Developments in the MathWorks System Identification Toolbox

Lennart Ljung, Rajiv Singh, Qinghua Zhang, Peter Lindskog,
Anatoli Ioudiski

Division of Automatic Control

E-mail: ljung@isy.liu.se, rajiv@mathworks.com,
zhang@irisa.fr, lindskog@niradynamics.se,
Anatoli.Iouditski@inrialpes.fr

13th May 2009

Report no.: LiTH-ISY-R-2906

Accepted for publication in 15th IFAC Symposium on System Identification, Saint-Malo, France

Address:

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

WWW: <http://www.control.isy.liu.se>

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

Abstract

The paper describes additions to the System Identification ToolboxTM offered by The MathWorks, Inc, that handle the estimation of nonlinear models. Both structured grey-box models and general, flexible black-box models are covered. The idea is that the look and feel of the syntax, and the graphical user interface (GUI) should be as close as possible to the linear case. This presentation is focused on the GUI functionality and the possibilities to simulate the estimated models in SimulinkTM.

Keywords: System Identification

Developments in The MathWorks System Identification Toolbox

Lennart Ljung* Rajiv Singh† Qinghua Zhang**
Peter Lindskog*** Anatoli Iouditski****

* *Div. of Automatic Control, Linköping University, Sweden.*
ljung@isy.liu.se

** *INRIA-IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France.*
zhang@irisa.fr

*** *Nira Dynamics. peter.lindskog@niradynamics.se*

**** *LMC, Univ. Grenoble, France. Anatoli.Iouditski@inrialpes.fr*

† *The Mathworks Inc, Natick MA, USA, rajiv.singh@mathworks.com*

Abstract: The paper describes additions to the System Identification ToolboxTM offered by The MathWorks, Inc, that handle the estimation of nonlinear models. Both structured grey-box models and general, flexible black-box models are covered. The idea is that the look and feel of the syntax, and the graphical user interface (GUI) should be as close as possible to the linear case. This presentation is focused on the GUI functionality and the possibilities to simulate the estimated models in SimulinkTM.

Keywords: nonlinear system identification, neural networks, nonlinear models, graphical user interface (GUI)

1. INTRODUCTION

The current version of the MathWorks System Identification Toolbox (SITB), Ljung (2007) integrates techniques for nonlinear and linear models. A general introduction to nonlinear black-box models can be found in Sjöberg et al. (1995). The basic idea is to do this integration in a transparent manner, so that the rather complex problem of estimating and analyzing nonlinear models will appear simple and natural. It is therefore desirable that it can be done with a syntax which has the same look and feel as for linear models. The underlying theory and methodology for the model estimation were described in Ljung et al. (2006). This presentation focuses on the new Graphical Interface features and the possibilities of simulating the estimated nonlinear models in Simulink.

The nonlinear model structures supported by the current version of the toolbox are:

- **idnlgrey**: nonlinear grey-box models: arbitrary nonlinear state-space equations in continuous or discrete time with a user-specified parametric form. The user supplies code in the form of MATLAB m-file or MEX file that defines the right hand side of these state-space equations.
- **idnlarx**: nonlinear ARX models: system output is modeled as a nonlinear regression of past inputs and past outputs.
- **idnlhw**: nonlinear Hammerstein-Wiener models: block-oriented structure with a linear dynamic block preceded and/or followed by a nonlinear static block.

For **idnlarx** and **idnlhw** models, a single command can be used both to specify a model structure and to estimate the model from data. For example:

```
m = nlarx(data,[2 2 1],'sigmoidnet')  
m = nlwh(data,[2 2 1],'poly1d','wavenet')
```

where **'sigmoidnet'**, **'poly1d'** and **'wavenet'** indicate the types of nonlinearity estimators. These two commands are indeed similar to their linear model counterparts **arx** and **oe**.

For an **idnlgrey** model, the model structure must be specified in an M-file or MEX-file before being estimated from data.

Like the linear models of the SITB, each nonlinear model is implemented as a MATLAB object. A model thus has a certain number of properties, like all MATLAB objects. Once a model is created, its properties can be accessed by the commands **get/set** or using the standard dot syntax. Property/Value pairs can also be used in other methods. For example, when a nonlinear model object **m0** has been created, the estimation syntax is the same as for linear models:

```
m = pem(data,m0,P1,V1,...,Pn,Vn)
```

where **Pk,Vk** are optional Property/Value pairs, and **data** is an **iddata** object.

The quality of estimated models **m1,m2,...,mn** can also be evaluated similarly to the linear case by using the following commands:

```
compare(data,m1,m2,...,mn)  
resid(data,m1)  
sim(data,m1)  
predict(data,m1)
```

(Model simulation error comparison, residual analysis, simulation and prediction, respectively). The nonlinearities in an estimated model are basically plotted by

plot(m1)

These functionalities of the System Identification Toolbox software were described in more detail in Ljung et al. (2006). We shall now illustrate its Graphical User Interface (GUI) and simulation facilities that are available in the toolbox Version 7.2.1 (R2008b) and higher.

2. GRAPHICAL USER INTERFACE FOR ESTIMATING NONLINEAR MODELS: BASIC USE

Let us consider a magneto-rheological fluid damper (or MR damper for short), to illustrate the GUI of the SITB. MR dampers are semi-active control devices to reduce vibrations of various dynamic structures. The data used in the following illustrations have been collected from a laboratory system (see the acknowledgment note at the end of the paper about the origin of the experimental data). The velocity V and the damping force F are sampled at 200 Hz (the velocity is obtained by numerical differentiation of the over-sampled displacement measurement). See Wang et al. (2008) and Wang et al. (2007) for more detailed descriptions of the experimental system.

First create an `iddata` object z from the input data V and the output data F (two column vectors containing 3499 data samples), and split it into two subsets ze and zv for estimation and validation purposes.

```
load MRdamper
z = iddata(F,V,0.005);
ze = z(1:2000);
zv = z(2001:end);
```

Type `ident` at the MATLAB prompt to open the GUI, then import ze and zv into the GUI. Try some simple linear models of different orders and keep the best ones from these trials, for example, `arx241` (linear ARX model with orders $[2 \ 4 \ 1]$), `oe421` (OE model with orders $[4 \ 2 \ 1]$), and `pss3` (a third order state-space model) using the GUI. We open the Model Output view and find that the fits to validation data are not that impressive (ca 50 % at best). Note that now the **Estimate** pull down menu offers nonlinear models:

If we select **Nonlinear models...** a new window opens (Figure 1), which allows us to select between two types of model structures:

- Nonlinear ARX
- Hammerstein-Wiener

(Nonlinear grey-box models must be specified in an M-file or in a C-mex file and estimated from command line. Then they can be imported into the GUI to be analyzed in the same manner as for the other models.)

If you select Nonlinear ARX, then model orders (the number of lagged past inputs and outputs used for the prediction) should be specified:

Fig. 1. The basic nonlinear model dialog window

Then, click on the **Model Properties** tab to select the nonlinearity type from a pull down menu:

By selecting **Sigmoid Network** and then clicking **Estimate** a nonlinear model `nlarx1` is estimated and inserted into the model board.

Similarly, by selecting the **Hammerstein-Wiener** pull-down option, you can select among several different static nonlinearities for the input and output nonlinearities:

To specify the orders of the linear system, select the Linear Block tab:

By clicking **Estimate** the corresponding model, **nlhw1**, is estimated and inserted into the model board:

3. GRAPHICAL USER INTERFACE FOR ANALYZING NONLINEAR MODELS

Note that linear and nonlinear models are treated equally in the model board (there is a background color shading distinction between linear, nlarx and nlhw models). They can be compared with each other, for example by residual analysis or in terms of model outputs on validation data:

We see that the nonlinear models are significantly better at reproducing the validation output than the linear ones.

We also see in the System Identification Tool window that the **Model Views Nonlinear ARX** and **Hammerstein-Wiener** have now been activated. By selecting Nonlinear ARX we get a view of the model nonlinearity as a function of the regressors, that can be interactively chosen. The view can also be arbitrarily rotated:

If several NLARX models of the same orders have been estimated, the nonlinearities may be superimposed.

Similarly, the, Hammerstein-Wiener view shows input output nonlinearities, as well as a choice of several characteristics of the linear part:

4. ACCESSING ADVANCED OPTIONS IN THE GUI

Nonlinear model estimation is characterized by a whole lot of options that may affect the result in an essential manner. From a command line syntax it may be difficult to grasp and figure out which options are available. A good thing with a GUI is that it may offer various options in a visible way, at the same time as it suggests the suitable default choices.

When estimating NLARX models, it is essential to choose the regressors. It could be as easy as to select the number of suitable input and output delays in Figure 1. But it may also be a matter of deciding which of these regressors should enter the prediction function in a linear or nonlin-

ear manner, resp. By pushing the **Edit Regressors ...** button a new dialog window opens up:

Here the check-boxes allow you to choose which regressors should enter in a nonlinear way. The pull-down menu allows you to search for a suitable subset of regressors that should be nonlinear. The window also allows you to define “custom-made” regressors. By pushing the **create** button, you can create your own regressors:

More options are associated with the estimation process. By selecting the tab **Estimation** in the Nonlinear Models window shown in Figure 1, you can modify a variety of Algorithms Options, including the option for continuing iterations to improve results. This tab also gives a report of the progress of the estimation process.

5. USING SIMULINK FOR SIMULATING ESTIMATED MODELS

After you estimate a nonlinear model, you can simulate it by the `sim` command as mentioned in Section 1. The models can also be linearized for a given input signal or about a given operating point using functions such as `linapp` and `linearize`. These functions allow creation of linear approximation of nonlinear models for analyzing the local behavior of nonlinear models about given operating points, and use them for specific applications such as controller design.

Simulink provides a powerful and popular avenue for use of identified (linear and nonlinear) models. You can open the Simulink Library Browser by clicking Start in the MATLAB Desktop and then select the System Identification Toolbox library.

This library contains blocks for various model and data objects related to this product. The blocks for representing nonlinear models are called Nonlinear ARX Model, Hammerstein-Wiener Model and Nonlinear Grey-Box Model.

A simple connection of IDDATA source block with a Nonlinear ARX Model block and an IDDATA sink block allows simulation of the identified model in Simulink and export of the simulation results to Workspace in the form of a data object. Such a configuration is shown below

wherein the measured output signal from validation data `zv` is plotted together with the simulated response of the estimated model `nlarx1`. The model `nlarx1` was made available to Simulink model by exporting it from the GUI to the MATLAB Workspace. Double clicking on the Nonlinear ARX Model block launches its parameter dialog where the model's name and initial conditions for simulation can be specified:

A Hammerstein-Wiener model or a nonlinear grey-box model may be similarly brought into a Simulink model for simulation.

The ability to import identified models into Simulink opens up lots of application avenues for these models. Some of these are:

- (1) Simulation of identified model using a variety of signals and using the model as a component in a physics based model of a larger system (e.g., a power amplifier in a model for aircraft engine may be more easily created using identification techniques rather than electrical circuits).
- (2) Inspection of the model's internal structure to understand their dynamics. Right-click on model's block and choose 'Look under mask' option. One can clearly see how input signal is propagated for computation of model's output:

- (3) Subsystem simplification: A complicated system (such as many physical modeling based systems) which may be intractable for certain tasks may be replaced by its data based approximation. The data may be collected by simulating the model about a given operating condition and measuring the input/output signals for the subsystem. This data can then be used with System Identification Toolbox software to derive suitable linear or nonlinear models. The identified model may then be imported into the Simulink model as a proxy for the original subsystem.
- (4) Code generation and control design tasks: The MathWorks provides several products as add-ons to Simulink that facilitate specific engineering tasks such as parameter estimation, linearization and code generation. These tools allow you to adopt identified models for a variety of uses. For example, C code for the identified models may be generated for generic and embedded targets using MathWorks Real Time WorkshopTM (RTW) software. In Simulink, an identified model block can be linearized using Simulink Control DesignTM software and used with Control System ToolboxTM software for controller design.

6. CONCLUSIONS

The System Identification Toolbox software is consistently being enhanced and expanded. The latest major extension was with version 7, which included several tools to estimate nonlinear models. In this presentation we have focused on the graphical tools for both estimating, analysing and using nonlinear models.

The GUI for model estimation and analysis follows closely the GUI for estimating linear models. In fact, it is a useful feature that linear and nonlinear models can be handled and compared side by side. Validating models in terms of how well they can reproduce validation data (GUI checkbox **Model output**) is a powerful and intuitive way to evaluate if a nonlinear model feature indeed results in better models. The residuals (prediction errors) resulting from applying a model to validation data should ideally be independent (uncorrelated) with past inputs, regardless of the model being linear or nonlinear. The GUI checkbox **Model resids** is therefore also a very useful tool to evaluate and compare models.

The feature to create Simulink blocks from estimated linear and nonlinear models in a very simple manner opens up a wide area of MATLAB applications for models obtained in the System Identification Toolbox Software.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Akira Sano (Keio University, Japan) and Dr. Jiandong Wang (Peking University, China) for having generously shared the MR damper data used in the illustrative examples of this paper.

REFERENCES

- L. Ljung. *System Identification Toolbox for use with MATLAB. Version 7*. The MathWorks, Inc, Natick, MA, 7th edition, 2007.
- L. Ljung, Q. Zhang, P. Lindskog, A. Juditsky, and R. Singh. An integrated system identification toolbox for linear and non-linear models. In *Proc. 14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006.
- J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.Y. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, 31(12):1691–1724, 1995.
- J. Wang, A. Sano, T. Chen, and B. Huang. Blind hammerstein identification for mr damper modeling. In *Proc. 2007 American Control Conf.*, pages 2277–2282, New York City, USA, 2007.
- J. Wang, A. Sano, T. Chen, and B. Huang. Identification of hammerstein systems without explicit parameterization of nonlinearity. *International Journal of Control*, xx:in press, 2008. DOI: 10.1080/00207170802382376.

Avdelning, Institution Division, Department Division of Automatic Control Department of Electrical Engineering		Datum Date 2009-05-13	
Språk Language ☐ Svenska/Swedish ☒ Engelska/English ☐ _____		Rapporttyp Report category ☐ Licentiatavhandling ☐ Examensarbete ☐ C-uppsats ☐ D-uppsats ☒ Övrig rapport ☐ _____		ISBN _____ ISRN _____ Serietitel och serienummer ISSN Title of series, numbering 1400-3902	
URL för elektronisk version http://www.control.isy.liu.se		LiTH-ISY-R-2906			
Titel Developments in the MATHWORKS System Identification Toolbox Title					
Författare Lennart Ljung, Rajiv Singh, Qinghua Zhang, Peter Lindskog, Anatoli Ioudiski Author					
Sammanfattning Abstract <p>The paper describes additions to the System Identification ToolboxTM offered by The MathWorks, Inc, that handle the estimation of nonlinear models. Both structured grey-box models and general, flexible black-box models are covered. The idea is that the look and feel of the syntax, and the graphical user interface (GUI) should be as close as possible to the linear case. This presentation is focused on the GUI functionality and the possibilities to simulate the estimated models in SimulinkTM.</p>					
Nyckelord Keywords System Identification					