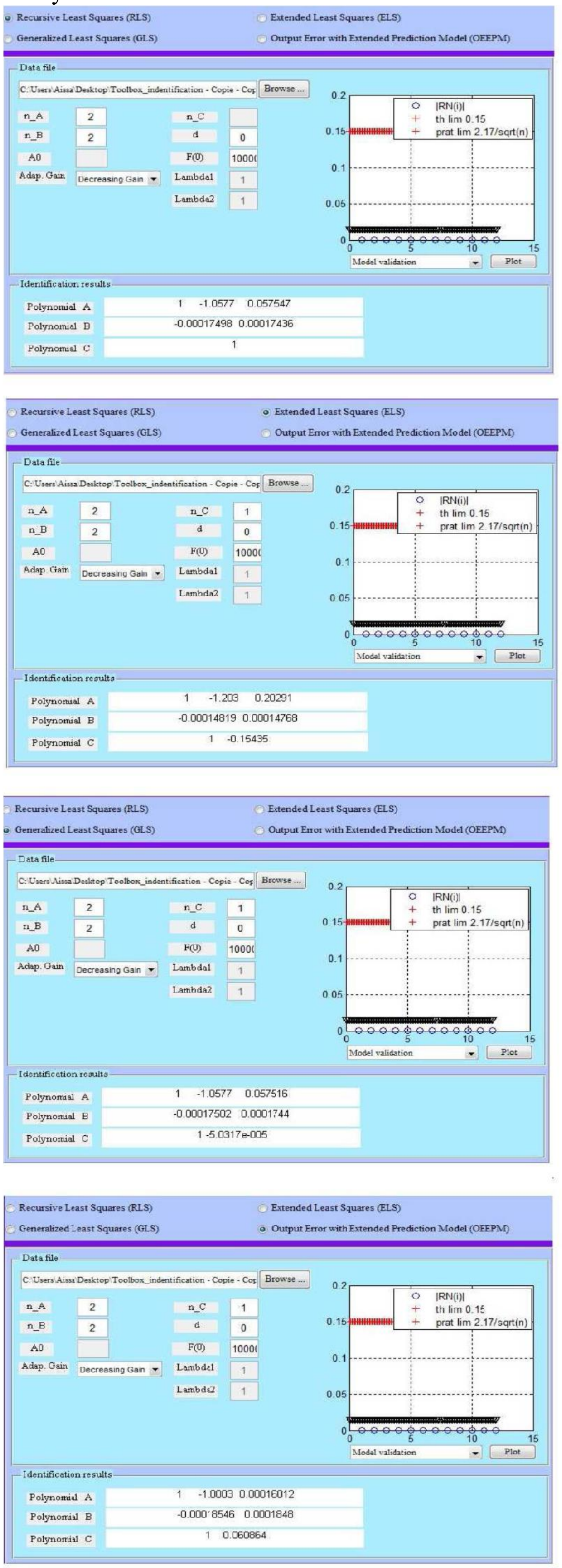
As mentioned before, one of the advantages of RIM is its capability to carry out a comparison between the various identification methods based on the error autocorrelation values. Indeed, the identification results for decreasing gain of all recursive methods are presented in Fig. 4a and Fig. 4b.

RIM also allows to the user to study the influence of identification parameters on the result. For example, in the case of decreasing gain, the model is valid, but with a constant gain, the model is not valid any more (see Fig. 5). The comparison results between the different cases presented in Table I using the parameters identification of Table II are illustrated in Table III.

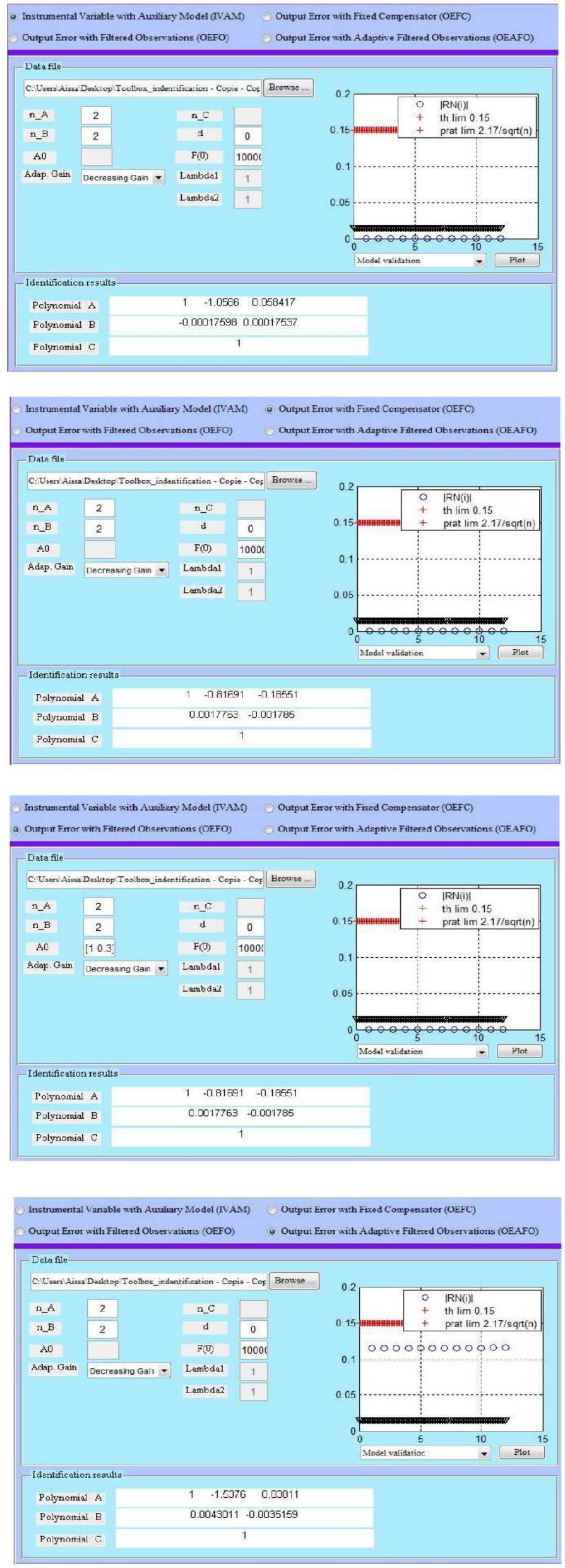
V. Conclusion

In this paper, we presented a Matlab GUI platform for education purposes. The proposed platform deals with the recursive identification methods and contains the necessary tools to validate many theoretical aspects in the domain of systems identification.

The RIM platform allows to users to make simulations without need to program the identification techniques or to have a programming experience in Matlab environment. RIM gives to users the possibility to study the influence of different identifications parameters and makes users more insight into the systems identification theory.

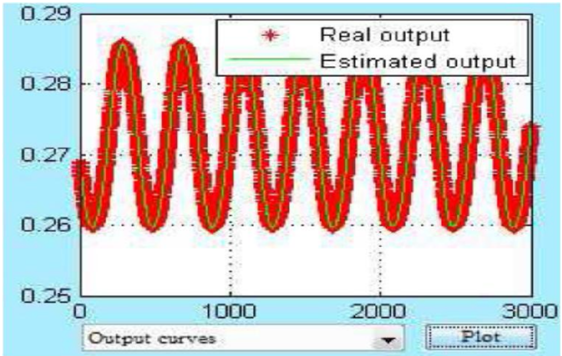


(a) Identification methods based on the whitening of the prediction error

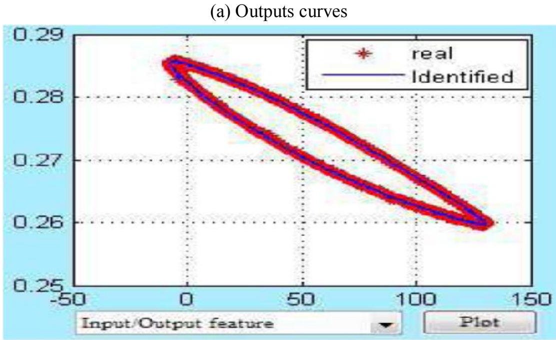


(b) Identification methods based on the uncorrelation of the observation vector and the prediction error

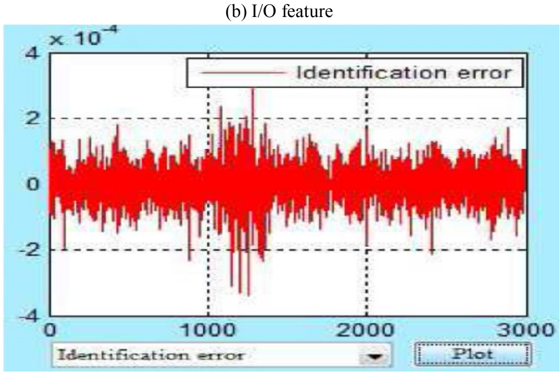
Fig. 4: Identification results with RIM



(a) Outputs curves



(b) I/O feature



(c) Identification error

Fig. 3: Figures which can be showed by RIM

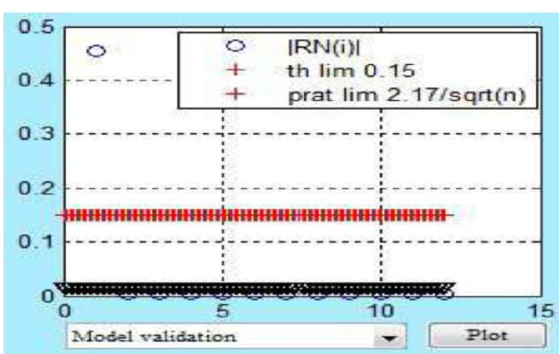


Fig. 5: Model validation for a constant gain

REFERENCES

[1] J. P. Corriou, *Process Control: Theory and Applications*. London: Springer-Verlag, 2004.

[2] C. Heij, A. Ran, and F. V. Schagen, *Introduction to Mathematical System theory: Linear systems, Identification and control*. Birkhäuser Verlag Basel - Boston - Berlin, 2007.

[3] B. Ninness, A. Wills, and A. Mills, "Unit: A freely available system identification toolbox," *Control Engineering Practice*, vol. 21, pp. 631 - 644, 2013.

[4] P. Navrátil and V. Bobál, "Recursive identification algorithms library, "in 17th *International Conference on Process Control*, Strbské Pleso, Slovakia, June 9-12 2009, pp. 516 – 523.

[5] L. Ljung, *MATLAB system identification toolbox users guide*, version 08 ed., The Mathworks, 2012.

[6] Mathworks, *MATLAB users guide*, 2013.

[7] I. D. Landau and G. Zito, *Digital Control Systems : Design , Identification and Implementation*. Springer, 2006.

[8] I. D. L. et al., *Adaptive Control*. Springer, 2012, ch. Recursive Plant Model Identification in Open Loop, pp. 153 – 191.

[9] M. S. Mahmoud and Y. Xia, *Applied Control Systems Design*. Springer, 2012, ch. System Identification Methods, pp. 35 – 148.