Computation of Supremal Controllable Sublanguages and Infimal Controllable Superlanguages for Fuzzy Discrete Event Systems

Klaus Schmidt*, Yiannis Boutalis†*

* Lehrstuhl für Regelungstechnik, Universität Erlangen-Nürnberg, Erlangen, Germany
† Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece

***Abstract*— Recently, several approaches for the study of fuzzy discrete event systems (FDES) in a supervisory control context have been proposed. Although controllability of fuzzy languages and their implementation by a supervisor could be verified by algorithmic procedures, the problem of supervisor synthesis was only solved for the case of FDES with crisp states but fuzzy state transitions. In this paper, we present algorithms to compute the supremal controllable fuzzy sublanguage and the infimal controllable prefix-closed fuzzy superlanguage of a given fuzzy language for the general case of FDES with fuzzy states, fuzzy state transitions and fuzzy event controllability properties, and formally prove their correctness.**

I. Introduction

Discrete event systems are discrete in both time and state space. Also changes in the system state occur asynchronously and driven by events rather than by a clock. Examples for discrete event systems are manufacturing systems, networks, digital circuits, communication protocols, etc. [1].

In most of the published research on DES, where deterministic or nondeterministic automata are used, it is assumed that their states and state transitions are crisp, that is, no uncertainty arises either in the states or in the state transitions. There are, however, situations in which being in a state or moving to another state via state transitions is not certain but determined by a *possibility degree*.

A typical example given in [2] considers modeling a person's health status, that can simultaneously assume different states, e.g., "poor", "fair", "excellent". To this end, possibility degrees are employed to describe the possibility of being in each respective state and to transition between the states. Further examples for such modeling requirements are for example mobile robots in unstructured environments [3], intelligent vehicle control [4], and wastewater treatment [5].

To cope with this situation, crisp DES were extended to fuzzy DES (FDES) by incorporating the fuzzy set theory in a DES framework in [2] based on existing literature on *fuzzy finite automata* [6], [7]. The application of the supervisory control theory to FDES was established in [8], [9]. In these works, it is desired to restrict the behavior of a given FDES to a desired subbehavior by employing a supervisory controller. In [9], the case of FDES with only fuzzy state transitions is considered, while [8] assumes fuzzy states, fuzzy state transitions, and fuzzy event controllability properties. Both approaches develop basic controllability results [8], [9], observability results [10], [11] and extensions to decentralized control [12], [10] including algorithmic support.

In this paper, the case where a given specification behavior cannot be implemented by supervisory control is studied. First, *fuzzy canonical recognizers* are introduced as an appropriate representation of fuzzy languages. Based on this representation, we develop algorithms for both computing the supremal controllable subbehavior and the infimal controllable superbehavior of a given specification in the general modeling framework for FDES in [8]. The algorithms are proved to be correct, and an example FDES illustrates the computational procedures.

Supervisory control for FDES has already been studied in the literature [8], [9]. However, for the general modeling framework in [8], only the existence of the supremal controllable subbehavior and the infimal controllable superbehavior is proved without supplying an algorithmic procedure. Although such algorithms are provided in [9], that paper considers the case of only fuzzy state transitions which is less general than our modeling framework.

The paper is organized as follows. After providing basic notation in Section II, the algorithms for computing the supremal controllable subbehavior and the infimal controllable superbehavior are presented in Section III-B and Section III-C, respectively. A brief discussion and conclusions are given in Section IV.

II. Preliminaries

A. Fuzzy Sets

Each *fuzzy set* $\mathcal{A}$ is defined in terms of a *universal set* X by a *membership function* assigning to each element x of X a value $\mathcal{A}(x)$ in the interval [0, 1]. The support of a fuzzy set $\mathcal{A}$ is a crisp set defined as $\mathrm{supp}(\mathcal{A}) = \{x : \mathcal{A}(x) > 0\}$. We denote by $\mathcal{F}(X)$ the set of all fuzzy subsets of X. For any $\mathcal{A}, \mathcal{B} \in \mathcal{F}(X)$, we say that $\mathcal{A}$ is contained in $\mathcal{B}$ (or $\mathcal{B}$ contains $\mathcal{A}$), denoted by $\mathcal{A} \subseteq \mathcal{B}$, if $\mathcal{A}(x) \leq \mathcal{B}(x)$ for all $x \in X$. We say that $\mathcal{A} = \mathcal{B}$ if and only if $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{B} \subseteq \mathcal{A}$. A fuzzy set is said to be empty, denoted by $\mathcal{O}$, if its membership function is identically zero on X.

Let $\mathcal{A}, \mathcal{B} \in \mathcal{F}(X)$. The union $\mathcal{A} \cup \mathcal{B}$ of $\mathcal{A}$ and $\mathcal{B}$ is defined by the membership function $(\mathcal{A} \cup \mathcal{B})(x) = \mathcal{A}(x) \vee \mathcal{B}(x)$ for all $x \in X$; the intersection of $\mathcal{A}$ and $\mathcal{B}$, denoted by $\mathcal{A} \cap \mathcal{B}$, is given by the membership function $(\mathcal{A} \cap \mathcal{B})(x) = \mathcal{A}(x) \wedge \mathcal{B}(x)$ for all $x \in X$, where $\vee$ and $\wedge$ stand for the maximum and minimum operators respectively [13].

B. Formalism for Fuzzy Discrete Event Systems

The following definitions and theorems provide a formal framework for the study of fuzzy discrete event systems, adopting the formalism in [8], [11], [12].

Let the crisp state set of a DES consist of the states $p_1, p_2, \ldots, p_n$. Then each *fuzzy state* in the setting of FDES can be written as a vector $[a_1, a_2, \ldots, a_n]$, where $a_i \in [0, 1]$. This way, each fuzzy state can be considered as a possibility distribution or alternatively as a fuzzy set determining the degree a_i by which the system participates in each crisp state p_i, provided it is in the current fuzzy state. Similarly, a *fuzzy event* σ is characterized by a matrix $[a_{ij}]_{nxn}$, in which every element $a_{ij} \in [0, 1]$ indicates the possibility of a transition in the FDES from the current state p_i to the new state p_j when the event σ occurs.

A *fuzzy finite automaton* (FFA) is 5-tuple $G = (Q, \Sigma, \delta, q_0, Q_m)$, where Q is a set of fuzzy states; Σ consists of fuzzy events; q_0 is the initial fuzzy state; Q_m stands for the set of marking fuzzy states; $\delta : Q \times \Sigma \to Q$ is the state transition relation, which is defined by $\delta(q, \sigma) = q \odot \sigma$. Note that $\odot$ denotes the max-min operation in the sense that for $q = [a_1, a_2, \ldots, a_n] \in Q$ and $\sigma = [a_{ij}]_{nxn} \in \Sigma$, $q \odot \sigma = [\max_{l=1}^{n} \min\{a_l, a_{l1}\}, \cdots, \max_{l=1}^{n} \min\{q_l, a_{ln}\}]$

The *fuzzy languages* generated and marked by G are denoted by $\mathcal{L}_G$ and $\mathcal{L}_{G,m}$, respectively. With the set Σ^* of all strings of fuzzy events over Σ, they are defined as functions from Σ^* to [0, 1] as follows: For any $s = \sigma_1\sigma_2 \ldots \sigma_k \in \Sigma^*$,

$$\mathcal{L}_G(s) = \max_{i=1}^{n} q_0 \odot \sigma_1 \odot \sigma_2 \odot \ldots \odot \sigma_k \odot s_i^T, \tag{1}$$

$$\mathcal{L}_{G,m}(s) = \sup_{q \in Q_m} q_0 \odot \sigma_1 \odot \sigma_2 \odot \ldots \odot \sigma_k \odot q^T, \tag{2}$$

where s_i^T is the transpose of $s_i = [0 \ldots 1 \ldots 0]$ and 1 is in the ith place.

$\mathcal{L}_G(s)$ represents the degree of the string $s \in \Sigma^*$ being physically possible, or alternatively the degree by which this string belongs to $\mathcal{L}_G$. $\mathcal{L}_{G,m}(s)$ stands for the possibility of the same string being marked (recognized) by the fuzzy automaton G. Let $s \in \Sigma^*$ and any $\sigma \in \Sigma$. Then the following relation follows from Equation (1) [8].

$$\mathcal{L}_{G,m}(s\sigma) \leq \mathcal{L}_G(s\sigma) \leq \mathcal{L}_G(s). \tag{3}$$

It should be mentioned that the set of fuzzy states $\{\delta(q_0, s) | s \in \Sigma^*\}$ in any max-min automaton is finite [8][14] while it can be unbounded for, e.g., max-product automata.

Example 1: Let a FDES have an initial fuzzy state $q_0 = [0.7, 0.3] \in Q$ and two events $\sigma_1,\sigma_2 \in \Sigma$ with:

$$\sigma_1 = \begin{bmatrix} 0.6 & 0.5 \\ 0.2 & 0 \end{bmatrix} \text{ and } \sigma_2 = \begin{bmatrix} 0 & 0.4 \\ 0.5 & 0.3 \end{bmatrix}$$

Starting from the initial state, the respective subsequent fuzzy states of the FFA are iteratively computed by performing the max-min operation between the current state and any of the fuzzy events. For example,

$$q_0 \odot \sigma_1 = \begin{bmatrix} 0.7 & 0.3 \end{bmatrix} \odot \begin{bmatrix} 0.6 & 0.5 \\ 0.2 & 0 \end{bmatrix} = \begin{bmatrix} 0.6 & 0.5 \end{bmatrix},$$

$$q_0 \odot \sigma_1 \odot \sigma_2 = q_0 \odot \sigma_1 \odot \begin{bmatrix} 0 & 0.4 \\ 0.5 & 0.3 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.4 \end{bmatrix}$$

This procedure can be repetitively applied until no new states are obtained. A graphical representation of the finite max-min automaton G of the system derived in this way is shown in Fig. 1, where each fuzzy state is labeled with the respective vector of possibility degrees and the initial fuzzy state is indicated by an incoming arrow. Example possibility degrees for fuzzy strings computed using Equation (1) are

$$\mathcal{L}_G(\epsilon) = \max\{0.7, 0.3\} = 0.7, \mathcal{L}_G(\sigma_1) = 0.6, \mathcal{L}_G(\sigma_1\sigma_2) = 0.5.$$

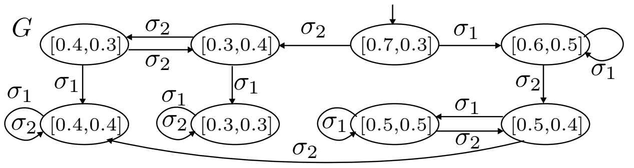

Fig. 1. Graphical representation of the FFA in Example 1.

C. Control of Fuzzy DES

In supervisory control, a fundamental issue is how to design a controller (or supervisor) whose task is to enable and disable controllable events such that the resulting closed-loop system obeys some pre-specified operating rules [1]. In the setting of FDESs, each fuzzy event $\sigma \in \Sigma$ is associated with a degree of controllability: the uncontrollable set Σ_{uc} is a fuzzy subsets of Σ such that $\Sigma_{\mathrm{uc}}(\sigma)$ expresses to which degree an event σ can occur without being disabled. Hence, e.g., $\Sigma_{\mathrm{uc}}(\sigma) = 0$ means that σ can be fully disabled while it is not possible at all to disable σ if $\Sigma_{\mathrm{uc}}(\sigma) = 1$.

A supervisor S for a FDES G is defined as a function $S : \Sigma^* \to \mathcal{F}(\Sigma)$, where for each $s \in \Sigma^*$ and each $\sigma \in \Sigma$, $S(s)(\sigma)$ represents the possibility of fuzzy event σ being enabled after the occurrence of a string s. Hence, $\min\{S(s)(\sigma), \mathcal{L}_G(s\sigma)\}$ can be interpreted as the degree to which the string $s\sigma$ can occur. Similar to the admissibility condition defined for crisp supervisors [1], S is usually required to satisfy that for any $s \in \Sigma^*$ and $\sigma \in \Sigma$

$$\min\{\Sigma_{\mathrm{uc}}(\sigma), \mathcal{L}_G(s\sigma)\} \leq S(s)(\sigma). \tag{4}$$

This condition is called the *fuzzy admissibility condition* of the supervisor S for a FDES G [8]. It states that a supervisor cannot restrict the possibility of an event occurrence σ after a string $s \in \Sigma^*$ more than specified by the uncontrollability degree $\Sigma_{\mathrm{uc}}(\sigma)$ of σ.

The fuzzy controlled system by S, denoted by S/G, is also a FDES, and the languages $\mathcal{L}_{S/G}$ and $\mathcal{L}_{S/G,m}$ generated and marked by S/G, are defined as follows: For any $s \in \Sigma^*$ and each $\sigma \in \Sigma$

$$\begin{aligned} \mathcal{L}_{S/G}(\epsilon) &= \mathcal{L}_G(\epsilon) \\ \mathcal{L}_{S/G}(s\sigma) &= \min\{\mathcal{L}_{S/G}(s), \mathcal{L}_G(s\sigma), S(s)(\sigma)\} \\ \mathcal{L}_{S/G,m} &= \mathcal{L}_{S/G} \cap \mathcal{L}_{G,m} \end{aligned} \tag{5}$$

For any string $t \in \Sigma^*$ and any fuzzy language $\mathcal{L} \subseteq \mathcal{F}(\Sigma^*)$, the *prefix-closure* $\mathrm{pr}(t)$ and $\mathrm{pr}(\mathcal{L})$ are defined as

$$\begin{aligned} \mathrm{pr}(t) &= \{s \in \Sigma^* | \exists r \in \Sigma^* \text{ s.t. } sr = t\} \\ pr(\mathcal{L})(s) &= \sup_{\{t | s \in pr(t)\}} \mathcal{L}(t) \end{aligned}$$

So $pr(\mathcal{L})(s)$ denotes the possibility of string s belonging to the prefix-closure of $\mathcal{L}$. The main task of supervisory control is to find a supervisor S that restricts the plant behavior modeled by a FFA G in order to comply with a specification behavior $K \subseteq \mathcal{F}(\Sigma^*)$, where $K \subseteq \mathcal{L}_G$. By means of the formulation of the above concepts, the controllability theorem concerning fuzzy DESs is stated below.

Theorem 1 (Controllability [8]): Let a FDES be modeled by FFA $G = (Q, \Sigma, \delta, q_0)$. Suppose the fuzzy uncontrollable subset $\Sigma_{\mathrm{uc}} \in \mathcal{F}(\Sigma)$, and the fuzzy legal subset $K \in \mathcal{F}(\Sigma^*)$ with $K \subseteq \mathcal{L}_G$ and $K(\epsilon) = \mathcal{L}_G(\epsilon)$ are given. Then there exists a supervisor $S : \Sigma^* \to \mathcal{F}(\Sigma)$ that satisfies the fuzzy admissibility condition and $\mathcal{L}_{S/G} = pr(\mathcal{K})$ if and only if for any $s \in \Sigma^*$ and any $\sigma \in \Sigma$

$$\min\{pr(K)(s), \Sigma_{\mathrm{uc}}(\sigma), \mathcal{L}_G(s\sigma)\} \leq pr(K)(s\sigma). \tag{6}$$

Here, (6) is called the *fuzzy controllability condition* of K with respect to $\mathcal{L}_G$ and Σ_{uc}. This condition indicates that the minimum degree of each string $s\sigma$ that can be achieved in the closed loop must be allowed by the specification.

Appealing to (6), we define the set of all controllable fuzzy sublanguages and superlanguages of K w.r.t. $\mathcal{L}_G$ and Σ_{uc} as $\mathcal{K}^{\uparrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}})$ and $\mathcal{K}^{\downarrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}})$, respectively.

$$\mathcal{K}^{\uparrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}}) = \{K' \subseteq K | K' \text{ is controllable w.r.t. } \mathcal{L}_G, \Sigma_{\mathrm{uc}}\}$$
$$\mathcal{K}^{\downarrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}}) = \{K' \supseteq K | K' \text{ is controllable w.r.t. } \mathcal{L}_G, \Sigma_{\mathrm{uc}}\}$$

The main task of this paper is the computation of controllable fuzzy sublanguages and superlanguages of a given specification K that is not controllable w.r.t. $\mathcal{L}_G$ and Σ_{uc}. A first result in this direction was provided in [8]. It states that the *supremal controllable fuzzy sublanguage* $K^{\uparrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}})$ and the *infimal prefix-closed fuzzy superlanguage* $K^{\downarrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}})$ exist. Furthermore, if K is prefix-closed, i.e., $K = \mathrm{pr}(K)$, then also $K^{\uparrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}}) = \mathrm{pr}(K^{\uparrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}}))$.

Proposition 1: Let G, K and Σ_{uc} be as above. Then

$$\begin{aligned} K^{\uparrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}}) &= \textstyle\bigcup_{K' \in \mathcal{K}^{\uparrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}})} K', \\ K^{\downarrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}}) &= \textstyle\bigcap_{K' \in \mathcal{K}^{\downarrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}})} K' \end{aligned}$$

In the following section, we provide algorithms for the computation of $K^{\uparrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}})$ and $K^{\downarrow}(\mathcal{L}_G, \Sigma_{\mathrm{uc}})$. For convenience, we will write $K^{\uparrow}$ and $K^{\downarrow}$ whenever $\mathcal{L}_G$ and Σ_{uc} are clear from the context.

III. Computation of Controllable Sublanguages and Superlanguages

In this section, we introduce *fuzzy canonical recognizers* as an alternative representation of fuzzy languages that is suitable for algorithmic computations by extending the *Nerode equivalence relation* [15] to fuzzy languages.

A. Nerode Equivalence for Fuzzy Languages

Definition 1 (Nerode Equivalence): Let $\mathcal{L} \subseteq \mathcal{F}(\Sigma^*)$ be a fuzzy language over Σ. The Nerode equivalence relation on Σ^* w.r.t. $\mathcal{L}$ (or $\mod \mathcal{L}$) is defined as follows. For $s, s' \in \Sigma^*$,

$$s \equiv s' \mod \mathcal{L} \Leftrightarrow \forall u \in \Sigma^* : \mathcal{L}(su) = \mathcal{L}(s'u).$$

Based on Definition 1, the *fuzzy canonical recognizer* (FCR) for a fuzzy language $\mathcal{L}$ can be defined as a five-tuple $C_{\mathcal{L}} = (\Sigma, X_{\mathcal{L}}, \nu_{\mathcal{L}}, x_{0,\mathcal{L}}, \chi_{\mathcal{L}})$. Here, Σ is the *alphabet*, and the set of *states* $X_{\mathcal{L}}$ corresponds to the set of equivalence classes of the Nerode equivalence relation on $\Sigma^* \mod \mathcal{L}$. Furthermore, the *initial state* x_0 is the equivalence class that contains the empty string ϵ. The *possibility function* $\chi_{\mathcal{L}} : X_{\mathcal{L}} \to [0, 1]$ relates each state $x \in X_{\mathcal{L}}$ to its possibility degree, and the *transition function* is defined as follows. For each $x \in X_{\mathcal{L}}$, let $s \in \Sigma^*$ s.t. s belongs to the equivalence class corresponding to x (in particular this means that $\mathcal{L}(s) = \chi_{\mathcal{L}}(x)$). Also, for each $\sigma \in \Sigma$, let x_σ be the state corresponding to the equivalence class of $s\sigma$. Then for each x and σ, the transition function is $\nu_{\mathcal{L}}(x, \sigma) := x_\sigma$.

Note that the FCR should not be confused with the FFA as defined in Section II-B. While the FFA is particularly useful for modeling purposes, the FCR rather represents the fuzzy language without any additional modeling information. However, it is possible to algorithmically construct a finite-state FCR that generates the same fuzzy language as a given FFA based on state minimization techniques as in [16]. Fig. 2 (a) depicts the FCR for the FFA G in Example 1. Here, each state is labeled with the associated possibility degree.

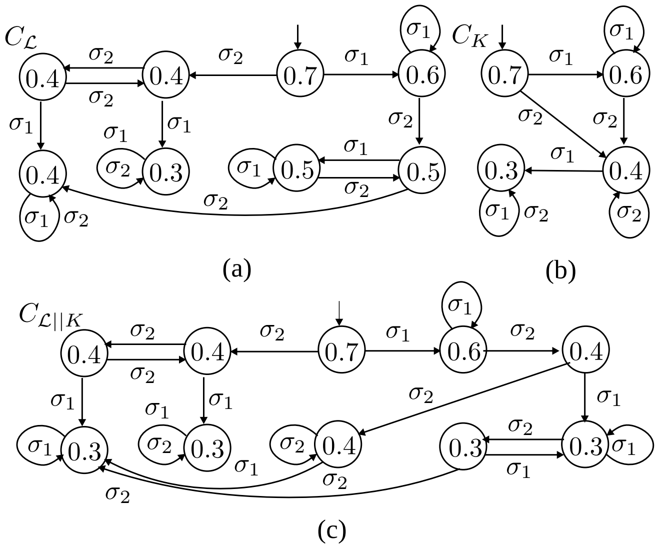

Fig. 2. Fuzzy Canonical Recognizers: (a) $C_{\mathcal{L}}$; (b) C_K; (c) $C_{\mathcal{L}||K}$

An important result of the classical supervisory control theory [17] states that the computation of the supremal controllable sublanguage of a given specification w.r.t. a plant language can be evaluated on the product state space of their respective canonical recognizers. Analogously, we define a *product composition* for FCRs, that will support the computation of the supremal controllable fuzzy sublanguage.

Definition 2 (Product Composition): Let $C_{\mathcal{L}}$, C_K be two FCRs for the fuzzy languages $\mathcal{L}$, K over the common alphabet Σ. The product composition $C_{\mathcal{L}||K} = (\Sigma, X_{\mathcal{L}||K}, \delta_{\mathcal{L}||K}, x_{0,\mathcal{L}||K}, \chi_{\mathcal{L}||K})$ is defined as follows

$$X_{\mathcal{L}||K} := X_{\mathcal{L}} \times X_K, \; x_{0,\mathcal{L}||K} := (x_{0,\mathcal{L}}, x_{0,K})$$

and for $(x_{\mathcal{L}}, x_K) \in X_{\mathcal{L}||K}$ and $\sigma \in \Sigma$:

$$\begin{aligned} &\chi_{\mathcal{L}||K}((x_{\mathcal{L}}, x_K)) := \min\{\chi_{\mathcal{L}}, \chi_K\} \\ &\delta_{\mathcal{L}||K}((x_{\mathcal{L}}, x_K), \sigma) := (\delta_{\mathcal{L}}(x_{\mathcal{L}}, \sigma), \delta_K(x_K, \sigma)) \end{aligned}$$

Example 2: Fig. 2 (c) shows the product composition $C_{\mathcal{L}||K}$ of the FCRs in Fig. 2 (a) and (b).

B. Supremal Controllable Fuzzy Sublanguage

In this section, a computational procedure to find the supremal controllable fuzzy sublanguage for a given plant G with the fuzzy language $\mathcal{L} := \mathcal{L}_G \subseteq \mathcal{F}(\Sigma^*)$ and the prefix-closed specification language $K \subseteq \mathcal{F}(\Sigma^*)$ with $K = \mathrm{pr}(K)$ is derived. Note that in this case, $\mathrm{pr}(K^{\uparrow}) = K^{\uparrow}$ as stated in Section II-C. To this end, we investigate how the specification K has to be modified in order to obtain $K^{\uparrow}$ if a controllability violation occurs according to (6), i.e. for a string $s \in \Sigma^*$ and an event $\sigma \in \Sigma$, we have that $K(s\sigma) < \min\{K(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}$.

Lemma 1: Let $s \in \Sigma^*$ and $\sigma \in \Sigma$. If $K(s\sigma) < K(s)$ and $K(s\sigma) < \mathcal{L}(s\sigma)$ and $K(s\sigma) < \Sigma_{\mathrm{uc}}(\sigma)$, then it holds that

$$K^{\uparrow}(s) \leq K(s\sigma) \tag{7}$$

Proof: Lemma 1 is proved by contradiction. Assume that $K^{\uparrow}(s) > K(s\sigma)$. We know that because of controllability of $K^{\uparrow}$ w.r.t. $\mathcal{L}$ and $\mathrm{pr}(K^{\uparrow}) = K^{\uparrow}$, $K^{\uparrow} = \mathcal{L}_{S/G}$ for some supervisor S. Then,

$$\begin{aligned} K^{\uparrow}(s\sigma) = \mathcal{L}_{S/G}(s\sigma) &= \min\{\mathcal{L}_{S/G}(s), \mathcal{L}(s\sigma), S(s)(\sigma)\} \\ &\geq \min\{K(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\} \\ &> K(s\sigma) \end{aligned}$$

by assumption of the above lemma. But this contradicts the fact that $K^{\uparrow}$ is a sublanguage of K. ■

That is, if neither the plant language nor a supervisor can enforce the possibility degree $K(s\sigma)$ after the string $s\sigma$, then already the desired possibility degree after the string s can maximally assume the value $K(s\sigma)$.

Based on the observation in Lemma 1, we define a language $\hat{K}_1 \subseteq \mathcal{F}(\Sigma^*)$ that fulfills the condition in Equation (7) and at the same time $\hat{K}_1 \subseteq K$.

Definition 3: Let $s \in \Sigma^*$. Then

$$\hat{K}_1(s) := \begin{cases} K(s) & \text{if } \forall \sigma \in \Sigma : K(s\sigma) \geq \\ & \min\{K(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\} \\ \min\limits_{\sigma \in \Sigma}\{K(s\sigma)\} & \text{otherwise} \end{cases} \tag{8}$$

From Lemma 1 and Definition 3 it follows immediately that $K^{\uparrow} \subseteq \hat{K}_1$. In addition to this result, it can be verified that a fuzzy recognizer of $\hat{K}_1$ can be determined using $C_{\mathcal{L}||K} := C_{\mathcal{L}}||C_K$. In particular, it holds that all strings s that lead to the same state in $C_{\mathcal{L}}||C_K$ also have the same value $\hat{K}_1(s)$.

Lemma 2: Let $s, s' \in \Sigma^*$ s.t. $\delta_{\mathcal{L}||K}(x_{0,\mathcal{L}||K}, s) = \delta_{\mathcal{L}||K}(x_{0,\mathcal{L}||K}, s')$. Then $\hat{K}_1(s) = \hat{K}_1(s')$.

Proof: Let $s, s' \in \Sigma^*$ s.t. $\delta_{\mathcal{L}||K}(x_{0,\mathcal{L}||K}, s) = \delta_{\mathcal{L}||K}(x_{0,\mathcal{L}||K}, s')$. Then $K(s) = K(s')$ and for all $\sigma \in \Sigma$, $K(s\sigma) = K(s'\sigma)$, $\mathcal{L}(s\sigma) = \mathcal{L}(s'\sigma)$, and $\Sigma_{\mathrm{uc}}(\sigma)$ is unique. According to Definition 3 this implies that $\hat{K}_1(s) = \hat{K}_1(s')$. ■

This means that for each state x in $X_{\mathcal{L}||K}$, all strings s that lead to x have the same possibility degree according to $\hat{K}_1$. As a consequence, a fuzzy recognizer $C_{\hat{K}_1}$ of $\hat{K}_1$ can be determined on the state space of $C_{\mathcal{L}||K}$ by setting $X_{\hat{K}_1} := X_{\mathcal{L}||K}$, $\delta_{\hat{K}_1} := \delta_{\mathcal{L}||K}$, $x_{0,\hat{K}_1} := x_{0,\mathcal{L}||K}$, and for each $x \in X_{\hat{K}_1}$ with $s \in \Sigma^*$ s.t. $\delta(x_{0,\hat{K}_1}, s) = x$,

$$\chi_{\hat{K}_1}(x) := \begin{cases} K(s) & \text{if } \forall \sigma \in \Sigma : K(s\sigma) \geq \\ & \min\{K(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\} \\ \min\limits_{\sigma \in \Sigma}\{K(s\sigma)\} & \text{otherwise} \end{cases}, \tag{9}$$

where in this case, $K(s) = \chi_{\mathcal{L}||K}(\delta(x_{0,\mathcal{L}||K}, s))$, $K(s\sigma) = \chi_{\mathcal{L}||K}(\delta(x_{0,\mathcal{L}||K}, s\sigma))$, and $\mathcal{L}(s\sigma) = \chi_{\mathcal{L}}(\delta(x_{0,\mathcal{L}}, s\sigma))$.

Considering the computational procedure, it has to be noted that by the second statement in (9), new controllability violations can be introduced. Hence, we propose to apply the above procedure iteratively until there are no more controllability violations (see Fig. 3). As in each step of the algorithm, Lemma 1 and Definition 3 are valid, it can be verified that $K^{\uparrow} \subseteq \hat{K} \subseteq K$ with the resulting fuzzy recognizer $C_{\hat{K}}$. Denoting the number of states of $C_{\mathcal{L}||K}$ as N and the number of different possibility degrees that are defined to describe G as P, $C_{\hat{K}}$ can be computed with complexity $\mathcal{O}(N^2 \cdot P)$.

/ Computation of $C_{\hat{K}}$: Initialization*/*
$i = 1; C_{\hat{K}_0} = C_{\mathcal{L}||K}$
Compute $C_{\hat{K}_1}$ from $C_{\hat{K}_0}$ according to Equation (9)
/ Computation of $C_{\hat{K}}$: Iteration */*
while $C_{\hat{K}_i} \neq C_{\hat{K}_{i-1}}$
    $i := i + 1$
    Compute $C_{\hat{K}_i}$ from $C_{\hat{K}_{i-1}}$ according to Equation (9)
end while
$C_{\hat{K}} = C_{\hat{K},i}$
return $C_{\hat{K}}$

Fig. 3. Algorithm for the Computation of $C_{\hat{K}}$

Example 3: In the control context, we assume that the controllability degrees of σ_1 and σ_2 are given as $\Sigma_{\mathrm{uc}}(\sigma_1) = 0.2$ and $\Sigma_{\mathrm{uc}}(\sigma_2) = 0.6$. $C_{\hat{K}}$ as determined from $C_{\mathcal{L}||K}$ in Fig. 2 is depicted in Fig. 4. Note that in the state that is shaded in gray, the original value of $K(\sigma_1) = 0.6$ had to be decreased to $\hat{K}(\sigma_1) = 0.4$ due to the controllability violation $K(\sigma_1\sigma_2) = 0.4 < \min\{K(\sigma_1), \mathcal{L}(\sigma_1\sigma_2), \Sigma_{\mathrm{uc}}(\sigma_2)\} = 0.5$.

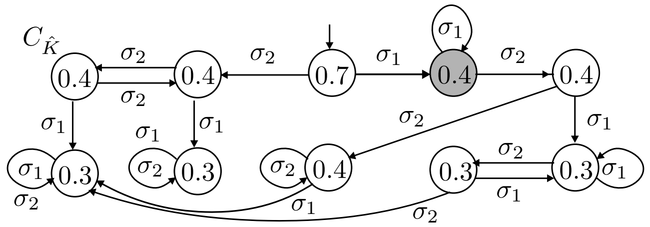

Fig. 4. Fuzzy Canonical Recognizer $C_{\hat{K}}$

Having determined the fuzzy recognizer $C_{\hat{K}}$ for the fuzzy language $\hat{K}$, we now define a supervisor that implements the possibility degree $\hat{K}(s)$ after each string $s \in \Sigma^*$ whenever a control action is needed and the possibility degree 1 otherwise.

Definition 4: Let $\hat{K}$ be defined as above. Then, the supervisor $S : \Sigma^* \to \mathcal{F}(\Sigma)$ for each $s \in \Sigma^*$ and $\sigma \in \Sigma$ is

$$S(s)(\sigma) := \begin{cases} \hat{K}(s\sigma) & \text{if } \hat{K}(s\sigma) < \hat{K}(s) \\ 1 & \text{otherwise.} \end{cases} \quad (10)$$

Note that $K^{\uparrow} \subseteq \hat{K}$ together with Definition 4 imply that for each string $s \in \Sigma^*$ and event $\sigma \in \Sigma$, it holds that $K^{\uparrow}(s\sigma) \leq S(s)(\sigma)$. Observing that $\hat{K}$ was computed from K by decreasing the possibility degrees of certain strings, it must also hold that $\hat{K} \subseteq K$. Together, this suggests that if S is fuzzy admissible and if the closed loop fuzzy language $\mathcal{L}_{S/G}$ is a subset of $\hat{K}$, then $\mathcal{L}_{S/G} = K^{\uparrow}$. The following theorem provides the desired result.

Theorem 2: Let the supervisor S be given as in Definition 4. S is fuzzy admissible and $\mathcal{L}_{S/G} \subseteq \hat{K}$ iff $\hat{K}(\epsilon) = \mathcal{L}(\epsilon)$.

Proof: "$\Rightarrow$": Assume that $\hat{K}(\epsilon) < \mathcal{L}(\epsilon)$.

$$\Rightarrow \hat{K}(\epsilon) < K(\epsilon) = \mathcal{L}(\epsilon)$$

$$\Rightarrow \exists \sigma \in \Sigma : \hat{K}(\sigma) < \min\{\hat{K}(\epsilon), \mathcal{L}(\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}$$

because of Definition 3 and the termination criterion of the algorithm in Fig. 3. Then $S(\epsilon)(\sigma) = \hat{K}(\sigma)$ according to Definition 4.

$$\Rightarrow S(\epsilon)(\sigma) = \hat{K}(\sigma) < \min\{\mathcal{L}(\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}$$

But this violates the admissibility of S.

"$\Leftarrow$": It is given that $\hat{K}(\epsilon) = \mathcal{L}(\epsilon)$. We have to show that $S(s)(\sigma) \geq \min\{\mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}$ for all $s \in \Sigma^*$ and $\sigma \in \Sigma$. This is true for $S(s)(\sigma) = 1$ in Equation (10). Hence, we only consider the case $S(s)(\sigma) = \hat{K}(s\sigma)$. But then, $S(s)(\sigma) \geq \min\{\mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}$ is implied by Definition 3. To show that $\mathcal{L}_{S/G} \subseteq \hat{K}$, we use induction. It holds that $\mathcal{L}_{S/G}(\epsilon) = \hat{K}(\epsilon) = \mathcal{L}(\epsilon)$. Now let $s \in \Sigma^*$ and $\sigma \in \Sigma$ and assume that $\mathcal{L}_{S/G}(s) \leq \hat{K}(s)$. We want to show that $\mathcal{L}_{S/G}(s\sigma) \leq \hat{K}(s\sigma)$. There are two cases: (i) $\hat{K}(s\sigma) < \hat{K}(s)$, (ii) $\hat{K}(s\sigma) \geq \hat{K}(s)$.

Case (i): Definition 4 implies that $S(s)(\sigma) = \hat{K}(s\sigma)$.

$$\begin{aligned} \Rightarrow \mathcal{L}_{S/G}(s\sigma) &= \min\{\mathcal{L}_{S/G}(s), S(s)(\sigma), \mathcal{L}_{S/G}(s)\} \\ &\leq S(s)(\sigma) = \hat{K}(s\sigma) \end{aligned}$$

Case (ii): $\mathcal{L}_{S/G}(s\sigma) \leq \mathcal{L}_{S/G}(s) \leq \hat{K}(s) \leq \hat{K}(s\sigma)$. ■

With the result in Theorem 2, it can finally be concluded that the supervisor S implements the supremal controllable sublanguage $K^{\uparrow}$.

Corollary 1: If the supervisor S is admissible, then the supremal controllable fuzzy sublanguage $K^{\uparrow}$ exists and is given by $K^{\uparrow} = \mathcal{L}_{S/G}$.

Proof: As S is admissible, $\mathcal{L}_{S/G}$ is fuzzy controllable w.r.t. $\mathcal{L}$. Furthermore, as $\mathcal{L}_{S/G} \subseteq \hat{K}$ according to Theorem 2, also $\mathcal{L}_{S/G} \subseteq K$. Hence, $\mathcal{L}_{S/G} \subseteq K^{\uparrow}$. On the other hand, because of Definition 4, $K^{\uparrow}(s\sigma) \leq S(s)(\sigma)$ for all $s \in \Sigma^*$ and $\sigma \in \Sigma$. As $K^{\uparrow}(s) \leq \mathcal{L}(s)$ for all $s \in \Sigma^*$, this implies that $K^{\uparrow} \subseteq \mathcal{L}_{S/G}$. Together it must hold that $K^{\uparrow} = \mathcal{L}_{S/G}$. ■

This means that the computational procedure carried out on the state space of the FCR $C_{\mathcal{L}||K}$ yields a supervisor S that implements the supremal controllable fuzzy sublanguage in conjunction with the given plant G. However, different from both the classical supervisory control theory with crisp states and events and the less general modeling framework with fuzzy state transitions in [9], the recognizer $C_{\hat{K}}$ resulting from the supervisor computation does not necessarily recognize $K^{\uparrow}$ but it rather holds that $K^{\uparrow} \subseteq \hat{K}$.

Example 4: For our example control problem, the required supervisor actions are illustrated by the bold arrows in Fig. 4, i.e., $S(\epsilon)(\sigma_1) = 0.4$, $S(\sigma_1^*\sigma_2)(\sigma_1) = 0.3$, $S(\sigma_2\sigma_2(\sigma_2\sigma_2)^*)(\sigma_1) = 0.3$, and $S(\sigma_1^*\sigma_2\sigma_2)(\sigma_1) = 0.3$. For all remaining strings s and events σ, it holds that $S(s)(\sigma) = 1$. It can also be verified that in this case, $\mathcal{L}_{S/G} = \hat{K}$.

C. Infimal Controllable Fuzzy Superlanguage

In this section, we establish an algorithm for the computation of the infimal prefix-closed controllable fuzzy superlanguage $K^{\downarrow}$ given a plant G with the fuzzy language $\mathcal{L} := \mathcal{L}_G \subseteq \mathcal{F}(\Sigma^*)$ and the specification language $K \subseteq \mathcal{F}(\Sigma^*)$ with $K = \mathrm{pr}(K)$. In order to prepare the desired result, we first derive a property of $K^{\downarrow}$ that relates the possibility degree of a string to the possibility degrees of its prefixes.

Lemma 3: Assume that $K^{\downarrow}(s)$ is known for a string $s \in \Sigma^*$. Then it holds for each $\sigma \in \Sigma$ that

$$K^{\downarrow}(s\sigma) = \max\{K(s\sigma), \min\{K^{\downarrow}(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}\} \quad (11)$$

Proof: As $K^{\downarrow}$ is controllable w.r.t. $\mathcal{L}$ and Σ_{uc}, we know that there is an admissible supervisor S such that $\mathcal{L}_{S/G} = K^{\downarrow}$. Employing this supervisor, we prove Lemma 3 by contradiction.

First assume that $K^{\downarrow}(s\sigma) > \max\{K(s\sigma), \min\{K^{\downarrow}(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}\}$. Now let $\tilde{S}$ be a supervisor such that $\tilde{S}(s')(\sigma) = S(s')(\sigma')$ for all $\sigma' \in \Sigma$ and $s'\sigma' \in \Sigma^* - \{s\sigma\}$ and $\tilde{S}(s)(\sigma) = \max\{K(s\sigma), \min\{K^{\downarrow}(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}\}$. Then admissibility of $\tilde{S}$ already holds for all $s' \in \Sigma^* - \{s\}$. To show admissibility of $\tilde{S}$ for s, the event σ has to be considered. It can be observed that $\tilde{S}(s)(\sigma) \geq K(s\sigma)$ and $\tilde{S}(s)(\sigma) \geq \min\{K^{\downarrow}(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}$. As $K^{\downarrow}(s) \geq K^{\downarrow}(s\sigma)$ and $K^{\downarrow}(s\sigma) > \min\{K^{\downarrow}(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}$, this means that $\tilde{S}(s)(\sigma) \geq \min\{\mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}$ which obeys the admissibility condition in Equation (4). Hence, it holds that $K \subseteq \mathcal{L}_{\tilde{S}/G} \subset \mathcal{L}_{S/G} = K^{\downarrow}$ and hence $K^{\downarrow}$ is not the infimal prefix-closed fuzzy superlanguage.

Now assume that $K^{\downarrow}(s\sigma) < \max\{K(s\sigma), \min\{K^{\downarrow}(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}\}$.

$$\Rightarrow K^{\downarrow}(s\sigma) < \min\{K^{\downarrow}(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}$$

as $K^{\downarrow}(s\sigma) < K(s\sigma)$ is not possible.

$$\Rightarrow \mathcal{L}_{S/G}(s\sigma) < \min\{K^{\downarrow}(s), \mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}$$

Because of Equation (5), $\mathcal{L}_{S/G}(s\sigma) = \min\{\mathcal{L}_{S/G}(s), \mathcal{L}(s\sigma), S(s)(\sigma)\}$ with $\mathcal{L}_{S/G}(s) = K^{\downarrow}(s)$.

$$\Rightarrow S(s)(\sigma) < \min\{\mathcal{L}(s\sigma), \Sigma_{\mathrm{uc}}(\sigma)\}.$$

According to (4) this violates the admissibility of S. ■

The result in Lemma 3 is now employed to construct a fuzzy recognizer $C_{K^{\downarrow}} = (\Sigma, X_{K^{\downarrow}}, \nu_{K^{\downarrow}}, x_{0,K^{\downarrow}}, \chi_{K^{\downarrow}})$ for the

/ Computation of $C_{K^\downarrow}$: Initialization */*
$x_{0,K^\downarrow} = (x_{0,\mathcal{L}}, x_{0,K}, \mathcal{L}(\epsilon))$, $X_{K^\downarrow} = \{x_{0,K^\downarrow}\}$, $\chi_{K^\downarrow}(x_{0,K^\downarrow}) = \mathcal{L}(\epsilon)$
/ set of states to be processed */*
$X_{\mathrm{waiting}} = \{x_{0,K^\downarrow}\}$
/ Computation of $C_{K^\downarrow}$: Iteration */*
while $X_{\mathrm{waiting}} \neq \emptyset$
  take an arbitrary element $\hat{x} = (x_{\mathcal{L}}, x_K, d)$ from X_{waiting}
  for all $\sigma \in \Sigma$
    $r = \max\{\chi_K(\nu_K(x_K,\sigma)),$
      $\min\{\chi_{K^\downarrow}(\hat{x}), \chi(\nu_{\mathcal{L}}(x_{\mathcal{L}},\sigma)), \Sigma_{\mathrm{uc}}(\sigma)\}\}$ (*)
    $\hat{x}' = (\nu_{\mathcal{L}}(x_{\mathcal{L}},\sigma), \nu_K(x_K,\sigma), r)$
    if $\hat{x}' \notin X_{K^\downarrow}$
      $X_{K^\downarrow} = X_{K^\downarrow} \cup \{\hat{x}'\}$
      $X_{\mathrm{waiting}} = X_{\mathrm{waiting}} \cup \{\hat{x}'\}$
      $\chi_{K^\downarrow}(\hat{x}') = r$
    end if
    $\nu_{K^\downarrow}(\hat{x},\sigma) = \hat{x}'$
  end for all
end while
return($C_{K^\downarrow}$)

Fig. 5. Algorithm for the Computation of $C_{K^\downarrow}$

fuzzy language $K^\downarrow$ from the two FCRs $C_{\mathcal{L}}$ and C_K. Note that the value $K^\downarrow(\epsilon) = K(\epsilon) = \mathcal{L}(\epsilon)$ is given by definition.

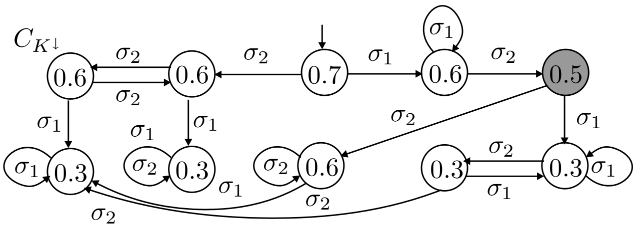

Fig. 6. Fuzzy Canonical Recognizer $C_{K^\downarrow}$

The algorithm in Fig. 5 iteratively evaluates (11) in line (*). Here, $\chi_K(\nu_K(x_K,\sigma))$ corresponds to $K(s\sigma)$, $\chi_{K^\downarrow}(\hat{x})$ corresponds to $K^\downarrow(s)$, and $\chi(\nu_{\mathcal{L}}(x_{\mathcal{L}},\sigma))$ corresponds to $\mathcal{L}(s\sigma)$. Each state $(x_{\mathcal{L}}, x_K, e)$ in $C_{K^\downarrow}$ is related to the state $x_{\mathcal{L}}$ in $C_{\mathcal{L}}$, the state x_K in C_K, and the possibility degree d. Note that generally the state space of $C_{K^\downarrow}$ is larger than the state space of $C_{\mathcal{L}||K}$, i.e., for each state $(x_{\mathcal{L}}, x_K)$ in $C_{\mathcal{L}||K}$ there may exist multiple states $(x_{\mathcal{L}}, x_K, \bullet)$ in $C_{K^\downarrow}$. With the number of states N of $C_{\mathcal{L}||K}$ and the number of events E, the algorithm has a computational complexity of $\mathcal{O}(N \cdot E)$. The following theorem states that $K^\downarrow$ as computed by the above algorithm is indeed the infimal prefix-closed controllable fuzzy superlanguage.

Theorem 3: The fuzzy language $K^\downarrow$ as computed in Fig. 5 is the infimal prefix-closed controllable fuzzy superlanguage of K w.r.t. $\mathcal{L}$ and Σ_{uc}.

Proof: The correctness of Theorem 3 follows from the iterative application of Equation (11) in each step of the algorithm starting from $\chi_{K^\downarrow}(x_{0,K^\downarrow}) = K^\downarrow(\epsilon) = K(\epsilon)$. ■

Example 5: Fig. 6 depicts the FCR $C_{K^\downarrow}$ for the FFA G in Example 1 and the fuzzy specification language K in Fig. 2. Here, the value of $K(\sigma_1\sigma_2) = 0.4$ had to be increased to $K^\downarrow(\sigma_1\sigma_2) = 0.5$ in the state shaded in gray.

IV. Conclusions

A framework for the supervisory control of Fuzzy Discrete Event Systems (FDES) with fuzzy states, fuzzy state transitions, and fuzzy event controllability properties has been established in [8]. In this paper, the methodology has been extended with algorithmic procedures for the computation of the *supremal controllable fuzzy sublanguage* and the *infimal controllable prefix-closed superlanguage* of a given specification. To this end, *fuzzy canonical recognizers* have been introduced as an appropriate representation of fuzzy languages, and the presented algorithms have been formulated based on this representation. Future work includes the computation of observable sublanguages in the case of limited event observability as studied in [11].

References

[1] C. G. Cassandras and S. Lafortune, "Introduction to discrete event systems," *Kluwer*, 1999.
[2] F. Lin and H. Ying, "Modeling and control of fuzzy discrete event systems," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 32, no. 4, pp. 408–415, August 2002.
[3] H. M. Barbera and A. G. Skarmeta, "A framework for defining and learning fuzzy behaviors for autonomous mobile robots," *Int. J. Intell. Syst.*, vol. 17, pp. 1–20, 2002.
[4] G. G. Rigatos, "Fuzzy stochastic automata for intelligent vehicle control," *IEEE Trans. Ind. Electron.*, vol. 50, no. 1, pp. 76–79, February 2003.
[5] J. Waissman, R. Sarrate, T. Escobet, J. Aguilar, and B. Dahhou, "Wastewater treatment process supervision by means of a fuzzy automaton model," in *Proceedings of the 2000 IEEE International Symposium on Intelligent Control*, Patras, Greece, 2000.
[6] W. Wechler, *The Concept of Fuzziness in Automata and Language Theory*. Berlin: Akademie-Verlag, 1978.
[7] J. N. Mordeson and D. S. Malik, *Fuzzy Automata and Languages: Theory and Applications*. London, Boca Raton, FL: Chapman & Hall, CRC, 2002.
[8] D. Qiu, "Supervisory control of fuzzy discrete event systems: A formal approach," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 35, no. 1, pp. 72–88, February 2005.
[9] Y. Cao and M. Ying, "Supervisory control of fuzzy discrete event systems," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 35, no. 2, pp. 366–371, 2005.
[10] ——, "Observability and decentralized control of fuzzy discrete-event systems," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 2, pp. 202–216, April 2006.
[11] D. Qiu and F. Liu, "Fuzzy discrete event systems under fuzzy observability and a test-algorithm," 2006. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0605107
[12] F. Liu and D. Qiu, "Decentralized supervisory control of fuzzy discrete event systems," in *European Control Conference*, 2007.
[13] L. A. Zadeh, "Fuzzy logic=computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 103–11, 1996.
[14] F. Steimann and K. P. Adlassning, "Clinical monitoring with fuzzy automata," *Fuzzy Sets and Systems*, vol. 61, pp. 37–42, 1994.
[15] J. E. Hopcroft and J. D. Ullman, "Introduction to automata theory, languages and computation," *Addison-Wesley*, 1979.
[16] J. E. Hopcroft, "An $n \log n$-algorithm for minimizing the states in a finite automaton," *In Z. Kohavi, editor, The theory of machines and computations, Academic Press*, pp. 189–196, 1971.
[17] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings IEEE, Special Issue Discrete Event Dynamic Systems*, vol. 77, pp. 81–98, 1989.