C 37

8th IFAC Symposium on Identification and System Parameter Estimation

Beijing, 27-31 August 1988

AN EXPERT WORKSTATION FOR SYSTEM IDENTIFICATION

Marc HAEST, Georges BASTIN, Michel GEVERS and Vincent WERTZ
Laboratoire d'Automatique, Dynamique et Analyse des Systèmes
Louvain University, Bâtiment Maxwell,
Place du Levant, 3
1348 LOUVAIN-LA-NEUVE, BELGIUM

Paper n° 582

Abstract. A Workstation, based on a SUN computer, specifically designed for System Identification is presented. The station is extremely user-friendly since all the manipulations and interactions with the user are performed through a mouse-windows oriented dialogue. All the complex tasks of identification are available, including validation tools, while graphical results are displayed at the time they are obtained. An automatic archivage of the session is performed allowing the user to obtain a summary of his work under the form of a well organized and easy to read printed copy of the results or to re-enter an old session that was temporarily stopped. Finally, it is possible to ask the station to classify the models from the best one to the worse according to a specified list of criteria, leading to the implementation of an Expert System to which the user can pass the deal.

Keywords. Artificial intelligence; computer applications; identification; modelling; parameter estimation; system analysis.

INTRODUCTION

In this paper, we describe in detail a **Workstation** (based on a SUN computer) that we have specifically designed for System Identification. The station provides tools that allow one to establish a dialogue between the user and the machine in terms of a window menus-mouse actions exchange. It was therefore a very convenient way to give the expert a particularly user-friendly environment.

The station can handle single output multiple input systems. At the beginning of an identification session, the user must provide the station with at most two output-inputs data files : the first for parameter estimation and the second, if any, for model validation. These data sets will be called, respectively, the estimation data set and the validation data set. They are stored once and for all in a vector after the necessary space has been allocated, so that no further readings are done in the sequel of the session, resulting in rather substantial savings in time. The characteristics of the data files (like number of samples, means and variances of the inputs and output, ...) are permanently shown in a window called **Data Characteristics**. The station may also be connected to a data processing package that includes sampling, smoothing, substraction of polynomial drifts, rejection of outliers and the like.

AN IDENTIFICATION SESSION

The standard configuration of the station during an identification session, shown in Fig. 1, is composed of three different types of windows. In windows of the first type the user can specify options under which tools activated in the second type window will be run and whose results will appear in windows of the third type. Note that you can move, resize, expose or hide windows so that the basic window configuration can be modified at leasure.

ADMISSIBLE MODELS AND PARAMETER ESTIMATION METHODS

The type 1 windows are **Model Structure Selection** and **Parametric Estimation Method**. The structure of the model under investigation is chosen by the user in the former. The station allows the identification of MISO discrete linear systems of the following form:

$$A(z^{-1})\, y(t) = \sum_{i=1}^{m} z^{-\tau_i} B_i(z^{-1})\, u_i(t) + \frac{C(z^{-1})}{D(z^{-1})}\, e(t) + CC$$

where A, B_i, C and D are polynomials in the shift operator z^{-1} with A,C and D monic, τ_i are the dead times, m is the number of inputs, e(t) is a noise driving term and CC is the offset term.

Parametric estimation methods are selected in the latter window, with the values of their associated design parameters (i.e. initial conditions, forgetting factor, gain matrix resetting frequency, bounds on the trace of the gain matrix, ...). The following methods are available: off-line least squares, recursive least squares, generalized least squares, recursive maximum likelihood, output error method.

TOOLS

In the **Tool Box** window, the only window of the second type, a number of tools from the Fortran 77 SYSID package, a System Identification package developed in our laboratory, can be activated (in the sequel, "prediction errors in estimation" refers to prediction errors computed on the estimation data set, while "prediction errors in validation" means prediction errors computed on the validation data set):

- run a parametric estimation,

- compute mean and variance of the prediction errors in estimation or in validation,

- compute Akaike's order tests (Akaike 1978, 1980; Rissanen 1978; Schwarz 1978):

$$FPE = \frac{N + \dim(\theta)}{N - \dim(\theta)}\, \sigma^2$$

$$AIC = N \ln(\sigma^2) + 2 \dim(\theta)$$

and $$BIC = N \ln(\sigma^2) + \ln(N) \dim(\theta)$$

where N is the number of samples, $\dim(\theta)$ is the number of parameters in the model under investigation and σ^2 is the variance of the prediction errors in estimation or in validation,

- test the mean of the prediction errors in estimation or in validation for nullity, i.e. test if

$$-t_{\alpha/2}(N-1) \le \frac{\mu}{\sigma/\sqrt{N}} \le t_{\alpha/2}(N-1)$$

where N is the number of prediction errors, μ and σ^2 are estimators for their mean and variance and $t_\alpha(N)$ is the α level of the Student distribution with N degrees of freedom,

- test the means of the prediction errors in estimation and validation for equality, i.e. test if

$$-t_{\alpha/2}(N_e+N_v-2) \le \frac{\mu_e - \mu_v}{\sqrt{\frac{\sigma_e^2}{N_e} + \frac{\sigma_v^2}{N_v}}} \le t_{\alpha/2}(N_e+N_v-2)$$

where μ_e and μ_v are estimators for the mean of the prediction errors in estimation and validation, respectively, and σ_e^2 and σ_v^2 are estimators of their variances,

- test that the variance of the prediction errors in validation is only a small constant times the variance of the prediction errors in estimation, i.e. test $\sigma_v^2 = k\,\sigma_e^2$ against $\sigma_v^2 > k\,\sigma_e^2$ by checking the following inequality

$$\frac{\sigma_v^2}{k\,\sigma_e^2} < F_\alpha(N_v-1, N_e-1)$$

where $F_\alpha(N_1, N_2)$ is the α level of the F distribution with N_1 and N_2 degrees of freedom,

- test the prediction errors in estimation or in validation for whiteness. First, if the prediction errors behave like a purely random process, then the number of changes of sign in the sequence of the prediction errors is normally distributed with mean N/2 and variance N/4 (Söderström 1987). Next, the autocorrelation coefficients $\rho_{\varepsilon\varepsilon}(k)$ of the prediction errors are asymptotically normally distributed with zero mean and variance 1/N (Priestley 1981). As those coefficients should be zero for $k > 0$, the α levels of confidence on them are given by

$$\pm \frac{N_{\alpha/2}}{\sqrt{N}}$$

where N_α stands for the α level of the normal distribution with zero mean and unit variance. We can thus count the number of coefficients which are outside this gap and divide by the total number of computed coefficients. If more than α % of them lie outside the critical range, we should normally reject the hypothesis that the prediction errors are white. Finally, if the prediction errors are white, then the following quantity should be asymptotically $\chi^2(M)$ distributed (Ljung 1987)

$$\zeta_{N,M} = N \sum_{k=1}^{M} \rho_{\varepsilon\varepsilon}^2(k)$$

where N stands for the number of data and M stands for the number of estimated autocorrelation coefficients. Independence between the prediction errors can thus be tested also by checking if

$$\zeta_{N,M} < \chi_\alpha^2(M)$$

where $\chi_\alpha^2(M)$ is the α level of the χ^2 distribution with M degrees of freedom,

- test the independence between prediction errors in estimation or in validation and past inputs. This is usually checked using estimates of their crosscorrelation coefficients $\rho_{\varepsilon u}(k)$. If the prediction errors are independent of past inputs, then (Ljung 1987)

$$\sqrt{N}\,\rho_{\varepsilon u}(k)$$

chaves asymptotically as a normal distribution with zero mean and variance

$$P = \sum_{k=-M}^{M} \rho_{\varepsilon\varepsilon}(k)\,\rho_{uu}(k)$$

where M stands for the number of computed coefficients. If N_α denotes the α level of the normal distribution with zero mean and unit variance, we can thus check if

$$|\rho_{\varepsilon u}(k)| \le \sqrt{\frac{P}{N}}\,N_\alpha$$

If not, the hypothesis that the prediction error at time t and input at time t-k are independent should be rejected,

- compute the zeros of the polynomials A, B_i, C and D (Jenkins 1975),

- compute the unit step-responses, the static gain, the settling time and the overshoot of the estimated model,

- compute simulated outputs of the estimated model using inputs only.

Each of these tools can be activated at any time during the identification session by just clicking on the button at the left of the corresponding tool label. They are performed in the chronological order of their selection. As will be seen, the software is designed so as to avoid repeating the same computations several times.

It is also possible to activate several tools at the same time by clicking on the button **Make a Study** in the **Tool Box** window. Those tools that will be run are selected through hidden menus associated with each of the tool labels. Doing this enables the user to define a standard combination of tools that will be applied to every case study by specifying only the tools that are relevant to him. If further informations are needed later, they can be obtained by clicking only on the buttons corresponding to the tools that are missing.

REPRESENTATION OF NUMERICAL AND GRAPHICAL RESULTS

All results are displayed automatically at the time they are computed. When a model structure and a parameter estimation method have been specified in windows **Model Structure Selection** and **Parametric Estimation Method**, all the results of the tools that have already been activated for this particular model structure-parameter estimation method combination can be obtained by clicking on the button **Available Results** in the **Tool Box** window. Some of the results, like the estimated parameters and their confidence levels, are presented only in a numerical manner. Others, like the test for whiteness of the prediction errors and unit step-responses, give rise both to numerical and graphical representation. The correlogram of the prediction errors and the cross-correlogram between the prediction errors and past inputs are displayed with their corresponding levels of confidence. Pole-zero

maps, unit step-responses and comparison between simulation results and the true data are also available on a graphical mode.

THE DATA BASE

All the identification results are collected at the time they are obtained in a data base organized as a filing box and set up using the concept of structure in the C language. Each card contains the already computed results for one particular model structure-parameter estimation method combination. New identification cards are lexicographically inserted in the data base or appended at its end each time a new model is proposed for investigation. So, the code knows at any time everything that has already been done by the user. If he or she asks for an old result, perhaps without remembering that it has already been computed, it can be restituted with no effort. Thus, we not only do the computations very rapidly, but also, old results are recovered at the price of only one search through the data base. It must be noticed that missing intermediate results are automatically computed and inserted in the data base when a complex tool needing unavailable results is activated.

A complete set of facilities is at the user's disposal in the **Data Base** subwindow to consult the data base and compare cards. One can read the cards chronologically, i.e. in the order they were obtained, or lexicographically as they are sorted in the data base, i.e. for increasing model complexity.

The user can erase cards from the data base or flag some of them that are considered definitively uninteresting ones. This enables one to remember that these cases have already been investigated while destroying their associated results at the same time

SCORER

Moreover, one can ask the code to classify the models from the best one to the worse one according to a specified list of criteria, some sort of preference ordering which is a typical problem of multiple criterion optimization theory. An intelligent summary of the best results already obtained during the session can be obtained in subwindow **Scorer**, including a graphical illustration of the decrease of the variance of the prediction errors in estimation or validation with the dimension of the parameter vector and a selection of the best models according to the most important validation tools.

For example, it is possible to obtain the model that gives the best variance of the prediction errors in estimation or validation, the model that leads to the best FPE, AIC or BIC or the one for which the test on the prediction error whiteness gives the best results.

SUMMARY AND RECORDING OF THE SESSION

The data base also allows one to get a well organized and easy to read printed copy of the results at the end of the session. When exiting, it is also possible to store in a particular file the current state of the study in a condensed form in order to initialize the restarting of a session which has been interrupted. To save space, some of the results are erased when saving the session such as the computed sequences of autocorrelation coefficients of the prediction errors that are used when testing them for whiteness. The results about the whiteness are retained. This means that these coefficients will have to be computed again only if it is desired to obtain a correlogram in a later study.

THE EXPERT SYSTEM

These classification abilities led to the implementation of an Expert System to which the user can pass the deal. This can be done by just clicking the corresponding button. The OPS83 Expert System that we have used to test our search algorithm consists of two components, a collection of If-Then rules and a global data base called working memory (Forgy 1986). Each rule contains a conditional expression called the rule's LHS (for Left Hand Side) and an unconditional sequence of actions called the rule's RHS (Right Hand Side). A LHS in turn consists of one or more patterns. A LHS is considered satisfied when every pattern in the LHS matches an element from working memory.

The rule interpreter executes a production system by performing a sequence of operations called the recognize-act cycle. The standard recognize-act cycle is:

1. Match: Evaluate the LHSs of the rules to determine which are satisfied given the current contents of the working memory,

2. Conflict Resolution: Select one rule from among the ones with satisfied LHSs. If no rules have satisfied LHSs, halt execution,

3. Act: Perform the operations specified in the RHS of the selected rule,

4. Go to step 1.

The changes made during the Act phase of the cycle generally result in a new set of LHSs being satisfied on the next cycle, and it is this that gives direction and continuity to a system's processing. That is, the typical sequence of events that is seen as a system runs is as follows: some rule makes a change to working memory, another rule becomes satisfied as a result of that change, and on the next cycle it is selected and allowed to make further changes. The changes made by the second rule cause a third rule to become satisfied and able to execute.

Processing continues in this manner, each rule responding to the changes made by its predecessors, making changes of its own in an attempt to drive the system closer to the solution it is seeking. The rules that we have implemented are described hereafter.

In the sequel, we will designate models by using the (n, m, d) notation where n and m stand for the number of coefficients in the polynomials A and B_1, i.e. the number of autoregressive and exogeneous parameters respectively, and d stands for the delay. The dimension of a particular model is thus given by n + m. Note also that, for the sake of simplicity, we hereafter consider only models with one input.

Delay detection: First of all, a first guess for the delay is obtained by trying several models of dimension 2, (1, 1, d), with d varying from 0 to 20. The one for which the variance of the prediction errors is minimum becomes the initial model for the search algorithm that is applied there after.

The search for the best model is performed by detecting an elbow in the graph between the variances of the prediction errors and the model dimension (Ljung 1987). As long as there is no elbow, a fast search is performed for models of increasing complexity. When an elbow is detected, a slow search is made around the best model to ascertain its adequation.

Fast search: In this method, a search for the best model in the sense of the minimization of the variance of the prediction errors in estimation is made through the set of all models with the same dimension. When that best model has been obtained, the dimension of the models under investigation is increased by one.

At constant dimension and if no elbow has been detected in the graph of the variance of the prediction errors versus the dimension of the models, a fast search is made where we apply the following rules sequentially :

- decrease the delay of the current model by one,

- add one pole and substract one zero to the current model,
- add one zero and substract one pole to the current model,

- increase the delay of the current model by one,

This is done until a model with better prediction error variance is encountered. Each time this happens, the sequence is interrupted, the new model becomes the current one and the rules are restarted immediately. If the four rules are applied without giving a better model, one pole is added to the current best model and the same search is restarted through the set of models with dimension increased by one.

Slow search: If an elbow has been detected in the relation between the variance of the prediction errors and the dimension of the models, i.e. if a model dimension such that an increase of the number of parameters results in an insignificant decrease of the prediction error variances is discovered, a slow search is peformed beginning with the best model obtained for this dimension and for the best model in the next available dimension.

This slow search is made by examining, this time, all the neighbouring models of the current one at constant dimension, i.e. those that can be obtained by modifying n, m and/or d by one at a time. Here, all the neighbours are computed before examining if one of them gives rise to a reduction of the variance of the prediction errors.

Elbow detection: Practically, the fast search is applied until an elbow in the graph of the variance of the prediction errors versus the dimension of the models is obtained. Note that this elbow must be observed between model dimensions differing at least by two to allow the insertion of two complex conjugate poles in the polynomial A. To avoid problems arising when the variance of the prediction errors decreases very slowly for low model dimensions, one does not try to detect an elbow if the model dimension is less than 4.

Adequacy of the sampling period: In what concerns the adequacy of the sampling period, the following rules are applied each time a fast search at constant dimension has been achieved. If, for the current sampling period, a dominant pole with module comprised between 0.9 and 1.0 has already been detected for at least two model dimensions, or if the time response of the system is less than one sampling period, or this time response is greater than 20 sampling periods, or if there is no elbow in the relation between the variance of the prediction errors and the model dimensions, or if all the exogeneous parameters are less than their standard deviations, then the current sampling period is consirered to be unadapted.

Increasing the sampling period clearly results in a decrease of the number of samples. A rule checks that enough data remain in the file to perform further estimations.

On the other hand, the process will be considered to contain an integrator if there is always one pole near the unit circle after 4 modifications of the sampling period. The data can then be replaced by their increments.

Finally, if the sampling period remains unadapted after 4 such modifications while the system cannot be considered to contain an integrator, a rule concludes to the failure of the study.

Model validation: To measure the quality of a particular model with respect to the validation tools, a Quality Index is attached to them which is incremented by one each time the model satisfies one of the following criteria:

- less than 20% of the autocorrelation coefficients of the prediction errors are rejected and the first four are accepted at the 5% level of confidence,

- the variance of the prediction errors in validation is only a small constant times the variance of the prediction errors in estimation. This constant is tipically chosen as 1.5,

- the model is such that an increase of the number of parameters results in an insignificant decrease of the prediction error variance (an elbow has been detected),

- the BIC of the model is minimal (among all the models already detected),

- the settling time is between 5 and 20 sampling periods,

Thus this Quality Index can take values between 0 and 5, with 5 indicating the best models. At the end of the study, a table is constructed that gives the best model obtained for each possible value of the Quality Index. The principle of parcimony is applied between two models with the same Quality Index: the one with the least number of parameters is prefered. If more than one model remains with the same Quality Index and the same number of parameters, the one which gives the smallest prediction error variance is retained.

The rules and procedure developed in our Expert System are of course subjective, but they are the result of many years of experience on both simulated and real-life applications of identification. One advantage of the Expert System approach is that rules and decision parameters (such as confidence levels) can be changed very easily. Once the Expert System button has been activated, the only thing the user has to do is to observe the modifications the Expert System running in the background brings to the window configuration.

CONCLUSION

Perhaps we haven't put enough emphasis on the advantages the Workstation offers to the user who has to make a compromise between the dimension of the model and its performances. The trade-off between parsimony and flexibility is at the heart of the identification problem. How shall we obtain a good fit to the data with few parameters?

The answer is usually to use as much a priori knowledge about the system as possible, intuition and ingenuity. One can often hear that identification can hardly be brought into a fully automated procedure because of those facts. Also, it is difficult to rely only on objective cost functions: a general feeling of the adequacy of the solution must be gained by inspecting graphical results. A common-sense analysis of such plots is very often the last step to be passed before accepting a model. With a dedicated Workstation giving those possibilities in a particularly user-friendly environment, an identification exercice can now be achieved at a speed that was unheard of before.

REFERENCES

Akaike, H. (1978). On newer statistical approaches to parameter estimation and structure determination. 7-th Triennal WorldCongress of the IFAC, Helsinki, 1877-1884.

Akaike, H. (1980). Modern development of statistical methods. Trends and Progress in System Identification, P. Eykhoff ed.,169-184.

Forgy, C. L. (1986). The OPS83 User's Manual. Production Systems Technologies, Inc.

Jenkins, M. A. (1975). Algorithm 493: Zeros of a Real Polynomial. ACM Transactions on Mathematical Software, Vol. 1, N° 2, 178-189.

Ljung, L. (1987). System Identification: Theory for the User. Prentice-Hall.

Priestley, M. B. (1981), Spectral Analysis and Times Series, Academic Press.

Rissanen, J. (1978). Modeling by shortest data description. Automatica, Vol. 14, 465-571.

Schwarz, G. (1978). Estimating the dimension of a model. The Annals of Statistics, Vol. 6, N° 2, 461-464.

Söderström, T. (1987). Model structure determination. Encyclopedia of Systems and Control, Pergamon Press, 2287-2293.

Fig. 1 : The standard configuration of the station during an identification session.