Mobile Robot Navigation based on Fuzzy Discrete Event Systems

Georgios Tampakis * **Klaus Schmidt** ** **Yiannis Boutalis** *,**

* *Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece. (e-mail:{gtampaki,ybout}@ee.duth.gr)*
** *Lehrstuhl für Regelungstechnik, Universität Erlangen-Nürnberg, Erlangen, Germany, (e-mail: klaus.schmidt@rt.eei.uni-erlangen.de)*

Abstract: Recently, several approaches for the control of fuzzy discrete event systems (FDES) have been proposed. First results towards the use of FDES in mobile robot navigation have also been presented, which however mainly build on sensory information processing. In this paper, we develop a methodology to compute control actions for the navigation of a mobile robot based on distributed FDES. The FDES description permits to take into account possible uncertainties in sensory information and enables a prediction of the future behavior of the robot depending on potential control actions. This prediction can then be used to select the most appropriate control action in each time instant. Our approach is tested by simulations of a mobile robot that encounters unknown obstacles on a pre-defined path.

Keywords: Fuzzy control, discrete event system, lookahead policy, mobile robot navigation.

1. INTRODUCTION

Discrete event systems (DES) 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 (Cassandras and Lafortune, 1999). In most of the published research on DES, 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. On the contrary, in Fuzzy Discrete Event Systems (FDES), being in a state or moving to another state via state transitions is not certain but determined by a *possibility degree*.

In (Lin and Ying, 2002) the formal framework for crisp DES was extended to FDES by incorporating the fuzzy set theory in a DES framework based on existing work on *fuzzy finite automata* (Wechler, 1978; Mordeson and Malik, 2002). Considering the control of FDES, different perspectives are taken in the literature. On the one hand, approaches such as (Qiu, 2005; Cao and Ying, 2005; Schmidt and Boutalis, 2008) investigate how the behavior of an FDES represented by a *fuzzy language* can be restricted in order to fulfill a given specification. On the other hand, an optimal control approach for FDES that employs fuzzy finite automata models is presented in (Lin and Ying, 2002). In this paper, we also develop an optimal control strategy based on a state representation of FDES.

DES were first used by Kosecka and Bajcy (1994) as a behavior arbitration technique in mobile robots. The DESbased methods employ sensor information to identify a new state of the robotic system. However, DES may lead to erroneous decision making if the sensor information is imprecise. The vagueness of sensor data can be better represented using *fuzzy logic* (FL) to describe the possibility degree of being in each state and to transition between the states (Barbera and Skarmeta, 2002). Hence, FDES facilitate the integration of DES and FL for implementing a control scheme under uncertain conditions.

FDES have been recently used in (Huq et al., 2006b,a) for robotic sensor information processing and robotic behavior modulation. To cope with multiple sensor data, distributed FDES with *fuzzy states* and *fuzzy events* are defined and the possibility of occurrence of an event varies according to the imprecision and uncertainty in the sensor data. However, the proposed behavior modulation and the selected control actions do not arise directly from an underlying FDES and its specifications but it rather emerges after a heuristic exploitation of the information gained from the distributed FDES. That is why the approach requires an additional post-decision evaluation.

In this paper, a new scheme for mobile robot navigation based on FDES is introduced. The proposed formulation clearly separates the sensor evaluation and the computation of control actions. In the first step, distributed subsystem models are employed to process uncertain sensor information. Then, a further set of FDES models allows the prediction of the future system state when applying pre-defined control actions. In the last step, these FDES models are evaluated in each sampling instant of the system evolution in order to apply the best instantaneous control action. The proposed approach is verified by simulations of a mobile robot, which has to follow a specified path in an environment with unspecified obstacles.

The paper is organized as follows. After providing basic notation in Section 2, we present the sensor evaluation and the state prediction for FDES in Section 3 and 4, respectively. The computation of optimal control actions is elaborated in Section 5 in conjunction with a mobile robot example. Conclusions are given in Section 6.

2. PRELIMINARIES

2.1 Fuzzy Sets

Each *fuzzy set* $\mathcal{A}$ is defined in terms of a *universal set* X by a *membership function* assigning to each element $x \in X$ a value $\mathcal{A}(x)$ in the interval [0, 1]. 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 $\mathcal{B}$, 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 empty, denoted by $\mathcal{O}$, if its membership function is identically zero on X.

2.2 Formalism for Fuzzy Discrete Event Systems

This section introduces fuzzy finite automata as a formal model of FDES in the formalism of (Lin and Ying, 2002).

Let the crisp state set $\mathcal{P}$ of a DES consist of the states $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$. Then each *fuzzy state* in the setting of FDES can be written as a vector $q = [q_1, q_2, \ldots, q_n]$, where $q_i \in [0, 1]$. This way, each fuzzy state can be considered as a possibility distribution or alternatively as a fuzzy set $q \subseteq \mathcal{F}(\mathcal{P})$ determining the degree q_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. Using the above notation, the successor fuzzy state can be evaluated using the max-min operation $\odot$ (Qiu, 2005; Schmidt and Boutalis, 2008) such that

$$q \odot \sigma = \left[\max_{l=1}^{n} \min\{q_l, a_{l1}\}, \cdots, \max_{l=1}^{n} \min\{q_l, a_{ln}\} \right]. \quad (1)$$

We use a generalized version of the formalism in (Huq et al., 2006a) to describe the behavior of a distributed FDES which is applicable to the sensing problem addressed in Section 3 and the control problem in Section 5. Each FDES models the occurrence of an event σ starting from an *initial fuzzy state* q_σ by a tuple $F_\sigma = (Q_\sigma, \Phi_\sigma, q_\sigma)$ with the *set of crisp states* Q_σ, with the dimension $|Q_\sigma| = n$. Accounting for the distributed nature of the FDES, the event σ is associated with the *fuzzy event set* Φ_σ in the sense that all events in Φ_σ contribute to the occurrence of σ: given the inital fuzzy state $q_\sigma \subseteq \mathcal{F}(Q_\sigma)$, the successor fuzzy state $\hat{q}_\sigma$ after the occurrence of σ evaluates to

$$\hat{q}_\sigma := \max_{f \in \Phi_\sigma} [q_\sigma \odot f], \quad (2)$$

where max takes the component-wise maximum of the respective successor state vectors. We write $\hat{q}_\sigma = q_\sigma \otimes \sigma$.

Example 1. We study the FDES $F_\sigma = (Q_\sigma, \Phi_\sigma, q_\sigma)$ with three crisp states ($Q_\sigma = \{q_1, q_2, q_3\}$), $\Phi_\sigma = \{\mathtt{f}^1_\sigma, \mathtt{f}^2_\sigma\}$, $q_\sigma = [0.3\ 0.8\ 0.4]$ and the fuzzy event matrices

$$\mathtt{f}^1_\sigma = \begin{bmatrix} 0 & 0.2 & 0.7 \\ 0 & 0.8 & 0 \\ 0.4 & 0 & 0.8 \end{bmatrix}, \qquad \mathtt{f}^2_\sigma = \begin{bmatrix} 0 & 0.4 & 0 \\ 0.6 & 0 & 0.9 \\ 0 & 0.8 & 0 \end{bmatrix}$$

Then, the evaluation of $q_\sigma \otimes \sigma$ yields

$\hat{q}_\sigma = q_\sigma \otimes \sigma = \max([0.4\ 0.8\ 0.4], [0.6\ 0.4\ 0.8]) = [0.5\ 0.6\ 0.6]$.

Finally, we introduce the *fuzzy state product* for two fuzzy states q, p with the dimension n as

$$q \oslash p := \big[\min\{q_1, p_1\} \cdots \min\{q_1, p_n\}\ \min\{q_2, p_1\} \\ \min\{q_n, p_1\} \cdots \min\{q_n, p_n\} \big].$$

Hence, $q \oslash p$ describes the fuzzy state corresponding to the product state space of the crisp states of q and p.

Example 2. For example, let $q = [0.5\ 0.6\ 0.6]$ and $p = [0.6\ 0.4\ 0.4]$. Then, $q \oslash p = \big[\min\{0.5, 0.6\} \cdots \min\{0.5, 0.4\} \cdots \min\{0.6, 0.4\} \big] = \big[0.5\ 0.4\ 0.4\ 0.6\ 0.4\ 0.4\ 0.6\ 0.4\ 0.4\big]$.

3. SENSOR INFORMATION PROCESSING USING DISTRIBUTED FDES

3.1 Approaches for Sensor Data Evaluation

There exist three broad categories of intelligent multiple sensor data processing for robotic control: *centralized*, *behavior-based*, and *hybrid* strategies (Rosenblatt and Handler, 1999). It is discussed in the literature that the latter seems to be the most appropriate because it alleviates the drawbacks of the first two, namely the significant delays introduced by the centralized processing and the conflicting sequences of actions triggered by different sensor data associated with independent behaviors.

Fuzzy logic (FL)-based approaches (Pirjanian and Mataric, 1999; P. Vadakkepat and Lee, 2004), are able to cope with uncertainties in sensory data and environment dynamics, but they suffer from the drawback of forming a large rule-base for complex behavior-based systems. Moreover, existing FL-based techniques do not provide a clearly structured framework for intelligent robotic control and lack the capability of system analysis. To cope with this situation (Huq et al., 2006a,b) propose a distributed FDES approach for the processing of multiple robotic sensor data. However, their approach depends on a heuristic weighting mechanism for control actions.

In our approach, robot navigation is based entirely on a DES framework and comprises two components. In the first component the multiple sensory information is used by distributed FDES, which determine the current state of the robot in respect to the pre-specified goals. The current state is then the initial information of the second FDES based component, which determines the control actions. In this section the first component is described, interweaved with a simple navigation example.

3.2 FDES Framework for Sensor Information Processing

In our framework, sensor data are evaluated with respect to several *goals* that have to be achieved by control. We consider the case, where for each element of the *set of goals* $\mathcal{G} = \{1, \ldots, p\}$, multiple *measurements* have to be evaluated. To this end, we introduce a *set of measurements* Σ_g and FDES models $S_{g,\sigma}$ that characterize the sensor evaluation for the measurement $\sigma \in \Sigma_g$ w.r.t. the goal $g \in \mathcal{G}$. Then, we perform the computation of the fuzzy state q_g in each sampling period w.r.t. to the objective $g \in \mathcal{G}$ as $q_g := \oslash_{\sigma \in \Sigma_g} (q_\sigma \otimes \sigma) = \oslash_{\sigma \in \Sigma_g} \hat{q}_\sigma$. In this computation, the fuzzy state product allows the combination of multiple independent measurements to an overall characterization of the current system state w.r.t. to each objective $g \in \mathcal{G}$. Using this approach, the main effort of the system designer is to determine the fuzzy event matrices related to the measurements based on the available sensor data.

3.3 FDES Modeling for a Mobile Robot

We now illustrate the formal framework by a mobile robot example. We consider a mobile robot that moves in a known environment and tries to reach a pre-specified target, where the path to be followed is determined using a path planning technique (details will be given in Section 5). Furthermore, the environment can change dynamically, in the sense that obstacles may be unexpectedly encountered. Therefore, there are two goals for the robot: it has to follow a given path in order to reach the target (goal 1), while avoiding any encountered obstacle (goal 2).

Figure 1 describes this situation, where bold dots denote path vertices (i.e., subgoals of the robot). Walls or other objects can be considered as obstacles. Based on sensor data, the distance d_{S} to the nearest subgoal, the angle a_{S} between the heading direction of the robot and the direction pointing to the next subgoal, the distance d_{O} to the nearest obstacle, and the angle a_{O} between the moving direction of the robot and the direction of the line connecting the center of the robot with the obstacle can be determined. Regarding goal 1, the two first values (d_{S},a_{S}) characterize the state of the robot. Similarly, d_{O} and a_{O} represent the state of the robot in respect to goal 2.

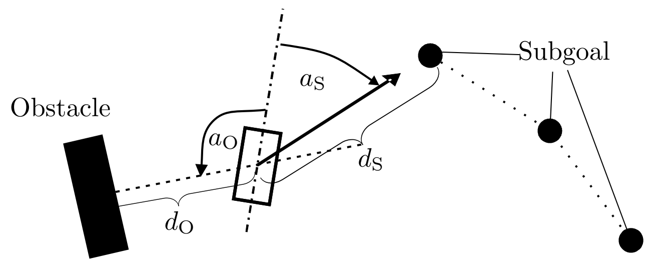

Fig. 1. Mobile robot with subgoals and obstacle.

In order to conform with the simulation example in Section 5.2, we measure distance values in numbers of pixels in an image area of 200×400 pixels. With an area of 20mm $\times$ 20mm per pixel, each image represents an area of 4m $\times$ 8m. The angle measurement is performed in degrees.

According to Section 3.2, we now develop separate FDES models that capture the sensor evaluation for d_{S}, a_{S}, d_{O} and a_{O}, and then compose the models to obtain a fuzzy state evaluation of the global position of the robot.

Path Following (Goal 1) We associate the events δ_{S} and α_{S} to the measurement of d_{S} and a_{S}, respectively. Hence, $\Sigma_1 = \{\delta_{\mathrm{S}}, \alpha_{\mathrm{S}}\}$. The FDES models are $S_{1,\delta_{\mathrm{S}}}$ and $S_{2,\alpha_{\mathrm{S}}}$.

For $S_{1,\delta_{\mathrm{S}}}$, we choose 3 crisp states (state set $Q_{1,\delta_{\mathrm{S}}}$) to characterize the distance d_{s} as shown in Fig. 2 (a), where state 1 corresponds to a high value, state 2 corresponds to a medium value and state 3 corresponds to a low value of d_{S}. In our scenario, these measurements carry a degree of uncertainty because they are extracted from a combination of data coming form probably inaccurate sensors located in different positions of the robot. Therefore, a fuzzy linguistic representation of their values is more appropriate. We associate the event δ_{S} with three fuzzy events such that $\Phi_{1,\delta_{\mathrm{S}}} = \{\mathtt{f}^{\mathrm{h}}_{1,\delta_{\mathrm{S}}}, \mathtt{f}^{\mathrm{m}}_{1,\delta_{\mathrm{S}}}, \mathtt{f}^{\mathrm{l}}_{1,\delta_{\mathrm{S}}}\}$. Here, $\mathtt{f}^{\mathrm{h}}_{1,\delta_{\mathrm{S}}}, \mathtt{f}^{\mathrm{m}}_{1,\delta_{\mathrm{S}}}, \mathtt{f}^{\mathrm{l}}_{1,\delta_{\mathrm{S}}}$ denote "high", "medium", "low" values of the respective variable.

Each fuzzy event is represented by a 3×3 fuzzy event matrix determining the state transitions triggered by this event. In order to determine the entries of these matrices, we follow an approach that is similar to (Huq et al., 2006a). Considering our choice of the crisp states in Fig. 2, we introduce state transitions that indicate the effect of each event in $\Phi_{\Sigma_{\mathrm{S}}} = \{\mathtt{f}^{\mathrm{h}}_{1,\delta_{\mathrm{S}}}, \mathtt{f}^{\mathrm{m}}_{1,\delta_{\mathrm{S}}}, \mathtt{f}^{\mathrm{l}}_{1,\delta_{\mathrm{S}}}\}$. Suppose the previous state of the robot was 2 ("medium" distance to the subgoal). Then, a new distance measurement, i.e., the occurrence of δ_{S}, causes the three different fuzzy events $\mathtt{f}^{\mathrm{h}}_{1,\delta_{\mathrm{S}}}, \mathtt{f}^{\mathrm{m}}_{1,\delta_{\mathrm{S}}}, \mathtt{f}^{\mathrm{l}}_{1,\delta_{\mathrm{S}}}$. The event $\mathtt{f}^{\mathrm{h}}_{1,\delta_{\mathrm{S}}}$ will make the system go to state 1 (high distance), $\mathtt{f}^{\mathrm{m}}_{1,\delta_{\mathrm{S}}}$ will lead to 2, and $\mathtt{f}^{\mathrm{l}}_{1,\delta_{\mathrm{S}}}$ will make the system go to 3 (low distance). The same rationale may be applied to all three states of the automaton, where state transitions triggered by each fuzzy event are denoted by arrows. In our setting, the entry in the i-th row and the j-th column of the fuzzy event matrix of each fuzzy event is nonzero if there is a transition from state j to state i with the respective fuzzy event. In addition, we assume that all nonzero entries for each particular fuzzy event are equal (values $g^{\mathrm{h}}_{d_{\mathrm{S}}}, g^{\mathrm{m}}_{d_{\mathrm{S}}}, g^{\mathrm{l}}_{d_{\mathrm{S}}}$). Having in mind the automaton of Fig. 2, $\mathtt{f}^{\mathrm{h}}_{\mu_{d_{\mathrm{S}}}}$, $\mathtt{f}^{\mathrm{m}}_{\mu_{d_{\mathrm{S}}}}$ and $\mathtt{f}^{\mathrm{l}}_{\mu_{d_{\mathrm{S}}}}$ are given by

$$\mathtt{f}^{\mathrm{h}}_{\mu_{d_{\mathrm{S}}}} = \begin{bmatrix} g^{\mathrm{h}}_{d_{\mathrm{S}}} & g^{\mathrm{h}}_{d_{\mathrm{S}}} & 0 \\ 0 & 0 & g^{\mathrm{h}}_{d_{\mathrm{S}}} \\ 0 & 0 & 0 \end{bmatrix}, \mathtt{f}^{\mathrm{m}}_{\mu_{d_{\mathrm{S}}}} = \begin{bmatrix} 0 & 0 & 0 \\ g^{\mathrm{m}}_{d_{\mathrm{S}}} & g^{\mathrm{m}}_{d_{\mathrm{S}}} & g^{\mathrm{m}}_{d_{\mathrm{S}}} \\ 0 & 0 & 0 \end{bmatrix}, \mathtt{f}^{\mathrm{l}}_{\mu_{d_{\mathrm{S}}}} = \begin{bmatrix} 0 & 0 & 0 \\ g^{\mathrm{l}}_{d_{\mathrm{S}}} & 0 & 0 \\ 0 & g^{\mathrm{l}}_{d_{\mathrm{S}}} & g^{\mathrm{l}}_{d_{\mathrm{S}}} \end{bmatrix}$$

Finally, the concrete values of $g^{\mathrm{h}}_{d_{\mathrm{S}}}$, $g^{\mathrm{m}}_{d_{\mathrm{S}}}$ and $g^{\mathrm{l}}_{d_{\mathrm{S}}}$ are determined using the membership function in Fig. 3 (a). That is, "low" distances are found when measuring less than 5.4 pixels, "medium" distances correspond to between 3.5 and 7.1 pixels, and high distances are found for more than 5.4 pixels. It has to be noted that it is the task of the system designer to define appropriate membership functions based on the particular system characteristics.

Together, the distance measurement is modeled by the FDES $S_{1,\delta_{\mathrm{S}}} = (Q_{1,\delta_{\mathrm{S}}}, \Phi_{1,\delta_{\mathrm{S}}}, q_{1,\delta_{\mathrm{S}}})$. In our experiments, an initial state of $q_{1,\delta_{\mathrm{S}}} = [0.1\ 0.8\ 0.1]$ proved to be suitable.

Similarly, the model $S_{1,\alpha_{\mathrm{S}}} = (Q_{1,\alpha_{\mathrm{S}}}, \Phi_{1,\alpha_{\mathrm{S}}}, q_{1,\alpha_{\mathrm{S}}})$ is constructed for the measurement of a_{S}. Here, the fuzzy events in $\Phi_{1,\alpha_{\mathrm{S}}} = \{\mathtt{f}^{\mathrm{h}}_{1,\alpha_{\mathrm{S}}}, \mathtt{f}^{\mathrm{m}}_{1,\alpha_{\mathrm{S}}}, \mathtt{f}^{\mathrm{l}}_{1,\alpha_{\mathrm{S}}}\}$ are evaluated based on the automaton in Fig. 2 (b) and the membership function in Fig. 3 (b). $q_{1,\alpha_{\mathrm{S}}} = [0.1\ 0.8\ 0.1]$ is the initial state.

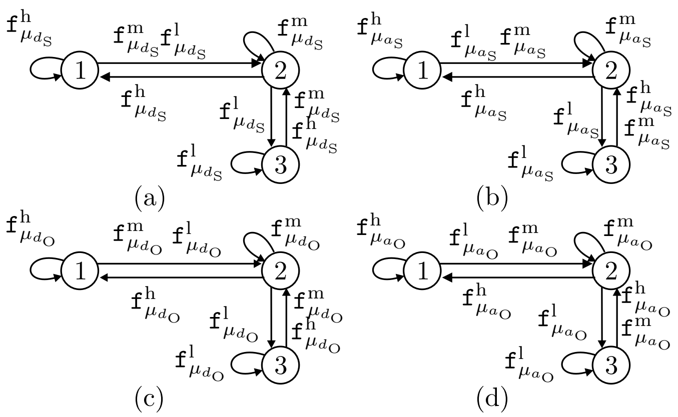

Fig. 2. Automata models for path following.

In each sampling instant, the overall path following objective is then described by the fuzzy state product of the fuzzy states for the distributed FDES $S_{1,\alpha_{\mathrm{S}}}$ and $S_{1,\delta_{\mathrm{S}}}$.

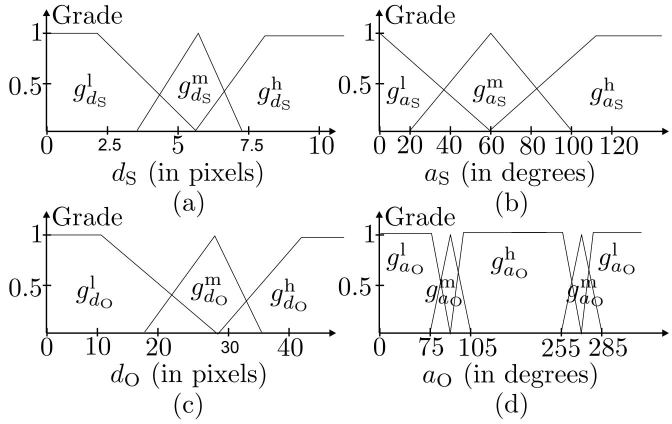

Fig. 3. Membership functions for goal 1 and goal 2.

Semantically, first, the successor states $\hat{q}_{1,\delta_{\mathrm{S}}} = q_{1,\delta_{\mathrm{S}}} \otimes \delta_{\mathrm{S}}$ and $\hat{q}_{1,\alpha_{\mathrm{S}}} = q_{1,\alpha_{\mathrm{S}}} \otimes \alpha_{\mathrm{S}}$ are evaluated, where the corresponding fuzzy event matrices are computed depending on the current values of d_{S} and a_{S}. Then, the fuzzy state product $\hat{q}_{1,\delta_{\mathrm{S}}} \oslash \hat{q}_{1,\alpha_{\mathrm{S}}}$ is taken, to determine the combined fuzzy state of the goal 1.

Obstacle Avoidance (Goal 2) Analogous to the previous section, we determine 2 FDES models $S_{2,\delta_{\mathrm{O}}}$ and $S_{2,\alpha_{\mathrm{O}}}$ with 3 crisp states each to characterize the distance d_{O} and the angle a_{O} as illustrated in Fig. 2 (c) and (d). We model the distance and angle measurement by the events δ_{O} and α_{O} and the associated fuzzy events in $\Phi_{2,\delta_{\mathrm{O}}} = \{\mathtt{f}^{\mathrm{h}}_{2,\delta_{\mathrm{O}}}, \mathtt{f}^{\mathrm{m}}_{2,\delta_{\mathrm{O}}}, \mathtt{f}^{\mathrm{l}}_{2,\delta_{\mathrm{O}}}\}$ and $\Phi_{2,\alpha_{\mathrm{O}}} = \{\mathtt{f}^{\mathrm{h}}_{2,\alpha_{\mathrm{O}}}, \mathtt{f}^{\mathrm{m}}_{2,\alpha_{\mathrm{O}}}, \mathtt{f}^{\mathrm{l}}_{2,\alpha_{\mathrm{O}}}\}$. The fuzzy event matrices can again be constructed combining the information in Fig. 2 (c) and (d) and Fig. 3 (c) and (d). Note that the angle measurement of α_{O} identifies angles smaller than 90° and larger than 270° as "low" (w.r.t. the goal of obstacle avoidance), angles around 90° and 270° as "medium" and the remaining angles as "high".

Together, the distributed FDES $S_{2,\delta_{\mathrm{O}}}$ and $S_{2,\alpha_{\mathrm{O}}}$ describe the sensor data evaluation for obstacle avoidance (goal 2). In each sample instant, the successor states $\hat{q}_{2,\delta_{\mathrm{O}}}$ and $\hat{q}_{2,\alpha_{\mathrm{O}}}$ of the initial fuzzy states $q_{2,\delta_{\mathrm{O}}}$ and $q_{2,\alpha_{\mathrm{O}}}$ are evaluated, where the fuzzy event matrices are computed depending on the current values of d_{O} and a_{O}. Then, the computation of the fuzzy state product $\hat{q}_{2,\delta_{\mathrm{O}}} \oslash \hat{q}_{2,\alpha_{\mathrm{O}}}$ yields the combined fuzzy state for goal 2. In our experiments, we used the initial states $q_{2,\delta_{\mathrm{O}}} = [0.1\ 0.8\ 0.1]$ and $q_{2,\alpha_{\mathrm{O}}} = [0.1\ 0.8\ 0.1]$.

4. STATE PREDICTION FOR FDES

We now introduce our strategy for the prediction of the future fuzzy state by modeling *control actions* as FDES.

4.1 FDES Framework for State Prediction

Our framework is based on the same set of goals $\mathcal{G}$ as in Section 3.2. We define a global *set of control actions* Γ, where the predicted behavior for each goal $g \in \mathcal{G}$ when applying a control action $\gamma \in \Gamma$ is modeled by an FDES $C_{g,\gamma}$. Furthermore, we introduce the set of control action combinations $\mathcal{A} \subseteq 2^{\Gamma}$ such that each $a \in \mathcal{A}$ denotes a potential set of control actions to be applied in each sampling instant. The initial fuzzy state $q_{g,\gamma}$ of $C_{g,\gamma}$ is chosen as $q_{g,\gamma} = q_g$ according to the sensor data evaluation in Section 3.2. We then characterize the predicted fuzzy state after the control action $a \in \Gamma$ for the goal $g \in \mathcal{G}$ by $\hat{q}_{g,a} = \max_{\gamma \in a} q_g \otimes \gamma$. Again, the task of the system designer is to find fuzzy event matrices that capture the future behavior for each control action.

4.2 Control Actions for the Mobile Robot

We now apply this strategy to our mobile robot example.

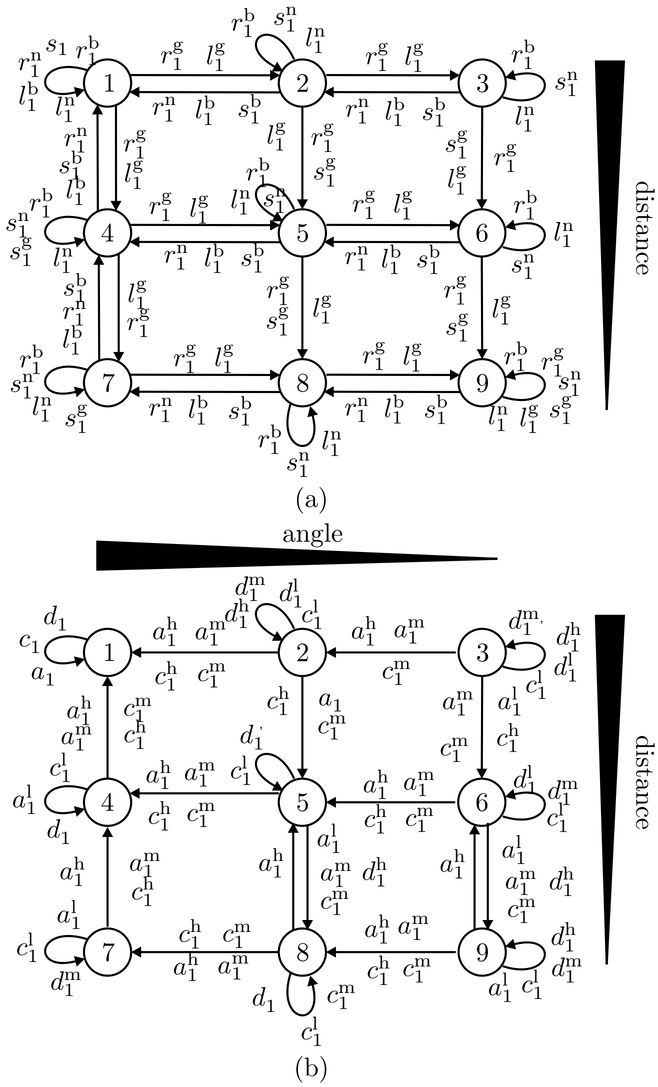

Fig. 4. State prediction for path following.

Path Following Objective (Goal 1) We introduce the events γ_{a}, γ_{c} and γ_{d} to model *acceleration* (+3% of the maximum speed), *constant speed*, and *deceleration* (−3%) of the mobile robot. In addition, the events γ_{r} (5° to the right), γ_{s} (straight) and γ_{l} (5° to the left) describe the angular movement of the robot such that $\Gamma = \{\gamma_{\mathrm{a}}, \gamma_{\mathrm{c}}, \gamma_{\mathrm{d}}, \gamma_{\mathrm{r}}, \gamma_{\mathrm{s}}, \gamma_{\mathrm{l}}\}$. For each of the events, we define an FDES that captures the expected behavior when applying the corresponding control action. For example, the event γ_{r} is modeled by $C_{1,\gamma_{\mathrm{r}}}$ with 9 crisp states that describe the expected angle and distance measure of the path following objective as depicted in Fig. 4 (a). Furthermore, we associate the three fuzzy events in $\Phi_{1,\gamma_{\mathrm{r}}} = \{r^{\mathrm{g}}_1, r^{\mathrm{n}}_1, r^{\mathrm{b}}_1\}$ with γ_{r}. Here, r^{g}_1, r^{n}_1 and r^{b}_1 model a "good", "neutral" and "bad" effect of turning to the right. The fuzzy event matrices for the fuzzy events in Φ_{1,γ_r} are constructed according to the respective state transitions in Fig. 4 (a).

Analogous FDES models C_{1,γ_l}, C_{1,γ_s}, C_{1,γ_a}, C_{1,γ_d} and C_{1,γ_c} are constructed for the remaining control actions. The corresponding fuzzy events are listed in Table 1. Finally, the non-zero entries of the fuzzy event matrices are determined by the membership functions for the angle changes in Fig. 5 (a) and for the speed actions in Fig. 5 (b). Here, the current speed of the robot is measured as a percentage of its maximum speed.

event	fuzzy events	explanation
γ_a	$\{a_1^h, a_1^m, a_1^l\}$	acceleration is "high", "medium", "low"
γ_c	$\{c_1^h, c_1^m, c_1^l\}$	const. speed is "high", "medium", "low"
γ_d	$\{d_1^h, d_1^m, d_1^l\}$	deceleration is "high", "medium", "low"
γ_r	$\{r_1^g, r_1^n, r_1^b\}$	turn right is "good", "neutral", "bad"
γ_s	$\{s_1^g, s_1^n, s_1^b\}$	go straight is "good", "neutral", "bad"
γ_l	$\{l_1^g, l_1^n, l_1^b\}$	turn left is "good", "neutral", "bad"

Table 1. Path following: fuzzy events.

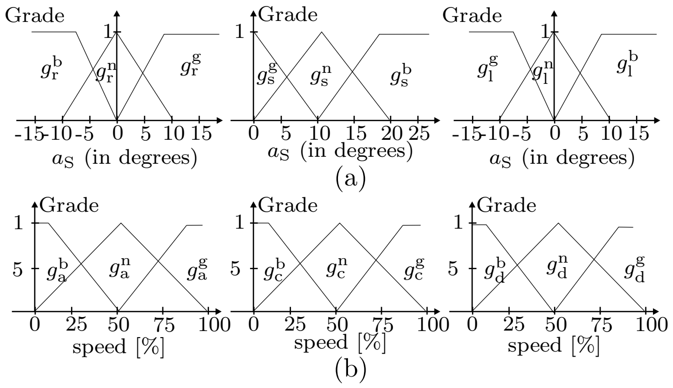

Fig. 5. Membership functions for speed and turning actions related to goal 1.

Obstacle Avoidance Objective (Goal 2) In order to model the control actions in Γ w.r.t. to goal 2, we use the FDES models C_{2,γ_r}, C_{2,γ_l}, C_{2,γ_s}, C_{2,γ_a}, C_{2,γ_d} and C_{2,γ_c} with 9 crisp states each. The construction of the fuzzy event matrices for the respective events in Γ is performed according the automata representation in Fig. 6 (a) for the turning actions and in Fig. 6 (b) for the speed actions.

Control Action Combinations Finally, we define the set of feasible control actions as $A = \{\{\gamma_a, \gamma_r\}, \{\gamma_a, \gamma_s\}, \{\gamma_a, \gamma_l\}, \{\gamma_c, \gamma_r\}, \{\gamma_c, \gamma_s\}, \{\gamma_c, \gamma_l\}, \{\gamma_d, \gamma_r\}, \{\gamma_d, \gamma_s\}, \{\gamma_d, \gamma_l\}\}$, i.e. we propose to combine a turning action with a speed action in each sampling instant. Hence, for each combination of control actions $a = \{\gamma, \gamma'\} \in \mathcal{A}$ and each subgoal $g \in \mathcal{G}$, we arrive at the predicted fuzzy state $q_{g,a} = \max\{q_g \otimes \gamma, q_g \otimes \gamma'\}$ as outlined in Section 4.1.

5. CONTROL APPROACH FOR FDES

This section elaborates our control method. It is based on the state prediction in the previous section that is computed in each sample instant of the system evolution. This prediction is then employed to determine an optimal control strategy to be applied in the subsequent sampling period.

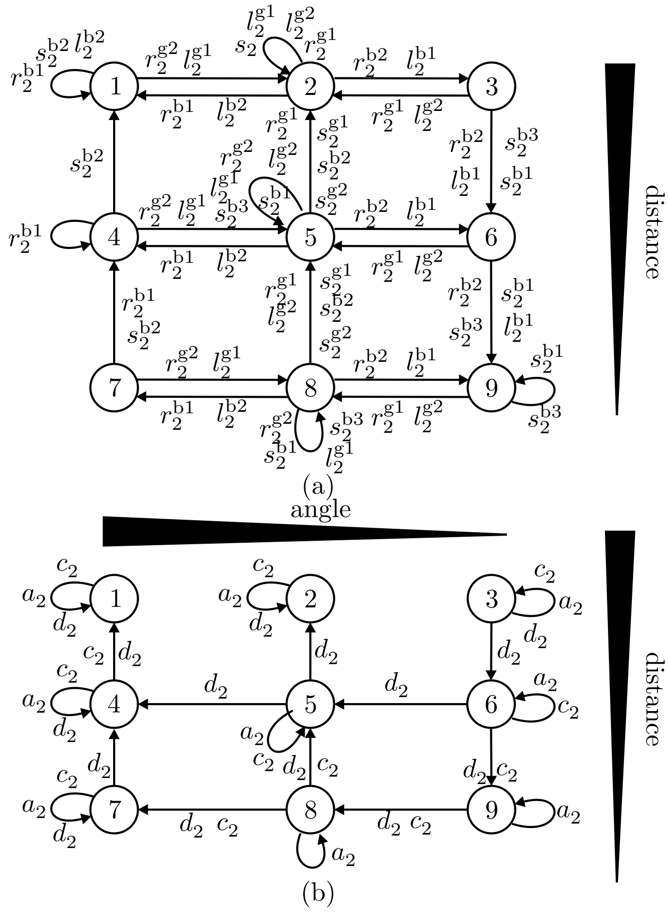

Fig. 6. State prediction for obstacle avoidance.

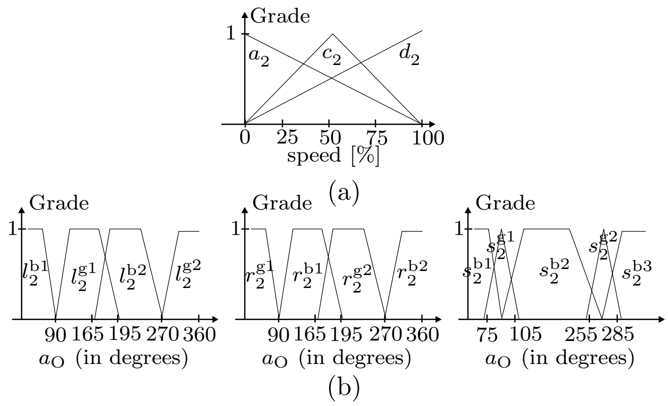

Fig. 7. Membership functions for speed and turning actions related to goal 2.

5.1 Control Problem

We formulate an optimal control problem. A similar control problem is investigated in (Lin and Ying, 2002). In our setting, we define a *weight vector* $w_g \in \mathbb{R}^n$ for each goal such that $\hat{q}_{g,a} w_g$ characterizes the benefit of the control action combination a for the goal g. Accordingly, our control objective for each $a \in \mathcal{A}$ is described by the function $J_a = \sum_{g \in \mathcal{G}} \hat{q}_{g,a} w_g$. Hence, we want to solve

$$\max_{a \in \mathcal{A}} J_a = \max_{a \in \mathcal{A}} \sum_{g \in \mathcal{G}} \hat{q}_{g,a} w_g. \quad (3)$$

5.2 Mobile Robot Example

We now apply the optimization described in the previous section to the mobile robot scenario in Fig. 8 (a). The robot has to move in a 'corridor' starting from an initial position $(0, 50)$, given in pixels, in order to reach its goal (small circle at $(325, 0)$). On its route, it has to go around the unknown obstacles located at $(60, 140), (70, 120), (150, 160), (150, 180), (60, 325)$. The path that has to be followed is marked by crosses. They are determined by using a well-established procedure ((Murphy, 2000)) to construct a Voronoi diagram and then apply A^* search to extract the safe path between the initial position of the robot and the target (obstacles are not taken into account). With a sampling period of 10 milliseconds, the maximum speed of the robot is 100 pixels/second, which corresponds to about 7.5 km/h.

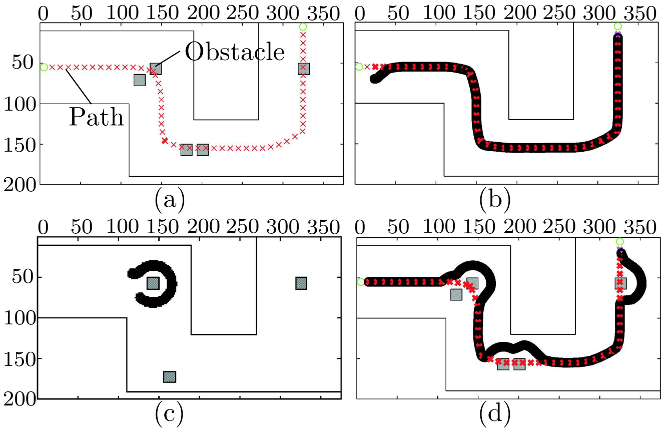

Fig. 8. (a) Scenario; (b) path following experiment; (c) obstacle avoidance; (d) combined experiment.

Path Following We first implement path following (goal 1) without any obstacles. We identify the crisp states that correspond to small d_S and small a_S (5, 6, 8, 9 in Fig. 4) as the most desirable states for goal 1. Hence, we choose $w_1 = [0.01\ 0.2\ 1.6\ 0.05\ 0.4\ 3.2\ 0.1\ 0.8\ 6.4]$. The application of the optimal control action according to Section 5.1 in each time instant results in the trajectory shown in Fig. 8 (b) that exactly follows the desired path.

Obstacle Avoidance In the second experiment, we implement goal 2, which is best achieved by a large distance d_O and a medium angle a_O (states 1, 2, 5 in Fig. 6). This is captured by $w_2 = [1.6\ 6.4\ 0.8\ 0.4\ 3.2\ 0.2\ 0.1\ 2.0\ 0.01]$. The result of our experiment is illustrated in Fig. 8 (c), where the robot always tries to move perpendicular to the direction pointing towards the obstacle.

Combination of Path Following and Obstacle Avoidance Finally, we study the joint implementation of goal 1 and goal 2 with the weight vectors from above. As can be seen in Fig. 8 (d), the robot follows the given path as long as no obstacles are encountered. Moreover, each obstacle is surrounded, whereas the robot returns to the desired path as soon as there is no more obstacle in its vicinity. It also has to be mentioned that the computational effort is small, since the evaluation of the optimal control strategy only involves the enumeration of the limited number of control action combinations in the set $\mathcal{A}$.

6. CONCLUSIONS

This paper presents a multi-objective framework for the control of systems that can be modeled as distributed fuzzy discrete event systems (FDES). First, in each sampling instant, a fuzzy representation of the current system state is obtained from potentially uncertain measurement data based on FDES models. This system state is then used to predict the future system state after pre-defined control actions, where again FDES are employed to model the predicted system behavior. Finally, the state predictions are used to find an optimal control strategy for the subsequent sampling instant. This feature is illustrated by simulation experiments with a mobile robot that has to both follow a given path and avoid unknown obstacles that can be encountered on the path.

REFERENCES

Barbera, H.M. and Skarmeta, A.G. (2002). A framework for defining and learning fuzzy behaviors for autonomous mobile robots. *Int. J. Intell. Syst.*, 17, 1–20.

Cao, Y. and Ying, M. (2005). Supervisory control of fuzzy discrete event systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 35(2), 366–371.

Cassandras, C.G. and Lafortune, S. (1999). Introduction to discrete event systems. *Kluwer*.

Huq, R., Mann, G.K.I., and Gosine, R.G. (2006a). Behavior modulation technique in mobile robotics using fuzzy discrete event system. *IEEE Transactions on Robotics*, 22(5), 903–916.

Huq, R., Mann, G.K.I., and Gosine, R.G. (2006b). Distributed fuzzy discrete event system for robotic sensory information processing. *Expert Systems*, 23(5), 273–289.

Kosecka, J. and Bajcy, R. (1994). Discrete event systems for autonomous mobile agents. *Robotics and Autonomous Systems*, 12, 187–198.

Lin, F. and Ying, H. (2002). Modeling and control of fuzzy discrete event systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 32(4), 408–415.

Mordeson, J.N. and Malik, D.S. (2002). *Fuzzy Automata and Languages: Theory and Applications*. Chapman & Hall, CRC, London, Boca Raton, FL.

Murphy, R. (2000). *Introduction to AI Robotics*. MA: MIT Press, Cambridge.

P. Vadakkepat, O.C. Miin, X.P. and Lee, T. (2004). Fuzzy behavior-based control of mobile robots. *IEEE Transactions on Fuzzy Systems*, 12(4), 559–565.

Pirjanian, P. and Mataric, M. (1999). A decision theoretic approach to fuzzy behavior coordination. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*.

Qiu, D. (2005). Supervisory control of fuzzy discrete event systems: A formal approach. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 35(1), 72–88.

Rosenblatt, J. and Handler, J. (1999). Architectures for mobile robot control. *Advances in Computers*, 48, 315–353.

Schmidt, K. and Boutalis, Y. (2008). Computation of supremal controllable sublanguages and infimal controllable superlanguages for fuzzy discrete event systems. *Control and Automation, Mediterranean Conference on*.

Wechler, W. (1978). *The Concept of Fuzziness in Automata and Language Theory*. Akademie-Verlag, Berlin.