Discrete-Time Finite Fuzzy Markov Chains Realized through Supervised Learning Stochastic Fuzzy Discrete Event Systems

Hao Ying, *Fellow, IEEE,* and Feng Lin, *Fellow, IEEE*

***Abstract*—The discrete-time finite Markov chains constitute a class of stochastic models employed across various industries for over 100 years. However, the binary nature of the states and state transitions renders this modeling methodology unsuitable for many practical systems, such as those in biomedicine, which are characterized by intrinsic vagueness and imprecision in states and events that cause state transitions. To address this fundamental limitation, we have extended in this paper Markov chains to fuzzy Markov chains capable of handling fuzzy states and fuzzy events. This innovative and significant advancement is founded on the theory of Stochastic Fuzzy Discrete Event Systems (SFDES) and the supervised learning algorithm for Fuzzy Discrete Event Systems (FDES), recently published by the authors. Our major technical contribution lies in mathematically generalizing a traditional Markov chain with N states to a fuzzy Markov chain with N fuzzy states, which is represented by an SFDES consisting of N^2 FDES. Each FDES has its own $N \times N$ event transition matrix that is automatically learned by the aforementioned learning algorithm. Crucially, the fuzzy Markov chain fully preserves the stochastic characteristics defined by the transition probability matrix of the binary Markov chain, ensuring identical stochastic behaviors. A defuzzifier is used to yield crisp model output. The structurally more complex fuzzy Markov chain encompasses its binary counterpart as a special case and degenerates into it when fuzzy states degenerate into binary states. A simulation example is provided to illustrate the systematic design procedure and demonstrate the higher prediction accuracy of the fuzzy Markov chain over its binary counterpart. Capable of effectively representing and processing vague and imprecise state and event information, fuzzy Markov chains hold a key advantage and have the potential to solve real-world stochastic problems beyond the reach of conventional Markov chains, especially in biomedicine.**

***Index Terms*—stochastic modeling, Markov chains, fuzzy Markov chains, stochastic fuzzy discrete event systems, fuzzy automaton, supervised learning**

I. Introduction

A Markov chain (or Markov process) is a stochastic model characterizing a system in which a sequence of events takes place that will cause system state to change in a random manner. Markov chains have achieved widespread success in countless practical applications since their first appearance in the turn of the last century. The Markov chains involved in this study are of the discrete-time type and have a finite number of states, which are frequently used and studied. They will simply be referred as the Markov chains from now on unless otherwise indicated. A Markov chain generates a random state sequence as its output, forming a time series. The state at time t_0 leads to the state at t_1, which subsequently leads to the state at t_2, and so forth. Therefore, the model can be viewed as a process that iteratively employs the current state as the pre-event state to generate a new state, which is the post-event state, when an event occurs.

A state of a Markov chain represents a M-dimensional hypercube that is formed by M intervals of M random variables (one interval for one variable). The states are mutually exclusive because the hypercubes do not overlap. The states hence are binary in that the system can be only in one state at any moment of time. Changes in the values of the M variables do not necessarily cause the system to change state. The system state remains the same as long as values of the M variables stay within the same M intervals. The system transfers from one state to another abruptly when value of at least one variable falls into a different interval.

The binary nature of the states and state transitions makes the Markov chains unsuitable for modeling systems whose states and events are intrinsically vague and imprecise. A case in point is biomedical systems. As an example, description of a hospitalized patient's clinical state necessitates the use of ambiguous and subjective terms like "Stable State," "Fair State," "Serious State," and "Improved State." Furthermore, a patient can be in multiple states (e.g., "Serious State" and "Stable State") simultaneously with different extents. Similarly, the notation of an event causing a system to change from one binary state to another cannot effectively handle the reality in biomedicine. For instance, disease treatment (e.g., surgery) is an event that may transfer a patient to more than one state concurrently with varying degrees (e.g., patient is more in "Improved State" and less in "Fair State" at the same time).

Initiatives to incorporate fuzzy sets theory to mitigate the limitations of traditional Markov chains commenced in the early 1980s [1]. One approach entails replacing probabilities in a transition probability matrix with fuzzy numbers or fuzzy sets, which can be viewed as subjective probability estimates expressed in vague terms (e.g., a "moderate" chance). The fuzzy numbers and sets, which can be either type-1 or type-2, are used to model stochastic uncertainties that are actually treated as possibilities (e.g., [1], [2], [3], [4], [5], [6], [7]). The transition probability matrix is then used as a fuzzy relation to infer a post-event state from a pre-event state. Treating a

Hao Ying is with the Department of Electrical and Computer Engineering, Wayne State University, 5050 Anthony Wayne Dr., Detroit, MI 48202, USA. (corresponding author; e-mail: hao.ying@wayne.edu). Feng Lin is with the same department (e-mail: flin@wayne.edu).This work is supported in part by by the National Science Foundation of USA under grant 2146615.

transition probability matrix as a transition possibility matrix (i.e., fuzzy relation) alters the stochastic characteristics of the original problem, a transformation undesirable for many practical applications. Another approach involves expanding the binary states of Markov chains into fuzzy states, wherein state transition probabilities are considered as possibilities (e.g., [8], [9], [10], [11]). A recent survey on fuzzy Markov chains is available [12], in which a list of references is provided.

In this paper, we present a totally different approach to addressing the aforementioned fundamental limitation of binary Markov chains. We have established a novel rigorous mathematical framework for generalizing traditional Markov chains to fuzzy Markov chains. These fuzzy Markov chains are capable of: (1) capturing and representing ambiguous states as fuzzy states, and (2) representing and processing vague events. The creation of fuzzy Markov chains is possible only because we discovered in this study an intriguing connection between conventional Markov chains and a class of Fuzzy Discrete Event Systems (FDES) called Stochastic Fuzzy Discrete Event Systems (SFDES), which we recently developed [13].

What we found is that a fuzzy Markov chain can be represented by an SFDES, and importantly, the SFDES retains the stochastic characteristics defined by the transition probability matrix of the original binary Markov chain, which is assumed to be available or obtainable through measurements and calculations. A supervised learning algorithm that we previously developed can be employed to learn the event transition matrices of the SFDES. Consequently, the fuzzy Markov chain preserves the stochastic characteristics of the binary Markov chain, leading to identical stochastic behaviors.

We will prove mathematically that the fuzzy Markov chain encompasses the binary Markov chain as a special case and degenerates into it when fuzzy states reduce to interval states.

The ability of fuzzy Markov chains to handle fuzzy states and events provides substantial advantages over conventional Markov chains, resulting in improved models and more accurate predictions. These capabilities make fuzzy Markov chains well-suited for addressing practical stochastic problems characterized by ambiguous states and vague events, particularly in fields like biomedicine, where conventional Markov chains fall short.

Clearly, the fuzzy Markov chains are SFDES-based. It should be pointed out that the notion of a fuzzy Markov chain is never mentioned in our previous publications and thus is innovative with respect to our previous studies. SFDES is founded on the theory of FDES that we introduced in 2001 [14] to model systems involving vague states and events. The theory extends and complements the conventional discrete event systems theory that was originated in the 1980s for modeling a class of discrete-time systems whose system state change is a result of occurrences of a sequence of events [15]. It is important to note that while Markov chains and discrete event systems are both event-driven, they represent distinct modeling methodologies developed for different systems and with different objectives.

For better presentation, we need to provide an introduction to FDES first.

A. Introduction to Fuzzy Discrete Event Systems

The foundation of the FDES theory rests on the notations of "fuzzy state" and "fuzzy event." A fuzzy automaton, similar to its binary counterpart, mathematically models an FDES and is represented by

$$G = (\mathbf{Q}, \Sigma, \varphi, \mathbf{q_o}) \tag{1}$$

where system state $\mathbf{Q}$ is a vector represented by N individual fuzzy states, $\mathbf{q_o}$ is an initial fuzzy state vector, Σ is a set of fuzzy events, each of which is characterized by an $N \times N$ event transition matrix, and $\varphi : \mathbf{Q} \times \Sigma \rightarrow \mathbf{Q}$ is an event transition mapping. Memberships of the individual fuzzy states are in [0, 1], so are the elements of the event transition matrix. That means the fuzzy state and fuzzy event can have partial memberships, and binary state and crisp event are special cases of fuzzy state and fuzzy event, respectively (0 or 1 is a special case of [0, 1]). For any specific FDES, given a pre-event fuzzy state vector and an event transition matrix, the post-event fuzzy state vector can be computed using a fuzzy inference operation such as the widely-used max-product composition operation. System state before and after occurrence of an event is called the pre-event state and post-event state, respectively. The corresponding state vectors are referred as the pre-event fuzzy state vector and post-event fuzzy state vector, respectively.

For a concrete FDES and its operation, the reader is referred to the illustrative numerical FDES example in the Fuzzy Logic Toolbox of MATLAB version R2024a. This example, developed collaboratively with the first author of the present paper, is publicly accessible free of charge on MathWorks' website, along with its MATLAB code [**?**].

The theoretical framework of FDES has been expanded significantly in various directions. These extensions include state-based control [16], state-feedback control [17], supervisory control [18], [19], [20], [21], [22], decentralized control [23], [24], [25], online control [26], detectabilities [27], diagnosability [24], [28], [29], prognosis [30], [31], predictability [32], and opacity [33], [34]. Additionally, controllability of FDES has been studied [35]. Type-2 fuzzy sets have been used to expand the type-1-fuzzy-set-dominated FDES framework [36], and the notions of generalized FDES [35] and semi-discrete events with fuzzy logic [37] have been proposed. The modeling of FDES based on a generalized linguistic variable has been explored [38]. Most efforts in the literature focus on the deterministic FDES. Nevertheless, modeling of the nondeterministic FDES is investigated in [39] while modeling and control of the probabilistic FDES are addressed in [40]. A recent survey of the FDES literature is available [41].

The event transition matrix governs the state-to-state transition of an FDES and hence is crucial for determining properties of the FDES such as observability [42] and predictability [32]. It is worthy noting that the matrix's elements are possibilities. Thus, they are fundamentally different from the elements in a Markov chain's transition matrix, which are probabilities. Manually building the event transition matrix of a FDES for a specific application can be a difficult and daunting task. This motivated us to develop stochastic-gradient-descent-based online supervised learning algorithms to automatically learn the event transition matrix from pre- and post-event fuzzy state vector pairs under various conditions that may arise in practice [43], [44], [44], [45], [46], [47]. As one will see below, one of the algorithms will be utilized to learn fuzzy automata in an SFDES representing a fuzzy Markov chain.

B. Overview of the Paper

SFDES is a new class of nondeterminstic FDES and is substantially different from the probabilistic FDES that we put forward earlier. An SFDES consists of two or more FDES that are represented by fuzzy automata. Each of the fuzzy automata represents a fuzzy event with an event occurrence probability. In the presence of a pre-event fuzzy state vector, one does not know which of the multiple fuzzy events will occur and thus cannot know or determine the post-event fuzzy state vector beforehand. At any moment, which event will take place is random. We have developed two techniques to identify fuzzy automata of an SFDES under different conditions on pre- and post-event fuzzy state vector pairs [13][48].

SFDES represents one of the latest advancements in FDES theory. Originating independently and not influenced by Markov chain theory, this development holds intrinsic significance. Consequently, the establishment of a connection between SFDES and Markov chains becomes intriguing and noteworthy. Furthermore, leveraging this connection allows for the systematic expansion of Markov chains into fuzzy Markov chains with mathematical rigor. This marks a pivotal theoretical advancement and represents our most significant contribution.

The second significant theoretical development and our contribution lie in our mathematical proof that Markov chains are a special case of SFDES. More specifically, a Markov chain with an $N \times N$ transition probability matrix can be represented by N^2 crisp automata of an SFDES whose occurrence probabilities are the same as the N^2 transition probabilities in the transition matrix. It is important to point out that this representation is exact - the SFDES fully preserves the stochastic nature of the Markov chain without any change. Next, on the basis of this representation, we extend the binary states to fuzzy states for the Markov chain through N^2 fuzzy automata of the SFDES without altering its stochastic properties.

We not only establish connections among conventional Markov chains, fuzzy Markov chains, and SFDES but also devise a systematic procedure for designing and constructing fuzzy Markov chains. This represents the third crucial theoretical development and our contribution. Given pre- and post-event data pairs for Markov chain modeling, the pairs will first be used to calculate/estimate the transition probability matrix of a Markov chain. Hence, the occurrence probabilities of the N^2 fuzzy automata of the SFDES representing the fuzzy Markov chain are known. The N^2 FDES of the N^2 fuzzy automata employ fuzzy sets to transform binary states into fuzzy states, and we elucidate the underlying design principles governing this process. The data pairs and one of the supervised learning algorithms that we developed previously will be utilized to learn the N^2 event transition matrices of the FDES. A centroid defuzzification algorithm is introduced to convert a post-event fuzzy state vector to a numerical value, if crisp output, as oppose to fuzzy output, is desired for the fuzzy Markov chain. We also show how the batch least-squares method can be utilized to optimize parameters in the defuzzification algorithm to potentially enhance model accuracy.

In the next section, we introduce SFDES, which will be followed by major theoretical developments of the fuzzy Markov chains in Sections III and IV. In Section V, a detailed simulation example is presented to illustrate the key design steps and also show preliminary evidence on the advantage of the fuzzy Markov chain over its binary counterpart in terms of model accuracy. Considerations regarding design choices are elaborated upon in Section VI. Conclusions are drawn in the last section.

II. INTRODUCTION TO STOCHASTIC FUZZY DISCRETE EVENT SYSTEMS

Assume an SFDES is comprised of H FDES, each of which is represented by a fuzzy automaton $\tilde{G}_k$, $k = 1, \ldots, H$. The fuzzy automata take place randomly one at a time based on their occurrence probabilities with the probability for $\tilde{G}_k$ being p_k. If two or more events occur consecutively for at least one of the fuzzy automata, the SFDES is said to be a multi-event SFDES. Otherwise, the SFDES is of the single-event type, meaning each of its fuzzy automata has only one event. This study involves single-event SFDES only.

$\tilde{G}_k$ meets the mathematical definition given in (1). $\tilde{G}_k$ has only one fuzzy event, which is denoted as $\tilde{\mathbf{\Psi}}_k$ and is represented by an $N \times N$ event transition matrix, that is, $\tilde{\mathbf{\Psi}}_k = (a_{ij}^k)_{N \times N}$ with all the matrix elements falling in [0, 1]. The elements in a row (or a column) of $\tilde{\mathbf{\Psi}}_k$ are not required to be summed to 1. The states in the row and column directions are arranged identically.

System state of an SFDES is represented by $\mathbf{Q}$ in (1) where $\mathbf{Q} = [Q_1, Q_2, \ldots, Q_N]$. Membership value of Q_i, denoted by S_i, is in [0, 1], and the fuzzy state vector $\tilde{\mathbf{\Theta}} = [S_1, S_2, \ldots, S_N]$ characterizes the system state. $\mathbf{Q}$ is usually expressed as an intuitive linguistic term set (e.g., $\mathbf{Q}$ = ["Improved State," "Fair State," "Stable State," "Serious State"]). S_i can be obtained by applying a fuzzy set (or sets) defined for Q_i to the value (or values) of the variable (or variables) related to Q_i. The fuzzy set can be multi-dimensional if more than one variable is involved. Alternatively, multiple one-dimensional fuzzy sets, one for a variable, can be used along with a fuzzy aggregator (e.g., a fuzzy AND operator) to produce S_i.

If an event takes place, the SFDES will transfer from a pre-event fuzzy state (first such state is the initial state) to a post-event fuzzy state through the event transition matrix of the fuzzy automaton representing the event. The post-event state can be computed by using a fuzzy inference operator. Assume pre-event fuzzy state vector is $\tilde{\mathbf{\Theta}}_0 = [S_{10}, \ S_{20}, \ \ldots \ , S_{N0}]$ and fuzzy event $\tilde{\mathbf{\Psi}}_k$ occurs. Then, the post-event fuzzy state vector of the system, denoted as $\tilde{\mathbf{\Theta}}_1$, is

$$\tilde{\mathbf{\Theta}}_1 = \tilde{\mathbf{\Theta}}_0 \circ \tilde{\mathbf{\Psi}}_k = [S_{10}, \quad S_{20}, \quad ... \quad , S_{N0}] \circ \begin{bmatrix} a_{11}^k & a_{12}^k & ... & a_{1N}^k \\ a_{21}^k & a_{22}^k & ... & a_{2N}^k \\ ... & & & \\ a_{N1}^k & a_{N2}^k & ... & a_{NN}^k \end{bmatrix} = [S_{11}, \quad S_{21}, \quad ... \quad , S_{N1}].$$

The symbol $\circ$ denotes a fuzzy inference operation such as the widely-used max-product composition method or max-min composition method. As an example, if the max-product composition operation is used, the result of $\tilde{\mathbf{\Theta}}_1$ above is attained as

$$S_{j1} = max(S_{10}a_{1j}^k, S_{20}a_{2j}^k, ..., S_{N0}a_{Nj}^k) \quad (2)$$

where $1 \leq j \leq N$.

An SFDES becomes a stochastic discrete event system if the fuzzy automaton is replaced by the conventional automaton whose states and events are binary. We have the following definition.

Definition 1: A stochastic discrete event system is an SFDES whose states are binary and each of its event transition matrix contains all 0's except for a single occurrence of 1.

For a stochastic discrete event system, S_i and a_{ij}^k are either 0 or 1, hence $\tilde{\mathbf{\Psi}}_k$ contains a single 1 (i.e., a state is allowed to only transfer to one state at a time). There will be no fuzzy sets for the variables related to Q_i. Like their SFDES counterparts, stochastic discrete event systems can also be classified as single-event stochastic discrete event systems and multi-event stochastic discrete event systems. Stochastic discrete event systems form a special class of SFDES.

It is noteworthy that the concept of a stochastic discrete event system is innovative; it has not been previously discussed in the literature of FDES or discrete event systems.

III. Connections between Markov Chains and Stochastic (Fuzzy) Discrete Event Systems

A. Discrete-Time Finite Markov Chains

Without loss of generality, assume the Markov chains involve M continuous random variables, denoted as x_i, that are defined on $[\alpha_i, \beta_i]$, $i = 1, 2, \ldots, M$. Let the random variable vector be $\mathbf{X} = [x_1, x_2, \ldots, x_M]$. The interval $[\alpha_i, \beta_i]$ is divided into N_i subintervals. Denote the first subinterval that starts with α_i, second subinterval,. . . , last subinterval that ends with β_i as L_1^i,L_2^i,. . . ,$L_{N_i}^i$, respectively. This results in a total of $N = N_1 \times N_2 \times \cdots \times N_M$ different combinations of the total $K = \sum_{i=1}^{M} N_i$ subintervals. Each combination represents a hypercube in the M-dimensional space. That means there are a total of N different states, one state for a hypercube. Denote the hypercube corresponding to State q of the Markov chains as C_q.

Example 1: Suppose x_1 is defined on [0, 20] that is divided into four subintervals (e.g., $N_1 = 4$). They are [0, 4], (4, 9], (9, 15], (15, 20], which are L_1^1, L_2^1, L_3^1 and L_4^1. Assume x_2 is defined on [-2 5] has two subintervals L_1^2 = [-2, 2] and L_2^2 = (2, 5]. There are $N = 4 \times 2 = 8$ different combinations of the subintervals. Each combination occupies a rectangle area (i.e., hypercube) in the x_1-x_2 space and represents a state.

We comment that there exist discrete-time Markov chains whose state definition does not rely on continuous random variables. These Markov chains are not covered in this study.

The transition probability matrix of the Markov chains, denoted as P, is $N \times N$. Conventionally, the N states are arranged in the same way in the row direction and the column direction, and the rows and columns represent pre- and post-event states, respectively. The matrix elements are fix and time-independent state transition probabilities, denoted as p_{ij}, $1 \leq i, j \leq N$, and p_{ij} are non-negative and satisfy $\sum_{j=1}^{N} p_{ij} = 1$. By definition, a Markov chain must meet the requirement that the probability of entering any new state depends only on the previous state and is unrelated to any other historical states. A transition matrix P together with the initial probability associated with each state *completely* defines a Markov chain [49].

The probability of State i transferring to State j is p_{ij} which in practice can be estimated by counting the number of times that State i transferring to State j in all the state transitions. State i transferring to State j forms a pair of pre-event state and post-event state, denoted as (State i, State j), and the number of such pairs is designated by Ω_{ij}. That means $p_{ij} \approx \Omega_{ij}/\Omega_{i*}$ where $\Omega_{i*} = \sum_{j=1}^{N} \Omega_{ij}$. The larger the Ω_{ij}, the more accurate the estimation of p_{ij}. The total number of state pairs is $\Omega = \sum_{i=1}^{N} \Omega_{i*}$.

To develop a Markov chain model, a sequence of n sets of values of the M variables generated by the system being modeled at n different times is assumed to be available. The values are mapped by the subintervals into hypercubes or states, resulting in the corresponding sequence of hypercubes or states. Then $n - 1$ pairs of pre- and post-event states (or hypercubes) can be obtained, with post-event state (or hypercubes) of current pair being pre-event state (or hypercubes) of next pair. We denote pre- and post-event state pair as (State i, State j) and the corresponding hypercube pair as (C_i, C_j).

B. Discrete-Time Finite Markov Chains Are a Class of Single-Event Stochastic (Fuzzy) Discrete Event Systems

We link the Markov chains to the stochastic discrete event systems through the following theorem.

Theorem 1: The discrete-time finite Markov chains are a class of the single-event stochastic discrete event systems.

Proof: Given a Markov chain with N states and transition probability matrix $P = (p_{ij})_{N \times N}$, we can always configure a single-event stochastic discrete event system using the same N states. We do not configure a multi-event stochastic discrete event system because for the Markov chain, the probability of entering any new state depends only on the previous system state and is unrelated to any other historical states. The single-event stochastic discrete event system will consist of N^2 discrete event systems, and the event transition matrices of their N^2 automata are $N \times N$ with the states arranged in the same way as those for the transition probability matrix P of the Markov chain in terms of the row and column. We denote the automata as G_{ij}, $1 \leq i, j \leq N$, and let the occurrence probability of G_{ij} be p_{ij}, which means G_{ij} covers transfer of system state from State i to State j. To make this state transfer happens, we let all the elements in the event transition matrix of G_{ij} be 0 except for the two elements corresponding to (i, j) which are set to 1. It can be easily proven that the post-event state vector resulted from fuzzy inference involving G_{ij} and a pre-event state vector whose only nonzero membership is 1 for the i-th element (i.e., State i) will have 0 membership value everywhere except 1 for the j-th element (i.e., State j) when either the max-product composition method or the max-min composition method is used for computing state transfer.

At this point, we have proved that functionality and characteristics of the Markov chain and the single-event stochastic discrete event system are exactly the same.

A Markov chain can be represented as a single-event stochastic discrete event system. The reverse is generally not true. A single-event stochastic discrete event system is a Markov chain only when its stochastic characteristics is defined by the transition probability matrix of the Markov chain. Hence, one can conclude that the Markov chains are a class of the single-event stochastic discrete event systems. **QED**

The proof of Theorem 1 is constructive. That is, the proof is accomplished by constructing the single-event discrete event systems that are equivalent to the Markov chains. Note that for automaton G_{ij}, only State i in the pre-event state vector and State j in the post-event state vector are involved in state transfer. The remaining states do not participate because their membership values are 0. We call an individual state the Dominant State if its membership value is the highest among the N states in a state vector (Two or more individual states can be the Dominant States simultaneously if they all have the same highest membership values). The remaining individual states are named the Non-dominant States. States i and j are the Dominant States of G_{ij}. These state designations will facilitate comprehension later during the transition from "State i transfers to State j" in a binary Markov chain to "Fuzzy State i transfers to Fuzzy State j" in the corresponding Markov chain with fuzzy states (i.e., the fuzzy Markov chains).

In summary, given a Markov chain described in Subsection III-A, a single-event discrete event system can be easily made, which consists of N^2 automata G_{ij} with pre- and post-event Dominant States being States i and j, respectively. The occurrence probability of G_{ij} is p_{ij}. Ω_{ij} pairs of (Dominant State i, Dominant State j) are associated with G_{ij}.

The stochastic nature of a Markov chain is dictated by the transition probability matrix P. This characteristic is fully preserved by the corresponding single-event discrete event system through its N^2 automata G_{ij} whose occurrence probabilities are p_{ij}. Note that (Dominant State i, Dominant State j) of G_{ij} are binary and correspond to (State i, State j) of the original Markov chains, and the two pairs are associated with the same p_{ij}. Thus, we can state the following result.

Corollary 1: The single-event stochastic discrete event systems share the identical stochastic nature characterized in the transition probability matrix P of the discrete-time finite Markov chains.

For brevity, a formal proof is omitted. The presentation in this subsection clearly indicates the correctness of this finding.

IV. Fuzzy Markov Chains Realized through Supervised Learning Stochastic Fuzzy Discrete Event Systems

A. Extending Binary States of the Discrete-Time Finite Markov Chains to Fuzzy States

For a discrete-time finite Markov chains, it remains in the same state as long as each value of the M variables stays within the same subinterval, no matter how big the changes of the values are. Such character causes abrupt state change, which often fails to represent gradual state changes in the reality (e.g., change of patient's clinical state). The binary nature of states makes it impossible for the Markov chain to be in two or more states simultaneously, let alone events transferring from one state to multiple states at the same time. All in all, the Markov chains are incapable of effectively modeling systems whose states and events are intrinsically imprecise or vague.

We now expand the binary states to fuzzy states. On the basis of the Markov chains in Subsection III-A, a continuous fuzzy set is defined on $[\alpha_i, \beta_i]$ for each of the N_i subintervals of x_i, resulting in N_i fuzzy sets. The total number of fuzzy sets for the M variables is K. The fuzzy sets dedicated to the subintervals L_1^i,L_2^i,...,$L_{N_i}^i$ are denoted respectively as $\tilde{F}_1^i(x_i)$,$\tilde{F}_2^i(x_i)$,...,$\tilde{F}_{N_i}^i(x_i)$. It should be stressed that the universe discourse of $\tilde{F}_h^i(x_i)$ is $[\alpha_i, \beta_i]$, not L_h^i. Given a specific value of x_i, some or all of the N_i fuzzy sets may have nonzero membership values, depending on how the fuzzy sets are defined.

All the K fuzzy sets are required to be convex. $\tilde{F}_h^i(x_i)$ is convex if and only if for any a, $b \in [\alpha_i, \beta_i]$ and any $\lambda \in [0, 1]$, $\tilde{F}_h^i(\lambda a + (1-\lambda)b) \geq min(\tilde{F}_h^i(a), \tilde{F}_h^i(b))$. $\tilde{F}_h^i(x_i)$ is also required to have at least one membership value being 1 and not have membership value 1 outside the subinterval L_h^i, which are common requirements for fuzzy systems in the literature. Finally, suppose $L_h^i = [d_{h-1}^i, d_h^i]$ and $x_i \in L_h^i$, $h = 1, 2, \ldots, N_i$. The K fuzzy sets must be so defined that (1) $\tilde{F}_1^i(d_1^i) = \tilde{F}_2^i(d_1^i)$, (2) for $h = 3, 4, \ldots, N_i - 2$, $\tilde{F}_{h-1}^i(d_{h-1}^i) = \tilde{F}_h^i(d_{h-1}^i) = \tilde{F}_h^i(d_h^i) = \tilde{F}_{h+1}^i(d_h^i)$, and (3) $\tilde{F}_{N_i-1}^i(d_{N_i-1}^i) = \tilde{F}_{N_i}^i(d_{N_i-1}^i)$. There is no other restriction on the fuzzy sets and the fuzzy sets can be symmetrical or asymmetrical. The reason for imposing these mild requirements is to ensure that when $x_i \in L_h^i$, the value of $\tilde{F}_h^i(x_i)$ is the highest among the N_i values from the N_i fuzzy sets fuzzifying x_i. We call $\tilde{F}_h^i(x_i)$ the Primary Fuzzy Set for L_h^i. The remaining $N_i - 1$ fuzzy sets are named the Secondary Fuzzy Sets as far as L_h^i is concerned. These distinctions in the fuzzy sets will help us introduce an important notion called the Dominant State later.

Example 2: Continue with Example 1. Suppose four fuzzy sets dedicated to the four subintervals of x_1, L_1^1 to L_4^1, are as shown in Fig. 1. They all meet the three requirements.

Most, if not all, popular fuzzy sets are convex and can be easily configured to meet the above-mentioned requirements. They include the Gaussian type, triangular type, and trapezoidal type.

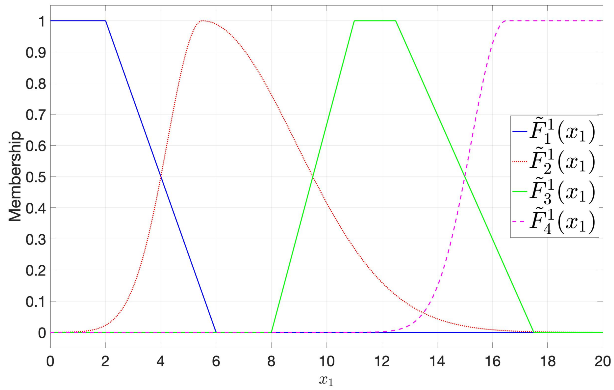

Fig. 1: Four hypothetical fuzzy sets for fuzzifying x_1.

The fuzzy sets fuzzify the N binary states through fuzzification of the values of the M variables, resulting in N fuzzy states. Specially, M Primary Fuzzy Sets, each of which is responsible for a subinterval in which one of the M variables lies, covers one hypercube, or equivalently one state. Because the relationship among the M subintervals is intersection (i.e., binary AND), the relationship among the corresponding M Primary Fuzzy Sets is fuzzy intersection (i.e., fuzzy AND). Hence either the product fuzzy AND operator or the min fuzzy AND operator can be used to calculate a combined/aggregated membership for the hypercube and the state. This aggregated membership is the membership for the fuzzy state covering the hypercube. The aggregated membership for hypercube C_q, hence for State q, is denoted as S_q. The N aggregated membership values form a pre-event fuzzy state vector. Given a particular value of $\mathbf{X}$, some or all of the N fuzzy states may have nonzero membership values, depending on the definitions of the K fuzzy sets.

At this point, we have extended the Markov chains with N binary states characterized by N M-dimensional hypercubes to the Markov chains with N fuzzy states covering the same N hypercubes. Basically, a Markov chain with fuzzy states consists of the transition probability matrix P of the original Markov chains and a set of N fuzzy states, each of which is determined by M Primary Fuzzy Sets fuzzifying M variables.

B. Enabling Fuzzy Event Handling for the Fuzzy Markov Chains through Stochastic Fuzzy Discrete Event Systems

As stated earlier, we require our target fuzzy Markov chains to not only have fuzzy states, but also have the ability to handle fuzzy events. As is, the Markov chains with fuzzy states cannot deal with fuzzy events. A fuzzy event means a pre-event fuzzy state transfers to a post-event fuzzy state. Since a fuzzy state implies that the N individual states (e.g., $\tilde{\mathbf{\Theta}}$ in Section II) can all have nonzero membership values, the question is how should a Markov chain with fuzzy states compute fuzzy state transfer? Recall that fuzzy state transfer is calculated using a pre-event fuzzy state vector, an event transition matrix, and a fuzzy inference method. Therefore, it is obvious that the Markov chains with fuzzy states are incapable of handling fuzzy events due to their structural deficiency - they do not have event transition matrices.

SFDES can bridge the gap. Given a Markov chain, a single-event stochastic discrete event system that is equivalent to the Markov chain can be constructed, per Theorem 1. The stochastic discrete event system consists of N^2 automata G_{ij}, $1 \le i, j \le N$. The occurrence probability of G_{ij} is p_{ij}, which is the probability of Dominant State i transferring to Dominant State j, or equivalently, State i transferring to State j for the original Markov chains. For a Markov chains with fuzzy states, we now show how to construct a single-event SFDES based on the stochastic discrete event system so that the resulting SFDES can not only represent the Markov chain with fuzzy states, but also is capable of computing fuzzy state transfer.

The single-event SFDES will have the same N fuzzy states as the Markov chain with fuzzy states does. There are N^2 FDES in the SFDES, each is represented by a fuzzy automaton denoted as $\tilde{G}_{ij}$, $1 \le i, j \le N$, and the occurrence probability of $\tilde{G}_{ij}$ is p_{ij}. The N fuzzy states in the $N \times N$ event transition matrix of $\tilde{G}_{ij}$, denoted as $\tilde{\mathbf{\Psi}}_{ij}$, are arranged in the same way as those in the Markov chain with fuzzy states in terms of the row and column. $\tilde{G}_{ij}$ is associated with the Ω_{ij} pairs of (State i, State j) like G_{ij} is. Recall that the pairs of (State i, State j) of the original Markov chains represent the state transfer whose probability is p_{ij}. These pairs are corresponded by the pairs of (Dominant State i, Dominant State j) of the nonfuzzy automaton G_{ij}. We now establish a condition for the pairs of (State i, State j) to be pairs of (Dominant State i, Dominant State j) of the fuzzy automaton $\tilde{G}_{ij}$.

Lemma 1: A sufficient condition for the pairs of (State i, State j) of the binary Markov chains that represent the state transfer with probability p_{ij} to become the pairs of (Dominant State i, Dominant State j) of the fuzzy automaton $\tilde{G}_{ij}$ having the same probability is that the element of the i-th row and j-th column of the event transition matrix $\tilde{\mathbf{\Psi}}_{ij}$ is 1.

Proof: The individual state i represents a hypercube formed by M subintervals, one for each of the M variables. When the value of x_i is fuzzified by its respective N_i fuzzy sets $F_h^i(x_i)$, the membership value of the Primary Fuzzy Set responsible for the subinterval where the value of x_i lies will be higher than those of the rest of the $N_i - 1$ Non-primary Fuzzy Sets because of the way the N_i fuzzy sets are defined in IV-A. This is true for all the M values of the M variables. Therefore, the aggregated membership value for the hypercube, that is, State i, will be higher than the rest of the $N-1$ hypercubes or states, regardless of the type of fuzzy AND operator used. This makes State i the Dominant State relative to the other individual states in the pre-event state vector.

If the element of the i-th row and j-th column of the event transition matrix $\tilde{\mathbf{\Psi}}_{ij}$ is 1, the membership value of the j-th individual state in the post-event fuzzy state vector will have the highest membership value among the N states, which means State j is the Dominant State. This is the case for both the max-product and max-min fuzzy inference methods. **QED**

We comment that the definitions of the K fuzzy sets given in IV-A are quite systematic and general. They represent one way to make State i the Dominant State i. There may exist other ways of defining the fuzzy sets that can achieve the same goal. For better presentation, we do not dive deeper into this matter in this paper.

All in all, $\tilde{G}_{ij}$ is the fuzzy counterpart of G_{ij}. At this point, we have successfully equipped the Markov chains with fuzzy states with the ability of handling fuzzy events. This capability is realized through single-event SFDES. In light of these developments, we have the following finding.

Theorem 2: When the condition set in Lemma 1 is satisfied, discrete-time finite fuzzy Markov chains with the same stochastic characteristics defined by the transition probability matrix P of the binary Markov chains can be realized through single-event SFDES.

Proof: The presentation in this and previous subsections has already shown constructively: (1) how the binary Markov chains can be extended to have fuzzy states, and (2) how the Markov chains with fuzzy states can be equipped with fuzzy-event-handling ability through single-event SFDES.

Other than fuzzy states and ability of handling fuzzy events, a third and final requirement for our target fuzzy Markov chains is to preserve the stochastic characteristics of the binary Markov chains. Recall that the Ω_{ij} pairs of (State i, State j) of the binary Markov chains represent the transfer from State i to State j and the state transition probability is p_{ij}. After the fuzzification of State i in these pairs in the fuzzy automaton $\tilde{G}_{ij}$ whose occurrence probability is p_{ij}, Ω_{ij} pre-event fuzzy state vectors will result, with the individual State i being the Dominant State. When the condition set in Lemma 1 is met, the individual State j will be the Dominant State in the corresponding Ω_{ij} post-event fuzzy state vectors as a result of applying either the max-product inference method or the max-min inference method to the pre-event fuzzy state vectors. Note that (Dominant State i, Dominant State j) of $\tilde{G}_{ij}$ degenerates into (State i, State j) of the binary Markov chains when all the Non-dominant States in the fuzzy state vectors have zero membership value. Hence, with an occurrence probability of p_{ij}, the fuzzy automaton $\tilde{G}_{ij}$ maintains the stochastic nature defined by the state transition probability p_{ij} of the binary Markov chain. **QED**

It is worth emphasizing that because the fuzzy Markov chains preserve the stochastic characteristics defined by the transition probability matrix of binary Markov chains, stochastic properties of the fuzzy and binary Markov chains, such as steady-state probabilities and multi-step transition probabilities [49], are identical.

We name the fuzzy Markov chains mentioned in Theorem 2 the Stochastic-Fuzzy-Discrete-Event-System-Based Fuzzy Markov Chains (SFDES-Based Fuzzy Markov Chains for short) and it is formally defined as follows:

Definition 2: A SFDES-Based Fuzzy Markov Chain is a single-event SFDES whose event transition matrices satisfy Lemma 1 and its stochastic characteristics is defined by the transition probability matrix of the corresponding binary Markov chain.

The following statement is obvious.

Corollary 2: The discrete-time finite SFDES-Based Fuzzy Markov Chains are a class of single-event SFDES.

Theorem 3: A discrete-time finite Markov chain is a special case of the discrete-time finite SFDES-Based Fuzzy Markov Chains.

Proof: A binary Markov chains does not involve with fuzzy sets, and hence the membership values for states can only be 0 or 1. To align an SFDES-Based Fuzzy Markov Chains with it, we introduce K special fuzzy sets for the fuzzy Markov chain as follows: Primary Fuzzy Set $\tilde{F}_h^i(x_i)$ responsible for the subinterval $L_h^i = [d_{h-1}^i, d_h^i]$ has a membership value of 1 everywhere in the subinterval and 0 everywhere else in $[\alpha_i, \beta_i]$. With these K special fuzzy sets, (State i, State j) of the binary Markov chains becomes a special case of (Dominant State i, Dominant State j) of the SFDES realizing the fuzzy Markov chain because all the Non-dominant States in the fuzzy state vectors have zero membership value. The special fuzzy sets also make each event transition matrix of the SFDES contain only one non-zero element. Actually, the value of that element is 1, and the element is at the i-th row and the j-th column of $\tilde{\mathbf{\Psi}}_{ij}$. At this point, the single-event SFDES degenerates into a single-event stochastic nonfuzzy discrete event system, which incorporates discrete-time finite Markov chains per Theorem 1. **QED**

Theorem 3 and Corollary 2 lead to the following conclusion.

Corollary 3: The discrete-time finite Markov chains are special cases of single-event SFDES.

C. Supervised Learning of the Event Transition Matrices of the SFDES-Based Fuzzy Markov Chains

With the structure of the SFDES-Based Fuzzy Markov Chains having been designed, a key question then is how to determine the element values for each of the N^2 event transition matrices of $\tilde{G}_{ij}$? The answer is that they can all be automatically learned, one at a time, using an online stochastic-gradient-based supervised learning algorithm that we developed previously [44]. That algorithm learns each of the event transition matrices independent using the paired pre- and post-event fuzzy state vectors that are related to it.

Such fuzzy state vector pairs are available for the N^2 fuzzy automata that we just constructed. Recall that there are a total of Ω state pairs that characterize state transfer for the binary Markov chains. Correspondingly, for the fuzzy Markov chains, state transfer is characterized by Ω pairs of pre- and post-event fuzzy state vectors. States i and j are the pre-event state and post-event state, respectively, with the transition probability of p_{ij} for the binary Markov chain. They are the respective Dominant States in the pre- and post-event fuzzy state vectors for the corresponding fuzzy automaton $\tilde{G}_{ij}$ of the SFDES-Based Fuzzy Markov Chains with the occurrence probability of $\tilde{G}_{ij}$ being p_{ij}.

The fuzzy state vector pairs corresponding to the Ω_{ij} pairs of (State i, State j) can be obtained for $\tilde{G}_{ij}$, $1 \le i, j \le N$ by fuzzifying the values of two sets of M variables that correspond to State i and State j. N_i membership values will result for the N_i Primary Fuzzy Sets $\tilde{F}_1^i(x_i)$,$\tilde{F}_2^i(x_i)$,…, $\tilde{F}_{N_i}^i(x_i)$ that cover the N_i subintervals of x_i. How many of the memberships will be nonzero depends on the definitions of $\tilde{F}_1^i(x_i)$,$\tilde{F}_2^i(x_i)$,…, $\tilde{F}_{N_i}^i(x_i)$ as well as specific values of the M variables. Aggregation of these memberships (see Section IV-A) will generate N membership values, one for each individual state. They constitute a pre- or post-event fuzzy state vector, depending on whether the M variables are pre- or post-event.

Therefore, Ω_{ij} pairs of pre- and post-event fuzzy state vectors, denoted as ($\tilde{\mathbf{\Theta}}_h^{ij}$, $\tilde{\mathbf{\Phi}}_h^{ij}$), $h = 1, 2, \ldots, \Omega_{ij}$, that correspond to (State i, State j) are available for learning of the event transition matrix $\tilde{\mathbf{\Psi}}_{ij}$ of $\tilde{G}_{ij}$. Owing to the definitions of the K fuzzy sets in Section IV-A, States i and j are the Dominant States in the fuzzy state vectors. Let the variable vectors corresponding to $\tilde{\mathbf{\Theta}}_h^{ij}$ and $\tilde{\mathbf{\Phi}}_h^{ij}$ be denoted as $\mathbf{X}_h^{ij}$ and $\bar{\mathbf{X}}_h^{ij}$, respectively, and they form a pre- and post-event variable vector pair ($\mathbf{X}_h^{ij}$, $\bar{\mathbf{X}}_h^{ij}$). We also let $\tilde{\mathbf{\Theta}}_h^{ij} = [S_{1,h}^{ij}, S_{2,h}^{ij}, \ldots, S_{N,h}^{ij}]$ and $\tilde{\mathbf{\Phi}}_h^{ij} = [\bar{S}_{1,h}^{ij}, \bar{S}_{2,h}^{ij}, \ldots, \bar{S}_{N,h}^{ij}]$. $S_{k,h}^{ij}$ and $\bar{S}_{k,h}^{ij}$ are the aggregated membership values of the M memberships resulted from fuzzification of the values of the M variables in $\mathbf{X}_h^{ij}$ and $\bar{\mathbf{X}}_h^{ij}$, respectively.

With ($\tilde{\mathbf{\Theta}}_h^{ij}$, $\tilde{\mathbf{\Phi}}_h^{ij}$), the learning algorithm can learn the $N \times N$ event transition matrix $\tilde{\mathbf{\Psi}}_{ij}$ of $\tilde{G}_{ij}$. The algorithm works iteratively. One pair of the fuzzy state vectors leads to one iteration of the matrix element value updating. Let $\tilde{\mathbf{\Psi}}_{ij}(h) = (a_{mn}^{ij}(h))_{N \times N}$, where the value of a_{mn}^{ij} modified after the h-th iteration of parameter updating is denoted by $a(h)_{mn}^{ij}$ while the corresponding event transition matrix is denoted by $\tilde{\mathbf{\Psi}}_{ij}(h)$. The h-th pair, ($\tilde{\mathbf{\Theta}}_h^{ij}$, $\tilde{\mathbf{\Phi}}_h^{ij}$), will update the value of $a_{mn}^{ij}(h-1)$, $1 \leq m, n \leq N$, using the following formula:

$$a_{mn}^{ij}(h) = a_{mn}^{ij}(h-1) - \lambda S_{m,h}^{ij}(\hat{S}_{n,h}^{ij} - \bar{S}_{n,h}^{ij})\delta_h^{ij} \qquad (3)$$

where

$$\delta_h^{ij} = \begin{cases} 1, \text{if} \hat{S}_{n,h}^{ij} = S_{m,h}^{ij} a_{mn}^{ij}(h-1) \\ 0, \text{otherwise.} \end{cases} \qquad (4)$$

Initial values of a_{mn}^{ij} (i.e., $a_{mn}^{ij}(0)$) are usually set randomly. λ is learning rate, which is a hyperparameter. Its value needs to be decided by the modeler. The same λ value may be used for the learning of all the matrices. Actual post-event fuzzy state vector due to pre-event fuzzy state vector $\tilde{\mathbf{\Theta}}_h^{ij}$ is denoted as $\hat{\mathbf{\Phi}}_h^{ij} = [\hat{S}_{1,h}^{ij}, \hat{S}_{2,h}^{ij}, \ldots, \hat{S}_{N,h}^{ij}]$. It is computed by using $\tilde{\mathbf{\Theta}}_h^{ij}$, $\tilde{\mathbf{\Psi}}_{ij}(h-1)$, and the max-product composition. More specifically, $\hat{\mathbf{\Phi}}_h^{ij} = \tilde{\mathbf{\Theta}}_h^{ij} \circ \tilde{\mathbf{\Psi}}_{ij}(h-1)$.

Because States i and j are the Dominant State of the pre- and post-event fuzzy state vectors, respectively, for the fuzzy automaton $\tilde{G}_{ij}$, it is likely that the element in the i row and j-th column of the event transition matrix $\tilde{\mathbf{\Psi}}_{ij}$ is 1 (i.e., a_{ij}^{ij} = 1) when the learning of $\tilde{\mathbf{\Psi}}_{ij}$ is completed. If not, it needs to be set to 1 so that the condition set in Lemma 1 is met.

The process of learning the event transition matrix $\tilde{\mathbf{\Psi}}_{ij}$ is as follows: Ω_{ij} pairs of ($\tilde{\mathbf{\Theta}}_h^{ij}$, $\tilde{\mathbf{\Phi}}_h^{ij}$) are fed to the algorithm consecutively one pair a time. The algorithm uses formula (3) to modify the values of the matrix elements. When all the Ω_{ij} data pairs are used, one round of parameter learning has been completed. More rounds of the sample feeding may be desired as it can lead to more parameter updating, which may produce better learning outcome. The learning ends either when the pre-set round of sampling feeding has been performed or changes in matrix element values between two consecutive rounds are smaller than a pre-set threshold.

Learning of the event transition matrices takes place one matrix a time, and learning of each matrix is independent one from another. The learning continues until all the N^2 event transition matrices of the single-event SFDES have been learned.

The size of Ω_{ij} affects learning outcome for $\tilde{\mathbf{\Psi}}_{ij}$. Our limited simulation study indicates that a large sample size may not be needed. For example, only 100 sample pairs were enough to learn a 4×4 event transition matrix accurately after 60 rounds of sample feeding [44].

Computing time for executing the learning of the event transition matrix $\tilde{\mathbf{\Psi}}_{ij}$ depends on the magnitudes of N and Ω_{ij}. The larger they are, the longer computing time. Our simulation experience suggests that a relatively new typical personal computer could complete the learning of one event transition matrix involving moderate magnitudes of N and sample pairs in less than 20 seconds. For a more complex FDES with larger magnitudes of N and sample pairs, the execution time is expected to be longer, but not prohibitively long. With the rapid advancement of computer hardware, computing time is not considered a bottleneck.

D. Defuzzifying Post-Event Fuzzy States of the SFDES-Based Fuzzy Markov Chains for Crisp Model Output

Given a binary Markov chain and a value of $\mathbf{X}$, the value will first be mapped to a pre-event individual state (i.e., hypercube), say State i. The post-event state will then be determined randomly based on the transition probability matrix. Suppose that State j is the post-event state.

For the corresponding fuzzy Markov chain, the value of $\mathbf{X}$ will first be fuzzified, resulting in a pre-event fuzzy state vector with N aggregated membership values, one for each of the N states. One of the states is the Dominant State, and in this case it is State i. Aligning with the binary Markov chain, the fuzzy automaton $\tilde{G}_{ij}$ is randomly selected to act, producing a post-event fuzzy state vector whose Dominant State is State j, if the condition in Lemma 1 is met. In some applications, output of the SFDES-Based Fuzzy Markov Chains in the form of fuzzy state vectors may be appropriate and natural from the standpoint of man-machine interface. In many applications, however, fuzzy output may not be acceptable and crisp output is required. If this is the case, the post-event fuzzy state vector needs to be defuzzified to yield a crisp model output.

There are different defuzzifiers in the literature, each with pros and cons. Here, we consider two major ones - the maximum defuzzifier and the centroid defuzzifier. The maximum defuzzifier takes the value of the universe of discourse that corresponds to the highest membership value of the fuzzy set being defuzzified as the crisp output. When this defuzzifier is applied to the fuzzy Markov chains, the following finding is attained.

Theorem 4: If a discrete-time finite SFDES-Based Fuzzy Markov Chain satisfying the condition in Lemma 1 employs the maximum defuzzifier, its input-output mapping is identical to that of the binary Markov chain.

Proof: The proof is straightforward. As mentioned earlier in this subsection, for any p_{ij}, output of an SFDES-Based Fuzzy Markov Chain meeting the condition in Lemma 1 is a fuzzy state vector whose Dominant State coincides with the post-event state of the corresponding binary Markov chain. Hence, when the maximum defuzzifier is applied to the fuzzy state vector, the defuzzification result is that Dominant State because its membership value is the highest among the N states. This is to say the Dominant State is the same as the post-event state of the corresponding binary Markov chain. This holds true for any values of i and j. **QED**.

This result is of theoretical interest only as practically speaking, there is no advantage to use a fuzzy Markov chain with the maximum defuzzifer. Using the binary Markov chain would be far simpler.

The centroid defuzzifer, which is the most widely-used defuzzifier in the fuzzy system literature, takes the membership values of all the N fuzzy states into account in the defuzzification process. In order for this defuzzifier to work for the fuzzy Markov chains, each of the N states needs to have a representative value that will be used in defuzzifcation. Theoretically speaking, any value in a hypercube can serve as the representative value for the state associated with that hypercube. Practically speaking, though, some values may offer better representation than other values. The choice can also be application-dependent. In what follows, we focus on using the centers of the N hypercubes as the representative values for the N states. The resulting formula can be easily modified if other types of representative values are desired.

The center of a M-dimensional hypercube is composed of M middle points of the M subintervals that form the hypercube. Denote the center of State k (i.e., the k-th hypercube) as $\mathbf{\Gamma}_k = [\gamma_1^k, \gamma_2^k, \ldots, \gamma_M^k]$ where γ_q^k is the middle point of the subinterval of the q-th variable in $\mathbf{X}$ that is associated with the hypercube. Given a post-event fuzzy state vector $[\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_N]$, the centroid defuzzifer produces the following M values, one for each of the M variables:

$$\hat{x}_q = \frac{\hat{S}_1\gamma_q^1 + \hat{S}_2\gamma_q^2 + \cdots + \hat{S}_N\gamma_q^N}{\hat{S}_1 + \hat{S}_2 + \cdots + \hat{S}_N}, q = 1, 2, \ldots, M. \quad (5)$$

Vector $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_M]$ is a point in the M-dimensional hypercube, which can serve as output of the SFDES-Based Fuzzy Markov Chains.

There are M center values for a state/hypercuble, leading to a total of $M \times N$ center values for the N states/hypercubes. All these center values are used in (5) and the N^2 fuzzy automata employ the same set of the center values.

E. Customizing Representative Values in Centroid Defuzzifier for Individual Fuzzy Automata via Batch Least-Squares Method

Instead of using the hypercube center of a state as the representative value for the state for all the fuzzy automata in the centroid defuzzifier, one may use the batch least-squares method to find different sets of representative values for different automata. The representative values found in this approach can lead to smaller errors between the defuzzified output of the fuzzy Markov chain and the desired model output represented by post-event variable values.

For fuzzy automaton $\tilde{G}_{ij}$, denote its to-be-found representative value vector for state k as $\mathbf{\Xi}_k = [\xi_1^k, \xi_2^k, \ldots, \xi_M^k]$, where $\xi_q^k \in [\alpha_q, \beta_q], q = 1, 2, \ldots, M$. We avoid the superscription or subscription ij to simplify presentation. Ω_{ij} pairs of pre- and post-event fuzzy state vectors ($\tilde{\mathbf{\Theta}}_h^{ij}$, $\tilde{\mathbf{\Phi}}_h^{ij}$) are available, where $\tilde{\mathbf{\Phi}}_h^{ij}$ are the results of fuzzification of $\bar{\mathbf{X}}_h^{ij}$ and $h = 1, 2, \ldots, \Omega_{ij}$. To simplify notations, we use $\tilde{\mathbf{\Phi}}_h = [\bar{S}_{1,h}, \bar{S}_{2,h}, \ldots, \bar{S}_{N,h}]$ and $\bar{\mathbf{X}}_h = [\bar{x}_{1,h}, \bar{x}_{2,h}, \ldots, \bar{x}_{M,h}]$, instead of $\tilde{\mathbf{\Phi}}_h^{ij}$ and $\bar{\mathbf{X}}_h^{ij}$, respectively. Replacing $\mathbf{\Gamma}_k$ by $\mathbf{\Xi}_k$, we can express (5) as

$$\begin{aligned} \hat{x}_{q,h} &= \frac{\bar{S}_{1,h}\xi_q^1 + \bar{S}_{2,h}\xi_q^2 + \cdots + \bar{S}_{N,h}\xi_q^N}{\bar{S}_{1,h} + \bar{S}_{2,h} + \cdots + \bar{S}_{N,h}} \\ &= \theta_h^1\xi_q^1 + \theta_h^2\xi_q^2 + \cdots + \theta_h^N\xi_q^N \end{aligned} \quad (6)$$

where

$$\theta_h^p = \frac{\bar{S}_{p,h}}{\bar{S}_{1,h} + \bar{S}_{2,h} + \cdots + \bar{S}_{N,h}}, \quad p = 1, 2, \ldots, N.$$

θ_h^p's are numbers that are calculated by using $\tilde{\mathbf{\Phi}}_h$. Thus, for any given q, (6) represents Ω_{ij} linear equations in terms of the M unknown parameters in $\mathbf{\Xi}_k$. The batch least-squares method can be directly used to find the optimal values of $\xi_q^1, \xi_q^2, \ldots, \xi_q^N$ that minimize the error $\sum_{h=1}^{\Omega_{ij}} (\hat{x}_{q,h} - \bar{x}_{q,h})^2$. Doing this for all the q values will minimize the total error $\sum_{q=1}^{M}\sum_{h=1}^{\Omega_{ij}} (\hat{x}_{q,h} - \bar{x}_{q,h})^2$ and achieve the optimal representative values for the fuzzy automaton $\tilde{G}_{ij}$.

For each fuzzy automaton, the least-squares method will result in N sets of optimal representative values, one for a state. Each set contains M values, one for a variable in $\mathbf{X}$. That is, the N sets of representative values constitute N points in the M-dimensional space, one point for a state. It is worth mentioning that the point for any specific state may lie outside the hypercube that is associated with that state. This situation cannot occur when the centers of the states are used in the defuzzification process.

V. Simulation Results

In this section, we provide an example to demonstrate how to apply the theory presented in Section IV to design and develop an SFDES-Based Fuzzy Markov Chain based on a given discrete-time finite binary Markov chain. Through computer simulations, we will show that owing to fuzzy states and the ability of handling fuzzy events, the fuzzy Markov chain outperforms the binary Markov chain in terms of less prediction errors.

We employed MATLAB (version R2021a) to write a program implmenting this example. The program ran on a 2021 MacBook Pro 13" equipped with M1 CPU chips, 16 GB RAM, and macOS Monterey.

Suppose that a stochastic process with continuous random variable x is modeled as a two-state discrete-time finite Markov chain (i.e., N = 2). The states are labeled as "Negative State" and "Positive State" that cover x = [-10, 0) and x = [0, 10], respectively (i.e., $M = 1$, $L_1^1 = [-10, 0)$, and $L_2^1 = [0, 10]$). A Gaussian random number generator with a mean of 0 and a standard deviation of 2 is employed to produce 5000 random numbers in [-10, 10] as samples of x. With an equal probability of being selected, 2240 of the 5000 numbers are randomly selected without replacement (i.e., a number can be selected only once) to form 1120 samples of pre- and post-event x-value pairs (i.e., $\Omega = 1120$). These sample pairs are independent one another and the sample size is assumed to be large enough to reflect the statistical features of the pre- and post-event pair population in the time series in x.

Based on the values of x, each pair is mapped to a pre- and post-event state pair, which belongs to one of the following four state pair groups - the Negative-Negative group, Negative-Positive group, Positive-Negative group, and Positive-Positive group. The naming of the groups reflects the nature of the pairs. For example, the Negative-Positive group has the pre- and post-event states that are Negative and Positive, respectively. The names of the other three groups can be interpreted similarly. The four groups cover four different state transitions. Subsequently, the transition probabilities of the Markov chain are estimated from the 1120 samples of state pairs (Table I). The integers in the table are numbers of state pairs (i.e., $\Omega_{ij}, i = 1, 2$) while the decimal numbers in the parentheses are the corresponding state transition probabilities (i.e., p_{ij}).

	Negative [-10, 0)	**Positive [0, 10]**
Negative [-10, 0)	112 (0.2)	448 (0.8)
Positive [0, 10]	160 (0.286)	400 (0.714)

TABLE I: Transition probability matrix of the example two-state discrete-time finite Markov chain. The rows and columns represent pre- and post-event states, respectively.

An SFDES-Based Fuzzy Markov Chain corresponding to the binary Markov chain is then constructed. We chose to use the fuzzy sets in Fig. 2 for the fuzzification of all the x values of the state pairs. The fuzzy sets satisfy all the requirements set in Section IV-A. The fuzzy sets "Negative" and "Positive" serve as the Primary Fuzzy Sets for the intervals [-10, 0) and [0, 10], respectively. Conversely, they function as the Non-primary Fuzzy Sets for [0, 10] and [-10, 0), respectively. With the fuzzy sets, the binary "Negative State" and "Positive State" of the Markov chain lead to respective "Fuzzy Negative State" and "Fuzzy Positive State" of the fuzzy Markov chain. "Fuzzy Negative State" is the Dominant State for [-10, 0) while "Fuzzy Positive State" is the Non-dominant State because the former always has a higher membership value than the latter for any x value in this interval. Likewise, for [0, 10], "Fuzzy Positive State" is the Dominant State and "Fuzzy Negative State" is the Non-dominant State.

With fuzzification, each x value will have two membership values, one for each of the two fuzzy sets. In other words, fuzzification allows each x value to belong to both (fuzzy) states simultaneously to different extents, better reflecting the nature of the x values than the interval-based binary categorization used by the conventional Markov chain. This may be especially desirable for values near 0, such as $x = 0.0001$ and $x = -0.0001$. These two example values would lead to different states in the binary Markov chain despite their very small difference. In contrast, both values lead to the two fuzzy sets with almost the same memberships (approximately 0.5), providing a more realistic and reasonable characterization. There are many x values that are near 0 as their mean and standard deviation are known to be 0 and 2, respectively. Hence, the fuzzy sets play a key role in enabling the fuzzy Markov chain to potentially outperform the binary Markov chain in terms of model accuracy.

The SFDES-Based Fuzzy Markov Chain consists of four fuzzy automata (i.e., $\tilde{G}_{ij}$), one for each of the four state pair groups. Together, they constitute a single-event SFDES. Correspondingly, we name these automata as the Negative-Negative fuzzy automaton, Negative-Positive fuzzy automaton, Positive-Negative fuzzy automaton, and Positive-Positive fuzzy automaton. Their occurrence probabilities are the same as their corresponding state pair groups (e.g., 0.8 for the Negative-Positive fuzzy automaton), which are the transition probabilities of the binary Markov chain.

To learn the four event transition matrices of the four fuzzy automata (i.e., $\tilde{\mathbf{\Psi}}_{ij}$), 75% of the x-value pair samples of each group is used to learn its event transition matrix (e.g., 120 pairs for learning the Positive-Negative fuzzy automaton). The remaining pairs will be used for testing purpose later. Before the learning can start, all the x-value pairs are converted to fuzzy state vector pairs through the fuzzy sets. Via the fuzzification, each pre-event or post-event x value is mapped to two membership values in a 1×2 fuzzy state vector. The first membership value is for the fuzzy set "Positive" and the second for the fuzzy set "Negative." This is to say each x-value pair becomes a fuzzy state vector pair with membership values. The pre- and post-event states are each represented by a 1×2 fuzzy state vector.

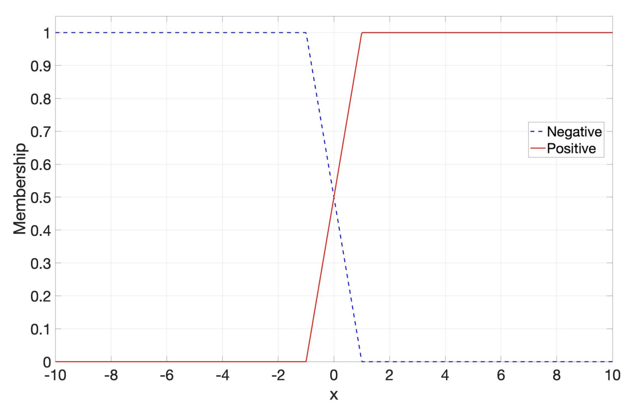

Fig. 2: Two fuzzy sets for fuzzifying x values associated with the state pairs in the four state groups.

Feeding all the fuzzy state vector pairs of any one of the four groups one by one to the above-mentioned learning algorithm will lead to learning of the event transition matrix of the fuzzy automaton associated with that state pair group. Multiple rounds of sample feeding were carried out when needed. The learning was terminated if the sum of the squared differences between the computed post-event fuzzy state vectors and the target post-event fuzzy state vectors stopped decreasing.

We experimented the learning rate λ in (3) from 0.1 to 1.5 with an increment of 0.1 and found $\lambda = 1.0$ to be the best in terms of the smallest sum of the squared errors. This value was subsequently used for the learning of all the four event transition matrices.

Each event transition matrix started with a random initial matrix and was learned separately. The learned matrices were as follows:

$$\begin{bmatrix} 1 & 0 \\ 0.4892 & 0.3750 \end{bmatrix} \quad \text{for Negative-Negative fuzzy automaton,}$$

$$\begin{bmatrix} 0.2741 & 0.9756 \\ 0.1496 & 0.3904 \end{bmatrix} \quad \text{for Negative-Positive fuzzy automaton,}$$

$$\begin{bmatrix} 1 & 0.1594 \\ 1 & 0 \end{bmatrix} \quad \text{for Positive-Negative fuzzy automaton,}$$

$$\begin{bmatrix} 0.0597 & 0.6805 \\ 0 & 1 \end{bmatrix} \quad \text{for Positive-Positive fuzzy automaton.}$$

Three rounds of the sample feeding were needed in order to produce the matrices for the Negative-Negative fuzzy automaton and Positive-Negative fuzzy automaton while two rounds were sufficient for the learning of the other two matrices. Computer CPU time was under 2.1 seconds for completion of the learning of any one of the matrices.

Examination of the four learned matrices shows that the elements corresponding to the Dominant State pairs are 1 for all but the matrix for the Negative-Positive fuzzy automaton. For that matrix, the element corresponding to (”Negative State,” ”Positive State”) is 0.9756, which is very close to 1. We set this element's value to 1 so that ”Fuzzy Negative State” and ”Fuzzy Positive State” are the Dominant States for the Negative-Positive fuzzy automaton per Lemma 1.

The x-value pair samples saved for testing purpose were then used to compare one-step prediction errors made by the Markov chain and SFDES-Based Fuzzy Markov Chain, group by group. There are 280 testing samples ($1120 \times 25\%$). For the Markov chain, if a pre-event x value presented was in [0, 10], -5 would be regarded as the predicted post-event x value for the Positive-Negative group and 5 would be used for the Positive-Positive group. The values of -5 and 5 were reasonable choices because they were the respective middle points of the intervals [-10, 0) and [0, 10] that covered respectively the ”Negative State” and ”Positive State” of the Markov chain. Similarly, if a pre-event x value presented was in [-10, 0), the predicted post-event x value would be either -5 for the Negative-Negative group or 5 for the Negative-Positive group. The prediction errors, as measured by the sum of the absolute values of the differences between the middle points and the post-event x values in the testing pairs, was then computed for each of the four x-value pair groups (Table II).

For the fuzzy Markov chain, each of the pre-event x values in the testing pairs of the four groups was first fuzzified by the two fuzzy sets in Fig. 2. The resulting 1×2 fuzzy state vectors were then utilized to calculate/predict the post-event 1×2 fuzzy state vectors for each of the four groups by using the max-product composition (2) and the relevant event transition matrix learned above. For each group, post-event fuzzy state vectors were defuzzified by the centroid defuzzifier (5) in which the representative values (i.e., γ_j^i) for the ”Fuzzy Negative State” and ”Fuzzy Positive State” were respectively chosen to be -5 and 5 (the middle points or centers of the respective intervals for x). The defuzzifier converted each 1×2 fuzzy state vector to a x value in [-10, 10]. The prediction error for each group was subsequently computed, which was the sum of the absolute values of the differences between these computed/predicted x values and the group's post-event x values in the testing pairs.

Table II shows the comparison results. Clearly, the fuzzy Markov chain made more accurate predictions than the binary Markov chain did for each and every group of testing pairs. For the four groups together, the total prediction errors yielded by the fuzzy Markov chain is 25.52% less than that produced by the Markov chain.

We comment that similar improved accuracy on one-step prediction can be expected if the SFDES-Based Fuzzy Markov Chain is applied to a time series in x to prospectively predict next x value of the process based on constantly changing current reading of x value. Actual current state of the process is known as it is determined by the current value of x. Which state the process will be in next depends on the current state as well as the state transition probabilities, and its probability is determined by Table I. The fuzzy Markov chain is anticipated to yield more accurate predicted x values associated with the possible next states than the binary Markov chain does. Per Table II, such potential accuracy improvement is for the times series as a whole, not for each individual prediction.

	NN (28)	NP (112)	PN (40)	PP (100)	Sum
MC	94.631	394.175	139.121	338.841	966.768
FMC	85.683	171.493	132.986	329.861	720.023

TABLE II: Comparing prediction errors made by the Markov Chain (MC) and SFDES-Based Fuzzy Markov Chain (FMC). NN, NP, PN, and PP stand for the Negative-Negative, Negative-Positive, Positive-Negative, and Positive-Positive x-value pair groups, respectively. The numbers in the parentheses indicate numbers of testing pairs.

VI. Design Considerations for SFDES-Based Fuzzy Markov Chains

The example Fuzzy Markov Chain provides evidence to support the central theme of this study - (1) an SFDES-Based Fuzzy Markov Chain can be developed based on a binary Markov chain, and (2) the resulting fuzzy Markov chain has the potential to outperform its binary counterpart in terms of prediction accuracy. We did not explore this fuzzy Markov chain with different design choices, which could potentially result in an even greater reduction in prediction errors.

One such a design choice is to use different trapezoidal fuzzy sets. By the MATLAB notations, the fuzzy sets in Fig. 2 are trapmf(x,[-11, -10, -1, 1]) for ”Negative” and trapmf(x,[-1, 1, 10, 11]) for ”Positive.” It is conceivable that one could use trapmf(x,[-11, -10, $-a$, a]), $a > 0$, for ”Negative” and trapmf(x,[$-a$, a, 10, 11]) for ”Positive” and conduct a search to find the best value of a that provides the most prediction error reduction. The search can be conducted manually or systematically (e.g., through a genetic algorithm).

Instead of the symmetrical fuzzy sets depicted in Fig. 2, one may opt for asymmetrical trapezoidal fuzzy sets like trapmf(x,[-11, -10, $-a$, a]) for "Negative" and trapmf(x,[$-b$, b, 10, 11]) for "Positive," $b > 0$, and conduct a search with the same objective of maximizing error reduction. The search would be more difficult and time consuming as two parameters are involved. An alternative option is to utilize a different type of fuzzy sets, such as the Gaussian type, whether symmetrical or asymmetrical. Lastly, distinct fuzzy sets and/or various types of fuzzy sets may be employed for different fuzzy automata. However, additional parameters will be involved, further complicating the search for optimal fuzzy sets.

Another design consideration for the SFDES-Based Fuzzy Markov Chain pertains to the magnitudes of the representative values γ_j^i in defuzzifier (5). In the above example, the middle points are used and no attempt is made to search for better representative values that could potentially produce even smaller prediction errors for the testing data. In general, if desired, the pursuit of such a goal can be undertaken for each individual fuzzy automaton through the least-squares approach outlined in Section IV-E. Alternatively, one may use the middle points as initial values and search, either manually or automatically, for optimal representative values that will minimize prediction errors. One systematic approach is to employ an evolutionary algorithm (e.g., genetic algorithm).

Last but not least, one may utilize an evolutionary algorithm to concurrently and systematically search for optimal parameters of the fuzzy sets and optimal representative values of the centroid defuzzifier.

In summary, we have leveraged the simulation example to address and discuss several design considerations. These issues are universal and, therefore, play a crucial role in the design of any fuzzy Markov chains introduced in this paper. These practical design considerations complement the theoretical developments presented in Section IV.

VII. Conclusion

Building upon the foundations of SFDES and supervised learning FDES presented in our recent papers, this paper introduces a fundamentally innovative and unique theory of SFDES-Based Fuzzy Markov Chains. Utilizing SFDES, a fuzzy Markov chain can be constructed from a traditional binary Markov chain. The event transition matrices of the fuzzy automata within SFDES can then be automatically learned from sample data. Despite the fuzzy nature of its states, the stochastic characteristics of the fuzzy Markov chain remain fully identical to those of the Markov chain. The fuzzy Markov chain encompasses the corresponding Markov chain as a special case and reverts to it when its states become binary.

The fuzzy Markov chain possesses the same capabilities as the conventional Markov chain to model and process random information. However, the incorporation of fuzzy states, fuzzy event transition matrices, and fuzzy inference empowers the fuzzy Markov chain to more effectively represent and process vague and imprecise information in states and events. This potentially surpasses the binary Markov chain in terms of model accuracy. This advantage can be important for many practical applications, especially in medicine, where a patient can be in multiple ambiguous clinical states simultaneously to different extents, and a treatment can transfer a patient to multiple states concurrently with varying degrees.

Like conventional Markov chains, fuzzy Markov chains are not blackbox models; their structure and information processing chain can be intuitively understood and easily checked by humans. This interpretability is especially crucial in medical applications, where patient safety is paramount. It also facilitates easier, quicker, and more cost-effective model design, development, refinement, and implementation.

We introduce the new theory in a mathematically rigorous manner. A detailed simulation example is provided to illustrate key design aspects and demonstrate the advantages of the fuzzy Markov chain over its binary counterpart. Additionally, important design issues are discussed.

A fuzzy Markov chain with N states has N^2 fuzzy event transition matrices to be learned by the supervised learning algorithm one by one. A sufficient amount of sample pairs must be available in order to learn each of the matrices. Therefore, constructing a fuzzy Markov chain may require more sample pairs than the corresponding binary Markov chain, which can be a limitation in practice.

One important issue worthy of future research is the conditions under which a fuzzy Markov chain can serve as a more accurate model than a binary Markov chain. This question is technically challenging and is of both theoretical and practical significance. We believe that the answer depends highly on the characteristics (e.g., distributions) of the values of the M variables.

Disease diagnosis and treatment will likely be a highly fruitful application area for the new technology. Another potential application area is the prediction of stochastic natural processes, such as weather. For instance, a fuzzy Markov chain has the inherent capability of simultaneously calculating and predicting both the probability of rain (e.g., 80%) and the extent of the rain (e.g., 2.7 mm) for a future time (e.g., tomorrow). Overall, the novel theory of SFDES-Based Fuzzy Markov Chains holds practical utility in modeling various systems across different industries, particularly within the field of biomedicine.

References

[1] R. E. Stanford, "The set of limiting distributions for a markov chain with fuzzy transition probabilities," *Fuzzy Sets and Systems*, vol. 7, no. 1, pp. 71–78, 1982.

[2] R. M. Kleyle and A. de Korvin, "Constructing one-step and limiting fuzzy transition probabilities for finite markov chains," *Journal of Intelligent & Fuzzy Systems*, vol. 6, no. 2, pp. 223–235, 1998.

[3] J. J. Buckley and E. Eslami, "Fuzzy markov chains: uncertain probabilities," *Soft computing. 2002 Vol. 9 Num. 1*, 2002.

[4] K. E. Avrachenkov and E. Sanchez, "Fuzzy markov chains and decision-making," *Fuzzy Optimization and Decision Making*, vol. 1, no. 2, pp. 143–159, 2002.

[5] M. Kurano, M. Yasuda, J.-i. Nakagami, and Y. Yoshida, "A fuzzy approach to markov decision processes with uncertain transition probabilities," *Fuzzy Sets and Systems*, vol. 157, pp. 2674–2682, 2006.

[6] G. Li and B. Xiu, "Fuzzy markov chains based on the fuzzy transition probability," in *The 26th Chinese Control and Decision Conference*, pp. 4351–4356, IEEE, 2014.

[7] J. C. F. Garcia, "Interval type-2 fuzzy markov chains: an approach," in *2010 Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 1–6, IEEE, 2010.

[8] R. Kleyle and A. De Korvin, "Transition probabilities for markov chains having fuzzy states," *Stochastic analysis and applications*, vol. 15, no. 4, pp. 527–546, 1997.
[9] M. Symeonaki and G. B. Stamou, "Theory of markov systems with fuzzy states," *Fuzzy Sets and Systems*, vol. 143, pp. 427–445, 2004.
[10] D. P. Filev and I. Kolmanovsky, "Generalized markov models for real-time modeling of continuous systems," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 983–998, 2014.
[11] D. P. Filev, I. Kolmanovsky, and R. Yager, *Developing fuzzy state models as Markov chain models with fuzzy encoding*, pp. 79–106. Cham, Switzerland: Springer, 2015.
[12] N. Li, I. Kolmanovsky, A. Girard, and D. Filev, "Fuzzy encoded markov chains: Overview, observer theory, and applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, pp. 116–130, 2021.
[13] H. Ying and F. Lin, "Stochastic fuzzy discrete event systems and their model identification," *IEEE Transactions on Cybernetics, in press*.
[14] F. Lin and H. Ying, "Modeling and control of fuzzy discrete event systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 4, pp. 408–415, 2002.
[15] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Springer Nature, 3rd ed., 2021.
[16] Y. Cao, M. Ying, and G. Chen, "State-based control of fuzzy discrete-event systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 410–424, 2007.
[17] F. Lin and H. Ying, "State-feedback control of fuzzy discrete-event systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 951–956, 2009.
[18] Y. Cao and M. Ying, "Supervisory control of fuzzy discrete event systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 2, pp. 366–371, 2005.
[19] D. Qiu, "Supervisory control of fuzzy discrete event systems: a formal approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 1, pp. 72–88, 2005.
[20] W. Deng and D. Qiu, "Supervisory control of fuzzy discrete-event systems for simulation equivalence," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 178–192, 2014.
[21] W. Deng and D. Qiu, "Bifuzzy discrete event systems and their supervisory control theory," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2107–2121, 2015.
[22] F. Lin and H. Ying, "Supervisory control of fuzzy discrete event systems under partial observation," *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 9, pp. 3161–3172, 2023.
[23] Y. Cao and M. Ying, "Observability and decentralized control of fuzzy discrete-event systems," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 2, pp. 202–216, 2006.
[24] W. Deng and D. Qiu, "State-based decentralized diagnosis of bi-fuzzy discrete event systems," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 854–867, 2016.
[25] A. Jayasiri, G. K. Mann, and R. G. Gosine, "Generalizing the decentralized control of fuzzy discrete event systems," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 4, pp. 699–714, 2011.
[26] X. Yin, "A belief-evolution-based approach for online control of fuzzy discrete-event systems under partial observation," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1830–1836, 2016.
[27] A. O. Mekki, F. Lin, and H. Ying, "On detectabilities of fuzzy discrete event systems," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 2, pp. 426–436, 2020.
[28] E. Kilic, "Diagnosability of fuzzy discrete event systems," *Information Sciences*, vol. 178, no. 3, pp. 858–870, 2008.
[29] F. Liu and D. Qiu, "Diagnosability of fuzzy discrete-event systems: A fuzzy approach," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 2, pp. 372–384, 2009.
[30] B. Benmessahel, M. Touahria, F. Nouioua, J. Gaber, and P. Lorenz, "Decentralized prognosis of fuzzy discrete-event systems," *Iranian Journal of Fuzzy Systems*, vol. 16, no. 3, pp. 127–143, 2019.
[31] T. Zhu, F. Liu, and C. Xiao, "Reliable fuzzy prognosability of decentralized fuzzy discrete-event systems and verification algorithm," *Information Sciences*, 2023.
[32] B. Benmessahel, M. Touahria, and F. Nouioua, "Predictability of fuzzy discrete event systems," *Discrete Event Dynamic Systems*, vol. 27, pp. 641–673, 2017.
[33] W. Deng, J. Yang, C. Jiang, and D. Qiu, "Opacity of fuzzy discrete event systems," in *2019 Chinese control and decision conference (CCDC)*, pp. 1840–1845, IEEE, 2019.
[34] W. Deng, D. Qiu, and J. Yang, "Opacity measures of fuzzy discrete event systems," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 9, pp. 2612–2622, 2020.
[35] M. Nie and W. W. Tan, "Theory of generalized fuzzy discrete-event systems," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 98–110, 2014.
[36] X. Du, H. Ying, and F. Lin, "Theory of extended fuzzy discrete-event systems for handling ranges of knowledge uncertainties and subjectivity," *IEEE Transactions on Fuzzy Systems*, vol. 17, pp. 316–328, 2008.
[37] S. Raczynski, "The space of models, semi-discrete events with fuzzy logic," in *Models for Research and Understanding: Exploring Dynamic Systems, Unconventional Approaches, and Applications*, pp. 207–228, Springer, 2023.
[38] S. Zhang and J. Chen, "Modelling of fuzzy discrete event systems based on a generalized linguistic variable and their generalized possibilistic kriple structure representation," in *Proceedings of the ICNC-FSKD 2022*, pp. 437–446, Springer, 2023.
[39] Y. Cao, Y. Ezawa, G. Chen, and H. Pan, "Modeling and specification of nondeterministic fuzzy discrete-event systems," *Decision Making Under Constraints*, pp. 45–58, 2020.
[40] F. Lin and H. Ying, "Modeling and control of probabilistic fuzzy discrete event systems," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 399–408, 2021.
[41] N. J. Khan, G. Ahamad, M. Naseem, and Q. R. Khan, "Fuzzy discrete event system (fdes): A survey," in *Renewable Power for Sustainable Growth: Proceedings of International Conference on Renewal Power (ICRP 2020)*, pp. 531–544, Springer, 2021.
[42] D. Qiu and F. Liu, "Fuzzy discrete-event systems under fuzzy observability and a test algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 3, pp. 578–589, 2008.
[43] H. Ying, F. Lin, and R. Sherwin, "Fuzzy discrete event systems with gradient-based online learning," in *2019 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, pp. 1–6, IEEE, 2019.
[44] H. Ying and F. Lin, "Online self-learning fuzzy discrete event systems," *IEEE Transactions on Fuzzy Systems*, vol. 28, pp. 2185–2194, 2019.
[45] H. Ying and F. Lin, "Learning fuzzy automaton's event transition matrix when post-event state is unknown," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 4993–5000, 2020.
[46] H. Ying and F. Lin, "Self-learning fuzzy automaton with input and output fuzzy sets for system modelling," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 2, pp. 500–512, 2023.
[47] F. Lin and H. Ying, "Supervised learning of multievent transition matrices in fuzzy discrete-event systems," *IEEE Transactions on Cybernetics*, vol. 53, no. 9, pp. 5596–5604, 2023.
[48] H. Ying and F. Lin, "Identification of single-event stochastic fuzzy discrete event systems: An equation-systems-based approach," *IEEE Transactions on Fuzzy Systems, in press*.
[49] H. A. Taha, *Operations Research: An Introduction*. Prentice Hall, sixth ed., 1997.

Hao Ying (S88-M90-SM97-F12-LF23) received his B.S. and M.S. degrees in electrical engineering from Donghua University, Shanghai, China, in February 1982 and July 1984, respectively. He obtained his Ph.D. in biomedical engineering from the University of Alabama at Birmingham in 1990. He is a Professor at the Department of Electrical and Computer Engineering, Wayne State University, USA. He has published two fuzzy control books and over 130 journal papers. His Google H-index is 52 and is ranked among 25,000 most-cited authors across all 22 scientific fields (176 subfields) selected from nearly seven million scientists worldwide. He receives the 2023 IEEE CIS Fuzzy Systems Pioneer Award, which is the highest level honor of the IEEE Computational Intelligence Society, with the citation "For fundamental contributions to model-free fuzzy control theory and its biomedical applications." He is serving as an Associate Editor or a Member of Editorial Board for 18 international journals, including the IEEE Transactions on Fuzzy Systems, and the IEEE Transactions on Systems, Man, and Cybernetics: Systems. He served as a Member of the Fellows Committee of both the IEEE Computational Intelligence Society and the IEEE Systems, Man, and Cybernetics Society. He served as Program Chair/Co-Chair for four international conferences and as a Program/Technical Committee Member for over 160 international conferences.

Feng Lin (S85-M88-SM07-F09) received his B.Eng. degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 1984 and 1988, respectively. He was a Post-Doctoral Fellow with Harvard University, Cambridge, MA, USA, from 1987 to 1988. Since 1988, he has been with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA, where he is currently a professor. His current research interests include discrete event systems, hybrid systems, robust control, and their applications in alternative energy, biomedical systems, and automotive control. He authored a book entitled “Robust Control Design: An Optimal Control Approach” and coauthored a paper that received a George Axelby outstanding paper award from the IEEE Control Systems Society. He was an associate editor of IEEE Transactions on Automatic Control.