

## FREQUENCY DOMAIN SYSTEM IDENTIFICATION TOOLBOX FOR MATLAB: AUTOMATIC PROCESSING – FROM DATA TO MODELS

István Kollár<sup>\*</sup>, Rik Pintelon<sup>\*\*</sup>, Yves Rolain<sup>\*\*</sup>, Johan Schoukens<sup>\*\*</sup>, and Gyula Simon<sup>\*</sup>

<sup>\*</sup>*Budapest University of Technology and Economics  
Department of Measurement and Information Systems  
H-1521 Budapest, Magyar tudósok krt. 2. Hungary  
Fax: +36 1 463-4112, email:[kollar,simon]@mit.bme.hu*  
<sup>\*\*</sup>*Vrije Universiteit Brussel, Dienst ELEC, Belgium  
Pleinlaan 2, B-1050 Brussel, Belgium, fax: +32 2 629-2850  
Email: [Johan.Schoukens,Rik.Pintelon,Yves.Rolain]@vub.ac.be*

**Abstract:** The Frequency Domain System Identification Toolbox for MATLAB is an effective tool for the identification of linear dynamic system models from measured data. Since the use of advanced system identification methods often requires a lot of programming work, the attention can be deflected from the modelling issues. Therefore, a Graphical User Interface (GUI) was developed which allows the experimenter to visually follow and control the data processing and modelling steps.

However, for many users it is still desirable to obtain good models with as few decisions to be made as it is possible. Therefore, automatic processing steps have been added to the GUI. Identification can be done now in this toolbox with a minimum of user interactions in the graphics windows, and a reasonable model is returned, ready for control or physical analysis. *Copyright © 2003 IFAC*

**Keywords:** system identification, frequency domain, automatic data processing, model order selection, Fourier analysis, transfer function, graphical user interface, GUI, MATLAB.

### 1 INTRODUCTION

System identification is a very difficult task. It needs a lot of insight into the physics of the phenomenon under test, and into the modelling process. The noisy nature of measured signals demands appropriate procedures for experiment design, measurement, data preprocessing, modelling, parameter estimation and model validation. In system identification theoretical and numerical considerations need to be followed, and for many users this is already prohibitive in the related methods.

These aspects were clear for us even after the first release of the Frequency Domain System Identification Toolbox in 1994. We made several steps to narrow the gap between the available functions and the users, as described below.

### 2 BRINGING THE PROGRAMS CLOSER TO THE USERS

If we want to let people use a program, we must make it as simple to use as it is possible. There are different steps we can make.

## 2.1 Simple function calls with default values of arguments

The first obstacle for a user is the requirement to give all the possible options in each routine. In general purpose programs many options are provided to control the type of iteration, stop criteria, data reduction, handling of messages and informative plots, etc. The choice among these settings requires good knowledge of the theory, and significant experience. However, in many cases, reasonable choices can be made by utilising some properties of the data, or sacrificing a part of the speed of the program for reliable convergence. Therefore, many function arguments can be given default values, sometimes in dependence on the data.

## 2.2 Data structures

An eminent difficulty in the use of general purpose programs is the handling of several interdependent aspect of the data. A measurement does not only consist of the measured samples, but also of the sampling frequency, circumstances like periodic or random excitation, frequency contents, amplitude units,

names of measurement channels, date, etc. If these are all to be given, function calls will usually get badly organized, difficult to check and debug, and frightening for the user. The solution for this problem was the use of complex data types: *cells* and *structures*, later *objects*. These are since several years available in Matlab. All function heads were carefully redesigned to be able to use as simply as it was possible. Moreover, some function calls were made dependent on the object type in the arguments. However, despite the efforts, we regularly got questions concerning the use of individual functions. It was apparent that what we did was not enough for the users to forget programming, and focus at the modelling task.

## 2.3 Graphical User Interface

The most drastic step towards user-friendliness was the introduction of a Graphical User Interface (GUI). This has been described in previous papers (Kollár et al, 1999; Kollár et al, 2000). The main window of the GUI can be seen in Fig. 1.

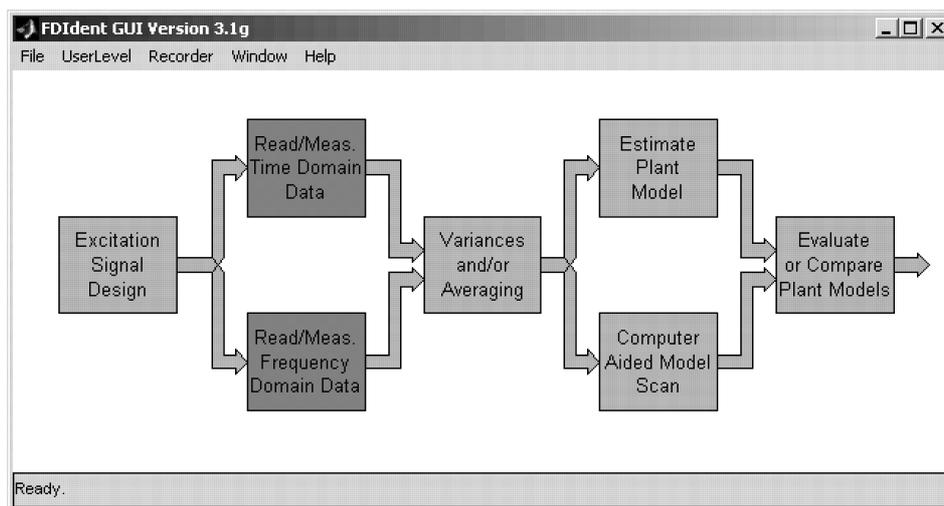


Fig. 1. Starting window of the Graphical User Interface

The basic idea is that opening any block, the user has access to processing tools (controls), and by proper selections the procedure can be directed to properly run.

Even in this intuitively clear tool, the users has several alternatives, that is, there are several decisions to make. This is sometimes still frustrating, and not always necessary. Therefore, already in the first design we made two decisions:

- Almost all parameters have default values, set properly for the given data. This not only allows the user to accept the computer-suggested settings, but also paves the way for automatic runs.
- Three different user levels allow to make a global choice among “Automatic”, “Interactive” and “Advanced” processing. Those who do not want to bother with many decisions can

run through the processing steps with default values.

## 3 NOVELTY NOW: AUTOMATIC PROCESSING

As we saw above, the direction of development is towards an easy-to-use tool. We try to let the computer program overtake as much as it is possible from the from the user. This proved to be the right direction, but we still had to pose several questions where the user needed to consider and answer. Therefore, the most logical step was to localize such places, and make the whole process as automatic as it is possible. The ultimate goal is to have a tool which absorbs the data, and returns verified results. Here are the points where we could automate, and overtake the task of decisions from the user.

### 3.1 Automatic Assembly of Data Structures (Data Objects)

Providing consistent and properly described data to an algorithm is a real pain. Despite of different helps and descriptions, creation of a proper full object was not easy for somebody who wanted to focus on identification rather than on technical details of data import. A set of data import windows made it possible to consistently generate objects from the descriptors of the data and of the experiment.

### 3.2 Reliable Preprocessing of Periodic Data

When measuring and processing periodic signals, there are two “catches”. One is that it can often happen that sampling and the signal frequencies are not coherent, despite of true efforts of the user; second is that it is often cumbersome to provide the set of frequencies of the signal components. An automatic tool has been developed which determines the period length, strong components, and variance of the measurement noise (Schoukens et al, 2002). This allows the user to simply plug the measured signals into the GUI, and watch how the estimated Fourier coefficients and variances come out.

### 3.3 Automatic Model Order Selection

Determining the proper model order is a very difficult task. Different criteria exist (Akaike, Rissanen, etc.), but they only work if nonlinearity errors are negligible compared to observation noise. Maybe the most popular solution is to scan the reasonable nu-

merator/denominator order combinations, and select on the basis of some user-defined criteria.

Analysis of the system equations in the proper mathematical basis allows to use a systematic approach: small singular values in the singular value decomposition tell how much overmodelled is the system, and careful “peeling” of the models allows to reach the proper minimum order model (Simon et al, 2000).

## 4 QUESTIONS STILL NEEDED TO BE POSED

The dream of a user is an intelligent machine which takes the data, and returns good results while the user drinks coffee. This sounds good, and *can almost be achieved*. In a typical “Automatic” session, the user needs to give only the following answers:

1) Bring the data into Matlab, and tell the GUI import window where are these. There are 2 possibilities: time domain measurements and frequency function measurements.

Time domain measurements (assuming periodic excitations):

- input and output time series
- sampling time
- antialias filters on/off (more precisely, whether lowpass filtered input/output measurements are made (bandpass experiment), or the input signal is computer generated, and the output is simply sampled (ZOH experiment))
- stationary/transient state

Optional parameters may also be given, but these are not obligatory.

Fig. 2 Composition of time domain data: the necessary fields (see above) are shown in the real life in red

Frequency function measurements, typically:

- FRF magnitude and phase, at which frequencies
- antialias filters on/off (more precisely, whether lowpass filtered input/output measurements are made (bandpass experiment), or the input signal is computer

generated, and the output is simply sampled (ZOH experiment))

Optional parameters may also be given, but these are not obligatory.

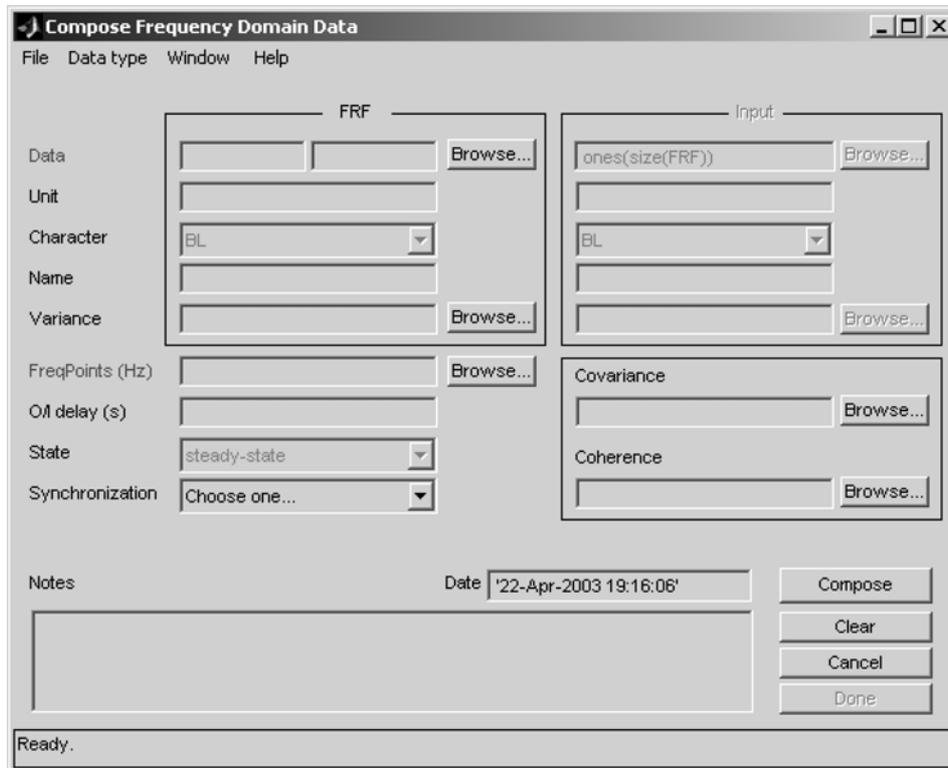


Fig. 3 Composition of frequency domain data: the necessary fields (see above) are shown in the real life in red

- 2) Choose between  $s$ -domain and  $z$ -domain (continuous-time or discrete-time models)

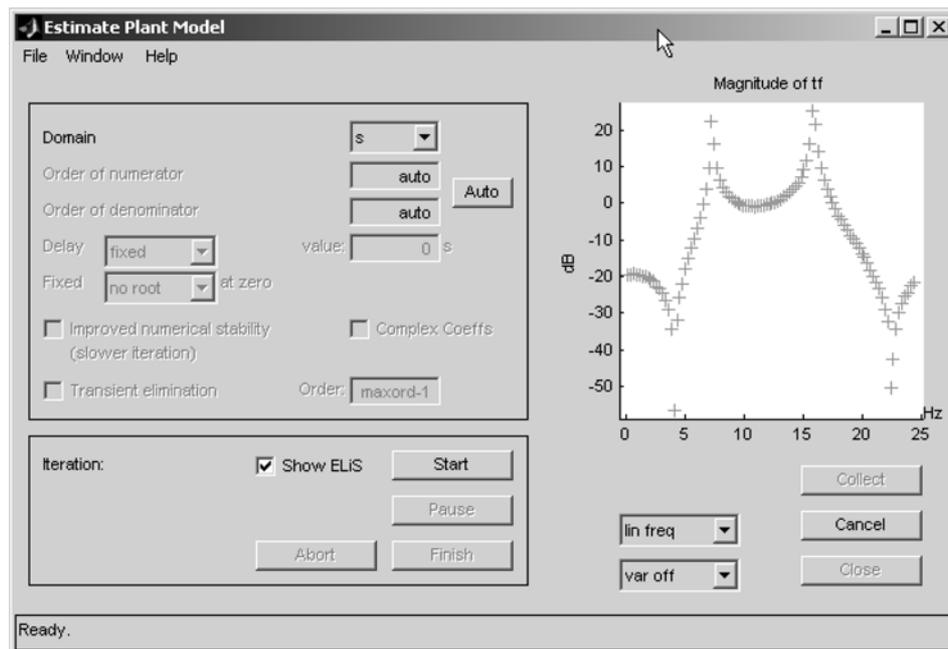


Fig. 4 Almost all controls are inactive in the “Estimate Plant Model” window: the user need only to select the domain, and press “Start”. The “auto” strings in the order windows mean that the best orders will be determined automatically.

The domain could also be fixed beforehand, but in many cases at least its choice is required, and this is not a complicated question to answer at all.

## 5. CONCLUSIONS

Automatic processing is already possible in identification, using careful program design and recently developed algorithms. An example for frequency domain identification is available in (FDIDENT, 1999-2003). The described automatisms allow the user to forget programming pains, and focus at identification itself.

## REFERENCES

- Balogh L., I. Kollár, and G. Gueret, "Variance of Fourier coefficients calculated from overlapped signal segments for system identification," *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE*, Vol. 1, pp. 1065-1070.
- FDIDENT (1999-2003), *Frequency Domain System Identification Toolbox Developers' Page*.  
<http://elec.vub.ac.be/fdident/>
- Kollár, I., R. Pintelon G. Román, G. Simon and J. Schoukens (1999), Graphical User Interface, Objects, and Improved Numerical Stability – New Developments in the Frequency Domain System Identification Toolbox. *Electronic publication*.  
<http://www.mit.bme.hu/~kollar/...>  
[...topics/fdidgui/fdident-paper.html](http://www.mit.bme.hu/~kollar/...topics/fdidgui/fdident-paper.html)
- Kollár, I., J. Schoukens, R. Pintelon, G. Simon and G. Román (2000), "Extension for the Frequency Domain System Identification Toolbox: Graphical User Interface, Objects, Improved Numerical Stability." *Preprints of the IFAC Symposium on System Identification, SYSID 2000, 21-23 June 2000, Santa Barbara, CA, USA. CD-ROM*.
- Pintelon, R., and J. Schoukens (2001). *System Identification - A Frequency Domain Approach*. IEEE Press, Piscataway, NJ.
- Schoukens, J. and R. Pintelon (1991). *Identification of Linear Systems: A Practical Guideline to Accurate Modeling*. Pergamon Press, Oxford.
- Schoukens, J., Y. Rolain, Simon G., and R. Pintelon (2002) "Fully automated spectral analysis of periodic signals," *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE*, Vol. 1, pp. 299-302.
- Simon, G., J. Schoukens, and Y. Rolain (2000), "Automatic Model Selection for Linear Time-Invariant Systems," *Proceedings of the 12th IFAC Symposium on System Identification, SYSID 2000, Santa Barbara, CA, USA, 21-23 June 2000, Vol. I.*, pp. 379-384.
- Extended electronic version: Simon, G., J. Schoukens, and Y. Rolain, "Automatic Model Selection for Linear Time-Invariant Systems – Practical Issues"  
<http://www.mit.bme.hu/~simon/...>  
[...publications/automodel.ps.zip](http://www.mit.bme.hu/~simon/...publications/automodel.ps.zip)

The Frequency Domain System Identification Toolbox for MATLAB is an effective tool for the identification of linear dynamic system models. Since the use of advanced system identification methods deflects the attention from the modelling issues, a Graphical User Interface (GUI) was developed which allows the experimenter to visually follow and control the data processing and modelling steps. However it is desirable to obtain good models with as few decisions to be made as it is possible. Therefore, automatic processing steps have been added to the GUI. Identification can be done now in this toolbox with just a minimum of user interactions in the graphics windows, and a reasonable model is returned, ready for control or physical analysis.