10.2478/acsc-2013-0026

Archives of Control Sciences
Volume 23(LIX), 2013
No. 4, *pages 427–446*

MULTI-EDIP – an intelligent software package for computer-aided multivariate signal and system identification

JERZY KASPRZYK and JAROSŁAW FIGWER

In the paper an intelligent software package MULTI-EDIP for computer-aided identification of multivariate signals and systems is presented. Purposes and main requirements for computer-aided identification are discussed. A summary of the most important MULTI-EDIP services with a focus on expert advice is described. An example of using the package in electro-acoustic plant identification for active noise control system development is presented.

Key words: computer-aided system identification, parametric model identification, non-parametric model identification, digital signal processing

1. Introduction

System and signal identification (SSI) may be looked upon as a generalized measurement technique, providing users with hardware and software necessary to transform raw measurement data into mathematical models representing, in a comprehensive form, the essential features of system or signal behaviour. It constitutes a unique blend of mathematical methods, manifold programming techniques and practical experience. The basic ingredients of this blend are the following:

- knowledge of the real-world process to be identified;
- knowledge of time- and frequency-domain models for dynamic systems and signals;
- knowledge of estimation procedures;
- knowledge of numerical methods;
- knowledge of some advanced programming language;

The Authors are with Institute of Automatic Control, The Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland. E-mails: Jerzy.Kasprzyk@polsl.pl, Jaroslaw.Figwer@polsl.pl
Received 14.12.2012. Revised 13.11.2013.

- knowledge of software techniques like data bases and computer graphics;
- knowledge of process simulation techniques;
- experience in real-world and computer simulated model identification;
- identification know-how as mastered by identification experts.

It is generally accepted that some form of computer support is a necessary prerequisite for successful system and signal identification, as the result of identification depends on a considerable extent upon the availability of a user-friendly software package giving access to robust and high-quality numerical procedures, some data base tools and some tools for producing graphical output. Therefore some classical software packages ([14], [29]), intelligent software packages called expert systems ([13], [15], [34]) and MATLAB like toolboxes ([2], [6], [7], [8], [12], [22], [24], [25], [33]) were developed to aid the user in system and signal identification. This paper is devoted to one of such software packages called MULTI-EDIP.

MULTI-EDIP is the latest chain in a long development project of a tool for computer-aided SSI. It started in 1980's at the Institute of Automatic Control, The Silesian University of Technology, under name EFPI (Expert for Process Identification) [30], [31]. EFPI was designated for identification of single-input single-output (SISO) systems and scalar signals. It has been continued during the years 1994-97 as MULTI-EDIP for DOS operating system [32] and is being pursued up to now as MULTI-EDIP for Windows [18], [20]. MULTI-EDIP is a user friendly intelligent software package supporting identification, analysis and simulation of mathematical models for multi-input multi-output (MIMO) systems and multivariate signals in time- and frequency-domain. It is destined for advanced as well as non-skilled users. In the paper its most important properties and services are described.

The paper is organized as follows: (1) selected problems of software development for computer-aided identification are discussed; (2) intelligent services of MULTI-EDIP are presented; (3) details on modes of working with MULTI-EDIP are given; (4) an example of expert identification of a MIMO electro-acoustic plant for active noise control system design is described; (5) final conclusions are drawn.

2. Computer-aided identification

In computer-aided system and signal identification (CASSI) cooperate interactively a person performing identification procedure and a computer executing strictly defined tasks during data processing. Complexity of the software package for CASSI depends on how general model structures are treated, which identification methods are implemented and how intelligently the user can be lead through meanders of identification. However, the problem of whether SSI can be automated and to what extent is still unsettled. Bohlin in [5] discusses prospects and pitfalls of interactive SSI that may well justify some fundamental questions related to the design of CASSI tools. Generally, it is known that for successful SSI some special and hard to get expertise is necessary, an expertise which is elusive and difficult to define but nevertheless very real. Thus, it is obvious to postulate that computer support for SSI should go beyond standard number crunching and data base services (as made available, *e.g.*, by appropriate SSI toolboxes) providing with services that may be expected from an identification expert.

2.1. Purposes and tool requirements

SSI is an iterative procedure consisting of the following steps: data acquisition from identification or simulation experiments, preprocessing of collected data, estimation of a mathematical model using preprocessed data, validation of the estimated model. Hence, the user expects that CASSI gives him assistance in carrying out these steps together with a suggestion what and how should be done. Basic purposes of CASSI tool can be then summarized as to support SSI by:

- guiding the user through all actions to be performed during identification procedure;
- providing data-base capabilities to store and retrieve data from identification or simulation experiments as well as the corresponding results of their processing;
- providing with high-quality numerical procedures and the most important identification algorithms integrated with graphical presentation for easy and quick employment;
- offering support in choosing and parameterizing appropriate numerical procedures and identification methods;
- offering the possibility to test identification methods and to learn identification techniques through identifying simulated, well-defined systems and signals;
- recollecting basic facts from identification theory through a context-sensitive *help*.

However, more advanced CASSI tool should also offer some kind of expert advice in solving such problems, like, *e.g.*:

- testing system or signal stationarity;
- choosing identification methods;
- choosing a model structure;
- testing if estimation task is well conditioned as it influences reliability of the solution;
- validating the model structure and estimated parameters.

If the tasks presented above (or other, similar to them) are implemented in CASSI tool, it tends to be qualified as intelligent. The word intelligent appears often in a control- and identification-related context, without being precisely defined. The default definition implied by most of what is being published seems to be that some control or identification algorithm or software is regarded as intelligent if it is based upon (or contains) paradigms which somewhat resemble human inference paradigms [35]. Thus, a piece of software using recursive least squares is hardly ever thought to be intelligent, but if it is extended by a front-end containing a rule-based knowledge base which extends the software scope so as to allow, *e.g.*, for automatic model order determination, one somehow succumbs to a temptation of calling it by a more impressive name. Some other paradigms, provided they seem to be remotely related to processes presumably going on in living systems (let say fuzzy logic, neural processing, genetic algorithms), are also used as excuses for calling a particular control or identification contraption intelligent. Nevertheless, it seems that all those using the word intelligent in a control- or identification-related context are referring rather to something to be above standard. Therefore we could safely in what follows confine ourselves to this meaning of intelligent and accept the following rather subjective definition of intelligence: a performance is considered intelligent, if experts comparing it with what is standard in their domain of expertise, consider it as such. Thus, if CASSI tool fulfills requirements presented above, it can be qualified as intelligent.

It is also generally assumed that CASSI software has to be a *user-friendly* tool, taking advantage of facilities supplied by Windows environment and this aspect is presented in the next subsection.

2.2. User interface

Very important part of CASSI software is the user interface. It should allow a novice to communicate with CASSI tool without the need of studying manuals and extensive identification expertise, especially that nowadays many users start their adventure with identification techniques with the aid of a software, and without familiarization with any textbooks on the topic considered. According to this, the user interface will also play a role of tutor in teaching principles of system and signal identification. Because SSI is a complicated task, combined with many possible purposes and aspects, hence a list of different methods, algorithms, model structures or designing parameters is uncommonly long, and for a novice it can be quite overwhelming. Thus, software for CASSI should be designed so that the user could be conscious what options are available and simultaneously not feel lost at the beginning. Later, he would be oriented in the different possibilities of CASSI software. A direct solution consists in showing only these options that are essential at the current stage of the identification procedure, whereas hiding those that are not necessary, or about which the user does not want to know. Particularly, this concerns all the parameters that are independent from data or chosen by the designer in a heuristic way.

This can be done using graphical user interface (GUI) that enables the user to communicate with CASSI software via different kinds of windows (menus, plots, reports) together with pop-up menus and hints, which are automatically switched on in accordance with the current state of the identification session. They should guide user through complicated meanders of identification procedure, *i.e.*, through identification experiment design, data preprocessing, models supported by CASSI software, the corresponding estimation methods up to model validation tests.

2.3. Simplicity versus flexibility

The basic problem in designing a tool for SSI is how to fit its features to skills and knowledge of the user. A novice expects to be guided through the system and signal identification step by step, whereas the experienced user would prefer rather to search his own way to solve the problem. In the first case, the structure of the CASSI software should be fixed, guiding the user through all numerical and graphical functions implemented in the system. It is assumed that in this case user knowledge about SSI is rather poor, so the amount of pieces of information delivered to him should be adequate for understanding what is going on and helping to take decisions. On the each step of identification procedure, the user meets only with options relevant to this step and he can only choose one of them. The number of parameters needed to declare is limited to an indispensable minimum and all actions that can be automated should be implemented in CASSI tool. However, to assure minimum flexibility, the user has the right to make decisions and he has permission to change or to veto the expert advice given by the system. Thus, working with CASSI software is similar to usage of an intelligent microprocessor instrument. It is the case where the user interface is oriented on the identification procedure and a classical example of such a tool is EFPI [30].

The approach mentioned above is safe and convenient for users inexperienced in SSI, as it enables them to benefit from achievements of theory of identification without the necessity of deeply studying the literature. However, for somebody that has some experience in this field, work with such CASSI software can be somewhat tiresome. It claims moving through the same paths and answering the same questions. Moreover, this solution is hardly flexible because it does not allow for testing new, dedicated procedures. Thus, on the opposite side, the advanced user would search CASSI software having an adequately rich set of functions that enables him to process data and present the outcomes in a free style, like, *e.g.*, in classical MATLAB or Octave. Usage of these systems rely upon programming algorithms of data processing, corresponding to the chosen identification method. Users have at their disposal sets of functions aggregated in toolboxes. In MATLAB there are, *e.g.*, the most famous System Identification Toolbox [24], Frequency Domain System Identification Toolbox [22], CONTSID [12] and others. In this approach the user has nearly unlimited capabilities of creating his individual procedures, but he has to know the syntax of applied language, as well as the list of implemented functions together with possible parameters, apart from a good knowledge of identification theory. Thus, the user should be not only an expert in his domain, but also in system and signal identification. However, new versions of toolboxes often offer some advice as well. Nevertheless, for users not being academic researchers, especially far from automatics or technical sciences, benefiting from these toolboxes may be too difficult.

Solution proposed in MULTI-EDIP is based on a compromise between simplicity and flexibility. It means that all numerical and graphical tasks are implemented in the system and there is no necessity (and capability) to self-reliant programming as it is stated below. On the other hand, the user can run identification procedure freely.

3. Presentation of MULTI-EDIP design approach

The sheer volume of MULTI-EDIP basic services clearly points at the need of creating a framework to host them. This framework has been developed first on a small-scale basis for EFPI. The basic assumptions formulated and implemented in EFPI for SISO systems and scalar signals, have been extended in MULTI-EDIP for MIMO systems and multivariate signals.

3.1. Basic prerequisites

It was assumed that designed CASSI tool will be meant for students learning system and signal identification as well as for researchers and engineers from different areas of interest, working on model building based on data coming from real-world or simulation experiments. Namely, a target group of MULTI-EDIP represents people whose knowledge and skills in SSI are not broad, but who are interested in identification of the most popular linear models, described in well known textbooks on system identification, like [23], [38] or [37]. Therefore, essential promises can be verbalized as follows:

1. System/signal identification and model analysis/validation should be done on the highest possible level of abstraction, *i.e.*, on the symbolic level. Therefore, the user is dealing only with basic concepts of SSI, as for example: input-output models, time series models, model classes (polynomial, spectral), types and structures, estimation procedures, model validation and model checking tests. He/she is not dealing with bits, bytes, instructions, syntax, semantics, algorithms, *etc*. MULTI-EDIP frees the user from doing any computer programming by providing full control of all functions and services through a system of windows and pull-down menus. The main motivation for this assumption came from the exigencies of teaching system and signal identification: students performing laboratory identification experiments should work on exactly the same conceptual level as the one used in exposing them to the main ideas of SSI. Roughly speaking, the basic philosophy of MULTI-EDIP is antiMATLAB.

2. The main techniques used to accomplish the user support are:

- combining the model to be identified with the most appropriate identification method (sec. 4.3);
- offering expert advice in model structure selection (sec. 3.2);
- suggesting a set of default values for some important parameters;
- checking correctness and reasonableness of parameters declared by the user;
- proposing a set of validation procedures;
- fixing, if the model has to be estimated as time-varying or not;
- a hypertext context sensitive help system, providing comments, suggestions and explanation of what has been obtained or what should be done next.

Besides, a simulation mode allows the user for testing any identification method first on a set of simulated data derived from a model of precisely defined structure with precisely known parameters. This service is crucial for building user confidence in CASSI software and preparing them to use it effectively.

Of course, the presented concept involves an important trade-off between user friendliness and tool flexibility. The features built into MULTI-EDIP enable less knowledgeable users to obtain satisfactory results and understand them. On the other hand, the most advanced users cannot create their own algorithms of data processing, modify implemented methods or identify some unusual models as they can using MATLAB identification toolboxes. Nevertheless, the package can be useful also for this kind of user as it offers identification and analysis of typical linear models to be done quickly and smoothly.

3.2. Intelligent support in model structure determination

Whereas identifying a parametric model, one of the most difficult problems is declaration of the model structure, *i.e.* model order and time delays. This problem does not exist in non-parametric identification, *e.g.* in spectral analysis. To support the user in parametric model identification, the following approach is suggested and partially performed automatically:

1. Start from a model structure that is consistent with a prior knowledge.
2. Estimate model parameters using suitable algorithm according to a chosen approximation criterion.
3. Validate the estimated model.
4. If the result is not good enough, try another model structure and repeat from step 2, until a suitable model is obtained.

The above identification loop can be repeated a number of times. Its crucial part is handling the search of "optimal" structure, *i.e.*, the structure for which the model is the best one according to the chosen approximation criteria. It is especially important when there is no reason to assume a model structure, *e.g.*, when a linear model is identified to approximate a dynamic behaviour of a nonlinear system working around the set point. The natural way to do this is an "exhaustive search", which means that models for all structures not exceeding the assumed upper bounds ought to be estimated. This approach assures that the "global optimum" will be achieved, but due to a huge number of possible model structures, particularly for MIMO models, it is too time-consuming. Therefore, the second method is implemented, based on the authors' experience. In this method, search through the structure space is driven by a set of heuristic rules that attempt to reach (sub) optimal point (the best structure) in a small number of steps. The first attempt to create this strategy was proposed in [17], the final version for the general linear model is presented in [20].

The next problem to be solved is the choice of a criterion for model selection in the searching loop. There is a lot of different criteria suggested in the literature, but basing on the authors' experience, four of them are proposed in MULTI-EDIP. Depending on the purpose of identification, the user can choose information criteria like AIC [1] or BIC [41], final output error FOE [43] or a minimum of prediction error calculated for a validation data set (so called cross-checking). Information criteria are suitable for purposes related to prediction - AIC is proposed when the quality of prediction is critical, whereas BIC allows for finding a simpler model, FOE is suitable for control and the last criterion can be applied if the validation data set is available. Detailed discussion on criteria for model structure selection is given in [20].

Thus, if the user of MULTI-EDIP does not know the proper structure of the model, it is sufficient to declare only the bounds for the searching procedure and a selection criterion with respect to the needs.

3.3. Principles of working with MULTI-EDIP

The fundamental idea assumed by the authors consists in separation of operations invoked by the user and parameter declaration for these actions. It means that the user can start any data processing without explicit setting of parameters required for this processing because programmed routines use parameters from the set declared in advance. This solution allows for obtaining results quickly, even after some "mouse clicks", however, on the other hand, the user can be unaware of implicitly used parameters. Hence, all information about executed operations and taken parameters as well as results of operations are presented in a report window (see Fig.1). Sequence of actions is not enforced in the package - the user is able to invoke consecutive functions from the set presented in the next section, according to his demands and results obtained during processing. After generating data, he can start directly identification task or he can begin with preprocessing data as well. If the outcomes of invoked operations are not satisfactory, he can change settings (*e.g.*, model structure, validation tests, subsequence of data, band for filtration,*etc.*) and repeat actions for new settings or induce a new sequence of actions.

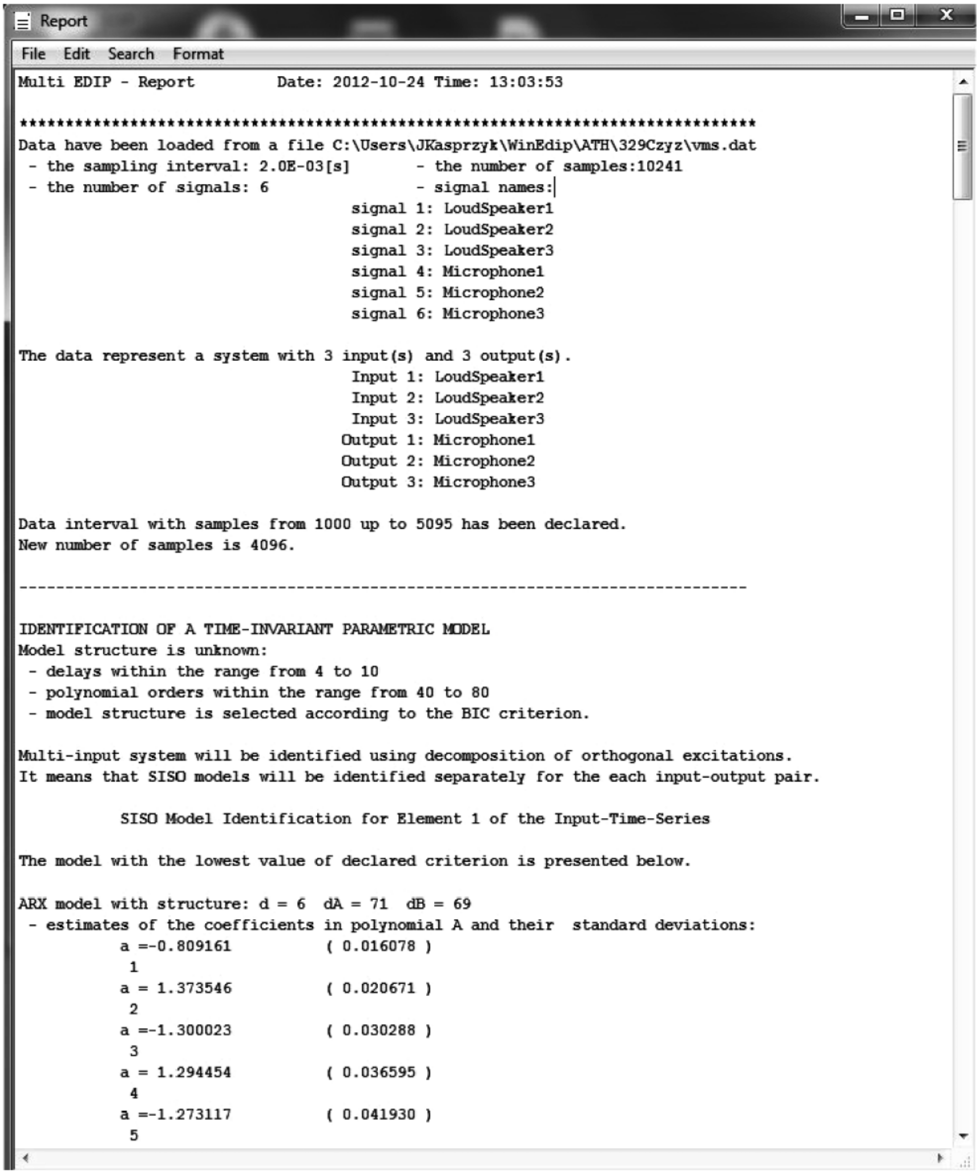

Figure 1. The report window of MULTI-EDIP.

3.4. Graphical user interface

The main window of MULTI-EDIP is a multi-document interface (see Fig. 2). This window is an area where other children-windows may appear. They display many kinds of plots like time- and frequency-domain plots, histograms, polar or Nyquist plots. The results of parameter model estimation, the effects of model validation tests, information about data preparation and operations done on files are presented in a report window. During the session, contents of this window can be edited, saved in a file or printed.

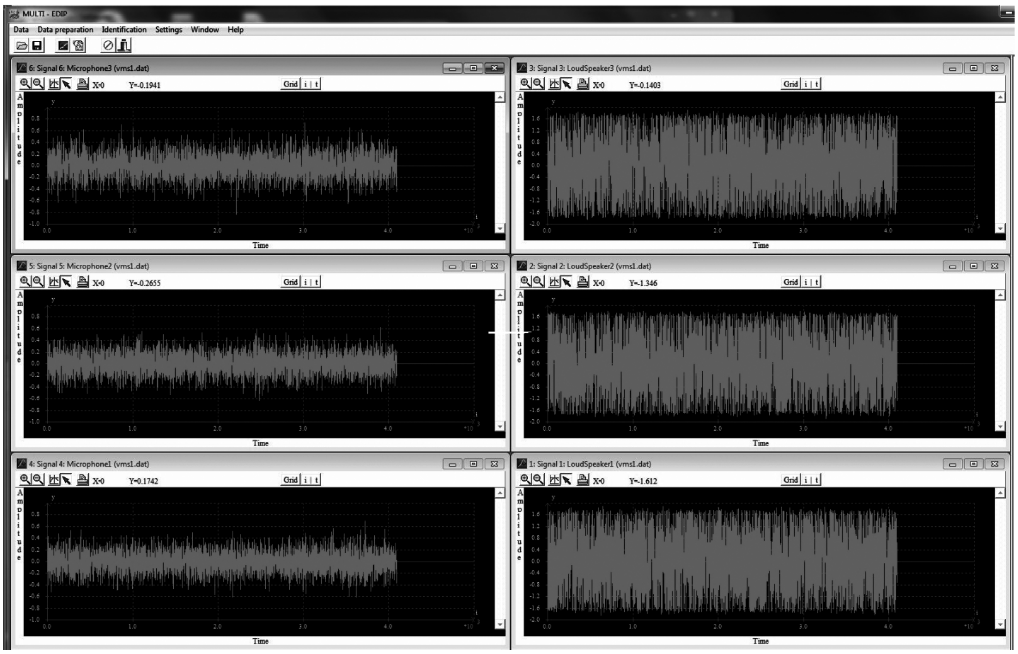

Figure 2. The main window of MULTI-EDIP with diagrams of signals taken in the example below.

Identified nonparametric models, results of some model validation tests as well as time-domain plots of prepared data are presented as diagrams. Each kind of a diagram allows for some actions such as reading of coordinates of any point in the diagram, change of scale (logarithmic, linear), zooming of a chosen part of the diagram, data carrying between windows, showing the difference between diagrams from two windows, comparing with results saved in a file or with a mean value of a group of files, saving results in a file and printing the diagram. It was assumed that background of a diagram should be black, as it resembles an oscilloscope monitor, but also because this solution saves eyesight.

For time-varying models, the results are presented as three-dimensional diagrams showing how identified process varies in time. In such a window it is possible to plot and move across-section plane, to change a point of watching on a diagram, to show a cross-section in a two-dimensional window and to print a diagram.

4. MULTI-EDIP features

MULTI-EDIP starts with the main menu presented in Fig. 2. It offers the following modes of MULTI-EDIP functioning: data generation (option Data), data preparation, model identification and model validation (option Identification). Their accomplishment depends on the configuration parameters declared by the user invoking option Settings from the main menu. Option Help offers basic knowledge concerning models supported by MULTI-EDIP, identification procedures and their parameterization as well as usage of MULTI-EDIP.

4.1. Data generation

The source of data used by MULTI-EDIP are data files. Data they contain may have been generated either by simulation, by real-world identification experiments or acquired with a specialized precision analog multi-input multi-output unit. The files can hold also data saved after preprocessing procedure. The associated application SIMULATOR MULTI-EDIP allows for simulating a broad spectrum of deterministic and stochastic time-series, scalar or vector, as well as MIMO dynamic systems. Simulation mode gives the user plenty of opportunities to test all identification procedures. Files can be created also in any other application generating data by simulation, by acquiring measurements directly from analogue input devices, by collecting them from data acquisition systems or reading from data bases. No matter what mechanism is used for data generation, the user is urged to generate two data sets for each experiment: an estimation data set, used for model identification, and a validation data set, used for model validation.

SIMULATOR MULTI-EDIP may be also used as a flexible tool for designing a wide range of stationary or nonstationary excitation signals for SISO, MISO and MIMO system identification. The user can generate scalar and vector signals consisting of step, impulse, sine and piece-wise deterministic components, pseudo-random multilevel and binary components, scalar and vector orthogonal and nonorthogonal multisine components [11], and stationary as well as nonstationary ARMA time-series. Generated signals can be used as excitations in identification experiment on a real-world system, as in the example presented in sec. 5.

4.2. Data preparation

Raw data, *i.e.*, data collected from some identification experiment, are not likely to be suitable for immediate processing by some identification algorithms because of their possible deficiencies or the need to enhance some features of particular interest. In data preparation mode, MULTI-EDIP offers the following services:

- data checking (removing outliers, calculating histograms or statistical parameters, testing time-invariability, *etc.*);
- data editing (decimation, interpolation, choice of a subsequence of interest, change of the sampling interval);

- data preprocessing (filtering the sample sequence, normalization, data scaling, removing averages, polynomial trends or periodical components, integration, differentiation).

4.3. Model identification

MULTI-EDIP provides support for identification of time-series models, scalar or vector, such as:

- stationary or time-varying stochastic parametric models (AR, MA, ARMA and theirs integrated versions);
- deterministic models (polynomial trends, discrete spectra);
- nonparametric models - correlation and frequency-domain models, stationary and time varying (auto- and cross-correlation functions, cepstra, power spectral density, and synch spectrum).

The next class of models supported by MULTI-EDIP consists of stationary and time-varying models of systems, SISO, MISO or MIMO, such as:

- parametric models (ARX, ARMAX, transfer function models, FIR, OE, BJ, *etc.*);
- nonparametric models - correlation and frequency domain models (*e.g.*, frequency transfer functions, coherence functions, power spectral density of disturbances, *etc.*).

The basic estimation methods used by MULTI-EDIP are:

- for parametric models: ordinary least squares, recursive least squares with exponential data discounting, recursive prediction error method, recursive pseudolinear regression, instrumental variable method;
- for nonparametric models: correlation and classical spectral estimation methods, parametric methods for discrete and continuous spectra identification.

For spectral analysis of time-varying processes there are two ways of processing data:

- direct methods, based on classical spectral analysis - consecutive models are estimated for a time window moving across the data;
- indirect methods - parametric model with assumed structure are identified using a recursive algorithm with exponential data discounting (so called forgetting factor) and used to compute a spectral model.

Details on the methods and models presented above, as well as validation tests, can be found in popular textbooks, see, *e.g.*, [3], [4], [16], [21], [26], [23], [36], [37] [38], [39] or [42].

4.4. Model validation

Model validation is the process of establishing or refuting the soundness of a particular model. This validation is usually done with a special data set, the so called validation data set, to be generated together with the basic estimation data set. The essence of validation is to subject an obtained model to some tests. MULTI-EDIP offers the following validation tests:

- visual tests (simulation, one-step ahead prediction, step- and pulse- responses, frequency transfer functions) which allow for detecting models that are obviously inadequate;
- correlation tests (whiteness of prediction error, correlation between inputs and prediction error) - to test the consistency of the model;
- condition number tests (condition number of data matrix, condition number of input- and output- correlation matrices) and overfitting tests (pole-zero cancellation) - to check for ill-conditioning of estimation problem and for the reasons for this;
- comparison tests (loss function, information criteria AIC, BIC) - to compare models of different structure;
- tests of model sensitivity on randomness of data - based on Monte Carlo methods.

MULTI-EDIP is asking by default for a validation data set for any of these tests. However, this may be overridden by the user who may apply the estimation data set for the same purposes.

5. Example of expert action

MULTI-EDIP proved its useful in solving many practical problems of model building. To illustrate its features, exemplary identification of an electro-acoustic plant for designing an adaptive control system for noise attenuation is presented, see [19]. In the example considered, a model is needed to design a feedforward multichannel active noise control (ANC) system (see, *e.g.*, [28]) creating a local 3-dimensional zone of quiet in an enclosure. The ANC system consists of three microphones measuring noise, and three loud speakers generating sound for noise attenuation. For proper parametrization of the ANC system, a dynamic model in the form of transfer functions between signals send to the loudspeakers and responses measured by the microphones is required. This electro-acoustic plant consists of D/A converters, reconstruction filters, amplifiers, 3 control loudspeakers, and acoustic space between the loudspeakers and microphones. Thus, identification of the plant is not a trivial task as it is a MIMO system with complicated dynamics and there are not any prerequisites for model structure assumption.

Moreover, parametric model of the plant is to be identified, but simultaneously the model should well describe the features of the plant in the frequency domain, as the accuracy of its frequency response determine the final result of ANC application. In this example MULTI-EDIP is used to: (1) generate input signals in Simulator, (2) identify structure and parameters of MIMO model of the plant, (3) validate model by comparison of its frequency responses with responses obtained by classical spectral analysis.

5.1. Model of the plant

Let $S_{sr}(z^{-1})$ denotes transfer function of an electro-acoustic path including the rth ($r = 1,2,3$) loudspeaker and sth ($s = 1,2,3$) microphone. The aim of identification is to determine estimates of all elements of the transfer function matrix

$$S(z^{-1}) = \begin{bmatrix} S_{11}(z^{-1}) & S_{12}(z^{-1}) & S_{13}(z^{-1}) \\ S_{21}(z^{-1}) & S_{22}(z^{-1}) & S_{23}(z^{-1}) \\ S_{31}(z^{-1}) & S_{32}(z^{-1}) & S_{33}(z^{-1}) \end{bmatrix}. \tag{1}$$

Elements of the matrix (1) can be parameterized as

$$S_{sr}(z^{-1}) = z^{-d_{sr}} \frac{B_{sr}(z^{-1})}{A_{sr}(z^{-1})}, \tag{2}$$

where:

$$A_{sr}(z^{-1}) = 1 + a_{1,sr}z^{-1} + \cdots + a_{dA_{sr},sr}z^{-dA_{sr}}, \tag{3}$$

$$B_{sr}(z^{-1}) = b_{0,sr} + b_{1,sr}z^{-1} + \cdots + b_{dB_{sr},sr}z^{-dB_{sr}}. \tag{4}$$

Structure of the model can be defined as the set of integer numbers characterizing all time delays d_{sr} and polynomial orders dA_{sr} and dB_{sr}. To obtain the model $\hat{S}(z^{-1})$, an identification experiment was performed in an enclosure. A vector white multisine signals with orthogonal components [10], [11] were generated by MULTI-EDIP Simulator for $N = 4096$ data and then used to excite three loudspeakers. Acoustic response measured by microphones was sampled at $T = 2[ms]$ and stored. Then all these signals (shown in Fig. 2) were processed by MULTI-EDIP to calculate the transfer function matrix (1) using decomposition method taking advantage of excitation orthogonality, see [9]. Fig. 3 presents power spectral densities calculated for input excitations showing its orthogonality.

Because there is not *a priori* knowledge about the proper model structure, it is assumed that the goal of identification is to find not only coefficients of the polynomials $A_{sr}(z^{-1})$ and $B_{sr}(z^{-1})$, but also to determine a triplet $(d_{sr}, dA_{sr}, dB_{sr})$ for all 9 channels. On the basis of previous experiments and analysis of the system geometry, it was assumed that the search space is to be bounded: for delays from 4 to 10 and for polynomial orders from 40 to 80, whereas model selection is based on BIC, see Fig. 4. Model structures selected by the procedure for all paths are presented in Table 1. It is worth noticing the very high orders of polynomials – it is the effect of sound reverberation

Figure 3. Power spectral densities calculated for three elements of input signal - a zoom in frequency.

in the enclosure that makes the frequency characteristic very complicated, having many peaks and deep valleys. Estimated delays agree well with layout of devices in the enclosure, *i.e.*, distances between loudspeakers and microphones. It is also worth emphasizing that MULTI-EDIP was able to give identification outcomes for all 9 channels after few dozen seconds. It is an effect of the proposed search algorithm as well as a specially programmed least squares method [20].

	Control loudspeaker 1	Control loudspeaker 2	Control loudspeaker 3
Microphone 1	$(6,71,69)$	$(5,80,79)$	$(7,73,73)$
Microphone 2	$(4,61,61)$	$(6,53,56)$	$(4,55,54)$
Microphone 3	$(7,67,67)$	$(7,79,80)$	$(6,78,75)$

Table 10. Results of model structure identification $(d_{sr}, dA_{sr}, dB_{sr})$ using BIC criterion

5.2. Model validation

Because frequency features of the identified model may influence its implementation in ANC system, model validation should be based, among others, on a comparison

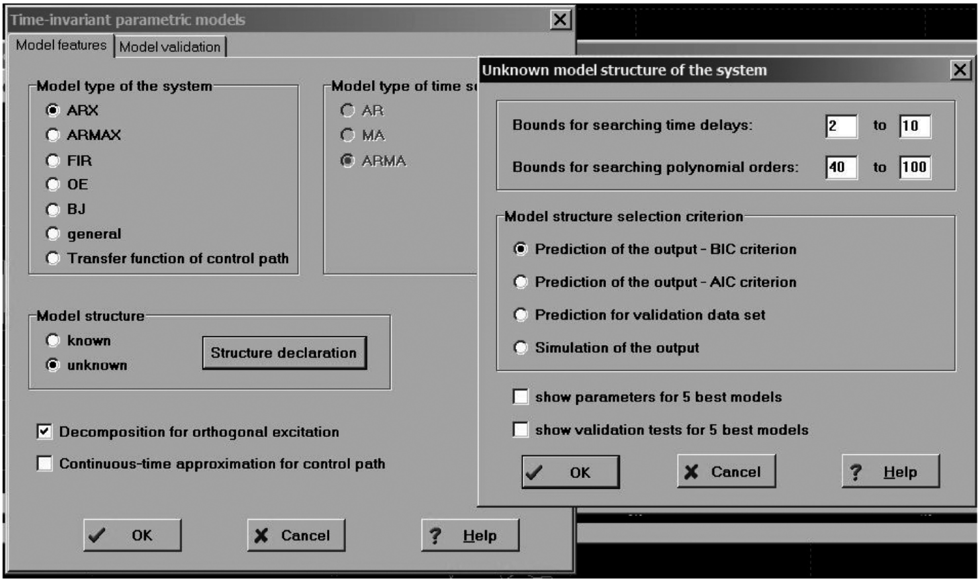

Figure 4. Windows for structure searching declaration.

of the frequency response calculated for identified transfer function with frequency response obtained *via* classical spectral analysis. MULTI-EDIP offers such a possibility, and for instance, Fig. 5 presents a gain of frequency response function of the path between the first loudspeaker and the second microphone calculated for identified $S_{12}(z^{-1})$ and by Blackman-Tuckey method. The plot shows a very good conformity of the results obtained by different methods. This laborious example show usefulness and versatility of MULTI-EDIP as a CASSI tool.

Obtained models were then applied in a multichannel ANC system and demonstrated their value providing good noise attenuation [27].

6. Conclusions

During nearly 30 years of evolution process, MULTI-EDIP has been used in many education, research and industrial projects, as diverse as military equipment or physiology and medicine, see *e.g*, a description of using MULTI-EDIP in identification of ventilatory response in congestive heart failure patients [40]. An intensive use of the package and user's remarks were helpful in development of its features. The authors believe, that MULTI-EDIP showed its usefulness and was able to support users in achieving relevant results. The main conclusion is that the automation of a knowledge-intensive task like SSI is feasible. However, it proved to be conceptually difficult, software-wise demanding and generally time-consuming. Much of the work has been really like sailing

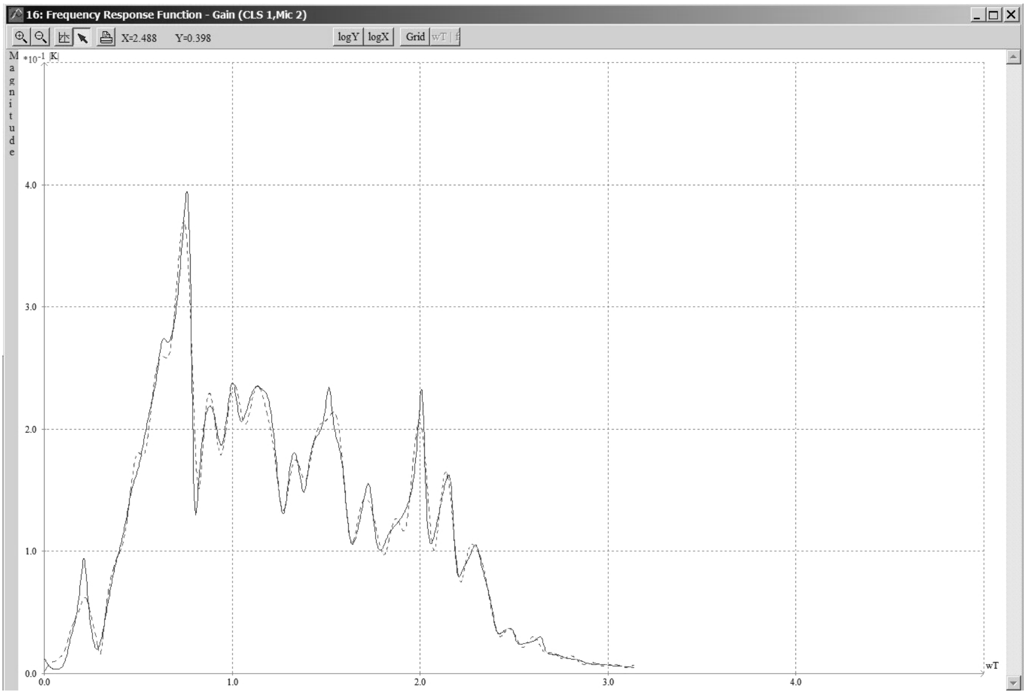

Figure 5. Comparison of the gain of frequency response function for the path between the first loudspeaker and the second microphone calculated for identified transfer function S_{12} (solid line) and obtained *via* Blackman-Tuckey method (dashed line).

upon uncharted waters. Thanks to extensive testing of MULTI-EDIP, both by a sizeable student population working on laboratory projects and Ph.D. students working on their theses, many simple and sophisticated errors have been discovered and removed. Some new ideas still await testing and implementing, *e.g*, model identification based on higher order spectra.

Acknowledgments

The partial financial support of this research by Polish Ministry of Science and Higher Education is gratefully acknowledged.

References

[1] H. AKAIKE: A new look at the statistical model identification. *IEEE Trans. on Automatic Control*, **19** (1974), 716-723.

[2] ADAPTx Automated Multivariable System Identification and Time Series Analysis Software. (http://www.adaptics.com/)

[3] J.S. BENDAT and A.G. PIERSOL: Random Data Analysis and Measurement Procedures. John Wiley & Sons, Inc., New York, 1986.

[4] J.S. BENDAT and A.G. PIERSOL: Engineering Applications of Correlation and Spectral Analysis. John Wiley & Sons, Inc., New York, 1993.

[5] T. BOHLIN: Interactive System Identification: Prospects and Pitfalls. Springer Verlag, Berlin, 1991.

[6] F.J. DE CANETE, S. GONZALEZ-PEREZ and P. SAZ-OROSCO: Software tools for system identification and control using neural networks in process engineering. World Academy of Science, Engineering and Technology, **47** (2008), 59-63.

[7] CAPTAIN - Computer-Aided Program for Time-Series Analysis and Identification of Noisy Systems. (http://www.es.lancs.ac.uk/cres/captain/)

[8] CUEDSID: Cambridge University System Identification Toolbox. (http://www-control.eng.cam.ac.uk/jmm/cuedsid/)

[9] J. FIGWER, A. NIEDERLIŃSKI and J. KASPRZYK: A new approach to the identification of linear discrete-time MISO systems. *Archives of Control Sciences*, **2**(3-4), (1993), 223-237.

[10] J. FIGWER and A. NIEDERLIŃSKI: Using the DFT to synthesize multivariate orthogonal white noise series. *Trans. of the Society for Computer Simulation*, **12**(2), (1995), 749-758.

[11] J. FIGWER: Synthesis and Simulation of Random Processes. Silesian Univeristy of Technology Press, Gliwice, 1999, ISSN 0434-0760, (in Polish).

[12] H. GARNIER, M. GILSON, T. BASTOGNE and M. MENSLER: CONTSID toolbox: a software support for continuous-time data-based modelling. In: Identification of continuous-time models from sampled data, H. Garnier and L. Wang (Eds.), Springer-Verlag, London, 2008, 249-290.

[13] S. GENTIL, A. BARRAUD and K. SZAFNICKI: SEXI: An expert identification package. *Automatica*, **26** (1990), 803-809.

[14] J. GUZMAN, D. RIVERA, S. DORMIDO and M. BRENGUEL: An interactive software tool for system identification. *Advances in Engineering Software*, **45** (2011), 115-123.

[15] M. HAEST, G. BASTIN, M. GEVERS and V. WERTZ: ESPION: an expert system for system identification. *Automatica*, **26** (1990), 85-95.

[16] G.M. JENKINS and D.G. WATTS: Spectral Analysis and its Applications. Holden-Day, San Francisco, 1968.

[17] J.KASPRZYK: Model structure determination in parametric model identification. *Systems Science*, **23**(2), (1997), 89-95.

[18] J. KASPRZYK: MULTI-EDIP – an interactive software package for process identification. *Proc. of the 13th IFAC Symp. on System Identification SYSID'2003*, Rotterdam, Netherlands, (2003), 1484-1489.

[19] J. KASPRZYK: Model identification for active noise control - a case study. *Archives of Control Sciences*, **14**(3), (2004), 219-243.

[20] J. KASPRZYK: Computer-Aided Process Identification. Silesian University of Technology Press, **1720**, Gliwice, Poland, 2006, ISSN 0434-0760, (in Polish).

[21] S.M. KAY: Modern Spectral Estimation: Theory and Applications. Prentice Hall, Englewood Cliffs, 1986.

[22] I. KOLLAR, R. PINTELON and J. SCHOUKENS: Frequency domain system identification toolbox for MATLAB: Characterizing nonlinear errors of linear models. *Proc. of the IFAC Symp. on System Identification, SYSID'2006*, Newcastle, Australia, Pergamon Press, (2006), 726-731.

[23] L. LJUNG: System identification: Theory for the User. (Second ed.), Prentice-Hall, Upper Saddle River, New Jersey 1999.

[24] L. LJUNG: System Identification Toolbox. User's Guide. R2011a. The MathWorks Inc., 2011.

[25] I. MARKOVSKY, J. WILLEMS, S. HUFFEL and B. MOOR: Software for approximate linear system identification. *Proc. of the 44th IEEE Conf. on Decision and Control*, and *The European Control Conf.*, Seville, (2005), 1559-1564.

[26] S.L. MARPLE: Digital Spectral Analysis with Applications. Prentice Hall, Englewood Cliffs, 1987.

[27] M.I. MICHALCZYK: Adaptive Control Algorithms for Three-Dimensional Zones of Quiet. Jacek Skalmierski Computer Studio, Gliwice, Poland, ISBN 83-89105-67-5, 2004.

[28] P.A. NELSON, S.J. ELIOT: Active Control of Sound. Academic Press, London, 1992.

[29] NI LabVIEW System Identification Toolkit. (http://sine.ni.com/nips/cds/view/p/lang/en/nid/209045)

[30] A. NIEDERLIŃSKI, J. KASPRZYK and J. FIGWER: EFPI – an integrated software environment for system and signal identification. *Proc. of the 9th IFAC/IFORS Symp. on Identification and System Parameter Estimation*, Budapest, **1** (1991), 567-572.

[31] A. NIEDERLIŃSKI, J. KASPRZYK and J. FIGWER: EFPI – Expert for Process Identification. Software and User's Manual. World Scientific Publishing Co. Ltd., 1994.

[32] A. NIEDERLIŃSKI, J. KASPRZYK and J. FIGWER: MULTI-EDIP – analyzer of multivariable signals and systems. Silesian University of Technology Press, Gliwice, Poland, 1997, ISSN 0434-0825, (in Polish).

[33] B. NINNESS, A. WILLS and S. GIBSON: The University of Newcastle identification toolbox. *16th IFAC World Congress*, Prague, Czech Republic, (2005).

[34] P. VAN OVERSCHEE, B. DE MOOR, S. BOYD, H. ALING and R. KOSUT: A fully interactive identification module for Xmath (ISID). *Prep. of 10th IFAC Symp. on System Identification*, Copenhagen, Denmark, **4** (1994), 1-2.

[35] K.M. PASSINO: Bridging the gap between conventional and intelligent control. *IEEE Control Systems*, **13**(3), (1993), 12-18.

[36] U.S. PILLAI and T.I. SHIM: Spectrum Estimation and System Identification. Springer-Verlag, New York, 1993.

[37] R. PINTELON and J. SCHOUKENS: System Identification. A Frequency Domain Approach. IEEE Press, New York, 2001.

[38] T. SÖDERSTRÖM and P. STOICA: System Identification. Prentice Hall International, London, 1989.

[39] P. STOICA and R.L. MOSES: Introduction to Spectral Analysis. Prentice Hall, 1987.

[40] Z. TOPOR, L. JOHANNSEN, J. KASPRZYK and J. REMMERS: Dynamic ventilatory response to CO2 in congestive heart failure patients with and without central sleep apnea. *J. of Applied Physiology*, **91** (2001), 408-416.

[41] G. SCHWARZ: Estimating the dimension of a model. *The Annals of Statistics*, **6** (1978), 461-464.

[42] H. UNBEHAUEN and G.P. RAO: Identification of Continuous-Time Systems. North Holland Systems and Control Series, Amsterdam, 1987.

[43] Y. ZHU: Use of error criteria in identification for control. *Proc. of the 12th IFAC Symp. on System Identification (SYSID'2000)*, Santa Barbara, CA, **1** (2000).