A FORMAL APPROACH TO FUZZY MODELING

by

John Lygeros

Memorandum No. UCB/ERL M95/15

1 March 1995

COVER PAGE

A FORMAL APPROACH TO FUZZY MODELING

by

John Lygeros

Memorandum No. UCB/ERL M95/15

1 March 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A FORMAL APPROACH TO FUZZY MODELING

by

John Lygeros

Memorandum No. UCB/ERL M95/15

1 March 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A Formal Approach to Fuzzy Modeling *

John Lygeros

Intelligent Machines and Robotics Laboratory
University of California, Berkeley
Berkeley, CA 94720
lygeros@robotics.eecs.berkeley.edu

Abstract

A formalism for coding fuzzy models of dynamical systems is presented. It is shown that the set of models consistent with this formalism contains models that are capable of duplicating the trajectories of an arbitrary conventional discrete time dynamical system, whose single step maps are polynomials with rational coefficients. The proof of this claim is constructive. The formalism also illustrates the similarities that exist between fuzzy systems and hybrid control systems. We hope to be able to exploit this similarity by extending results from the area of hybrid systems to the fuzzy domain and vice versa.

1 Introduction

During the early seventies fuzzy logic was introduced as a way of formally describing and manipulating linguistic information ([1, 2, 3]) Soon, however, it became apparent that it could also be used for control, given a plant and a task which are both simple enough to describe linguistically (see for example [4, 5]). Moreover, the control design process does not have to make use of an explicit model of the plant; all that is needed is an idea of how the system behaves and some common sense, which gets coded in terms of fuzzy rules. The absence of a model made the designer task a lot simpler and allowed engineers to come up with satisfactory controllers with minimum effort. As a result, the field became very popular and controllers were developed for many systems, with considerable success.

Despite numerous success stories many people in the control community are still skeptical when it comes to fuzzy control. Most of the criticism probably originates from the fact that fuzzy controllers do not make use of an explicit model. One consequence of this fact is that the dynamics of the plant and the task in question have to be describable linguistically. This in turn implies that they either have to be really simple or that the fuzzy control has to be at a high level and rely on conventional controllers to do most of the job. Moreover the

*Research supported by ARO under grants DAAL03-91G-0191 and DAAL03-92G-0124

absence of a model makes it very difficult to formulate proofs which is a major drawback in systems where precise performance is needed (for example for safety).

In response to the criticism many researchers in the field of fuzzy logic decided to rethink their approach and tried to introduce mathematical modeling to it. The work presented in this paper moves roughly along these lines. Our approach is based on the belief that in order to formulate proofs for fuzzy controllers one needs to express both the plant and the controller in the fuzzy domain. This line naturally leads to the concept of fuzzy modeling. Progress in this direction has already been made, it seems, however, that most researchers are concerned with facilitating the encoding of linguistic information about the plant. Even though this fits in well with the conventional fuzzy logic methodology, we believe that, if the objective is to construct fuzzy proofs, the modeling formalism should be chosen to facilitate mathematical analysis, sacrificing some of the elegance of the linguistic description if necessary. Therefore our approach will move along the lines of general dynamical system modeling (see [6] for a quick outline). In addition the modeling formalism will be such that the closed loop system fits in the framework of hybrid systems, a field that is currently receiving a lot of attention. The idea behind this is to facilitate the extension of any results from hybrid to fuzzy systems and vice versa.

The rest of this paper is arranged in three sections. In Section 2 the modeling formalism we propose is presented. Definitions consistent with the general dynamical systems description are postulated, desirable properties that we would like the fuzzy system to satisfy are specified and the conditions under which they are satisfied are investigated. In Section 3 the descriptive power of these models is investigated. It is shown that they are almost as descriptive as a general, nonlinear, discrete time, finite state conventional model. More specifically an algorithm for obtaining a fuzzy model that approximates an arbitrary nonlinear map on a compact set arbitrarily closely is presented. In the concluding section a brief outline of the way such models can fit in the general framework of fuzzy control is discussed and directions of further research are outlined.

2 Model Formalism

As already discussed in the introduction we will try to establish a framework for carrying out modeling in terms of fuzzy sets. Our basic framework (described in Section 2.1) falls in the general framework for modeling dynamical systems described in [6]. Even though the immediate goal is to be able to use fuzzy logic to model systems in discrete time we believe that the framework can also be applied to continuous time systems with minor modifications. Some more work will be needed to carry the proofs of Section 3 over to the continuous domain, but the rest of the analysis should go through almost identically.

As with almost all modeling problems the first step is to identify all the relevant quantities whose interaction the model will specify. These relevant quantities can be classified into input, output and state variables. Let U, Y and Σ denote the input, output and state spaces respectively; in conventional models all these spaces will be subsets of $\Re^n$ for (possibly) different values of n. Because we are interested in dynamical systems, i.e. systems that evolve with time, we also need to specify a set $T \subset \Re$ of times of interest; typically $T = \Re$ or $\Re^+$ for continuous time systems and $T = \{n\tau / n \in \mathcal{Z} \text{ or } \mathcal{N}\}$ for discrete time systems. Given these sets a general dynamical system is defined in [6] as a quintuple $D = (\mathcal{U}, \Sigma, \mathcal{Y}, s, r)$ where:

- $\mathcal{U}$ is a set of *input functions*:
$$u() : T \longrightarrow U$$

- Σ is the *state space*

- $\mathcal{Y}$ is a set of *output functions*:
$$y() : T \longrightarrow Y$$

- s is the *state transition function*:
$$\begin{array}{rcl} s : T \times T \times \Sigma \times \mathcal{U} & \longrightarrow & \Sigma \\ (t_1, t_0, x_0, u()) & \longmapsto & x_1 = s(t_1, t_0, x_0, u()) \end{array}$$

  It produces the value of the state x_1 at time t_1 given the value x_0 of the state at time t_0 and the input for all times. The map is only defined for $t_1 \geq t_0$.

- r is the *read-out function*:
$$\begin{array}{rcl} r : \Sigma \times \mathcal{U} \times T & \longrightarrow & \mathcal{Y} \\ (x_t, u(t), t) & \longmapsto & y(t) = r(x_t, u(t), t) \end{array}$$

  It produces the output function at time t given the value of the state and input at time t.

In order to keep the definition consistent two axioms are imposed on the state transition function:

1. *Causality:* for all $t_0 \leq t_1$ in T, for all $x_0 \in \Sigma$ and all $u_1, u_2 \in \mathcal{U}$ such that $u_1(t) = u_2(t)$ for all $t \in [t_0, t_1] \bigcap T$
$$s(t_1, t_0, x_0, u_1) = s(t_1, t_0, x_0, u_2)$$

2. *Semigroup:* for all $t_0 \leq t_1 \leq t_2$ in T, for all $x_0 \in \Sigma$ and all $u \in \mathcal{U}$
$$s(t_2, t_0, x_0, u) = s(t_2, t_1, s(t_1, t_0, x_0, u), u)$$

2.1 Model Components

We now restrict our attention to discrete time models and in particular models whose time stamps take values in the set $T = \{n\tau / n \in \mathcal{N}\}$ for some $\tau > 0$. Without loss of generality we will assume $\tau = 1$. Let also $I = [0, 1]$ denote the unit interval in $\Re$. In accordance with the above we give the following definition:

Definition 1 *: A* **fuzzy dynamical system** *is a quintuple* $D = (\mathcal{U}^F, \Sigma^F, \mathcal{Y}^F, FR, RO)$ *where:*

- Σ^F *is the* fuzzy state space*:*
$$\Sigma^F \subset I^{a_1} \times \ldots \times I^{a_n}$$
that is every $x^F \in \Sigma^F$*:*
$$x^F = \begin{bmatrix} x_1^F \\ \vdots \\ x_n^F \end{bmatrix}$$
$$x_i^F = \begin{bmatrix} p_1^i \\ \vdots \\ p_{a_i}^i \end{bmatrix} \in I^{a_i}$$
where $0 \leq p_j^i \leq 1 \quad i \in 1,\ldots,n \quad j \in 1,\ldots,a_i$

- $\mathcal{U}^F$ *is a set of* fuzzy input functions*:*
$$u^F() : T \longrightarrow U^F \subset I^{b_1} \times \ldots \times I^{b_m}$$

- $\mathcal{Y}^F$ *is a set of* fuzzy output functions*:*
$$y^F() : T \longrightarrow Y^F \subset I^{c_1} \times \ldots \times I^{c_l}$$

- $FR = \{FR_1, \ldots, FR_n\}$ *is a set of* firing rules*:*
$$\begin{aligned} FR_i : \Sigma^F \times U^F \times T &\longrightarrow I^{a_i} \\ FR : \Sigma^F \times U^F \times T &\longrightarrow \Sigma^F \end{aligned}$$
The firing rules produce the value of the state at the next time instant given the value of the state and the input at the current time instant.

- $RO = \{RO_1, \ldots, RO_l\}$ *is a set of* read-out maps*:*
$$\begin{aligned} RO_i : \Sigma^F \times U^F \times T &\longrightarrow I^{c_i} \\ RO : \Sigma^F \times U^F \times T &\longrightarrow Y^F \end{aligned}$$
It produces the value of the output function at the current time given the value of the state and input.

Note that the firing rules are one step transition functions. We can extend them to general transition functions by repeatedly applying them. The resulting state transition function will clearly satisfy the causality and semigroup axioms.

Given the above model we can also infer extended firing rules to describe the input output behavior:

$$\begin{aligned} FRO : \Sigma^F \times U^F \times T &\longrightarrow Y^F \\ (x^F(t), u^F(t), t) &\longmapsto y^F(t+1) = RO(FR(x^F(t), u^F(t), t), u^F(t), t) \end{aligned}$$

Rules like this may be useful when doing input/output plant inversion or output feedback for the purpose of control.

For every I^{a_i} each of the a_i entries represents a fuzzy set. It is customary to assign linguistic labels to these fuzzy set that convey some characteristic of the set in question (e.g. negative, small, hot etc.). The value of the corresponding p^i_j determines to what extend the label associated with the fuzzy set j characterizes the current value of the fuzzy state. Small values of p^i_j indicates that the state component x^F_i has little to do with set j while large values indicate that the label of j is a good description of x^F_i. The same also holds for fuzzy inputs and outputs. Only a finite number of fuzzy sets are allowed for each quantity.

It should be noted that, unlike probability distributions, the fuzzy state, input and output spaces are not required to satisfy $\sum_{j=1}^{a_i} p^i_j = 1$ (or similar relations for U^F and Y^F). However fuzzy variables that satisfy this property will be particularly useful in the next section so we will give them a special name.

Definition 2 : *A fuzzy quantity is called* **normalized** *if every fuzzy vector* $[p_1 \dots p_a]^T$ *related to this quantity satisfies* $\sum_{j=1}^{a} p_j = 1$.

2.2 Interface with the "Real" World

The model described above evolves exclusively in the fuzzy domain. Describing the model in this form may be sufficient for designing fuzzy controllers for it, observing the system performance and even doing proofs. It is however desirable to be able to describe the interaction of the fuzzy model with the real world. For example external signals (e.g. reference signals that need to be tracked or disturbances that need to be rejected) as well as initial conditions for the system are usually described in terms of real numbers rather than fuzzy sets. Moreover it may be desirable to be able to connect a fuzzy system with conventional controllers and vice versa. Finally it may be desirable to be able to observe the behavior of the fuzzy system in terms of real numbers so that it is easier to quantify its performance and compare it with the performance of similar conventional systems.

2.2.1 Real to Fuzzy

The transition from the real domain to the the fuzzy domain is done via the process of *fuzzification*. This process consists of associating to each fuzzy set a membership function. These functions can be though of as maps from the real numbers to the interval $I = [0, 1]$. If there are n fuzzy sets associated with a given quantity $x \in \Re$, n such maps are defined:

$$F_i : \Re \longrightarrow I \quad i = 1, \dots, n \tag{1}$$

They determine to what extend the label associated with fuzzy set i characterizes the current value of of x. As before, small values of $F_i(x)$ indicate that the value of x has little to do with set i while large values indicate that the label of i is a good description of x.

Fuzzification consists of associating a fuzzy vector with quantity x. The fuzzy vector is obtained by passing x through all the membership functions.

$$F : \Re \longrightarrow I^n$$

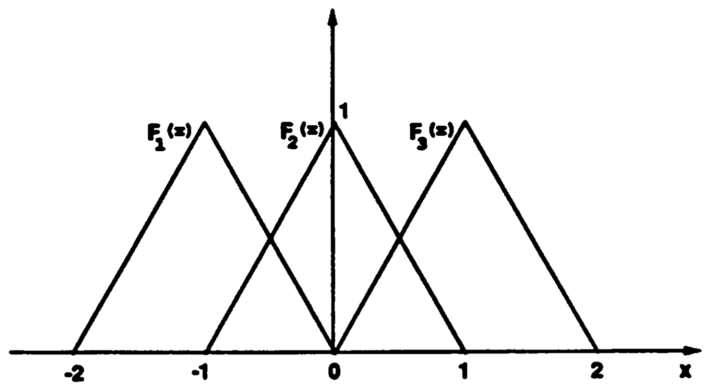

Figure 1: Triangular membership functions

$$x \longmapsto \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} F_1(x) \\ \vdots \\ F_n(x) \end{bmatrix} \tag{2}$$

Many different kinds of membership functions have been used in the literature. The most common choices are functions whose graphs are triangles (Figure 1), trapezoids or Gaussian functions. It should be noted that usually the fuzzification map F is not surjective, even though the individual membership functions are. There are many fuzzy vectors in I^n that are not the image of a real number under F. For example the range of the fuzzification process shown in Figure 1 does not contain any fuzzy vectors for which both p_1 and p_3 are non zero. Note also that typically the membership functions are not injective. For example, each triangular membership function shown in Figure 1 is two-to-one as it maps pairs of points symmetric about the center to the same value. Moreover the typical fuzzification map F will not be injective either, hence F will not be invertible even when restricted to fuzzy vectors in its range. This observation will be investigated further in Section 2.2.3.

2.2.2 Fuzzy to Real

The transition from the fuzzy domain to the real domain is done by the process of *defuzzification.* In a sense this is the inverse of the fuzzification even though mathematically speaking the maps need not be inverses of one-another (in light of the last comment of Section 2.2.1). In general defuzzification can be viewed as a map DF, mapping a fuzzy vector x^F with n fuzzy sets to a real number.

$$DF : I^n \longrightarrow \Re \tag{3}$$

Usually in the literature the defuzzification process makes explicit use of the membership functions F_i or at least their graphs and therefore is directly related to the fuzzification process. A popular choice for defuzzification is the *center of mass* technique, where the graph of the i^{th} membership function if clipped at the value of the i^{th} entry of the fuzzy vector and the center of mass of the resulting two dimensional figure is used as the real value corresponding to the fuzzy vector (Figure 2). A different defuzzification technique that does not make explicit use of the membership functions will be presented in the next section.

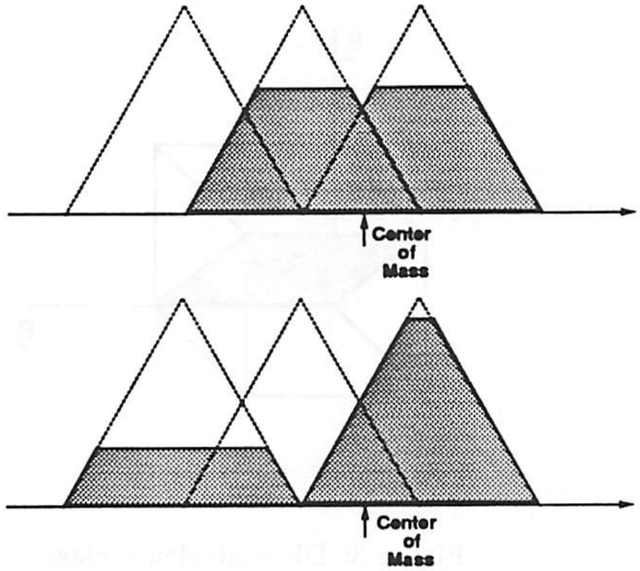

Figure 2: Center of mass defuzzification

Similarly to the fuzzification process, the map DF will typically not be surjective. Moreover, DF is usually not injective either. For example both the fuzzy vectors shown in Figure 2 will defuzzify to the same value under the center of mass defuzzification method. This observation leads us to the following definition:

Definition 3 *: Two fuzzy vectors are* **DF-equivalent** *if their images under the defuzzification map DF are equal.*

It is easy to show that DF-equivalence is indeed an equivalence relation on fuzzy vectors of a given dimension n. If the domain of the defuzzification map is the entire I^n we can say that DF partitions I^n to a disjoint union of equivalence classes. As usual we will denote the equivalence class of a fuzzy vector x^F under DF-equivalence by $[x^F]$. For example consider the defuzzification map on I^3 defined by:

$$DF(x^F) = -1p_1 + 0p_2 + 1p_3 \tag{4}$$

where $x^F = [p_1\ p_2\ p_3]^T$. Then the equivalence class corresponding to $x_o^F = [p_1^o\ p_2^o\ p_3^o]^T$ will be:

$$[x_o^F] = \{x^F \in I^3 \ / \ p_3 - p_1 = p_3^o - p_1^o\}$$

This defines a family of planes, parallel to the p_2 axis whose intersection with the (p_1, p_3) plane is a line of slope 45 degrees. The equivalence class corresponding to $p_1^o = p_3^o$ is shown in Figure 3.

Note that the defuzzification map can now be viewed as a map from the quotient space to the real numbers, by defining:

$$DF([x^F]) = DF(x^F) \tag{5}$$

In fact the map DF is injective as a map from the quotient space, therefore it is invertible on its range, that is for every $x \in \Re$ in the range of DF there exists a unique equivalence class

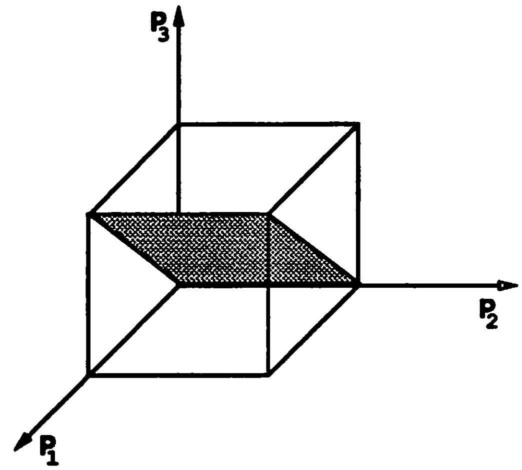

Figure 3: DF-equivalence class

$[x^F]$ whose elements are all mapped to x by DF. It would be nice to be able to associate this inversion with some form of fuzzification process. This question is addressed in the next section.

2.2.3 Properties of the Interface

The definitions given above are very general and should, in principle, encompass all the examples of fuzzy systems found in the literature (for an overview see [7] and [8]). We would, however, like to be able to say something more about fuzzy systems than we can just from the general definitions. In this section we will define a particularly interesting class of interfaces.

Definition 4 *: An interface (F, DF) is called* **consistent** *over a set $U \subset \Re$ if the map $DF \circ F$ restricted to U is the identity, i.e.:*

$$DF \circ F(x) = x \ \ \forall x \in U$$

Using this definition it is easy to show the following:

Lemma 1 *: Given a fuzzification map $F : \Re \rightarrow I^n$ and a set $U \subset \Re$ there exists a defuzzification map $DF : I^n \rightarrow \Re$ such that the pair (F, DF) is consistent over U if and only if F is injective over U.*

Proof: Assume (F, DF) is consistent and F is not injective over U. Then there exists $x_1, x_2 \in U$ with $x_1 \neq x_2$ such that $F(x_1) = F(x_2)$. Applying the defuzzification map we obtain $DF(F(x_1)) = DF(F(x_2))$ which contradicts our original assumption of consistency, as $x_1 \neq x_2$. For the converse assume F is injective over U. Then, $F : U \longrightarrow F(U) \subset I^n$ is bijective. Therefore there exists an inverse map:

$$F^{-1} : F(U) \longrightarrow U$$

Choose any $DF : I^n \rightarrow \Re$ whose restriction to $F(U)$ is the same as F^{-1}. Then:

$$\begin{aligned} DF(x^F) &= F^{-1}(x^F) \ \text{ for all } \ x^F \in F(U) \\ \Rightarrow DF \circ F(x) &= F^{-1} \circ F(x) = x \ \text{ for all } \ x \in U \end{aligned}$$

□

This lemma shows that consistency of the interface (F, DF) implies that the fuzzification map F is invertible over a certain subset of $\Re$, hence given the fuzzy vector $F(x)$ the real number x that led to it can be unambiguously determined. In other words, the interface is consistent if and only if the fuzzy vector $F(x)$ contains exactly the same information as the real number $x \in U$. The fact that no information is lost during fuzzification is in a sense undesirable as it defeats one of the main reasons behind all linguistic and abstract descriptions (like fuzzy logic) namely that information is condensed by the abstraction. Note however that a higher level of abstraction is still obtained by fuzzification, even if information is not condensed, as, after fuzzification, the emphasis is placed on the linguistic labels of the fuzzy sets rather than the real values of the membership functions. For example if a standard fuzzy controller is to be designed for such a system only the labels of the controller input fuzzy sets will be used by the firing rule base that codes the controller dynamics and determines the controller output fuzzy sets. The actual values of the membership functions will just be along for the ride.

Definition 4 also implies another nice property.

Lemma 2 *: If the interface (F, DF) is consistent over a set U then*

$$F \circ DF([x^F]) \in [x^F]$$

for all x^F such that $DF([x^F])$ belongs to U.

Proof: Assume that (F, DF) is consistent. Let:

$$\begin{aligned} DF(x^F) &= x \in U \\ \Rightarrow F \circ DF(x^F) &= F(x) \\ \Rightarrow DF \circ F \circ DF(x^F) &= DF \circ F(x) \\ \Rightarrow DF \circ F \circ DF(x^F) &= x \quad \text{by consistency} \\ \Rightarrow F \circ DF(x^F) &\in [x^F] \quad \text{by definition of DF-equivalence} \end{aligned}$$

□

This property gives us a way of consistently assigning a representative element to the equivalence class $[x^F]$, namely $F \circ DF([x^F])$. All the elements of $[x^F]$ are indistinguishable from this representative from the real output point of view. Note that the converse is not necessarily true, i.e. $F \circ DF([x^F]) \in [x^F]$ does not imply that the interface (F, DF) is consistent. For example consider the trivial defuzzification map $DF : x^F \mapsto 0$ for all $x^F \in I^n$. For this map $[x^F] = I^n$ and therefore $F \circ DF(x^F) \in [x^F]$ but $DF \circ F(x) = 0 \neq x$ in general.

By looking upon fuzzification and defuzzification as an interface between the real and fuzzy domains we are reducing the fuzzy control problem to a special case of the general *hybrid* or *intelligent control* problem. For example, coupling a dynamic (discrete time) fuzzy controller of the form described in Section 2.1 with a real plant through an interface (F, DF) like the one described above leads to the system shown in Figure 4. The block diagram indicates that the resulting closed loop system has the same structure as the systems considered in the hybrid control literature (see for example [9, 10, 11, 12]). This similarity can hopefully be exploited by extending results obtained for hybrid systems so that they can be applied to fuzzy systems and vice versa.

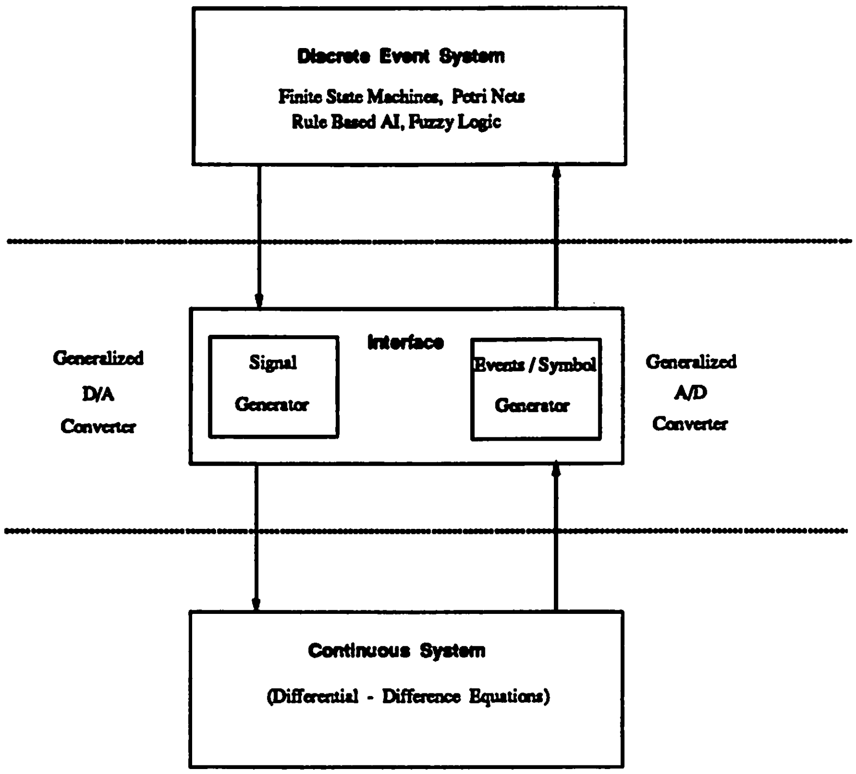

Figure 4: Fuzzy control system viewed as hybrid system

3 Descriptive Power of the Fuzzy Models

In the previous section a formalism for modeling dynamical systems in the fuzzy domain was introduced. In this section we will address the question what classes of systems can be accurately modeled using this formalism. A result in this direction can be found in [7], where it is shown that fuzzy models of sufficient generality can be used to approximate any nonlinear function on a compact subset of Euclidean space arbitrarily closely. In [7] only an existence proof is given; we will show explicitly how one can obtain fuzzy functions to carry out the approximation. Our main result can be summarized in the following theorem:

Theorem 1 *: Consider a map f on a bounded set $U \subset \Re^n$:*

$$\begin{aligned} f : U &\longrightarrow \Re \\ \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} &\longmapsto x_0 = f(x_1, \ldots, x_n) \end{aligned}$$

Assume f is a polynomial of degree N in $x_1, \ldots, x_n$ whose coefficients are rational. Then there exists a fuzzy model whose input-output map is identical to f.

An similar theorem for conventional linear dynamical systems can be found in [13].

Note: In certain cases the above claim becomes trivial. For example consider an arbitrary continuous map $f : U \rightarrow \Re$ where $U \subset \Re$ is compact. It is trivial to derive a fuzzy model that duplicates f. Indeed, let:

$$M = \max_{x \in U} f(x) \qquad m = \min_{x \in U} f(x)$$

Then, assuming $m \neq M$, we can define a fuzzy set *Trivial* with membership function:

$$F : x \longmapsto x^F = \frac{f(x) - m}{M - m}$$

a firing rule:

$$FR : x^F \in \text{Trivial} \longmapsto x^F \in \text{Trivial}$$

and a defuzzification map:

$$DF : x^F \longmapsto m + (M - m)x^F$$

Clearly this fuzzy system belongs to the class described in Section 2 and matches f on the compact set U. Unfortunately this construction does not apply to functions of more than one variable.

The proof of the theorem will be constructive. We will proceed by specifying an interface (F, DF) and a set of firing rules FR:

$$\begin{array}{rcl} F : \Re^n & \longrightarrow & I^{a_1} \times \ldots \times I^{a_n} \\ FR : I^{a_1} \times \ldots \times I^{a_n} & \longrightarrow & I^{a_0} \\ DF : I^{a_0} & \longrightarrow & \Re \end{array}$$

We will then show that the input-output map of the resulting fuzzy system:

$$DF \circ FR \circ F : \Re^n \longrightarrow \Re$$

is identical to f when restricted to U.

3.1 Fuzzification

By assumption the set $U \subset \Re^n$ is bounded, so we can find integers $a_{i_{min}}$ and $a_{i_{max}}$ for $i = 1, \ldots, n$ such that:

$$U \subset [a_{1_{min}}, a_{1_{max}}] \times \ldots \times [a_{n_{min}}, a_{n_{max}}] \tag{6}$$

Let $a_i = a_{i_{max}} - a_{i_{min}} + 1$. Then for each x_i consider a fuzzification map:

$$\begin{array}{rcl} F^i : \Re & \longrightarrow & I^{a_i} \\ x_i & \longmapsto & \begin{bmatrix} F^i_{a_{i_{min}}}(x) \\ \vdots \\ F^i_{a_{i_{max}}}(x) \end{bmatrix} \end{array} \tag{7}$$

Let α_i be the integer for which $x_i \in [\alpha_i, \alpha_i + 1)$. Then define F^i_l by:

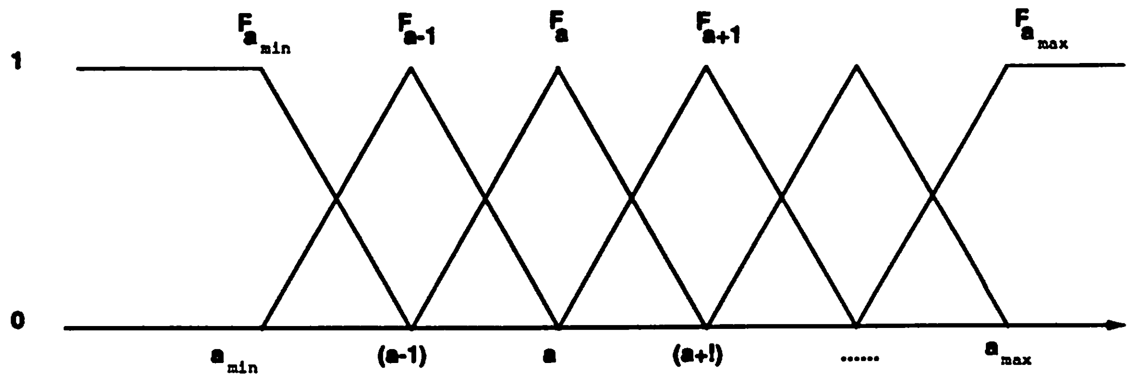

Figure 5: Membership functions

- If $[\alpha_i, \alpha_i + 1) \subset [a_{i_{min}}, a_{i_{max}}]$

$$F_l^i(x) = \begin{cases} \alpha_i + 1 - x_i & \text{if } l = \alpha_i \\ x_i - \alpha_i & \text{if } l = \alpha_i + 1 \\ 0 & \text{else} \end{cases}$$

- If $\alpha_i < a_{i_{min}}$:

$$F_l^i(x) = \begin{cases} 1 & \text{if } l = a_{i_{min}} \\ 0 & else \end{cases}$$

- If $\alpha_i + 1 > a_{i_{max}}$:

$$F_l^i(x) = \begin{cases} 1 & \text{if } l = a_{i_{max}} \\ 0 & else \end{cases}$$

The graphs of the resulting membership functions for a variable x_i are given in Figure 5. To each of the fuzzy sets defined in this way we assign a label from the set of integers $\mathcal{Z}$ in the natural way, i.e. the label $\alpha \in \mathcal{Z}$ is assigned to the fuzzy set whose membership function peaks at α. Note that the choice of fuzzy sets is such that we can carry out some algebraic calculations using their labels. This property will be crucial for the design of consistent firing rules. Note also that the fuzzy vectors produced in this way are normalized, that is the sum of all their entries is equal to 1.

To simplify the calculations we will assume that the indices of the fuzzy sets are symmetric about the origin for all the quantities considered. This can be done without loss of generality by defining $n_i = \max\{\|a_{i_{min}}\|, \|a_{i_{max}}\|\}$ and extending the fuzzification map to:

$$\begin{array}{rcl} F^i : \Re & \longrightarrow & I^{2n_i+1} \\ x_i & \longmapsto & \begin{bmatrix} F^i_{-n_i}(x) \\ \vdots \\ F^i_{n_i}(x) \end{bmatrix} \end{array}$$

by adding zeros in the extra entries. This is a rather wasteful way of storing information as many of the entries of the vector will never be used, but it will hopefully simplify the notation somewhat.

A similar fuzzy set structure can be assumed for the output x_0 of the map f. Let $x_0^F \in I^{2n_0+1}$. The problem in this case is that we can not a-priori determine the the value of n_0. We will assume for the moment that it is chosen "large enough" and we will specify how "large" this is after defining the firing rules.

3.2 Firing Rules

Let $\mathcal{I}$ be the set of normalized fuzzy vectors of arbitrary, odd dimension and let x^F and y^F be two elements of $\mathcal{I}$ of dimension $2n+1$ and $2m+1$ respectively:

$$x^F = \begin{bmatrix} p_{-n} \\ \vdots \\ p_n \end{bmatrix} \quad y^F = \begin{bmatrix} q_{-m} \\ \vdots \\ q_m \end{bmatrix}$$

Define three operations on $\mathcal{I}$, denoted $\otimes, \oplus$ and $\odot$, according to the following relations:

-
$$\begin{array}{rcl} \otimes : \mathcal{I} \times \mathcal{I} & \longrightarrow & \mathcal{I} \\ I^{2n+1} \times I^{2m+1} & \longrightarrow & I^{2mn+1} \\ (x^F, y^F) & \longmapsto & \begin{bmatrix} r_{-nm} \\ \vdots \\ r_{nm} \end{bmatrix} \end{array}$$

  where:

$$r_i = \sum_{j=-n,k=-m,jk=i}^{j=n,k=m} p_j q_k$$

-
$$\begin{array}{rcl} \oplus : \mathcal{I} \times \mathcal{I} & \longrightarrow & \mathcal{I} \\ I^{2n+1} \times I^{2m+1} & \longrightarrow & I^{2(n+m)+1} \\ (x^F, y^F) & \longmapsto & \begin{bmatrix} r_{-(n+m)} \\ \vdots \\ r_{(n+m)} \end{bmatrix} \end{array}$$

  where:

$$r_i = \sum_{j=-n,k=-m,j+k=i}^{j=n,k=m} p_j q_k$$

-
$$\begin{array}{rcl} \odot : \mathcal{Z} \times \mathcal{I} & \longrightarrow & \mathcal{I} \\ a \times I^{2n+1} & \longrightarrow & I^{2(an)+1} \\ (a, x^F) & \longmapsto & \begin{bmatrix} r_{-(an)} \\ \vdots \\ r_{an} \end{bmatrix} \end{array}$$

where:

$$r_i = \begin{cases} p_j & \text{if } i = aj \\ 0 & \text{else} \end{cases}$$

Before we go any further we need to verify is that the operations defined in this way are indeed meaningful as maps on $\mathcal{I}$. Clearly the output of every one of these operations is a fuzzy vector of odd dimension so the only thing that needs to be shown is that they are also normalized. Indeed if we add the entries of $x^F \otimes y^F$ we obtain:

$$\begin{aligned}
\sum_{i=-nm}^{nm} r_i &= \sum_{i=-nm}^{nm} \sum_{j=-n,k=-m,jk=i}^{j=n,k=m} p_j q_k \\
&= \sum_{j=-n,k=-m}^{j=n,k=m} p_j q_k \\
&= \sum_{j=-n}^{j=n} p_j \sum_{k=-m}^{k=m} q_k \\
&= 1
\end{aligned}$$

as x^F and y^F are both normalized. The same calculation goes through for the other two operations as well.

A simple calculation reveals that, under the three operations $\otimes, \oplus$ and $\odot$, the set $\mathcal{I}$ becomes a $\mathcal{Z}$ algebra, not unlike polynomials with integer coefficients. This observation allows us to define firing rules that imitate any polynomial map with integer coefficients in a natural way. Let $f : \Re^n \rightarrow \Re$ be such a polynomial. The firing rules that would imitate the behavior of f can be derived by substituting all variables by their fuzzy counterparts, all multiplications by $\otimes$, all additions by $\oplus$ and all scalar multiplications by $\odot$. For example consider:

$$f(x, y) = a_0 + a_1 y + a_2 x^2 + a_3 xy$$

where $a_1, \ldots a_3$ are integers. Then the firing rules FR will be of the form:

$$FR(x^F, y^F) = a_0 \odot 1^F \oplus a_1 \odot y^F \oplus a_2 \odot x^F \otimes x^F \oplus a_3 \odot x^F \otimes y^F$$

where 1^F is a normalized fuzzy 3 vector with a 1 at the entry corresponding to 1 and zeros everywhere else. Note that in order to code the output of f in this example we would need $2n' + 1$ fuzzy sets, where, if x^F is of dimension $2n + 1$ and y^F is of dimension $2m + 1$:

$$n' = a_0 + a_1 m + a_2 n^2 + a_3 nm$$

This calculation can be carried out for a general polynomial as well. The number n' of fuzzy sets required to code the output can always be obtained by substituting the corresponding values of n, m, etc. in the polynomial and computing. Typically n' will be large, but it will always be finite.

3.3 Defuzzification

Having specified a fuzzification map and a set of firing rules that, in a sense, duplicate a polynomial map f, we now have to define a defuzzification map. Consider the following map on normalized fuzzy vectors of odd dimension:

$$\begin{aligned} DF : I^{2n+1} &\longrightarrow \Re \\ \begin{bmatrix} p_{-n} \\ \vdots \\ p_n \end{bmatrix} &\longmapsto \sum_{i=-n}^{n} ip_i \end{aligned} \tag{8}$$

In probability terms this could be called the expectation of the fuzzy vector. Defining the defuzzification process in this way leads to two interesting properties:

Lemma 3 : *The defuzzification map DF is such that the interface (F, DF) is consistent over the set of interest U.*

Proof: for every $x \in U$ and $x \in [\alpha, \alpha + 1)$ for an integer α:

$$DF(F(x)) = \alpha(\alpha + 1 - x) + (\alpha + 1)(x - \alpha) = x$$

□

Lemma 4 : *The maps $\otimes, \oplus, \odot$ on $\mathcal{I}$ are well defined on the quotient space induced by DF-equivalence.*

Proof: It is easy to see that the Lemma holds for scalar multiplication $\odot$, so we will focus our attention to the other two operations. Consider 4 fuzzy vectors in $\mathcal{I}$:

$$x_1^F = \begin{bmatrix} p_{-n_1} \\ \vdots \\ p_{n_1} \end{bmatrix} \quad x_2^F = \begin{bmatrix} q_{-n_2} \\ \vdots \\ q_{n_2} \end{bmatrix}$$
$$y_1^F = \begin{bmatrix} r_{-m_1} \\ \vdots \\ r_{m_1} \end{bmatrix} \quad y_2^F = \begin{bmatrix} s_{-m_2} \\ \vdots \\ s_{m_2} \end{bmatrix}$$

Assume that $x_2^F \in [x_1^F]$ and $y_2^F \in [y_1^F]$, i.e.:

$$\begin{aligned} \sum_{-n_1}^{n_1} ip_i &= \sum_{-n_2}^{n_2} iq_i \\ \sum_{-m_1}^{m_1} ir_i &= \sum_{-m_2}^{m_2} is_i \end{aligned}$$

We need to show that $x_1^F \otimes y_1^F$ and $x_2^F \otimes y_2^F$ belong to the same DF-equivalence class (and similarly for $\oplus$).

$$\begin{aligned}
DF(x_1^F \otimes y_1^F) &= \sum_{i=-n_1m_1}^{n_1m_1} i(x_1^F \otimes y_1^F)_i \\
&= \sum_{i=-n_1m_1}^{n_1m_1} i \sum_{kj=i} p_j r_k \\
&= \sum_{i=-n_1m_1}^{n_1m_1} \sum_{kj=i} i p_j r_k \\
&= \sum_{i=-n_1m_1}^{n_1m_1} \sum_{kj=i} (jp_j)(kr_k) \\
&= \sum_{k,j} (jp_j)(kr_k) \\
&= \sum_{j=-n_1}^{n_1} (jp_j) \sum_{k=-m_1}^{m_1} (kr_k) \\
&= \sum_{j=-n_1}^{n_2} (jq_j) \sum_{k=-m_2}^{m_2} (ks_k) \\
&= DF(x_2^F \otimes y_2^F)
\end{aligned}$$

Similarly for $\oplus$:

$$\begin{aligned}
DF(x_1^F \oplus y_1^F) &= \sum_{i=-n_1+m_1}^{n_1+m_1} i(x_1^F \oplus y_1^F)_i \\
&= \sum_{i=-n_1+m_1}^{n_1+m_1} i \sum_{k+j=i} p_j r_k \\
&= \sum_{j,k} (j+k) p_j r_k \\
&= \sum_{j,k} j p_j r_k + \sum_{j,k} k p_j r_k \\
&= \sum_j j p_j \sum_k r_k + \sum_j p_j \sum_k k r_k \\
&= \sum_j j p_j + \sum_k k r_k \\
&= DF(x_2^F \oplus y_2^F)
\end{aligned}$$

$\square$

Note that this procedure can easily be generalized to polynomials whose coefficients are rational rather than integer. Let $\hat{f}$ be such a polynomial and D be the least common multiple of the denominators of all the coefficients. Then $f = D\hat{f}$ is a polynomial with integer coefficients, for which the above construction (firing rules and defuzzification) is applicable.

To reproduce $\hat{f}$ from this we need only alter the defuzzification map to $\hat{DF} = \frac{1}{D}DF$. This augmentation leaves the dynamics (coded in the firing rules) unaffected. Moreover the equivalence classes of $\hat{DF}$ are the same as the ones of DF, therefore Lemma 4 still holds. However the new interface $(F, \hat{DF})$ is no longer consistent.

3.4 Proof of Theorem

With all the elements of the fuzzy model specified we are now ready to prove the main theorem of this section. As above, let $\hat{f} : U \longrightarrow \Re$ be a polynomial map with rational coefficients whose least common multiple is D. In the previous sections the procedure for defining a fuzzification process F, a set of firing rules FR and a defuzzification process $\hat{DF}$ based on $\hat{f}$ was outlined. In this section our objective is to show that the input- output map of the resulting fuzzy system is the same as $\hat{f}$.

As the only effect of the rational coefficients is during defuzzification we will first prove the claim for a polynomial with integer coefficients f and then trivially generalize it to the rational coefficient case. Restating the theorem we would like to prove that the following diagram commutes:

$$\begin{array}{ccc} U & \xrightarrow{f} & \Re \\ F\downarrow & & \uparrow DF \\ \mathcal{I}^N & \xrightarrow[FR]{} & \mathcal{I} \end{array} \tag{9}$$

We will proceed by first proving that the corresponding diagrams for the three operations $\otimes, \oplus, \odot$ and their counterparts for integers (i.e. addition and multiplication) do commute. In other words we would like to show that that given any pair of real numbers, x and y (in a certain range), then:

$$xy = DF(F(x) \otimes F(y)) \tag{10}$$

$$x + y = DF(F(x) \oplus F(y)) \tag{11}$$

$$ax = DF(F(x) \odot F(y)) \tag{12}$$

Assuming $x \in [\alpha, \alpha+1)$ and $y \in [\beta, \beta+1)$ for α, β integers:

$$F_l(x) = \begin{cases} \alpha+1-x & \text{if } l = \alpha \\ x-\alpha & \text{if } l = \alpha+1 \\ 0 & \text{else} \end{cases}$$

$$F_l(y) = \begin{cases} \beta+1-y & \text{if } l = \beta \\ y-\beta & \text{if } l = \beta+1 \\ 0 & \text{else} \end{cases}$$

If the product of these two fuzzy vectors is taken using the $\otimes$ operator the resulting fuzzy vector will have zeros in all the entries except:

$$\begin{aligned} \alpha\beta &\rightsquigarrow (\alpha+1-x)(\beta+1-y) \\ \alpha(\beta+1) &\rightsquigarrow (\alpha+1-x)(y-\beta) \\ (\alpha+1)\beta &\rightsquigarrow (x-\alpha)(\beta+1-y) \\ (\alpha+1)(\beta+1) &\rightsquigarrow (x-\alpha)(y-\beta) \end{aligned}$$

Passing this through the defuzzification map we obtain, after some algebra:

$$\begin{aligned}DF(F(x)\otimes F(y)) &= \alpha\beta(\alpha+1-x)(\beta+1-y)+\alpha(\beta+1)(\alpha+1-x)(y-\beta)\\&\quad+(\alpha+1)\beta(x-\alpha)(\beta+1-y)+(\alpha+1)(\beta+1)(x-\alpha)(y-\beta)\\&= xy\end{aligned}$$

Similarly, the fuzzy vector obtained for $F(x)\oplus F(y)$ will have the following non zero entries:

$$\begin{aligned}\alpha+\beta &\rightsquigarrow (\alpha+1-x)(\beta+1-y)\\ \alpha+(\beta+1) &\rightsquigarrow (\alpha+1-x)(y-\beta)\\ (\alpha+1)+\beta &\rightsquigarrow (x-\alpha)(\beta+1-y)\\ (\alpha+1)+(\beta+1) &\rightsquigarrow (x-\alpha)(y-\beta)\end{aligned}$$

Thus the real number obtained after defuzzification will be:

$$\begin{aligned}DF(F(x)\oplus F(y)) &= (\alpha+\beta)(\alpha+1-x)(\beta+1-y)+(\alpha+(\beta+1))(\alpha+1-x)(y-\beta)\\&\quad+((\alpha+1)+\beta)(x-\alpha)(\beta+1-y)+((\alpha+1)+(\beta+1))(x-\alpha)(y-\beta)\\&= x+y\end{aligned}$$

Finally, the fuzzy vector obtained from $a\odot F(x)$ for any integer a will have non zero entries at:

$$\begin{aligned}a\alpha &\rightsquigarrow \alpha+1-x\\ a(\alpha+1) &\rightsquigarrow x-\alpha\end{aligned}$$

After defuzzification:

$$\begin{aligned}DF(a\odot F(x)) &= a\alpha(\alpha+1-x)+a(\alpha+1)(x-\alpha)\\&= ax\end{aligned}$$

Overall we have shown that equations 10, 11 and 12 are indeed valid. Recall however that Lemmas 2 and 3 guarantees that every equivalence class $[x^F]$ has exactly one element corresponding to the fuzzified value of a real number, namely $F(DF([x^F]))$. Further, Lemma 4 guarantees that, as long as we plan to defuzzify the final result, whatever holds for this particular element holds for the whole equivalence class. Combining these three facts concludes the proof of the theorem for polynomials with integer coefficients. Dividing the defuzzification map with the least common multiples of the denominators of all the coefficients (as described in the previous section) extends this result to polynomials with rational coefficients.

□

A direct corollary of the above theorem is the following:

Corollary 1 *: Given any continuous function f on a compact subset $U \subset \Re^n$ there exists a fuzzy model that approximates it arbitrarily closely.*

Here "arbitrarily closely" means that given any $\epsilon > 0$ we can find a fuzzy approximation to f such that the supremum as the arguments vary over U of the absolute value of the difference between f and its fuzzy approximation is less than ϵ. To prove Corollary 1 we just need to note that, as a consequence of the Stone-Weierstrass theorem, we can approximate the map f arbitrarily closely by a polynomial with rational coefficients. Then we can duplicate this polynomial by a fuzzy system as described by Theorem 1. This corollary is known and its proof can also be found in [7]. The main difference between our approach and the one in [7] is that our proof is constructive.

3.5 Dynamical Systems

Besides static maps, Theorem 1 can easily be extended to discrete time dynamical systems, as a discrete time system can be characterized by a single step map that maps the current state and input to the state at the next sample time.

Corollary 2 : *Given any discrete time dynamical system whose single step map is a polynomial with rational coefficients, there exists a fuzzy model that duplicates its behavior, provided that the state and input trajectories are guaranteed to lie in a compact set.*

The last assumption is somewhat restrictive, as it forces us to exclude all unstable plants, among other things. In certain cases however it is justified. In many systems, for example, there are saturation bounds on the actuators and the state that enforce this assumption. Moreover, the purpose of control is usually to stabilize the system, in which case the fuzzy model can adequately describe the plant once the loop is closed.

In view of Corollary 1 and 2 it seems plausible that we could hope to construct a fuzzy model that reproduces the trajectories of an arbitrary, discrete time dynamical system arbitrarily closely. Unfortunately this assertion is probably not true. We can only prove it for a very restricted class of systems:

Corollary 3 : *Given an autonomous discrete time dynamical system whose one step map is contracting there exists a fuzzy system that tracks all trajectories that lie in a compact set arbitrarily closely.*

Proof: Let $f : \Re^n \rightarrow \Re^n$ be the single step map of the system in question and let $\hat{f} : U \rightarrow U$ be a fuzzy approximation such that:

$$\|f(x) - \hat{f}(x)\| \leq \epsilon \quad \forall x \in U \subset \Re^n$$

where U is the compact set were the trajectories of f lie. $\|.\|$ is any complete norm on $\Re^n$. f is a contraction over U if there exists $\rho < 1$ such that:

$$\|f(x) - f(y)\| \leq \rho\|x - y\| \quad \forall x, y \in U$$

Let $\{x_k \in U/k \in \mathcal{N}\}$ and $\{\hat{x}_k \in U/k \in \mathcal{N}\}$ be the trajectories with initial condition $x^0 \in U$ of the conventional dynamical system and the fuzzy approximation respectively, i.e.:

$$\begin{aligned} x_0 = \hat{x}_0 &= x^0 \\ x_{k+1} &= f(x_k) \\ \hat{x}_{k+1} &= \hat{f}(x_k) \end{aligned}$$

Then:

$$\|x_1 - \hat{x}_1\| = \|f(x_0) - \hat{f}(\hat{x}_0)\| = \|f(x^0) - \hat{f}(x^0)\| \leq \epsilon \tag{13}$$

Proceed by induction. Assume that, for some $n \geq 1$:

$$\|x_n - \hat{x}_n\| \leq \epsilon \sum_{i=0}^{n-1} \rho^i \tag{14}$$

Then:

$$\begin{aligned}
\|x_{n+1} - \hat{x}_{n+1}\| &= \|f(x_n) - \hat{f}(\hat{x}_n)\| \\
&= \|f(x_n) - f(\hat{x}_n) + f(\hat{x}_n) - \hat{f}(\hat{x}_n)\| \\
&\leq \|f(x_n) - f(\hat{x}_n)\| + \|f(\hat{x}_n) - \hat{f}(\hat{x}_n)\| \\
&\leq \rho\|x_n - \hat{x}_n\| + \epsilon \\
&\leq \rho\epsilon \sum_{i=0}^{n-1} \rho^i + \epsilon \\
&\leq \epsilon \sum_{i=0}^{n} \rho^i
\end{aligned}$$

Combining this last statement with equations 13 and 14 we conclude, by induction, that:

$$\begin{aligned}
\|x_k - \hat{x}_k\| &\leq \epsilon \sum_{i=0}^{k-1} \rho^i \ \ \forall k \in \mathcal{N} \\
&\leq \epsilon \sum_{i=0}^{\infty} \rho^i \\
&= \frac{\epsilon}{1-\rho}
\end{aligned}$$

Thus if we want to construct a fuzzy system to track the original to within δ we have to choose a fuzzy map $\hat{f}$ that approximates f to within $\epsilon = (1-\rho)\delta$.

□

4 Concluding remarks

The discussion presented above can be viewed as an attempt to link fuzzy logic to the well established field of dynamical systems and the rapidly evolving field of hybrid control. In the process links to probability were also discovered. For example the fuzzy construction that proved useful in formulating the theorem is very similar to a set of probability distributions. Note that some of the results proved here are trivial when looked at from the probability point of view; they are merely a restatement of well known facts such as that the expectation of the product of two independent random variables is equal to the product of their expectations.

Another interesting fact about this formalism, that relates more to hybrid systems, is that it provides a technique for coding dynamics in a semi-discrete way. Because of the injectivity of the fuzzification maps no information is lost when moving from the "real" to the fuzzy domain. However, once in the fuzzy domain, attention is restricted to the fuzzy set labels, rather than the exact values of the membership functions. So, even though the fuzzy world is effectively discrete, the continuous information is not lost, as is the case with most standard abstractions.

Overall the above analysis indicates that fuzzy models are probably a rich enough set to be useful. Moreover, the introduction of a rigorous technique for obtaining firing rules and interfacing to the real world gives us hope that proofs may be in sight for fuzzy logic controllers designed in this framework.

Acknowledgment: The author would like to thank Charles Coleman, Datta Godbole and Prof. Shankar Sastry for their helpful comments and encouragement

References

[1] L. Zadeh, "Fuzzy sets," *Informat. Control*, vol. 8, pp. 338–353, 1965.

[2] L. Zadeh, "Toward a theory of fuzzy systems," in *Aspects of Network and System Theory* (R. Kalman and N. DeClaris, eds.), Holt, Rinehart and Winston, 1971.

[3] L. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-3, no. No 1, pp. 28–44, 1973.

[4] E. Mamdani and S. Assilian, "An experiment in linguistic sysntesis with fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. No 1, pp. 1–13, 1975.

[5] R. Yager and L. Zadeh, eds., *An Introduction to Fuzzy Logic Applications in Intelligent Systems.* Boston, MA, USA: Kluwer Academic Publishers, 1992.

[6] F. M. Callier and C. A. Desoer, *Linear System Theory.* Springer-Verlag, 1991.

[7] L.-X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis.* Prenice Hall, 1994.

[8] D. Driankov, *An Introduction to Fuzzy Control.* Berlin, DGR: Springer-Verlag, 1993.

[9] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, eds., *Hybrid Systems.* Springer-Verlag, 1993.

[10] R. W. Brockett, "Hybrid models for motion control systems," in *European Control Conference*, 1993.

[11] K. M. Passino and P. J. Antsaklis, "Modeling and analysis of artificially intelligent planning systems," in *An Introduction to Intelligent and Autonomous Control* (P. J. Antsaklis and K. M. Passino, eds.), pp. 191–214, Boston: Kluwer Academic Publishing, 1993.

[12] A. Nerode and W. Kohn, "Models for hybrid systems: Automata, topologies, controllability, observability," in *Hybrid System* (R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, eds.), pp. 317–356, New York: Springer Verlag, 1993.

[13] J. Lygeros, D. N. Godbole, and C. P. Coleman, "Model based fuzzy logic control," Tech. Rep. UCB/ERL-M94/60, Electronic Research Laboratory, University of California Berkeley, 1994. Also in Proceedings of the 1994 Conference on Decision and Control.