Journal of Applied Statistics, Vol. 18, No. 1, 1991

FILTEX: toward an expert system for time-series filter design

RADOVAN KRTOLICA[1], **IVAN OBRADOVIĆ**[2] & **RADMILO BOŽINOVIĆ**[3], *Mihailo Pupin Institute*[1], *University of Belgrade*[2], *Apple Computer Inc, CA, USA*[3]

SUMMARY *This paper describes an attempt to develop a statistical expert system (FILTEX) as an intelligent aid for time-series filter design. To this end the knowledge of the filter design strategy is represented in Prolog and coupled with numerical routines of a general purpose signal processing package. This knowledge-based system is conceived as a set of independent knowledge sources integrated into a system by a blackboard mechanism which embodies overall control of the filter design process. Modularity and flexibility of knowledge representation in such a framework preserve usability of the evolving system during its development from the original numerical package to an expert system for filter design. This approach seems to be more flexible than the use of shells and less time consuming than building from scratch. A novel method for incorporating classical statistical information into an uncertainty management mechanism is presented. Experimental results confirm the feasibility of the approach and set directions for further research.*

1 Introduction

Available measurements of a dynamic system generally represent the sum of the signal and noise. An important task of signal processing is to eliminate as much of the noise as possible through processing of the measured time series by a filter. The development of an intelligent aid for time-series filter design can be viewed as a paradigmatic example of an expert system for a specific statistical domain (Gale, 1986). Optimal design of time-series filters is one of the classical problems of mathematical statistics and the theory of stochastic processes (Wold, 1938; Kolmogorov, 1939; Wiener, 1949; Kalman, 1960). However, the attempt to adapt this model-based approach to the general problem of signal processing had only limited success mainly because of the following reasons:

(a) lack of information on validity of assumptions used by the theory;
(b) inability of theoretical criteria (such as minimum variance, maximum likelihood, etc.) to account for all the requirements posed by practical signal processing problems (e.g. computational efficiency or level of representation sought by the application).

Moreover, from the very early stages of implementation of optimal (statistical) filter design theory (see for example James *et al.*, 1947; Schmidt, 1966), it was clear that even in the classical domains of application a knowledgeable engineer is needed to perform it.

We believe that the use of artificial intelligence techniques to represent the knowledge of a time series specialist in addition to the existing software for signal processing makes a sound basis for a useful and practically achievable expert system for filter design. A knowledge-based approach allows us to model a skilled engineer's strategy in combining formal and informal methods for solving signal processing problems.

Currently, our attention is restricted to time series generated by linear SISO systems. A feasibility study was performed on an experimental system for simultaneous parameter and order evaluation of ARMAX models (Krtolica *et al.*, 1988). To perform this task, a general-purpose signal processing package SIG™* (Lager & Azevedo, 1985) was coupled with programs for symbolic processing written in Prolog. We pursued this approach by incorporating more expert knowledge into SIG which ultimately lead to a knowledge-based system for time series analysis and design (Krtolica *et al.*, 1989). At that point a standard Prolog interpreter without special features was used, which resulted in shallow coupling of SIG and Prolog where information was exchanged via external files. Shortly afterwards, we switched to IF/Prolog, a professional compiler/interpreter which provides a direct interface to numerical routines. This enables deeper coupling of numerical and symbolic parts of the system such that numerical routines can be incorporated into IF/Prolog through its interface to Fortran which offers the possibility of writing numerical, Fortran-based predicates and using them like standard Prolog predicates. The modular and multilayered Fortran-based organization of SIG gave us the opportunity to use its numerical routines which can carry out the statistical analysis of the time series data and the estimation of model parameters to fit the data, as the basis for FILTEX numerical predicates. The whole system is being developed on a VAX/VMS 750/11 installation.

In Section 2 we state the problem of filter design at hand. In Section 3 we describe the software components on top of which an evolving expert system is built. The current state of development is described in Section 4, and a possibilistic approach to the selection of filter structure in Section 5. Finally, Section 6 outlines the experimental framework and results. An example of an interactive FILTEX session is given in Appendix A.

2 Statement of the problem

Design and selection of appropriate processing algorithms (filters) for a given time series (signal) are an important part of the signal processing problem. Solving this

* SIG is a trademark of the Lawrence Livermore National Laboratory.

task requires some knowledge about the *signal source* (the dynamic system that generates signals in the application at hand), and knowledge of the *aim* and *constraints* of a particular implementation of the algorithm. This knowledge is used to develop a *design strategy* by which an *expert* in the field generates and selects the processing algorithms in accordance with the *data available*. However, a human expert in his current practice needs to possess only part of the knowledge encompassed by the outlined design strategy. The aim of a particular signal processing application, its constraints, as well as the structure of the model of the signal source, are all domain specific, while the statistical analysis of time series data and the estimation of model parameters are domain independent. As general-purpose signal processing CAD systems represent efficiently the main (procedural) part of this domain-independent body of knowledge, for an expert designer it is enough to possess the remaining, non-procedural part of this knowledge and knowledge of the domain-specific part of the design strategy (e.g. the goals and constraints of the algorithm implementation, and the structure of the signal sources).

This paper describes an attempt to complete the representation of the filter design strategy which is only partially automated by general-purpose signal processing CAD systems. To this end, we must represent the non-procedural part of the domain-independent knowledge and the domain-specific knowledge (e.g. the constraints to be met and the goals to be achieved by an implementation of the selected filter); we must also represent the available knowledge about the class of allowable signal source models in the domain of interest.

From the classical CAD point of view, the difficulty in representing this kind of knowledge lies in the fact that it is qualitative and thus cannot be translated into numerical algorithms easily. Our concept of knowledge-based filter design takes advantage of the development of symbolic computation in order to overcome this difficulty.

3 Structure of the FILTEX system

The expert system for design of time-series filters (FILTEX) is conceived as a set of knowledge-based modules integrated into a system using the blackboard architecture (Nii, 1986; Engelmore & Morgan, 1988). Each module represents an independent source of knowledge, which incorporates one or more numerical routines and a knowledge base of its own, and performs a task that contributes to the problem-solving process, thus realizing part of the overall filter–selection strategy. Individual tasks generally belong to one of the following groups:

(a) input/output and dialogue handling (performed by the INOUT knowledge source group);
(b) preprocessing: tests of periodicity (seasonability) and stationarity, normalization by non-linear transformations, smoothing, etc. (performed by the PREPROC group);
(c) correlation and spectral estimation of stationary time series (performed by the SPECORR group);
(d) selection of shaping filters (low-pass, high-pass, band-pass . . .) for the given set of signals (performed by the SHAPE group);
(e) model-based approach to signal processing (optimal filtering and prediction of stationary time series, performed by the MODEL group).

The overall structure of FILTEX is depicted in Fig. 1.

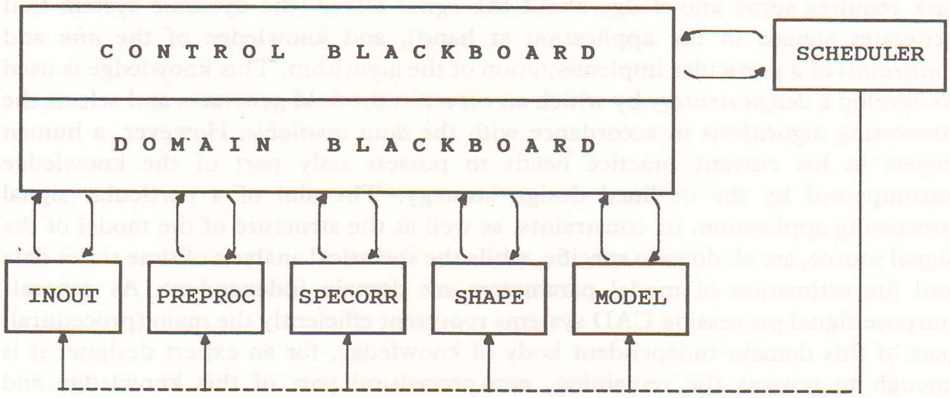

FIG. 1. Overall structure of the FILTEX system.

The blackboard architecture offers a mechanism for achieving opportunistic reasoning and flexible problem solving. In this approach domain knowledge is organized into independent knowledge sources which correspond to different areas of specialization within the problem-solving process and communicate mutually through a global data base called the blackboard. All input information, partial solutions and alternatives which lead to the final solution are kept on the domain blackboard. The current state of the solution is kept on the control blackboard and it determines which of the knowledge sources can, in the next step of the problem-solving process, contribute to the final solution. The scheduling mechanism (scheduler) selects and activates one of these knowledge sources which creates a new partial solution, or modifies an existing, according to the specialized knowledge encoded within. This mechanism can be based on different criterion functions (e.g. priority of knowledge sources, knowledge source confidence level, etc.), which provides for the realization of flexible problem-solving strategies. In other words, the strategy can be easily modified by changing either the values of the criterion function for particular knowledge sources or the criterion function (i.e. scheduling mechanism) itself.

The knowledge sources and the blackboard can take various representation forms. It is well known that frames represent a clearly defined concept for knowledge representation (Winston, 1984). At the same time, this concept, being flexible and open, can easily accommodate the needs of specific problem-solving. Each frame contains slots which describe stereotyped properties of a given object, act or event and its connections to other frames within the hierarchical structure of the frame system. A particular slot can be filled in three different ways: by giving it a value; by inheritance from a frame at a higher level of abstraction; by assigning a procedure to be fired in order to compute the value of the slot when it is required, which provides for an active nature of the frame system. Since slots can contain both declarative and procedural information it seemed convenient to use frames for holding knowledge encoded in knowledge sources, as well as various kinds of information about the knowledge sources themselves, such as the priority of execution, the circumstances in which knowledge sources can apply their knowledge, etc. Frames were also used to represent the control and domain blackboard as well. The values of their slots represent the relevant control and domain information (partial solutions, alternatives, etc.).

The choice of Prolog by which to implement the FILTEX frame system was based on the following reasons:

(a) versus procedural high-level languages ('non-artificial intelligence')—ease of processing and adequate power;
(b) versus LISP—logic oriented, easier inference engine building, due to built-in resolution and unification;
(c) versus commercially available shells—flexibility of specific expert system design (see for example Merritt, 1989).

4 Current state of development

FILTEX is an evolving expert system whose modular structure allows for an incremental implementation. Presently, it encompasses the core of the blackboard control mechanism (Hayes-Roth, 1985) and the knowledge sources related to the selection of shaping filters and to the model-based approach. What follows is a brief description of the salient features of the part implemented so far.

The filter design process starts with a dialogue, led by an appropriate knowledge source, in which the user specifies the problem at hand, i.e. the (output) time series that requires filtering, the appropriate excitation time series (optionally), and other relevant information available (e.g. information concerning the signal source). The results of this dialogue are registered on the blackboard. If these results indicate that direct noise elimination can be achieved, i.e. that the frequency spectra (range) of the signal and noise can be separated by means of a shaping filter, then the selection of such a filter can represent a satisfactory solution. If this is not the case, then only a filter based on linear prediction might solve the problem. Since the numerical part of FILTEX offers algorithms for both classes of filters, the main strategic goal can be decomposed into two disjoint subgoals: selection of an appropriate shaping filter or the selection of a filter based on linear prediction.

The selection of a shaping filter generally represents a simpler task since it is based on heuristic rules established by traditional engineering practice. The corresponding strategy represents a less interesting part of FILTEX, based on a simple and practical characterization of a limited number of filter types, which leads to a more or less 'classical' rule-based system for the selection of the appropriate filtering algorithm.

The filters based on linear prediction (predictive filters) require a model of the filtered signal (see for example Candy, 1986). For practical purposes, we restricted the allowable models to the class of linear, finite-dimensional, parametric SISO models. Even so, the number of possible solutions is extremely large, and the effects of particular models on the quality of filtering have not yet been totally clarified. The selection of a predictive filter, being such a complex problem, allows for multiple problem-solving strategies. Consequently, the representation of relevant knowledge (both theoretical and empirical) in the expert system leads to interesting knowledge-based design problems. We shall therefore focus our discussion on the current implementation of strategy related to predictive filters.

The adopted strategic paradigm for predictive filter design was the search for a least complex acceptable solution (the principle of parsimony). Engineering experience points out that this approach represents a general characteristic of predictive filter design (as well as the design of technical solutions in general). A linear predictive model can either belong to one of the elementary classes (the autoregressive (AR), the moving average (MA) or the input-only (X) model), or it can be represented as a combination of elementary models and thus belong to one of the complex classes (ARMA, ARX, MAX or ARMAX). The design strategy relies on subsequent creation of models from different classes, according to the strategic paradigm, followed by numerical realization (simulation) of appropriate residual sequences (parameter identification and prediction error simulation).

Following the principle of parsimony, the first strategic goal is an elementary model which satisfies the assumed constraints such as allowable model complexity or acceptable final prediction error. In an attempt to satisfy this goal, a model from each of the elementary classes (AR, MA, X) is proposed by corresponding knowledge sources, following their local optimization rules. A set of such rules relates to the knowledge used by an expert performing visual analysis of the shape of the following four functions (Box & Jenkins, 1976; Hippel *et al.*, 1977):

— autocorrelation function (ACF);
— partial autocorrelation function (PACF);
— inverse autocorrelation function (IACF);
— inverse partial autocorrelation function (IPACF).

A sample rule from this set is given (handcompiled into an informal syntax) in Fig. 2. Another set encompasses the rules based on such criteria as structure complexity and statistical parameters of the residual sequence (mean, variance, etc.). All elementary models that satisfy the constraints are model candidates and represent possible solutions. However, if no elementary model candidates were found then the search for a model candidate is continued in complex classes by combining the two (three) elementary models. Knowledge sources which propose models from complex classes are invoked and a model from each class is created. The obtained complex models that satisfy the constraints now represent the set of model candidates. However, if again no model candidates were found, an extensive search is performed in the limits of the allowable model complexity, i.e. all models within these limits are created and checked. If even this time-consuming application of 'brute force' fails, i.e. no model candidates emerge, then the user must be informed that under present constraints an appropriate solution cannot be found.

If at any point in the aforementioned strategy at least one model candidate appears, the set of acceptable models becomes non-empty. Following again the strategic paradigm, an attempt is made to widen this set by looking for possible less complex model candidates. The least complex candidate in the initial set is identified and models obtained by lowering the complexity of this model candidate are examined. If model candidates are found among them they are included in the set of acceptable solutions. Finally, a choice of model candidates from this set is offered to the user according to a set of rules for final selection. The aim of these rules is primarily to choose an appropriate selection criterion for pruning the set of model candidates. Currently the criteria include Pareto optima (on mean, variance and possibly model complexity), clustering/partitioning of variances, specialised criteria (e.g. Akaike's),

(R1)	q is the order of the MA part of the model with probability $p(\varepsilon)$	
	IF	the absolute value of ACF of the given output signal is less than ε, when the argument t is greater than or equal to $q+1$,
	AND	the input signal does not exist.

FIG. 2. A rule for the analysis of the autocorrelation function (ACF).

(R2) Apply clustering of variances
IF the signal/noise ratio is greater than α,
AND complex poles are present in the model.

(R3) Complex poles are present in the model
IF fast Fourier transformation of the input signal reveals identifiable poles.

FIG. 3. Sample rules from the second group.

and a (novel) possibilistic approach which, being also of theoretical interest, is described in some detail in Section 5. A sample of these rules is given in Fig. 3. As a result of this final selection of model candidates, based on numerical processing and rule-based reasoning, a most appropriate model (or set of models) is presented to the user. He then has the opportunity to either confirm the system's choice and, in the case of a set of models, pick one of them as the most suitable or overrule this proposal by choosing a model he considers to be the most appropriate. Upon this final choice a predictor model (filtering algorithm) is automatically generated. Simulation facilities can be used to test the selected algorithm on various time sequences.

The outlined strategy obviously represents only one possible path to the solution. The available pieces of knowledge can be combined in other ways to achieve the same strategic goal: the selection of an appropriate filter for the elimination of noise from a measured signal. Different strategies with the blackboard architecture can be realized by rescheduling the knowledge source activation, i.e. by changing the scheduling mechanism. Obviously, the rescheduling cannot be totally arbitrary since the current state of the solution on the blackboard determines which of the knowledge sources can be activated at that very moment. In order to formalize all interdependences between the knowledge sources and the blackboard, and to identify different possibilities of both control and data flow, a specialized inference network was developed (Obradović *et al.*, 1990).

5 An approach to the selection of filter structure based on possibility theory

One element of the knowledge base within the knowledge source for final selection deserves particular attention and relates to a specific (and more theoretically justified) decision mechanism for selecting (one or more) model candidates. This approach is based on a generalization of appropriate t and F tests for residual error sequences of various model candidates and the use of possibility theory to interpret and combine them. Namely, classical interpretation of the statistical tests is given as binary decision making, based on arbitrary confidence levels. Also, such an approach does not allow for a meaningful combination of independent statistics test results. In contrast, here we generate a possibility distribution over the candidate structures where the degree of possibility for various structures is based upon the value of their F and t statistics (more precisely, on a generalization of what is labeled in the classical test setting as the confidence level or risk of rejecting the true hypothesis). In what follows, we describe the key ideas, assuming some acquaintance with both possibility and statistical testing theories.

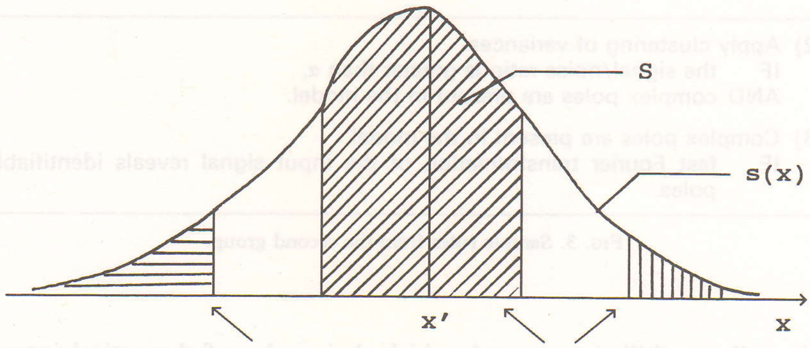

FIG. 4. Analyzing the distribution hypothesis, based on a realization of the random variable (t or F). The arrows indicate possible locations of x_i that determine the zone R and integral r_i; the limits of the zone S are chosen so that equal areas appear on both sides of x', and the total area is $1-2c^{-1}$.

Let us observe a finite set of n candidates, $\mathbb{K}$. The proposed method includes generating a possibility distribution over all candidates $k_i \in \mathbb{K}$ $(i=\overline{1,n})$ based on the following steps:

(1) Apply a 'modified test', i.e. compute, for each value of i

$$r_i = \int_R s(x)\,dx \tag{1}$$

where $s(x)$ represents the density function for the appropriate statistics (random variable, e.g. t or F). The integration domain R depends on the realization of the statistics x_i for a given candidate k_i: when x_i is greater than the mean x', $R=\{x|x>x_i\}$, otherwise $R=\{x|x<x_i\}$ (Fig. 4).

(2) Compute

$$\begin{aligned} \pi(k_i) &= cr_i \quad && \text{if } r_i < c^{-1} \\ &= 1 && \text{if } r_i \geqslant c^{-1} \end{aligned} \tag{2}$$

where $c \geqslant 2$ is a given constant.

(3) In the case where $\pi(k_i) > 1$ for all $i=\overline{1,n}$, introduce another event, 'other candidates' (k_0), with possibility $\pi(k_0)=1$ (in order to normalize the possibility distribution).

(4) Repeat steps (1)–(3) for different test statistics and combine the obtained distributions using the possibilistic combination rule (Prade, 1985). For two distributions $\pi_1(\text{—})$ and $\pi_2(\text{—})$ defined over some set $\mathbb{P}$, a new possibility distribution $\pi_3(\text{—})$ can be defined as follows:

$$\forall p \in \mathbb{P},\ \pi_3(p) = \min\{\pi_1(p), \pi_2(p)\}/d$$

where

$$d = \sup_{p \in \mathbb{P}} \min\{\pi_1(p), \pi_2(p)\}$$

is a (re)normalizing coefficient.

We now give a justification of this procedure.

(1) The apparata of possibility measures and distributions address basically the same phenomena as probability theory, which is at the core of mathematical

statistics. Since there is insufficient information for the use of probabilities, we use the less strict possibilistic setting.

(2) The distribution defined through steps (1)–(3) satisfies the conditions for a possibility distribution. It is easy to see that it satisfies the definition

$$0 \leqslant \pi(k_1) \leqslant 1; \quad \exists k_i \in \mathbb{K},\ \pi(k_i) = 1$$

Moreover, it can be considered as generated from n separate possibility distributions $\pi_i(\text{—})$ such that

$$\forall i, \forall j,\ \pi_i(k_j) = 1 \qquad \text{if } i \neq j$$

$$0 \leqslant \pi_i(k_j) \leqslant 1 \qquad \text{if } i = j$$

which are uniquely combined via the possibilistic rule of combination, since it is both commutative and associative. Thus, we have

$$\forall k_i \in \mathbb{K},\ \pi(k_i) = \min\{\pi_i(k_i), 1\} = \pi_i(k_i)$$

and the normalization coefficient d is equal to 1.

(3) Each of the n distributions is an intuitively satisfactory interpretation of the associated modified tests, since the information it carries is possibilistic in nature. As with classical testing, no quantification of the null hypothesis H_0 can be made, as opposed to the alternative hypothesis H_1. The integral (1) directly generalizes the notion of risk (confidence level) in statistical tests having in mind that in this case, for preset values of risk, we determine the probability (relative frequency) of falling in or out of the critical region. With the modified test, for a given value x_0 of the statistics (the realized value of the test), we seek the significance of its distance from the expected value x' as a measure of hypothesis validity. Simply, the 'closeness' of the realized value x_0 and the expected value x' is only a necessary condition for testing H_0, therefore large values of $|x' - x_0|$ render evidence in favor of H_1 (in a measurable fashion), whereas small values provide no specific evidence whatsoever.

Further details about the choice of c, practical results and other aspects of this approach are given by Božinović *et al.* (1988) and Obradović *et al.* (1989).

The following can be said of the presented possibilistic approach to filter selection:

(a) it conforms with the intuitive meaning of statistical tests;
(b) it preserves all the information present in the test statistics which is lost in classical testing;
(c) it gives a clearer picture about hypothesis quality, through the necessity/possibility pair of values;
(d) it allows for a meaningful and tractable combination of independent statistics test results via the possibilistic combination rule.

Preliminary experimentation has proved the viability of this concept.

6 Experimental issues

A number of experiments have been performed with the implemented part of the system, mainly to test the model-based case, using the simulation facilities to control fully the experimental framework. In other words, the output signals were generated using an algorithm for linear system simulation, by specifying the following:

— the input signal
— the noise signal

— the model of the system itself (delay function polynomials of the ARMAX model)

The nature and values of all three parameters have been varied over a broad range. Some of the types of input signals used were as follows:

— a delayed step function
— a ramp function
— a piece-wise periodical function
— Gaussian noise as the input

The noise signals were all white Gaussian signal, with zero mean and a variance changing from 10^{-5} to 10^{2}. The case without noise was also examined.

As to the models of the system, various delay function polynomials have been hand-picked in order to account for, among other cases

— different orders of the model (polynomials)
— polynomials with real and complex zeros
— models with and without zero-pole cancellation
— stable and unstable models

In particular, for each model defined we varied the input and noise signals to produce a set of 10–20 output signals, given a total of approximately 100 test cases. Each of the output signals generated, together with the corresponding input signal, was then subjected to filter design via the FILTEX system. Since all these signal sources were 'identified' as linear, time-invariant SISO models, the design procedure ended with a model (or set of models) chosen by the system as most appropriate, or a report of failure to accomplish the task. In the case of a successful ending, the chosen model(s) could then be compared with the one which originally generated the output signal.

This experimental framework was primarily intended to serve as a means for testing system performance, but it also turned out to be a valuable source of new knowledge, which could then be incrementally built in the system. Thus FILTEX evolved not only by incorporating established ('external') signal processing heuristics, but also the 'internal' ones which appeared in the experimentation phase. We now briefly state a few of the major conclusions offered by the experimentation material.

(a) Identification ended successfully only for stable models, since in the case of instability, almost always numerical overflow was encountered.
(b) Mildly non-stationary input signals do not result in higher identification failure rate.
(c) For a successful identification, excitation persistance is complementary with noise magnitude.
(d) Partitioning the set of model candidates into two classes (of 'possible' and 'inferior' models) by clustering the residual error sequence variances was possible in most cases: the variances in the 'possible' set were of the same order of magnitude, and considerably higher in the 'inferior' set. The first set always included the true model.
(e) Within the 'possible' set, the true model was often indistinguishable from the rest; in this sense the set seemed to form an equivalence class of models.
(f) A larger signal-to-noise ratio results in better discrimination within the set of model candidates.

(g) The presence of complex poles in the model allowed for a good discrimination even for lower signal-to-noise ratios.
(h) The variances of residual sequences in the 'possible' set of models were of the same order of magnitude as the variance of the input noise.
(i) When the assumed degree of the input signal polynomial was inferior to the true degree the appropriate model usually belonged to the set of 'inferior' models, whereas no systematic pattern of overfitting has been observed.
(j) Within the possibilistic approach certain values of the constant c yielded good discrimination.

Currently, when the system has to perform an extensive search for model candidates the response time tends to be large (several minutes).

7 Conclusions

An attempt to represent the knowledge of the filter design strategy by upgrading a general-purpose signal processing package is proposed. Incorporating this expert knowledge in such a numerical package is a practical alternative to the classical dilemma between use of shells (often too restrictive) and building from scratch (often time consuming). This results in a coupled system that integrates the existing numerical routines with symbolic computation in a blackboard architecture realized through a system of frames written in Prolog. Such an approach offers the possibility to preserve usability during the evolution from the original numerical package to the expert system for filter design. Two main features of such an evolving system are system modularity and flexibility of its knowledge representation framework. This framework allows incremental building of the knowledge acquired during the experimental work with the system. We also proposed a novel method to incorporate classical statistical information into an uncertainty management mechanism. Initial experimental results confirm the feasibility of the overall approach.

Acknowledgements

This work was supported by the Serbian National Foundation for Scientific Research, under Grant T-142, and the Yugoslav Federal Research Committee, under Grant PR-84. We would like to express our appreciation to Lawrence Livermore National Laboratory for the use of SIG and to Jim Candy, one of the authors. We are grateful to Bill Gale of AT&T Bell Laboratories for his encouragement and support, and to Sanja Petrović of the Mihailo Pupin Institute Knowledge Engineering Laboratory for her useful comments and assistance.

Correspondence: Radovan Krtolica, Knowledge Engineering Laboratory, Mihailo Pupin Institute, Volgina 15, 11000 Belgrade, Yugoslavia.

REFERENCES

Box, G. E. P. & Jenkins, G. M. (1976) *Time Series Analysis: Forecasting and Control,* 2nd ed (San Francisco, CA, Holden-Day).

Božinović, R., Krtolica, R. & Obradović, I. (1988) An interpretation of the results of statistical tests using the Dempster–Shafer theory, *Prep. XV Yugoslav conference SYM-OP-IS '88*, pp. 3–6 (in Serbo-Croatian).

Candy, J. V. (1986) *Signal Processing. The Model-Based Approach* (New York, McGraw-Hill).

Engelmore, R. & Morgan, T. (eds) (1988) *Blackboard Systems* (Reading, MA, Addison-Wesley).
Gale, W. A. (ed.) (1986) *Artificial Intelligence and Statistics* (Reading, MA, Addison-Wesley).
Hayes-Roth, B. (1985) A blackboard architecture for control, *Artificial Intelligence*, 26, pp. 251–321.
Hippel, K. W., McLeod, A. I. & Lenox, W. C. (1977) Advances in Box–Jenkins modeling, 1 model construction, 2 application, *Water Resources Research*, 13, 3.
James, H. N., Nichols, N. B. & Phillips, R. S. (1947) *Theory of Servomechanisms* (New York, McGraw-Hill).
Kalman, R. E. (1960) A new approach to linear filtering and prediction problems, *Journal of Basic Engineering, Trans. of the ASME, Series D*, 82, 1, pp. 35–45.
Kolmogorov, A. (1939) Sur l'interpolation et extrapolation des suites stationnaires, *C.R. Acad. Sci. Paris*, 208, pp. 2043–2045.
Krtolica, R., Obradović, I. & Božinović, R. (1988) Application of logic programming in linear system identification, *Prep. 4th IFAC Symposium on Computer Aided Design in Control Systems*, pp. 605–608.
Krtolica, R., Obradović, I. & Božinović, R. (1989) A knowledge-based system for time-series analysis and representation, in: M. H. Hamza (ed.), *Proc. IASTED Int. Symp. on Expert Systems, Theory and Applications*, pp. 290–293 (ACTA Press).
Lager, D. & Azevedo, S. (1985) SIG—a general purpose signal processing program, *Report UCID-19912 Rev. 1* (Lawrence Livermore National Laboratory).
Merritt, D. (1989) *Building Expert Systems in Prolog* (New York, Springer).
Nii, H. P. (1986) The blackboard model of problem solving and the evolution of blackboard architectures, *The AI Magazine*, Summer 1986, pp. 49–64.
Obradović, I., Krtolica, R. & Božinović, R. (1989) Using possibility theory to incorporate statistical knowledge in the inference engine, *Proc. XXXIII Yugoslav ETAN Conference*, pp. XIII.27–34 (in Serbo-Croatian).
Obradović, I. *et al.* (1990) A blackboard approach to problem-solving strategies in filter selection, *Proc. XII International Symposium: Computers at the University*, pp. 8.5.1–8.5.6.
Prade, H. (1985) A computational approach to approximate and plausible reasoning with applications to expert systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7, pp. 260–283.
Schmidt, S. F. (1966) Applications of state space methods to navigation problems, in: C. T. Leondes (ed.), *Advances in Control Systems*, Vol. 3 (New York, Academic Press).
Wiener, N. (1949) *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications* (Cambridge, MA, MIT Press).
Winston, P. H. (1984) *Artificial Intelligence* (Reading, MA, Addison-Wesley).
Wold, H. (1938) *A Study in the Analysis of Stationary Time Series* (Upsalla, Almqvist & Wiksells).

Appendix A: Example of an interactive FILTEX session

WELCOME TO FILTEX

The EXpert system for FILTer design

Please enter the data store number of the output signal.
–> 30
Does the input (excitation) signal exist?
–> yes
Please enter the data store number of the input signal
–> 31
Is the signal/noise ratio favorable?
–> explain
A favorable signal to noise ratio is a ratio over 100 : 1.
Is the signal/noise ratio favorable?
–> dont know
Then I have to assume an unfavorable signal/noise ratio and look for a solution in the class of filters based on linear prediction.

Do you want to set a limit to model complexity?
–⟩ yes
In that case, please enter the allowable model complexity.
–⟩ 7
Can you give any information about the variance of the white noise?
–⟩ no
Can you set a limit to the final prediction error?
–⟩ yes
Please enter the acceptable final prediction error.
–⟩ 2.0
The default strategic paradigm is the search for a least complex acceptable solution.
Do you accept this strategy?
–⟩ explain
An attempt is made to find a model from an elementary class (AR, MA, X).
If that fails, a model is searched for in complex classes (ARMA, ARX, MAX, ARMAX). If that also fails, an extensive search is performed.
(The alternative strategy is an immediate extensive search within the limits of allowable complexity).
Do you accept this strategy?
–⟩ yes
According to the selected strategy, I shall try to identify an elementary model which satisfies the assumed constraints. The identification is based on a set of rules for local optimization.

The Levinson algorithm identifies the following model:

Model class: AR
Final prediction error: 2.28847 Complexity: 2 Variance: 2.23545
A-order: 2 Parameters: [1.0, –1.75355, 0.789704]
B-order: 0 Parameters: [0.0]
C-order: 0 Parameters: [1.0]

The harmonic lattice algorithm identifies the following model:

Model class: AR
Final prediction error: 2.28916 Complexity: 2 Variance: 2.23612
A-order: 2 Parameters: [1.0, –1.74064, 0.77758]
B-order: 0 Parameters: [0.0]
C-order: 0 Parameters: [1.0]

The geometric lattice algorithm identifies the following model:

Model class: AR
Final prediction error: 2.29233 Complexity: 2 Variance: 2.23923
A-order: 2 Parameters: [1.0, –1.77248, 0.806374]
B-order: 0 Parameters: [0.0]
C-order: 0 Parameters: [1.0]

The minimum lattice algorithm identifies the following model:

Model class: AR
Final prediction error: 2.31017 Complexity: 2 Variance: 2.25666
A-order: 2 Parameters: [1.0, –1.80504, 0.835763]
B-order: 0 Parameters: [0.0]
C-order: 0 Parameters: [1.0]

Each of the applied algorithms proposed the same structure: [2,0,0]. This structure will be used for the ABC-algorithm (recursive least square algorithm).

The ABC-algorithm identifies the following model:

Model class: AR
Final prediction error: 2.28934 Complexity: 2 Variance: 2.2363
A-order: 2 Parameters: [1.0, −1.7412,0.777598]
B-order: 0 Parameters: [0.0]
C-order: 0 Parameters: [1.0]

Be aware of the fact that this identification ignores the input signal!

I shall now propose a moving average (MA) model:

The ABC-algorithm identifies the following model:

Model class: MA
Final prediction error: 12.2187 Complexity: 4 Variance: 11.7506
A-order: 0 Parameters: [1.0]
B-order: 0 Parameters: [0.0]
C-order: 4 Parameters: [1.0,0.807131,0.735356,0.702332,0.723644]

Be aware of the fact that this identification ignores the input signal!
Since excitation input exists I shall now propose an input-only (X) model.

The ABC-algorithm identifies the following model:

Model class: X
Final prediction error: 135.016 Complexity: 3 Variance: 130.862
A-order: 0 Parameters: [1.0]
B-order: 3 Parameters: [192.605, −41.2294,6.78511,99.2071]
C-order: 0 Parameters: [1.0]

The following set of elementary models is obtained:

AR [2,0,0] Final prediction error: 2.28934 Complexity: 2
MA [0,0,4] Final prediction error: 12.2187 Complexity: 4
X [0,3,0] Final prediction error: 135.016 Complexity: 3

Models with allowable complexity are listed below:

AR [2,0,0] Final prediction error: 2.28934 Complexity: 2
MA [0,0,4] Final prediction error: 12.2187 Complexity: 4
X [0,3,0] Final prediction error: 135.016 Complexity: 3

There are not models with acceptable final prediction error!
None of the proposed elementary models satisfies both constrains: allowable model complexity and acceptable final prediction error. The search for a model is continued in complex classes that combine the two (three) elementary models.
Combining the two elementary models AR and MA results in the ARMA-model [2,0,4].

The ABC-algorithm identifies the following model:

Model class: ARMA
Final prediction error: 12.6558 Complexity: 6 Variance: 11.9821
A-order: 2 Parameters: [1.0, −1.75826,0.800797]

B-order: 0 Parameters: [0.0]
C-order: 4 Parameters: [1.0,−0.324869,−0.333956,−0.177451,−0.108827]

Be aware of the fact that this identification ignores the input signal!
Identification of the ARX-model is based on the following heuristic: B-order is taken from the identified X-model, A-order is increased until a local optimum is reached.

The ABC-algorithm identifies the following model:

Model class: ARX
Final prediction error: 1.71155 Complexity: 7 Variance: 1.60782
A-order: 4 Parameters: [1.0,−2.31059,2.05395,−0.980142,0.256049]
B-order: 3 Parameters: [−125.191,156.177,−76.4071,13.2458]
C-order: 0 Parameters: [1.0]

Identification of the MAX-model is based on the following heuristic: B-order is taken from the identified X-model, C-order is increased until a local optimum is reached.

The ABC-algorithm identifies the following model:

Model class: MAX
Final prediction error: 24.1885 Complexity: 10 Variance: 22.1954
A-order: 0 Parameters: [1.0]
B-order: 3 Parameters: [111.848,91.3862,−14.9579,343,849]
C-order: 7 Parameters: [1.0,0.742962,0.580706,0.694335.0.49722,
0.516644,0.357807,0.198458]

Identification of the ARMAX-model is based on the following heuristic: B-order is taken from the identified X-model, A-order and C-order are taken from the ARX and MAX models and decreased until a local optimum is reached.

The ABC-algorithm identifies the following model:

Model class: ARMAX
Final prediction error: 1.68277 Complexity: 6 Variance: 1.59319
A-order: 2 Parameters: [1.0,−1.79429,0.81718]
B-order: 3 Parameters: [−141.24,109.43,−3.08864,−15.2422]
C-order: 1 Parameters: [1.0,0.724317]

The following set of complex model is obtained:

ARMA	[2,0,4]	Final prediction error: 12.6558	Complexity: 6
ARX	[4,3,0]	Final prediction error: 1.71155	Complexity: 7
MAX	[0,3,7]	Final prediction error: 24.1885	Complexity: 10
ARMAX	[2,3,1]	Final prediction error: 1.68277	Complexity: 6

Models with allowable complexity are listed below:

AR	[2,0,0]	Final prediction error: 2.28934	Complexity: 2
MA	[0,0,4]	Final prediction error: 12.2187	Complexity: 4
X	[0,3,0]	Final prediction error: 135.016	Complexity: 3
ARMA	[2,0,4]	Final prediction error: 12.6558	Complexity: 6
ARX	[4,3,0]	Final prediction error: 1.71155	Complexity: 7
ARMAX	[2,3,1]	Final prediction error: 1.68277	Complexity: 6

Models with acceptable final prediction error are listed below:

ARX [4,3,0] Final prediction error: 1.71155 Complexity: 7
ARMAX [2,3,1] Final prediction error: 1.68277 Complexity: 6

The set of the models which satisfy both constrains (allowable model complexity and acceptable final prediction error) is no more empty. An attempt is made to widen this set with additional, less complex models.
I shall start with the least complex model: ARMAX [2,3,1].
By decreasing it's complexity successively the following models can be obtained.

The ABC-algorithm identifies the following model:

Model class: ARMAX
Final prediction error: 1.68372 Complexity: 5 Variance: 1.6066
A-order: 2 Parameters: [1.0, −1.78967,0.812438]
B-order: 2 Parameters: [−138.546,110.364, −15.5921]
C-order: 1 Parameters: [1.0,0.685271]

The ABC-algorithm identifies the following model:

Model class: ARMAX
Final prediction error: 1.68456 Complexity: 4 Variance: 1.62002
A-order: 2 Parameters: [1.0, −1.84254,0.868383]
B-order: 1 Parameters: [−125.67,97.6667]
C-order: 1 Parameters: [1.0,0.537024]

The ABC-algorithm identifies the following model:

Model class: ARX
Final prediction error: 1.91759 Complexity: 3 Variance: 1.85859
A-order: 2 Parameters: [1.0, −1.85316,0.881658]
B-order: 1 Parameters: [−78.1468,51.2152]
C-order: 0 Parameters: [1.0]

Since the final set of model-candidates includes only 5 models, no further pruning will be done. Please select one of the following criteria for ranking the set of model-candidates:

A. Final prediction error
B. Model complexity
C. Residual sequence variance
D. Akaike's information criterion

Which criterion do you want to select?
-> A

The order of the proposed model-candidates is:

Model class: ARMAX
Final prediction error: 1.68277 Complexity: 6 Variance: 1.59319
A-order: 2 Parameters: [1.0, −1.79429,0.81718]
B-order: 3 Parameters: [−141.24,109.43, −3.08864, −15.2422]
C-order: 1 Parameters: [1.0,0.724317]

Model class: ARMAX
Final prediction error: 1.68372 Complexity: 5 Variance: 1.6066
A-order: 2 Parameters: [1.0, − 1.78967,0.812438]
B-order: 2 Parameters: [− 138.546,110.364, − 15.5921]
C-order: 1 Parameters: [1.0,0.685271]

Model class: ARMAX
Final prediction error: 1.68456 Complexity: 4 Variance: 1.62002
A-order: 2 Parameters: [1.0, − 1.84254,0.868383]
B-order: 1 Parameters: [− 125.67,97.6667]
C-order: 1 Parameters: [1.0,0.537024]

Model class: ARX
Final prediction error: 1.71155 Complexity: 7 Variance: 1.60782
A-order: 4 Parameters: [1.0, − 2.31059,2.05395, − 0.980142,0.256049]
B-order: 3 Parameters: [− 125.191,156.177, − 76.4071,13.2458]
C-order: 0 Parameters: [1.0]

Model class: ARX
Final prediction error: 1.91759 Complexity: 3 Variance: 1.85859
A-order: 2 Parameters: [1.0, − 1.85316,0.881658]
B-order: 1 Parameters: [− 78.1468,51.2152]
C-order: 0 Parameters: [1.0]

Do you want ARMAX [2,3,1] to be the chosen candidate?
−> no
Please enter the structure of the candidate you want to choose.
−> [2,1,1]
I shall now prepare a file containing the predictor coefficients for this model.
The predictor coefficients are in the file "pred.dat", and the error predictor coefficients in the file"epred.dat"; both can be transferred to SIG data stores via the RCREAD command.

GOOD BYE FROM FILTEX!