Polska Akademia Nauk • Instytut Badań Systemowych

AUTOMATYKA STEROWANIE ZARZĄDZANIE

Książka jubileuszowa
z okazji
70-lecia urodzin

PROFESORA KAZIMIERZA MAŃCZAKA

pod redakcją
Jakuba Gutenbauma

Polska Akademia Nauk • Instytut Badań Systemowych

AUTOMATYKA STEROWANIE ZARZĄDZANIE

**Książka jubileuszowa
z okazji
70-lecia urodzin**

PROFESORA KAZIMIERZA MAŃCZAKA

**pod redakcją
Jakuba Gutenbauma**

Warszawa 2002

Książka jubileuszowa z okazji
70-lecia urodzin
Profesora Kazimierza MAŃCZAKA

Redaktor
prof. dr hab. inż. Jakub Gutenbaum

Warszawa 2002

ISBN 83-85847-78-2

Wydawca: Instytut Badań Systemowych PAN
ul. Newelska 6 01-447 Warszawa
http://www.ibspan.waw.pl

Opracowanie składopisu: Anna Gostyńska, Jadwiga Hartman

Druk: KOMO-GRAF, Warszawa
nakład 200 egz., 34 ark. wyd., 31 ark. druk.

A STRATEGY AND TOOL FOR COMPUTER AIDED PROCESS IDENTIFICATION

Antoni Niederliński, Jerzy Kasprzyk, Jarosław Figwer
Instytut Automatyki, Politechnika Śląska, Gliwice

Abstract: *The problem of computer aided process identification is discussed. Reasons for the development of intelligent software for computer aided process identification are presented. The main functions and properties of an integrated software environment MULTI-EDIP for computer aided process identification are described. A summary of the most important MULTI-EDIP expert advice (model structure determination) is presented together with an application example for active noise control.*

Keywords: *Computer aided identification, intelligent identification, expert systems, active noise control.*

1. The essence of identification

One of the fundamental activities in science and technology is *modelling*. It can be defined as the attempt of representing *processes* we try to understand by means of *models*, which we already comprehend. The main goal of *identification* is to determine *mathematical models* describing the behaviour of processes based on prior knowledge and measurements collected – if possible – in specially designed identification experiments.

Further, two groups of processes are distinguished: signals and systems. The essential attribute of *signals* is that they can be only acquired, whereas it is not possible to influence them (e.g. meteorological or seismic phenomena). Mathematical models of signals are necessary, for example, to predict signals in some limited sense or to depict quantitatively their properties. *Systems* are characterized by the possibility of controlling their *outputs* by manipulating their *inputs*. Models of systems are useful, e.g., in more effective control of these systems.

Process identification is nowadays done in areas as diverse as *engineering*, including aeronautics, acoustics and vibration analysis, bioengineering, chemical engineering, control engineering, electronics, mechanical engineering, signal processing, technical diagnostic, telecommunication; *hard sciences* including physics and chemistry; *life sciences* including biology, ecology, agriculture, horticulture and medicine, *earth sciences* including geology, hydrology and seismography, *social sciences* including economics and sociology.

Process identification 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 systems 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 time– and frequency-domain models for dynamic systems and time-series,
- knowledge of estimation techniques,
- knowledge of process simulation techniques,
- knowledge of numerical methods,
- knowledge of some advanced programming language,
- knowledge of software techniques like data bases and computer graphics,
- experience in *real-world* and *computer simulated identification*,
- knowledge of the *real-world process* to be identified,
- identification know–how as mastered by identification experts.

2. Computer supported identification

The problem of whether process identification can be automated and to what extent is still unsettled. However, many early attempts at designing software tools for interactive intelligent identification, e.g. ESPION (Haest et al. 1990), SEXI (Gentil et al. 1990), EFPI (Niederliński et al. 1991), ISID (Overschee et al. 1994) and others, or monographs by Mańczak and Nahorski (1983), and Bohlin (1991), discussing basic techniques as well as prospects and pitfalls of interactive identification may well justify some fundamental questions related to the design of such tools.

It is generally accepted that some form of computer support is a necessary prerequisite for successful process identification, as the result of identification depends to a considerable extent upon the availability of a user-friendly system giving access to robust and high-quality numerical identifications software, some data base tools and some tools for producing graphical output.

It is, however, also generally known that for the successful process identification 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 identification should go beyond standard number crunching – and data base services (as made available e.g. by appropriate identification toolboxes) and deliver those services that may be expected from true identification experts. To use a catchword: computer supported identification should provide some *intelligent services.*

3. What is meant by intelligent process identification?

The word *intelligence* appears more and more 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 (Passino 1993). E.g. 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. automatic 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 (lets say *fuzzy logic, neural processing, genetic algorithms*), are also used as excuses for calling a particular control or identification contraption *intelligent*. This usage seems to contradict a long established pattern of utilizing the word *intelligent* in everyday and not-so-everyday speech. This pattern can be summarized as follows:

1. The adjective *intelligent* conveys both the meaning of *reasonableness* or *soundness* (which could be regarded as defining some good but standard action) as well as the meanings of *brilliancy*, *cleverness* and *brightness* (which are used to describe something that is above standard and somehow surprising). 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 the discussion to the second meaning of intelligent.

2. The adjective *intelligent* is granted some action as a result of judgement that might be called *expert acceptance*. This judgement is passed by experts, which enjoy the reputation of having some standard of what constitutes intelligent behaviour in similar actions.
3. The fact of calling some action *intelligent* does not seem to depend on the actor arriving at the action with the help of some particular philosophy of acting, which *eo ipso* guarantees intelligence of actions. *Intelligence* of action is generally and universally judged by its fruits, not by the method the action was determined. This method is anyway very seldom known.
4. The nature of *intelligence* seems to be rather transitory, it seems to belong to a receding horizon: what was considered to be intelligent at some particular instant of time might well be considered *routine, obvious* or *standard* at some later instant of time. E.g. to those steam-engine operators which were Maxwell's contemporaries, the steam engine speed controller might appear to be a highly intelligent piece of hardware, whereas to contemporary control engineers it is just a dull standard.

The above arguments make it easier to 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.*

This is conveyed by saying that the *Multivariate Systems and Signals Analyser MULTI-EDIP* is a tool, which not only provides basic number-crunching and data base services, but using some expert know-how, intelligently supports the user in consecutive steps of the process identification: from designing an experiment to verifying the model.

4. Basic services of MULTI-EDIP

MULTI-EDIP is the latest chain in a long development project. It started in the early 1980s at the Institute of Automatic Control; Silesian University of Technology, under the name *EDIP* (*Expert for Process Identification*), see Niederliński et al. (1991, 1994a). It has been continued during the years 1994-97 as *MULTI-EDIP for DOS* (Niederliński et al. 1997) and is being pursued up to now as *MULTI-EDIP for Windows* (Kasprzyk 2001).

On the basis of configuration parameters declared by the user, *MULTI-EDIP* offers the following modes: *data generation*, *data preparation*, *model identification* and *model validation*.

4.1. Data generation

The only source of data used by *MULTI-EDIP* are its own *data files*. The data they contain may have been generated either by *simulation* or by *real-world identification experiments*, using two associated applications:

1. The associated application *SIMULATOR MULTI-EDIP* allows to simulate a broad spectrum of deterministic and stochastic time-series, polynomial or neural, scalar or vector, as well as multidimensional dynamic systems. The *simulation mode* gives the user plenty of opportunities to test all identification procedures.
2. The associated application *PAIO MULTI-EDIP* allows to interface *MULTI-EDIP* with specialized precision analog input-output units. These units provide up to four analog-digital inputs to connect with measurement transducers and up to four digital-analog outputs to excite the system under consideration.

No matter what mechanism is used for data generation, the user is urged to generate for each experiments two data sets: an *estimation data set*, used for model identification, and a *validation data set*, used for model validation.

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 the 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),
- data preprocessing (filtering of the sample sequence, normalization, data scaling, removing of averages, polynomial trends or periodical components, integrating, differencing, etc.).

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 as well as their neural counterparts NAR, NMA, NARMA),
- deterministic models (models of polynomial trends, discrete spectra models),
- nonparametric models – correlation and frequency-domain models, stationary and time varying (auto- and cross-correlation functions, cepstrum, power spectral density, and synch spectrum).

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

- parametric models (*ARX*, *ARMAX*, their neural counterparts NARX and NARMAX, transfer function models, *FIR*, *OE,* 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, backpropagation, Levenberg-Marquardt,
- 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 the data:

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

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),
- basic tests (whiteness of prediction error and conditioning number of data matrix),
- additional tests (conditioning number tests for input- and output-correlation matrices, correlation between inputs and prediction error, pole-zero cancellation tests),
- comparison tests (loss function, information criteria such as *AIC, BIC*),
- tests of model sensitivity on randomness of data.

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. This option provides, in particular, an opportunity to learn about the inappropriateness of using estimation sets for model validation purposes.

The *main window* of *MULTI-EDIP* is a multi-document interface (MDI). 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 and 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, the contents of this window can be edited, saved in a file or printed.

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 diagram allows 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 transfer between windows, comparison 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.

For time-varying models, the results are shown as three-dimensional diagrams, presenting changes over time for identified models. In such a window it is possible to plot and move a cross-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.

5. Intelligent services of MULTI-EDIP

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 *EDIP*. The basic assumptions formulated and implemented in *EDIP* for single-input-single output systems and scalar signals, have been extended in *MULTI-EDIP* for multi-input multi-output systems and vector signals. They can be verbalized as follows:

1. Process/signal identification and model analysis/verification 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 process identification,* as for example: input-output models, time series models, model classes (polynomial, spectral, neural), types and structures, estimation procedures, model validation and model checking tests, not 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 process 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 identification.
2. Intelligent user support should be provided by:
    - combining the model to be identified with the most appropriate identification method,
    - offering expert advice for model structure selection,
    - providing a set of default values for some important parameters,
    - checking the correctness of some parameters declared by the user,
    - proposing a set of verification procedures.

Roughly speaking, the basic philosophy of *MULTI-EDIP* is anti*MATLAB*. Of course, the presented design concept involves an important trade-off between user friendliness and tool flexibility:

- the features built into *MULTI-EDIP* enable even less knowledgeable users to obtain satisfactory results and understand them,
- however, in *MULTI-EDIP* 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 spectrum of basic and intelligent services is broad enough to support typical *MULTI-EDIP* users, e.g. academic teachers running courses on process identification and engineers implementing advanced control systems. It covers, among others, the following supports:

- support for the process of designing an identification experiment including the choice of *sampling interval*, *choice of data record length* (e.g. advice on process-generated constraints on signal variance and data record length, advice on the interchangeability of signal variance and data record length), *type of signal prefiltering*, *type and parameterisation of the input signal*,
- support for the choice of a *type* of model from the given class, e.g. AR, MA or ARMA time-series, neural NAR, NMA or NARMA time series,
- support for the choice of a model *structure*, i.e. the polynomial orders of AR, MA or ARMA time-series and the number of hidden layers and neurons in the layers for neural NAR, NMA or NARMA time series,
- support for the choice and parameterisation of a proper numerical procedure like Least Squares, Recursive Least Squares, Backpropagation, Levenberg-Marquards etc.,
- support for the tasks of interpreting, analysing and verifying the identified model.

To support parametric model identification the following approach is suggested and partially performed automatically:

1. Start from a model structure that is consistent with prior knowledge.
2. Estimate model parameters using suitable algorithm according to estimation criterion.
3. Validate the estimated model.

4. If the result is not good enough, try another model structure and repeat from step 2, until suitable model is obtained.

This identification loop can be repeated a number of times. Its crucial part is handling of the search for the "optimal" structure, i.e. the structure for which the model is the best one according to the chosen 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 around the set point. The natural way to do this is an "exhaustive search", meaning that models for all structures not exceeding the assumed upper bounds be estimated. This approach assures that the "global optimum" will be achieved, but due to a huge number of possible model structures it is time-consuming. Therefore, a second method is implemented, based on the authors' experience. In this method, the search through the structure space is driven by a set of heuristic rules that attempt to reach a (sub) optimal point (the best structure) in a small number of steps (Kasprzyk, 1997).

The main techniques used to implement some of the services are

1. *Default values* and *default procedures,* suggested to users as they proceed using *MULTI-EDIP.* Needles to say, all the default suggestions may be overriden by users who think they know better or who just want to experiment in order to gain deeper knowledge of the techniques made available.
2. *Proper menu sequencing* to guide the user through actions to be performed. That means that at each phase of using *MULTI-EDIP* the user is first presented only with the options relevant at that phase. However, they can be overriden by any user wishing to proceed otherwise.
3. Checking correctness and reasonableness of parameters declared by users or procedures initiated by the user. Values not passing the check may be used further only by a separate users decision. Procedures, which are not basically sound (e.g. using the *estimation data set* instead of the *validation data set* for validation purposes) produce a warning, that may be overridden by the user.
4. *A hypertext context sensitive help system,* providing comments, suggestions and explanation of what has been obtained or what should be done next.
5. *A simulation mode,* allowing the user to test 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 users confidence in the tool and preparing them to use it effectively.

6. Example of expert identification

MULTI-EDIP proved useful for the development of active noise control systems (Niederliński 1999). A typical identification problem occurring in active noise applications is the identification of acoustic plant models for the development of feedforward control systems. In the example considered the model is needed to design an adaptive feedforward active noise control system creating a local 3-dimensional zone of quiet surrounding a single (error) microphone in a reverberant enclosure.

The error microphone was placed in a laboratory enclosure of about 23 m^3 of volume. This enclosure is disturbed by a noise (generated by primary source) which should be reduced using two secondary sources (control loud-speakers). The control loud-speakers were placed at about 0.6 m and 1 m apart from the error microphone. The block diagram of the feedforward active noise control system is shown in Fig. 1.

A reference microphone placed near to the primary noise source was used to measure the reference signal $x(i)$. $P(z^{-1})$ is the transfer function of the disturbance path. It represents an acoustic space between the reference and error microphones. $S_1(z^{-1})$ and $S_2(z^{-1})$ are secondary path transfer functions representing D/A converters, reconstruction filters, amplifiers, control loudspeakers and an acoustic space between these loudspeakers and the error microphone. $F_1(z^{-1})$ and $F_2(z^{-1})$ denotes acoustic feedback path transfer functions composed of D/A converters, reconstruction filters, amplifiers, control loudspeakers and the acoustic space between these loudspeakers and the reference microphone. A digital FIR filters are used as the compensators $W_1(z^{-1})$ and $W_2(z^{-1})$. Their coefficients are tuned on the basis of the error signal $e(i)$ and the reference signal $x(i)$ filtered through the secondary path transfer function estimates. The goal of the adaptation algorithm is to calculate the coefficients of digital FIR filters that minimise mean square value of the error signal.

The structure of control system implies that there is a need to identify the secondary as well as the acoustic feedback path transfer functions before activating the active noise control system. It should be noticed that models of secondary paths $\hat{S}_1(z^{-1})$ and $\hat{S}_2(z^{-1})$ are used to filter signal $x(i)$ while models of acoustic feedback paths $\hat{F}_1(z^{-1})$ and $\hat{F}_2(z^{-1})$ are used to cancel this feedback. In both identification cases the structure of plant to be identified is of two-input single-output (TISO) type.

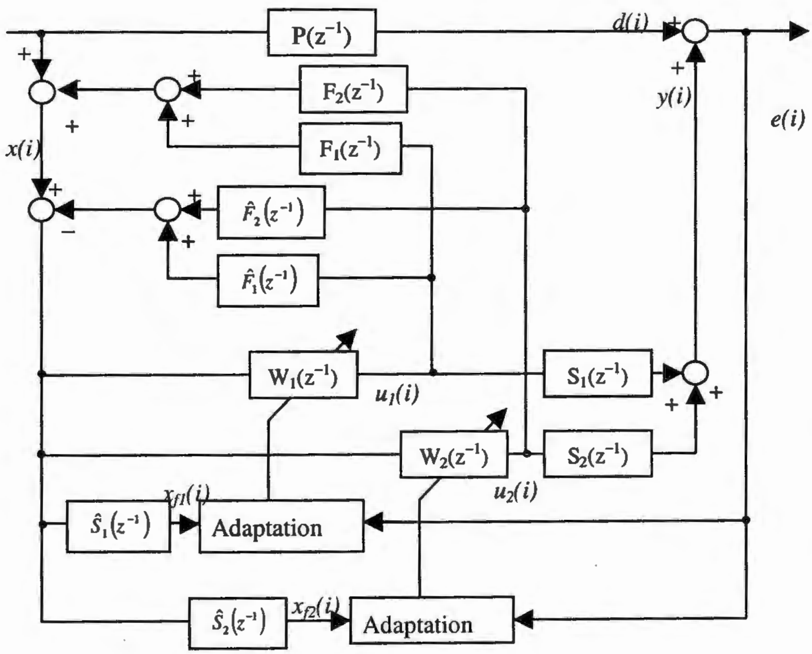

Fig.1. The block diagram of active noise control system.

Only the experimental identification of the secondary path transfer functions will be presented in details as the models of acoustic feedback paths may be identified in a similar way.

For the identification experiment bivariate multisine excitation (Niederliński and Figwer 1995) with the period $N = 2048$ and standard deviation 0.71 was generated and used to drive the control loud-speakers *via* PAIO. The sampling frequency was 500 Hz. The excitation was repeated 4 times after the plant reached steady state condition.

It was assumed that the TISO plant between inputs $u_1(i)$ and $u_2(i)$ and output $y(i)$ can be modelled by an ARX model:

$$A(z^{-1})y(i) = \sum_{r=1}^{2} z^{-d_r} B_r(z^{-1})u_r(i) + v(i)\,, \tag{1}$$

where: z^{-1} is the backward shift operator, $u_r(i)$ and $y(i)$ are $i-$th samples of $r-$th $(r=1,2)$ input and output respectively, d_r denotes the pure delay in the *r*-th control path, $A(z^{-1})$ and $B_r(z^{-1})$ are polynomials of the operator z^{-1} with orders dA and dB_r respectively, $v(i)$ is a white noise used to model disturbances acting on the output signal.

A model structure is specified by a set of integer numbers defining polynomial orders and delays. In this case the model structure can be defined as:

$$([d_1, d_2], [dB_1, dB_2], dA). \tag{2}$$

Simulation and real-data identification experiments carried out by the authors showed that model quality and computing time depend, above all, on the proper choice of a starting point, so a heuristic algorithm of searching a (sub) optimal structure was implemented in *MULTI-EDIP*. In this case it may be summarized as follows:

1. For each pair – input $u_r\,(r=1,2)$ and output y – search for the optimal structure of a SISO model.
2. As the starting point in the TISO system identification, for consecutive *r*-th control channel take the orders and delay obtained for the *r*-th SISO model, but for common polynomial (i.e. $A(z^{-1})$) choose the greatest order of appropriate polynomial in SISO models. Then find (by varying orders or delays) such a TISO model structure that minimizes the chosen criterion (e.g. *AIC*).

As a rule, an overparametrisation (i.e. too high orders) occurs while SISO model identification is performed for a TISO system. It is rather obvious, because the model attempts to explain the overall input-output behaviour only by means of one input-output relation. Therefore, handling the search of polynomial orders in the second step is usually done by decreasing the polynomial orders.

Step 1 consists of the following stages:

1.1. For each input identify the first order SISO models for consecutive delays within the admissible set and choose this delay, for which the minimal cost function occurs.

1.2. For the chosen delay look for the SISO model with the same orders of each polynomial, which minimizes the chosen criterion.

1.3. For the selected structure try to change the orders of consecutive polynomials until a minimum of the criterion is reached.

1.4. Repeat the searches for the admissible delay until a minimum of the criterion is reached.

The complexity of the identified acoustic plant (e.g. the reconstruction filters are Butterworth 8th order) implies that identified model should be of a high order, so models having polynomial orders within the range 8 – 22 were tested. For greater orders data matrices were ill conditioned. Similarly, taking into account the dimensions of the plant, it was assumed that delays may vary within the range of 2-10.

The results of consecutive steps in model structure selection using *AIC* criterion are as follows:

1. Identification of the first channel.
   - In the first step the delay $d_1 = 2$ has been determined.
   - Models of orders $dA_1 = dB_1 = 8,\ldots,22$ were tested and a model of order 21 has been determined.
   - Finally, a model with the structure *(3,21,22)* was found, minimizing *AIC*.
2. Identification of the second channel.
   - In the first step the delay $d_2 = 10$ has been determined.
   - Models of orders $dA_2 = dB_2 = 8,\ldots,22$ were tested and a model of order 21 has been determined.
   - Finally, a model with the structure *(10,21,22)* was found, minimizing *AIC*.

3. Identification of the entire TISO model.

    - Starting from the structure ([*3,10*],[*21*,*22*],*22*) and changing consecutive delays and orders, a final model with the structure ([*9*,6],[*14,21*],*19)* was found.

It is worth to emphasize that in the procedure 23 SISO models for the first channel, 19 SISO models for the second channel and 63 TISO models were estimated and tested while in an "exhaustive search" 273375 models ought to be estimated and tested. However, it can not be guaranteed that the global minimum of *AIC* has been achieved.

7. Conclusions

The main conclusion is that automation of a knowledge-intensive task like process identification is feasible. However, it proved to be a conceptually difficult, software-wise demanding and generally time-consuming. Much of the work has been really like sailing 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 theses on active noise control, many simple and sophisticated errors have been discovered and removed. Some new ideas still wait to be tested.

References

Bohlin T. (1991) *Interactive System Identification: Prospects and Pitfalls.* Springer Verlag, Berlin.

Gentil S., Barraud A., Szafnicki K. (1990) SEXI: An Expert Identification Package. *Automatica*, **26**, 803-809.

Haest M., Bastin G., Gevers M., Wertz V. (1990) ESPION: an Expert System for System Identification. *Automatica*, **26**, 85-95.

Kasprzyk J. (1997) Model Structure Determination in Parametric Model Identification, *Systems Science*, **23**, 2, 89-95.

Kasprzyk J. (2001) Computer Aided Process Identification Using MULTI-EDIP, *Proceedings of the IASTED International Conference Applied Simulation and Modelling*, Marbella, 13-18.

Ljung L. (1987) *System identification - Theory for the user.* Prentice-Hall, Englewood Cliffs, N.J.

Mańczak K., Nahorski Z. (1983) *Komputerowa identyfikacja obiektów dynamicznych* PWN, Warszawa.

Niederliński A., Kasprzyk J., Figwer J. (1991) EFPI- An Integrated Software Environment for System- and Signal Identification. *Proceedings of the 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, Budapest, 567-572.

Niederliński A., Kasprzyk J., Figwer J. (1994a) Computer Aided Process Identification. In: *Computer Aided Control System Design.* M. Brdyś and K. Malinowski, eds. World Scientific Publishing Co. Pte. Ltd., Singapore, 73-98.

Niederliński A., Kasprzyk J., Figwer J. (1994b) Intelligent Identification: Decision Trees or Knowledge Bases. *Preprints of 10th IFAC Symposium on System Identification*, Vol. 2, Copenhagen, 471-476.

Niederliński A., Figwer J. (1995): Using the DFT to Synthesize Bivariate Orthogonal White Noise Series. *IEEE Transactions on Signal Processing*, **3**, 749-758.

Niederliński A., Kasprzyk J., Figwer J. (1997) *MULTI-EDIP - Analizator Wielowymiarowych Sygnałów i Obiektów.* Wydawnictwo Politechniki Śląskiej, Gliwice (in Polish).

Niederliński A. (1999). Identyfikacja i adaptacja dla aktywnmego tłumienia hałasu. Referat plenarny, *XIII Krajowa Konferencja Automatyki, Materiały Konferencyjne*, **1**, Oficyna Wydawnicza Politechniki Opolskiej, Opole, 37-44.

Overschee P. van et al. (1994) ISID II A Fully Interactive Identification Module for Xmath. *Preprints of 10th IFAC Symposium on System Identification*, **4**, Copenhagen, 1-2.

Passino K. M. (1993) Bridging the Gap between Conventional and Intelligent Control. *IEEE Control Systems*, June 1993, 12-18.

Acknowledgment

The partial financial support of this research by the State Committee for Scientific Research (KBN) under the grant BK-215/RAU-1/02 and grant 4 T11A 010 22 is gratefully acknowledged.

ISBN 83-85847-78-2