

# ESPION: an Expert System for System Identification†

MARC HAEST,‡ GEORGES BASTIN,‡ MICHEL GEVERS‡ and  
VINCENT WERTZ‡

*Artificial intelligence techniques are used to implement an expert system that mimics human behaviour during an identification exercise.*

**Key Words**—Artificial intelligence; computer applications; heuristic programming; system identification.

**Abstract**—An expert system for the identification of linear multiple input single output systems in ARARX form, written with the OPS83 rule-based programming language, is presented. Provided with a data set, the expert system will organize an intelligent search through the set of candidate structures, and end up with a “best” model according to a “quality index” that incorporates a number of validation criteria. Tests on both industrial and simulated data have shown that the expert system behaves as well as human experts, with considerable savings in time.

## 1. INTRODUCTION

ALMOST any book, paper or lecture on identification of linear systems starts with the by now classical block-diagram of the *identification loop*, which in words can be paraphrased as follows.

- (1) start from a model structure that is consistent with prior knowledge and data,
- (2) estimate its parameters using an estimation criterion and a suitable algorithm,
- (3) validate the estimated model, possibly using other data,
- (4) if unhappy, try another model structure and repeat from step 2, until a “good enough” or “best” model is obtained.

An extensive body of literature and a coherent theory is now available about steps 1–3 above (see, e.g. Ljung (1987)). Properties of model structures, of estimation criteria and of algorithms are well understood and they can even be shaped to suit the objective for which the identified model is to be used. A battery of validation criteria are at the user’s disposal to

decide whether his or her estimated model is “good enough” or “better than all other estimated ones”. However, there is surprisingly no result available to guide the user in step 4 which, in a typical identification exercise, will have to be repeated dozens of times. Every user slowly builds up his or her own set of rules, based on heuristic experience.

A model structure is specified by a set of integer numbers. For example, the structure of an autoregressive model with a deterministic input (ARX model) is specified by the triplet of integers  $n_a$ ,  $n_b$  and  $\tau$  defining the orders of the polynomials  $A(z^{-1})$  and  $B(z^{-1})$  and the delay. For a single input single output system, and with a search restricted to ARX model structures, the identification loop described above can then be thought of as a way of travelling through the points of the space  $(Z_+)^3$  until a triplet  $n_a^*$ ,  $n_b^*$ ,  $\tau^*$  is obtained for which the corresponding ARX model is considered “good enough” or “best”. For more complicated model structures, the same is true with  $(Z_+)^3$  replaced by  $(Z_+)^q$ , where  $q$  is the number of integers defining the structure.

There are several ways of handling this search through the space  $(Z_+)^q$ . If an upper bound on the integers can reasonably be assumed, one could use an “*exhaustive search*” approach, which means estimate the models for all structures the structure indices of which do not exceed the upper bound. However, this could lead to the investigation of a horrendous number of models. For example, an ARX model with 2 inputs, upper bounds 5 on the polynomial orders, and upper bounds 10 on the delays would require the estimation of 12 500 models. A second way would be to use an “*integer programming*” algorithm to minimize a model validation criterion over the set of structure

† Received 2 November 1988; received in final form 11 May 1989. This paper was not presented at any IFAC meeting. This paper was recommended for publication by Associate Editor A. Bagchi under the direction of Editor P. C. Parks.

‡ Laboratoire d’Automatique, de Dynamique et d’Analyse des Systèmes, Université Catholique de Louvain, Place du Levant 3, B-1348 Louvain-La-Neuve, Belgium.

indices in  $(Z_+)^q$ . While this approach cannot be rejected off hand, we believe that it is perhaps unreasonable to forego the insight that can be gained, after a number of iterations of the identification loop, from the already estimated models. Therefore, starting in 1986, we developed an “*expert systems*” approach, in which the travel path through  $(Z_+)^q$  is driven by a set of rules that attempt to mimic the way in which experienced identification buffs (hereafter called experts) proceed when they sit at their terminal and identify a model using an identification package such as MATLAB’s IDENTIFICATION TOOLBOX. Only experience will tell whether our approach is justified from a practical point of view but, whatever the outcome, it has already had the beneficial effect of forcing us, the experts, to rationalize, justify and sometimes put in question our own rules for selecting model structures.

In this paper we shall describe the present state of our expert system called ESPION for “Expert System for Process IdentificatiON”, a first version of which was presented at the First IFAC Workshop on Artificial Intelligence in Real-time Control (Haest *et al.* 1988). One of our main requirements was that the expert system should be able to run without any human intervention; once the data are fed into the system, it starts its voyage into  $(Z_+)^q$ , driven by the data and the rules, until it encounters a final stopping rule. In the meantime we can work on other problems or drink capuccino. So far, it is limited to multiple input single output systems with delays modelled in ARARX, also called GLS, structure (see Section 3). Its main two ingredients are rules that determine the search path through  $(Z_+)^q$  based on the already estimated models, and quality criteria that attach a score to each model.

A few other expert systems are being developed specifically for the purpose of system identification: the SEXI Expert System for Identification from the Institut National Polytechnique de Grenoble (Gentil and Conraux, 1985), the Knowledge Database for System Identification developed by Larsson and Persson (1987, 1988) at Lund Institute of Technology, or that of Monsion *et al.* (1988) in Bordeaux. But, to the best of our knowledge, ESPION is the only fully data-driven expert system that can handle multiple input single output discrete linear models.

Our paper is organized as follows. Section 2 describes in a very schematic way the overall behaviour of the expert system. In Section 3 we describe the class of model structures our expert

system is presently able to handle and we introduce notations and definitions that will be used all through the paper. ESPION is organized around a number of goals, each of which is achieved via a set of rules. These goals, and the rules used to achieve them, are described in some detail in Section 4, which constitutes the core of our paper. Section 5 gives some practical details about the implementation of the working memory and the activation of the rules, while in Section 6 we illustrate the typical performances of ESPION on both simulated and industrial data. As has by now become standard practice, we have put our conclusions at the end.

## 2. OVERALL BEHAVIOUR OF THE EXPERT SYSTEM

We consider the situation where the data have already been collected at a sufficiently high rate, i.e. much larger than the system bandwidth. The task of the identifier is then to find a “best” or “good enough” model, at a reasonable sampling period. By reasonable sampling period we mean that it should be coherent with the dynamics of the system, and that it may therefore be necessary to choose a model sampling period that is larger than the period with which the data were collected.

To perform this task, our expert system is provided with the following.

- (a) An efficient sampling device (including appropriate filters) to undersample the data set without loss of significant information.
- (b) A set of admissible model structures which, at this moment, is the set of ARARX models. This set is henceforth called the *model set*.

Then the overall behaviour of ESPION can be roughly summarized by the three-step loop shown in Fig. 1.

### 2.1. Parsimonious walk through the model set

The aim of this first step is obviously to avoid examining and comparing all the models in the

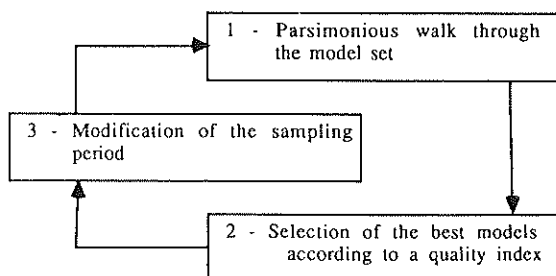


FIG. 1. Overall behaviour of the expert system.

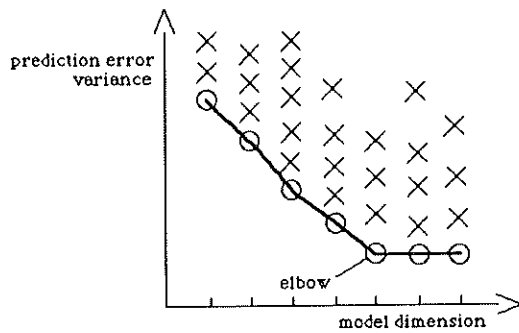


FIG. 2. A walk is done until an elbow is detected.

model set as would be done in an exhaustive search. The principle of the walk is to start with a model structure of small dimension, i.e. a model structure with few parameters, the delays of which have been roughly estimated. Then the dimension is progressively increased, while minimizing the number of investigated model structures at each dimension value. These investigations imply the estimation of the model parameters by using a standard prediction error method. The walk in the model set is stopped when a significant *elbow* is detected in the graph representing the prediction error variance with respect to the model dimension. The principle of the elbow detection is illustrated in Fig. 2 where estimated models are indicated by crosses except the best one at each model dimension, which is represented by a circle.

### 2.2. Selection of the best models according to a quality index

Once the first step has been completed, the expert system is left with a set of presumably good models at a specific sampling period. The next task is to select the best one(s). To do that, we introduce the concept of *quality index* of a model. The quality index is an intuitive, though technically well specified criterion (see Section 4.5). It is designed to reflect, in a single concise expression, the best trade off among the multitude of both empirical and theoretically founded *validation tools* which are commonly used by identification practitioners to compare and assess models.

### 2.3. Modification of the sampling period

When using shift-operator discrete time models for designing predictors or controllers, it is well known that the choice of the sampling period may be critical. This choice is therefore a relevant part of the system identification problem and, hence, an important step in our expert system. There is usually no universally accepted theory for the optimal selection of sampling rates, but a number of "rules of

thumb", from engineering experience, are available which constitute the basis of the expertise in ESPION. If these rules are found to be violated, the sampling period is increased and the data set is filtered appropriately. Steps 1 and 2 are then repeated on the new data set until the sampling period is found to be reasonable or until the resulting number of data points is found to be too small.

### 3. ADMISSIBLE MODELS AND NOTATIONS

The expert system allows the identification of multiple input single output discrete linear ARARX models (for AutoRegressive models with eXogeneous inputs and an AutoRegressive noise model). They are of the following form:

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

where polynomials in the shift operator  $z^{-1}$  are in capital letters, with  $A(z^{-1})$  and  $D(z^{-1})$  monic, the  $\tau_i$  are the delays of the system,  $m$  the number of inputs,  $e(t)$  a noise driving term and offset stands for a constant offset term. The  $A(z^{-1})$  and  $B_i(z^{-1})$  polynomials are sometimes called the deterministic polynomials while  $D(z^{-1})$  is often referred to as the noise polynomial. Note that the widely used ARX models are a subset of this model set.

A standard off-line prediction error method is used for parameter estimation so that prediction errors are defined as follows:

$$e(t) = y(t) - \hat{y}(t)$$

where predicted outputs are given by the following expression:

$$\hat{y}(t) = y(t) - A(z^{-1})[D(z^{-1})y(t)] + \sum_{i=1}^m z^{-\tau_i} B_i(z^{-1}) \times [D(z^{-1})u_i(t)] + D(z^{-1}) \text{offset.}$$

Note that a data file divider mechanism is used which separates the original data set into two different subsets, one for parameter estimation and the other for model validation, so that prediction errors may be computed either on the first, called the estimation data set, or on the second, called the validation data set.

The following notations will be used in the sequel:

$n_a$ : number of coefficients in the autoregressive polynomial  $A(z^{-1})$ ,

$n_{b_i}$ : number of coefficients in the exogeneous polynomial  $B_i(z^{-1})$ ,

$n_d$ : number of coefficients in the noise polynomial  $D(z^{-1})$ ,

$n_0$ : an integer set to 1 or 0 depending on the presence or absence of an offset term,

$\dim(\theta)$ : model dimension or number of parameters in the model

$$\dim(\theta) = n_a + \sum_{i=1}^m n_{bi} + n_d + n_0,$$

$N_e$ : number of data samples used for parameter estimation,

$N_v$ : number of data samples used for model validation,

$\sigma_e^2$ : experimental variance of the prediction errors computed on the estimation data set, hereafter called estimation variance

$$\sigma_e^2 = \frac{1}{N_e} \sum_{t=1}^{N_e} e^2(t),$$

$\sigma_v^2$ : the experimental variance of the prediction errors computed on the validation data set, hereafter called validation variance

$$\sigma_v^2 = \frac{1}{N_v} \sum_{t=1}^{N_v} e^2(t),$$

$t_{si}$ : model settling time with respect to input  $i$  expressed as a multiple of the sampling period (we call settling time the time between the start of the response to a step and the time the response settles to within 5% of its steady-state value).

Many of the rules which will be described in the following sections involve the modification of the structure (delays, polynomial orders, presence or absence of an offset term) of the models and the computation of the estimation variance of the ensuing model. Two important remarks are in order here.

First, each time a new model structure is investigated the following quantities are computed and stored in a global data base: parameter estimates with their confidence intervals, the estimation variance, static gains, poles and settling times. Hence, those characteristics are recovered with no effort if they have already been computed in previous steps: the same computation is never performed twice.

The second remark is that some rules are necessary to check the admissibility of new structures. An admissible model is trivially defined as follows:

$$\begin{aligned} \tau_i &\geq 0 \quad \forall i \\ n_a &\geq 1 \\ n_{bi} &\geq 1 \quad \forall i \\ n_d &\geq 0 \\ n_0 &= 0 \text{ or } 1. \end{aligned}$$

Those are the present default admissibility rules, but the user can modify them. Only admissible structures are considered in the structure related operations described in the following sections. With the present rules, ESPION decides whether an offset term and/or a noise polynomial should be used or not. However, the user can prohibit the use of those terms. He or she can also specify the signs of some of the static gains known in advance so as to avoid the unnecessary analysis and validation of models estimated with the wrong signs.

#### 4. THE RULES

As already explained (Fig. 1), we can distinguish between two categories of rules: those related to the walk through the space of admissible structures and those related to the analysis of the results, namely the validation of the estimated models and of the sampling period.

Inside these two categories, the behaviour of human experts has been further split into several more or less hierarchically related goals, each set of rules tending towards one particular goal. The main goals that are pursued during an expertise are now described with the rules that were implemented in order to achieve them. They are depicted in Fig. 3, in which the three steps of Fig. 1 are expanded.

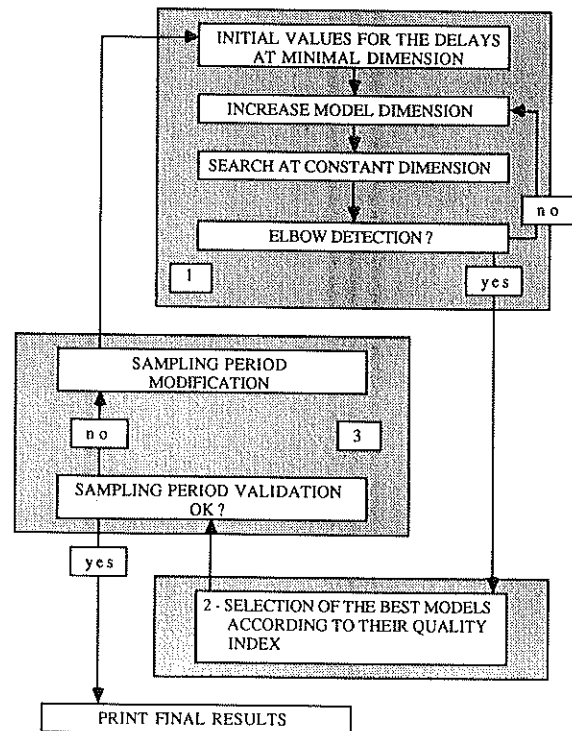


FIG. 3. The main goals of the expertise.

#### 4.1. Initial values for the delays at minimal dimension

*Operations.* Two different (and obvious) operations can be performed on the delays:

$D_1$ :  $\tau_i \rightarrow \tau_i + 1$  (increase the delay of some input  $i$ ),

$D_2$ :  $\tau_i \rightarrow \tau_i - 1$  (decrease the delay of some input  $i$ ).

*Expertise rules.* The procedure of initializing the delays is performed at minimal dimension ( $n_a = 1$ ,  $n_{bi} = 1 \forall i$  and  $n_d = n_0 = 0$ ). Two initial candidate structures are defined as follows. The cross-correlation coefficients between the output and each of the inputs are computed up to lag  $\tau_{\max}$ , where  $\tau_{\max}$  is typically set to 20. Denote by  $\tau_{if}$  the lag of the first cross-correlation coefficient outside the 95% confidence interval around zero, and by  $\tau_{im}$  the lag of the cross-correlation coefficient with maximum absolute value, both for input  $i$ . The two candidate structures are then

$$\tau_i = \tau_{if} \quad \forall i$$

and

$$\tau_i = \tau_{im} \quad \forall i.$$

For each of these initial structures, the following procedure is applied.

(a) Compute their estimation variance and that of all their neighbouring structures (structures obtained from the initial one by applying operations  $D_1$  or  $D_2$  to one or more inputs as shown in the example below).

(b) Choose the structure with the smallest estimation variance to replace the initial structure.

(c) Back to step (a), until the structure has no better neighbours.

To illustrate the concept of neighbouring structures, we take as an example a two input process with initial delay structure  $\tau_1 = 2$  and  $\tau_2 = 5$ . Then the neighbouring delay structures are given hereafter, the original one being in bold characters:

$$(\tau_1 = 1, \tau_2 = 6)(\tau_1 = 2, \tau_2 = 6)(\tau_1 = 3, \tau_2 = 6)$$

$$(\tau_1 = 1, \tau_2 = 5)(\tau_1 = \mathbf{2}, \tau_2 = \mathbf{5})(\tau_1 = 3, \tau_2 = 5)$$

$$(\tau_1 = 1, \tau_2 = 4)(\tau_1 = 2, \tau_2 = 4)(\tau_1 = 3, \tau_2 = 4).$$

It should be stressed that, while this procedure is expected to provide good estimates for the delay structure of the system, it does not mean that these delays will not change in the sequel, since they are only initial estimates.

#### 4.2. Increase the model dimension; elbow detection

At the end of the delay estimation step at minimal dimension or at the end of a search at constant dimension (see Section 4.3), one is left with the model with the smallest estimation variance in this dimension. The expert system will then check whether increasing this dimension can provide a better model.

*Operations.* Only one operation is used in order to increase the model dimension.

$$ID_1: n_a \rightarrow n_a + 1 \text{ (add one pole).}$$

*Expertise rules.* The real expertise in this task is to decide when to stop increasing the model dimension. The idea of the procedure that has been implemented is to detect an elbow in a graph where estimation variances are plotted vs model dimension.

The elbow must be observed for two successive dimensions, to allow the insertion of two complex conjugate poles in the autoregressive polynomial. The procedure is as follows.

(a) When the best model at dimension  $k$  has been obtained, compare the best estimation variances of the models of dimension  $k-2$ ,  $k-1$  and  $k$  using the following test procedure.

(b) If there is a significant decrease either between  $\sigma_e^2(k-2)$  and  $\sigma_e^2(k-1)$  or between  $\sigma_e^2(k-2)$  and  $\sigma_e^2(k)$ , increase the model dimension ( $ID_1$  on the best model at dimension  $k$ ). If an elbow has been detected at dimension  $k-2$ , the exploration is stopped and the validation tasks begin (see Sections 4.4 and 4.5).

To test whether the estimation variance is significantly reduced when the number of parameters is increased from  $n_1$  to  $n_2$ , the following test quantity is used:

$$F = \frac{\sigma_e^2(n_1) - \sigma_e^2(n_2)}{\sigma_e^2(n_2)} \frac{N_e - n_2}{n_2 - n_1}$$

which is asymptotically  $F$ -distributed with  $(n_2 - n_1)$  and  $(N_e - n_2)$  degrees of freedom provided the residuals are Gaussian (Ljung, 1987). Note that there is a maximal model dimension which is usually set to six times the number of polynomials in the model structure. If it is reached without an elbow being detected, the sampling period will be considered unadapted (see Section 4.5).

#### 4.3. Search at constant dimension

*Operations.* The following operations can be performed at constant dimension:

$$P_1: n_a \rightarrow n_a + 1, n_{bi} \rightarrow n_{bi} - 1 \text{ for some } i$$

$$P_2: n_a \rightarrow n_a + 1, n_d \rightarrow n_d - 1$$

$$P_3: n_a \rightarrow n_a + 1, n_0 \rightarrow n_0 - 1$$

- $I_1: n_{bi} \rightarrow n_{bi} + 1, n_a \rightarrow n_a - 1$  for some  $i$   
 $I_2: n_{bi} \rightarrow n_{bi} + 1, n_{bj} \rightarrow n_{bj} - 1$  for some  $i$  and  $j$ ,  
 $i \neq j$   
 $I_3: n_{bi} \rightarrow n_{bi} + 1, n_d \rightarrow n_d - 1$  for some  $i$   
 $I_4: n_{bi} \rightarrow n_{bi} + 1, n_0 \rightarrow n_0 - 1$  for some  $i$   
 $N_1: n_d \rightarrow n_d + 1, n_a \rightarrow n_a - 1$   
 $N_2: n_d \rightarrow n_d + 1, n_{bi} \rightarrow n_{bi} - 1$  for some  $i$   
 $O_1: n_0 \rightarrow n_0 + 1, n_a \rightarrow n_a - 1$   
 $O_2: n_0 \rightarrow n_0 + 1, n_{bi} \rightarrow n_{bi} - 1$  for some  $i$ .

*Expertise rules.* The aim of the search at constant model dimension is to find the structure with the smallest estimation variance as fast as possible, rather than performing an exhaustive search. The procedure for this search is as follows.

(a) Start with the structure generated by the previous step, namely when increasing the model dimension.

(b) Examine other structures using the operations described above, plus  $D_1$  and  $D_2$ , until a model with better estimation variance is encountered which is then retained as the new initial structure. The examination is performed in the following order (see example in Section 6.2, Fig. 5):

$O_1, O_2$  ( $i = 1, \dots, m$ ),  $N_1, N_2$  ( $i = 1, \dots, m$ ),  
 $D_2$  ( $i = 1, \dots, m$ ),  $P_1$  ( $i = 1, \dots, m$ ),  $I_1$  ( $i = 1, \dots, m$ ),  $I_2$  ( $i = 1, \dots, m$  ( $j = 1, \dots, m$ )),  $P_2, I_3$   
 $(i = 1, \dots, m)$ ,  $D_1$  ( $i = 1, \dots, m$ ),  $P_3, I_4$  ( $i = 1, \dots, m$ ).

Recall that only the admissible structures generated by this rule are considered.

(c) Back to (b), until no further improvement of the estimation variance can be obtained in the current dimension.

Although this procedure appears to be quite long, it will be shown with examples that at constant model dimension, the number of estimated models is far less than with an exhaustive search. Obviously, the order in which these operations are performed does have an effect upon the overall performance of the expert system: number of estimated models, discovery of the best structure, . . . In fact, here is where the bulk of the expertise resides: how to obtain a good fit to data at low cost? The present ordering results from the expertise of the authors and from tests that have been carried out on both industrial and simulated data. Nevertheless, due to our choice of expert system formulation and language, it is very easy for the user to modify this ordering in order to compare

results and derive conclusions on the optimal strategy.

#### 4.4. Selection of the best models according to their quality index

Up to this step, the only criterion used in the search for "best models" has been the estimation variance. Now the aim is to apply more complex validation tools to some of these models (the selection will be made precise later) in order to attach to each model a quality index and to produce as final result a *ranking* of the best models.

*Operations.* The following logical assertions, commonly referred to as validation tools in the literature, can be tested by the expert system for any estimated model:

- $V_1$ : the given model has minimal BIC,  
 $V_2$ : the given model has minimal  $\sigma_v^2$ ,  
 $V_3$ :  $\sigma_v^2$  does not differ significantly from  $\sigma_e^2$ ,  
 $V_4$ : the dimension of the model is the elbow dimension,  
 $V_5$ :  $t_{si} < t_{\max} \forall i, i = 1, \dots, m$  where  $t_{\max}$  is currently set to 20,  
 $V_6$ : the prediction errors in estimation are white,  
 $V_7$ : the prediction errors in validation are white,  
 $V_8$ : the prediction errors in estimation are independent from past samples of the input signal number  $i$  ( $i = 1, \dots, m$ ),  
 $V_9$ : the prediction errors in validation are independent from past samples of the input signal number  $i$  ( $i = 1, \dots, m$ ).

The precise meaning of these tests and the statistical tools used in these operations are described in the Appendix.

*Expertise rules.* ESPION performs a first selection of the models previously estimated by retaining only those models the estimation variances of which are less than the best estimation variance obtained in the preceding dimension. If prior information on static gains has been provided, models having at least one static gain with a wrong sign are also rejected. The quality index of each of these selected models is then incremented by one each time the model satisfies one of the logical tests defined above.

Once the quality indices have been computed, the models are hierarchically sorted with respect to the following criteria: decreasing values of the quality index, then increasing values of the model dimension (parsimony principle), finally increasing values of the estimation variance. This means that the quality index is the top selection criterion; if several models have the same

highest quality index, then the one with the smallest dimension is chosen; among models with the same highest quality index and the same lowest model dimension, the one with the smallest estimation variance is chosen.

#### 4.5. Sampling period validation and modification

*Operations.* No operations are needed when validating the sampling period, since all tools required are already available once the models have been estimated: standard deviations on the parameter estimates, model poles and settling times.

*Expertise rules.* A parameter is said to be inconsistent if the one  $\sigma$  interval around that parameter contains zero, where  $\sigma$  is the estimated standard deviation of that coefficient. A polynomial is called inconsistent if all its parameters are inconsistent.

The sampling period is modified if at least one of the facts given hereafter holds:

$T_1$ : no elbow has been detected,

$T_2$ : the exogeneous polynomials of the best model are all inconsistent,

$T_3$ : the best model has poor transient behaviour ( $t_{si} > t_{max}$  for some  $i$ ),

$T_4$ : the best model has at least one real pole within the interval [0.95; 1.05].

If  $T_3$  holds, the new sampling period  $T_c$  is chosen so that  $t_{si}$ , expressed as a multiple of the new  $T_c$ , is such that

$$t_{si} < t_{max} \quad \forall i.$$

If  $T_3$  does not hold while at least one of the other facts does, then  $T_c$  is simply incremented by one unity. The data are then undersampled and filtered and a model search is reinitiated with this new data set, beginning with the computation of  $\tau_{if}$  and  $\tau_{im}$  (Section 4.1).

Increasing the sampling period clearly results in a decrease of the number of samples. A rule checks that enough samples remain in the data sets to perform further estimations. The expertise is stopped if this is not the case or if the sampling period has been judged adapted ( $T_i$  false  $\forall i, i = 1, \dots, 4$ ).

The validation of the sampling period and its automatic modification is a delicate operation, because it is difficult to clearly distinguish between data that have been collected at an excessively fast rate, and data originating from a process that contains a pure integrator. The rules just mentioned are those that are presently being experimented, but more research is needed on this particular set of rules.

#### 5. IMPLEMENTATION ASPECTS

The expert system we have developed with the OPS83 rule-based programming language consists of two components: a collection of *IF-THEN* rules and a global data base called *working memory* (Forgy, 1986).

Elements of the working memory are used to store particular facts about the current state of knowledge and future goals. For example some elements of type *model* are used to store estimated model structures and their characteristics. Other elements of type *goal* are used to specify to the inference engine some tasks to be performed. The current model structure is also stored in an element the type of which is *current structure*, while the modifications that can be applied on it are stored in elements of type *structure modification*. In addition to its type, an element can contain field descriptors to further characterize it. For example, an element of type *goal* possesses a field descriptor called *status* and the goal is performed only if this field descriptor is set to *on*.

As already explained, the rules are goal oriented. One can consider that every set of rules is organized around a particular goal. Some goals that are difficult to achieve can be expressed as an ordered or unordered sequence of simpler ones. The rules corresponding to complex tasks, like the expertise rules described in Section 4, have been split into a collection of more rudimentary goals to which they make several uses following a scenario that will depend on the data at hand.

Each rule contains a conditional expression consisting of one or more *patterns*, followed by an unconditional sequence of *actions*. A particular rule is considered satisfied when every pattern in the conditional expression matches an element from the working memory.

The unconditional sequence of actions, in turn, can be viewed as the body of a function in a classical imperative programming language. Some actions specify modifications to be carried out on elements from the working memory. New elements may be created while old ones may be modified or deleted. Other actions include calls to Fortran tools from the SYSID package (a SYStem IDentification library developed in our laboratory). Besides a set of validation tools, this package contains two off-line parameter estimation algorithms: one off-line least-squares algorithm for the estimation of ARX models, and a generalized least-squares algorithm for the estimation of ARARX models.

For example, the syntax of a rule used during the search at constant dimension is given on the following page.

```

IF {
  —there is an element from working
  memory whose type is goal, whose name
  is search and whose status is on;
  —there is an element from working
  memory whose type is current structure;
  —there is an element from working
  memory whose type is structure modifica-
  tion and whose status is unused;
  —the modification of the current model
  structure does not produce an already
  computed admissible model;
}
THEN {
  —estimate the new model and store the
  results in a new working memory element
  of type model;
  —set the status of the element of type
  structure modification to used;
  —if {a decrease of the estimation
  variance has occurred}
    then {
      —replace the structure stored in
      the element current structure by
      the structure that has just been
      estimated;
      —set the status of all used
      structure modification elements
      to unused;
    }
}

```

So, if the *status* of one element of type *goal* the *name* of which is *search* is *on*, modifications of the current structure are examined as long as there exist unused elements of type *structure modification* in the working memory. Nevertheless, only modifications producing unknown admissible model structures are considered. Each time a new structure has been estimated, the modification that produced the new model is disabled by setting its *status* to *used*. However, when a better model is encountered, the process must be reinitialized by resetting the *status* of all used modifications.

What should be kept in mind here is that, even if the control exerted by the rules may seem to be excessive from the production system point of view, the expert system remains basically data driven. Through the search rules, the sampling period validation rules or those that detect an elbow, the behaviour of the expert system is highly sensitive to the characteristics of the data under investigation. If the general profile of the study remains the same from one data file to another, the contents of the working memory, the number of estimated models and

their peculiarities, the travel path through  $(Z_+)^q$ , the explored sampling periods and the final model selection will evolve in different ways.

## 6. PERFORMANCES

The expert system has been tested on a number of both industrial and simulated data sets. We illustrate here some typical behaviours.

### 6.1. A glass tube drawing bench

Run on industrial data from a glass tube drawing bench on which human experts had already worked (Wertz *et al.*, 1987), ESPION proved to behave as well as them and, in some cases, gave better results. For example, the expert system took only 20 min to identify an ARX model between drawing speed and glass tube diameter in this production line from a 2700 sample file. As can be seen from Fig. 4, which shows part of the data, a pseudo random binary signal with amplitude 5 (variance 25) has been added to the setpoint of the drawing speed, about  $123 \text{ m min}^{-1}$ .

First, the search was forcibly limited to ARX structures with a possible offset term. At the end of the search in dimension 9, an elbow was detected in dimension 7. Fifty four models had been estimated and 18 of them validated. The best model was found to be the structure chosen by the specialists, but in the final ranking ESPION also suggested some other very good candidates among which a best choice is not clear cut. The final decision remains to the user. Note that in the case of this particular one input one output system, an exhaustive search would have needed the estimation of 64 models if the delay had been known. An exhaustive search over all models with delays up to the correct one plus one would have required the estimation of 384 models.

An expertise on the same data set but with a possible denominator polynomial in the noise model took 1 h. It estimated 78 models and led

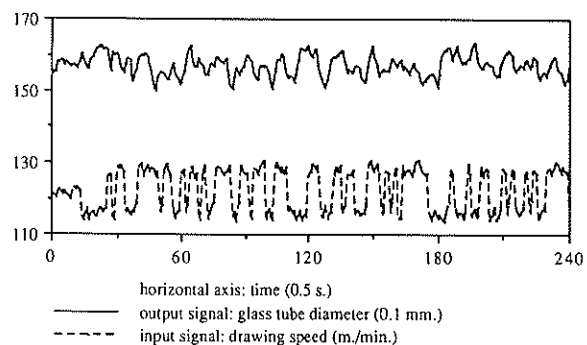


FIG. 4. Data from the glass tube drawing bench. Horizontal axis: time (0.5 s). Output signal (—): glass tube diameter (0.1 mm). Input signal (---): drawing speed ( $\text{m min}^{-1}$ ).



to the same final conclusions. The analysis and validation phases are time consuming, but here the increase in computing time is mainly due to the estimation method, which becomes an iterative one when ARARX models are estimated. Here, an exhaustive search with unknown delay would have required the estimation of 1224 models.

6.2. Simulated data

The expert system was also run on simulated data obtained from the following nine parameter, two input-one output system

$$(1 + z^{-1} + 0.5z^{-2})y(t) = z^{-2}(1 - z^{-1} + 0.5z^{-2})u_1(t) + z^{-7}(1 + 0.2z^{-1})u_2(t) + \frac{e(t)}{1 - 0.49z^{-2}}$$

where pseudo random binary signals, with amplitudes 1 and 2 (variances 1 and 4), respectively, were used for  $u_1$  and  $u_2$  and a Gaussian white noise with zero mean and variance 0.25 was taken for  $e$ , leading to an output signal variance of around 22.

It took 1 h to carry out the task on 1000 samples with no offset term. Table 1 summarizes the overall behaviour of the expert system for this particular case study. For each investigated model dimension, the starting and final model structures are given (in and out). Note the existence of two initial structures at minimal dimension 3 where initial estimates for the

delays are obtained. This is due to the fact that  $\tau_{1f} = 2$ ,  $\tau_{1m} = 4$  while  $\tau_{2f} = \tau_{2m} = 7$ . Each model structure is further characterized by its estimation variance  $\sigma_c^2$ . The number of estimated models, noted  $n$ , is also given for each model dimension.

Note that the initial estimates of the delays ( $\tau_1 = 3$ ,  $\tau_2 = 7$ ), obtained at minimal dimension, are only slightly modified in higher dimensions. Of course, this is a desirable property. In general, the number of estimated models tends to increase dramatically when the delays have not been approximated correctly. Such behaviour, where the delays are frequently changed must be considered pathological: when the system does not seem to know which way to turn, the sampling period may be suspected to be unadapted, but it may also happen that the system is nonlinear or nonstationary.

Figure 5 is a part of the script file that allows the user to track the expert system behaviour. It shows the models that were estimated during the search performed at dimension 8 in which a noise model has been constructed for the first time. The models are listed in the order of their investigation, with the following notation:

$$\text{model structure} = n_a(\tau_1 - n_{b1})(\tau_2 - n_{b2})n_d$$

First, a search is performed from the model built by adding a pole to the best model obtained in dimension 7. The first modification of this structure, i.e. adding a parameter in  $D(z^{-1})$  by deleting one in  $A(z^{-1})$ , leads to a better estimation variance. So the search is restarted

TABLE 1. OVERALL BEHAVIOUR OF THE EXPERT SYSTEM

dim ( $\theta$ )		$n_a$	$\tau_1$	$n_{b1}$	$\tau_2$	$n_{b2}$	$n_d$	$\sigma_c^2$	$n$
3	in	1	2	1	7	1	0	4.24121	16
	in	1	4	1	7	1	0	5.02561	
	out	1	3	1	7	1	0	3.60241	
4	in	2	3	1	7	1	0	1.69535	8
	out	2	2	1	7	1	0	1.63716	
5	in	3	2	1	7	1	0	1.63520	12
	out	2	2	2	7	1	0	0.75233	
6	in	3	2	2	7	1	0	0.70032	13
	out	2	2	3	7	1	0	0.44811	
7	in	3	2	3	7	1	0	0.39520	18
	out	2	2	3	7	2	0	0.35712	
8	in	3	2	3	7	2	0	0.35697	8
	out	2	2	3	7	1	2	0.33691	
9	in	3	2	3	7	1	2	0.28699	15
	out	2	2	3	7	2	2	0.26515	
10	in	3	2	3	7	2	2	0.26506	12
	out	2	2	3	7	2	3	0.26439	
11	in	3	2	3	7	2	3	0.26438	13
	out	3	2	3	7	2	3	0.26438	

```

criterion modification in dimension 7 : -0.09099
18 models estimated with Te = 1
best one : 2 (2-3) (7-2) 0 with criterion = 0.35712
no elbow
DIMENSION 8 - Te = 1
search from 3 (2-3) (7-2) 0 with criterion = 0.35697
2 (2-3) (7-2) 1 : 0.35652 better !
search from 2 (2-3) (7-2) 1
1 (2-3) (7-2) 2 : 0.79143 worse
2 (2-2) (7-2) 2 : 0.40685 worse
2 (2-3) (7-1) 2 : 0.33691 better !

search from 2 (2-3) (7-1) 2
1 (2-3) (7-1) 3 : 1.11489 failed
2 (2-2) (7-1) 3 : 0.70187 worse
2 (1-3) (7-1) 2 : 0.70475 failed
2 (2-3) (6-1) 2 : 3.42270 worse
3 (2-2) (7-1) 2 : 0.65752 failed
1 (2-4) (7-1) 2 : 1.13460 worse
3 (2-3) (7-1) 1 : 0.36280 failed
2 (2-4) (7-1) 1 : 0.41361 failed
2 (3-3) (7-1) 2 : 1.30408 failed
2 (2-3) (8-1) 2 : 4.31957 failed

criterion modification in dimension 8 : -0.02022
8 models estimated with Te = 1
best one : 2 (2-3) (7-1) 2 with criterion = 0.33691
no elbow
    
```

FIG. 5. The search at dimension 8.

from this latter structure. This time the estimation variance does not decrease until a second noise parameter is added by deleting a zero from  $B_2(z^{-1})$ . The search is then restarted once again but, now, with no success: all the admissible modifications give poorer results. At this stage, the expert system is ready to undertake a search in dimension 9. Note that structures for which the estimation algorithm fails to converge ("failed" warnings) are simply ignored.

At the end of the search in dimension 11, an elbow was detected in dimension 9. One hundred and fifteen models were estimated and 34 of them were validated. Here, an exhaustive search would have needed the estimation of 495 models assuming the delays were known while an exhaustive search over all models with delays up to the correct one plus one would have required the estimation of 17 820 models!

The ten best models are presented in Table 2 according to their quality index (QI). As expected, the structure with which the data were generated emerged much more clearly from the final selection than for the industrial data. Note how models in higher dimensions are penalized despite their lower estimation variances.

Another test has been made on simulated data obtained from the same system with the same inputs except that the real zeros of the denominator polynomial in the noise model were replaced by a pair of complex conjugate ones

$$(1 + z^{-1} + 0.5z^{-2})y(t) = z^{-2}(1 - z^{-1} + 0.5z^{-2})u_1(t) + z^{-7}(1 + 0.2z^{-1})u_2(t) + \frac{e(t)}{1 + 0.6z^{-1} + 0.18z^{-2}}$$

and that a Gaussian white noise with zero mean and variance 0.0625 was used for  $e$ . Despite the low level of the noise it took again about 1 h to estimate 113 models and propose the structure

TABLE 2. VALIDATION RESULTS

QI	$n_a$	$\tau_1$	$n_{b1}$	$\tau_2$	$n_{b2}$	$n_d$	$\sigma_e^2$	dim ( $\theta$ )
8†§	2	2	3	7	2	2	0.26515	9
7‡	2	2	3	7	2	3	0.26439	10
6	3	2	3	7	2	3	0.26438	11
6	2	2	4	7	2	3	0.26439	11
5	2	2	3	7	2	0	0.35712	7
5	3	2	3	7	1	0	0.39520	7
5	2	2	3	7	2	1	0.35652	8
5	3	2	3	7	2	0	0.35697	8
5§	2	2	3	7	1	3	0.32708	9
5	3	2	3	7	2	2	0.26506	10

† Best BIC.

‡ Best validation variance.

§ Elbow.

which served to generate the data as the best choice.

## 7. CONCLUSIONS

The primary objective we set ourselves when we started working on an expert system for identification was to develop a software tool that would be able to identify a reasonably good model on most industrial data, thereby allowing us to save the considerable amount of engineering time that is normally spent identifying models from data. Even though ESPION is presently still limited to ARARX model structures, it has already achieved this objective to a considerable degree. With the industrial sets that we have handled, it has rarely been possible to find models that were significantly better than the best model obtained by ESPION. In addition, as already indicated, the model structure search strategy of ESPION significantly reduces the number of trial structures compared to, say, an exhaustive search. Our expert system seldom estimates more than 15% of the number of structures an exhaustive search would estimate, and this rate drops markedly (down to about 1%) with the complexity of the true or best model.

But if the development of ESPION has an engineering objective, it also has a scientific objective, namely to evaluate whether the structure estimation part of the system identification problem is amenable to an expert system approach. More precisely, the question at hand is whether the heuristics that different practitioners have developed to organize their own search rules through the space  $(Z_+)^q$  is based on experience and personal preference, or whether a rationale and, perhaps, some theoretical tools can be developed to organize this search in an efficient way. In this respect the present state of development of ESPION has already delivered two useful benefits: it has forced us to justify and sometimes alter our own search rules; it has also provided us with a true laboratory tool in which any proposed change in the search strategy or in the decision parameters (such as confidence levels) can be immediately put to a test by simply changing a few rules in the expert system accordingly. We can thus quickly evaluate the benefits and drawbacks of one particular identification scenario over another one.

Our expert system is still in an evolving stage. Many of the present rules as well as their ordering may still be modified, as the expert system is provided with more sophisticated learning capabilities. For example, at present the estimation variance is the only criterion used

during a search at constant model dimension and it does not indicate which structure modification is most appropriate given the current state of knowledge. On the other hand, the standard deviations of the parameter estimates and residual analysis could give such indications. In addition, the set of candidate model structures must be expanded to include more complex structures, such as ARMAX and Box-Jenkins model structures, and several open problems remain to be solved. However, we believe that, even at this stage of its development, our expert system has already proved its usefulness, both from an engineering and from a scientific point of view.

*Acknowledgements*—The authors would like to thank R. Hanus and J. L. Henrotte from the Service d'Automatique de Brussels University who made helpful comments about this work and the members on the staff of S.A. ABSY who have supported the development of a first version of this expert system. The financial support of IRSIA is also acknowledged. The results presented in this paper have been obtained within the framework of the Belgium Program on Concerted Research Actions and on Interuniversity Attraction Poles initiated by the Belgium State, Prime Minister's Office, Science Policy Programming. The scientific responsibility rests with its authors.

REFERENCES

Forgy, Ch. L. (1986). *The OPS83 User's Manual—System Version 2.2*. Production Systems Technologies, Inc., Pittsburgh, PA.

Gentil, S. and Ph. Conraux (1985). S.E.X.I. Système Expert en Identification. DEA Report, Laboratoire d'Automatique de Grenoble, Institut National Polytechnique de Grenoble.

Haest, M., G. Bastin, M. Gevers and V. Wertz (1988). An expert system for system identification. *Proc. 1st IFAC Workshop on Artificial Intelligence in Real-time Control*, Swansea, U.K., pp. 101–106.

Larsson, J. E. and P. Persson (1987). An expert system interface for Idpac. Technical Report LUTFD2/(TFRT-3184)/1-96/(1987), Department of Automatic Control, Lund Institute of Technology.

Larsson, J. E. and P. Persson (1988). The knowledge database used in an expert system interface for Idpac. *Proc. 1st IFAC Workshop on Artificial Intelligence in Real-time Control*, Swansea, U.K., pp. 107–112.

Ljung, L. (1987). *System Identification. Theory for the User*. Prentice-Hall, Englewood Cliffs, New Jersey.

Monson, M., B. Bergeon, A. Khaddad and M. Bansard (1988). An expert system for industrial process identification. *Proc. 1st IFAC Workshop on Artificial Intelligence in Real-time Control*, Swansea, U.K., pp. 95–99.

Wertz, V., G. Bastin and M. Haest (1987). Identification of a glass tube drawing bench. *Prepr. 10th IFAC World Congress on Automatic Control*, München, Federal Republic of Germany, Vol. 10, pp. 334–339.

APPENDIX

The BIC notation stands for the Bayesian Information Criterion as proposed by Akaike

$$BIC = N_c \ln(\sigma_e^2) + \ln(N_c) \dim(\theta).$$

To test that  $\sigma_v^2$  does not differ significantly from  $\sigma_e^2$ , the following inequality is checked:

$$\frac{\sigma_v^2}{k\sigma_e^2} < F_{95}(N_v - 1, N_c - 1)$$

where  $F_{95}(n_1, n_2)$  is the 95% level of the  $F$  distribution with  $n_1$  and  $n_2$  degrees of freedom and  $k$  is typically chosen as

$$k = \frac{\sigma_v^2(y)}{\sigma_e^2(y)}$$

where  $\sigma_v^2(y)$  and  $\sigma_e^2(y)$  stand for the variances of the output computed on the estimation and validation data sets, respectively.

The prediction errors in estimation are considered white if less than 5% of the absolute values of the autocorrelation coefficients of the prediction errors are greater than

$$\frac{1.96}{\sqrt{N_e}}$$

where 1.96 is the 97.5% level of the normal distribution with zero mean and unit variance. A similar test is used for prediction errors in validation with  $N_e$  replaced by  $N_v$ .

Finally, the prediction errors in estimation are considered independent from past samples of an input signal if less than 5% of the absolute values of the cross-correlation coefficients between these prediction errors and past samples of the corresponding input signal are greater than

$$1.96 \sqrt{\left(\frac{P}{N_e}\right)}$$

with

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

where  $M$  autocorrelation coefficients have been estimated both for the prediction errors ( $\rho_{ee}(k)$ ) and for the considered input signal ( $\rho_{uu}(k)$ ). A similar test is used for prediction errors in validation with  $N_e$  replaced by  $N_v$ .

