ARTIFICIAL INTELLIGENCE IN SCIENTIFIC COMPUTATION:
Towards second generation systems
R. Huber et al. (editors)
J.C. Baltzer AG, Scientific Publishing Co. © IMACS, 1989
pp. 337-342

ADAPTIVE CONTROL WITH A SUPERVISOR LEVEL USING A RULE BASED INFERENCE SYSTEM WITH APPROXIMATE REASONING

Rafael SANZ[1], Joseph AGUILAR MARTIN[2], Carles SIERRA[3], Lluis GODO[3] and Anibal OLLERO[1]
[1] *Departamento de Ingenieria Eléctrica y de Computadores y Sistemas, Universidad de Santiago, E.T.S.I.I., Ap. 62, 36280 Vigo, Spain*
[2] *L.A.A.S.-C.N.R.S., 7, Avenue du colonel Roche, 31077 Toulouse Cedex, France*
[3] *Grup de Recerca en Intel.ligència Artificial i Lògica, Centre d'Estudis Avançats C.S.I.C., Camí de Santa Bárbara s/n, 17300 Blanes, Spain*

ABSTRACT

Practical design and application of self-tuning controllers have shown that, besides the parameter estimation and control algorithms, a significant amount of heuristics logic must be considered. Moreover self-tuning controllers must also be able to handle changes in environment conditions as well as pathological behaviour of the estimators, caused by numerical and informational problems in recursive parameter estimation algorithms. In all real process implementations the existence of supervisory functions considered in order to avoid unacceptable closed-loop behaviour can be observed. In this paper we formalize this supervisor level and adopt a rule based inference system. This expert supervisor must cope with both precise and imprecise data and as well as with heuristic decision rules, the validity of their application being partially known. Simulations with an expert system shell, managing uncertainty and approximate reasoning, are described in this paper.

1. INTRODUCTION

It is well known that advanced Control Theory has only had so far a low echo on industrial automation although it yields a great amount of sound theoretical solutions to complex problems in the field of Stochastic and Adaptive Control, Coordination of Interconnected Systems, and Optimisation of well defined cost dependent problems. Unfortunately those solutions stay mostly on research reports and seldom reach an operative situation in the context of the control and regulation of industrial plants.
In this paper an attempt to configurate an elementary adaptive regulation situation is described. It contains a rule based supervisor level built using an expert system shell, having the particular feature of accepting approximate knowledge and imprecise data. After showing through this modular experiment that such a structure can work, and that it considerably improves classical adaptive regulation schemes, a further work should take into account the complete plant model by adjoining simulation modules and inserting a monitoring level between the expert system and the classical control algorithms: regulation, identification...
Most supervisory functions are based on the designer's expertise to handle heuristic knowledge. Artificial Intelligence techniques happen to be particularly suitable tools. Knowledge based Expert Systems have achieved a great success with some practical problems such as medical diagnosis, computer fault analysis, mineral exploration... Nevertheless only recently have those techniques rised some interest in Real Time Control. Sometimes an attempt has been made to replace controllers by rules. In our opinion this may lead to a misunderstanding of the role of the expertise with respect to Control problems, thus bypassing all the efficient real-time procedures that Control Theory can afford. Most of the reasoning operations of Artificial Intelligence must be a help tool for the control device supervision level taking advantage of the fast and well known abilities of closed loop control systems for driving continuous or sampled processes. Expert systems have already been proposed for

the supervision of controllers in Aström and Anton [1], Samaan [2], Sanoff and Wellstead [3], and Sanz [4].

2. GENERAL PRINCIPLES OF SUPERVISED CONTROL

The aim of this work is to show an on-line supervised process built using some updated tools of Artificial Intelligence, connected to Adaptive Control Algorithms in order to adjust the parameters and choose the operating methodologies.
Between the process and the human operator, four levels are to be considered they are represented in the diagram of figure 1, they are grouped in two main tasks:

I) CONTROL: Control devices give numerical or analogic informations to be processed in procedural routines. Direct control actions are taken after computing in real time (immediate response).

II) SUPERVISION: This task manipulates symbolic information used in declarative programmed units. Decision making responses are produced after in human-like reasoning time (response after reflexion delay).

HUMAN OPERATOR		
LEVEL 3	Artificial Intelligence: Operator dialogue and Expert Supervision	Supervision
LEVEL 2	Monitoring and Evolutive Data Base	Supervision
LEVEL 1	Automata, regulators and optimizers	Control
LEVEL 0	Sensors and Actuators	Control
PROCESS PLANT		

Figure 1

3. NUMERICAL-SYMBOLIC TRANSLATION

In the several attempts for an Intelligent Supervised Control mentionned above, it appears clearly that one of the main of the dialogue difficulties between Control Algorithms (classical or advanced) and Knowledge Based Systems is due to the fact that Artificial Intelligence is better suited to deal with symbolic data, whereas procedural methods in Control work uniquely with numerical data. Moreover, the description of engineers knowledge about suitable methodologies does not deal only with non dichotomic situations, and is, usually, corrupted by uncertainty and . Therefore it appears that a Numerical-symbolic interface, or **Linguistic Evaluator**, must be included between the control procedures and the expert supervisor. The principle of the numerical-symbolic interfaces is illustrated in figure 2, where the measured control variable is converted into linguistic variables with certainty coefficients:

"Output error = x" ↔
"Output error is HIGH (almost sure),
Output error is MEDIUM (possible),
Output error is LOW (impossible)"

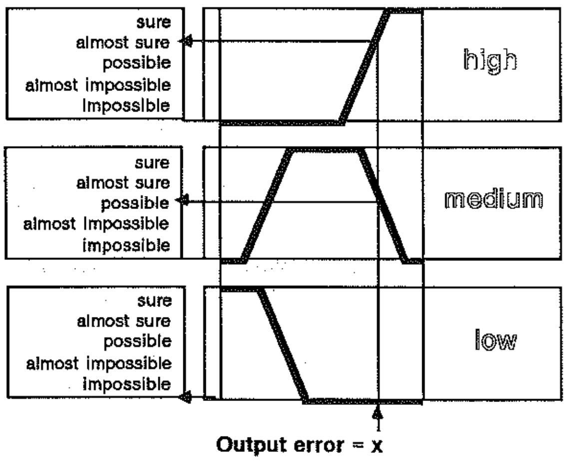

Figure 2

This device has to perform the two following functions:

1) Linguistic evaluation of the quality of the present behaviour and state of the system.
2) Decision translation of the supervision expert system answer into control setting values and parameter adjustment.

In the following example, it can be seen that the attribution of linguistic labels is accompanied by a "truth level" or "likelihood coefficient" that is also a linguistic label. The translation mechanism can be illustrated as follows:
This evaluator can be driven by the Human Operator in order to be adapted to different situations, as for example *start-up*, *test*, *normal-operation*... Another necessary interface of the

same nature is needed to translate the decisions or advices of the expert system into control actions. Those actions can be either quantities, as the value of a parameter, or logical as the switching of a procedure. The **Decision Translator** is a Symbolic-numerical converter, and shares the same principles as the Linguistic Evaluator.
Finally it shall be pointed that the Adaptive Control procedures chosen here assume a heuristic extension of the Certainty Equivalence Principle by assuming that an acceptable control is reached by the estimating parameters of a reduced order lineal model of the plant, followed by a suitable pole-assignement procedure. The following diagram represents the standard structure of a supervised regulated system and locates the modules.

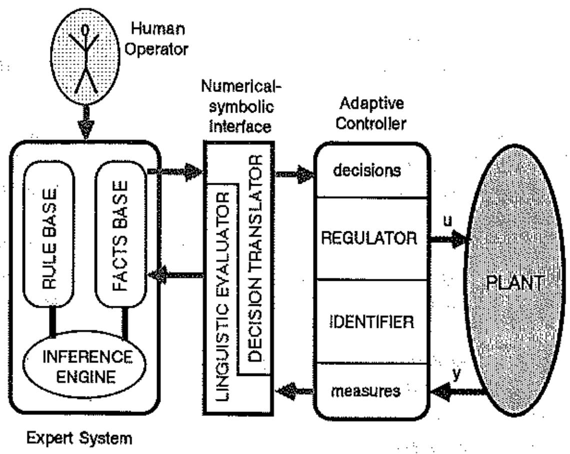

Figure 3

4. EXPERT SYSTEM CONFIGURATION

The Supervision task is to be performed by an Expert System. It must have facilities to handle approximate reasoning, different search procedures, explicit control representation (e.g. meta-rules) and time reasoning management.
In a first implementation Sanz and Ollero [5], a simple two layer expert system was designed: the first layer concerned rules using uniquely primary information and producing intermediary facts used by the selection layer as shown in figure 4.

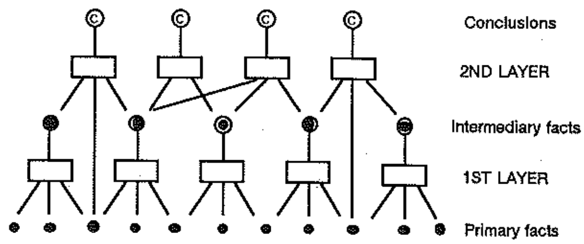

Figure 4

The program was implemented in PASCAL and no uncertainty representation was provided. Taking advantage of the experience acquired by that first implementation, it has been possible to implement the expert system on the shell MILORD (Godó, López de Mántaras, Sierra and Verdaguer [6].
MILORD is an expert systems building tool consisting of two inference engines and an explanation module. The system allows to perform different calculi of uncertainty on an expert defined set of linguistic terms expressing uncertainty. Each calculus corresponds to specific conjunction, disjunction, and implication operators. The internal representation of each linguistic uncertainty value is a fuzzy subset of the interval [0,1]. The different calculi of uncertainty applied to the set of linguistic terms, give, as a result, a fuzzy subset that is approximated, by means of a linguistic approximation process, to a linguistic certainty value belonging to the set of linguistic terms. This linguistic approximation keeps closed the calculus of uncertainty. This has the advantage that, once the linguistic certainty values have been defined, the system computes, off-line, the conjunction, disjunction and implication operations for all the pairs of linguistic uncertainty values in the term set, and stores the results in matrices, see fig. 5. Therefore, when MILORD is run, the prop-

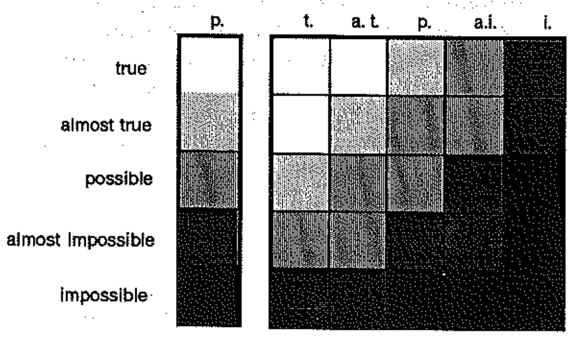

Figure 5

agation and combination of uncertainty is performed by simply accessing these precomputed matrices. MILORD also deals with non-monotonic reasoning in the same framework of uncertainty management. The architecture of MILORD presents a multilevel structure in both the domain representation elements and the control representation. There is a relation between these two representations as can be seen in figure 6.

	Domain Level		Control Level
Heuristical Level	Strategies	←	Metarules over strategies
Solution Level	Modules	←	Metarules over modules
Inferential Level	Rules	←	Metarules over rules
Conceptual Level	Facts	←	Semantic Network

Figure 6

The graph acts over the facts, structuring their interdependencies and controlling mechanisms such as semantic subsumption between concepts.
The metarule hierarchy controls the application of their corresponding domain knowledge. Before moving down in the domain level hierarchy (from strategies to rules and facts), the corresponding control level is consulted.
This separation between different levels of control and of domain representation matches quite well various design processes in KB definition:

1.- It allows a top-down design of the KB, because the decisions about domain knowledge and the corresponding control items are independent of the later refinements. In the case of strategy definitions and resolution of conflicts between strategies, for example, one can first define strategies based only on domain knowledge and later define how to solve the possible conflicts between them.
2.- It allows defining different problem solving methods, due to the flexibility at the operational level. Thus, one can define, for example, simple classification or heuristic classification processes.
3.- It allows defining different hierarchical structures: mixed hierarchies or pure hierarchies in the domain knowledge, and the corresponding control level that monitorizes the use of the different strategies.

5. APPLICATION TO EXPLICIT ADAPTIVE CONTROL

The Certainty equivalence principle is supposed to hold in such a way that the adaptive control scheme is obtained by an Identifier followed by a Controller. Let us consider the classical sample data linear model:

$A(z)\,y = B(z)\,u + C(z)\,e$ dynamics $\quad y(t) = h^T(t)\,\theta(t) + \eta(t)$ observation

$\eta(t)$ coloured noise

And the feedback regulator is given by:

$H(z)u=G(z)(y-y_{ref})$

where **u** is taken as output and the measured error between the output and the reference $y\text{-}y_{ref}$ is the input.
The general pole placement procedure consists of choosing a given polynomial $T(z)$, having the desired poles, and to solve the so-called Diophantine equation to find the polynomials G and F. The expertise of the control engineer consists of choosing T according to the identified system, or to impose G and H in particular circumstances as *start-up, emergencies,...*
The estimation is realized by means of least-square type algorithms.
The working vectors and the correspondent estimator equations are:

$$h^T(t) = [\, -y(t-1), -y(t-2), \ldots, -y(t-n_y), u(t-1), u(t-2), \ldots, u(t-n_u)\,]$$

$$\theta = [\, a_1 \; a_2 \ldots a_{n_y} \; b_0 \; b_1 \; \ldots \; b_{n_u}\,]$$

Prediction error:

$$e(t) = y(t) - h^T(t)\,\theta(t-1)$$

Estimator updating recurrent equation:

$$\theta(t) = \theta(t-1) + K(t)\,\varepsilon(t)$$

The estimator equation is given by:

$$\theta(t+1) = \theta(t) + \alpha(t)\ M(t)\ h(t)\ \varepsilon(t+1)$$

$$M(t) = M(t-1) - \frac{M(t-1)\ h(t)\ h^T(t)\ M(t-1)}{\lambda(t) + h^T(t)\ M(t-1)\ h(t)} + Q(t)$$

$\alpha(t)$ and $\lambda(t)$: forgetting factors; $Q(t)$: fluctuation;

$\theta(0)$ and $M(0)$: initialisation.

Supervisor tasks: initialization of working vectors, initialization, choice of forgetting factors.

Variables to be observed:
The System output variables are to be observed in order to evaluate the quality of the behaviour: error: $e(t)=(y(t)-y_0)$, speed $s(t)= dx/dt$, variability $v(t) = \Sigma_{(t-H)} e(t)^2$.

The Regulator variables chosen to indicate the quality of control are the value of the control, $u(t)$, its variability $\tilde{\delta}(t) = \Sigma_{(t-H)} u(t)^2$ and the margin $\tilde{a}(t)=\mathbf{min}\ [u(t)-u_{min},\ u(t)-u_{max}]$.

The Identifier is evaluated after its precision index, or generalized covariance matrix trace: $\sigma(t)=$**trace [M(t)]**, the estimated value $\theta(t)$, and its variability $v(t)= \Sigma_{(t-H)} \theta(t)^2$.

Finally the closed loop system can be evaluated from the estimated model and the chosen controller by its pole-zero configuration.

6. CONTROLLER ALGORITHM

The evaluation variables are stored in a *fact base*, whereas the *knowledge base* houses all rules concerning the use of the different possibilities of adjustment and choice of algorithms.
A meta-control mechanism is implemented in the form of *Contexts* for every type of running situations; the more elemental discrimination is the difference between **Normal Operation** and **Starting-up** *contexts*. The set of rules to be activated and the covering of the linguistic labels may be different for each *context*, e.g. the output error can be considered **medium** and thus acceptable, in the **Starting-up** *context* whereas it could be called **very high**, so unacceptable, in **Normal Operation**.
As an example the following rule can be implemented:

IF
[[σ(t) IS high] IS sure]
AND
[[v (t) IS low] IS AT LEAST possible]
AND
[[e(t) IS high] IS AT LEAST possible]
AND
[[v(t) IS AT LEAST medium] IS sure]
THEN :
make **[a(t) = 1]**
AND
re-initialize **[M(0) IS high]**

The underlined linguistic variables are translated numerical quantities whereas not underlined are truth values.
Logical operators are IF, IS, AND, IS AT LEAST, THEN, =.
Actions are writen ***in bold italics***, and propositions are between square brackets [].

7. EXPERIMENTAL IMPLEMENTATION AND RESULTS

The above concepts for merging automated reasoning and adaptive control techniques have been applied to a solar power plant simulation model located in southern Spain. This plant is fully described in Rubio, Sanz, Camacho and Ollero [7]. The main particularities of this model are its variability with respect to the uncontrolled input solar radiation, submitted to great perturbations when clouds appear. The amplitude of these perturbations may cause great problems in the parameter estimation because the gain of the system may be interpreted by the algorithm as changing its sign.
The final practically implemented program is given by the diagram of figure 7. It has been implemented on a VAX system under VMS. The simulation of the plant and the control and identifivation mechanism are PASCAL packages, the Expert System has been built with MILORD in a standard version implemented in semi-compiled VAX-LISP. The communi-

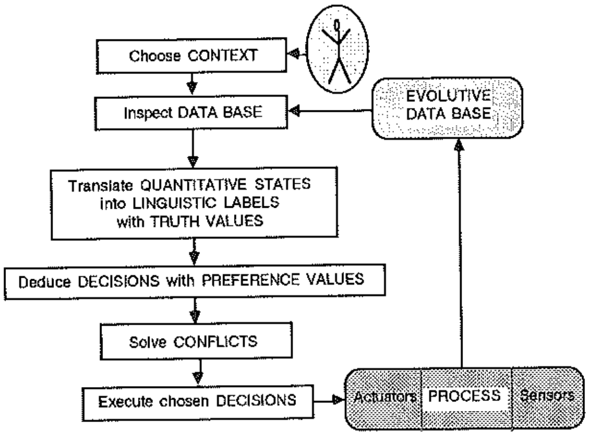

Figure 7

cation mechanism is a shared file of mailbox type called alternatively by the supervisor and the controller.
A sample of the output, in both an supervised and a supervised run, is shown in figure 8, as well as the decisions taken by the supervisor.

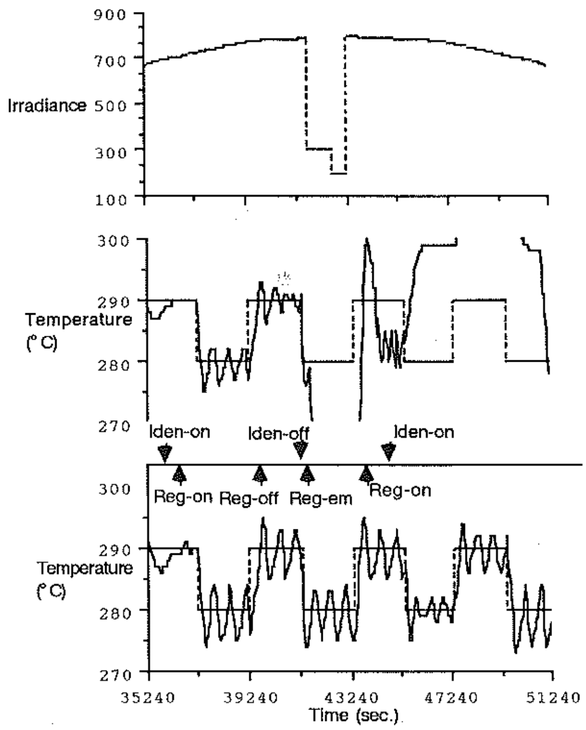

Figure 7

8. CONCLUSIONS

The feasibility of the supervisor level using automated reasoning with representation of uncertain knowledge has proved to be as an useful tool to improve adaptive regulation. It is the only way for applying control devices to a complex and model changing system.
Workstations using the facilities of Artificial Intelligence programming environments will be the next step to practically implement supervision in real industrial processes. Important characteristics, present in MILORD environment, is the management of imprecision and uncertainty, as well as the capability of giving information about its reasoning. Time events management should be the object of further research in Artificial Intelligence applied to Process Control.

REFERENCES

[1] Aström, K.J. and Anton, J.J., Expert Control, Proceedings IFAC 9th World Congress, vol VI, pp. 240-245, Budapest, 1984.

[2] Samaan, M., Station de travail et systèmes experts en supervision, Rapport DEA d'Automatique, LAG, Grenoble, 1985.

[3] Sanoff, S.P. and Wellstead, P.E., Expert identification and control, Proceedings IFAC Int. Symp. on Identification and Systems Parameter Estimation, pp. 1046-1051, York, 1985.

[4] Sanz, R., Diseño de Controladores Autoajustables. Nivel Supervisor Experto, Tesis doctoral, E.T.S.I.I. de Vigo, Univ. de Santiago, 1987.

[5] Sanz, R. and Ollero, A., A rule based inference method for supervision of self-tuning controllers, Proceedings IFAC Int. Symp. on Low Cost Automation, pp 43-48, Valencia, 1986.

[6] Godó, L., López de Mántaras, R., Sierra, C. and Verdaguer, A., Managing linguistically expressed uncertainty in MILORD. Application to medical diagnosis, AI Communications, Vol 1-1, 1988.

[7] Rubio, F.R., Sanz, R., Camacho, E.F., and Ollero, A., Studies on expert self-tuning control of distributed collector fields, Proceedings IMACS-88 Symp., pp 231-236, Barcelona, 1987.