A Fuzzy MHT Algorithm Applied to Text-Based Information Tracking

Santiago Aja-Fernández, Carlos Alberola-López, *Member, IEEE*, and George V. Cybenko

***Abstract*—In this paper, we carry out a detailed analysis of a fuzzy version of Reid's classical multiple hypothesis tracking (MHT) algorithm. Our fuzzy version is based on well-known fuzzy feedback systems, but the fact that the system we describe is specialized for likelihood discrimination makes this study particularly novel. We discuss several techniques for rule activation. One of them, namely, the *sum–product*, seems particularly useful for likelihood management and its linearity makes it tractable for further analysis. Our analysis is performed in two stages. First, we demonstrate that, with appropriately chosen rules, our system can discriminate the correct hypothesis. Second, the steady-state behavior with constant input is characterized analytically. This enables us to establish the optimality of the *sum–product* method and it also gives a simple procedure to predict the system's behavior as a function of the rule base. We believe this fact can be used to devise a simple procedure for fine-tuning the rule base according to the system designer needs. The application driving our fuzzy MHT implementation and analysis is the tracking of natural language text-based messages. That application is used as an example throughout the paper.**

***Index Terms*—Fuzzy feedback system, hypotheses discrimination, information tracking, multiple hypothesis tracking (MHT) algorithm, natural language processing.**

I. Introduction

NATURAL language messages are present in many information processing and analysis applications. However, to-date most systems for natural language processing have been used for database querying or machine translation. New and more powerful text processing techniques need to be developed and analyzed to handle other important applications that require correlation of text-based messages such as intelligence analysis, computer security incidents databases, and customer service reporting.

These applications have several common attributes: they involve tracking possibly ambiguous reports generated by different observers over time (in this context *tracking* means finding which messages deal with the same pieces of information and, therefore, they should be correlated somehow over time). Each such application also tends to be narrow in scope so a few important keywords should be carefully searched for and processed. These applications areas are all in need of more advanced automatic analysis techniques given the increasing amount of networked text-based information available to them.

TEXTTRACK, described in [1], is a software system whose goals are to apply advanced signal processing tracking concepts to natural language processing. TEXTTRACK addresses the problems of correlating and tracking observations of multiple moving vehicles reported by natural language messages that are generated by multiple observers asynchronously over time. The system has demonstrated that such problems can be tackled using relatively mature concepts from radar signal processing, namely the multiple hypothesis tracking (MHT) algorithm [20]. The prototype accepts simple natural language messages about vehicle types and locations, correlates the messages and associates groups of messages into the most likely tracks based on a succession of positions. The correlation procedure is solved in two steps: first, an appropriately modified, but still classical, Bayesian framework is used to handle the ambiguity in natural language descriptions. A formal theorem shows that under very mild conditions, the correct solution is eventually achieved. The second step uses a fuzzy inference engine (FIE), specifically, a fuzzy version of the classical Bayesian Reid's multiple hypothesis tracking algorithm. Since the purpose is to model natural language ambiguity, linguistic variables (i.e., *computing with words* in Zadeh's terminology [24]) are a natural choice for this purpose. However, [1] does not include a rigorous analytical study of the TEXTTRACK system. That work presented an intuitive argument for the system's effectiveness and was illustrated with several working examples.

In this paper, we give the fuzzy MHT algorithm originally developed in [1] a solid theoretical foundation by analytically characterizing the FIE on which the algorithm is based. Due to the fact that its mathematical characterization is application-independent, a natural byproduct of this paper is the broadening of the range of possible applications of the text-based MHT philosophy. That is, not only is it possible to track mobile man-made objects, but we will see it is possible to handle information about any time-varying phenomenon, as long as the phenomenon can be described by means of a few keywords, and the phenomenon itself is statistically causal in the sense that the distribution of future states is statistically dependent on past observed states.

The principal ingredient of the FIE arising in the MHT algorithm is a variant of well-known fuzzy feedback systems

Manuscript received April 6, 2001; revised October 6, 2001 and December 5, 2001. The work of S. Aja-Fernández and C. Alberola-López was supported in part by the Comisión Interministerial de Ciencia y Tecnología under Research Grants TIC97-0772 and 1FD97-0881 and by Junta de Castilla y León under Research Grants VA78/99 and VA91/01. The work of G. V. Cybenko was supported by the National Science Foundation under Grant 9813744 and DARPA Grant F30602-98-2-0107.

S. Aja-Fernández is with the Department of Teoría de la Señal y Comunicaciones e Ingenier a Telemática, University of Valladolid, 47011 Valladolid, Spain.

C. Alberola-López is with the Department of Teoría de la Señal y Comunicaciones e Ingeniería Telemática of the University of Valladolid, Spain, ETSI Telecomunicación, Campus Miguel Delibes, 47011 Valladolid, Spain (e-mail: caralb@tel.uva.es).

G. V. Cybenko is with the Thayer School of Engineering, Dartmouth College, Hanover, NH 03755 USA.

Publisher Item Identifier S 1063-6706(02)04829-4.

(FFSs). Traditionally such systems have been applied to control theory so that previous research has focused on issues specific to this theory. In our case, the quantities that are fed back to the system are likelihoods accumulated over time and the consequences of this fact will be explored in detail in the paper. Though initially motivated by TEXTTRACK, our study is a general and thorough mathematical analysis of single-fuzzy-input–single-fuzzy-output feedback systems[1] for hypotheses likelihood determination. Consequently, this system can be used in more applications other than just land vehicle tracking.

This paper is organized as follows. Section II gives a description of the MHT algorithm and its fuzzy version as well as a review of well-known techniques for rule activation. In Section III we demonstrate the ability of the *fuzzy MHT* to discriminate the most likely hypothesis in a worst-case scenario. This is made possible by the use of linear operators, such as the *sum–product*. Section IV is devoted to calculate the long-term behavior of the system with a constant input, which turns out to be a function of the rule database. Our results are validated in Section V by means of several examples.

We believe that the results obtained in Section IV give great insight into the system's behavior, and they provide a great deal of information relevant to the design of appropriate rule bases for specific applications even in the absence of data.

II. BACKGROUND

A. The MHT Algorithm

Reid's [20] MHT algorithm is a well known and widely used Bayesian approach to multiple target tracking. It is based on deferring decisions until enough evidence is collected to make a correct choice. MHT is implemented by explicitly storing as many hypotheses (i.e., possible classifications into tracks of all the vehicles so far observed) as possible, together with estimates of a probability measure of these hypotheses. When a decision is made, the hypothesis with the current highest likelihood is taken to be the truth.

Multiple hypothesis tracking algorithms are typically used in radar applications involving several sensors. The sensors' probability of detection is high so there is a large flow of information within the system. Sensors return *reports* at discrete time intervals, producing a *scan*. Reports from different scans are organized into *tracks* according to a probabilistic calculus based on the dynamics of the objects being detected. A track consists of reports of the same underlying object. A collection of consistent tracks is called a *hypothesis* and each such hypothesis has a likelihood. The goal is to maintain the most likely hypotheses bearing in mind that future reports may dramatically change the hypotheses' likelihoods.

To be more specific, and to highlight the recursive nature of the algorithm, suppose that at time instant k, a scan consists of M_k measurements, which are stored in vector $Z(k)$. Suppose that before receiving this set of measurements, a number of hypotheses Θ_n^{k-1}, i.e., a disjoint set of pre-established tracks, were stored (n ranges from 1 to the overall number of hypotheses at time $k-1$). Now, each of the M_k observations could be either a new observation corresponding to one of the existing tracks, a new target that appears in the sensor's field of view for the first time, or a false alarm due to clutter or thermal noise. The goal of the MHT algorithm is to effectively characterize these new M_k observations, i.e., to classify them correctly according to the above three cases, giving rise to a new set of hypotheses Θ_m^k. To that end, it is necessary to build several extended hypotheses derived from the most likely associations of new observations to existing tracks, to compute the new probabilities, and then to prune the less likely hypotheses (to satisfy finite storage limitations). The probabilities of the hypotheses are updated recursively as follows.

Hypothesis Θ_m^k, i.e., the mth hypothesis that includes the observations $Z(k)$, is built by means of appropriately appending some permutation, $\theta_m(k)$, of the new measurements to the hypothesis $\Theta_{l(m)}^{k-1}$ from the previous time. Thus

$$\Theta_m^k = \left\{\Theta_{l(m)}^{k-1}, \theta_m(k)\right\}. \tag{1}$$

Let Z^k denote the set of measurements until time instant k; it consists of

$$Z^k = \{Z^{k-1}, Z(k)\}. \tag{2}$$

The goal is to obtain

$$P\{\Theta_m^k | Z^k\} \tag{3}$$

that is, to calculate the probability of the mth active hypothesis after associating the observations collected until time instant k. Using Bayes rule, this probability can be shown to be proportional to the product of the following three probabilities

$$P\{\Theta_m^k|Z^k\} \propto P\left\{Z(k)\left|\Theta_{l(m)}^{k-1},\ \theta_m(k),\ Z^{k-1}\right.\right\} \cdot P\left\{\theta_m(k)\left|\Theta_{l(m)}^{k-1},\ Z^{k-1}\right.\right\} P\left\{\Theta_{l(m)}^{k-1}\left| Z^{k-1}\right.\right\}. \tag{4}$$

The last term of this equation is the probability of the parent hypothesis $P\{\Theta_{l(m)}^{k-1}|Z^{k-1}\}$ and is, therefore, available from the previous iteration. The first factor is the likelihood of the measurements given an association of measurements to tracks (which is computed by means of the information on the dynamics of the objects involved in the process) and the second factor is the probability of that specific assignment of observations to pre-existing tracks.

If text-descriptions (as opposed to radar sensor measurements) are collected, the foregoing scheme is valid provided that appropriate changes are made. The first factor of equation (4) is directly used since vehicle dynamics are assumed known; the third factor does not need any change either; the second factor, on the other hand, is understood as a *vehicle compatibility measurement*, i.e., how likely it is that an observer reports a *car* for example and the association is made to a track in which the vehicle is, possibly, different from a car (a jeep, a small van or others). The authors demonstrated in [1] that an MHT-like algorithm so built, will eventually reach a correct solution in a worst case scenario under a very mild condition, specifically, provided that the probability of reporting vehicle

[1] We will only be concerned with single-input single-output systems, so they will be hereafter referred to as FFS.

i when vehicle i is the actual vehicle is greater than the probability of reporting any other type of vehicle j with $j \neq i$.

B. A Fuzzy MHT Algorithm

Because the inputs to the MHT application described in last section are natural language text messages, a natural alternative to the Bayesian framework described above is a fuzzy MHT-like algorithm [1]. The different sources of ambiguity that arise in this problem, namely, the ambiguity in the description of the vehicles and the uncertainty in the prediction of future positions of the objects (since the vehicle dynamics are given in statistical terms) can be handled with fuzzy logic in a way that appears to be closer to the human reasoning.

It is clear, however, that a fuzzy MHT algorithm can be easily generalized to finding associations in a set of reports that describe, possibly with some ambiguity, some sort of reality about which previous knowledge is available. Denote by *objects* the entities to be tracked, and assume some knowledge is available about the objects' behavior (and thus some prediction of the future states of the objects is possible). In such a scenario, the sources of ambiguity would be

1) ambiguity in the description of the *objects*;
2) uncertainty in the prediction of the future state of these objects (some dynamical model is assumed known, perhaps only in statistical terms).

This ambiguity, inherent both in human natural language and in stochastic dynamical models, can be modeled by two linguistic variables, namely, *reported object* (in the case of vehicle tracking such objects will be *truck, van, car* …) and *prediction error* (*small, medium, large* …).[2] These two variables, together with a variable called *likelihood* (the values of which are labels such as *unlikely*, *very likely*, and so forth) can be fused together to create an MHT-like procedure by means of a triple fuzzy reasoning, motivated by (4) and driven by the following rules.

1) Rules that play the role of probability $P\{\theta_m(k)|\Theta_{l(m)}^{k-1}, Z^{k-1}\}$

Rule:	If $\mathcal{O}_1$ is $Class_i$
	and $\mathcal{O}_2$ is $Class_i$,
	then $\mathcal{C}_V$ is $Very\ Likely$.
Rule:	If $\mathcal{O}_1$ is $Class_i$
	and $\mathcal{O}_2$ is $Class_{i+1}$,
	then $\mathcal{C}_V$ is $Likely$.
…	…
Rule:	If $\mathcal{O}_1$ is $Class_k$
	and $\mathcal{O}_2$ is $Class_i$,
	then $\mathcal{C}_V$ is $Very\ Unlikely$.
Fact:	$\mathcal{O}_1$ is A'
	and $\mathcal{O}_2$ is B'
Conclusion:	$\mathcal{C}_V$ is C'

where, the facts $\mathcal{O}_1$ is A' and $\mathcal{O}_2$ is B' are to be understood as "the last object in the track under analysis is A' and the observation is B'."

2) With respect to probability $P\{Z(k)|\Theta_{l(m)}^{k-1}, \theta_m(k), Z^{k-1}\}$ the following rules are appropriate:

Rule:	If $\mathcal{C}_V$ is $Very\ Likely$
	and $\mathcal{E}$ is $Almost\ zero$,
	then $\mathcal{P}^k$ is $Very\ Likely$.
Rule:	If $\mathcal{C}_V$ is $Very\ Likely$
	and $\mathcal{E}$ is $Small$,
	then $\mathcal{P}^k$ is $Likely$.
…	…
Rule:	If $\mathcal{C}_V$ is $Very\ Unlikely$
	and $\mathcal{E}$ is $Excessive$,
	then $\mathcal{P}^k$ is $Very\ Unlikely$.
Fact:	$\mathcal{C}_V$ is A'
	and $\mathcal{E}$ is B'
Conclusion:	$\mathcal{P}^k$ is C'

where $\mathcal{C}_V$ is the likelihood derived from the first set of rules and $\mathcal{E}$ is the prediction error. $\mathcal{P}^k$ is the likelihood of the hypothesis.

3) Finally, the recursion with the hypothesis likelihood history

Rule:	If $\mathcal{H}^{k-1}$ is $Very\ Likely$
	and $\mathcal{P}^k$ is $Very\ Likely$,
	then $\mathcal{H}^k$ is $Very\ Likely$.
Rule:	If $\mathcal{H}^{k-1}$ is $Very\ Likely$
	and $\mathcal{P}^k$ is $Possible$,
	then $\mathcal{H}^k$ is $Likely$.
…	…
Rule:	If $\mathcal{H}^{k-1}$ is $Very\ Unlikely$
	and $\mathcal{P}^k$ is $Possible$,
	then $\mathcal{H}^k$ is $Unlikely$.
Fact:	$\mathcal{H}^k$ is A'
	and $\mathcal{P}^k$ is B'
Conclusion:	$\mathcal{H}^k$ is C'

where $\mathcal{H}^{k-1}$ is the hypothesis likelihood history at instant k.

These ideas can be graphically represented as in Fig. 1, where the first block gives the likelihood of the association of the current reported object with the stored object; the second block weighs this label with the prediction error, to obtain a second label of likelihood. This second label updates the overall likelihood by means of an inference with the likelihood accumulated until the previous time instant. As it can be seen, the overall hypothesis likelihood is fed back to the FIE, so we can naturally denote it as the *hypothesis history*.

[2]This prediction error must be understood as a measure of the mismatch between the predicted object state and the observed state. In the case of vehicle tracking it is a distance error. In other cases appropriate changes must be made.

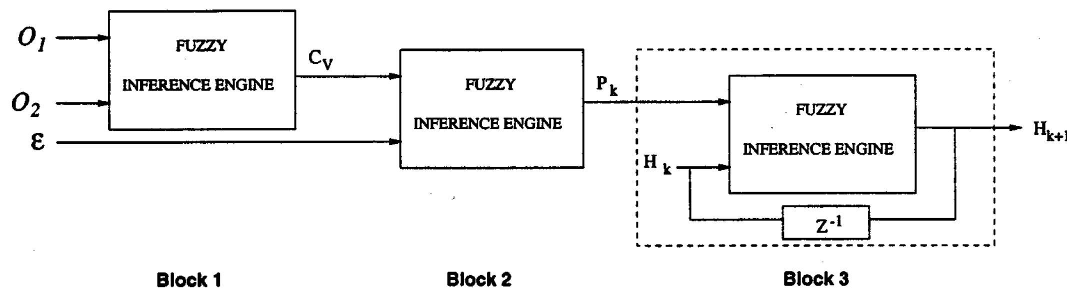


Fig. 1. A sketch of the fuzzy MHT system. $\mathcal{E}$: prediction error, $\mathcal{C}_V$: object compatibility, $\mathcal{P}_k$ probability of observation, $\mathcal{H}_k$: likelihood history.

The key of this system is the last block since it updates all the previous knowledge coherently with new incoming observations. As the first and the second blocks do not accumulate information (they work on "new entries" every time instant), the way they work is not critical, provided that probable associations are given higher likelihoods that unprobable. But the third block is the one which updates the system information using an FFS. A proper set of rules and fuzzy operations will have to be chosen to guarantee the correct update of the likelihood history.

We will demonstrate, in the worst case, that provided that the rule base is correctly designed, the system is able to discriminate the most likely hypothesis. In addition, we will also prove that the system output is stable when the input is constant, so we can guarantee the correct functioning of the whole system in time.

For the sake of clarity, we illustrate the procedure of the fuzzy MHT algorithm by means of two examples. First, we consider a land-vehicle tracking application, as in [1]. Over a known space, we receive text information on the position of these vehicles. The input patterns will be the description of the vehicles (*truck, van, car* ...). So the rules that play the role of $P\{\theta_m(k)|\Theta_{l(m)}^{k-1}, Z^{k-1}\}$ will be

Rule:	If $\mathcal{V}_1$ is $Truck$
	and $\mathcal{V}_2$ is $Truck$,
	then $\mathcal{C}_V$ is $Very\ Likely$.
Rule:	If $\mathcal{V}_1$ is $Truck$
	and $\mathcal{V}_2$ is Van,
	then $\mathcal{C}_V$ is $Likely$.
...	...
Rule:	If $\mathcal{V}_1$ is $Motorcycle$
	and $\mathcal{V}_2$ is $Truck$,
	then $\mathcal{C}_V$ is $Very\ Unlikely$.
Fact:	$\mathcal{V}_1$ is A'
	and $\mathcal{V}_2$ is B'
Conclusion:	$\mathcal{C}_V$ is C'.

The prediction error in this application will be the absolute difference between the predicted and the reported position of the vehicles. This measure could be fuzzified, so the input to the system will be a label such as *large*.

Another interesting (though still unimplemented) example would be the tracking of nonauthorized users on a computer network. When a user tries to enter a machine by a certain port, a message could be sent to the tracking system, which could predict the next step of the user. Input patterns to the system would be some kind of user identifier, and the prediction error could be represented by a linguistic variable which modeled the mismatch of the "next step" prediction.

C. System Inputs and Activation of Rules

The inputs to our system have the particularity of being linguistic variables, the values of which for the third block are likelihood labels. This obliges us to define a method for rule activation with fuzzy sets as inputs.

The problem is posed as finding the activation of a rule R_j

$$R_j(\mathcal{X})\text{: If } \mathcal{X} \text{ is } a_j(x), \qquad \text{then } \mathcal{Y} \text{ is } C_j(y)$$

when $A'(x)$ is input, with $a_j(x)$, $C_j(y)$ and $A'(x)$ fuzzy sets. The solution must be some sort of composition of $a_j(x) \circ A'(x)$ in a way that the intersection of the sets $a_j(x)$ and $A'(x)$ is weighed to end up with

$$R_j(A') = a_j(A')C_j. \tag{5}$$

Several strategies can be considered to solve the above mentioned problem. These strategies have been called *interpolation methods* elsewhere [12].

Definition 1 (Max–Min Method) [12], [16]: The composition is

$$a_j(A') = \sup_{x \in X} \{\mu_{a_j \cap A'}\} = \sup_{x \in X} \min\{a_j(x), A'(x)\}. \tag{6}$$

This method is the *Modus Tollens* proposed by Zadeh [23], and seems to be so far the most popular method.

Definition 2 (Sum–Product Method): Kosko [14] proposes the following rule activation method:

$$a_j(A') = \int_{R^n} A'(x) a_j(x)\, dx \tag{7}$$

for x a continuous variable. Its discrete version can be trivially rewritten as

$$a_j(A') = \frac{1}{K} \sum_i a_j(x_i) A'(x_i) \tag{8}$$

with K a normalizing factor.[3] This method, as opposed to the former, takes into account the area enclosed under the whole product of the two fuzzy sets, and not only the maximum value of the product. Kosko's operator is linear which makes analysis far more tractable, and it is the basis for SAMs procedures.

Definition 3 (Max–Product Method) [5]: An alternative to the foregoing proposals is a hybrid method expressed as

$$a_j(A') = \frac{1}{K} \max\{a_j(x)A'(x)\}. \tag{9}$$

This method seems to inherit the advantages of performing an intersection with a product, but does not consider the whole area under the intersection; the max operator will greatly reduce the tails that may show up in the output fuzzy set.

Definition (Sum–Min Method): A different possibility is to calculate the area under the intersection of the fuzzy sets $a_j(x)$ and $A'(x)$, as follows:

$$a_j(A') = \frac{1}{K} \sum_i \min\{a_j(x_i),\, A'(x_i)\} \tag{10}$$

or, for a continuously valued variable

$$a_j(A') = \int_{R^n} \min\{a_j(x),\, A'(x)\}\, dx. \tag{11}$$

III. Hypothesis Discrimination Capability of the Fuzzy MHT Algorithm

A fuzzy MHT system must be able to associate a greater likelihood to the actual hypothesis after enough information has been collected. In this section, we demonstrate that, if the rule base is correctly designed, the system proposed in [1] and generalized in Section II-B will correctly discriminate the most likely hypothesis versus others. As we have seen, the third block in Fig. 1 is in charge of making this discrimination.

Denote by P_m^k the likelihood label associated to the mth hypothesis that uses the data received at time instant k and by H_m^{k-1} the accumulated likelihood associated to this hypothesis until the previous time instant. H_m^k is the updated likelihood history.

The variable *likelihood* will be modeled as a linguistic variable, say *probability*, defined by means of several fuzzy sets (*likely, very likely, unlikely*, and so forth) [12] with *pseudotrapezoid-shaped* (PTS) membership functions [25], as is shown in Fig. 2.

P_m^k is then a fuzzy set with membership function $\mu_{P_m^k}(x)$ and H_m^{k-1} is a second fuzzy set with membership function $\mu_{H_m^{k-1}}(x)$. In order to avoid excessive notation, we will refer to the membership functions with the same symbol as the fuzzy set.

According to [25], if Ai are consistent and normal fuzzy sets in $U \subset R$ with PTS membership functions $A_i(x) = A_i(x;\; a_i, b_i, c_i, d_i)$ $(i = 1, 2, \ldots N)$, then there exists an ordering $\{i_1, i_2, \ldots, i_N\}$ in $\{1, 2, \ldots, N\}$ such that

$$A_{i_1} < A_{i_2} < \cdots < A_{i_N}. \tag{12}$$

[3] In subsequent sections, we will drop the x_i in the notation unless necessary, i.e., we will write $a_j(A') = (1/K)\sum a_j A'$.

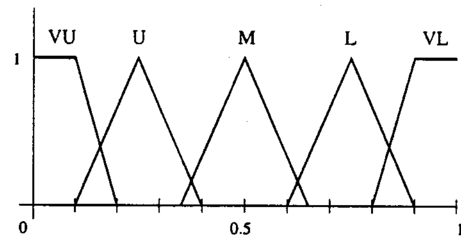


Fig. 2. Components of the linguistic variable *Likelihood*.

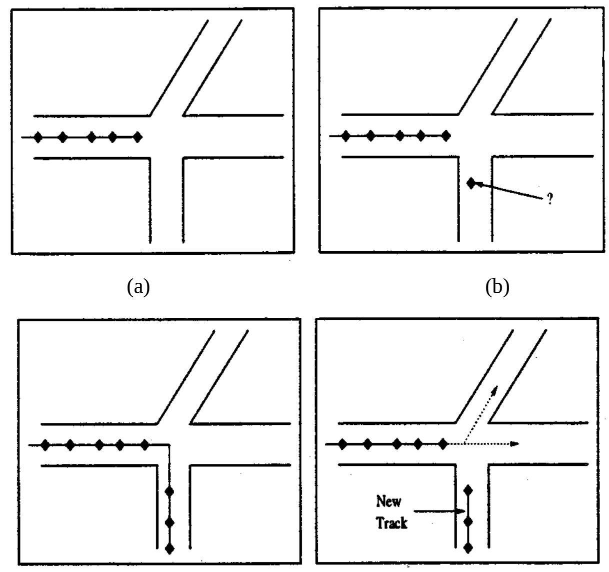


Fig. 3. An illustration of an ambiguous situation. (a) A track associated to a single vehicle exists. (b) A new observation from the same vehicle comes up. (c) Hypothesis 1 (right): The track is enlarged with the new observation. (d) Hypothesis 2 (wrong): A new track is created. Two vehicles are now assumed to be present in the scenario.

In our case

$$A_0 = VU < A_1 < \cdots < A_M = VL.$$

The fuzzy MHT algorithm should guarantee that the accumulated likelihoods of two hypotheses are ordered $H_1^k > H_2^k$ provided that at time instant k the hypothesis Θ_1^k is more likely than hypothesis Θ_2^k. In order to demonstrate that this is actually the case, we consider a worst-case scenario (see Fig. 3), in which two hypotheses only differ in the probability of one assignment.

1) At time instant t the system has one hypothesis in memory, with some likelihood label. Assume that, without loss of generality, the likelihood label is the maximum, i.e., $H^{t-1} = A_M$.
2) At this time, and observation comes up. Assume two different associations of the observation to tracks are possible, giving rise to two possible hypotheses.
   a) Hypothesis Θ_1^t, from which a label of likelihood $P_1^t = A_M$ is calculated (by the two first blocks in Fig. 1), with A_M the maximum fuzzy set of the linguistic variable, i.e., *very likely*.
   b) Hypothesis Θ_2^t, from which a label of likelihood $P_2^t = A_K$ is calculated, with $A_k \neq A_M$.

3) Subsequent observations will give rise in both hypotheses to the maximum value of likelihood, i.e., $P_1^{t+i} = P_2^{t+i} = A_M$ with $i > 0$.

Fig. 3 depicts graphically the problem statement for the case of *vehicle tracking*. In Fig. 3(a), a track associated to a single vehicle exists (H^{t-1}); for instance, observations through time of a truck. Suppose a new observation of the same vehicle comes up [Fig. 3(b)]; however, due to the ambiguity in the description and the right turn of the vehicle (which is difficult to predict), two hypotheses are sensible to be considered.

- Hypothesis 1 (Θ_1^t): the new observation is considered to come from the same truck, which has made a right turn [Fig. 3(c)]. The track is enlarged with this new observation.
- Hypotheses 2 (Θ_2^t): the new observation is in this case considered as the first report of a second vehicle which has just appeared in the field of view of the reporter. No information is assumed to be given about the truck in this time instant [Fig. 3(d)].

Theorem: We state that the fuzzy MHT algorithm is able to discriminate that

$$H_1^k > H_2^k, \qquad \forall\, k \geq t.$$

Proof: Our demonstration will have two stages. The first step shows that we can state that $H_1^t > H_2^t$. The second step is an inductive method derived from the former, and allows us to order the hypotheses at arbitrary future time instants.

A. First Stage: Analysis at Time Instant t

Consider a reasoning with a SAM (as proposed by [13] and [14]) inference engine and the *sum–product* method for rule activation. We have chosen these methods because they both are linear, making a formal study easier; furthermore, we follow the conclusions presented in [1], where the results using Kosko's operators were similar to those of Bayesian case, versus the max–min approach which had the poorest results.

For rules as the following:

$$R_j\text{: If } H_{l(m)}^{k-1} \text{ is } a_j \quad \text{and} \quad P_m^k \text{ is } b_j, \qquad \text{then } \mathcal{Y} \text{ is } C_j$$

we can write that at time instant k

$$H_m^k = \sum_j C_j a_j \left(H_{l(m)}^{k-1}\right) b_j(P_m^k) \tag{13}$$

which can be further expressed

$$H_m^k = \frac{1}{K_1} \sum_j C_j \left(\sum a_j H_{l(m)}^{k-1}\right) \left(\sum b_j P_m^k\right). \tag{14}$$

According to our problem statement, the likelihood at time instant $t-1$, $H_1^{t-1} = A_M$, so, according to (14) we can write

$$H_1^t = \frac{1}{K_1} \sum_j C_j \left(\sum a_j A_M\right) \left(\sum b_j A_M\right) \tag{15}$$

and

$$H_2^t = \frac{1}{K_1'} \sum_j C_j \left(\sum a_j A_M\right) \left(\sum b_j A_K\right). \tag{16}$$

A further step needs the following.

Proposition 1: If A_i are consistent and normal fuzzy sets that give rise to a complete partition of $U \subset R$, with PTS membership functions $A_i(x) = A_i(x;\ a_i, b_i, c_i, d_i)$ $(i = 1, 2, \ldots, N)$, each membership function, but those of the smallest and the greatest set, will intersect one and only one membership function in every extreme.

Remark 1: Each fuzzy set intersects with two sets, each at every side of the maximum of the set, but the sets A_1 and A_N will only intersect one fuzzy set.

Proof: Since we are dealing with normal and consistent fuzzy sets each set will only intersect with one set because, otherwise, the set would have a nonnull value in the normal subset of the other set, and thus the property of consistency would not hold. In addition, since the partition is complete, two consecutive sets should intersect.

In the demonstration, we assume, without loss of generality that $A_K = A_{M-1}$. If the property holds for $A_K = A_{M-1}$ it will necessarily hold for $A_K < A_{M-1}$.

For Θ_1^t, since proposition 1 holds, the following table of activated rules applies:

$a_j(x)$	$b_j(x)$	C_j	$R_j = a_j(A_M)b_j(A_M)c_j$
A_M	A_M	C_I	c_I
A_M	A_{M-1}	C_{II}	$\delta_1 c_{II}$
A_{M-1}	A_M	C_{III}	$\delta_1 c_{III}$
A_{M-1}	A_{M-1}	C_{IV}	$\delta_1^2 c_{IV}$

with $c_I, c_{II}, \ldots$ the centroids of the fuzzy sets C_j. With respect to the antecedent composition, Proposition 1 allows us to write

$$A_i \circ A_j = \begin{cases} =1, & \text{if } i = j \\ =\delta_i, & \text{if } |i-j| = 1 \\ =0, & \text{if } |i-j| > 1 \end{cases} \tag{17}$$

with $\delta_i < 1$. In our case, $A_M \circ A_{M-1} = \delta_1$ and $A_{M-1} \circ A_{M-2} = \delta_2$, where both δ_i depend on the actual activation method (max–min, sum product, and so forth).

Similarly, for Θ_2^t

$a_j(x)$	$b_j(x)$	C_j	$a_j(A_M)b_j(A_{M-1})c_j$
A_M	A_M	C_I	$\delta_1 c_I$
A_M	A_{M-1}	C_{II}	c_{II}
A_{M-1}	A_M	C_{III}	$\delta_1^2 c_{III}$
A_{M-1}	A_{M-1}	C_{IV}	$\delta_1 c_{IV}$
A_M	A_{M-2}	C_V	$\delta_2 c_V$
A_{M-1}	A_{M-2}	C_{VI}	$\delta_1 \delta_2 c_{VI}.$

The crisp likelihood value will be obtained by the method of centroids, i.e.,

$$c(A', B') = \frac{\sum_j a_j(A')b_j(B')c_j}{\sum_j a_j(A')b_j(B')} \tag{18}$$

which in our case results in

$$c_1 = \frac{c_I + \delta_1 c_{II} + \delta_1 c_{III} + \delta_1^2 c_{IV}}{(1+\delta_1)^2} \tag{19}$$

for H_1^t and

$$c_2 = \frac{c_{II} + \delta_1 c_I + \delta_1 c_{IV} + \delta_2 c_V + \delta_1\delta_2 c_{VI} + \delta_1^2 c_{III}}{(1+\delta_1)(1+\delta_1+\delta_2)} \tag{20}$$

for H_2^t. Mild restrictions are needed in our rule base to guarantee correct discrimination. Since our system is a likelihood comparison system, greater inputs draw greater outputs. Specifically, output for inputs A_M and A_M should be greater than output for inputs A_{M-1} and A_{M-1}. When inputs are A_M and A_{M-1} we can only state that the output will be less than or equal to the output for inputs A_M and A_M. These restrictions give rise to the following relations: $c_I \geq c_{II}$, $c_I \geq c_{III}$, $c_I > c_{IV}$, $c_I \geq c_V$, $c_I > c_{VI}$, $c_{II} \geq c_{IV}$, $c_{II} \geq c_V$, $c_{II} \geq c_{VI}$, $c_{III} \geq c_{IV}$, $c_{III} \geq c_V$, $c_{III} \geq c_{VI}$, $c_{IV} \geq c_{VI}$, and $c_V \geq c_{VI}$.

Centroids can be compared by writing

$$\begin{aligned} c_1 &= \frac{(c_I + \delta_1 c_{II} + \delta_1 c_{III} + \delta_1^2 c_{IV})(1+\delta_1+\delta_2)}{(1+\delta_1)^2(1+\delta_1+\delta_2)} \\ &= \frac{N_1}{(1+\delta_1)^2(1+\delta_1+\delta_2)} \end{aligned} \tag{21}$$

$$\begin{aligned} c_2 &= \frac{(c_{II}+\delta_1 c_I+\delta_1 c_{IV}+\delta_2 c_V+\delta_1\delta_2 c_{VI}+\delta_1^2 c_{III})(1+\delta_1)}{(1+\delta_1)^2(1+\delta_1+\delta_2)} \\ &= \frac{N_2}{(1+\delta_1)^2(1+\delta_1+\delta_2)}. \end{aligned} \tag{22}$$

If we calculate the difference of the numerators

$$\begin{aligned} N_1 - N_2 = &(c_I - c_{II}) + \delta_1(c_{III} - c_{IV}) \\ &+ \delta_2(c_I - c_V) + \delta_1^2(c_{II} - c_I) \\ &+ \delta_1\delta_2(c_{II} - c_V + c_{III} - c_{VI}) \\ &+ \delta_1^3(c_{IV} - c_{III}) + \delta_1^2\delta_2(c_{IV} - c_{VI}) \end{aligned}$$

and making use of the relations just mentioned

$$\begin{aligned} (c_I - c_{II}) &\geq 0 & \delta_1(c_{III} - c_{IV}) &\geq 0 \\ \delta_2(c_I - c_V) &\geq 0 & \delta_1^2(c_{II} - c_I) &\leq 0 \\ \delta_1\delta_2(c_{II} - c_V) &\geq 0 & \delta_1\delta_2(c_{III} - c_{VI}) &\geq 0 \\ \delta_1^3(c_{IV} - c_{III}) &\leq 0 & \delta_1^2\delta_2(c_{IV} - c_{VI}) &\geq 0 \end{aligned} \tag{23}$$

we observe that all the elements are positive but $\delta_1^2(c_{II} - c_I)$ and $\delta_1^3(c_{IV} - c_{III})$. However, the following inequalities hold:

$$\begin{aligned} (c_I - c_{II}) &\geq \delta_1^2(c_I - c_{II}) \\ \delta_1(c_{III} - c_{IV}) &\geq \delta_1^3(c_{III} - c_{IV}) \end{aligned} \tag{24}$$

since $\delta_1 \leq 1$ and $\delta_2 \leq 1$. Therefore, we can write

$$\begin{aligned} (c_I - c_{II}) - \delta_1^2(c_I - c_{II}) &\geq 0 \\ \delta_1(c_{III} - c_{IV}) - \delta_1^3(c_{III} - c_{IV}) &\geq 0 \end{aligned} \tag{25}$$

and consequently $N_1 - N_2 \geq 0$, or equivalently, $c_1 \geq c_2$.

The equality is obtained when $c_I = c_{II} = c_V$ and $c_{III} = c_{IV} = c_{VI}$, which is avoidable by a proper selection of the rule base. If we assume the rule base is properly designed, we can state

$$H_2^{t+1} < H_1^{t+1}. \tag{26}$$

Q.E.D.

B. Second Stage: Likelihoods at Arbitrary Future Time Instants

The problem is now posed as follows.

- The two hypotheses Θ_1^t and Θ_2^t have accumulated likelihoods H_1^t and H_2^t, respectively, and we have shown that $H_2^t < H_1^t$.
- Suppose now that $P_1^{t+i} = P_2^{t+i} = A_M$, $\forall\, i > 0$, i.e., that subsequent observations causes the maximum output in the second block of Fig. 1 for both hypotheses.

Our goal is to demonstrate that $H_2^k < H_1^k$ for $\forall\, k = t + t_1 > t$. This can be easily done by simple induction, using the result of the previous stage.

Defining the following parameters:

$$\begin{aligned} P_{a_j} &= \sum a_j A_M \\ P_{b_j} &= \sum b_j A_M \\ M_{a_j} &= \sum a_j A_{M-1} \end{aligned} \tag{27}$$

the accumulated histories at time instant $t + t_1$ are

$$\begin{aligned} H_1^{t+t_1} = &\frac{1}{K_{t_1}} \sum_j C_j P_{b_j} \\ &\cdot \left(\cdots \sum_j C_j P_{b_j} \left(\sum a_j \left(\sum_j C_j P_{b_j} \right.\right.\right. \\ &\left.\left.\left.\cdot \left(\sum a_j \left(\sum_j C_j P_{b_j} P_{a_j} \right)\right)\right)\right) \cdots \right) \end{aligned} \tag{28}$$

$$\begin{aligned} H_2^{t+t_1} = &\frac{1}{K_{t_1}} \sum_j C_j P_{b_j} \\ &\cdot \left(\cdots \sum_j C_j P_{b_j} \left(\sum a_j \left(\sum_j C_j P_{b_j} \right.\right.\right. \\ &\left.\left.\left.\cdot \left(\sum a_j \left(\sum_j C_j P_{b_j} M_{a_j} \right)\right)\right)\right) \cdots \right) \end{aligned} \tag{29}$$

which can be written recursively as

$$\begin{aligned} H_1^{t+t_1} &= \frac{1}{K_{t_1}} \sum_j C_j P_{b_j} \left(\sum a_j H_1^{t+t_1-1} \right) \\ H_2^{t+t_1} &= \frac{1}{K_{t_1}} \sum_j C_j P_{b_j} \left(\sum a_j\, H_2^{t+t_1-1} \right). \end{aligned} \tag{30}$$

From these two equations it is obvious that if $H_2^t < H_1^t$ holds, then $H_2^{t+t_1} < H_1^{t+t_1}$ must necessarily hold. Q.E.D.

The behavior of the system when $t_1 \rightarrow \infty$ in (30) is not obvious; common sense dictates that if more observations with the greatest likelihood are given to both hypotheses, the overall likelihood of both hypotheses should increase, they should be progressively closer to the maximum and, consequently, the differences between both should decrease. In the next section, we accomplish such an study and we obtain closed-form expressions of the long term behavior of the system when the input is held constant. Interesting comparative conclusions can be drawn about the methods used for rule activation.

IV. System Analysis for a Steady Input

As previously mentioned, the key of the system is the third block in Fig. 1, which is known as an FFS. This block is critical since it performs the accumulation of likelihoods in time, so it is the responsible of assuring the long term behavior of the algorithm.

FFSs have been studied in the literature. Some brief background material follows.

A. FFS

FFSs have been traditionally used as controllers, where they have demonstrated their effectiveness in a myriad of applications. The stability of these controllers is usually studied by nonlinear analysis techniques, such as Lyapunov's methods [9], [14], [17], which give a confidence of the reliability of the system, but they do not characterize its recursive behavior. So, the design of these controllers has been traditionally done *ad hoc*, due to the difficulty in developing a theoretical basis that mathematically characterizes that behavior.

There have been several interesting attempts to rigorously formalize FFSs. The first one is due to Tong [22] in 1980. The FFS proposed is the one shown in Fig. 4.

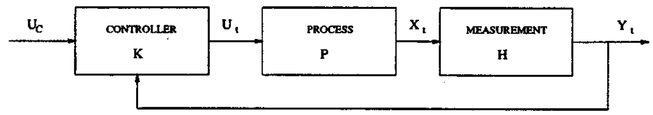


Fig. 4. Closed-loop system (Tong).

The author obtains a theoretical description of the closed loop response as a function of the initial state, specifically, $X_{t+1} = X_0 \circ S_t$, where $\circ$ is a mapping defined by a *max–min* composition. The transfer function S_t depends on the constant input U_c and on the system components (H, P and K), which are defined as fuzzy relations. Due to the difficulty to solve some complex relational equations to find S_t, the problem has only solution under certain conditions in which the author forces the controller K into some desired form.

Despite the fact that the result is not directly applicable to others systems in use, it is clearly a first approach to a rigorous characterization of the FFS.

An interesting further effort in this direction is made by Chen and Tsao in [4]. The authors state that *the main cause of the failure of FFSs is due to the use of the max–min operator*. This operator has the side effect of flattening the fuzzy set membership functions. When a FFS is described recursively, the accumulation of this effect makes the monitorization of the system evolution an impossible task. To overcome this problem, the authors propose the use of concepts of nonlinear system analysis, specifically, the *cell-to-cell mapping* developed by Hsu [10], to analyze the global behavior of nonlinear dynamical systems.

The FFS used by the authors is represented in Fig. 5, and it has been used as the paradigm of fuzzy control by others [7], [17]. The input is basically the difference (the error) between the output of the plant and a control signal. The input–output relation is expressed recursively by $Y_{t+1} = f_1(Y_t)$, with f_1 a relation derived from the rule set.

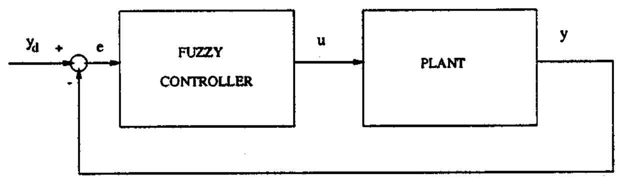


Fig. 5. Fuzzy control system (Chen–Tsao).

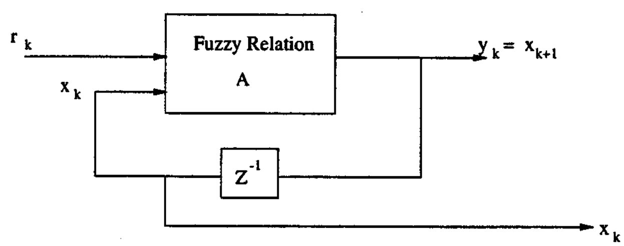


Fig. 6. Single-input–single-output fuzzy dynamic system (Kang).

Broadly speaking, the key in [4] is to map the fuzzy system into a nonfuzzy system so that its behavior can be better understood. This way, the global behavior of the system dynamics can be extracted by Hsu's method. The problem of the accumulation of *fuzziness* in each iteration is circumvented, since the fuzzy part is restricted to the transformation from one domain to the other. However, this method can only give an approximate prediction of the behavior of the system; furthermore, the cell-to-cell method cannot be applied to all dynamical fuzzy systems, and only *intuitive* criteria to find out which systems fit within this framework are given in [4].

Kang [11], in the early 1990s, proposed a systematic design method of linguistic fuzzy controllers. His control system is shown in Fig. 6.

When the input r_k is suppressed, the author can express the system output with the recursive equation $X_{k+1} = A^k \circ X_0$ (the matrix composition is not a power, but a kth composition of the form $A^k = A \circ A \circ \cdots \circ A$), with A a transition matrix which depends on the rule set and on the fuzzy sets membership functions. This matrix has a size $m \times m$, being m the size of the discrete fuzzy sets considered. The operator *max–min* is always used. The author states that if there exists a positive integer n from which $A^{n+1} = A^n$ then X_k reaches a steady-state, i.e., $X_{n+1} = X_n$. The author forces a fuzzy relation K (not indicated in Fig. 6) between the two inputs, r_k and X_k, to guarantee the system stability.

Although the three papers give a good theoretical basis for the analysis of stability of FFS, they are not suitable for our purpose. The first two give only an approximate solution of the system behavior, and the stability relation K proposed in the third method does not fit our system. Therefore, in next section we develop a different and complete steady state analysis, but that will make use of some of the ideas previously proposed.

Further information on the area of fuzzy control and FFSs can be found in [2], [17], and [18]. Other sources of introductory material and related topics on FFS are [6], [7], [15], [19], and [21].

B. The Fuzzy MHT Architecture as a FFS

As we have seen, the third block in Fig. 1 is a FFS, similar to the one proposed in [11] (Fig. 6). However, our system has a number of particularities that should be taken into account in any analysis. First, it has a single input (the *likelihood* label) but two entries to the FIE: the system input, and the accumulated history of likelihoods that comes directly from the feedback. Second, the input and the output of this block are all fuzzy sets.

According to [4], where the authors state that *the main cause of the failure of FFSs is due to the use of the max–min operator*, we have built the FIE upon linear operators, such as SAM and *sum–product*, as described in Section III. This eases considerably the process of theoretically describing the system behavior. Furthermore, as we will show in Section V, the *sum–product* is the only method that gives an acceptable response from the system.

C. Analysis for a Steady Input: General Case

We will use a similar paradigm as others previously reported [11] (Section IV-A): we will keep an input constant and we will study the system output as it evolves in time. We will search a recursive input–output relation and we will end up with a transition matrix between states. A composition of the matrix gives the output at arbitrary future time instants, so, at infinity, the steady-state behavior will be obtained.

As we have previously done, we will consider the linguistic variable likelihood, consisting of M fuzzy sets $A_1, A_2, \dots A_M$ with a normal PTS membership function, and the sets constitute a consistent partition. If we use the SAM philosophy, at time instant, say 0, a hypothesis Θ_l^0 will have a likelihood given by

$$H_l^0 = \frac{1}{K}(\beta_M^0 A_M + \beta_{M-1}^0 A_{M-1} + \cdots + \beta_2^0 A_2 + \beta_1^0 A_1) \quad (31)$$

with β_i^0 a set of coefficients which depend on the rule activation method, and K is a normalizing constant which turns out to be $K = \max_i\{\beta_i^0\}$.

The system output at time instant 1 for a given input can be written

$$H_l^1 = \frac{1}{K_2}(\beta_M^1 A_M + \beta_{M-1}^1 A_{M-1} + \cdots + \beta_2^1 A_2 + \beta_1^1 A_1)$$

where the coefficients depend on the rule activation method. The coefficients β_i^1 will be function of the coefficients in the previous time step β_i^0

$$\begin{aligned} \beta_M^1 &= f_M(\beta_1^0, \beta_2^0, \dots, \beta_M^0) \\ \beta_{M-1}^1 &= f_{M-1}(\beta_1^0, \beta_2^0, \dots, \beta_M^0) \\ &\vdots \quad \vdots \\ \beta_2^1 &= f_2(\beta_1^0, \beta_2^0, \dots, \beta_M^0) \\ \beta_1^1 &= f_1(\beta_1^0, \beta_2^0, \dots, \beta_M^0). \end{aligned}$$

As we use a linear activation method (*sum–product*), the relation between coefficients β_i^0 and β_i^1 will be also linear as follows:

$$\begin{aligned} \beta_M^1 &= f_M(\beta_1^0, \beta_2^0, \dots, \beta_M^0) = \sum_i r_{Mi}\beta_i^0 \\ \beta_{M-1}^1 &= f_{M-1}(\beta_1^0, \beta_2^0, \dots, \beta_M^0) = \sum_i r_{(M-1)i}\beta_i^0 \\ &\vdots \qquad \vdots \\ \beta_2^1 &= f_2(\beta_1^0, \beta_2^0, \dots, \beta_M^0) = \sum_i r_{2i}\beta_i^0 \\ \beta_1^1 &= f_1(\beta_1^0, \beta_2^0, \dots, \beta_M^0) = \sum_i r_{1i}\beta_i^0. \end{aligned}$$

We can generalize the foregoing expression to any time instant, provided that the input is held constant

$$\begin{aligned} \beta_M^k &= f_M(\beta_1^{k-1}, \beta_2^{k-1}, \dots, \beta_M^{k-1}) = \sum_i r_{Mi}\beta_i^{k-1} \\ \beta_{M-1}^k &= f_{M-1}(\beta_1^{k-1}, \beta_2^{k-1}, \dots, \beta_M^{k-1}) = \sum_i r_{(M-1)i}\beta_i^{k-1} \\ &\vdots \qquad \vdots \\ \beta_2^k &= f_2(\beta_1^{k-1}, \beta_2^{k-1}, \dots, \beta_M^{k-1}) = \sum_i r_{2i}\beta_i^{k-1} \\ \beta_1^k &= f_1(\beta_1^{k-1}, \beta_2^{k-1}, \dots, \beta_M^{k-1}) = \sum_i r_{1i}\beta_i^{k-1}. \end{aligned}$$

These linear relations suggest that a transition matrix Ω (as in [11], but here we use linear operators) can be defined so

$$[\beta_M^k \quad \beta_{M-1}^k \quad \cdots \beta_1^k]^T = \Omega[\beta_M^{k-1} \quad \beta_{M-1}^{k-1} \quad \cdots \beta_1^{k-1}]^T. \quad (32)$$

Proceeding recursively

$$[\beta_M^k \quad \beta_{M-1}^k \quad \cdots \beta_1^k]^T = \Omega^k[\beta_M^0 \quad \beta_{M-1}^0 \quad \cdots \beta_1^0]^T. \quad (33)$$

Matrix Ω is diagonizable, since it is the matrix of a system for which a solution always exists. So, if we write

$$\Omega^k = B\Lambda^k B^{-1} \quad (34)$$

with Λ a diagonal matrix (the entries of which are the eigenvalues of matrix Ω) (33) becomes

$$[\beta_M^k \quad \beta_{M-1}^k \quad \cdots \beta_1^k]^T = B\Lambda^k B^{-1}[\beta_M^0 \quad \beta_{M-1}^0 \quad \cdots \beta_1^0]^T \quad (35)$$

and the output crisp value is easily found as the centroid of the fuzzy set

$$C_k = \frac{[c_M \quad \cdots \quad c_2 \quad c_1]B\Lambda^k B^{-1}[\beta_M^0 \quad \beta_{M-1}^0 \quad \cdots \beta_1^0]^T}{[1 \quad \cdots \quad 1 \quad 1]B\Lambda^k B^{-1}[\beta_M^0 \quad \beta_{M-1}^0 \quad \cdots \beta_1^0]^T} \quad (36)$$

where the c_i are the centroids of the fuzzy sets A_i.

The problem now is to calculate the limiting value of equation (36) when $k \to \infty$. To that end, the *power method for the dominant eigenvalue* [8] can be readily applied; assume the largest eigenvalue in matrix Ω is located at the hth row of matrix Λ, i.e., $|\lambda_{\mathrm{MAX}}| = |\lambda_h| > |\lambda_p|, \forall p \neq h$. We can write (34) as

$$\Omega^k = \begin{pmatrix} b_{11} & b_{21} & \cdots & b_{M1} \\ b_{12} & b_{22} & \cdots & b_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1M} & b_{2M} & \cdots & b_{MM} \end{pmatrix} \lambda_h^k \cdot \begin{pmatrix} \left(\frac{\lambda_1}{\lambda_h}\right)^k & & & & & \\ & \left(\frac{\lambda_2}{\lambda_h}\right)^k & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & \left(\frac{\lambda_m}{\lambda_h}\right)^k \end{pmatrix} \cdot \begin{pmatrix} b'_{11} & b'_{21} & \cdots & b'_{M1} \\ b'_{12} & b'_{22} & \cdots & b'_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ b'_{1M} & b'_{2M} & \cdots & b'_{MM} \end{pmatrix}. \tag{37}$$

As can be seen, as $k \to \infty$ all the entries in the diagonal matrix tend to zero, but the one in the hth row. Therefore, after some algebra, the centroid turns out to be

$$C_k = \frac{[c_M \quad \cdots \quad c_2 \quad c_1]\lambda_h^k \begin{bmatrix} b_{h1} \sum_{i=1}^{M} b'_{ih}\beta^0_{M+i-1} \\ b_{h2} \sum_{i=1}^{M} b'_{ih}\beta^0_{M+i-1} \\ \vdots \\ b_{hM} \sum_{i=1}^{M} b'_{ih}\beta^0_{M+i-1} \end{bmatrix}}{[1 \quad \cdots \quad 1 \quad 1]\lambda_h^k \begin{bmatrix} b_{h1} \sum_{i=1}^{M} b'_{ih}\beta^0_{M+i-1} \\ b_{h2} \sum_{i=1}^{M} b'_{ih}\beta^0_{M+i-1} \\ \vdots \\ b_{hm} \sum_{i=1}^{M} b'_{ih}\beta^0_{M+i-1} \end{bmatrix}} \tag{38}$$

i.e.,

$$C_k = \frac{\sum_{j=1}^{M} c_{(M+1-j)} b_{hj}}{\sum_{j=1}^{M} b_{hj}} \tag{39}$$

with c_{M+1-j} the centroids of the output space fuzzy sets and $b_h = b_{\mathrm{MAX}} = [b_{h1} \cdots b_{hM}]^T$ the eigenvector associated to λ_{MAX} [remember that λ_i are the eigenvalues of the transition matrix between the system output at time j and $j+1$, defined in (32)]. Note that the dependence with β_j^0 has disappeared, i.e., there is not any dependence on the initial state.

Several conclusions can be drawn from (39).

- The convergence value of the system with a steady input is independent of the initial system state, and it only depends on the input value and on the rule base.
- The convergence value is a weighted sum of the centroids of the rule base. The weighting coefficients are the components of the eigenvector associated to the maximum eigenvalue of the system matrix. Moreover, these values depend on the rule activation method applied. Therefore, for a given data set we will obtain different convergence values according to the activation method.

Equation (39) is, in fact, the operation that calculates a centroid from the involved fuzzy set. This lets us state that this is the centroid of $H_\infty = \sum_j b_{\mathrm{MAX}j} A_{M+1-j}$. Consequently, we can state that the steady-output fuzzy set of our FFS with constant input is

$$H_\infty = \beta_M^\infty A_1 + \cdots + \beta_2^\infty A_{M-1} + \beta_1^\infty A_M \tag{40}$$

with β_j^∞ the components of the eigenvector associated to the largest eigenvalue of the system matrix Ω.

Finally, we can conclude that we can know *a priori* the state that the system will converge to with the only knowledge of the rule set. Conversely, this result enables us to fine tune the rule set so as to converge to a desired state when the input is held constant. To further illustrate convergence, we consider a particular case.

D. A Particular Case: A Three-Set Space

We study the case of the linguistic variable likelihood with three values $A_1 < A_2 < A_3$, expressed in an increasing order of plausibility. For simplicity, since normal and consistent PTS fuzzy sets are considered, we will assume that

$$A_i \circ A_j = \begin{cases} =1, & \text{if } i = j \\ =\delta, & \text{if } |i-j| = 1 \\ =0, & \text{if } |i-j| > 1. \end{cases} \tag{41}$$

The history at time instant 0 will be

$$H^0 = A_3\beta_3^0 + A_2\beta_2^0 + A_1\beta_1^0.$$

We use the (fairly strict) set of rules shown in the first three columns of Table I. We also assume the input is constant and equal to A_3; with this input, every rule is activated as shown in the last column of Table I. The activated rules give the relation among coefficients shown in Table II; matrix Ω turns out to be

$$\Omega = \begin{pmatrix} 1 & \delta & 0 \\ \delta(2+\delta) & (1+\delta)^2 & (1+\delta+\delta^2) \\ 0 & \delta^2 & \delta \end{pmatrix}. \tag{42}$$

Centroids are calculated as in (36)

$$C_k = \frac{[c_3 \quad c_2 \quad c_1] B\Lambda^k B^{-1} [\beta_3^0 \quad \beta_2^0 \quad \beta_1^0]^T}{[1 \quad 1 \quad 1] B\Lambda^k B^{-1} [\beta_3^0 \quad \beta_2^0 \quad \beta_1^0]^T}. \tag{43}$$

TABLE I
RULE BASE FOR THE EXAMPLE (THREE LEFT COLUMNS) AND ACTIVATION OF RULES WHEN THE INPUT IS A_3 (RIGHT COLUMN)

a_j	b_j	C_j	R_j
A_3	A_3	A_3	$A_3(\beta_3^0+\delta\beta_2^0)$
A_3	A_2	A_2	$A_2(\beta_3^0+\delta\beta_2^0)\delta$
A_3	A_1	A_2	0
A_2	A_3	A_2	$A_2(\delta\beta_3^0+\beta_2^0+\delta\beta_1^0)$
A_2	A_2	A_2	$A_2(\delta\beta_3^0+\beta_2^0+\delta\beta_1^0)\delta$
A_2	A_1	A_1	0
A_1	A_3	A_2	$A_2(\beta_1^0+\delta\beta_2^0)$
A_1	A_2	A_1	$A_1(\beta_1^0+\delta\beta_2^0)\delta$
A_1	A_1	A_1	0

TABLE II
COEFFICIENT RELATION

$$\begin{aligned}\beta_3^1 &= \beta_3^0+\delta\beta_2^0\\ \beta_2^1 &= \beta_3^0(2\delta+\delta^2)+\beta_2^0(1+2\delta+\delta^2)+\beta_1^0(1+\delta+\delta^2)\\ \beta_1^1 &= \beta_2^0\delta^2+\beta_1^0\delta\end{aligned}$$

For our steady input we can write (44) as shown at the bottom of the page.[4] The limit of the foregoing expression can be calculated by analyzing the asymptotic behavior of each of the matrix eigenvalues. After some algebra, the behavior of the eigenvalues is

$$\lim_{k\to\infty}\lambda_1^k=\lim_{k\to\infty}\left(1+\left(1+\sqrt{3}\right)\delta+\left(1+\frac{\sqrt{3}}{2}\right)\delta^2 -\frac{7}{8\sqrt{3}}d^3+O(\delta^4)\right)^k=\infty \tag{45}$$

$$\lim_{k\to\infty}\lambda_2^k=\lim_{k\to\infty}(\delta-\delta^2-\delta^4+O(\delta^5))^k=0 \tag{46}$$

$$\lim_{k\to\infty}\lambda_3^k=\lim_{k\to\infty}\left(1+\left(1-\sqrt{3}\right)\delta+\left(1-\frac{\sqrt{3}}{2}\right)\delta^2 -\frac{7}{8\sqrt{3}}d^3+O(\delta^4)\right)^k=0 \tag{47}$$

and, consequently

$$\lim_{k\to\infty}C_k=\frac{c_3b_{11}+c_2b_{12}+c_1b_{13}}{b_{11}+b_{12}+b_{13}}. \tag{48}$$

[4]This expression has been obtained setting $\beta_2^0=\beta_1^0=0$ for simplicity, since the limit does not depend on the initial state.

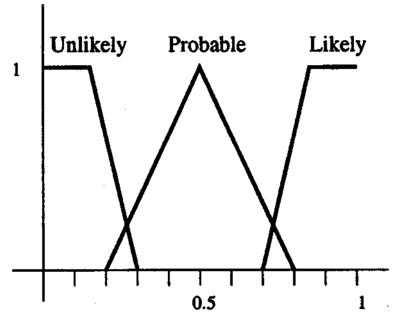


Fig. 7. Fuzzy sets of the linguistic variable likelihood.

The behavior of the system is as shown in (39). According to (40) the output fuzzy set can be written

$$H_\infty=\beta_3^\infty A_1+\beta_2^\infty A_2+\beta_1^\infty A_3 \tag{49}$$

with β_i^∞ the components of the eigenvector associated to the largest eigenvalue of the system matrix Ω.

In order to show the goodness of fit of equations (48) and (49) consider the three normal PTS fuzzy sets shown in Fig. 7; the system input is held constant an equal to the maximum fuzzy set in Fig. 7. We will consider two initial states, namely, a maximum and a minimum likelihood.[5] The evolution of the two cases [represented in dashed line in Fig. 8(a)] has been calculated by simulating a SAM system with a sum–product procedure for rule activation; the solid line in the figure shows the application of (44) for different values of k, for the two initial states. The final limiting value, which analytically turns out to be 0.5927, coincide with the two simulated cases. Convergence is achieved both in terms of centroids and in terms of the whole output fuzzy set. Fig. 8(b) shows the output fuzzy set after 300 inputs (dashed line), and the output fuzzy set calculated with (49) (in solid line). As can be seen, the match is virtually perfect.

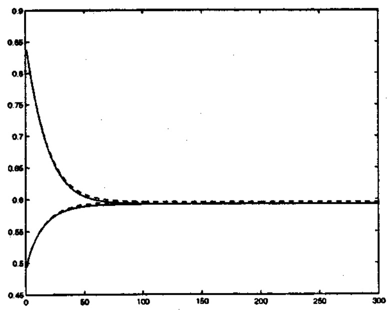
V. A COMPARISON OF STRATEGIES FOR RULE ACTIVATION

As our foregoing analysis has highlighted, the position of the output centroid depends on both the rule base and the method for rule activation. In this section, we will show that the method for rule activation plays a critical role in a FFS applied to likelihood discrimination.

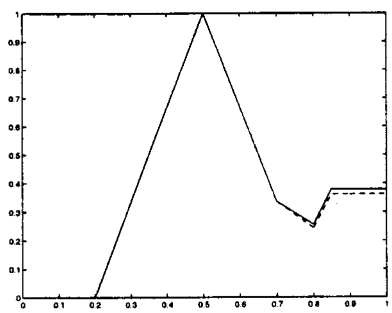
We model the linguistic variable *likelihood* as a five-valued variable, the values of which are *Very Unlikely, Unlikely, Possible, Likely, Very Likely*, were all of them are consistent and normal fuzzy sets with a PTS membership function (see Fig. 2). These sets give rise to a complete partition. The sets are ordered as *Very Unlikely* being the smallest and *Very Likely* the greatest.

[5]For this second case, (48) does not hold, but only in the limit. However, only trivial changes in the expression are needed to make it hold $\forall\, k$.

$$C_k=\frac{\lambda_1^k b'_{11}(c_3b_{11}+c_2b_{12}+c_1b_{13})+\lambda_2^k b'_{12}(c_3b_{21}+c_2b_{22}+c_1b_{23})+\lambda_3^k b'_{13}(c_3b_{31}+c_2b_{32}+c_1b_{33})}{\lambda_1^k b'_{11}(b_{11}+b_{12}+b_{13})+\lambda_2^k b'_{12}(b_{21}+b_{22}+b_{23})+\lambda_3^k b'_{13}(b_{31}+b_{32}+b_{33})}. \tag{44}$$



(a)



(b)

Fig. 8. (a) Centroid positions as a function of time. Solid line: analytical results [(44)] of C_k for the values of k shown in the horizontal axis. Dashed line: results after simulation. (b) Output fuzzy sets for the example. Solid line: (49); Dashed line: simulated results.

For simplicity, we will use the following notation: VU, U, P, L, VL. Therefore, $VU < U < P < L < VL$. We will use the set of rules shown in Table III, taken directly from [1].

As a previous example, we have entered the inputs *Very Likely* and *Possible* to the rule base, and we have made the inference process with each of the four methods for rule activation proposed in Section II-C. Fig. 9 shows the output fuzzy sets. Table IV shows the defuzzified values with the method of centroids.

The figure shows that the *sum–product* method draws the least-significant side tails off the main lobe. This is due to the fact that the product operator for the intersections makes values smaller than one decrease, while when the maximum value of the intersection is unity, its height is held. This gives rise to a fuzzy set that is very concentrated about its maximum value, and, consequently, the defuzzified value is fairly close to the maximum.

The case of the max–min operator is quite the opposite: the fuzzy output for the max–min operator is "fuzzier" (i.e., there is a greater dispersion about the maximum value of the set) and will be progressively farther from the original values of the linguistic variable. Indeed, the output set will have nonnull values in more points with this method than with the other methods, and this will produce a shift of the defuzzified output value toward the average of those values. The other two operators show intermediate behaviors.

With this previous knowledge, we will monitor the system behavior (i.e., the behavior of the third block in Fig. 1) by means of three illustrative cases whose labels and descriptions are as follows.

TABLE III
RULE SET FOR THE FUZZY MHT

$H_m^{k-1} \setminus P_m^k$	VU	U	P	L	VL
VU	VU	VU	U	U	U
U	VU	VU	U	U	U
P	U	U	P	P	L
L	U	U	P	L	VL
VL	U	U	L	VL	VL

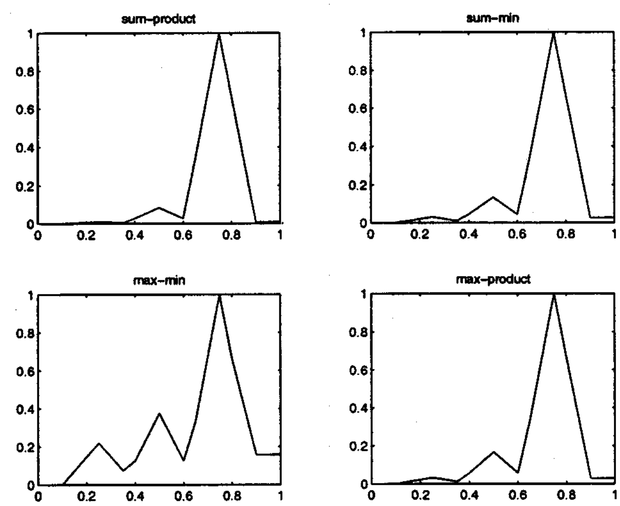


Fig. 9. Output fuzzy sets for each of the four activation methods.

TABLE IV
DEFUZZIFIED VALUES USING THE METHOD OF CENTROIDS

Method	S-P	S-M	M-M	M-P
Value	0.7279	0.7129	0.6492	0.7069

1) *Maximum:* The input will always be the maximum label of likelihood (say, the label *Very Likely*). The output centroid as a function of time will be represented with a cross (+).
2) *Maximum After Medium:* An intermediate label of likelihood (say, label *Possible*) starts the procedure, and, thereon, maximum inputs will be fed into the system. The output will be represented with a dashed line.
3) *Maximum After Minimum:* In this case, a *Very Unlikely* fuzzy set is input, and thereafter, as before, the inputs will be maximum. We will represent it with a solid line.

Output centroid values as a function of time are shown in Fig. 10 for each of the four rule activation procedures. It is clear that there is a convergence in all of the four cases, but there is a clear difference both in the limit and in the time instant at which convergence is obtained.

- For *Sum-product* [Fig. 10(b)] the limit value is 0.9, which lies in the range of the label *Very Likely*. In addition, this procedure reaches a steady state before any other. Also,

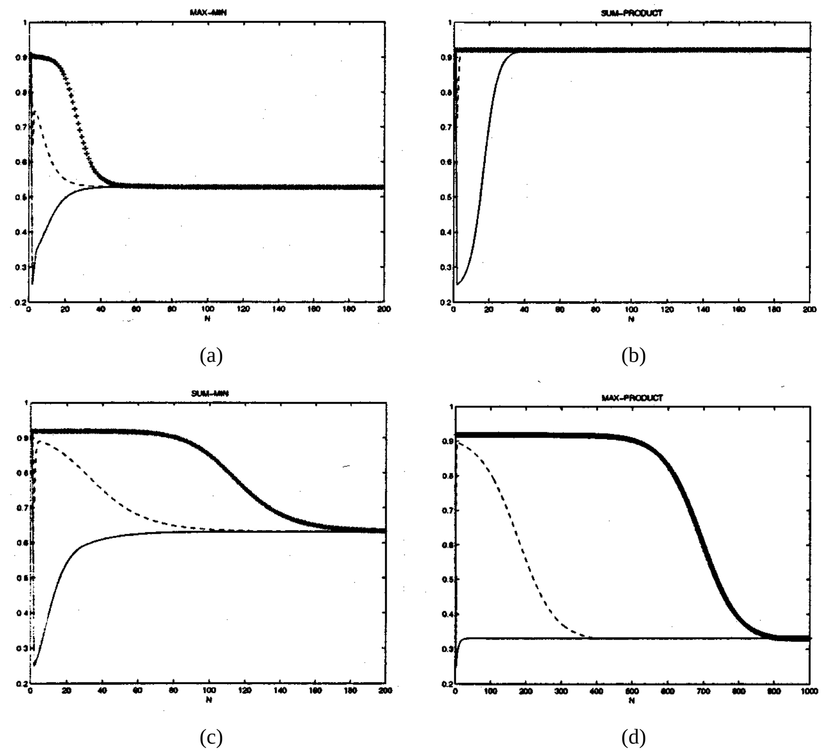


Fig. 10. Output time evolution. (a) Max–min. (b) Sum–product. (c) Sum–min. (d) Max–product.

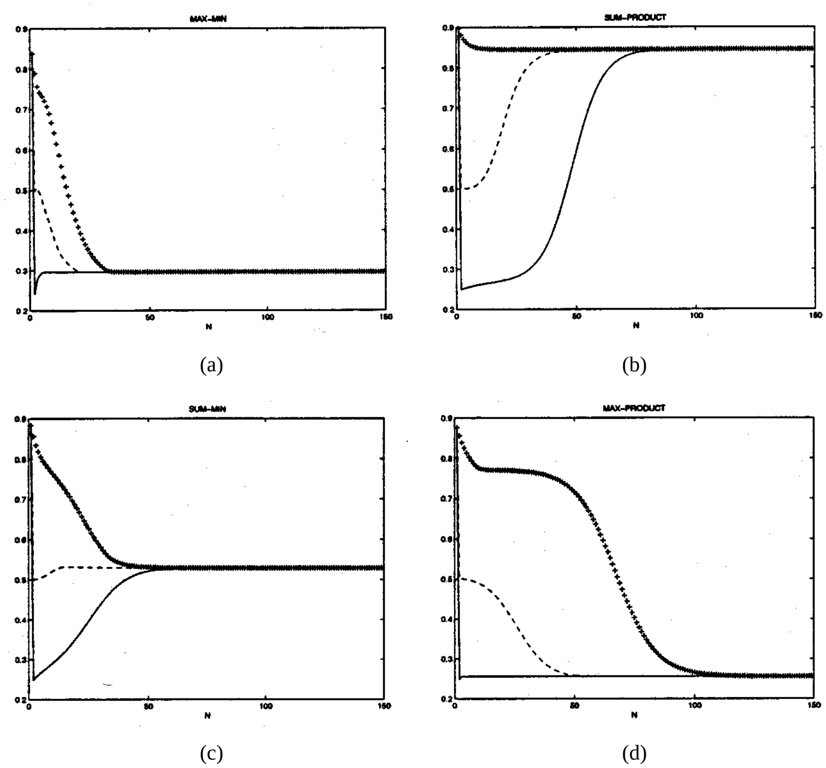


Fig. 11. Output time evolution for the second rule set. (a) Max–min. (b) Sum–product. (c) Sum–min. (d) Max–product.

the output for the constant-input case (line with crosses) is fairly constant as well.

- In the three other cases, the situation is very different: the convergence value is some intermediate value in the allowable range. This is due to the importance of the side tails out of the main lobe of the fuzzy set. In the case of the *max–min* procedure, the centroid at steady-state lies in the range of the label *Possible*. For the two other methods [Fig. 10(c) and (d)] results are far from the maximum as well.

TABLE V
STRICTER RULE SET FOR THE FUZZY MHT

$H_t \setminus P_t$	VU	U	P	L	VL
VU	VU	VU	VU	U	U
U	VU	VU	U	U	U
P	VU	U	P	P	P
L	U	U	P	L	L
VL	U	U	P	L	VL

We have repeated the aforementioned experiments for a stricter set of rules (see Table V). The outputs are shown in Fig. 11.

The results in this case confirm the former. Indeed, since rules are now stricter, i.e., unlikely labels have greater weights, side tails get higher and convergence is achieved much faster than in the previous case (note the difference in the time scale in Fig. 11). The results also show that we can calculate a dynamic range in which all the output values will be encountered. The range, as many other parameters we have described in the paper, depends on the rule base and on the activation method used.

These examples show clearly that the *sum–product* method is the only suitable method for our application. Since we are managing likelihoods, we are interested in maintaining a maximum output when the input is maximum. Because of the influence of the side tails, the max–min have a deflection to middle values when the number of iterations in the FFS is high. The result of using this operator will be a growth of the uncertainty for long tracks.

VI. CONCLUSION

In this paper, we have presented a thorough analysis of a FFS that implements a fuzzy version of the well-known MHT Reid's algorithm. An important result in the paper is the demonstration that, under very mild conditions, the fuzzy MHT algorithm will necessarily be able to discriminate the most likely hypothesis.

In addition, a second important contribution of the paper is the analytical characterization of the FFS as a function of time, and, in particular, its asymptotic behavior when the input is held constant. The importance of this result is twofold.

1) Since the asymptotic behavior is a function of the rule activation method, a designer can choose the method that matches his/her needs. In our case, in which we have dealt with probabilities, we can conclude that the *sum–product* methods clearly outperforms the others. However, different applications may prefer other methods.
2) The asymptotic behavior is also a function of the rule base itself; an analytical fine tune of the rules can be devised with our closed-form expressions. This issue is an interesting effort to be considered in the future.

As a concluding remark, we want to highlight the importance of the linear operations and the *sum–product* activation method to analytically characterize FFSs. Other types of operations make this analysis far more difficult.

REFERENCES

[1] C. Alberola and G. Cybenko, "Tracking with text-based messages," *IEEE Intell. Syst.*, vol. 14, pp. 70–78, Aug. 1999.
[2] H. R. Berenji, *Fuzzy Logic Controllers: An Introduction to Fuzzy Logic Applications and Intelligent Systems*. Norwell, MA: Kluwer, 1992.
[3] S. G. Cao and N. W. Rees, "Identification of dynamic fuzzy systems," *Fuzzy Sets Syst.*, vol. 74, pp. 307–320, 1995.
[4] Y. Y. Chen and T. C. Tsao, "A description of the dynamical behavior of fuzzy systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, pp. 745–755, July 1989.
[5] F. Chung and T. Lee, "On fuzzy associative memory with multiple-rule storage capacity," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 375–384, Aug. 1996.
[6] E. Czogala and W. Pedrycz, "Control problems in fuzzy systems," *Fuzzy Sets Syst.*, vol. 7, no. 3, pp. 257–273, 1982.
[7] M. Figueiredo, F. Gomide, A. Rocha, and R. Yager, "Comparison of Yager's level set method for fuzzy logic control with Mamdani's and Larsen's methods," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 156–159, May 1993.
[8] J. L. Goldberg, *Matrix Theory With Applications*. New York: McGraw-Hill, 1991.
[9] P. Grim, "Self-reference and chaos in fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 237–253, Aug. 1994.
[10] C. S Hsu, "A theory of cell-to-cell mapping dynamical systems," *ASME Trans. J. Dyna. Syst. Meas. Control*, vol. 47, pp. 931–939, 1980.
[11] H. Kang, "Stability and control of fuzzy dynamic systems via cell-state transitions in fuzzy hypercubes," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 267–279, Aug.. 1993.
[12] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*. Upper Saddle River, NJ: Prentice-Hall, 1995.
[13] B. Kosko, *Neural Networks and Fuzzy Systems*. Upper Saddle River, NJ: Prentice-Hall, 1992.
[14] ——, *Fuzzy Engineering*. Upper Saddle River, NJ: Prentice-Hall, 1997.
[15] C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller—Parts I–II," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 404–435, Mar. 1990.
[16] J. M. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, pp. 345–377, Mar. 1995.
[17] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Reading, MA: Addison-Wesley, 1998.
[18] W. Pedrycz, *Fuzzy Control and Fuzzy Systems*, 2nd ed. London, U.K.: Research Studies, 1993.
[19] ——, "Fuzzy dynamic systems," in *Proc. IJCAI Workshops Fuzzy Logic Fuzzy Control, LNAI*, vol. 833, Aug. 1994, pp. 37–44.
[20] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 843–854, Dec. 1979.
[21] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets Syst.*, vol. 45, pp. 135–156, 1992.
[22] R. M. Tong, "Some properties of fuzzy feedback systems," *IEEE Trans. Syst., Man., Cybern.*, vol. SMC-10, pp. 327–330, June 1980.
[23] L. Zadeh, "Fuzzy logic and its application to approximate reasoning," *Inform. Processing*, vol. 74, pp. 591–594, 1973.
[24] ——, "Fuzzy logic = computing with words," *IEEE Trans. Fuzzy Syst.*, vol. 2, pp. 103–111, Apr. 1994.
[25] X. Zeng and M. G. Singh, "Approximation theory of fuzzy systems—SISO case," *IEEE Trans. Fuzzy Syst.*, vol. 2, pp. 162–176, Apr. 1994.

Santiago Aja-Fernández received the Ingeniero de Telecomunicación degree from the University of Valladolid, Spain, in 1999. He is currently working toward the Ph.D. degree in computer engineering at the same university.

He is a Research Assistant with the ETSI Telecomunicación of the University of Valladolid, Spain, where he also collaborates in teaching activities. His research interests include fuzzy techniques for image and signal processing and he is currently involved in several medical imaging applications.

Carlos Alberola-López (S'94–M'96) received the Ingeniero de Telecomunicación and Ph.D. degrees, both from the Politechnical University of Madrid, Spain, in 1992 and 1996, respectively.

He is an Associate Professor with the ETSI Telecomunicación, University of Valladolid, Spain. In 1997, he was a Visiting Scientist at the Thayer School of Engineering, Dartmouth College, Hanover, NH. His research interests are statistical and fuzzy methods for signal and image processing applications. He is currently involved in several on-going projects in medical imaging applications and 3-D visualization.

George V. Cybenko received the B.Sc. degree in mathematics from the University of Toronto, ON, Canada, and the Ph.D. degree in mathematics from Princeton University, Princeton, NJ, in 1974 and 1978, respectively.

He is the Dorothy and Walter Gramm Professor of Engineering at Dartmouth College, Hanover, NH. Prior to joining Dartmouth, he was Professor of electrical and computer engineering at the University of Illinois at Urbana-Champaign, where he was also a director of the Center for Supercomputing Research and Development. His research interests include signal processing, distributed information systems, and computational intelligence.

Dr. Cybenko is Editor-in-Chief of IEEE TRANSACTIONS ON COMPUTING IN SCIENCE AND ENGINEERING.