functions, neural networks, optimal interfaces, optimization, semantic importance of the proposed constraints are included. Implementation issues and one example illustrating the constraints that are coded within a given optimization scheme, such as and discussed. These properties are translated in terms of nonlinear semantic properties that membership functions should have is postulated interpreting membership functions. To achieve this a comprehensive set of when used within an optimization scheme obviate the subjective task of This work addresses this difficulty and points out a set of constraints that optimized in such a way cannot provide a linguistic interpretation, techniques, is becoming more and more popular. In general, fuzzy systems strategies, such as neural network learning rules or evolutionary optimization which is comparable to let the fuzzy systems (such as controllers, models, or classifiers) using biological inspired strategies is that the only indicator guiding the search process is concerned exclusively with the system quality in terms of the approximation task (e.g., with the errors between system output and training data). For example, this is case with the so-called performance index of the neural networks learning rules [7] and with the fitness function of the genetic algorithms [5]. In the presence of such unconstrained optimization processes the parameters of membership functions (e.g., centers and widths) are treated just like any other parameter of the system. Furthermore, usually a fuzzy system has very many independent parameters (degrees of freedom). As a consequence, several local minima of the performance function are likely to exist. In face of this we simply cannot ensure that the resulting membership functions have any semantic value in what concerns their ability to be interpretable as linguistic terms. To ignore this aspect during fuzzy system design is first of all to renounce the richness of concepts and techniques provided by fuzzy sets and systems. It is comparable to let the fuzzy parameters outgoing their range (usually [0, 1]) violating the set or logic nature of fuzzy calculus. It is equivalent to consider fuzzy systems as a tool for initializing artificial neural networks (in the sense that without semantics a fuzzy system can be viewed as a neural net). While this is undeniable, one may legitimately ask 1) whether constraining the optimization to preserve semantics would not degrade the fuzzy system performance, and 2) what would be the reasons to translate the functioning of, e.g., a ready-to-implement fuzzy control algorithm into a natural-language equivalent description. For addressing the first issue, it is recalled that, in general, unconstrained optimization methods are more susceptible to get stuck in local minima than conveniently constrained ones. One the other hand, minima are necessarily local in that region of parameters space where there is no possible interpretation. A typical case being the absence of membership functions in a subset of the universe of discourse. This being so, there would be input values for which the system would have no response. Constraints may help to avoid local minima. The systematic avoidance of local minima leads to the global minimum, and thus to optimal performance. This contradicts the naive belief that constrained optimization cannot perform better than unconstrained optimization. Take adaptive control as an example. In certain model-reference schemes unconstrained optimization of the controller gains leads to unstable closed-loop responses. However when equipped with convenient constraints, these control schemes leads to guaranteed stability, cf. [1]. Later on, this semantics/performance synergy is further illustrated by means of an included example where a fuzzy system optimized with semantic constraints performs better than other with the same structure and initial conditions, but optimized without taking care of semantics. Notice also that the first applied fuzzy (control) systems were designed using heuristic methods based on the knowledge and experience of experts, and thus they were necessarily semantically valid. However this has not degraded their performances; by the opposite in

Semantic Constraints for Membership Function Optimization

J. Valente de Oliveira

Abstract—The optimization of fuzzy systems using bio-inspired strategies, such as neural network learning rules or evolutionary optimization techniques, is becoming more and more popular. In general, fuzzy systems optimized in such a way cannot provide a linguistic interpretation, preventing us from using one of their most interesting and useful features. This work addresses this difficulty and points out a set of constraints that when used within an optimization scheme obviate the subjective task of interpreting membership functions. To achieve this a comprehensive set of semantic properties that membership functions should have is postulated and discussed. These properties are translated in terms of nonlinear constraints that are coded within a given optimization scheme, such as backpropagation. Implementation issues and one example illustrating the importance of the proposed constraints are included.

Index Terms—Adaptive systems, fuzzy systems, learning, membership functions, neural networks, optimal interfaces, optimization, semantic constraints.

I. INTRODUCTION

When compared with other nonlinear modeling techniques such as artificial neural networks, Hammerstein, Wiener, or more generally NARMAX models (cf. [3]), fuzzy systems have the important ad

Manuscript received April 15, 1996; revised February 10, 1997 and April 10, 1998.

The author is with the Department of Mathematics and Computer Science, Universidade Da Beira Interior, 6200 Covilhã, Portugal.

Publisher Item Identifier S 1083-4427(99)00895-4.

1083-4427/99$10.00 © 1999 IEEE
many cases better performances were obtained when compared with the other available solutions, cf. [14].

For addressing the second issue, we can consider an industrial plant where a new variable was made observable to improve the performance of an existent fuzzy controller. In this case, the maintenance of the controller will be easier to accomplish by someone who finds a set of legible if–then rules instead of a set of meaningless parameters such as those found in neural networks. While it is very difficult to include additional functionality in a trained neural network, it is relatively easy to add, e.g., a new production rule into a trained legible fuzzy system. On the other hand, for those fuzzy systems dealing with user interfaces using a natural language, semantics is a crucial requirement. The fuzzy I/O controller is one of such systems [14].

This work is devoted to the study of semantically driven conditions to constraint the optimization process of a fuzzy system in such a way that the resulting membership functions can still represent human readable linguistic terms. In Section II, a comprehensive set of semantic properties membership functions should meet is postulated and discussed. In Section III, constraints are introduced and analyzed. Section IV discusses some implementation issues, and presents one example illustrating the importance of the proposed constraints. A section on conclusions ends the paper.

II. ON MEMBERSHIP FUNCTIONS AND LINGUISTIC TERMS

Conceptually the architecture of a fuzzy system involves three different stages: the so-called input interface, the fuzzy processing stage, and the output interface.

An interface deals with linguistic variables. The notion of linguistic variable formally requires a semantic rule for associating a meaning to each one of its values, the linguistic terms represented by the referential or primary fuzzy sets [20]. Hence, each referential fuzzy set should represent a linguistic term with a clear semantic meaning. If the system processes the linguistic variable temperature, then its values might be “high,” “medium,” and “low” temperature, for instance.

In this section, an attempt is made to postulate a set of properties that the membership functions of the referential fuzzy sets should have to obviate the subjective task of assigning linguistic terms to them. The focus is on the meaning of a linguistic term within its context i.e. relatively to the other linguistic terms, instead of on the absolute meaning of an isolated linguistic term. The set of semantic properties includes, cf. [16]. A moderate number of membership functions, distinguishability, normality, natural zero positioning, and coverage.

A. A Moderate Number of Membership Functions

The number of membership functions should not be arbitrary. Although this number has been recognized as application dependent, there are some reasons for imposing an upper bound on the number of membership functions.

The main reason pertains to the Zadeh’s principle of incompatibility [19]. As soon as the number of membership functions increases the precision of the system may increase too, but its relevance will decrease. In the limit, when the number of membership functions approaches the number of data, a fuzzy system becomes a numeric system. What would be then an appropriated limit?

A well-known and interesting result from Cognitive Psychology is that the typical number of different entities efficiently handled at the short-term memory (also known as active, or working memory) is 7 ± 2 [9]. See also [9]. This observation on the human information process limitations has guided most of the strategies for segmenting, factoring, or decomposing problems in Computer Sciences and other fields. The problem is that above this number the errors generated by us while juggling, dealing with, or handling objects, entities, or concepts rise in a strong nonlinear way. Thus we can and should adopt this typical number as an upper-bound on the number of membership functions. Curiously enough, it is interesting to verify that most of the successful industrial and commercial applications of fuzzy systems do not exceed this limit, cf. [14].

B. Distinguishability

By definition of linguistic variable, there should be a semantic rule associating a meaning to each value of the variable. That is, a membership function should represent a linguistic term with a clear semantic meaning. To obviate the subjective establishment of this membership functions/linguistic term association, membership functions of a given linguistic variable should be distinct enough from each other.

In the absence of this property, a variable could have equal membership functions what would be superfluous, tautological, and misleading. The existence of close or nearly identical membership functions (linguistic terms) may cause inconsistencies at the processing stage. As an illustration, consider, e.g., the following fuzzy rules: IF A THEN B and IF A’ THEN C, where A, A’, B, and C are linguistic terms. If A and A’ are nearly identical then an inconsistency is likely to happen, since for the same input two distinct conclusions, induced by B and C, are derived.

If required, fine shades of meaning can be realized at the processing stage using logical connectives (such as conjunction and negation) and hedges (such as very) on the linguistic terms [19], [20].

C. Normality

Membership functions should be normal. Since each membership function should represent a linguistic term with a clear semantic interpretation, then at least one of the datum in the universe of discourse should exhibit full matching with each membership function. More formally

\[ \exists v \in V \forall i = 1, 2, \ldots, n : \mu_i(v) = 1.0 \] (1)

where v is any external variable defined in V, n is the number of membership functions and \( \mu_i \) is the i-th membership function.

Semantically consider nonnormal membership functions is to question the meaning of the represented linguistic terms. On the other hand, normality allows one to use the entire range of membership grades, usually [0, 1].

D. Natural Zero Positioning

If required by the nature of the problem, one of the membership functions should be unimodal, convex, and centered in zero, representing the linguistic term nearly zero. This applies to most fuzzy controllers essentially because a nearly zero tracking error (exactly zero, if possible) is desirable. It should be emphasized the difference between the location of the linguistic term nearly zero, and its necessity for solving a given problem.

E. Coverage

Membership functions \( \mu_i \) of the referential sets \( \mathcal{X}_i \) should cover the entire universe of discourse, so that every piece of data may have a linguistic representation. Formally

\[ \forall v \in V \exists i : 1 \leq i \leq n \mu_i(v) > 0. \] (2)

More generally, one may defined a certain level of coverage, \( \epsilon \), given raise to the concept of strong coverage, as follows:

\[ \forall v \in V \exists i : 1 \leq i \leq n \mu_i(v) > \epsilon. \] (3)
This property is closely related to completeness. Completeness is a property of deductive systems (or theories) that has been used in the context of Artificial Intelligence to indicate that the knowledge representation scheme can represent every entity within the intended domain. When applied to fuzzy systems, and informally, this property states that a fuzzy system should be able to derive a proper action for every input [8]. Obviously, if a given region of the UoD is not cover by any membership function then fuzzy system cannot take any action for an input in that region.

Even for the case in which the system is required to take no action, this should be explicitly handled at the processing stage. For instance, using a rule such as IF $x$ is $X$ THEN $y$ is $ZE$, where $x$ represents the state of the system, $X$ is an intended linguistic term, $y$ represents the output of the system, and $ZE$ is the nearly zero linguistic term.

### III. CONSTRAINTS FOR SEMANTIC INTEGRITY

In this section, a set of constraints able to ensure the postulated semantic properties is given.

Some of the semantic properties of membership function can be satisfied easily by an a priori selection of certain membership function features. This is the case with the number of membership functions, and the use of normal membership functions, or even by forcing a membership function to be positioned at zero, if required. However the coverage and distinguishability properties should be monitorized and enforced during the optimization process of the overall system.

#### A. Constraints for Coverage

To ensure coverage one could use a constraint based directly on condition (3). This constraint can be expressed in terms of the sigma-count measure of the internal representation of the data. The sigma-count gives a measure of the cardinality of a fuzzy set $A$, as follows [7]:

$$
M_{s}(A) = \frac{\nu}{a_{1} + \cdots + a_{n}}
$$

where $a_{i} (i = 1, \cdots, n)$ are the degrees of membership defining the fuzzy set $A$. Therefore (3) can be ensured by

$$
\forall \nu \in V \forall M_{s}(\mathcal{L}(\nu)) < \epsilon
$$

where $\mathcal{L}(\nu)$ gives the internal representation of the external variable $\nu$, a fuzzy (sub)set.

Another more general constraint is based on the concept of optimal interfaces. This concept, introduced in [15], and further analyzed in [16], can be presented as follows.

**Definition 1:** Let $V = [a, b] \subseteq \mathbb{R}$. The ordered pair $(\mathcal{L}, \mathcal{N})$ is said a pair of optimal interfaces (or simply optimal interfaces) iff

$$
\forall \nu \in V \forall \mathcal{N}(\mathcal{L}(\nu)) = \nu
$$

where $\mathcal{L}: V \rightarrow [0, 1]^{n}$ is the mapping provided by the input interface of the variable $\nu$. $\mathcal{N}: [0, 1]^{n} \rightarrow V$ with $\mathcal{N}([0, \cdots, 0]) = \emptyset$ is the mapping provided by the output interface associated with the variable $\nu$.

For illustration of the application of this concept consider the single-input single-output system of Fig. 1(a). The application of the concept to the input variable yields

$$
\forall \nu \in X \mathcal{N}(\mathcal{L}(\nu)) = x
$$

as illustrated in Fig. 1(b). Notice that $\mathcal{N}$ is an auxiliary synthesis method sharing the same membership functions of $\mathcal{L}$, and used for ensuring that no information is discarded by the $\mathcal{L}$ transformation, instead of the output interface of the system, $\mathcal{N}$. Similarly, the application of the concept of optimal interfaces to the output variable yields

$$
\forall \nu \in Y \mathcal{N}(\mathcal{L}(\nu)) = y
$$

as illustrated in Fig. 1(c). Again, notice that $\mathcal{N}$ is not equal to $\mathcal{L}$; it is an auxiliary input mapping sharing the same membership functions of $\mathcal{N}$, and used for ensuring information equivalence at the output mapping.

Notice that it is straightforward to extend this concept to the general formats of external data, such as (crisp or fuzzy) intervals, or linguistic data.

The main motivation beyond this constraint is the following: when we perform a conversion of information formats it is desirable to ensure that we did not lose any information. One way to ensure this is to perform the inverse conversion and verify whether it is possible to obtain the original information. This idea has found elegant and useful results in the past. One of these results is the sampling theorem first suggested by Nyquist and then generalized by Shannon [13]. The sampling theorem gives us the conditions under which a continuous-
time signal can be univocally represented and precisely reconstructed by a collection of its instantaneous values equally spaced in time. The importance of this theorem pertains to the necessity of knowing when a continuous-time signal is uniquely represented by its discrete version. This validates the use of digital computers in the processing of continuous-time signals. Also in information theory, namely in the communications field, this idea constitute the formal basis for the analysis and synthesis procedures providing an elegant tool for information decomposition and encoding.

In the context of fuzzy systems condition, (6) gives us some important achievements also: it ensures coverage, it gives a condition for assuring information equivalence, and provides a simple constraint allowing an efficient implementation. The following subsections deal with each one of these achievements.

1) On Coverage:

**Proposition 1:** A class of membership functions satisfying the optimal interface definition (6) ensure the restrict form of coverage given by (2).

**Proof:** Consider, by absurdity, that \( \exists i \in [a, b] \forall \mu_i(v_0) = 0 \) or equivalently, \( \exists i \in [a, b] \mu_c(S_i(v_0)) = 0 \). However and by definition, the output mapping \( \mathcal{N}_c \) is undefined for the empty set \( \emptyset \), thus \( \mathcal{N}_c(S_c(v_0)) \neq v_0 \), which contradicts (6). Therefore, \( \forall i \in [a, b] \exists \mu_i(x) > 0 \) or \( \forall i \in [a, b] \exists M_i(S_i(v)) > 0 \).

Notice that this proposition does not dependent on the conversion methods used at the input or at the output interfaces. On the other hand, a strong form of coverage can be assured as soon as both \( \mathcal{L}_c \) and \( \mathcal{N}_c \) have been specified. For backing up this claim, one can observe that for triangular membership functions and for the COG defuzzification method, (6) assures a level of coverage of 0.5. Let

\[
\mathcal{L}_c = [\Delta(v; C_1, W_1) \Delta(v; C_2, W_2)]
\]

be the mapping provided by an input interface, where \( \Delta(v; C_i, W_i) \) represents the \( i \)th membership function, given by

\[
\Delta(v; C_i, W_i) = \Delta_i(v) = 1 - \frac{|v - C_i|}{W_i} \quad i = 1, 2
\]

such that \( 0 \leq \Delta(v; C_i, W_i) \leq 1 \) as illustrated in Fig. 2.

**Proposition 2:** Let \( \mathcal{L}_c, \mathcal{N}_c \) be a pair of interfaces where \( \mathcal{L}_c \) is given by (9) with triangular membership functions \( \Delta_1, \Delta_2 \) defined in \( V = [a, b] \); the operator \( \mathcal{N}_c \) is given by the COG method. If \( \mathcal{L}_c, \mathcal{N}_c \) satisfies (6) then \( \Delta_1, \Delta_2 \) ensures a coverage level of 0.5.

**Proof:** Under the conditions of the proposition it follows that

\[
\forall i \in [a, b] \frac{\Delta_i(v)C_i + \Delta_2(v)C_2}{\Delta_1(v) + \Delta_2(v)} = v
\]

that can be rewritten as

\[
\Delta_1(v)[C_1 - v] + \Delta_2(v)[C_2 - v] = 0;
\]

\[
\Delta_1(v) + \Delta_2(v) \neq 0.
\]

If \( v = C_1 \) then from (12) \( \Delta_2(v)[C_2 - C_1] = 0 \), i.e., \( \Delta_2(v) = 0 \lor C_2 = C_1 \). Notice that, \( C_2 = C_1 \) is an invalid solution since it violates (6)

\[
C_2 = C_1 \Rightarrow \forall v \in [a, b] v = C_1 \lor v = C_2.
\]

Therefore, for \( v = C_1 \) we have \( \Delta_1(v = C_1) = 1 \land \Delta_2(v = C_1) = 0 \). Moreover, from

\[
\Delta_2(v = C_1) = 1 - \frac{|C_1 - C_2|}{W_2} = 0
\]

it comes that \( W_2 = |C_1 - C_2| \land W_2 \neq 0 \).

Similarly, if \( v = C_2 \) then \( \Delta_1(v = C_2) = 0, \Delta_2(v = C_2) = 1.0, W_1 = |C_2 - C_1| \). Thus \( W_1 = W_2 = |C_1 - C_2| = |C_2 - C_1| \neq 0 \).

Now solving an equation system resulting from replacing successively \( v \) by \( a \) and \( b \) (the extremes of UoD) in (12), i.e., solving

\[
\begin{align*}
1 - |a - C_1| & \frac{1}{C_1 - C_2} (C_1 - a) \\
+ & 1 - \frac{|a - C_2|}{C_1 - C_2} (C_2 - a) = 0 \\
1 - |b - C_1| & \frac{1}{C_1 - C_2} (C_1 - b) \\
+ & 1 - \frac{|b - C_2|}{C_1 - C_2} (C_2 - b) = 0
\end{align*}
\]

it comes \( C_1 = a \land C_2 = b \), and thus \( W_1 = W_2 = |a - b| \). The membership functions are therefore

\[
\begin{align*}
\Delta_1(v) & = \max \left( 0, 1 - \frac{|v - a|}{a - b} \right) \\
\Delta_2(v) & = \max \left( 0, 1 - \frac{|v - b|}{a - b} \right).
\end{align*}
\]

The intersection of these functions gives a degree of coverage of 0.5.

**Corollary 1:** Let \( \mathcal{L}_c, \mathcal{N}_c \) be a pair of interfaces where \( \mathcal{L}_c \) is given by (9) with triangular membership functions \( \Delta_1, \Delta_2 \) defined in \( V = [a, b] \); the operator \( \mathcal{N}_c \) is given by the COG method. If \( \mathcal{L}_c, \mathcal{N}_c \) satisfies (6) then the two fuzzy sets form a fuzzy partition of the UoD, in the strict sense, i.e.,

\[
\forall v \in [a, b] \Delta_1(v) + \Delta_2(v) = 1.0
\]

**Proof:** It follows from the sum, in the UoD, of the functions (15).

In plain words, under the conditions of the Corollary 1, the membership functions represent linguistic terms that can be interpreted as partitioning the UoD with the intuitive notions of small and large, as shown in Fig. 3.

A straightforward (but not unique) way of generalizing the result of Proposition 2 for a generic number, \( n \), of membership functions is to split the UoD (the interval \([a, b]\)) into \( n - 1 \) subintervals, and use the membership functions determined in the previous proof to assign pairs of membership functions. This is shown in Fig. 4.
2) On Information Equivalence: Condition (6) is a sufficient condition for ensuring equivalence between external (data presented to the overall system) and internal (data processed by the fuzzy processing unit) formats of information [16], i.e., \( \forall v, \mu, \kappa \in \mathcal{V}; \quad v' \neq v'' \Rightarrow \mathcal{L}_c(v') \neq \mathcal{L}_c(v'') \).

In this respect, it should be stressed that (6) excludes the commonly used trapezoidal membership function distribution, while it elects triangular membership functions with 0.5 of overlapping as able of meeting the criterion (6). This does not means that input or output interfaces have to provide linear mappings.

**Proposition 3:** Let \((\mathcal{L}_c, \mathcal{N}_c)\) be a pair of interfaces where

\[
\mathcal{L}_c = [\Delta_1(v) \cdots \Delta_n(v)]'
\]

with all the membership functions \(\Delta_i(v)\) defined in \(\mathcal{V} = [a, b]\) and such that 1) In each point \(v \in \mathcal{V}\) only two functions are nonnull, 2) two consecutive functions overlap at 0.5, and 3) the following boundary conditions are satisfied \(\Delta_1(v = a) = 1.0 (\Delta_1(v = b) = 0.0)\) and \(\Delta_n(v = b) = 1.0 (\Delta_n(v = a) = 0.0)\). The \(\mathcal{N}_c\) operator is given by the COG method. Under these conditions \((\mathcal{L}_c, \mathcal{N}_c)\) satisfies the optimal interfaces criterion (6).

**Proof:** Since the membership functions \(\Delta_i(v); i = 1, \ldots, n\) are normal and unimodal, from the boundary conditions the extremes \(a\) and \(b\) are the centers of \(\Delta_1(v)\) and \(\Delta_n(v)\), \(C_1\) and \(C_n\), respectively. That is, \(a = C_1 < C_2 < \cdots < C_j < \cdots < C_n = b\). Between any two consecutive centers \(C_j, C_{j+1}\) with \(j = 1, \ldots, n - 1\) there are only two membership functions, and these are defined by the equations

\[
j = -\frac{1}{C_{j+1} - C_j} v + \frac{C_{j+1}}{C_{j+1} - C_j}
\]

and

\[
y_{j+1} = \frac{1}{C_{j+1} - C_j} v - \frac{C_j}{C_{j+1} - C_j}
\]

valid in \([C_j, C_{j+1}]\). According to the COG method, the converted value \(\bar{v}\) for \(v \in [C_j, C_{j+1}]\) is shown in (18) at the bottom of the page.

This result is a formal reason for using this type and distribution of membership functions (Fig. 4), which are quite common in the applications of fuzzy sets and systems (e.g., [8] and [14]).

3) On Implementation: Condition (6) provides a simple constraint allowing an efficient implementation. In this respect, and from a practical stand point (i.e., assuming that we are using digital computers) we can consider that the UoD of a given variable is represented by a sequence of samples, say \(\{x[k]\} k = 1, \ldots, N\), discretized according to some appropriated sampling criterion used in the context of the problem to be solved by the overall system. Therefore condition (6) gives rise to the following nonlinear constraint:

\[
J_1 = \frac{1}{2} \sum_{k=1}^{N} (x[k] - \bar{x}[k])^2
\]

where \(\bar{x}[k]\) is given by \(\mathcal{N}_c(\mathcal{L}_c(x))\). Let

\[
\mathcal{L}_c(x) = [\mu_1(x) \mu_2(x) \cdots \mu_n(x)]'
\]

and \(\mathcal{N}_c\) can be given by any differentiable defuzzification method, such as the COG. Thus

\[
\bar{x} = \mathcal{N}_c(\mathcal{L}_c(x)) = \frac{\sum_{i=1}^{n} \mu_i(x) a_i}{\sum_{i=1}^{n} \mu_i(x)}
\]

where \(a_i\) is the mean (or modal) value of the \(i\)th membership function in (20). Appendix A includes the details required for the implementation of this constraint.

B. Constraints for Distinguishability

For ensuring distinguishability we will adopt a constraint on the region of the hypercube where the internal (fuzzy) representation of the fuzzy system may exist. This can be done using again the sigma-count measure. As a brief motivation consider two membership functions, say \(\mu_i\) and \(\mu_j\), very close to each other. In this case, there will be a certain \(x\) in the UoD for which its internal representation will have components \(\mu_i(x)\) and \(\mu_j(x)\) with about the same high value, as shown in Fig. 5. On the other hand, if the membership functions are far enough, these components will not have simultaneously high values for none of the elements in the UoD.

We can formalize this reasoning in the following constraint:

\[
\forall x \in \mathcal{V}; M_p(\mathcal{L}_c(v)) \leq 1.
\]

Notice that asking for a sigma-count always strictly less than 1 is not feasible since we are assuming normal membership functions. On the other hand, asking for a sigma-count greater than 1 is equivalent to remove the constraint.

The parameter \(p\) in (22) is used for controlling the strength of the constraint. For \(p = 1\) we have a strong constraint whereas it can be eliminated as \(p \to +\infty\).

Appendix B includes the detailed formulae required for the implementation of this constraint.
IV. IMPLEMENTATION ISSUES

This section discusses application oriented aspects of membership function optimization. One example is included showing both the evolution of the optimization process, and the impact of the proposed constraints on the performance of a fuzzy system.

A. Implementing Constraints in Optimization Algorithms

Numerical solutions of optimization problems with constraints are one of the main concerns of the mathematical programming area, cf. [21]. There are a variety of techniques for dealing with equality and inequality constraints. Here we will describe one of these techniques, known as the penalty function method. This method is computationally appealing and has been used both in parameter and function optimization problems. For easy reference, it is presented below [21].

Let $J(\theta)$ be a performance index to be minimized, such as the MSE (mean squared errors), with respect to the parameter $\theta$ subject to the constraint $f(\theta) \leq 0$. The penalty function method allows us to consider the optimization of

$$J(\theta) = J(\theta) + Kf(\theta)^2 1(f(\theta))$$

subject to no constraints, where $K$ is a positive penalty factor, and $1(x)$ is the unit step function defined as

$$1(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0. \end{cases}$$

This enables us to solve constrained optimization problems using unconstrained techniques such as the backpropagation algorithm. To be more precise, we can add the two semantic constraints given in Appendixes A and B, $J_1$ and $J_2$, respectively, to the backpropagation algorithm simply by constructing a performance index, $\mathcal{J}$ as the sum of three components, the usual MSE performance index, $J$, together with a linear combination of $J_1$ and $J_2$. That is

$$\mathcal{J} = J + K_1J_1 + K_2J_2.$$  

Naturally, the derivatives of $J$ with respect to the parameter $\theta$ are now given by

$$\frac{\partial \mathcal{J}}{\partial \theta} = \frac{\partial \mathcal{J}}{\partial \theta} + \frac{\partial \mathcal{J}_1}{\partial \theta} + \frac{\partial \mathcal{J}_2}{\partial \theta}$$

where $\mathcal{J}/\partial \theta$ is given as before by the backpropagation algorithm, and the other derivatives are given in Appendixes A and B, respectively.

B. Example

Most fuzzy systems are involved in approximation tasks (e.g., fuzzy models, fuzzy controllers or fuzzy pattern matchers). Therefore, it might be relevant to present an example showing the impact of the proposed constraints on a fuzzy modeling problem. The task is to model the sigmoid function $d = 1/(1 + e^{-\tau})$ in the range [-2, 6] based on a pre-collected data set. The data set consists of input–output pairs $(x[k], d[k])$, $k = 1, \ldots, 51$ nonuniformly distributed. The following performance index is considered:

$$J = \frac{1}{2} \sum_{k=1}^{\infty} (d[k] - y[k])^2$$

where $d[k]$ is the $k$th desired value of the data set, and $y[k]$ is the output of the model when the $k$th input value, $x[k]$, is presented at the input of the model. Hereafter, the index $k$ will be omitted for simplicity of notation, whenever there will be no ambiguity.

Two fuzzy models are considered. These have exactly the same processing structure, and the same initial conditions, namely for the initial value of interfaces parameters. However they are designed in two different ways: one of the models has no constraints on its membership functions, while the other model uses the proposed constraints.

For both model, the numeric/linguistic interface is implemented as

$$\mathcal{L}_i(x) = X = [\mu(x, \theta_i^1), \ldots, \mu(x, \theta_i^n), \ldots, \mu(x, \theta_i^n)]^T$$

$\mu(x, \theta_i^k)$ being the following parametrized Gaussian membership function

$$\mu(x, \theta_i^k) = \exp\left(-\frac{(x - \theta_i^c)^2}{2\theta_i^w^2}\right)$$

where $\theta_i^c = [\theta_i^c_1, \theta_i^c_2]^T$, $\theta_i^c_1$ being the center, and $\theta_i^w$ the width of the $i$th membership function, defined in the input space.

The processing stage common to both models is defined by

$$Y = X \cdot R$$

where $Y = [Y_1, \ldots, Y_m]^T$, $X = [X_1, \ldots, X_n]^T$, $R = [R_{ij}]^T$ for $i = 1, \ldots, n; j = 1, \ldots, m$, and * denotes T's composition, such that

$$Y_{2} = T_{x=s}X_{s}R_{ij}$$

where $T$ is a t-norm, assumed as the product, and * is a s-norm assumed as $s(x, y) = x + y - xy$, cf. [6]. For both models, the linguistic/numeric interface is implemented according to the following expression:

$$y[k] = \frac{\sum_{i=1}^{m} Y_{j} \theta_{j}^y}{\sum_{i=1}^{m} Y_{j}}$$

Using a notation for the membership functions defined at the output space similar to that used for the membership function at the input space, then $\theta_{j}^y$ represents the center of the $j$th output membership function. For both models, five Gaussian membership functions were assumed at both the input and output interfaces, i.e., $m = n = 5$.

The parameters of the models are computed using the gradient method, i.e., they are computed using the general updating expression

$$w_{k+1} = w_k + \eta \Delta w_k$$

where $w_k$ is a generic parameter at the iteration $k$, $\Delta w_k$ its increment at the same iteration, and $\eta$ is the size of the optimization steps (or learning rates).
The initial fuzzy relation $R$ used for both models was set up randomly. For the unconstrained model, the increments of the parameters are as follows.

- For the output membership functions
  \[
  \Delta \theta_j^y = -\frac{\partial J}{\partial \theta_j^y} = -\frac{1}{N} \sum_{k=1}^{N} (d[k] - y[k]) Y_j[k] \sum_{i=1}^{m} Y_i[k].
  \]  
  (33)

- For the entries of the fuzzy relation
  \[
  \Delta R_{ij} = -\frac{\partial J}{\partial R_{ij}} = -\frac{1}{N} \sum_{k=1}^{N} (d[k] - y[k]) \frac{\partial y[k]}{\partial Y_j[k]} \frac{\partial Y_j[k]}{\partial R_{ij}} \]  
  (34)

  with
  \[
  \frac{\partial y}{\partial Y_j} = \frac{\theta_j^y - y}{\sum_{i=1}^{m} Y_i}.
  \]  
  (35)

  and
  \[
  \frac{\partial Y_j}{\partial R_{ij}} = (1 - X_i) T_{i=1}^{m} (X_i R_{ij}).
  \]  
  (36)

The above expression is only valid for the selected $T$ and $s$ norms.

- For the input membership functions
  \[
  \Delta \theta_i^x = -\frac{\partial J}{\partial \theta_i^x} = -\frac{1}{m} \sum_{j=1}^{m} (d[k] - y[k]) \frac{\partial y[k]}{\partial Y_j} \frac{\partial Y_j}{\partial \theta_i^x} \]  
  (37)

  with
  \[
  \frac{\partial y}{\partial X_i} = \sum_{j=1}^{m} \frac{\partial y}{\partial Y_j} \frac{\partial Y_j}{\partial X_i} \]  
  (38)

The derivatives $\partial y/\partial Y_j$ are computed as in (35), while

\[
\frac{\partial Y_j}{\partial X_i} = (1 - R_{ij}) T_{i=1}^{m} (X_i R_{ij}).
\]  
(39)

valid only for the selected $T$ and $s$ norms, and

\[
\frac{\partial X_i}{\partial \theta_i^x} = \frac{\partial R}{\partial X_i} \frac{\partial X_i}{\partial \theta_i^x}.
\]  
(40)

The optimization step used is $\eta = 0.01$. Fig. 6 shows the evolution of the performance index for this unconstrained model. A local minimum of $J$ was reached around iteration 2000, and then the performance of the model has degraded. This behavior can be explained looking at the interim status of the membership functions. These are shown in Figs. 7 and 8, for the input and output interface, respectively.

From these figures it is clear the degradation of membership functions, in the sense that it has become difficult to assign linguistic terms to them. The membership function of Fig. 7 centered around 1.0, has become a singleton, and there are uncovered regions of the UoD output interface in Fig. 8.

The optimization process of the unconstrained model.

The estimated fuzzy relation for this unconstrained model at the
The result of validating this model (with the parameters corresponding to the local minimum of $J$) using inputs in the range $[-2,6]$ is shown in Fig. 9. A poor performance can be seen. Without appropriated constraints, a number (usually big) of successive initializations of the learning process is required to find a better local minimum.

Using the same structure, and the same initial conditions, a model is now built with the proposed semantic constraints. According to the last sections, the following performance indexes are built to each group of parameters of the model, the parameter increments being computed accordingly.

- For the parameters of the input interface

\[ \mathcal{T}^X = J + K_1^X J_1^X + K_2^X J_2^X \]  
\[ \Delta \theta_i^X = -\frac{\partial \mathcal{T}^X}{\partial \theta_i^X} \]  

(42)

- For the entries of the fuzzy relation $R$: $J$ is defined as before, see (27), as well as its increments [see (34)].

- For the parameters of the output interface

\[ \mathcal{T}^Y = J + K_1^Y J_1^Y + K_2^Y J_2^Y \]  

(44)

with

\[ \Delta \theta_j^Y = -\frac{\partial \mathcal{T}^Y}{\partial \theta_j^Y} = -\frac{\partial J}{\partial \theta_j^Y} - K_1^X \frac{\partial J_1^Y}{\partial \theta_j^Y} = K_2^X \frac{\partial J_2^Y}{\partial \theta_j^Y} \]  

(45)

where $\partial J/\partial \theta_j^Y$ is computed by (33) as previously, $\partial J_1^Y/\partial \theta_j^Y$ is computed applying the derived formulae of Appendix A to the input data, and similarly $\partial J_2^Y/\partial \theta_j^Y$ is computed applying the formulae derived in Appendix B to the input data; the constants $K_1^X$ and $K_2^X$, are determined empirically, and set as $K_1^X = 100$, $K_2^X = 1000$, after some preliminary tests.

- For the parameters of the input interface

\[ \mathcal{T}^X = J + K_1^X J_1^X + K_2^X J_2^X \]  

(43)

where $\partial J/\partial \theta_i^X$ is computed by (37) as previously, $\partial J_1^X/\partial \theta_i^X$ is computed applying the derived formulae of Appendix A to the input data, and similarly $\partial J_2^X/\partial \theta_i^X$ using the formulae derived in Appendix B in the output data; the constants are $K_1^X = 2$, $K_2^X = 200$.

From Fig. 10, it can be seen that the performance index $J$ decreases monotonically with the number of iterations. Looking at the evolution of the indexes reflecting the constraints, $J_1^X$, $K_2^X$,$ J_1^Y$, and $J_2^Y$ shown in Fig. 11 a similar monotonic tendency is verified. Notice that by minimizing $\mathcal{T}^X$, and $\mathcal{T}^Y$, we are minimizing $J$ subject to the constraints.

Figs. 12 and 13 shows the interim status for the input and output membership functions, respectively. As it can be visualized the membership function are kept semantically meaningful, i.e., they can be easily identified with the linguistic terms big, medium big, medium, medium small, and small, for instance.
Fig. 10. Evolution of the performance index $J$ over the number of iterations for the constrained model.

Fig. 11. Evolution of the indexes representing the constraints coverage and distinguishability for both the input and output interfaces.

The optimization process was stopped for $J < 5e^{-6}$, and the fuzzy relation found at this minimum is

$$R = R_{ij} = egin{bmatrix} 0.0000 & 0.2840 & 0.5123 & 0.4890 & 1.0000 \\ 1.0000 & 0.3103 & 0.0000 & 0.0984 & 0.5660 \\ 0.9000 & 0.0032 & 0.7238 & 0.7655 & 0.4481 \\ 0.8218 & 0.5326 & 0.4849 & 0.0563 & 0.2719 \\ 0.8045 & 0.1940 & 0.4988 & 0.1561 & 0.4677 \end{bmatrix} \quad (46)$$

The result of validating this constrained model using a set of inputs in the range $[-2, 6]$ (some of them not used in the optimization) is shown in Fig. 14. The model has succeeded the proposed modeling task.

V. CONCLUSIONS

The optimization of fuzzy systems (such as controllers, models, or classifiers) using biological inspired strategies (such as neural networks learning rules, or evolutionary optimization techniques like genetic algorithms) cannot ensure the linguistic interpretation of the resulting systems. Namely, one cannot guarantee that the resulting membership functions have any semantic value in what concerns their ability to be interpretable as linguistic terms. In this case, we are neglecting one of the most interesting features of fuzzy systems, i.e., the insight provided on the linguistic relationship between variables.

A set of comprehensive semantic properties for membership functions was outlined. This includes a moderate number of membership functions, distinguishability, normality, natural zero positioning, and coverage. These properties are viewed as basic requirements for semantic integrity, as they form a minimum set required for rendering membership functions interpretable for human beings. From these coverage and distinguishability properties should be monitored during the optimization process. Two constraints were proposed for this purpose. These are the concept of optimal interfaces (for coverage) and the sigma-count of an internal variable (for distinguishability). The constraints have been shown to be able to meet the semantic properties. The importance of the proposed constraints has been illustrated by an example on an approximation task. Several application oriented issues were considered in this study. These
include the relevant formulae required for an efficient implementation of the semantic constraints and the inclusion of (nonlinear inequality) constraints in optimization algorithms.

APPENDIX A

In this appendix, the derivatives required for implementing optimal interfaces serving as the coverage constraint are developed. The sequence of numeric data \{\{x[1], x[2], \ldots, x[N]\}\} corresponding to the values of an external variable is assumed available.

Given the constraint (19)

$$J_1 = \frac{1}{2} \sum_{k=1}^{N} (\bar{x}[k] - x[k])^2$$

where \(x[k]\) is the \(k\)th numeric sample, and \(\bar{x}[k]\) is given by any differentiable defuzzification method such as

$$\bar{x}[k] = \frac{\sum_{i=1}^{n} \mu_i(x[k]) \theta_{i1}}{\sum_{i=1}^{n} \mu_i(x[k])}. \quad (47)$$

The \(i\)th membership functions is represented by a generic membership function \(\mu_i\) parametrized in the vector \(\theta_i; (i = 1, \ldots, n)\), \(n\) being the total number of membership functions. The vector \(\theta_i\) has two elements

$$\theta_i = [\theta_{i1} \quad \theta_{i2}]^T$$

with \(\theta_{i1}\) standing for the center and \(\theta_{i2}\) standing for the width. Therefore

$$\frac{\partial J_1}{\partial \theta_{i1}} = -\sum_{k=1}^{N} (\bar{x}[k] - x[k]) \frac{\partial \bar{x}[k]}{\partial \theta_{i1}} \quad (48)$$

as

$$\frac{\partial \bar{x}[k]}{\partial \theta_{i1}} = \left[ \frac{\partial \bar{x}[k]}{\partial \theta_{11}} \frac{\partial \bar{x}[k]}{\partial \theta_{12}} \right]^T$$

then from (47) we have (49) (shown at the bottom of the next page).

$$\frac{\partial \bar{x}[k]}{\partial \theta_{12}} = \left( \begin{array}{c} \frac{\partial \mu_i(x[k])}{\partial \theta_{11}} + \mu_i(x[k]) \sum_{i=1}^{n} \mu_i(x[k]) \frac{\partial \mu_i(x[k])}{\partial \theta_{11}} \\ \sum_{i=1}^{n} \mu_i(x[k]) \end{array} \right) \frac{\partial \mu_i(x[k])}{\partial \theta_{12}} \quad (50)$$

APPENDIX B

In this appendix, the derivatives required for implementing the distinguishability constraint are developed. Again, the sequence of numeric data \{\{x[1], x[2], \ldots, x[N]\}\} corresponding to an external variable is assumed available.

The constraint (22) can be rewritten for any \(k = 1, \ldots, N\), as

$$M_r(x[k]) - 1 \leq 0 \quad (51)$$

Since this is an inequality constraint we will have to introduce the unit step function (24) that only “activates” the constraint if necessary. This function included in (51) gives rise to the following operational expression:

$$J_2 = \frac{1}{2} \sum_{k=1}^{N} (M_r(x[k]) - 1)^2 \mathbf{1}(M_r(x[k]) - 1) \quad (52)$$

Now assuming that \(\mathbf{1}(M_r(x[k]) - 1) = 1\), and bearing in mind that

$$M_r(x[k]) = \sqrt[n]{\sum_{i=1}^{n} \mu_i(x[k])}$$

\(J_2\) reads as

$$J_2 = \frac{1}{2} \sum_{k=1}^{N} \left[ \left( \sum_{i=1}^{n} \mu_i(x[k]) \right)^{1/p} - 1 \right]$$

thus

$$\frac{\partial J_2}{\partial \theta_{i1}} = \sum_{k=1}^{N} \left( \left( \sum_{i=1}^{n} \mu_i(x[k]) \right)^{1/p} - 1 \right)$$

$$\frac{\partial J_2}{\partial \theta_{i2}} = \sum_{k=1}^{N} \frac{n}{\left( \sum_{i=1}^{n} \mu_i(x[k]) \right)^{1/p}} \left( \sum_{i=1}^{n} \mu_i(x[k]) \right)^{1/p} \frac{\partial \mu_i(x[k])}{\partial \theta_{i1}}$$

$$= \sum_{k=1}^{N} \frac{\left( \sum_{i=1}^{n} \mu_i(x[k]) \right)^{1/p} - 1}{\left( \sum_{i=1}^{n} \mu_i(x[k]) \right)^{1/p}} \frac{\partial \mu_i(x[k])}{\partial \theta_{i1}} \quad (53)$$

(49)
Therefore these derivatives can be written in the following general expression:
\[
\frac{\partial J_S}{\partial \theta_i} = \sum_{k=1}^{N} \left( \mathcal{M}_p(X[k]) - 1 \right) \left( \mathcal{M}_p(x[k]) - 1 \right) \cdot \left( \sum_{i=1}^{N} \frac{\partial \mu_i(x[k])}{\partial \theta_i} \right). 
\] (54)

Fig. 13. Interim status of the output membership functions during the optimization process of the constrained model.

Fig. 14. Evaluation of the constrained model. Solid line: target data, dashed line: model output.

REFERENCES