

Online Frequency Domain System Identification Based on a Virtual Instrument

József G. Németh, Balázs Vargha, and István Kollár, *Fellow, IEEE*

Abstract—System identification is an inherently iterative process. Yet, limited attempts have been made so far to implement the whole identification loop in a single device. This paper discusses the difficulties of the task and presents a solution based on a Matlab toolbox and a set of virtual instruments.

During the identification session, the obtained models may call for refinement or validation by new experiments. Using this integrated software-hardware tool, these subsequent experiments can be accomplished online with the identification procedure. As a demonstration, the modeling of a hairdryer is described.

The universal applicability of this solution is believed to be guaranteed by the modular architecture of virtual instrumentation and the general definition of the software interface developed. The interface allows combining Matlab-based identification packages with virtual instruments or pure hardware interfaced to Matlab.

Index Terms—Matlab, online recursive measurement, SigLab, system identification, virtual instrument.

I. INTRODUCTION

SYSTEM identification is used in various fields of engineering to determine the model that best corresponds to our observations about a system. A parametric model extracts the important aspects of the dynamic behavior of the system under test. Linear time invariant systems (LTIs) are often applied as models for their relative simplicity and well-established theory. In fact, many systems are almost linear within their range of operation.

The procedure of identification of linear systems is often quite involved. It comprises the design of the excitation signal, the measurement of the excitation and the response and the estimation and validation of parametric models. Finding the best solution implies several iterations through these steps that can also include repeated measurements.

The iterative nature of the system identification task [1] has already been recognized by developers in instrumentation who have made attempts to implement *the whole identification loop in one device*. Examples are the swept-sine based analyzer HP3563A of Hewlett Packard, the 2600 Series of Tektronix and DSP Technologies Siglab using the recursive lattice time-domain technique. However, these instruments use a

single method for the identification and do not make use of the rich variety of techniques to validate models, etc [1]. They are built as stand-alone tools with limited facilities for extensions.

This paper discusses the reasons for this and presents a solution. For practical reasons, we consider here a study case based on a frequency-domain system identification package and a commercially available instrument. Yet, we would like to stress that this solution is applicable

- to time-domain system identification; and
- to different instruments or devices as well.

II. DIFFICULTIES OF ONLINE IDENTIFICATION

There are several questions and problems that need to be tackled before developing an online identification tool.

First, the elaborated frequency-domain methods [1], [2] are currently nonrecursive, i.e., they assume the availability of the whole data record needed to perform the necessary calculations. This implies that the measurement and the parameter estimation stages are inherently separated in time. From another point of view, physical system modeling usually does not require quick reactions. Hence, no direct need arises for a real-time identification tool.

Second, *specifying the requirements* for such an instrument is also cumbersome. Calculations for s-domain modeling are quite involved, especially for orders over 10–15. Thus, depending on the task, the requirements can vary in a wide range. An additional difficulty in creating an integrated tool is that the hardware is almost always nonstandard. Even if it is commercially available, so in principle technical data are available, a lot of different data acquisition boards/equipment are in use. Consequently, it is extremely difficult to write standardized interface code that provides seamless data transfer from any hardware to a standard identification package.

Third, even if the interface is given to handle several different kinds of hardware, the *user interface* is expected to hide the irrelevant particularities of each type of hardware and allow the logical setting of parameters.

Finally, many users do not know about the details of the measurement setup necessary for effective and precise system identification (e.g., time-domain methods inherently assume that piecewise constant excitation is used, and the system response is measured without using an anti-alias filter [3]–[5], whereas frequency-domain system identification in the s-domain requires anti-alias filtered measurement of both the input and the output signals.) It is also often overlooked that periodic excitations and measurement of full periods in steady-state can significantly improve the accuracy of the result [6]. Therefore,

Manuscript received May 26, 1999; revised September 6, 2000. This work was supported by the Belgian National Fund for Scientific Research, the Flemish Government under Grants GOA-IMMI and BIL99/18, and the Belgian Government as a part of the Belgian Programme on Interuniversity Poles of Attraction (IUAP4/2) initiated by the Belgian State, Prime Minister's Office, Science Policy Programming.

The authors are with the Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Hungary, H-1521 (e-mail: nemeth@mit.bme.hu; vargha@mit.bme.hu; kollar@mit.bme.hu).

Publisher Item Identifier S 0018-9456(00)10520-0.

the interface of the instrument should make the best choices natural for the user and guide their activity in the right direction.

The solution for the above problems presented here is a *virtual instrument*, i.e., an instrument composed of layers of software and hardware having a virtual control panel that only appears on a computer display.

III. BASIS FOR AN INTEGRATED VIRTUAL INSTRUMENT

Virtual instrumentation has its strength in the modular and scalable architecture based on software and hardware building blocks. The widespread use of portable PCs makes this line of instrumentation even more appealing.

Our software platform for an integrated virtual instrument was Matlab, as it is well suited to the task of matrix computations exceedingly used in identification.

Its *Frequency-Domain System Identification Toolbox* [2] is a universal off-line tool that contains sophisticated methods and services. The graphical interface of the toolbox [9] offers a route through each step of the identification loop [1], except for accomplishing the online measurement. Data can be exported from and imported into the toolbox.

Meanwhile, another Matlab-based tool, *SigLab* (DSP Technologies) [7], is a collection of virtual instruments based on a common data acquisition hardware.

The hardware comprises 18-b sigma delta converters for high-precision measurements in the audio range and three DSPs for real-time computing and communication. It is accessible via SCSI bus from a host computer. This high-speed bus allows using more SigLabs together.

SigLab provides virtual instruments for both time-domain and frequency-domain analysis. These address successfully the problem of making multi-channel experiments with real-time preprocessing (averaging, FFT, statistical estimates) but offer limited services as far as data processing and analysis are concerned.

The great advantage of SigLab compared to other virtual instruments is that the user can manipulate the measurement data directly in Matlab without the overhead of transferring and converting data between programs. Together with the Frequency-Domain System Identification Toolbox, they provide a flexible environment for measurement and identification. The idea to combine these two tools into an integrated virtual instrument followed naturally. In order to do so, we needed to build an intelligent interface that handles some previously mentioned difficulties of online tools.

IV. ON-LINE IDENTIFICATION TOOL

A. Modular Structure of the Online Tool

Fig. 1 shows the modular structure of the integrated measurement-identification tool. The shaded ovals represent the existing components, while the nonshaded ones represent possible extensions. The ovals labeled as Interface GUI (Graphical User Interface) and V.I. (Virtual Instrument) Description represent the components of the newly developed interface.

The uppermost layer is related to the design and analysis of experiments; let us call it the identification process. The Frequency-Domain Identification package (Freq.D.Ident in Fig. 1) has been complemented by a GUI for launching measurements (*Interface GUI*). This component is a window that opens up from the identification package displaying the excitation signal and the list of available measuring instruments (after having checked the hardware) and asks for the necessary experiment parameters from the user. This component is independent from the measuring instrument and communicates with the middle layer through high-level calls. These calls are defined as a standard interface.

In the *V.I. Description* component of the interface, these high-level calls are broken down into procedures. The description of the hardware is contained here.

Thanks to the modular design, at each layer the components can be technically interchanged for a different tool of the same functionality. Beyond the implementation issues, the definition of the interface is general enough to support the use of different hardware or a different identification package.

B. Technical Details of the Interface

The role of the interface program is to

- return information on the virtual instruments;
- execute actions with the instruments;
- help the user to adjust the parameter settings for these actions so that the parameters agree with hardware capabilities (e.g., automatic adjustments can be carried out during the design of the excitation signal in the identification package.).

The *V.I. description* contains routines to implement these services. The *Interface GUI* handles the settings at a logical (abstract) level for the user with comfortable services such as a channel calibration facility, user help structured in windows, etc.

The interface files are available by e-mail for anyone, who uses Siglab and the Frequency-Domain Identification Toolbox for Matlab. The interface can be used together with the Extension of the Frequency-Domain Identification Toolbox [8]. Data elements of the *V.I. description* are explained in the header comments of the M-file [9], and a few of them are listed below:

- List of available devices, associated virtual instruments and predefined measurement scenarios, with used data types and excitation sources
- Allowed values for: excitation clock frequency, excitation record length, analysis sampling frequency, amplitude peak, excitation record length
- Options for: reconstruction and anti-alias filters

Allowed values can be given in analytical or explicit form in the interface, as they are encapsulated in calls.

The technical data about the hardware as well as the properties of the virtual control panel are mostly collected dynamically in the interface code, so that the modifications in subsequent versions of the virtual instrument do not necessarily lead to inconsistencies.

Some difficulties we encountered stemmed from the fact that SigLab supports, first of all, nonparametric identification. It does not calculate variances of obtained Fourier amplitudes

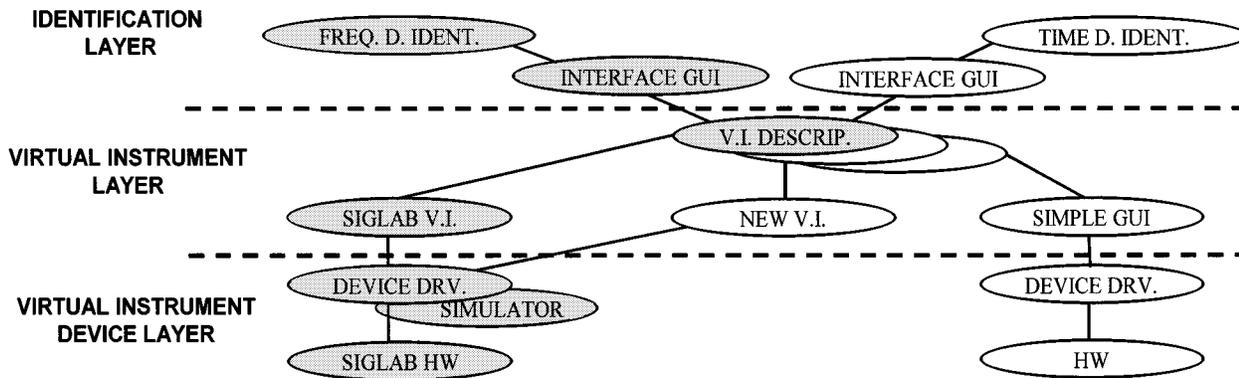


Fig. 1. Modular structure of the integrated measurement-identification tool and possible extensions.

and in some cases, it even returns coherence values over 1 because of 32-b floating-point processing. It is arguable how well variances can be approximated from coherence values (even supposing that the system is linear). Erroneous coherence values close to one can completely corrupt the parameter estimation, because these correspond to noiseless measured values allowing no freedom at the model fitting.

As a consequence, we had to add some new controls to the existing virtual instruments in order to allow easy execution of repeated measurements. All these extensions are also made dynamically without making any modifications to the original code of SigLab.

C. Extensions to the Implemented Tool

As an extension to the presented tool, a student has developed a hardware simulator program (Fig. 1). This is an M-file overriding the DLL device driver file. The simulator is useful mainly for the multi-site development of new V.I.s for the same hardware (Fig. 1) but might be used for education purposes as well: the teacher could check the students' performance on simulated identification problems.

Although SigLab has a high-precision data acquisition device, the audio bandwidth can be a serious limitation. One of our future goals is to build an automated impedance analyzer tool, which gives a more complex model than a simple parallel or serial model. A real capacitor, for instance, can be modeled by four parameters: serial inductance and resistance of the lead wires, capacitance and leakage resistance. In order to get enough information about all the poles and zeros we need a bandwidth of some megahertz. In our design we can integrate with reasonable effort an appropriate hardware device supporting this bandwidth (Fig. 1).

V. EXPERIMENTAL RESULTS

The presented online solution works. We have executed measurements that validate the abilities of the whole environment.

We used a hair-dryer setup (Fig. 2) as the prototype model for an industrial process. The input voltage controls the firing angle of a triac. The output power from that controls the hairdryer heating. The temperature of the air flood is measured with a temperature sensor. This signal is measured back as output voltage of the overall system.



Fig. 2. Block-model of the setup.

This setup is a popular example in control because, despite its relative simplicity, it contains nonlinearity and a pure time delay, which makes it difficult to identify the few parameters or to control the output.

The modulation of the power by the triac introduces a nonlinearity in the system. The transfer function of the system in a linear operating range can be described as

$$\frac{V_{OUT}}{V_{IN}} = e^{-s\tau_D} \frac{K_{TRIAC} \cdot K_1 \cdot K_2}{1 + s(\tau_1 + \tau_2) + s^2\tau_1\tau_2} \quad (1)$$

where K_1 , K_2 , τ_1 , τ_2 are the dc gains and time constants of the hairdryer and the sensor, respectively, while τ_D is the time delay mainly introduced by the air flood. K_{TRIAC} is a scaling factor introduced by the triac.

Our primary concern was to measure the nonlinearity and the frequency response function simultaneously by recently developed methods. Yet, we made some experiments to identify parametric models, too.

We compare four models fitted on the data: 0/2-(0 order numerator, 2nd order denominator) and 2/2-order models, with and without pure time delay.

For the 0/2-order model, the pure time delay was easily detectable because the attainable cost function value (Cf) without allowing the delay was much higher. The amount of delay was estimated to be about 20 ms.¹ 5 ms could be attributed to the average delay introduced by the firing angle of the triac.

With the 2/2 model, however, the Cf is even lower and gets close to the theoretical minimum even without time delay. This means that the delay is hardly detectable. The reason is that with such a small delay the exponential term in the transfer function can be very closely approximated by a second-order polynomial introduced in the numerator. By allowing a delay the ELiS algorithm [1] estimated 15 ms for the 2/2 case.

The two real poles of the system were around -0.5 and -1.6 , respectively, in all the above cases.

¹This was very low compared to the delay reported with similar setups and can probably be explained by the higher velocity of the air flood.

The *a priori* model [0/2 system in (1)] and the experiment results (2/2 system) do not quite agree. Clearly, we need to make further experiments to evaluate our models by cross-validating them with data coming from independent measurements, etc.

These and numerous other measurements have been carried out during a relatively short stay abroad at Vrije Universiteit Brussel. The integrated online identification tool—running on a portable PC—proved to be extremely useful to save time and to avoid data handling mistakes.

VI. CONCLUSION

- The difficulties of online identification were discussed, and a modular architecture was proposed.
- The implemented integrated online tool works and is open to extensions, which are under development.
- The specification of the developed interface (available at [8]) is general enough to support other kinds of identification packages (e.g., the time-domain System Identification Toolbox of Matlab).
- It allows users of different hardware to build their own module into it.
- The usefulness of this integrated environment has been tested and demonstrated in a real situation.

REFERENCES

- [1] J. Schoukens and R. Pintelon, *Identification of Linear Systems: A Practical Guide to Accurate Modeling*. New York: Pergamon, 1991.
- [2] I. Kollár, *Frequency Domain System Identification Toolbox*. Natick, MA: The MathWorks, 1994.
- [3] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*. Reading, MA: Addison-Wesley, 1990.
- [4] L. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [5] J. Schoukens, R. Pintelon, and H. Van hamme, "Identification of linear dynamic systems using piecewise constant excitations: Use, misuse and alternatives," *Automatica*, vol. 30, no. 7, pp. 1153–1169, 1994.

- [6] K. Godfrey, *Perturbation Signals for System Identification*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [7] *SigLab 2.0 Manual*, DSPT Inc., Fremont, CA, 1995.
- [8] J. G. Németh and I. Kollár, "Comment lines of the interface," <http://www.mit.bme.hu/~nemeth/Vicomment/>.
- [9] I. Kollár *et al.*, "Graphical user interface, objects, and improved numerical stability—New developments in the frequency domain system identification toolbox," <http://www.mit.bme.hu/~kollar/topics/fdidgui/ch13mod.html>.

József G. Németh was born in Budapest, Hungary, in 1973. He received the M.S. degree in electrical engineering in 1998 from the Technical University of Budapest, where he is currently pursuing the Ph.D. degree in the Department of Measurement and Information Systems. His main research interests are in frequency-domain system identification and DSP applications.

Balázs Vargha was born in Budapest, Hungary, in 1975. He received the M.S. degree in electrical engineering in 1998, from the Technical University of Budapest, where he is currently pursuing the the Ph.D. degree in the Department of Measurement and Information Systems. His main research interests are digital-to-analog converters and virtual instrumentation.

István Kollár (M'87–SM'93–F'97) was born in Budapest, Hungary, in 1954. He received the M.S. degree in electrical engineering from the Technical University of Budapest in 1977, and the "Candidate of Sciences" and "Doctor of the Academy" degrees from the Hungarian Academy of Sciences in 1985 and 1998, respectively.

From 1989 to 1990, he was a Visiting Scientist at the Vrije Universiteit Brussel, Belgium. From 1993 to 1995, he was a Fulbright Scholar and Visiting Associate Professor in the Department of Electrical Engineering, Stanford University. Currently, he is Professor of electrical engineering at the Budapest University of Technology and Economics. His research interests span the areas of digital and analog signal processing, measurement theory, and system identification. He has published about 70 scientific papers, and is coauthor of the book *Technology of Electrical Measurements* (New York: Wiley, 1993). was Editor of *Periodica Polytechnica Series in Electrical Engineering* from 1988 to 1991. He was also Chief Editor of *Periodica Polytechnica* from 1991 to 1997.

Dr. Kollár is an active member of EUPAS.