Development of A Systematic Methodology of Fuzzy Logic Modeling

Mohammad R. Emami, I. Burhan Türksen, Senior Member, IEEE, and Andrew A. Goldenberg, Fellow, IEEE

Abstract—This paper proposes a systematic methodology of fuzzy logic modeling as a generic tool for modeling of complex systems. The methodology conveys three distinct features: 1) a unified parameterized reasoning formulation; 2) an improved fuzzy clustering algorithm; and 3) an efficient strategy of selecting significant system inputs and their membership functions. The reasoning mechanism introduces four parameters whose variation provides a continuous range of inference operation. As a result, we are no longer restricted to standard extremes in any step of reasoning. Unlike traditional approach of selecting the inference mechanism a priori, the fuzzy model itself can then adjust the reasoning process by optimizing the inference parameters based on input–output data. The fuzzy rules are generated through fuzzy c-means (FCM) clustering algorithm. Major bottlenecks of the algorithm are addressed and analytical solutions are suggested. Furthermore, we also address the classification process in fuzzy modeling to extend the derived fuzzy partition to the entire output space. This issue remains unattained in the current literature. In order to select suitable input variables among a finite number of candidates (unlike traditional approaches) we suggest a new strategy through which dominant input parameters are assigned in one step and no iteration process is required. Furthermore, a clustering technique called fuzzy line clustering is introduced to assign the input membership functions. In order to evaluate the proposed methodology, two examples—a nonlinear function and a gas furnace dynamic procedure—are investigated in detail. The significant improvement of the model is concluded compared to other fuzzy modeling approaches.

Index Terms—Approximate reasoning, fuzzy clustering, fuzzy modeling, fuzzy systems.

I. INTRODUCTION

TRADITIONAL quantitative techniques of systems modeling have significant limitations. In most cases, it is quite difficult to adequately describe the behavior of a nonlinear system by mathematical models, especially when the structure of the system is unknown. Even if one knows the structure, numerical model representations usually become irrelevant and computationally inefficient as the complexity grows. After all, there are a lot of uncertainties, unpredictable dynamics, mutual interactions, and other unknown phenomena that cannot be mathematically modeled at all.

In attempt to obtain more flexibility and more effective capability of handling and processing uncertainties in complicated and ill-defined systems, Zadeh [1] proposed a linguistic approach as the model of human thinking, which introduced the fuzziness into systems theory. This idea was developed further to a new class of systems called fuzzy systems by Tong [2], Pedrycz [3], Gupta et al. [4], Sugeno and Yasukawa [5], and Yager and Filev [6]. The linguistic approach of system modeling can be formulated in three distinguishing features [1]:

1) the use of linguistic variables in place of or in addition to numerical variables;
2) the characterization of simple relations between variables by IF–THEN fuzzy rules;
3) the formulation of complex relations by fuzzy reasoning algorithms.

Therefore, the central characteristic of fuzzy systems is that they are based on the concept of fuzzy partitioning of the information. The decision-making ability of the fuzzy model depends on the existence of rule base and fuzzy reasoning mechanism. In the most general form, the encoded knowledge of a multi-input multi-output (MIMO) system can be interpreted by fuzzy models consisting of IF–THEN rules with multi-antecedent and multiconsequent variables (with r antecedents, s consequents, and n rules)

\[
\text{IF } U_1 \text{ is } B_{11} \text{ AND } U_2 \text{ is } B_{12} \text{ AND } \ldots \text{AND } U_r \text{ is } B_{1r}.
\text{THEN } V_1 \text{ is } D_{11} \text{ AND } V_2 \text{ is } D_{12} \text{ AND } \ldots \text{AND } V_n \text{ is } D_{1n},
\]

\text{ALSO}

\ldots
d

\text{ALSO}

\text{IF } U_1 \text{ is } B_{n1} \text{ AND } U_2 \text{ is } B_{n2} \text{ AND } \ldots \text{AND } U_r \text{ is } B_{nr}.
\text{THEN } V_1 \text{ is } D_{n1} \text{ AND } V_2 \text{ is } D_{n2} \text{ AND } \ldots \text{AND } V_n \text{ is } D_{ns},
\]

where \(U_1, U_2, \ldots, U_r\) are input variables and \(V_1, V_2, \ldots, V_n\) are output variables, \(B_{ij}\) (\(i = 1, \ldots, n, j = 1, \ldots, r\)) and \(D_{ik}\) (\(i = 1, \ldots, n, k = 1, \ldots, s\)) are fuzzy sets of the universes of discourse \(X_1, X_2, \ldots, X_r\), and \(Y_1, Y_2, \ldots, Y_s\) of \(U_1, U_2, \ldots, U_r\) and \(V_1, V_2, \ldots, V_s\), respectively. The set of rules operating with linguistic values of input–output variables appears to be analogous to the system of equations used for presentation of linear and nonlinear systems. The fuzzy sets \(B_{ij}\)’s and \(D_{ik}\)’s are parameters of the fuzzy model; the number of the rules determines its structure.

Conceptually, a system with multiple independent outputs can be considered as several groups of single output systems, separately. Consequently, the general rule structure of a MIMO fuzzy system can also be presented as a collection of multi-input single-output (MISO) fuzzy systems such that for a system with \(q\) output, each multiconsequent rule is broken into \(s\) single-consequent rules. Although the number of rules in the new fuzzy system will be increased, modeling and inference
would be more straightforward for MISO fuzzy systems. That is the reason why the literature concentrates on MISO rules as a generic presentation of fuzzy systems.

In investigation about fuzzy modeling, there are several issues that must be clearly addressed. In regard to where system information is presented, two basic categories have been suggested so far. In the first category, both antecedent and consequent parts of IF–THEN rules usually consist of vague predicates. In these models, fuzzy quantities are associated with linguistic labels and the fuzzy model is essentially a qualitative expression of the system that uses natural language expressions. This type of fuzzy model has been investigated by many researchers [2]–[5]. The second category of fuzzy models is formed by logical rules that have a fuzzy antecedent part and a functional consequent part; essentially they are a combination of fuzzy and nonfuzzy models. This category was initially proposed by Takagi–Sugeno–Kang [7] and is referred to as TSK fuzzy models. TSK fuzzy models have effective potential for expressing quantitative information and are computationally efficient. However, we are more concerned with a general framework of qualitative modeling, therefore, it is preferable to adopt the more general type of fuzzy IF–THEN rules, which consist of fuzzy variables in both their antecedent and consequent.

Regarding the methodology of fuzzy modeling, two basic approaches can be recognized through current literature. In traditional methods of fuzzy modeling, it is assumed that expert information is available. Therefore, in this approach the linguistic description is constructed subjectively on the basis of the a priori knowledge about the system. The first significant applications of fuzzy logic in the modeling of complex systems, especially in modeling operator’s actions in the area of control engineering [8], was a clear evidence of the great power of this novel approach for dealing with the complexity of the real world. This direct approach to fuzzy modeling has some inherent limitations. The main drawbacks of this approach are subjectivity and lack of generalization and dependence on expert’s knowledge that sometimes could be faulty. In search for more objectivity in constructing fuzzy models, scientists tried to develop more formal techniques that could use available data to augment human knowledge or even generate new knowledge. Therefore, the second direction in the development of fuzzy models, inspired by the classic systems theory, is based on the use of input–output data. In the language of systems theory, this approach can be regarded as a process of system identification [5].

The main issues of fuzzy modeling are about the reasoning mechanism, i.e.: 1) the way information is implied in each rule; 2) the way it is aggregated among a set of rules; 3) the way new information is inferred from the set of rules; and 4) the way fuzzy inferred values are translated into their crisp correspondences. In current methods of fuzzy modeling, the connectives of the reasoning mechanism are selected a priori the identification procedure without any theoretical basis. In the proposed systematic fuzzy modeling methodology, by using the fundamentals of the theory of approximate reasoning, we introduce a unified parameterized formulation for the inference process as a general framework for reasoning with a fuzzy model. In this way, we attempt to spread the objectivity to the selection of the reasoning mechanism in fuzzy modeling and control. Section II discusses the above subject matter.

Other issues of the objective approach of fuzzy modeling are due to the fuzzy system identification problem, which is the subject of Section III. The procedure of fuzzy system identification and modeling can be represented by the flow chart in Fig. 1.

Similar to conventional system identification, the problem can be divided into two types: structure identification and parameter identification. In fuzzy modeling, structure identification proceeds through two steps: 1) input variables are recognized among all possible input candidates and suitable membership functions are assigned for them and 2) input–output relations (IF–THEN rules) are specified. In this paper, fuzzy clustering is considered as an intuitive approach to objective rule generation in fuzzy modeling. In this direction, in Section III-A, we inspect the fuzzy c-means (FCM) clustering algorithm and attempt to improve the objectivity of this technique by deriving suitable indexes for selection of the number and the level of fuzziness of clusters. An efficient method is also suggested for the assignment of initial cluster center locations. Furthermore, we address the classification process to extend the derived fuzzy partition for the sample data to the entire space. In Section III-B, we suggest a new strategy through which dominant input parameters are assigned in one step and no iteration is required. Furthermore, the convex input membership functions are also derived through fuzzy line clustering. The second type of fuzzy system identification, i.e., parameter identification, consists of the derivation of the optimum inference parameters and adjustment of input and output membership functions. This is discussed in Section III-C. Finally, the proposed systematic fuzzy modeling algorithm is summarized in Section IV. Two examples are illustrated in Section V followed by conclusions in Section VI.
II. REASONING MECHANISM

The reasoning process is divided into the following steps.

1) Fuzzy aggregation of antecedents in each rule (AND connective).
2) Implication relation for each individual rule (IF–THEN connective).
3) Aggregation of the rules (ALSO connective).
4) Inference from the set of rules using the input to obtain the fuzzy output.
5) Defuzzification of the output.

In existing methods of fuzzy modeling, the connectives in all of the above steps of reasoning are selected a priori the identification procedure without any theoretical basis. The number of selections is limited to a few known aggregation and implication operators satisfying the compatibility conditions [9]. A general and unified framework for the reasoning process is constructed in [10]. Starting from the basic elements of reasoning, i.e., the connective operators (AND, ALSO, and IF–THEN), we adopt a special parameterized family of triangular functions, which, due to its simplicity and symmetry, is appropriate for our purpose. By extending the binary operation to n-ary operation and by proving the validity of De Morgan law for n-ary operation, we are able to parameterize the reasoning formulation for MISO systems. Focusing on systems with crisp input, which is the case in most applications of fuzzy modeling and control, and by proving the property of distributivity for t-norm and t-conorm operators in this case, it is proved that the two methods of inference from a rule set, first-aggregate-then-infer (FATI) and first-infer-then-aggregate (FITI), always give the same fuzzy output. Moreover, we present uniform reasoning formulations for Mamdani’s approximation and formal logical reasoning approaches that lead to a unified parameterized fuzzy reasoning formulation. As a result, four reasoning parameters \( p, q, \alpha, \) and \( \beta \) are introduced whose variation will cause a continuous range of variation for the reasoning mechanism. Therefore, we are no longer restricted to the extremes in any step of the reasoning process. Consequently, unlike traditional approach of selecting the inference mechanism a priori for fuzzy modeling, the fuzzy model itself can adjust the reasoning process by adjusting the above parameters according to input–output data. In order to reduce the computational effort, a fast algorithm for the calculation of the parameterized family of triangular functions is introduced.

A. Fuzzy Sets Intersection, Union, and Negation

The properties generally expected to be satisfied by the commutative-associative class of intersection-union operators, which are known as the t-norm \((T)\) and t-conorm \((S)\) set of operators, are identified with the following axioms [11].

\[ F1 \text{ Boundary} \]
\[ T(a, 1) = a, \quad T(a, 0) = 0 \quad \text{for all } a \in [0, 1] \]

\[ S(a, 0) = a, \quad S(a, 1) = 1 \quad \text{for all } a \in [0, 1] \]

\[ F2 \text{ Commutativity} \]
\[ T(a, b) = T(b, a) \quad \text{for all } a, b \in [0, 1] \]

\[ S(a, b) = S(b, a) \quad \text{for all } a, b \in [0, 1] \]

\[ F3 \text{ Associativity} \]
\[ T(T(a, b), c) = T(a, T(b, c)) \quad \text{for all } a, b \in [0, 1] \]

\[ S(S(a, b), c) = S(a, S(b, c)) \quad \text{for all } a, b \in [0, 1] \]

\[ F4 \text{ Monotonicity} \]
\[ \text{If } a \geq a' \quad \text{and } b \geq b' \text{ then:} \]
\[ T(a, b) \geq T(a', b') \quad \text{for all } a, a', b, b' \in [0, 1] \]

\[ S(a, b) \geq S(a', b') \quad \text{for all } a, a', b, b' \in [0, 1] \]

The main features of this parameterized form are its analytical simplicity and symmetry. These are basic advantages for future implementations, as will be demonstrated in a sequel. Another advantage is that as \( p \) (\( p > 0 \)) changes continuously, this parametric form does cover all t-norms and t-conorms from the special class of Zadeh operators to drastic operators.

Analogous to classical set theory, De Morgan laws establish a link between union and intersection via complementation. If a t-norm \( T \) and a t-conorm \( S \) and a strong negation \( \alpha \) satisfy De Morgan laws as

\[ T(a, b) = 1 - [(1 - a)^p + (1 - b)^q - (1 - (a^p)(1 - b)^q)^{1/p} \quad \text{for all } a, b \in [0, 1] \]

\[ S(a, b) = \left[ a^p + b^q - a^p b^q \right]^{1/p} \quad \text{for all } a, b \in [0, 1] \]

Theorem 1 [13]: The \( n \)-ary operators \( T_n \) and \( S_n \) on \([0, 1]\) satisfy all properties as their binary correspondence.

Theorem 2 [10]: If the binary operators \( T \) and \( S \) are dual with standard negation, then their corresponding \( n \)-ary extensions \( T_n \) and \( S_n \) are also dual.

We may call \( T_n \) and \( S_n \) the extension of triangular norm and conorms to \( n \) arguments and we will omit the subscript \( n \) and simply write \( T \) and \( S \) for the class of mapping generated by the triangular norms and conorms.
The extension of the introduced $t$-conorm operators to $n$ arguments reduces to the following formulation [10]:

$$S(a_1, a_2, \ldots, a_n) = \left[ \sum_{i=1}^{n} a_i^p - \sum_{i=1}^{n} a_i^p a_j^p + \sum_{i=1}^{n} a_i^p \prod_{k \neq i} a_k^p \right]^{1/p}$$

(4)

In order to obtain the extended $t$-norm by using Theorem 2, we simply use De Morgan law

$$T(a_1, a_2, \ldots, a_n) = 1 - S((1 - a_1), (1 - a_2), \ldots, (1 - a_n))$$

(5)

As it is obvious from the (4), the computational complexity exponentially increases when the number of argument $(n)$ becomes larger. With a rather good algorithm we need $(2^n - 2)$ additions and $(2^n - n - 1)$ multiplications and $(n)$ power operations to compute the above formulation. Hence, the computational complexity is of $O(2^n)$. This problem is a bottleneck for further applications of fuzzy reasoning in fuzzy systems and control. In order to significantly reduce the number of arithmetic operations, we change the (4) to the following, which can be confirmed by inspection:

$$S(a_1, a_2, \ldots, a_n) = \left[ a_1^p + (1 - a_1^p) a_2^p + (1 - a_1^p)(1 - a_2^p) \right]^{1/p} \cdot \left[ \prod_{i=1}^{n-2} a_i^p (1 - a_i^p) (1 - a_i^p) \right]^{1/p}$$

(6)

Then, for a set $(a_1, a_2, \ldots, a_n)$ ($n > 2$), the new formulation can be computed by the following algorithm:

**STEP 1:** Compute $a_1^p, a_2^p, \ldots, a_n^p$ as $A_1, A_2, \ldots, A_n$, respectively.

**STEP 2:** $S = A_n$

**STEP 3:** Loop $i$ FROM $n - 1$ TO 1 STEP -1;

   $S = A_i + (1 - A_i) \times S$

   NEXT $i$

**STEP 4:** $S = S^{1/p}$

The number of arithmetic operations in the above algorithm is $(2n - 2)$ additions and $(n - 1)$ multiplication and $(n + 1)$ power operations, reducing the computational complexity to $O(n)$. Therefore, the computational complexity is a linear function of the number of arguments, which is quite sufficient for further implementation. For computation of $t$-norm, an initial step which is to complement all arguments $a_1, a_2, \ldots, a_n$ and a final step, which is to subtract $S$ from one should be added to the above algorithm.

According to the theory of approximate reasoning [1], each fuzzy rule of the form

$$\text{IF } U_1 \text{ is } B_1 \ \text{AND } U_2 \text{ is } B_2 \ \text{AND} \cdots \ \text{AND } U_r \text{ is } B_r, \ \text{THEN } V \text{ is } D$$

(7)

can be translated into a canonical proposition of the form

$$(U_1, U_2, \ldots, U_r, V) \text{ is } R$$

(8)

where $R$ is a fuzzy relation defined on the Cartesian product universe $X_1 \times X_2 \times \cdots \times X_r \times Y$.

According to the analysis presented in Section II-A, it is suitable to use $t$-norm operators for defining conjunctions in the antecedent of the multi-input rule. Furthermore, modeling of implication relation-based fuzzy logic is not unique. Two extreme paradigms for forming the implication relation are conjunctive method and disjunctive method. Under conjunctive implication, the fuzzy relation $R$ is simply the conjunction of the antecedent and the consequent spaces. Therefore

$$R_c(x_1, x_2, \ldots, x_r, y) = T(T'(B_1(x_1), B_2(x_2), \ldots, B_r(x_r)), D(y))$$

(9)

in which $R_c(x_1, x_2, \ldots, x_r, y), B_i(x_i),$ and $D(y)$ are the membership degrees of $R_c, B_i,$ and $D$ is $t$-norm operator (with parameter $p$) for rule implication and $T'$ is $t$-norm operator (with parameter $q$) for rule antecedent aggregation. On the other side, disjunctive approach is obtained directly by generalizing the material implication defined in classical set theory as $B \rightarrow D \equiv \overline{B} \cup D$.

Therefore, we have

$$R_d(x_1, x_2, \ldots, x_r, y) = S(S'(\overline{B_1(x_1)}, \overline{B_2(x_2)}, \ldots, \overline{B_r(x_r)}), D(y))$$

(10)

in which $S$ and $S'$ are $t$-conorm operators with parameters $p$ and $q$, respectively.

The selection of the rule aggregation function depends on the selection of the implication function for individual rules [10]. For conjunctive implication (9) the whole combination of the rules should be as a disjunction (union) operator. In other words, the $\text{ALSO}$ connective should be an $\text{OR}$ operator; a $t$-conorm

$$R_M(x_1, x_2, \ldots, x_r, y) = S(R_1(x_1, x_2, \ldots, x_r, y), R_2(x_1, x_2, \ldots, x_r, y), \cdots, R_m(x_1, x_2, \ldots, x_r, y))$$

(11)

This method of aggregation coincides with the approximation reasoning approach introduced originally by Mamdani [14]. On the other hand, if each basic proposition is regarded as "$[n(U_j \text{ is } B_j)] \cup [V \text{ is } D_j]$," which is the disjunction implication approach, then the knowledge "$(U_1, U_2, \ldots, U_r, V)$ is $R'$ should be considered as a conjunction (intersection) of the
rules. In other words, the ALSo connective is AND operator; a t-norm
\[ R_T(x_1, x_2, \ldots, x_r, y) = T(R_{dt_1}(x_1, x_2, \ldots, x_r, y), R_{dt_2}(x_1, x_2, \ldots, x_r, y), \ldots, R_{dt_n}(x_1, x_2, \ldots, x_r, y)), \] 
(12)
This method is called formal logical reasoning.
Considering single-input single-output (SISO) system and given the relationship “is” and the information that \( U \) equals to a fuzzy set \( A \), the problem of interest becomes that of finding the fuzzy value for \( V \). From the mathematical point of view, this can be seen as solving the equation. From a logical point of view, this can be seen as a generalized form of modus ponens (GMP). According to Zadeh’s compositional rule of inference (CRI) [1], the membership function of the output is defined as
\[ F(y) = \vee_x [G(x; y)] = \vee_x [T(A(x), R(x; y))] = \max_x [T(A(x), R(x; y))], \]
(13)
in which \( \vee_x \) means the maximum for all values of \( x \).
Two different approaches defining the relation \( R \)—Mamdani’s and formal logical reasoning—lead to two formulations for obtaining the reasoning solution [10].

**Mamdani’s Approximation Reasoning:**
\[ F_M(y) = \vee_x [T[A(x), S[T(B_1(x), D_1(y))], T(B_2(x), D_2(y)), \ldots, T(B_n(x), D_n(y))]], \]
(14)
**Formal Logical Reasoning:**
\[ F_L(y) = \vee_x [T[A(x), T[S[(1 - B_1(x)), D_1(y)], S[(1 - B_2(x)), D_2(y)], \ldots, S[(1 - B_n(x)), D_n(y))]], \]
(15)
In general, t-norm and t-conorm classes do not have the property of distributivity. Therefore, for a fuzzy input \( A(x) \), we cannot distribute the t-norm operation among the t-conorms in (14) and (15). This means that for the case of fuzzy input, we have to first combine all rules to get the rule set relation \( R \) and then compose it with the fuzzy input. This is called FATI and, regarding the computation and memory considerations, this method is time and memory consuming and has practical limitations.

Fortunately, there is a more efficient method if the input of the reasoning is crisp, which is the case in most of fuzzy modeling and control applications. If the input \( x^* \) is crisp, then the input fuzzy set \( A \) is interpreted as a fuzzy singleton with membership function
\[ A^*(x) = \begin{cases} 1, & \text{if } x = x^* \\ 0, & \text{if } x \neq x^* \end{cases}. \]
(16)
In this case, by proving the distributivity property for any type of t-norm and t-conorm family, we illustrate that it is possible to fire each single rule first and calculate individual fuzzy output \( F_i \) and finally aggregate fuzzy output of all rules to obtain the inferred fuzzy output \( F(y) \). This method, FITA, is computationally more efficient than FATI.

**Theorem 3 [10]:** If \( A(x) \) is a singleton and \( M \) and \( N \) are fuzzy relations defined on \( X \times Y \), then the following distributivity relation is correct for t-norm \( T \) and t-conorm \( S \):
\[ T[A^*(x), S[M(x, y), N(x, y)]] = S[T[A^*(x), M(x, y)]] T[A^*(x), N(x, y)]. \]
(17)

**Theorem 4 [10]:** For crisp input, the FITA and FATI methods always give the same fuzzy output.
For MISO systems, the antecedent of each rule \( i \) is a conjunction of \( r \) fuzzy sets \( B_1^i, B_2^i, \ldots, B_r^i \). For crisp input \( x^* = (x_1^*, x_2^*, \ldots, x_r^*) \), the rule firing step for the rule \( i \) is to compute
\[ \tau_i(x^*) = T(B_1^i(x_1^*), B_2^i(x_2^*), \ldots, B_r^i(x_r^*)) \]
(18)
\( \tau_i \) is called the degree of firing (DOF) of rule \( i \). The fuzzy output membership function is
\[ F_M(y) = S[T(\tau_1(x^*), D_1(y)), T(\tau_2(x^*), D_2(y)), \ldots, T(\tau_n(x^*), D_n(y))]. \]
(19)
\[ F_L(y) = 1 - S[T(\tau_1(x^*), D_1(y)), T(\tau_2(x^*), D_2(y)), \ldots, T(\tau_n(x^*), D_n(y))]. \]
(20)

It is essential to mention that in the case of crisp input, we introduced a uniform formulation for both Mamdani’s approach and the logical approach. Therefore, unlike Yager and Filev’s approach, we are not concerned with the problem of aggregating the inferred output fuzzy set by such an operator that would be a compromise between the aggregating operators used in two extremes [10].

One of the most important issues in the reasoning process, especially in fuzzy modeling and control, is the problem of selection of a crisp value \( y^* \) based on the output fuzzy set \( E(y) \). Yager and Filev [15] suggest a general defuzzification method, based on the probabilistic nature of the selection process among the values of a fuzzy set called the basic defuzzification distribution (BADD) method
\[ y^* = \int_{y_m}^{y_M} y E^a(y) \, dy y^a \]
(22)
where the real interval \( Y = [y_m, y_M] \) is the universe of discourse of the output. As we can see, the BADD method is essentially a family of defuzzification methods parameterized by parameter \( \alpha \). By varying \( \alpha \) continuously in the real interval, it is possible to have more appropriate mappings from the fuzzy set to the crisp value depending on the system behavior.
As a result of different steps of the proposed reasoning process, four parameters $q$ (the parameter of $t$-norm operator for rule antecedent connection $T^*$), $p$ (the parameter of $t$-norm and $t$-conorm operators for rule implication and rule aggregation $T$ and $S$), $\beta$ (the compromise parameter of two inference approaches), and the defuzzification parameter are introduced whose variation will cause a continuous range of variation for reasoning mechanism. There is no need then to select the reasoning mechanism a priori, as the above parameters adjust the inference mechanism for each system based on its input–output data.

III. SYSTEM IDENTIFICATION

A. Fuzzy Structure Identification

Structure identification of fuzzy systems is possible by constructing enough rules with the appropriate input and output membership functions. This goal can be divided into two separate phases as: 1) rule generation and 2) input selection and membership assignment.

1) Rule Generation—Fuzzy Clustering: An intuitive approach to the objective rule generation is based upon clustering of input–output data, which has been suggested by several researchers [6], [16]. One simple and applicable idea, especially for systems with large number of input variables, was suggested by Sugeno and Yasukawa [5]. In this approach, we first cluster only the output space, which, as explained before, can be always considered as a single-dimensional space in fuzzy models. The fuzzy partition of the input space will be specified at the next step by generating the projection of the output clustered space on each input variable separately. Using this method, the rule generation step could be separated from the input selection step, as will be discussed in sequel.

The idea of fuzzy clustering is to divide the output data into fuzzy partitions that overlap with each other. Therefore, the containment of each data to each cluster is defined by a membership grade in $[0, 1]$. In formal words, clustering in unlabeled data $X = \{x_1, x_2, \ldots, x_N\} \subset \mathcal{R}^h$, where $N$ is the number of data vectors and $h$, the dimension of each data vector, is the assignment of $c$ number of partition labels to the vectors in $X$. $c$-partition of $X$ are sets of $(c \times N)$ membership values $\{u_{ik}\}$ that can be conveniently arrayed as a $(c \times N)$ matrix $U = [u_{ik}]$. The problem of fuzzy clustering is to find the optimum membership matrix $U$. The most widely used objective function for fuzzy clustering in $X$ is the weighted within groups sum of squared-errors objective function $J_m$, which is used to define the constrained optimization problem [17]

$$\min_{(U,V)} \left\{ J_m(U, V; X) = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m |x_i - v_i|^2 \right\} \quad (23)$$

where

$$U \in M_{c \times n} = \left\{ U \in \mathcal{R}^{c \times N} \left| \begin{array}{c}
0 \leq u_{ik} \leq 1 \quad \forall i, \forall k \\
0 < \sum_{k=1}^{N} u_{ik} < n \quad \forall i \\
0 < \sum_{i=1}^{c} u_{ik} = 1 \quad \forall k
\end{array} \right. \right\} \quad (24)$$

and

$V = \{v_1, v_2, \ldots, v_c\}$ is vector of (unknown) cluster centers (prototypes) and $|x|^A = \sqrt{\mathbf{x}^T A \mathbf{x}}$ is any inner product norm. $A$ is an $h \times h$ positive definite matrix that specifies the shape of the clusters. The matrix $A$ is usually selected to be as the identity matrix. This leads to the definition of Euclidean distance and, consequently, to the spherical clusters.

Fuzzy partitions are carried out by the FCM algorithm through an iterative optimization of (23) according to the following steps [18]:

**STEP 1:** CHOOSE

number of clusters ($c$), weighting exponent ($m$), iteration limit ($iter$), termination criterion ($\varepsilon > 0$), norm for $J_m$ ($|x_i - v_i|^A$), norm for error $= |V_t - V_{t-1}|$.

**STEP 2:** GUESS

initial position of cluster centers $V_0 = \{v_{0,1}, v_{0,2}, \ldots, v_{0,c}\} \subset \mathcal{R}^h$.

**STEP 3:** ITERATE

FOR $t = 1$ to $iter$

CALCULATE

$$u_{ik, t} = \left[ \sum_{j=1}^{c} \left( \frac{|x_i - u_{ik, t-1}|^A}{|x_i - u_{ij, t-1}|^A} \right)^{2/(m-1)} \right]^{-1} \quad (25)$$

CALCULATE

$$v_{i, t} = \frac{\sum_{k=1}^{c} (u_{ik, t})^m x_k}{\sum_{k=1}^{c} (u_{ik, t})^m} \quad (26)$$

IF error $= |V_t - V_{t-1}| \leq \varepsilon$, THEN stop and put $(U_f, V_f) = (U_t, V_t)$.

NEXT $t$.

The FCM clustering algorithm suffers from three major difficulties that are usually treated through heuristics for a specific problem at hand [18].

1) It is not always possible to assign the number of clusters as a priori. It is required to obtain a cluster validity criterion in order to determine the optimal number of clusters presented in the data.

2) No theoretical basis for an optimal choice for weighting exponent ($m$) has emerged to date.

3) Equations (25) and (26) are necessary conditions for local extrema of $J_m$, starting from an initial guess of $V_0$: but different choices of initial $V_0$ might lead to different local extrema. Therefore, the knowledge of proper initial location of cluster centers is required. This knowledge is not necessarily available a priori.

In order to complete the systematic methodology of fuzzy system identification and modeling, it is required to suggest generic solutions for the above problems. In [19], a theoretical basis for the first two problems is proposed and for the initial value problem a hybrid approach is introduced that produces a more efficient strategy than other solutions. Moreover, we address the classification process in fuzzy modeling, which remained unattained in the current literature.
1) Cluster Validity—Specification of the Number of Clusters: As a prerequisite for FCM algorithms, it is necessary to assign the number of underlying partitions that appear in the data set. In an attempt to derive a generic criterion for assignment of number of clusters from a theoretical point of view, we perform a proper generalization of scattering criteria that are mainly applied as suitable tools for expressing the compactness of and separation between the hard clusters [20] and introduce the following generalized fuzzy scatter matrices [19].

**Fuzzy within-cluster scatter matrix:**

\[
S_W = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^m (x_k - \bar{v}_i)(x_k - \bar{v}_i)^T.
\]  

(27)

**Fuzzy between-cluster scatter matrix:**

\[
S_B = \sum_{i=1}^{c} \left( \sum_{k=1}^{N} (u_{ik})^m \right) (\bar{v}_i - \bar{v})(\bar{v}_i - \bar{v})^T.
\]  

(28)

where the fuzzy total mean vector \(\bar{v}\) is a weighted mean of data considering their membership to each of the clusters in fuzzy partition, defined as

\[
\bar{v} = \frac{1}{c} \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^m x_k
\]  

(29)

and \(\bar{v}_i\) is the fuzzy cluster centers (prototypes)

\[
\bar{v}_i = \frac{1}{N} \sum_{k=1}^{N} (u_{ik})^m x_k.
\]  

(30)

\(S_B\) represents the separation between the fuzzy clusters and \(S_W\) is an expressive index for the compactness of fuzzy clusters. Hence, for obtaining the best clusters, we minimize \(\text{tr}(S_W)\) to increase the compactness of clusters and maximize \(\text{tr}(S_B)\) to increase the separation between clusters. In other words, we minimize

\[
s_{cs}(U; V; X) = \text{tr}(S_W) - \text{tr}(S_B) = \text{tr}(S_W - S_B)
\]  

\[
= \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m (||x_k - \bar{v}_i||^2 - ||\bar{v}_i - \bar{v}||^2)
\]  

(31)

where, \(\text{tr}(\cdot)\) is the trace of a matrix. It should be mentioned that unlike the Fukuyama–Sugeno index [21], in the proposed index, \(\bar{v}\) is the fuzzy extension of total mean vector. The difference is significant for large values of weighting exponent \(m\) in the FCM algorithm [19]. The effectiveness of this index to specify the best number of clusters for fuzzy clustering was justified by applying it to different examples mentioned in the related literature [19].

2) Selection of Weighting Exponent \((m)\) in Fuzzy Clustering Algorithm: Another parameter whose value should be decided in fuzzy clustering is the weighting exponent \(m\) in (25) and (26). In general, the weighting exponent controls the extent of membership sharing between fuzzy clusters in the data set. Therefore, in the range of \((1, \infty)\), the larger \(m\) is the “fuzzier” are the membership assignments to each data point. This parameter must be chosen to actually implement the FCM algorithm a priori. To date, no theoretical basis for an optimal choice of \(m\) has been suggested. We introduce a guideline for selection of \(m\) considering the cluster validity criterion \(s_{cs}\) derived in the previous section. First we examine the behavior of \(s_{cs}\) as \(m\) approaches its extremes, one, and infinity, through the following theorem.

**Theorem 5 [18], [19]:** In order to have a reliable index for cluster validity \(s_{cs}\), the weighting exponent \(m\) should be far enough from both of its limits, one, and infinity.

Now, in order to specify this condition, we should clearly identify the limits. The limit one is definitely clear, but what is infinity for \(m\)? This mainly depends on the data at hand and the question is, “Is there any index to indicate when \(m\) is about to become large?” The answer is “yes,” in accordance with our strategy, which is discussed next. We define a new scatter matrix, the fuzzy total scatter matrix \(S_T\) as the sum of fuzzy within-cluster and fuzzy between-cluster scatter matrices

\[
S_T = S_W + S_B = \sum_{k=1}^{N} \left( \sum_{i=1}^{c} (u_{ik})^m \right) (x_k - \bar{v})^T.
\]  

(32)

**Theorem 6 [19]:** The trace of \(S_T\) decreases monotonically from a constant value \(K\) to zero, as \(m\) varies from one to infinity. \(K\) depends only on the data set

\[
K = \text{tr} \left( \sum_{k=1}^{N} \left[ \left( x_k - \frac{1}{N} \sum_{k=1}^{N} x_k \right) \right]^T \left( x_k - \frac{1}{N} \sum_{k=1}^{N} x_k \right) \right).
\]  

(33)

Therefore, for a data set to be clustered, an appropriate value for \(m\) is what holds \(\text{tr}(S_T)\) somewhere in the middle of its domain \([K, 0]\). Since \(\text{tr}(S_T)\) is a function of number of clusters \(c\) as well as \(m\), the process of choosing \(m\) and \(c\) should be performed by a few iterations. We start with a suitable selection of \(m\), then derive \(s_{cs}\) for several \(c\’s\) and find the optimum \(c\). Then we check if for this \(c\) and \(m\); \(\text{tr}(S_T)\) should be satisfactorily far from its limits, otherwise, we repeat the process by a new \(m\). The reader is referred to [19] for more details.

3) Initial Guess and Local Optimality in FCM Algorithm: The third problem in FCM clustering algorithm arises from the fact that this algorithm may produce only local minima or partially optimal points [22]. Therefore, different initial guesses for mean vectors \(\bar{v}_i\) may lead to different optimum results. This fact affects the cluster validity as well as cluster
In order to efficiently obtain a preference for initial locations of cluster prototypes, we implement an agglomerative hierarchical clustering (AHC) algorithm as an introductory procedure to find properly identified hard-cluster centers as the initial locations of cluster prototypes in the FCM algorithm. The specific method used in this research is Ward’s method [23]. Having an unlabeled object data set \( X = \{x_1, x_2, \ldots, x_N \} \), the basic algorithm for this method is as follows:

**STEP 1:** CHOOSE

- number of clusters \( c \); the matrix of dissimilarities \( D = [d_{ij}] \) as the following Euclidean-based distance:

\[
d_{ij} = d(X_i, X_j) = \sqrt{n_i + n_j} ||\bar{x}_i - \bar{x}_j||,
\]

where \( \bar{x}_i \) and \( \bar{x}_j \) are mean vectors of hard clusters \( X_i \) and \( X_j \), respectively, and \( n_i \) (\( n_j \)) is the number of data in the hard cluster \( X_i \) (\( X_j \)).

**STEP 2:** LOOP

- FOR \( t = N \) to \( c \) and \( X_i,N = x_t \)

  - i = 1, 2, \ldots, N

  - FIND the pair of distinct clusters that have the minimum \( d_{ij} \), say \( X_{i,t} \) and \( X_{j,t} \);

  - MERGE \( X_{i,t} \) and \( X_{j,t} \);

  - DELETE \( X_{j,t} \);

  - NEXT \( t \).

The result of the above process is \( c \) hard clusters for a given data, which would be a good start for the fuzzy clustering procedure. With this method, we can choose the initial prototypes without any a priori knowledge of the data. This approach is much more efficient than random searches among different initial guesses.

4) **Formation of Membership Functions—Classification Problem:** In the fuzzy modeling algorithm, the output sample data are assorted in several fuzzy clusters. In order to extend the assigned fuzzy clusters to the entire output space, one more step called the classification process is required.

So far this step has not been addressed in the literature. We should emphasize the difference between clustering and classification. In the clustering process, we make a suitable partitioning for the data set \( X \subset \mathcal{R}^h \), whereas in the classification procedure, every data point in the entire space \( \mathcal{R}^h \) is labeled. Therefore, the problem of membership function formation for the entire output space is a classification problem. Since classifier design is usually performed using labeled data, clustering a sample output data is a good tool in the design of appropriate classifiers for the entire output space.

A fuzzy version of the probabilistic classification method called K-nearest neighbor has been introduced by Keller et al. [24] that is summarized in the following algorithm:

**STEP 1:** DERIVE

- \( X = \{x_1, x_2, \ldots, x_n \} \subset \mathcal{R}^h \) of labeled data set with membership grade \( U = [u_{ij}] \).

For any \( x \) of unknown classification,

**STEP 2:** CHOOSE

- number of neighbors \( k \): 1 \( \leq \) \( k \) \( \leq \) \( n \); norm for the distance \( ||x - x_j|| \).

**STEP 3:** FIND

- \( k \)-nearest neighbors to \( x \) among \( X = \{x_1, x_2, \ldots, x_n \} \).

**STEP 4:** LOOP

- FOR \( i = 1 \) to \( c \)

  - CALCULATE the membership grade to \( x \) in class \( i \) as:

\[
u_i(x) = \frac{\sum_{j=1}^{k} u_{ij} ||x - x_j||^2/(m-1)^{m-1}}{\sum_{j=1}^{k} ||x - x_j||^2/(m-1)^{m-1}}
\]

- NEXT \( i \).

After assignment of the data in the entire space to the fuzzy partition, in order to obtain simple membership functions we approximate the classified data by trapezoidal functions in such a way that for each fuzzy cluster, convex points are picked up and a trapezoid is fitted to them [25].

The phase of input selection in system identification is to find the most dominant input variables that affect the output among a finite number of input candidates. Theoretically, this problem belongs to a more general field of data analysis, i.e., *dimension reduction*. In the analysis of multivariate data, it is common practice to look for the dimension reduction via linear combinations of the initial variables. The classical techniques, such as principal components [20], canonical correlation [20], and discriminant analysis [26], are examples of this approach. From a practical point of view, another type of dimension reduction is selecting a subset of the variables. The main advantage of this approach is that there is an actual reduction in the number of measured variables. In this way, we can avoid the interpretational difficulties that could arise in looking at linear combinations of very different kinds of variables. Although it is a common practice to check the weights of variables in a linear combination and to discard those that have “negligible” weights, this is neither always easy to do nor are negligible weights always guaranteed.

The problem of variable selection is also referred to as “feature selection,” especially in some areas such as pattern recognition and information processing. Three major techniques are suggested for selecting a subset from an initial set of features, i.e., multiple regression [27], discriminant analysis [28], and cluster analysis [29]. A good comparison of these techniques can be found in [30]. All the aforementioned methods are expressed in the context of statistical analysis, which, in most cases, benefits from the formal analytical background. However, in order to apply these techniques, many conditions such as normal distribution, adequate amount
of data, independence, etc. should be satisfied, which is rather crucial in real situations. There are some efforts of using informal techniques such as search method [31], genetic algorithms [32], and techniques based on fuzzy sets [33] and possibility theory [34] in the context of feature selection in order to relieve the restricted formal conditions of statistical approach.

In the context of feature selection, there is no distinction between input and output variables of the investigated system. However, the specific problem of “input selection” can be considered taking this distinction into account. For instance, in input selection problem, it is quite possible to consider the dependence of the output variable(s) to each input variable, separately. In this way, the complexity of the problem at hand would be reduced significantly. Following this more specific approach, in fuzzy modeling, three major ideas have been suggested for selecting significant input variables among all finite candidates. Sugeno and Yasukawa [5] propose a combinatorial approach in which all possible combinations of input candidates are considered. For each combination, they build two fuzzy models based on two separated sets of data and calculate a performance index called regularity criterion based on a method of analyzing two groups of data in an attempt to cause data independence in model formation [35]. A combination of input variables is chosen, which has minimum value of the performance index. For \( r_0 \) input candidates, the number of fuzzy models to be built and tested for input variable selection is \( r_0!/(r_0+1)/2 \). In another investigation, Takagi and Hayashi [36] propose a fuzzy reasoning neural network system to identify the significant input variables by eliminating each input candidate and checking a performance index. Those candidates that have less or no improvement effect on the performance index are considered as nonsignificant. Again, for \( r_0 \) input candidates, a possible \( r_0!(r_0+1)/2 \) neural nets should be trained with this technique. Building \( r_0!(r_0+1)/2 \) fuzzy or neural network models is quite time-consuming, especially for real systems with a large number of potential input variables. Moreover, in our fuzzy modeling methodology, we desire to separate the “input selection” stage from other stages, specifically because the inference mechanism is not fixed in the proposed methodology. As a matter of fact, unlike Sugeno–Yasukawa approach, we believe that the significance of each input variable in the system is a real property of the system itself and should not depend on a selection of inference method and, hence, the manner of interpreting the model of the system. The third method of input selection is suggested by Lin and Cunningham [37]. In their technique, for each input variable the input–output data are plotted and each sample point is fuzzified to a Gaussian membership function and then, for each sample point, a fuzzy rule is constructed. Next, for potential input values, the defuzzified outputs are derived from the set of rules using Sugeno’s heuristic reasoning formulation. As a result of this process, a fuzzy curve will be produced in the input–output plane. This procedure is repeated for other input variables, one at each time. Significant input variables are supposed to have a wider range for their fuzzy curves. Lin and Cunningham illustrated the validity of their method by several examples [38].

In our proposed fuzzy modeling for the selection of the significant input variables, we propose an approach that is consistent with the whole idea of fuzzy models. In fuzzy mod-
EMAMI et al.: DEVELOPMENT OF SYSTEMATIC METHODOLOGY OF FUZZY LOGIC MODELING

For construction of the input and output membership functions, the ideal case would be to partition the \((r + 1)\)-dimensional input–output space where \(r\) is the number of significant input variables. Due to limitation of clustering techniques and potential high dimensionality of the problem, this approach is not promising in most of the applications. One practical alternative is to derive the fuzzy partition of the output space from the data in order to obtain the required number of rules for expressing the system behavior, i.e., the output variation. The next step in this approach is to extract the input membership functions from the output partitions. After deriving the output space clusters and before discussing about the input membership assignment, we are able to implement the proposed strategy of selecting the significant input variables. The reason is that by output space clustering for each cluster (rule), the points whose corresponding output membership grades are equal to one are specified. No matter how the input membership functions are assigned, these points should have the same input membership grades (equal to one) in the corresponding rules, as illustrated in Fig. 3.

Based on the above reasoning, the following simple algorithm is proposed for the input selection stage:

**STEP 1:** GET the output–space membership matrix 
\[ u_{ik}, i = 1, 2, \ldots, n, k = 1, 2, \ldots, N, \text{ where } N \text{ is the number of rules (output clusters) and } n \text{ is the number of data.} \]

**STEP 2:** LOOP

FOR \( j = 1 \) to \( n \)

PUT \( \pi_j = 1 \)

FOR \( i = 1 \) to \( n \)

FIND \( \Gamma_{ij} = \{\text{set of inputs } x_j \text{ with } u_{ik} = 1, k = 1, 2, \ldots, N\} \)

FIND \( \Gamma_{ij} = \{\text{set of all } x_j\} \)

FIND \( \Gamma_{ij} = \{\text{set of all } x_j\} \)

CALCULATE

\[ \pi_j = \pi_j \times \left( \frac{\text{Max}(\Gamma_{ij}) - \text{Min}(\Gamma_{ij})}{\text{Max}(\Gamma_{ij}) - \text{Min}(\Gamma_{ij})} \right) \]  

(39)

NEXT \( i \)

NEXT \( j \)

**STEP 3:** COMPARiE the values of \( \pi_j, j = 1, 2, \ldots, n \)

**STEP 4:** REMOVE input variables with large value of \( \pi_j \),

One important point should be remarked regarding the above strategy and the following strategy of the input membership assignment as well. In some applications, there might be more than one input clusters corresponding to one output cluster, as illustrated in Fig. 4. In this case, the input variable having this characteristic is certainly significant and two (or more) membership functions should be defined for this variable using the following strategy. Therefore, in such cases the number of rules \( (n) \) is more than the number of output clusters \( (c = n) \), although for the above cases the same strategy can be applied considering all input clusters corresponding to one output cluster as separate clusters.

After selecting the significant input variables, suitable membership functions should be defined for them. One simple approach is to set the membership grade of each input datum equal to its corresponding output membership grade obtained from the output data clustering process [5]. Therefore, for each output datum, all the corresponding input variables will have the same membership grade. The problem with this technique is that the membership functions assigned in this way are not convex and further approximation is required to shape the convex membership functions. Besides, there is no reason for the input membership grades to be the same and equal to the output membership grade at each sample point.

In our proposed fuzzy modeling methodology, a new technique is suggested based on the previous discussion about input selection strategy. As it was discussed (Fig. 3), assuming that the membership functions are smooth enough, the only points that can claim to have the equal input and output membership grades are those that have the unit (or close to unit) membership grade. This was our basic assumption in identification of the significant input variables. We use this conclusion to construct the convex input membership functions. In Fig. 3, suppose that for each output cluster \( i \) \( (i = 1, 2, \ldots, n) \) there are several points on the axis of input variable \( x_j \) \( (j = 1, 2, \ldots, r) \) that have output membership

![Fig. 3. “Peak” points should be the same for input and output clusters.](image1)

![Fig. 4. One output cluster with two corresponding input clusters.](image2)
grades equal or close to one. These points lie between \( v_{ij}^1 \) and \( v_{ij}^2 \) \((v_{ij}^1 < v_{ij}^2)\). We define the “distance” of each point \( x_{jk} \)\((k = 1, 2, \ldots, N)\) located on the axis \( x_j \) to the line \( v_{ij}^1 v_{ij}^2 \) with the following function:

\[
\text{dist}(x_{jk}, v_{ij}^1 v_{ij}^2) = \begin{cases} 
    d(x_{jk}, v_{ij}^1) = v_{ij}^1 - x_{jk}, & \text{if } x_{jk} < v_{ij}^1 \\
    0, & \text{if } v_{ij}^1 \leq x_{jk} \leq v_{ij}^2 \\
    d(x_{jk}, v_{ij}^2) = x_{jk} - v_{ij}^2, & \text{if } x_{jk} > v_{ij}^2.
\end{cases}
\]

(40)

Now, for input data \( x_{jk} \)\((j = 1, 2, \ldots, r; k = 1, 2, \ldots, N)\), the membership grades \( u_{ik}^j \) corresponding to output cluster \( i \)\((i = 1, 2, \ldots, n)\) are formed such that those points which are “closer” to the line \( v_{ij}^1 v_{ij}^2 \) obtain higher membership grades. Obviously, those points that are between \( v_{ij}^1 \) and \( v_{ij}^2 \) are assigned to have a unit membership grade. We call this clustering procedure the line fuzzy clustering algorithm.

Although the introduced clustering concept can be generalized to the multidimensional space, having line or surface fuzzy clustering, in this paper, we stick to the one-dimensional case and postpone the general case to a further research. Analogous to the weighted within-group sum of squared errors in FCM algorithm for the line fuzzy clustering algorithm, the following objective function is defined:

\[
J_m(U^j; X_j) = \sum_{k=1}^{N} \sum_{i=1}^{n} (u_{ik}^j)^m \text{dist}(x_{jk}, v_{ij}^1 v_{ij}^2).
\]

(41)

It should be mentioned that in our application, for each input variable \( x_j \), the lines \( v_{ij}^1 v_{ij}^2 \)\((i = 1, 2, \ldots, n)\) are known and already specified from output clusters. Therefore, the problem of fuzzy clustering here is to find optimum membership grades \( u_{ik}^j \) such that \( J_m \) becomes minimum. Like FCM algorithm, in-line fuzzy clustering algorithm weighting exponent \( m \) specifies the degree of fuzziness of the clusters. Solutions to the above optimization problem are directly obtained by distinguishing between two cases when \( x_{jk} < v_{ij}^1 \) and \( x_{jk} > v_{ij}^2 \) and solving \( (d/dx_{jk}) (J_m) = 0 \) for each case. The final result is obtained as in (42), shown at the bottom of the page, for \( i = 1, 2, \ldots, n, j = 1, 2, \ldots, r \), and \( k = 1, 2, \ldots, N \), where \( n \) is the number of rules, \( r \) is the number of significant input variables, and \( N \) is the number of data.

The above membership formation procedure is repeated for all selected input variables. The following algorithm summarizes the process of input membership formation:

**LOOP:**

FOR \( j = 1 \) to \( r \)

FOR \( i = 1 \to n \)

FIND \( \Gamma_{ij} \) = \{set of inputs \( x_j \) which correspond to output membership grade \( u_i = 1 \}

PUT \( u_{ij}^1 = \text{Min} (\Gamma_{ij}) \) AND \( u_{ij}^2 = \text{Max} (\Gamma_{ij}) \)

NEXT \( i \)

CALCULATE \( u_{ij}^j \) from equation (42) for \( i = 1, 2, \ldots, n \) and \( k = 1, 2, \ldots, N \)

NEXT \( j \)

The stage of parameter identification consists of two steps. First, we specify the “best” inference mechanism for the system represented by its input–output data. This is a distinguishing feature of our fuzzy modeling methodology. The inference mechanism (parameters \( p, q, \alpha, \beta \)) are optimized according to the data such that it provides the most adequate mechanism for the system under investigation, which is not necessarily one of the known extreme inference mechanisms.

In order to find the optimum value of the inference parameters, we model the problem as the following boundary nonlinear optimization problem:

Derive the set of parameters \( [p, q, \alpha, \beta] \) such that

\[
\text{PI}(p, q, \alpha, \beta) = \sum_{i=1}^{N} \frac{(y_i - \hat{y}_i)^2}{N} \]

(43)

becomes minimum subject to the following boundaries:

\[
0 < p, q < \infty \quad \& \quad 1 < \alpha < \infty \quad \& \quad 0 < \beta < 1
\]

(44)

where \( N \) is the number of data, \( y_i \) is the \( i \)th actual output, and \( \hat{y}_i \) is the \( i \)th model output. It should be mentioned that at this step, the input and output membership functions that are obtained from the structure identification are used. In order to solve the above optimization problem, function constr of MATLAB has been used, which is based on the sequential quadratic programming (SQP) method [39].

As mentioned before, we have already found the input–output membership functions in the structure identification stage. However, it is better to tune the parameters as we do in the ordinary system identification methods. In our methodology, trapezoidal fuzzy sets are used as approximations to convex fuzzy sets

\[
\text{trapezoid} (y, y_a, y_b, y_c, y_d) = \max \left( \min \left( \frac{y - y_a}{y_b - y_a}, 1 \right), \min \left( \frac{y_d - y}{y_d - y_c}, 0 \right) \right).
\]

(45)
In order to adjust the trapezoid function parameters \( (y_b, y_m, y_s, y_s) \), we apply Sugeno–Yasukawa’s tuning algorithm [5] with one modification that a variable adjustment value is used at each tuning step. This modification makes the tuning procedure more efficient. Besides, unlike Sugeno–Yasukawa’s method, we adjust the parameters of both input and output membership functions. The following algorithm summarizes the tuning process:

**STEP 1:** CHOOSE
the initial value of adjustment for input \( (\eta) \) and output \( (\delta) \) membership functions (5–10% of the range of the universe of discourse would be good start), number of iterations \( (\text{iter}) \), number of adjustment changes \( (\text{div}) \).

**STEP 2:** ITERATE
FOR \( i = 1 \) to \( \text{iter} \)

**STEP 2.1:** for \( i = 1, 2, \ldots, n \) (number of rules)
and \( j = 1, 2, \ldots, r \) (number of input variables), and \( k = 1, 2, 3, 4 \):

SET \( \theta^k_j \) as the \( k \)th input membership parameter
of the \( j \)th fuzzy set in the \( i \)th rule.

SET \( \eta = \theta^k \).

**STEP 2.2:** LOOP
FOR \( m = 1 \) to \( \text{div} \)

SET \( \eta = \epsilon/m \).

CALCULATE \( \theta^k_j + \epsilon \) and \( \theta^k_j \).

IF \( k = 2, 3, 4 \) AND \( \theta^k_j > \theta^k \)
THEN \( \theta^k_j = \theta^k \); ELSE \( \theta^k_j = \theta^k \) + \( \epsilon \).

IF \( k = 1, 2, 3 \) AND \( \theta^k_j < \theta^k \)
THEN \( \theta^k_j = \theta^k \); ELSE \( \theta^k_j = \theta^k \) + \( \epsilon \).

CHOOSE the parameter, which shows the least \( \text{PI} \) among \( (\theta^k, \theta^k_j, \theta^k_j) \) and REPLACE \( \theta^k_j \)
with it.

IF the new \( \text{PI} \) is less than the old \( \text{PI} \) THEN
break LOOP \( m \).

NEXT \( m \)

**STEP 2.3:** for \( i = 1, 2, \ldots, n \) (number of rules)
and \( k = 1, 2, 3, 4 \):

SET \( \xi^k \) as the \( k \)th output membership
parameter in the \( i \)th rule.

SET \( \xi = \xi \).

**STEP 2.4:** REPEAT **STEP 2.2** for output membership parameters \( \xi^k \).

NEXT \( i \).

IV. Fuzzy System Identification Algorithm

Summing up the subject of this section, we obtain the flow chart shown in Fig. 5.

A. Example 1

In [5], Sugeno and Yasukawa derive the fuzzy model of a nonlinear static system with two input variables \( x_1 \) and \( x_2 \) and a single output \( y \) as follows:

\[
y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \leq x_1, x_2 \leq 5.
\] (46)
having the identified value of $m$ for the system as $m = 2.92$. Obviously, the optimum number of clusters is $c = 8$. It should be mentioned that in all the above processes, for the FCM algorithm, the HAC algorithm is applied to derive the initial locations of cluster centers for FCM algorithm. This is also critical for deriving the appropriate $m$ and $c$. In [19], we illustrated that in this example, choosing random cluster centers for FCM algorithm is not reliable for assignment of optimum number of clusters. Furthermore, classification process makes the distribution of the output clusters more uniform in the entire output space as is obvious from Figs. 8 and 9. After deriving the output fuzzy clusters and performing the classification for the entire output space, significant input variables should be identified and their membership functions are to be assigned. Unlike the Sugeno–Yasukawa’s approach of dividing the data into two sets and using a time-demanding combinatorial strategy, we apply the straightforward strategy described in Section III-A.2. The values of $\pi$ (45) for the four input candidates are

$$
\pi_1 = 0.60 \times 10^{-4}; \quad \pi_2 = 0.69 \times 10^{-4},
$$

$$
\pi_3 = 5.72 \times 10^{-4}; \quad \pi_4 = 4.60 \times 10^{-4}.
$$

Clearly, the first two input variables ($x_1$ and $x_2$) have the minimum $\pi$'s, which are less than those for the other two variables in one order of magnitude. Input membership functions are also assigned through fuzzy line clustering as explained in Section III-A.2. After approximating the input and output fuzzy clusters by suitable trapezoidal functions, the first-step rough fuzzy model of the system is derived and shown in Fig. 10. Note that up until now, no inference mechanism is required to derive the input–output clusters. The next step is to select and tune the set of parameters of the fuzzy model, which consists of the inference parameters ($p$, $q$, $\alpha$, and $\beta$) and
Fig. 11. Final fuzzy model of the nonlinear system after parameter identification.

Fig. 12. Identification of $\frac{1}{10^9}$ for gas furnace process.

Fig. 13. Specification of $\frac{1}{99}$ for gas furnace process.

Fig. 14. Final fuzzy model of gas furnace process after parameter identification.

By optimizing the inference parameters, the fuzzy model performance index (51) is $\text{PI} = 0.171$. The second step of parameter identification is to adjust the input and output membership function parameters. At the first step of parameter identification, the optimum inference parameters are identified as explained in Section III-B.1. The optimum values are

$$p = 19.9939; \quad q = 0.4556; \quad \alpha = 5.3715; \quad \beta = 0.0255.$$  \hspace{1cm} (48)

By optimizing the inference parameters, the fuzzy model performance index starts from $\text{PI} = 0.318$ and after 20 iterations, it reaches to $\text{PI} = 0.010$ for this example, the proposed position-type fuzzy model shows a better performance.
The whole effort is to achieve a systematic and objective technique of fuzzy modeling, which is reliable for a wide range of applications supported by its theoretical background. The validity of the methodology was examined through two examples and a comparison study was made with Sugeno–Yasukawa fuzzy modeling technique. The results are quite superior.

V. CONCLUSIONS

We proposed a systematic approach of fuzzy modeling and system identification. The methodology considers the inference mechanism as a computation ground for fuzzy systems as well as the structure and parameter identification of the fuzzy system. For reasoning process, a unified parameterized formulation is applied by which the suitable inference mechanism is adjusted for the system based on its input–output data. Therefore, no selection of inference mechanism is required a priori and no restriction on any steps of reasoning is necessitated.

For structure identification of the fuzzy system, suitable indexes are applied for identifying number of rules and level of fuzziness of the fuzzy system based on fuzzy c-means clustering technique. Moreover, we used fuzzy classification techniques to extend the clusters obtained from the sample data to the entire space. Significant input variables are identified immediately by a new strategy as a result of output data clustering. Considering only the output space to identify the structure of the fuzzy system gives us the advantage of simplicity and applicability. However, by introducing fuzzy line clustering problem, we suggest a more appropriate methodology to specify the input-space partition from the output partition than simple projection of the output clusters onto the input space. For parameter identification, an efficient algorithm for tuning membership parameters of input and output fuzzy sets was introduced.

REFERENCES

Mohammad R. Emami received the B.A.Sc. and M.A.Sc. degrees in mechanical engineering from Sharif University of Technology, Tehran, Iran, in 1987 and 1990, respectively, and the Ph.D. degree in robotics and mechatronics from the University of Toronto, Toronto, ON, Canada, in 1997. He was awarded a scholarship by the Ministry of Culture and Higher Education of Iran in 1992. From 1986 to 1996, he was with the Research Industries Institute of Tehran, Iran, and in 1996 became a Research Associate of the Robotics and Mechatronics Laboratory at the University of Toronto. His current research interests include the application of nontraditional approaches such as fuzzy logic, neural networks, and genetic algorithms to modeling and control of complex and ill-defined systems.

Andrew A. Goldenberg (S’73–M’76–SM’87–F’96) received the B.A.Sc. and M.A.Sc. degrees from the Technion-Israel Institute of Technology, Haifa, Israel, in 1969 and 1972, respectively, and the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 1976, all in electrical engineering. From 1975 to 1981, he was employed by Spar Aerospace Ltd., Toronto, where he worked mainly on control, analysis, and design of the space shuttle remote manipulator system and satellite controls. From 1981 to 1982 he was an Assistant Professor of electrical engineering and from 1982 to 1987 he was an Associate Professor of mechanical engineering at the University of Toronto. Since 1987 he has been a Professor of mechanical engineering at the University of Toronto where he holds cross appointments in the Department of Electrical Engineering and the Institute of Biomedical Engineering. He founded both the Robotics and Automation Laboratory (1982) and the Mechatronics Laboratory (1996) at the University of Toronto. His current research interests are in the field of robotics and industrial automation, kinematics, control, and dynamics of robots and dexterous end effectors. He is the founder and President of Engineering Services Inc., a high-technology company involved in the development of prototype robotic-based automation and products. He is a consultant to the Department of National Defence, Ottawa, Canada, Ontario Ministry of Agriculture and Food, Ottawa, Canada, RCMP, Ottawa, Canada, Consumers Gas, Toronto, Canada, Vulkan-Vulcap Inc., Toronto, Canada, DCIEM, Toronto, Canada, and more.

Dr. Goldenberg is an Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION. He is a member of the American Society of Mechanical Engineers and the Professional Engineers of Ontario.