A Processing Algorithm for an Intelligent Production System

Associated Professor, Ph.D. Vasile MAZILESCU
Accounting and Economic Informatics Department
Faculty of Economic and Administrative Sciences University „Dunarea de Jos" of Galati

*The aim of this paper is to present the aspects of a control fuzzy knowledge based system **(CFKBS)** for a flexible manufacturing system that must be predictable. Typical examples include control software for monitoring, safety-critical and risky economic and industrial applications. We give an algorithm for processing a knowledge control model.*
Keywords: *predictability, real-time expert systems, inference engine.*

1 Introduction

In this paper we concentrate on the applicability of a processing algorithm for a specific knowledge model, with respect to the timing requirements. The use of temporal aspects refers to the design of those tools to integrate time in process control applications. These aspects are formally found on the inference engine algorithms, able to make full use of the specific knowledge to the process control [1, 5]. The symbolic aggregation metaoperator 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 nondeterministic state machine, while the expert system will operate like a finite deterministic state machine. The closed-loop control expert system can be modeled like a nondeterministic 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 [2]. 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 and a fuzzy inference engine. The engine represents a method of fast fuzzy logic inference [3, 4]. The fuzzy expert system **CFKBS** predictability has been specified in section 2. To illustrate the theoretical results we provide in section 3 an example of a fuzzy model based on metalevel knowledge for flexible production system in a specific structural definition [6]. Section 4 present concluding remarks to develop **AI** reasoning systems that utilize learning and planning capabilities.

2. The discrete models of CFKBS

In **AI**, the problem domain must be defined as a collection of problems that the expert system desires to solve. In conventional control, the plant is a dynamical system, described with linear or non-linear differential/ difference equations. An **AI** expert system consists of the planner or the inference engine, the problem domain, the exogenous inputs, and their interconnections. The outputs of the expert system are the inputs (control actions) to the problem domain. There are unmeasured exogenous inputs to the problem domain (disturbances) that represent specific uncertainty. The measured exogenous input to the expert system is the goal. An expert system can be modeled using predicate or temporal logic or other symbolic techniques such as finite state machine. **AI** feedback expert systems are analogous to conventional feedback control systems that do not use state estimation (they do not use situation assessment).

Our **CFKBS** fuzzy real-time expert systems must represent imprecise, time and temporal data, encode temporal knowledge and manage temporal fuzzy reasoning [6]. Following a conventional planning-theoretic approach, we can introduce a mathematical model for the plant P and the possibilistic expert control system (**PECS**), which consists of the possibilistic expert system (PES) and the plant. The **PES** must be designed so that it can coordinate the use of the plant outputs and reference (user) inputs, to decide what plant command inputs (or hypothesis/ conclusions) to generate so that the closed-loop specifications are met. Although the **PES** (viewed as an expert system) are frequently being used to perform complex control functions, most often it is the case that no formal analysis of the dynamics is conducted because mathematical analysis of such systems is often considered to be beyond the scope of conventional control theory.

It is assumed that the economic process can be represented with the following model: **P**=(X, E, f_e, δ_e, g, E_v), that can represent certain class of discrete event systems, where X is the set of plant states denoted by x, E is the set of all events, f_e are the state transition map, f_e: X→X, $e_k \in$ P(E), $k \in$ T, δ_e are the output maps, g is the enable function, g:X→ P(E), and E_v is the set of all valid event trajectories (that are physically possible). Note that E is the union of the command-input events (E_u), the disturbance input events (E_d) and the output events (E_o) of the plant. When discussing the states and events at time k, $k \in$ T or k is a fuzzy instant or a fuzzy time interval, $x_k \in X$ is the plant state, $e_{uk} \in E_u$ is a command input event of the plant, $e_{dk} \in E_d$ is a disturbance input event of the plant, $e_{ok} \in E_o$ is an output event of the plant, that is equal to input event $e_{pk} \in E_p$ for **PES**. Each $e_k \subset g(x_k)$ is an event that is enabled at time k, and it represents a set of command and disturbance input events of the plant. If an event $e_k \in E$ occurs at time k and the current state of plant is x_k, then the next state is $x_{k+1} = f_{ek}(x_k)$ and the output is $e_{ok} = e_{pk} = \delta_{ek}(x_k)$. Any sequence $\{x_k\}$ such that for all k, $x_{k+1} = f_{ek}(x_k)$, where $e_k \subset g(x_k)$ is called a *state trajectory*. The **PES** has two inputs: the reference input events $e_{rk} \in E_r^{PES}$ (user inputs) and the output events of the plant $e_{ok}=e_{pk}$, $e_{rk} \in E_p^{PES}$. Based on its fuzzy state and these inputs, the **PES** generates enable command input events to the plant $e_{0_k}^{PES} \in E_0^{PES}$.

A fact in a fuzzy database may be a property ω of objects; an object x_0 has, or has not the property ω, or in other words $\omega(x_0)$ is true to degree α, $\alpha \in [0,1]$.

$A(x) \in F_J$ itself is not sufficient to characterize the vagueness of ω. A couple (A,α), where A $\in F_J$ and $\alpha \in [0,1]$ is a truth degree of A is called an uncertain clause. The vagueness of ω is formally characterized by a set of uncertain clauses A so that $A=\{(A_x(t); \alpha_t) \mid t \in J$, t= term of the language J$\}$; $A_x(t)$ is a formula obtained from A by replacing all free occurrences of x by the term t. An inference rule is a scheme $([A_1; \alpha_1],...,[A_n;\alpha_n])/[B;b]$, where $B=r^{syn}(A_1,...,A_n)$ is a formula syntactically derived from A_i, i=1,n and $b= r^{sem}(\alpha_1,...,\alpha_n)$ is its resulting evaluation. In approximate reasoning used in process management and control, the basic situations are defined by specifying a linguistic description or model based on fuzzy rules $R_1,...,R_k$. Each of R_i is interpreted by considered as a set of special axioms, being the basis of approximate reasoning at the given moment based on some inference rules [3].

3. A case study

A Flexible Production System (**FPS**) can be represented by a G= (M,A) graph, where M= {1,…,N} represents the set of the identical subsystems (agents) from its structure and A⊂MxM. We consider that G is strongly connected, meaning that for (∀) $i \in M$ there is a path from i to (∀) $j \in M$ and additionally, if $(i,j) \in A$, $i \neq j$. Each subsystem has a quantity of tasks that can be processed by the i agent or by any other agent. We consider the afferent quantity of i, marked by $x_i \geq 0$ for $x_i \in N^*$ or $\mu_{xi} \in R^*_+$ (for the fuzzy case), the quantity that can be partially transferred to the j subsystem. The afferent **CFKBS** to this problem domain must transfer the quantities from i to j if there is $(i,j) \in A$. We mark with $e^c_{k,}{}^{ij}$ the command fact by which the quantity ck from i to j is transferred. We consider a **FPS** structure with N=6 connected such: {(1,2), (2,1), (1,3), (3,4), (4,3), (4,2), (3,5), (5,6), (6,5), (6,4)} ⊂ {1,...,6}x {1,...,6}. The problem that **CFKBS** must to solve is for a given **FPS** structure with initial given values $x_i \geq 0$ to establish an equilibrated distribution of the each agent load, $i \in M$. The model that is to be compiled $\mathbf{M_k}$ of the **FPS** is inserted within the knowledge base of the **CFKBS** and it implies the graph structure (M, A), $\Sigma_{i=1,6}\, x_i/6 = c$, c? $\mathbf{N}^*_+$ or c? $\mathbf{R}^*_+$.

For the synthesis of the fuzzy knowledge model afferent to this **FPS** it was necessary to consider the loads as T-fuzzy numbers, linguistically under the form of "about the value" or "approximate" as well as the introduction of some intermediary variables into the model structure of the type: the degree of global equilibration ge (at the whole FPS level) with *satisfactory* fuzzy values and *non-satisfactory* ones for the partial equilibration degree gep_i, i = 1,…5 (for subsystem groups) corresponding to the certain unsolved situations within the crisp model as well the fuzzy variables d_{56}, d_{42}, d_{13}, d_{21}, d_{43}, d_{35}, which can get T- fuzzy numbers *small*, *big*, and *zero*. The actual partial equilibration degrees applied, are characterized by uncertain linguistic values such *good* and *non-satisfactory* (the non-satisfactory values is similar as a fuzzy number to the linguistic value non-satisfactory for the global equilibration degree). It was needed to introduce knowledge under the form of some uncertain facts of the type (X1 X2 X3 X4 ?v_q), (X1 X2 X4 ?v_q), (X3 X4 X5 X6 ?v_q), (X1 X3 X4 X5 X6 ?v_q), (X2 X4 X5 X6 ?v_q) certifying the fact that the equilibration can take place continuously, partially, or gradually as well as a large number of meta-rules having all these facts. They are to be sequentially activated (under the form (R13, R14), (R13, R15), (R16, R17), (R16, R24), (R18, R19), (R18, R25), (R20, R21), (R22, R23)) and they support the fuzzy decision synthesis. Obviously this guiding model observes the stages for the linguistic model synthesis from the knowledge acquisition point of view and the knowledge and meta-knowledge representation as per the **CFKBS**. The exploration of this model implies the calculation of the fuzzy unification and of the partial shaped conclusion through the diagram of the generalized modus ponens, the apply for the procedures within the rules consequent and a certain control model, the dynamic update of the rules priorities as per the current uncertainty of all the involved knowledge at a certain moment in the synthesis process of the decision, the sequence of the fuzzy meta – rule as well as the demonstration of the global asymptotic stable behavior of the closed- loop system. The ≥∗ and =∗ of the $\mathbf{M}_k$ compiled model structure allow the uncertain comparison of the current loads values X1,….,X6.

The $\mathbf{M}_F$ fuzzy model for this case is:

1. *If* ((≥*(X1 ?x) (X2 ?y)) ∧ (≥*(X1 ?x) (X3 ?z))∧(≥*(X1 ?x) (X4 ?v)) ∧ (≥*(X1 ?x) (X5 ?w)) ∧ (≥*(X1 ?x) (X6 ?ξ)) ∧ (¬(Fa1 ?v_1)) ∧ (¬(Fa3 ?v_3)) ∧ (¬(=*(X1 ?x)(X2 ?y))) ∧ (ge *n)) *else* ((x1b 1) ∧(x3b 0) ∧ (x5b 0) ∧(x7b 0) ∧ (x9b 0) ∧(x10b 0) ∧ (x11b 0) ∧ (x2b 1))

2. *If* ((≥*(X1 ?x) (X2 ?y)) ∧ (≥*(X1 ?x) (X3 ?z)) ∧ (≥*(X1 ?x) (X4 ?v)) ∧ (≥*(X1 ?x) (X5 ?w)) ∧ (≥*(X1 ?x) (X6 ?ξ)) ∧ (¬(Fa2 ?v_2)) ∧ (¬(=* (X1 ?x) (X3 ?z))) ∧ (≥* (X1 ?x) (CFS *zero)) ∧ (ge *n)) *else* ((x1b 2) ∧ (x2b 0) ∧(x4b 0) ∧ (x6b 0) ∧ (x7b 0) ∧ (x9b 0) ∧ (x10b 0) ∧ (x11b 0) ∧ (x3b 1))

3. *If* ((≥*(X2 ?y) (X1 ?x)) ∧ (≥*(X2 ?y) (X3 ?z)) ∧ (≥* (X2 ?y) (X4 ?v)) ∧ (≥* (X2 ?y) (X5 ?w)) ∧ (≥* (X2 ?y) (X6 ?ξ)) ∧ (¬(Fa3 ?v_3)) ∧ (¬(Fa1 ?v_1)) ∧ (¬(=* (X1 ?x) (X2 ?y))) ∧ (ge *n)) *else* ((x1b 3) ∧ (x3b 0) ∧ (x5b 0) ∧ (x7b 0) ∧ (x9b 0) ∧ (x10b 0) ∧ (x11b 0) ∧ (x4b 1))

4. *If* ((≥*(X3 ?z)(X1 ?x)) ∧ (≥* (X3 ?z) (X2 ?y)) ∧ (≥* (X3 ?z) (X4 ?v)) ∧ (≥* (X3 ?z) (X5 ?w)) ∧ (≥*(X3 ?z) (X6 ?ξ)) ∧ (¬ (Fa4 ?v_4)) ∧ (¬(=*(X3 ?z) (X5 ?w))) ∧ (≥*(X1 ?x) (CFS *zero)) ∧ (ge *n)) *else* ((x1b 4) ∧ (x2b 0) ∧ (x4b 0) ∧(x6b 0) ∧ (x7b 0) ∧ (x9b 0) ∧ (x10b 0) ∧ (x11b 0) ∧ (x5b 1))

5. *If*((≥*(X3 ?z)(X1 ?x)) ∧ (≥* (X3 ?z) (X2 ?y)) ∧ (≥* (X3 ?z) (X4 ?v)) ∧ (≥* (X3 ?z) (X5 ?w)) ∧ (≥* (X3 ?z) (X6 ?ξ)) ∧ (¬(Fa5 ?v_5)) ∧ (=* (X1 ?x) (CFS *zero)) ∧ (ge *n)) *else* ((x1b 5)∧(x2b 0)∧(x3b 0) ∧ (x4b 0) ∧ (x5b 0) ∧ (x7b 0) ∧ (x9b 0) ∧ (x10b 0) ∧ (x11b 0) ∧ (x6b 1))

12. *If* (ge *n) ∧ (x12b 1) *else* ((x1b 0) ∧ (x2b 0) ∧ (x3b 0) ∧ (x4b 0) ∧ (x5b 0) ∧ (x6b 0) ∧ (x7b 0) ∧ (x8b 0) (x9b 0) ∧ (x10b 0)) ∧ Init(Modul de control)

13. *If* (X1 ?x)∧(X2 ?y)∧(X3 ?z)∧(X4 ?v) ∧ (ge *n) ∧ (gep_1 *b) *else* (X1 X2 X3 X4 ?v_q) ∧ (gep_1 *b_1)

14. *If* (X1 X2 X3 X4 ?v_q) ∧ (ge *n) ∧ (gep_1 *b) ∧(d_{56} *ma) *else* (gep_1 *b_1) ∧ Init(r8)

15. *If* (X1 X2 X3 X4 ?v_q) $\wedge$ (ge *n) $\wedge$ (gep$_1$ *b) $\wedge$(d_{35} *ma) *else* (gep$_1$ *b_1) $\wedge$ Init(r4)
16. *If* (X1 ?x) $\wedge$(X2 ?y) $\wedge$ (X4 ?v) $\wedge$ (ge *n) $\wedge$ (gep$_2$ *b) *else* (X1 X2 X4 ?v_q) $\wedge$ (gep$_2$ *b_1)
17. *If* (X1 X2 X4 ?v_q) $\wedge$ (ge *n) $\wedge$ (gep$_2$ *b) $\wedge$(d_{13} *ma) *else* (gep$_2$ *b_1) $\wedge$ Init(r2)
18. *If* (X3 ?z)$\wedge$(X4 ?v)$\wedge$(X5 ?w)$\wedge$(X6 ?ξ)$\wedge$(ge *n)$\wedge$(gep$_3$ *b) *else* (X3 X5 X6 ?v_q)$\wedge$(gep$_3$ *b_1)
19. *If* (X3 X4 X5 X6 ?v_q) $\wedge$ (ge *n) $\wedge$ (gep$_3$ *b) $\wedge$(d_{21} *ma) *else* (gep$_3$ *b_1) $\wedge$ Init(r3)
20. *If* (X1 ?x) $\wedge$(X3 ?z)$\wedge$(X4 ?v)$\wedge$(X5 ?w)$\wedge$(X6 ?ξ)$\wedge$(ge *n) $\wedge$ (gep$_4$ *b) *else* (X1 X3 X4 X5 X6 ?v_q) $\wedge$ (gep$_4$ *b_1)
21. *If* (X1 X3 X4 X5 X6 ?v_q) $\wedge$ (ge *n) $\wedge$ (gep$_4$ *b) $\wedge$(d_{42} *ma) *else* (gep$_4$ *b_1) $\wedge$ Init(r6)
22. *If* (X2 ?y)$\wedge$(X4 ?v)$\wedge$(X5 ?w)$\wedge$(X6 ?ξ) $\wedge$ (ge *n) $\wedge$ (gep$_5$ *b) *else* (X2 X4 X5 X6 ?v_q) $\wedge$ (gep$_5$ *b_1)
23. *If* (X2 X4 X5 X6 ?v_q) $\wedge$ (ge *n) $\wedge$ (gep$_5$ *b) $\wedge$(d_{43} *ma) *else* (gep$_5$ *b_1) $\wedge$ Init(r7)
24. *If* (X1 X2 X4 ?v_q) $\wedge$ (ge *n) $\wedge$ (gep$_2$ *b) $\wedge$(d_{56} *ma) *else* (gep$_2$ *b_1) $\wedge$ Init(r8)
25. *If* (X3 X4 X5 X6 ?v_q) $\wedge$ (ge *n) $\wedge$ (gep$_3$ *b) $\wedge$(d_{13} *ma) *else* (gep$_3$ *b_1) $\wedge$ Init(r2)

The distinct motives occurring within the guiding model as per the **CFKBS** can be grouped into subsets of motives according to the length feature such:

i) the motives of 2 length are: M_1=(X1 ?x), M_2=(X2 ?y), M_3=(X3 ?z), M_4=(X4 ?v), M_5=(X5 ?w), M_6=(X6 ??), M_7=(Fa1 ?v_1), M_8=(Fa2 ?v_2), M_9=(Fa3 ?v_3), M_{10}=(Fa4 ?v_4), M_{11}=(Fa5 ?v_5), M_{12}=(Fa6 ?v_6), M_{13}=(Fa7 ?v_7), M_{14}=(Fa8 ?v_8), M_{15}=(Fa9 ?v_9), M_{16}=(Fa10 ?v_{10}), M_{17}=(x12b 1), M_{18}= (ge *n), M_{19}=(gep$_1$ *b), M_{20}=(gep$_2$ *b), M_{21}=(gep$_3$ *b), M_{22}=(gep$_4$ *b), M_{23}=(gep$_5$ *b), M_{24}=(d_{56} *ma), M_{25}=(d_{35} *ma), M_{26}=(d_{13} *ma), M_{27}=(d_{21} *ma), M_{28}=(d_{42} *ma), M_{29}= (d_{43} *ma);

ii) the motives of 3 length are: M_{30}=(≥*(X1 ?x)(X2 ?y)), M_{31}=(≥*(X1 ?x)(X3 ?z)), M_{32}=(≥* (X1 ?x)(X4 ?v)), M_{33}=(≥*(X1 ?x)(X5 ?w)), M_{34}= (≥* (X1 ?x)(X6 **?**)), M_{35}=(=*(X1 ?x)(X2 ?y)), M_{36}=(≥*(X1 ?x)(CFS *zero)), M_{37}=(≥*(X2 ?y)(X1 ?x)), M_{38}=(≥*(X2 ?y)(X3 ?z)), M_{39}=(≥*(X2 ?y)(X4 ?v)), M_{40}=(≥*(X2 ?y)(X5 ?w)), M_{41}=(≥*(X2 ?y)(X6 ??)), M_{42}=(=*(X1 ?x) (X3 ?z)), M_{43}=(≥*(X3 ?z)(X1 ?x)), M_{44}=(≥*(X3 ?z)(X2 ?y)), M_{45}=(≥*(X3 ?z)(X4 ?v)), M_{46}=(≥* (X3 ?z)(X5 ?w)), M_{47}=(≥*(X3 ?z)(X6 ??)), M_{48}= (=* (X3 ?z)(X5 ?w)), M_{49}=(=*(X1 ?x) (CFS *zero)), M_{50}= (≥*(X4 ?v)(X1 ?x)), M_{51}=(≥*(X4 ?v) (X2 ?y)), M_{52}= (≥* (X4 ?v) (X3 ?z)), M_{53}= (≥*(X4 ?v)(X5 ?w)), M_{54}=(≥*(X4 ?v)(X6 ?ξ)), M_{55}= (=*(X4 ?v)(X2 ?y)), M_{56}=(≥*(X5 ?w)(X1 ?x)), M_{57}=(≥*(X5 ?w) (X2 ?y)), M_{58}=(≥*(X5 ?w)(X3 ?z)), M_{59}=(≥*(X5 ?w)(X4 ?v)), M_{60}=(≥* (X5 ?w)(X6 ?ξ)), M_{61}= (=* (X5 ?w)(X6 ?ξ)), M_{62}= (≥* (X6 ?ξ)(X1 ?x)), M_{63}= (≥*(X6 ?ξ)(X2 ?y)), M_{64}=(≥*(X6 ?ξ)(X3 ?z)), M_{65}=(≥*(X6 ?ξ)(X4 ?v)), M_{66}=(≥*(X6 ?ξ)(X5 ?w)), M_{67}=(=* (X6 ?ξ)(X4 ?v)), M_{68}=(=*(X1 ?x)(X4 ?v)), M_{69}= (=* (X1 ?x)(X5 ?w)), M_{70}=(=*(X1 ?x)(X6?ξ));

iii) the motive of 4 length is: M_{71}=(X1 X2 X4 ?v_q);

iv) the motives of 5 length are: M_{72}=(X1 X2 X3 X4 ?v_q), M_{73}=(X3 X4 X5 X6 ?v_q), M_{74}=(X2 X4 X5 X6 ?v_q);

v) the motives of 6 length are: M_{75}=(X1 X 3 X4 X5 X6 ?v_q).

The processing algorithm

1) The fuzzy initial loads μ_{xi0} are introduced;

2) The expected average value μ_{c0}= $\Sigma_{i=1,6}$ $\mu_{x_{i_0}}$/6, the fuzzy distances $\mu_{d_{i_0}}$=$\mu_{x_{i_0}}$-μ_{c0}, as well as the initial equilibration degree ge_0= **max**$_{i=1,6}$ $\{\mu_{d_{i_0}}\}$ are to be calculated;

3) The fuzzy sets *s (by testing it is controlled in accordance with the initial values of the characterized loads of the s* fuzzy multitude), *n=t_p(3 ge_0 2 2) (by testing it is controlled d and δ in accordance with the loads initial values) as well as the facts (ge*n) and ge * s) are generated;

4) Initialize $\mathbf{x}^b$=($\mathbf{x}^{b1}$,$\mathbf{x}^{b2}$) where $\mathbf{x}^{b1}$=**0** and dim($\mathbf{x}^{b1}$)=12, $\mathbf{x}^{b2}$ contains the added initial facts to all the linguistic variables that interfere into the allocation model of the type: ((X1 ?x), (X2 ?y), (X3 ?z), (X4 ?v), (X5 ?w), (X6 ?ξ), (X1 X2 X3 X4 ?v_q), (X1 X2 X4 ?v_q), (X3 X4 X5 X6 ?v_q), (X1 X3 X4 X5 X6 ?v_q), (X2 X4 X5 X6 ?v_q), (X1 X2 X4 ?v_q), (X12b 1), (ge *n), (CFS *zero), (gep$_1$ *b), (d_{56} *ma), (d_{35} *ma), (gep$_2$ *b), (d_{13} *ma), (gep$_3$ *b), (d_{21} *ma), (gep$_4$ *b), (d_{42} *ma), (gep$_5$ *b), (d_{43} *ma), with the effective evaluation of the all fuzzy variables occurring into the structure of the **CFKBS** status component. It is launched as an initial fact added to the motive (ge* n) fact (ge * v_0) where * v_0 is generated as a fuzzy set about the value ge_0 of the form (**constfaz** *v_0(tp ge_0-1 ge_0+1 2 2)) and x ^{int} = 0 is also initiated.

5) Within the consequents of the rules R1-R10, the calculation of the ge equilibration degree with its new fuzzy value is also added, meaning that the new fact (ge * v_k), k ≥ 1 is generated

6) If the inference engine stops on the other facts rather that the activation one and the execution of the 11 rule, then the fuzzy differences $\mu_{d_{ik}}=\mu_{x_{ik}}-\mu_{c_0}$ are recalculated with the determination of the *i* system (i=1,…6) and of the corresponding *j* rule (j=1,….,25) that will be activated, as per the satisfaction of the equilibration object at a certain current moment, using the meta -rules or keeping on the execution the control module.

Finally, the organization way/methods for the tests is shown in the below table. They have been achieved with the help of the **CFKBS** system interference motor, using a compiled specific model **FPS**. The "x_0" mark has been used for a fuzzy T- number x_0.

If there are introduced the fuzzy linguistic models within an expert system, this system becomes more complicated because of the taking into consideration of the fuzzy processing at all the system levels of the type: the fuzzy filtering/pattern-matching, the compatibility of the fuzzy sets, the fuzzy unification, the calculus of the inferred conclusion together with the calculation of the parameters propagation, which manage the uncertainty, the selection strategies in which they are naturally included and imprecise elements of the factual knowledge.

Test Nr.	μ_{x1}	μ_{x2}	μ_{x3}	μ_{x4}	μ_{x5}	μ_{x6}	$\mu_{\overline{X}}$	The degree of equil.	*s,*n,*b, *mi,*ma	The characteristics of the tests 1,2,3,4,5,6
1.	"200"	"200"	"200"	"0"	"0"	"0"	"100"	A3	specific	General behavior of the fuzzy inference engine
2.	"200"	"0"	"0"	"0"	"0"	"0"	"33"	A3	specific	Non-accomplishment of the guiding object through the neutralization of meta-rules
3.	"200"	"0"	"0"	"0"	"0"	"400"	"100"	A3	specific	The accomplishment of the guiding object using only fuzzy rules
4.	"1223"	"310"	"445"	"907"	"38"	"742"	"610.8"	A3	specific	The occurrence of the dynamic circularity
5.	"77"	"88"	"205"	"382"	"166"	"0"	"153"	A3	specific	The fuzzification influence
6	"100"	"100"	"100"	"0"	"100"	"0"	"50"	A3	specific	The control module use

The organization of FPS tests

4. Conclusions

It is obvious that the open-loop plant has cyclic properties that may prevent the open-loop from achieving the desired control objective. When closed-loop fuzzy expert control is used, as in our example, the invariant set exists, by simple analysis of the system dynamics. Using a search algorithm, we show that there exists at least one path from any given initial part distribution in the **FPS**.

The reachability result (the **FPS** described above is reachable for all initial states, because there exists a sequence of events to occur that produces a state trajectory, so that the end state of the plant is in the invariant set). In our fuzzy expert system, any rule whose "partially matches" the current data can "fire" (i.e., contribute to specifying the control input). In the fuzzy expert system we consider here, there may be more than one rule whose antecedent "exactly matches" the current data, but our inference engine allows only one rule to fire at a time.

We have shown that conventional knowledge-based debugging tools can ignore important dynamic behavior that can result from connecting the full fuzzy expert system (i.e., with an inference engine) to user inputs and a dynamical process. We have illustrated the results by modeling and analyzing expert systems that solve a **FPS** as a simple process control problem.

The results of this paper shows that fuzzy expert control system are a class of (heuristically constructed) nonlinear control systems that can be studied with the analytical tools available from conventional control theory.

It is hoped that the work reported in this paper serves to promote the development of a firm mathematical foundation on which to perform careful analysis for the verification and validation of the dynamics of expert control systems that operate in critical environments. There are important another future directions for this work, investigating the dynamics of AI reasoning systems that utilize learning and planning in various complex applications, studying computational complexity issues relative to conflict resolution strategies and metaknowledge representation, and modeling realistic industrial or economic application that involve knowledge-based systems.

In the future we should develop several differencing features of learning forms in our conceptual **CFKBS**, as a multiagent system, for structuring: *the degree of decentralization* (concerns distributedness and parallelism), *interaction-specific features*, required for realizing a decentralized learning process (e. g. planning, inference or decision steps, that are executed to achieve a particular learning goal), *involvement-specific features* (the relevance of involvement and the role played during involvement), *goal-specific features*, *the learning method*, *the learning feedback*.

These features characterize learning in multiagent systems from different points of view and at different levels. Agents having a limited access to relevant information run the risk of failing in solving a given learning task. This risk may be reduced by enabling the agents to explicitly exchange information, to communicate with each other. Generally, the following two forms of improving learning by communication may be distinguished:

- learning based on low-level communication, that is, relatively simple query-and-answer interactions for the purpose of exchanging missing pieces of information (knowledge and belief);
- learning based on high-level communication, that is, more complex communicative interactions like negotiations and mutual explanation for the purpose of combining and synthesizing pieces of information.

Our **PES** must be designed so that it can coordinate the use of the plant outputs and reference (user) inputs, to decide what plant command inputs (or hypothesis/conclusions) to generate so that the closed-loop specifications are met. Although the **PES** are frequently being used to perform complex control functions, most often it is the case that no formal analysis of the dynamics is conducted because mathematical analysis of such systems is considered to be beyond the scope of conventional control theory.

We have shown that conventional knowledge-based debugging tools can ignore important dynamic behavior that can result from connecting the full fuzzy expert system (i.e., with an inference engine) to user inputs and a dynamical process. The results of this paper shows that the dynamic of a fuzzy expert control system is equivalent to improve the knowledge about a class of (heuristically constructed) nonlinear control systems that can be studied with the analytical tools available from conventional control theory.

There are important another future directions for this work, investigating the dynamics of **AI** reasoning systems that utilize learning and planning in various complex applications, studying computational complexity issues relative to conflict resolution strategies and metaknowledge representation, and modeling realistic industrial or economic applications that involve **CFKBS**s.

References

1. Albert Mo Kim Cheng, Jeng-Rung Chen, 2000 – ***Response Time Analysis of OPS5 Production Systems***, IEEE Transactions on Knowledge and Data Engineering, Vol. 12, no. 3. May/June, pp. 391-409

2. Bouaud J., 1992 - ***TREE: une stratégie de jointure heuristique pour un algorithme de filtrage***, Revue d'Intelligence Artificielle, Vol. 6, nº **4**, 1992, Hermes, Paris, p. 457 - 493

3. Dubois D., Lang J., Prade H., 1992 - ***Inconsistency in possibilistic knowledge bases: To live with it or not live with it***, Fuzzy Logic for the Management of Uncertainty, Edited by Lotfi Zadeh and Janusz Kacprzyk, Wiley Professional Computing, p. 335-351

4. Gottwald S., 1993 - ***Fuzzy Sets and Fuzzy Logic/Foundations of Application - from a Mathematical Point of View***, Vieweg, ISBN 3-528-05311-9

5. Lange D.B., 1998 – ***Mobile agents: environments, technologies, and applications***. In Proceedings of the Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, 11-14. The Practical application Company