Expert Systems With Applications, Vol. 7, pp. 85–92, 1994
Printed in the USA.

0957-4174/94 $6.00 + .00

Frame-Based Blackboard Expert System for Signal Processing

SANJA PETROVIĆ

Knowledge Engineering Laboratory, Mihajlo Pupin Institute, Belgrade, Serbia

IVAN OBRADOVIĆ

University of Belgrade, RGF, Belgrade, Serbia

Abstract—*The selection of an appropriate filter for a given signal is a complex problem that requires both numerical and symbolic processing. To that end a coupled expert system (FILTEX) was developed by upgrading a professional digital signal processing software package (SIG) with symbolic knowledge representing expert heuristics and experience required for filter selection. In order to realize opportunistic filter selection strategies, the blackboard architecture was applied. The design of the FILTEX blackboard components and the frame-based realization in IF/Prolog programming language are given, ending with a sample session with the system.*

1. INTRODUCTION

SYSTEM ARCHITECTURE is one of the main issues in the design process of an expert system. The complexity of the problem to be solved and the availability and reliability of relevant information have a heavy impact on the reasoning process that may be applied. A large class of problems for which expert systems are built are not solved using only backward or forward chaining. Thus a standard rule-based architecture is not the appropriate solution for these systems. For solving complex, ill-defined problems an opportunistic reasoning approach is in use. This approach proceeds in a step-by-step manner interchanging the reasoning methods according to the current state of the solution.

It is well known that the blackboard architecture supports best the implementation of opportunistic reasoning (Nii, 1986; Engelmore & Morgan, 1988). Thus, expert systems aimed at solving problems requiring an opportunistic reasoning approach are often built using the blackboard concept.

One of the problems typical of the aforementioned class of complex problems is the filter selection problem in signal processing. The selection of an appropriate filter model for a given measured signal is based both on information that can be extracted from the signal and information about the dynamic system generating the signal. The latter is usually supplied by the user and often may be incomplete or unreliable, or both. In developing FILTEX, the expert system for filter selection (Krtolica, Obradović, & Božinović, 1991), we tried to overcome the complexity of the problem-solving strategy implementation by selecting the blackboard concept as the main architectural approach.

In this paper we outline the main design issues of FILTEX, that is, the realization of the blackboard architecture in this expert system. The implementation of the blackboard concept is based on a frame system, and the underlying programming language is IF/Prolog. We also used a general purpose signal processing package SIG[1] for numerical routines (Lager & Azevedo, 1985), which were transformed into Prolog predicates and coupled with a symbolic processing (Obradović, 1991). FILTEX was developed in the Mihajlo Pupin Institute Knowledge Engineering Laboratory on a VAX 11/750 machine under VMS.

The next section briefly discusses the filter selection strategy and the need for an opportunistic reasoning approach. In Section 3 we describe the main components of the blackboard architecture in FILTEX, while in Section 4 the frame-based implementation is given. Section 5 is devoted to FILTEX validation and testing. In Section 6 an example of a session with the expert system is given. The main conclusions are outlined in Section 7.

Requests for reprints should be sent to Sanja Petrović, Knowledge Engineering Laboratory, Mihajlo Pupin Institute, P.O. Box 15, 11000 Belgrade, Yugoslavia.

[1] SIG is a trademark of Lawrence Livermore National Laboratory.

2. OPPORTUNISTIC REASONING IN FILTER SELECTION

The solution of the filtering problem in signal processing may be described briefly as the selection of an appropriate filter model that enables the elimination of noise from a measured signal (Anderson & Moore, 1979). Direct noise elimination can be achieved easily if the frequency spectra of the signal and noise can be separated by means of a shaping filter. If this is not the case, then a more sophisticated filter model, based on the analysis of data obtained by the measurements and the available information about the nature of the signal source, has to be designed.

Being a very complex problem, filter design requires an appropriate problem-solving strategy. Thus, in building FILTEX we started by developing an opportunistic filter design strategy. For practical purposes, the allowable filter models were restricted to the class of linear finite-dimensional parametric SISO (single-input single-output) models. Even so, the number of possible filter models for a given signal was practically infinite.

2.1. Filter Selection Strategy

We shall briefly illustrate, now, the salient features of the filter selection strategy. The adopted generalized model for a discrete signal in FILTEX is the *autoregressive moving average model with exogenous input* (ARMAX) (Candy, 1986):

$$A(q^{-1})y(t) = B(q^{-1})u(t) + C(q^{-1})e(t), \tag{1}$$

where A, B, and C are polynomials, and q represents the backward-shift operator with the property $q^{-k}y(t) = y(t-k)$. Then,

$$A(q^{-1})y(t) = y(t) + a_1 y(t-1) + \cdots + a_{N_a}y(t-N_a)$$

$$B(q^{-1})y(t) = b_0u(t) + b_1u(t-1) + \cdots + b_{N_b}u(t-N_b)$$

$$C(q^{-1})y(t) = e(t) + c_1e(t-1) + \cdots + c_{N_c}y(t-N_c),$$

where $y(t)$ stands for the discrete output signal, $u(t)$ for the exogenous input, and $e(t)$ for a white-noise source. The ARMAX model, usually abbreviated by ARMAX(N_a, N_b, N_c), represents the general form of the model. Other model forms are deduced from ARMAX. For example, autoregressive (AR) model is obtained for $B(\cdot) = 0$, $C(\cdot) = 1$, the moving average (MA) model for $A(\cdot) = 1$, $B(\cdot) = 0$, the autoregressive model with exogenous input (ARX) for $C(\cdot) = 1$, etc.

The modeling problem is viewed as the selection of the ARMAX model *structure* (N_a, N_b, N_c) and the estimation of the *parameters,* that is, polynomial coefficients to fit the data. The criterion for model evaluation in FILTEX is the final prediction error (Ljung & Söderstrom, 1983). The problem of ARMAX model structure selection is based mainly on the knowledge and experience of an expert in the field, while parameter estimation is made by recursive algorithms (Obradović, 1991).

FILTEX is conceived to solve the whole problem, that is, (1) to design the filter model by selecting the ARMAX model structure, and (2) to determine the corresponding parameters. The adopted strategic paradigm for filter design in FILTEX is the search for the least complex acceptable solution—the principle of parsimony. This search is subject to user constraints on model complexity ($s = N_a + N_b + N_c$) and/or final prediction error. To give the flavor of the strategy we shall describe in brief how it solves the case when both allowable complexity and acceptable final prediction error are limited by the user.

In an attempt to follow the principle of parsimony, the search starts within the classes of the least complex elementary models: (AR, MA, X). One model from each of the elementary classes is chosen, starting from the simplest model in the class and proceeding by increasing the model complexity until a local minimum for the final prediction error is reached. The models obtained in such a manner, which satisfy user constraints, represent the initial set of "model-candidates," that is, possible solutions. If no elementary model-candidates are found, more complex classes (ARMA, ARX, MAX, ARMAX) are searched using the same philosophy. If the set of model-candidates remains empty, the search proceeds in the "vicinity" of a model that satisfies one of the user constraints. For example, if a model with allowable complexity but unacceptable final prediction error exists, by lowering or increasing its complexity, a model with acceptable error might be found. Similarly, the lowering of model complexity for a model with acceptable final prediction error could lead to a model-candidate. However, if no model that satisfies at least one of the constraints is found, an exhaustive search is performed within the limits of allowable model complexity. If this final attempt does not succeed, the user is asked to relax the constraints imposed. If at any point in the aforementioned strategy at least one model-candidate appears, then an attempt to widen the set of model-candidates is made by lowering the complexity of the least complex model-candidate. Finally, the list of model-candidates is offered to the user with models sorted according to a desirable criterion: complexity, final prediction error, or variance.

2.2. Need for Opportunistic Reasoning

The initial experience with FILTEX indicated that the implementation of the filter selection problem-solving strategy could not be based on a single method of reasoning (e.g., backward or forward chaining), but required their opportunistic combination (Obradović, Krtolica, Petrović, & Radojević, 1991). The filter selection problem is characterized by a varying amount of available information and lots of intermediate results to which various algorithms and heuristics potentially can be applied. There is a need for a careful choice of action at each problem-solving step. Such an approach to the search for a solution that selects an action from the set of possible actions which will contribute to the solution most, and at the same time uses the most favorable method of reasoning, is referred to as *opportunistic problem solving* (Erman, Hayes-Roth, Lesser, & Reddy, 1980). Opportunistic problem solving is based on *opportunistic reasoning,* which is defined as reasoning combining backward and forward chaining according to the need of the problem solver.

3. COORDINATION OF HETEROGENEOUS KNOWLEDGE SOURCES

The blackboard concept was selected for system architecture in FILTEX, since it suits best the need for opportunistic problem-solving. This section outlines the design of the FILTEX blackboard system components: knowledge sources, blackboard, and control mechanism. The description that follows refers to Figure 1.

3.1. Knowledge Sources

The knowledge needed to solve the filter selection problem is partitioned into logically independent *knowledge sources* (KSs). FILTEX contains 33 KSs. They are organized into five groups: *Elementary MODels* (*EL_MOD*), *COMplex MODels* (*COM_MOD*), *SELECT, WIDEN,* and *IN_OUT,* and the problem-solving strategy is realized by their cooperative activity. KSs of the *EL_MOD* and *COM_MOD* groups propose models from the AR, MA, X and ARMA, ARX, MAX, ARMAX classes, respectively. These KSs use SIG numerical routines to estimate model parameters and calculate model performances: model complexity, final prediction error, and variance. These models are treated as hypotheses pertaining to the final solution and are examined further by KSs from the *SELECT* group, which selects models that satisfy user constraints imposed on complexity and final prediction error. KSs in the *WIDEN* group use heuristic rules in order to produce more model-candidates on the basis of the hypothesis set. Communication with the user is handled by the *IN_OUT* KS group.

3.2. Blackboard

The communication among KSs takes place exclusively through the *blackboard* (BB)—a global data base that holds all information related to filter selection. This information takes the form of attribute-value pairs. At each step of the problem-solving process, the state of the BB determines which KSs can contribute to the solution. The KS selected for activation takes all the current information from the BB and creates new information, changing the state of the BB.

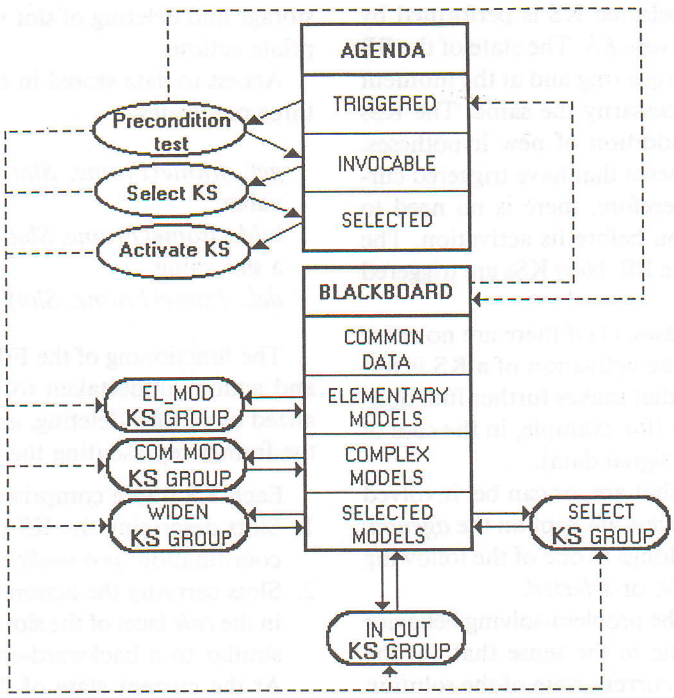

FIGURE 1. The blackboard architecture in FILTEX.

The outlined knowledge partitioning suggested a hierarchical organization of the BB. Each KS group has access to attributes on the corresponding level as well as to attributes on all higher levels. The hierarchical organization enables a faster and more transparent functioning of FILTEX.

3.3. Control Mechanism

The coordination of heterogeneous KSs is handled by a *control mechanism.* The control mechanism is realized through a loop where, in each problem-solving step, three control procedures are invoked sequentially: *Precondition test, Select KS,* and *Activate KS.*

The activation of a KS results in adding or deleting the values of selected BB attributes. These events trigger KSs that are potentially applicable in the next step of the problem-solving process. The reasoning process of a KS may need particular BB data. The existence of these data is a precondition for its activation. The fulfillment of preconditions is tested by the control procedure *Precondition test.* Only the triggered KSs that pass the precondition test can participate in the problem-solving process. These KSs are called "invocable." Activation of the remainder of triggered KSs is postponed until the BB reaches the appropriate state.

Constraints on the filter model, imposed by the user at the beginning of the FILTEX session, determine the strategy to be used in problem-solving. The adopted strategy is realized by assigning fixed priorities to the KSs. Priority assignment is done by the set of rules. The control procedure *Select KS* selects the KS with the highest priority from the set of invocable KSs.

The activation of the selected KS is performed by the control procedure *Activate KS.* The state of the BB at the moment of the KS triggering and at the moment of its activation is not necessarily the same. The KSs interact by incremental addition of new hypotheses. They do not delete hypotheses that have triggered currently invocable KSs. Therefore, there is no need to reevaluate KS precondition before its activation. The KS modifies the state of the BB. New KSs are triggered and the loop restarts.

The loop ends in two cases: (1) if there are no more invocable KSs, (2) when the activation of a KS is unsuccessful due to an error that makes further functioning of FILTEX impossible (for example, in the case of an unrecoverable error in signal data).

The names of the KSs that are, or can be, involved in the problem-solving process are kept on the *agenda.* Each KS on the agenda belongs to one of the following classes: *triggered, invocable,* or *selected.*

We may conclude that the problem-solving behavior in FILTEX is opportunistic in the sense that: (a) the knowledge applicable at a current state of the solution is represented through the set of invocable KSs, and (b) the KS with the highest priority contributes to the solution by the most profitable action at the moment.

4. FRAME-BASED IMPLEMENTATION OF BLACKBOARD ARCHITECTURE

The KSs and the BB can take various representation forms. Using frames as the basic building blocks enables representing both declarative and procedural information inherent to the BB system. A frame captures all available knowledge about an object in a single data structure, avoiding its distribution over production rules or logical formulae (Barr & Feigenbaum, 1981). A frame integrates declarative terms that describe the properties of the object and procedural terms for the representation of its dynamic behavior and the expertise from the domain of application.

The PROLOG implementation of the frame system suggested by (Merritt, 1989) fully satisfied all the requirements for FILTEX components. The internal data structure holding frame information is a predicate *frame(Name, Slot_list)* with two arguments, the first being the name of the frame and the second being the list of slots. Slots describe the properties—attributes of the object represented by the frame. The list of slots contains the values of slots and facets that describe the nature of these values. There are five possible facets: *val, def, calc, add,* and *del.* The *val* and *def* facets denote, respectively, the current and default value of the slot that is valid in the absence of other information. The *calc, add,* and *del* facets provide for an active nature of the frames. The procedure attached to the *calc* facet calculates the value of the slot upon request. Procedures attached to *add* and *del* facets monitor the storage and deleting of slot values and perform appropriate actions.

Access to data stored in the frames is controlled by three predicates:

get_frame(Frame, Slot-Value): retrieves a slot value,
add_frame(Frame, Slot-Value): adds or updates a slot value,
del_frame(Frame, Slot): deletes a slot value.

The functioning of the FILTEX control mechanism and actions undertaken to solve the problem are realized by setting, deleting, and asking for slot values of the frames representing the KSs and the BB.

Each KS frame comprises two groups of slots:

1. Slots describing the KS properties relevant to KSs coordination: *precondition* and *priority.*
2. Slots carrying the action of the KS. The procedure in the *calc* facet of the slot demonstrates the behavior similar to a backward-chaining inference process. At the current state of the solution, the action of the KS becomes the goal that has to be satisfied.

Therefore, the action of the KS is encapsulated in procedures denoted by the *calc* facets. The control mechanism activates the KS by asking for the values of these slots.

The BB is presented by a set of hierarchically organized frames. A frame inherits slot values of its parent frame on the higher hierarchical level.

The procedures denoted by *add* or *del* facets, which are activated when the slot value is changed, enable the forward chaining inference process. This property is used for KS triggering. Procedures in the *add* or *del* facets are attached to particular slots of the BB frames. Each procedure reports to the control mechanism which KSs are triggered after the KS action has changed the slot value.

The procedures attached to the slots enable alternative usage of the forward chaining realized by KSs triggering, and the backward chaining performed by KSs activations.

As an illustration, consider a KS named *x* that belongs to the *EL_MOD* KS group (Fig. 2). It proposes a model from the input-only X-model class. The KS *x* is triggered when the user enters the identification data store number containing the input signal. The procedure *trigger_KS* sets *x* to the *triggered* class of KSs kept on the agenda. Since the KS *x* needs no domain data from the BB, its precondition is always satisfied, that is, KS *x* is invocable as soon as it is triggered. In order to implement the adopted strategy, which gives the elementary models a preference, the KS *x* obtains a high priority. Asking for the values of the slots named in the *working_slot* causes the procedure *x_model* to be invoked. The proposed X-model becomes the value of the *el_models* slot and causes the triggering of the KS *list_el_models*. The B-polynomial order of the model is placed in the *x_nb* slot and will be used later when complex models are investigated by KSs *arx* and *max*.

The overhead cost of implementing the described control mechanism in FILTEX is not significant. It turned out that the FILTEX control mechanism requires less than 10% of total processing time consumed by KSs actions. It means that the time spent for the coordination of KSs is noticeably smaller than the time used for filter selection activities.

5. VALIDATION AND TESTING

FILTEX builders have been validating their system continually in every stage of the development. A lot of attention is paid to adequacy of the knowledge representation applied, consistency and completeness of embedded knowledge, correctness of reasoning, accuracy, and ease of interaction with the system. FILTEX modularity simplified validation process.

FILTEX successfully passed all feasibility demonstrations and has reached the stage of more formal testing. We ran the program on a large and representative sample of test cases. The experiments were performed in a fully controlled experimental environment. The signals submitted to FILTEX were generated using the simulation facilities offered by the underlying signal processing package SIG. The model of the system generating the signals, the input applied, and noise were known for each test example, and thus results of the expertise performed by FILTEX could be validated

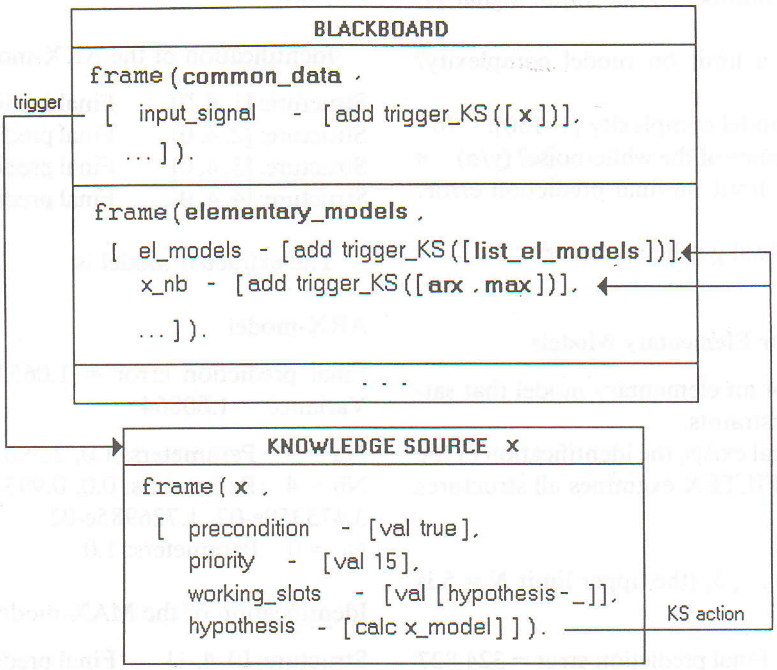

FIGURE 2. The interaction of the KS x and BB frames.

precisely. The experiments were performed on selected sets of 64 signals, produced by 40 different models, two possible input signals: pseudorandom binary sequences with zero mean and variances 1 and 10: PRBS(0; 1), PRBS(0; 10), the noise being always Gaussian white with variance 1. In each test set FILTEX produced satisfactory solutions in 98.4% test examples on an average. Statistics collected show that FILTEX could be considered worthy of the name expert system.

6. EXAMPLE OF A FILTEX SESSION

In order to illustrate the way FILTEX performs its expertise, an example of a session through which a filter model is selected by the system is given. We offered to FILTEX a signal generated by an ARMAX(4, 3, 1) model with a pseudorandom binary sequence PRBS(0; 10) as input signal and discrete white noise with a normal distribution $N(0; 1)$ added. The limit of the model complexity was set to 16 (twice the original $s = 4 + 3 + 1$), while the final prediction error limit was 2 (twice the noise variance). The FILTEX session is composed of seven parts, which will be presented with comments. User input is shown in italic.

6.1. Part 1. Data Entry

WELCOME TO FILTEX

EXpert System for FILTer Design

Enter the data store number of the output signal [1, 899]: *255*
Does the input signal exist? (y/n) *y*
Enter the data store number of the input signal [1, 899]: *2*
Do you want to set a limit on model complexity? (y/n) *y*
Enter the allowable model complexity [1–150]: *16*
Do you know the variance of the white-noise? (y/n) *n*
Do you want to set a limit on final prediction error? (y/n) *y*
Enter the acceptable final prediction error: *2*

6.2. Part 2. Search for Elementary Models

FILTEX is looking for an elementary model that satisfies the imposed constraints.

Since the input signal exists, the identification of the X-model is possible. FILTEX examines all structures from the set:

{[0, Nb, 0], Nb = 1, . . . , 5, (the upper limit $N = 5$ is heuristics)}.

Structure: [0, 1, 0] Final prediction error = 324.827
Structure: [0, 2, 0] Final prediction error = 309.446
Structure: [0, 3, 0] Final prediction error = 299.133
Structure: [0, 4, 0] Final prediction error = 297.616
Structure: [0, 5, 0] Final prediction error = 299.791

The extracted model is:

X-model

Final prediction error = 297.616 Complexity = 4
Variance = 288.458

Na = 0 Parameters: 1.0
Nb = 4 Parameters: 0.0, 1.13513, −1.31722, 1.09389, −0.61979
Nc = 0 Parameters: 1.0

6.3. Part 3. Examination of the Elementary Models

The elementary model is:

X [0, 4, 0] Final prediction error = 297.616 Complexity = 4

Models with allowable complexity are:

X [0, 4, 0] Final prediction error = 297.616 Complexity = 4

There are no models with acceptable final prediction error!

6.4. Part 4. Search for Complex Models

B-order (4) for ARX (MAX) model is taken from the identified X-model. A-order (C-order) is increased until a local minimum for the final prediction error is reached.

Identification of the ARX-model.

Structure: [1, 4, 0] Final prediction error = 20.1538
Structure: [2, 4, 0] Final prediction error = 1.68501
Structure: [3, 4, 0] Final prediction error = 1.06535
Structure: [4, 4, 0] Final prediction error = 1.07011

The extracted model is:

ARX-model

Final prediction error = 1.06535 Complexity = 7
Variance = 1.00864

Na = 3 Parameters: 1.0, 2.39056, 1.96248, 0.54938
Nb = 4 Parameters: 0.0, 0.993485, 1.06007, 3.475359e-02, 1.726985e-02
Nc = 0 Parameters: 1.0

Identification of the MAX-model.

Structure: [0, 4, 1] Final prediction error = 85.606
Structure: [0, 4, 2] Final prediction error = 342.785

The extracted model is:

MAX-model

Final prediction error = 85.606 Complexity = 5
Variance = 82.3261

Na = 0 Parameters: 1.0
Nb = 4 Parameters: 0.0, 1.16953, −1.43925, 1.36222, −0.737844
Nc = 1 Parameters: 1.0, −0.903006

FILTEX applies heuristics:

IF C-order of the MAX-model is small (1)
THEN there is no need to search for the ARMAX-model.

6.5. Part 5. Examination of the Complex Models

The set of complex models is:

MAX [0, 4, 1] Final prediction error = 85.606 Complexity = 5
ARX [3, 4, 0] Final prediction error = 1.06535 Complexity = 7

Models with allowable complexity are:

MAX [0, 4, 1] Final prediction error = 85.606 Complexity = 5
ARX [3, 4, 0] Final prediction error = 1.06535 Complexity = 7
X [0, 4, 0] Final prediction error = 297.616 Complexity = 4

Models with acceptable final prediction error are:

ARX [3, 4, 0] Final prediction error = 1.06535 Complexity = 7

6.6. Part 6. Widening the Set of Model-Candidates

Since the set of models with allowable model complexity and acceptable final prediction error is not empty (ARX [3, 4, 0]), FILTEX is trying to widen it with less complex models.

Structure: [2, 4, 0] Final prediction error = 1.68501 (identified before)
Structure: [3, 3, 0] Final prediction error = 1.05913

Model with the smallest final prediction error is added to the set:

ARX-model

Final prediction error = 1.05913 Complexity = 6
Variance = 1.01062

Na = 3 Parameters: 1.0, 2.40715, 1.99473, 0.565617
Nb = 3 Parameters: 0.0, 0.994465, 1.07693, 4.312419e-02
Nc = 0 Parameters: 1.0

The complexity lowering of the structure [3, 3, 0] is performed.

Structure: [2, 3, 0] Final prediction error = 1.97827
Structure: [3, 2, 0] Final prediction error = 1.05919

Model with the smallest final prediction error is added to the set:

ARX-model

Final prediction error = 1.05919 Complexity = 5
Variance = 1.01861

Na = 3 Parameters: 1.0, 2.37148, 1.92202, 0.526481
Nb = 2 Parameters: 0.0, 0.991932, 1.04001
Nc = 0 Parameters: 1.0

The complexity lowering of the structure [3, 2, 0] is performed.

Structure: [2, 2, 0] Final prediction error = 2.7681
Structure: [3, 1, 0] Final prediction error = 4.75411

Final prediction error of the models surpassed the allowable limit!

Lowering of the complexity stops!

6.7. Part 7. Models Ranking

User chooses the criterion for model ranking. FILTEX produced three possible solutions, all of them with the structure close to the original model, and what is more important, an error variance close to the variance of the noise, which is considered to be optimal (Ljung & Söderstrom, 1983).

CHOOSE ONE OF THE CRITERIA FOR THE RANKING OF MODELS OR EXIT FILTEX.

1. Final prediction error
2. Model complexity (Na + Nb + Nc)
3. Variance of the residual sequence
4. Exit FILTEX

ENTER OPTION [1 − 4]: 1

MODELS RANKED ACCORDING TO FINAL PREDICTION ERROR:

ARX-model

Final prediction error = 1.05913 Complexity = 6
Variance = 1.01062

Na = 3 Parameters: 1.0, 2.40715, 1.99473, 0.565617
Nb = 3 Parameters: 0.0, 0.994465, 1.07693, 4.312419e-02
Nc = 0 Parameters: 1.0

ARX-model

Final prediction error = 1.05919 Complexity = 5
Variance = 1.01861

Na = 3 Parameters: 1.0, 2.37148, 1.92202, 0.526481
Nb = 2 Parameters: 0.0, 0.991932, 1.04001
Nc = 0 Parameters: 1.0

ARX-model

Final prediction error = 1.06535 Complexity = 7
Variance = 1.00864

Na = 3 Parameters: 1.0, 2.39056, 1.96248, 0.54938
Nb = 4 Parameters: 0.0, 0.993485, 1.06007, 3.475359e-02, 1.726985e-02
Nc = 0 Parameters: 1.0

7. CONCLUSIONS

The blackboard architecture offered a suitable mechanism for developing a knowledge-based and flexible strategy in the FILTEX expert system, which solves the complex problem of filter design in signal processing. The idea of upgrading a commercial signal processing package SIG by adding expert knowledge and combining both symbolic and numerical processing has shown good results. FILTEX is able to solve the whole problem of filter design by selecting an appropriate filter structure and determining its parameters. Frames implemented in IF/Prolog on a VAX machine running under VMS have been successfully used as the building blocks in the FILTEX blackboard architecture.

The special features offered by the blackboard architecture are fully exploited: (a) complex knowledge was represented in a modular way, by partitioning into a number of knowledge sources, and thus can easily be extended and tuned, (b) experiments with different problem-solving strategies were performed by simply changing the priorities of knowledge sources.

The signal processing experiments performed with FILTEX on a representative set of examples proved the viability of the concept and the effectiveness of its implementation.

Acknowledgements—The authors are very grateful to Dr Radovan Krtolica of the Mihajlo Pupin Institute, now on leave at Canon Research Laboratories, Palo Alto, CA, who inspired and coordinated the Knowledge Engineering Laboratory team during the research presented in this paper.

REFERENCES

Anderson, B.D.O., & Moore, J.B. (1979). *Optimal filtering.* Prentice-Hall, Englewood Cliffs, N.J.

Barr, A., & Feigenbaum, E.A. (1981). *The handbook of artificial intelligence* (Vol. I). London: Pitman Books Ltd.

Candy, J. (1986). *Signal processing: The model-based approach.* New York: McGraw-Hill Book Company.

Engelmore, R., & Morgan, T. (Eds.). (1988). *Blackboard systems.* Reading, MA: Addison-Wesley.

Erman, L. D., Hayes-Roth, F., Lesser, V.R., & Reddy, D.R. (1980). The Hearsay-II Speech-Understanding System: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys,* **12**(2), 213–253.

Krtolica, R., Obradović, I., & Božinović, R. (1991). FILTEX: Toward an expert system for time-series filter design. *Journal of Applied Statistics, Special Issue on Expert Systems and Statistics,* **18**(1), 121–137.

Lager, D., & Azevedo, S. (1985). *SIG, a general purpose signal processing program.* User's manual, Lawrence Livermore National Laboratory, UCID-19912-Rev 1, Livermore, CA.

Ljung, L., & Söderstrom, T. (1983). *Theory and practice of recursive identification.* Cambridge, MA: The MIT Press.

Merritt, D. (1989). *Building expert systems in Prolog.* New York: Springer-Verlag.

Nii, P. H. (1986). The blackboard model of problem solving and the evolution of blackboard architectures (Part 1). *AI Magazine,* **7**(2), 38–53.

Obradović, I. (1991). *An approach to symbolic upgrading of numerical time-series processing.* Unpublished Ph.D. Thesis, University of Belgrade, Yugoslavia (in Serbian).

Obradović, I., Krtolica, R., Petrović, S., & Radojević, D. (1991). An opportunistic reasoning approach to intelligent time-series modeling. In R. Hanus, P. Kool and S. Tzafestas (Eds.), *Mathematical and intelligent models in system simulation* (pp. 331–336). Basel: J. C. Baltzer AG, Scientific Publishing Co.