A Real Time Control System based on a Fuzzy Compiled Knowledge Base

VASILE MAZILESCU
Department of Economic Informatics
“Dunarea de Jos” University of Galati, 47 Domneasca Street
vasile.mazilescu@ugal.ro

Abstract: - This paper presents the predictability problem in a real-time expert control system. It is a special possibilistic expert system (FESPC), developed in order to focus on fuzzy knowledge. We are not dealing high with level reasoning methods, because we think that real-time problems can only be solved by rather low-level reasoning. When humans engage in problem solving, the qualitative aspects of knowledge are hierarchically organized to provide concept association and reasoning, based on fuzzy logic inference. Most of the overall run-time of fuzzy expert systems is used in the match phase. To achieve a fast reasoning the number of fuzzy set operations must be reduced. For this, we use a fuzzy compiled structure of knowledge, like Rete, because it is required for real-time responses. Solving the match-time predictability problem would allow us to built much more powerful reasoning techniques.

Key-Words: Compiled fuzzy model, real-time, fuzzy reasoning, expert control, FESPC

1 Introduction

Expert systems were introduced as an intelligent tool for diagnosis and it is now widely used in classification and control tasks in a variety of human activity fields. Fuzzy logic is an attempt to capture valid reasoning patterns about uncertainty.

In addition to modelling the gradual nature of properties, fuzzy sets can be used to represent incomplete states of knowledge. In general, a more complex model may provide the capability to obtain a better representation of a system and may facilitate design, but it may not lend itself to straightforward analysis. If a simpler model is used, one may ignore some of the dynamical behaviour of the plant (problem domain) and be able to get more analytical results, but such results may only be valid in an approximate way for the real system. There will be different analysis techniques that are appropriate for different models (conventional, discrete event models, distributed architectures etc.).

The aim of this paper is to present the predictability problem in a real-time expert control system. It is a special possibilistic expert system, developed in order to focus on fuzzy knowledge. In this approach we are not dealing high with level reasoning methods, because we think that real-time problems can only be solved by rather low-level reasoning. Section 2 presents the reasoning algorithm of fuzzy compiled rules and in Section 3 the main properties of **FESPC** (a Fuzzy Expert System based on the use of the Possibility theory for expert Control) are stated and the analogy between expert and classical control systems and a qualitative analysis of possibilistic expert control systems. Section 4 makes concluding remarks.

2 The fast reasoning of fuzzy compiled rules

The fuzzy logic inference plays an important role in human intelligent activities. When humans engage in make decisions, the approximate, qualitative aspects of knowledge are hierarchically organized to provide concept association and reasoning. By using a compiled structure of a fuzzy rule-base, the reasoning process is efficiently and fast performed.

All works related to decision-making under fuzziness stem from Bellman and Zadeh [1] framework. Its basic elements are: the fuzzy goal FG in X, the fuzzy constraints FC in X and the fuzzy decision FD in X; X is a (non-fuzzy) space of decision (alternatives). Before we describe how to improve control, we must describe what it means to improve; in other words, we must chose a characteristic function f that will describe to what extent a control or a decision is good. It may be time, it may be cost, it may be fuel consumption.

The general decision-making problem formulation: given a (crisp) function $f : X \to R$ and a fuzzy set $FC \subseteq X$, to find $x \in X$ for which $f(x) \to \max_{x \in FC}$. What is given can be easily formalized. By a maximization problem under fuzzy constraints FC we mean a pair (f, FC), where f is a (crisp) function from a set X into the set R of all real numbers, and FC $\subseteq$ X is a fuzzy subset of X. Generally speaking, there are two possibilities here:

a) In decision making, what we want is some help for a decision maker. Therefore, we want the computer to produce several possibly optimal solutions, with the corresponding degree of possibility optimal. In fuzzy terms, we want a membership function $\mu_{FD}(X)$ that describes an optimal solution; **b**) In control, we want an automated device that controls without asking a human operator every time; in this case, we would prefer a number x. Notice that if f: X $\rightarrow$ R is a conventional objective (performance) function, then

$$\mu_{FG}(x) = f(x) / \sup_{x} f(x)$$

is a plausible choice provided that $0 \neq \sup_{x} f(x) < \infty$;

so, the fuzzy decision-making framework considered may therefore be viewed as a generalization of the conventional one. We wish to satisfy FC and attain FG which leads to fuzzy decision $\mu_{FD}(X)= \mu_{FC}(X) \wedge \mu_{FG}(X)$ which yields the "goodness" of an $x \in X$ as a solution to the decision-making problem considered from 1 for definitely perfect to 0 for definitely unacceptable, through all intermediate values. The "$\wedge$" (minimum) operation is commonly used. It is by no means the only choice, and may be replaced any t-norm or am suitable operation. For an optimal (non-fuzzy) solution to this problem, an x* $\in$ X such that

$$\mu_{FD}(x^*) = \sup_{x \in X} \mu_{FD}(x) = \sup_{x \in X} (\mu_{FC}(x) \wedge \mu_{FG}(x))$$

is a natural (but not the only possible) choice. In the general setting assumed here we have a deterministic system under control, whose dynamics is described by a state transition equation

$$x_{t+1} = f(x_t, u_t), \; t=0.1,\ldots,$$

where x_t, $x_{t+1} \in X = \{x\} = \{s_1,\ldots,s_n\}$ are the states at time (control stage) t and t+1, respectively, and $u_t \in U = \{u\} = \{u_1,\ldots,u_m\}$ is the control (input) at t; X and U are assumed finite.

At each t, u_t is subjected to the fuzzy constraints $\mu_{FC\,t}(u_t)$ and a fuzzy goal $\mu_{FG\,t+1}(x_{t+1})$ is imposed on x_{t+1}. The performance of the multistage decision - making (control) process is evaluated by the fuzzy decision which is assumed to be a decomposable fuzzy set. It may readily be seen that this general formulation max be viewed as a starting point for numerous extensions (our aim is the conditional optimization problem in terms of compiled fuzzy if-then rules).

An important application of the fuzzy logic inference refers to the problem of possibilistic and temporal reasoning in real-time fuzzy expert systems [2, 4, 6].

Let $s_0 \in U$ denote the unknown current state of a process under consideration. U may be viewed as the Cartesian product of domains $U^{(i)}$, attached to attributes $P^{(i)}$ that are chosen to characterize s_0. We suppose that s_0 is a n-tuple $(s^{(1),0},...,s^{(n),0})$ of attribute values $s^{(i),0} \in U^{(i)}$, i=1,...,n. The definition and application of fuzzy expert systems consists of four phases, which can be distinguished conceptually as follows: *i*) In the first phase the knowledge acquisition which leads to appointing the attributes $P^{(1)},...,P^{(n)}$, $n \in N$ and their domains $U^{(1)},...,U^{(n)}$. Fixing the universe $U = \Pi(U^{(i)})_{i \in Nn}$, $N_n \subset N$ provides the representation structure for the expert knowledge and forms the set of all states that are a priori possible; *ii*) In the second phase rules are formulated that express general dependencies between the domains of the involved attributes $P^{(1)},...,P^{(n)}$. The single rule R_j, j=1,...,m, $m \in N$, do not concern all attributes normally, but only a small number $P^{(i)}$, $i \in M_j$, which are identified by an index set $M_j \subseteq N_n$ of low cardinality.

The matching window is either a point, or a rectangle, depending on whether the matched fuzzy proposition holds at a time point or in a time interval. First, we should determine the time domains of variables in the database, or in other words, determine the size of the matching window and its position, by giving priority to the temporal matching. In the case that the event described by a fuzzy fact has appeared or is appearing, we can continue to perform the numeric matching. The application of the fuzzy formulation is advantageous in cases when small violations of specific constraints may be tolerable for the decision-maker with the goal to achieve a more reasonable objective.

Therefore, there exist some unique problems in the fuzzy reasoning procedure: the successful pattern-matching of a fuzzy rule not only requires that all the fuzzy propositions in the rule's premise should match the data in the database in a fuzzy sense, but also requires that the temporal relations among these fuzzy propositions should match the temporal relations implicitly formed by the corresponding dynamic situations in the database in a fuzzy sense.

ISSN: 1790-5109

ISBN: 978-960-474-099-4

A model associated with a possibilistic expert system and which is also based on a temporal reasoning should meet the following requirements, as outlined in the following algorithm:

Context
- *A fuzzy compiled rule base*
- *Fuzzy database with fuzzy temporal relations*

1. *Find* a time range associated with the time variable $X^{(i)}$, i= 1,...,n from the database according to the fuzzy descriptor DT, where $\Delta T = \left(\int_T \frac{\mu_1(t)}{t}, \int_T \frac{\mu_2(t)}{t} \right)$, the sentence P_i associated with variable $X^{(i)}$ is assumed to be within on interval DT formally described by

$$DT\left(P_i, \int_T \frac{\mu_1(t)}{t}, \int_T \frac{\mu_2(t)}{t}, m \right)$$

This way, we can find the size and the position of the matching window, priority been given to the temporal matching

2. *Perform* the temporal pattern matching in compliance with the existing temporal attributes. If (the temporal pattern-matching is successful) then compute its degree of confidence and proceeds to step 3 otherwise rejected situation

3. *Perform* the numeric pattern matching by using the pair Π and N. If (the numeric pattern-matching is successful) then continue the fuzzy reasoning algorithm based on compiled fuzzy rule base otherwise rejected fact. The numeric pattern-matching calls for the synthesis of $X^{(i)}$ based on associated values $x^{(i)}(t)$, $t \in DT$ into a single value

4. *Complete* the global pattern matching with both new facts derived from the process and already with the inferred facts. More specifically finish the fuzzy reasoning process starting from a given fuzzy state up to its (finite) limit passing through a sequence of internal states of the possibilistic expert system

5. *Defuzzify* outputs to obtain the results for all output variables

The possibilistic expert system has to be designed so that it can eliminate the undesirable system behaviours. There is a need to specify the initial state of the closed-loop system to reduce the combinations that may complicate the model. In analysis, the focus is on testing the closed-loop properties [5]: reach ability (firing a sequence of rules to derive a specific conclusion), cyclic behaviour of the fuzzy inference loop, stability (the ability to concentrate on the control problem).

3 The predictability in FESPC system

We start with a simple model of an expert system (the database is $BF=\{F_1,...,F_m\}$ and the rule base is $R=\{R_1,...,R_n\}$). The rule R_i has the form $C_1,...,C_k \rightarrow A_1,...,A_p$. The conditions of rule R_i are under the set of causes $COND(R_i)=\{C_1,...,C_k\}$.

Let $VAR(C_j)$ (j=1,...,k) be the set of variables that occur in condition C_j and $VAR(COND(R_i))$ the variables present in $COND(R_i)$. The pattern-matching algorithm entails two steps: the conditions/fact pattern matching and the variables linking.

A condition C filters a fact F if it can be determined a substitution σ so that $F=\sigma \cdot C$. The substitution σ can be represented through a list of pairs under the form $\sigma=\{t_1/v_1,..,t_s/v_s\}$, where the pair t_i/v_i means that the variable v_i in condition C will be replaced by the term t_i. Applying the substitution σ to condition C we obtain its instantiation C', resulting the relation $C'=\sigma \cdot C$. When a fact filters a condition, it is an instantiation of the condition. The condition C may filter several facts in database, which may be reunited in the instantiation, set of the condition C, noted I(C). This set satisfy the following relations: $I(C) \subset BF$, $(\forall)F_i$, $F_i \in I(C)$, where F_i is an instantiation of the condition C, and there is a corresponding substitution σ_i, so that $F_i = \sigma_i \cdot C$. It follows that I(C) can be represented by the list $I(C)=\{(\sigma_1,F_1),(\sigma_2,F_2), ...,(\sigma_q,F_q)\}$ and $F_i=\sigma_i \cdot C$.

Repeat the elementary pattern-matching for all the rules until obtain the instantiation sets of all the conditions. The algorithm based on the repeated condition/fact pattern matching is inefficient because of the numerous redundancies. The purpose of the second step is to find the antecedent instantiations for all the rules. This step occurs on the level of the global conditional part evaluation of the rules and a delicate operation is the linking of the variables (it permits the substitutions compatibility verification) shown as follows: for a rule R_i with $COND(R_i)$, it is required to find a set $\{(\sigma_1,F_1),..., (\sigma_k,F_k)\}$ so that $(\sigma_i,F_i) \in I(C_j)$ and $F_j= \sigma_j \cdot C_j$, j=1,...,k. If the terms associated to the common variables are identical, then the substitutions $\sigma_1,...,\sigma_n$ are consistents. The consistent substitutions composition are noted $\sigma=\sigma_1 \cdot \sigma_2 \cdot ... \cdot \sigma_k$ which contains all the distinctive variables of the substitutions. The substitutions consistence verification consists on a symbolic comparison. If there is in database fuzzy facts, the consistence verification of the substitutions is much more difficult, like in classical one.

The fuzzy condition/fact pattern matching. The fuzzy pattern-matching aims to determine the instantiations set of the causes. It is stronger than classic one because of its capacity of processing the fuzzy knowledge. It is a matter of evaluating the degree of this pattern matching between a fuzzy cause and a fuzzy fact (the fact filters more or less the cause). In order to put a fact in touch with a cause we can build up a recursive algorithm, comparing the two associated trees step by step. It follows beyond doubt that the knowledge pattern matching is the basic operation. Generally speaking, it is a matter of pattern-matching between a model P and a data D to which we attach μ_P respectively π_D ($\mu_P(u)$ represents the degree of the compatibility between the value u and the meaning of P, while $\pi_D(u)$ represents the possibility degree that the value u represents the value of the attribute which describes an object modelled through the data D). The degree of compatibility has the membership function $\mu_{P|D}$ defined through the extension principle. Though it translates relevant information related to the degree of the pattern matching between P and D, it is difficult to use $\mu_{P/D}$. We prefer two scalar measures in order to evaluate the compatibility: Π(P,D) and N(P,D).

The fuzzy constants compatibility. Let us consider the most simple case ((*f, *m→*c),*c'), where *m is the cause of the rule *m→*c, *f is the fact, each of them being expressed by fuzzy sets. In order to deduce the conclusion *c', it is to be known if the fact is compatible with the rule condition. We can try to calculate generalized modus ponens (GMP) for the inference conclusion *c', else the calculating process stops. The theory of possibilities provides two measures, which are very useful to evaluate the compatibility of the fuzzy sets [ ]:

$$\Pi(*m,*f) = \sup_u \min(\mu_{*m}(u), \mu_{*f}(u))$$
$$N(*m,*f) = 1-\Pi(\neg *m, *f) = \inf_u \max(1-\mu_{*m}(u), \mu_{*f}(u))$$

Generally, it is much complicated to calculate N than Π. A simple calculating method is based on the separation of the complementary of *m. Analysing the form of ¬*m we find that this can be divided into two fuzzy sets L_s and L_d.

The fuzzy set $L_s=(-\infty,g_m-\varphi_m,-\infty,\varphi_m)$ is always on the left of *m while $L_d=(d_n+\delta_m,\infty,\delta_m,\infty)$ is always on the right of *m, and $L_s \cap L_d=\varnothing$. It follows that $\neg *m = \max(L_s,L_d)$. We obtain:

$$N(*m,*f) = 1-\Pi(\neg *m,*f) = 1 - \Pi(\max(L_s,L_d),*f) = 1-\max(\Pi(L_s,*f),\Pi(L_d,*f))$$

Having Π and N, defined and calculated this way, we distinguish several classes of decreasing compatibility. Even if the measure Π and N correctly estimates the degree of compatibility between the fuzzy constants, these measures can not be used directly to infer the conclusions in the case of an inference engine based on GMP. If the measures Π and N satisfy some thresholds, then the pattern matching is successful. To calculate GMP we need the parameters θ and K, in the following form [ ]:

$$\theta = (*m,*f) = \max(\mu_{*f}(g_m-\gamma_m),\mu_{*f}(d_m-\varphi_m))$$
$$K = (*m,*f) = \min(\mu_{*m}(g_f),\mu_{*m}(d_f))$$

At the end of the fuzzy condition/fact pattern-matching stage for the cause C and the fact F, if the degrees of the pattern matching satisfy the chosen thresholds and if there is a consistent substitution σ, then pattern matching is successful. The substitution σ is a particular case when the variables in the causes can be associated to some fuzzy constants present in the facts. The instance σ·C obtained through the application of the fuzzy substitution σ to the condition C is not totally equal with F, i.e. the expression F=σ·C is not always true then σ is fuzzy. We can take into account the problem of finding the proper thresholds of the measures Π and N in order to determine the facts that do not filter the causes at all. The choice is not made at random, as between the two parameters of GMP it must be a tight link. Because of all these remarks and in order to correctly solve the problem, there are the links between Π, N, θ, K. As already shown GMP verifies the following important proposition:

Proposition. i) $K = 0 \Leftrightarrow \theta = 1$; $K > 0 \Leftrightarrow \theta < 1$; ii) The conclusion *c' inferred through GMP is uncertain: $(\mu_{*c'}=1) \Leftrightarrow \theta = 1$; iii) $N(*m, *f) > 0 \Leftrightarrow \theta < 1$.

The linking of the fuzzy variables. The fuzzy condition/fact pattern matching constitutes the first stage in the running of the inference engine, which takes into account the imprecision. After this stage, it results a lot of instantiations of the causes. Each instantiation of reason will be associated to a fuzzy substitution and to the four parameters Π, N, θ, K. The second stage is represented by the linking of the variables and it aims at determining the consistent instantiations at the full conditions level of the rules.

The fuzzy unification. The fuzzy unification aims at verifying the consistence of the fuzzy substitutions where the variables can be associated to fuzzy sets. Let's consider a rule (*D *H ?x) (B ?x)$\rightarrow$(act(C *E ?x)). In the antecedent of the rule there are two causes C_1= (*D *H ?x) and C_2=(B ?x). We suppose the facts to be specified: F_1 = (*d_1 *h_1 *w) and F_2 = (B *r). For some chosen fuzzy sets, the fuzzy constant *d_1 filters *D and *h_1 filters *H. The only result for the pattern-matching between C_2 and the fact F_2 is the fuzzy substitution σ=(* /?x) and the pattern-matching parameters. If all the parameters satisfy the designed thresholds, then the facts totally unify with the causes. After the fuzzy condition/fact pattern-matching, we obtained two fuzzy substitutions: σ={*w/?x} and σ = {*r/?x} where *w and *r are fuzzy sets.

The fuzzy unification contains on the one hand the evaluation of the consistence degree of the fuzzy substitutions on a certain norm and on the other hand, the fuzzy substitutions composition. Let us consider a rule R with k conditions, under the form COND(R)=(C_1,...,C_k). After the fuzzy condition/fact pattern-matching, if each condition C_i, filters a fact F_i, then there is a fuzzy substitution σ_i so that $F_i = \sigma_i \cdot C_i$ and the four parameters Π_i, N_i, θ_i, K_i. Let us consider a variable ?v within the rule; we suppose to appear n times in the conditional part of the rule. ?v_i is used for the representation of i^{th} of the variable ?v. In this case, all the occurrences of the variable ?v within the global condition of the rule can be represented through the following list: {?v_1, ?v_2,...,?v_n}. Each ?v_i will be certainly associated with a term t_i, which can be an atomic or a fuzzy constant, denoted: {t_1/?v_1, t_2/?v_2,..., t_n/?v_n}.

All the various variables present in a rule are independent. Each variable can occur in a rule several times. Each occurrence of the variable is independent of the other occurrences. Nearly all expert systems preserve this hypothesis. The fuzzy unification consists of: *i)* The consistence verification of the element in list {t_1/?v_1, t_2/?v_2,...t_n/?v_n}$\rightarrow${t_p/?v_p} as against a certain norm; *ii)* The composition of the fuzzy substitutions. In order to eliminate any confusion, ?v_p is used to represent the variable ?v after the fuzzy unification. Finally, the fuzzy unification can be represented through the following expression: {t_1/?v_1, t_2/?v_2,...t_n/?v_n} {t_p/?v_p} where t_p is going to be calculated [4]. Let us consider a simple case. If t_i is a fuzzy set, i.e. t_i = *t(i), (i=1,2), then the symbolic or numerical comparison is no longer sufficient to evaluate the consistence between *t(1) and *t(2). When ?v_1 and ?v_2 are independent, the Cartesian product *t(1) x *t(2) is defined by *t(1)x*t(2) = $\{((x_1, x_2), \mu_{*t(1)x*t(2)}(x_1,x_2)/x_1 \in X_1, x_2 \in X_2, X_1, X_2 \subset R\}$, $\mu_{*t(1)x*t(2)}(x_1,x_2) = \min(\mu_{*t(1)}(x_1)_{*t(2)}(x_2))$.

The compatibility between *t(1) and *t(2) can only by clarified through a reasonable explanation of the criterion relative to which compatibility is judged. In the classic situation, the criterion is made up by the equality relation. It is quite natural to introduce appropriate criteria for fuzzy unification in both stages: to check the consistence and to make up the fuzzy substitutions. These criteria should be more general; the equality relation can be defined by a binary fuzzy relation R. Making up the fuzzy set *t(1) and the relation R, we obtain $\mu_{R\circ *t(1)}(x_2)$, defined by:

$$\mu_{R\circ *t(1)}(x_2) = \sup_u \min(\mu_R(x_1,x_2),\mu_{*t(1)}(x_1))$$

Since we know both relation R and Cartesian product (t(1) x *t(2), we can use measures Π and N to estimate the consistence of fuzzy sets *t(1) and *t(2) relative to R. Thus, we have:

$$\Pi(R, *t(1) \text{ x } *t(2)) = \sup_{x1,x2}\min(\mu_R(x_1, x_2), \mu_{*t(1)}(x_1), \mu_{*t(2)}(x_2))$$

$$N(R, *t(1) \text{ x } *t(2)) = \inf_{x1,x2}\max(\mu_R(x_1, x_2), 1- \mu_{*t(1)}(x_1), 1 - \mu_{*t(2)}(x_2))$$

It is interesting to note that the fuzzy binary relation R, can be interpreted in various ways. The equality relation may be regarded as a particular case of relation R. A last important problem is the parameters propagation. An example is the surge tank with two fill valves A and B, and an empty valve C, located at the bottom of the tank. When valve A is opened, it automatically closes itself after pouring enough liquid so that the liquid level in tank increases by one level (small), and for valve B the liquid level in the tank increases by three levels (medium). We require that only one fill valve can be opened at once. The opening of valve C is random and unpredictable. Once this valve opens, the liquid level in the tank decreases by two level units.

The control objective for this problem is to control the liquid level in the tank so that it lies in the normal range for any given initial liquid level. We search one strategy that human expert can use to meet the control objectives, based on the seven rules of plant model [5]. We present in the figure 1 the simulation results for the priority scheme used in the select phase.

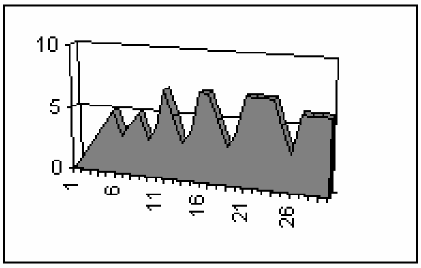

Fig. 1 Simulation results

The set of knowledge base states is X^b, where $x^b \in X^b$, $x^b=[x^{b,1},...,x^{b,6}]^t \in X^b$. The $x^{b,1}$ component represents the current liquid level in the tank; $x^{b,2}$ =0,1,2 indicating that e_{NUL},e_A,e_B event is enabled respectively; $x^{b,3}$=0 indicating that the liquid level has never reached 5 and $x^{b,3}$=1 indicating that the level has reached 5; $x^{b,4}$=0 indicating that rules 1 and 2 are allowed to fire; $x^{b,5}$=0 indicating that rules 3 and 4 are allowed to fire and $x^{b,5}$=1 indicating that rules 3 and 4 are not allowed to fire; $x^{b,6}$=0 indicating that rule 5 is allowed to fire and $x^{b,6}$=1 indicating that rule 5 is not allowed to fire. The set of expert system command inputs is $E^{PES} = E_0 = X$. It is obvious that cycles exist in the open-loop plant. To eliminate these undesirable properties and meet the closed-loop specifications, the expert system as above is employed.

4 Conclusions

The use of temporal aspects refers to the design of those tools to solve the equation **time** = complexity $\oplus$ real time $\oplus$ temporal reasoning, which is employed in order to integrate time into a process control application. This equation is formally found on the inference engine algorithm, able to make full use of the specific knowledge to the process control. The symbolic aggregation metaoperator $\oplus$ can be instantiated into different classes of specific operators, depending on the goal pursued by the control model. We assume that the process operates like finite no deterministic state machine, while the expert system will operate like a finite deterministic state machine. The closed-loop control expert system can be modelled like a no deterministic state machine, whose outputs are the process outputs. A major obstacle to the widespread use of (possibilistic) expert systems in real-time domains is the non-predictability of rule execution time. A widely used algorithm for real-time production systems is the Rete algorithm. To achieve a fast reasoning the number of fuzzy set operations must be reduced. For this, we use a fuzzy compiled structure of knowledge in FESPC, like Rete, because it is required for real-time responses and a fuzzy inference engine. The FESPC engine represents a method of fast fuzzy logic inference. It must provide guaranteed response times, completing its reasoning within a deterministic amount of time. Systematic analysis methods must be used so that the possibilistic expert system behaviour can be studied quantitatively within the developed modelling framework.

References:

[1] Bellman R.E. and Zadeh L.A. - 1970, D*ecision-making in a fuzzy environment*, Management Sci. 17(1970) B141-B164

[2] Dubois D., Prade H. - 1991, *Fuzzy Sets in Approximate Reasoning*, Part 1: Inference with Possibility Distributions, Fuzzy Sets and Systems, 40: 143-202, Part 2: Logical Approaches, p. 203-244

[3] Ghallab M. - 1988, *Moteur d'inférence pour les règles de production*, Actes Colloque International de l'Intelligence Artificielle, Marseille

[4] Klawonn F., Noväk V. - 1996, *The relation between inference and interpolation in the framework of fuzzy systems*, Fuzzy Sets and Systems, 81, p. 331-354

[5] Passino K.M., Antsaklis P.J. - 1989, *A System and Control Theoretic Perspective on Artificial Intelligent Planning Systems*, IJAAI, p. 31-32

[6] Zadeh L. - 1987, *The concept of a Linguistic Variable and its Applications to Approximate Reasoning*, Information Sciences, 8, p. 199-249