SUPERVISORY CONTROL OF FUZZY DISCRETE EVENT SYSTEMS WITH APPLICATIONS TO MOBILE ROBOTICS

AWANTHA JAYASIRI DEWAGE DON

Supervisory Control of Fuzzy Discrete Event Systems with Applications to Mobile Robotics

By

Awantha Jayasiri Dewage Don

A thesis submitted to the School of Graduate Studies
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Faculty of Engineering and Applied Science
Memorial University of Newfoundland
St. John's, Newfoundland, Canada

April, 2012

Abstract

Fuzzy Discrete Event Systems (FDES) were proposed in the literature for modeling and control of a class of event driven and asynchronous dynamical systems that are affected by deterministic uncertainties and vagueness on their representations. In contrast to classical crisp Discrete Event Systems (DES), which have been explored to a sufficient extent in the past, an in-depth study of FDES is yet to be performed, and their feasible real-time application areas need to be further identified. This research work intends to address the supervisory control problem of FDES broadly, while formulating new knowledge in the area. Moreover, it examines the possible applications of these developments in the behavior-based mobile robotics domain.

An FDES-based supervisory control framework to facilitate the behavior-based control of a mobile robot is developed at first. The proposed approach is modular in nature and supports behavior integration without making state explosion. Then, this architecture is implemented in simulation as well as in real-time on a mobile robot moving in unstructured environments, and the feasibility of the approach is validated.

A general decentralized supervisory control theory of FDES is then established for better information association and ambiguity management in large-scale and distributed systems, while providing less complexity of control computation. Furthermore, using the proposed architecture, simulation and real-time experiments of a tightly-coupled multi-robot object manipulation task are performed. The results are compared with centralized FDES-based and decentralized DES-based approaches.

A decentralized modular supervisory control theory of FDES is then established for complex systems having a number of modules that are concurrently operating and also containing multiple interactions.

Finally, a hierarchical supervisory control theory of FDES is established to resolve

the control complexity of a large-scale compound system by modularizing the system vertically and assigning multi-level supervisor hierarchies. As a proof-of-concept example to the established theory, a mobile robot navigation problem is discussed. This research work will contribute to the literature by developing novel knowledge and related theories in the areas of decentralized, modular and hierarchical supervisory control of FDES. It also investigates the applicability of these contributions in the mobile robotics arena.

Acknowledgements

Throughout this endeavor many kind-hearted people have helped me to achieve my academic goals. At this milestone of my life, I will take the opportunity to appreciate them for all their boundless help.

I would like to thank my supervisors Prof. George Mann and Prof. Raymond Gosine for their friendly encouragement and worthy suggestions. Especially, I am indebted to them for allowing me enough freedom to explore new ideas, and at the same time, keep their faith in my research work. Also, I thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for providing financial aid.

Colleagues in the Intelligent Systems Lab (ISLAB) always have been supportive and inspiring. I would like to thank Dilan, Dilhan and Yuanlong for their warm company. Without their helping hands this effort would have been mind-numbing.

I am also deeply grateful to my parents and sisters who have been pillars of strength for me throughout my life. This accomplishment is a tribute to all the hardships that they have faced in raising me to be who I am today.

Finally, I thank my beloved wife Nisanka for having patience with many late night arrivals and taking care of our son Deegayu alone most of the time.

To my parents, Chandra and Jayasiri...

Contents

Abstract i

Acknowledgement iii

List of Tables ix

List of Figures x

List of Abbreviations and symbols xii

1 Introduction 1

1.1 Problem statement 3

1.1.1 Decentralization and modularization 4

1.1.2 Hierarchical structuring 4

1.1.3 Modeling of behavior-based systems 4

1.2 Objectives and the contributions 5

1.2.1 Develop a novel framework for behavior-based systems 5

1.2.2 Establish a general decentralized control theory of FDES . . . 6

1.2.3 Establish a decentralized modular control theory of FDES . . 7

1.2.4 Establish a hierarchical control theory of FDES 7

1.3 Organization of the thesis 8

2 Background **10**

2.1 On supervisory control of FDES 10

2.1.1 Centralized supervisory control of FDES 12

2.1.2 Decentralized supervisory control of FDES 14

2.2 On control of mobile robotics 16

2.2.1 Multi robot coordination 18

2.2.2 Distributed architectures for multi robot coordination 19

2.2.3 Fuzzy logic-based approaches 24

2.2.4 crisp DES-based approaches 25

2.2.5 FDES-based approach to control SRS 26

2.2.6 A taxonomy for MRS 26

2.2.7 Properties of a better coordination mechanism 27

2.2.8 Proposed approach 29

3 Single robot control using FDES **30**

3.1 Preliminaries 30

3.2 Supervisory control of FDES 37

3.2.1 Problem definition 37

3.2.2 Identification of Σ_c and Σ_{uc} 39

3.2.3 Supervisor definition 41

3.3 Application to single robot control 42

3.3.1 Perception to action mapping 42

3.3.2 Behavior coordination of a single robot 43

3.3.3 Behavior modeling 44

3.3.4 Supervisor Synthesis 47

3.4 Implementation of the proposed approach 50

3.4.1 Simulation results 51

3.4.2 Performance evaluation 55
3.4.3 Real-time implementation 57
3.5 Summary 59

4 Decentralized control of FDES 60
4.1 The decentralized control architectures of FDES 60
4.1.1 Conjunctive and Permissive (C&P) architecture 61
4.1.2 Disjunctive and Anti-permissive (D&A) architecture 63
4.1.3 General architecture 65
4.2 Application to decentralized control of MRS 68
4.2.1 Modelling of the MRS 69
4.2.2 Behavior modeling 70
4.2.3 Implementation 73
4.2.4 The computational complexity of the proposed approach . . . 75
4.2.5 Simulation results 76
4.2.6 Performance evaluation 81
4.2.7 Real-time implementation 83
4.3 Summary 84

5 Decentralized modular control of FDES 85
5.1 Modular and concurrent FDES 85
5.1.1 Separability of a fuzzy language 87
5.1.2 A Modular supervisory control theory of FDES 89
5.2 Summary 89

6 Hierarchical control of FDES 90
6.1 The hierarchical control architecture of FDES 90
6.1.1 Computation of *main-path* 92

6.1.2 The *output-control-consistency* of FDES 94
6.1.3 Calculation of a controllable sub language 95
6.1.4 The *strictly-output-control-consistency* of FDES 97
6.1.5 Dealing with the unobservability of low-level fuzzy events . . . 99
6.1.6 A Hierarchical supervisory control theory of FDES 100
6.2 A mobile robot navigation example 101
6.3 Summary 105

7 Conclusion 106

7.1 Discussion 106
7.1.1 Research limitations 108
7.2 Research contributions 109
7.2.1 Publications resulting from the research 110
7.3 Future directions 112

Appendix: Proofs of theorems 113

Bibliography 120

List of Tables

2.1 A classification based on behavior coordination mechanism 27

3.1 Fuzzy rule base for $\Sigma_c(\sigma_{FR})$ 45

3.2 Fuzzy rule base for $\Sigma_c(\sigma_{GT})$ 45

3.3 Fuzzy rule base for $\Sigma_{uc}(\sigma_{AO})$ 46

3.4 Fuzzy rule base for $\Sigma_{uc}(\sigma_{FW})$ 46

3.5 Fuzzy rule base for $\Sigma_{uc}(\sigma_{AD})$ 47

3.6 Fuzzy rule base for $T_s(\sigma_{FR})$ 47

3.7 Performance evaluation for the three test environments 57

4.1 Fuzzy rule base for $S_{t_i}^{P_i}(\sigma_{ao,i})$ 71

4.2 Fuzzy rule base for $S_{t_i}^{P_i}(\sigma_{fw,i})$ 71

4.3 Fuzzy rule base for $S_{t_i}^{P_i}(\sigma_{fr,i})$ 72

4.4 Fuzzy rule base for $S_{t_i}^{P_i}(\sigma_{gt,i})$ 72

4.5 Performance evaluation of the presented architectures 82

List of Figures

3.1 Fuzzy automaton modeling two behaviors 38

3.2 The information flow diagram showing perception to action mapping 43

3.3 Fuzzy automaton G modeling the behaviors of a mobile robot 44

3.4 Membership functions used to calculate $\Sigma_c(\sigma_{FR})$ 45

3.5 Membership functions used to calculate $T_s(\sigma_{FR})$ 47

3.6 Plots related to Test I 52

3.7 Plots related to Test II 53

3.8 Plots related to Test III 54

3.9 Navigation in complex environments using proposed approach 55

3.10 Performance in the real-world with unmodeled obstacles 58

3.11 Performance in the real-world with a dead end 58

4.1 C&P decentralized supervisory control architecture of FDES 61

4.2 D&A decentralized supervisory control architecture of FDES 63

4.3 General decentralized supervisory control architecture of FDES . . . 65

4.4 Two robots moving a beam 68

4.5 Fuzzy automaton G_i 70

4.6 Membership functions used to calculate $S_{t_i}^{P_i}(\sigma_{fr,i})$ for i^{th} robot 71

4.7 Navigation results for Test I 77

4.8 Plots related to centralized FDES-based control for Test I 77

4.9 Plots related to decentralized crisp DES-based control for Test I . . . 77
4.10 Plots related to decentralized FDES-based control for Test I 78
4.11 Navigation results for Test II 78
4.12 Plots related to centralized FDES-based control for Test II 79
4.13 Plots related to decentralized crisp DES-based control for Test II . . . 79
4.14 Plots related to decentralized FDES-based control for Test II 79
4.15 Navigation results for Test III 80
4.16 Plots related to centralized FDES-based control for Test III 80
4.17 Plots related to decentralized crisp DES-based control for Test III . . 81
4.18 Plots related to decentralized FDES-based control for Test III 81
4.19 Real-time navigation: In an environment with 1 obstacle 83
4.20 Real-time navigation: In an environment with 2 obstacles 83

5.1 Fuzzy automata G_1 and G_2 86
5.2 Fuzzy automaton $G_1 \parallel G_2$ 86

6.1 A two-level hierarchical supervisory control system 91
6.2 Fuzzy automata models of low-level and high-level FDES 92
6.3 Modified fuzzy automaton G'_{hi} 98
6.4 The environment for mobile robot navigation 101
6.5 Fuzzy automaton modeling AO 102
6.6 Fuzzy automaton G_{lo} modeling robot tasks 102
6.7 Fuzzy automaton G_{hi} 104

List of Abbreviations and symbols

List of Abbreviations

C&P	Conjunctive and Permissive
D&A	Disjunctive and Anti-permissive
ABA	Architecture for Behavior-based Agents
AuRA	Autonomous Robot Architecture
BLE	Broadcast of Local Eligibility
CAMPOUT	Control Architecture for Multi robot Planetary OUTposts
CM	Competence Modules
DES	Discrete Event Systems
EFDES	Extended Fuzzy Discrete Event Systems
FD	Feature Detectors
FDEDS	Fuzzy Discrete Event Dynamical Systems
FDES	Fuzzy Discrete Event Systems
FIS	Fuzzy Inference System
FL	Fuzzy Logics
FSA	Finite State Automaton
MRS	Multi Robot Systems
PAB	Port Arbitrated Behavior paradigm
PDES	Probabilistic Discrete Event Systems
SRS	Single Robot Systems

UGV	Unmanned Ground Vehicle

List of Symbols - Chapter 3

$P^{-1}(P(s))$	A subset, which defines the possibility of the natural projection of a string, to be seen as the same as that of s
$\bar{L}$	The prefix-closure of L
$\cap$	Set intersection
$\circ$	Max-Min or Max-Product operation
$\cup$	Set union
δ	Transition mapping
$\otimes$	The tensor product
Σ	Set of fuzzy events
σ	A fuzzy event
Σ^*	Kleene-closure of Σ
Σ_c	The set of fuzzy controllable events
$\Sigma_c(\sigma)$	The degree to which, σ being controllable
Σ_o	The set of fuzzy observable events
Σ_{uc}	The set of fuzzy uncontrollable events
Σ_{uo}	The set of fuzzy unobservable events
$\tilde{\cap}$	Fuzzy intersection
$\tilde{\cup}$	Fuzzy union
ε	Null-event (and empty string)
B_E	Bending energy
C_t	Fuzzy state-based controllability
D^{obs}_{mean}	Mean distance to obstacles
D^{obs}_{min}	Minimum distance to obstacles
G	Fuzzy automaton modeling a system

$G_1 \parallel G_2$	The parallel composition of G_1 and G_2
I	The unit matrix
L_G	Language generated by a system G
$L_G(s)$	Physical possibility of occurring s
$L_{G,m}$	The fuzzy language of marked fuzzy strings
$L_{S/G}$	The fuzzy language generated by S/G
L_{tot}	Total length of the trajectory
M_{t+1}	The matrix of all fuzzy events at time $t+1$
N_{tot}	Number of collisions
$P(\sigma)$	The natural projection of σ
Q	Set of fuzzy states
q_0	Initial fuzzy states
S	The fuzzy supervisor
s	A fuzzy string of events
S/G	The fuzzy supervisor S controlling the FDES G
S^P	The fuzzy partially observation supervisor
$S_s(\sigma)$	The possibility of σ enabled by S after observing the string s
$T_s(\sigma)$	The degree of conformity of σ with all fuzzy uncontrollable event, given s has occurred
$\underline{\vec{A}_{t+1}}$	The coordinated action of all behaviors at time $t+1$
$\vec{\mathbf{a}}_{i,t+1}$	The unit vector representing i^{th} behavior at time $t+1$

List of Symbols - Chapter 4

d_b	The half of the beam length
d_c	The distance to the current way-point from beam center
d_f	The distance to the next way-point from beam center
$D_{Mean,i}$	The mean distance to obstacles from i^{th} Robot

$D_{Min,i}$	The minimum distance to obstacles from i^{th} Robot
$G_{i,t}$	The FDES model of the MRS according to the i^{th} robot perception at time t
G_{sys}	The complete FDES model of the MRS
α_i	The angle between i^{th} robot heading and the current way-point
β_i	The angle between θ_i and the line connecting beam center and the current way-point
δ_i	The angle between θ_i and the line connecting beam center and the next way-point
γ_i	The angle of the beam with respect to the horizontal axis measured from i^{th} robot
Σ_i	fuzzy event set of S^{P_i}
$\Sigma_{cp,c}$	fuzzy controllable event set of S^{cp}
$\Sigma_{cp,o}$	fuzzy observable event set of S^{cp}
Σ_{cp}	fuzzy event set of S^{cp}
$\Sigma_{da,c}$	fuzzy controllable event set of S^{da}
$\Sigma_{da,o}$	fuzzy observable event set of S^{da}
Σ_{da}	fuzzy event set of S^{da}
Σ_g	fuzzy event set of S^g
$\Sigma_{i,c}$	fuzzy controllable event set of S^{P_i}
$\Sigma_{i,d,c}$	subset of fuzzy controllable events of $\Sigma_{i,d}$
$\Sigma_{i,d}$	fuzzy event set of S^{P_i} where the default setting is disablement
$\Sigma_{i,e,c}$	subset of fuzzy controllable events of $\Sigma_{i,e}$
$\Sigma_{i,e}$	fuzzy event set of S^{P_i} where the default setting is enablement
$\Sigma_{i,o}$	fuzzy observable event set of S^{P_i}
θ_b	The heading of the beam center with respect to the horizontal axis

θ_i	The i^{th} robot heading with respect to the horizontal axis
d_i	The distance between i^{th} robot center point and the current way-point
$P_i(s)$	Projection of s observed by S^{P_i}
S^{cp}	The $C\&P$ decentralized supervisor
S^{da}	The $D\&A$ decentralized supervisor
S^{g}	The general decentralized supervisor
S^{P_i}	The i^{th} fuzzy partially observation supervisor

List of Symbols - Chapter 5

$P^{-1}(L)$	The inverse projection of the fuzzy language L
$P_i(\bar{k})$	The distributed local knowledge of $\bar{k}$ at i^{th} module
$P_i(\bar{k})(s)_{max}$	The highest degree of possibility of s which is belongs to $P_i(\bar{k})$
L_{G_1}	The fuzzy language generated by the local plant G_1

List of Symbols - Chapter 6

Σ	The set of fuzzy events of low-level FDES model
τ	A high-level fuzzy event
$F_{s_i}(\sigma_j)$	The feasibility of σ_j being available in s_i
G_{hi}	The high-level FDES
G_{lo}	The low-level FDES
m_{lo}	The language containing low-level fuzzy strings given by $\theta^{-1}(\tau)_m$
S_{hi}	The high-level supervisor
S_{lo}	The low-level supervisor
T	The set of fuzzy events of high-level FDES model
t	A of high-level fuzzy string
T_{uc}	The set of fuzzy uncontrollable events of high-level FDES model

L_{lo}	The fuzzy language generated from low-level FDES
$\bar{k}_{lo}^{\uparrow}$	The supremal fuzzy controllable prefix-closed sub language of $\bar{k}_{lo}$
Con_{hi}	The channel for controlling G_{hi} by S_{hi}
θ	The map from low-level to high-level
$\theta^{-1}(\tau)_m$	The language of marked low-level fuzzy strings, which generate τ in high-level with the mapping θ
Com_{hi-lo}	The channel for passing commands from S_{hi} to S_{lo}
Con_{lo}	The channel for controlling G_{lo} by S_{lo}
Inf_{hi}	The channel for informing results to S_{hi} from G_{hi}
Inf_{lo-hi}	The information channel from G_{lo} to G_{hi}
Inf_{lo}	The channel for informing results to S_{lo} from G_{lo}
L_{hi}	The fuzzy language generated from high-level FDES
L_{lo}^{MP}	The main-path of G_{lo}

Chapter 1

Introduction

A class of systems, in which their dynamics can be characterized by asynchronous occurrences of events with discrete state representations, is called Discrete Event Systems (DES) [1]. Examples of such systems include computer and communication networks, automated manufacturing systems, air traffic control systems, and so forth. Present complex system requirements with the rapid advances of technology demand high-level control specifications of the system, designed to achieve the desired outcome, to be stipulated. These specifications typically include abstractions of explicit descriptions of what must be performed at the sensor-actuator level of the system, which are often modeled as event-driven sequences that are triggered from stimuli resulting in a change of the system status. The conventional engineering approach that is accompanied by differential equations for modeling and control of time-driven dynamical systems seems to be incompatible for high-level discrete event driven processes. Hence, new modeling frameworks, analysis and design techniques and control methodologies are required for DES-based systems.

The behavior of the DES is defined in terms of an event sequence. This is called the *language* of the system and represented using an appropriate modeling formalism such as *automata* [1]. The desired high-level (discrete) behavior specification is achieved via exerting control by means of a supervisor and hence it is termed the *supervisory control of DES* [2,3]. Complying with the commands issued by the supervisor, which are in the form of events, the continuous-variable controllers are employed to control the servo mechanisms of the system at the low-level. The supervisor is updated by the information from sensors and actuators, which is also transmitted in the form of events.

Most real-world systems suffer from deterministic uncertainties when defining events and state transitions. This is mainly due to the vagueness of their models and the imprecision of their sensors. Under these circumstances the crisp representations of DES will not be sufficient and are simply incapable of representing the exact nature of the system at a given time. To rectify these limitations a new field of study, termed Fuzzy Discrete Event Systems (FDES) theory, has emerged [4–6]. FDES is capable of representing events and states using possibility distributions, which are more appropriate for such scenarios. Hence, it provides the flexibility to integrate associated uncertainties into events and state transitions. Consequently, the specifications are defined in terms of *fuzzy languages*, which facilitate the representation of possibility distributions of events, and *fuzzy automata* are employed to model FDES, in contrast to the (crisp) automata used in DES.

Although a complete theoretical foundation covering all aspects analogous to those of crisp DES is yet to be established, FDES theory has been employed effectively in recent literature for modeling and control of complex systems that are affected by event and state uncertainties. Examples of such systems include determining treatment regimens in drug delivery [7,8], control of air-conditioning systems [9], and behavior-based robotics [10,11]. With its strong mathematical background and proven success, utilization FDES-based techniques in arenas where uncertainty handling of event driven systems plays a key role, brings promising results.

As previously mentioned, one of the promising application domains for FDES-based system modeling and control is behavior-based robotics. Autonomous robotic systems performing the tasks in dynamic environments react to unexpected events by controlling their diverse sensors and actuators in real-time. However, they often experience difficulties posed by issues such as sensor uncertainty and ambiguity of obstacle locations and their shapes, which lead them to make inaccurate decisions due to the errors in state representations. Since robot applications are becoming more commonplace, the demand for more robust fault-tolerant and efficient deployments is inevitable and can only be fulfilled by employing sophisticated control techniques.

The FDES-based modeling and control of robotic systems generates event-driven responses to environmental stimuli and facilitates uncertainty management by using Fuzzy Logic (FL) techniques. Furthermore, its modular state representation by the use of fuzzy automata eases behavior integration and helps to perform better behavior coordination in behavior-based systems.

1.1 Problem statement

The present literature of FDES theory and its supervisory control aspects pose several key issues. In order to make a robust formalism, a significant amount of theoretical establishments still need to be set up due to the incompleteness of FDES theory relative to that of crisp DES. This will be particularly favorable for guaranteeing safety, uncertainty management and the information association of FDES. Furthermore, in the applications perspective it is worthwhile to discuss more examples with better modeling and in-depth analysis to show the effectiveness of the theory. These limitations lead to the formulation of several problems.

1.1.1 Decentralization and modularization

Decentralized and modular supervisory control, in which the supervision is distributed among a set of local supervisors, is of much interest since it minimizes the horizontal complexities of control computation and increases the availability of controlled performance. Although this has been extensively addressed under crisp DES, the existing studies of decentralized supervisory control of FDES discuss only one event fusion rule. Consequently, some of the useful information is neglected and hence such an approach is simply inadequate to develop a robust formalism. For better ambiguity management in the decentralized decision making of FDES, a new architecture with enhanced information association is necessary. However, developing such a framework having a low computational power will be challenging.

1.1.2 Hierarchical structuring

Exerting supervisory control for large-scale compound plants is often computationally intensive. For such systems, hierarchical structuring of supervision helps to resolve the control complexities in a vertical direction. Several studies covering this topic can be found under the crisp DES setting. However, to the best of our knowledge, there is no research in the literature relating to hierarchical supervision of FDES. For efficient control of a complex plant, which has been intricately modeled as an FDES, hierarchical supervisory control of FDES needs to be established.

1.1.3 Modeling of behavior-based systems

With its involved sensor uncertainties and vagueness in state representation, the behavior-based robotics arena serves as a good candidate for applying FDES theory to practice and investigate its effectiveness. Nevertheless, the FDES framework adopted by Huq et al. in [10] for modeling and control of behavior-based robotics is inflexible for behavior integration due to its complexity and also computationally expensive. Hence, for better modeling of such systems it is necessary to have a new structure, which is modular and readily scalable with relatively low computational demand.

1.2 Objectives and the contributions

The purpose of this research work is to explore the field of FDES, and it attempts to address the issues raised in section 1.1. As a result, it moves toward establishing a more complete theory of supervisory control of FDES and investigates the practical applications of the theory in the mobile robotics arena. More precisely, the following objectives and related contributions have been identified.

1.2.1 Develop a novel framework for behavior-based systems

An FDES-based supervisory control framework, which exploits the well-defined Ramadge-Wonham mathematical structure [2, 3], is introduced for modeling and control of behavior-based systems. The proposed approach eases the behavior representation and facilitates the behavior coordination of those systems. Furthermore, it exhibits a modular scheme, in which the behaviors can be added efficiently without significantly increasing the computational complexity. For validating the proposed approach, both simulation and real-time implementations are performed for a mobile robot navigating in an unmodeled environment. Moreover, a performance evaluation is conducted to compare the approach with its counterparts.

The contributions are:

1. Development of a novel supervisory control framework for behavior-based systems, which exploits the *command fusion* type of coordination.

2. Application of the framework in simulation as well as in real-time for mobile robot navigational tasks and comparison of the performances.

1.2.2 Establish a general decentralized control theory of FDES

In decentralized control the local supervisors cooperatively achieve the desired system-level behavior. This approach also helps to decrease the horizontal control complexities present in large-scale systems. Existing studies on decentralized supervisory control of FDES such as [12–14] discuss an architecture, which combines the locally recommended degrees of fuzzy events using the *intersection* operator. Hence, this process is termed *fusion by intersection*. To retain the information that is left out from the above mentioned fusion method, another architecture, which fuses the locally enabled degrees of fuzzy events using the *union* operator, is presented in this research. Such an operation is called *fusion by union*. Moreover, since both of these architectures individually possess limitations, a more general architecture for decentralized supervisory control of FDES that combines both above fusion rules is established. This is performed by extending the general decentralized architecture of crisp DES in [15] to fuzzy domain. This new architecture will pave the way for better information association and uncertainty management of decentralized decision making. For investigating the performances of the new general architecture, both simulation and real-time experiments are carried out.

The contributions are:

1. Development of a decentralized supervisory control architecture, which utilizes the *fuzzy union* operator for fusion of fuzzy events (by extending [15]).

2. Formulation of a general decentralized supervisory control architecture, which uses both *fuzzy intersection* and *fuzzy union* operators for event fusion (by extending [15]).

3. Implementation of the proposed architecture in both simulation and real-time in a robot team engaged in a tightly-coupled object manipulation task.

4. Comparison of the performances with the FDES-based centralized and the DES-based decentralized control approaches.

1.2.3 Establish a decentralized modular control theory of FDES

The decentralized modular control explores the supervisory control problem related to complex systems that are composed of concurrently operating and multiple interacting modules. In this case, a set of non-communicating local supervisors individually achieve the system-level specification using their own sensing and acting capabilities. In this research, we study the decentralized modular supervision of FDES assuming the event sets of each local supervisor are mutually disjoint. This is useful for detecting the existence of the local supervisors for a complex system composed of a number of FDES modules, each simultaneously performing their own tasks to attain the global specification.

The contribution is:

Formulation of a decentralized modular supervisory control theory for concurrent FDES.

1.2.4 Establish a hierarchical control theory of FDES

The complexity of modeling large-scale systems can be reduced by modularizing them into detailed low-level and abstract high-level representations. The desired system behavior is then specified on the high-level model with the help of a virtual high-level supervisor and it is realized via a low-level supervisor exerting control on the low-level model. Such a structuring of control commands is termed *hierarchical* supervision and it plays a key role in model reduction and decreasing the vertical complexities of control computation. In this research, we study the hierarchical supervisory control problem of FDES for large-scale systems. Moreover, we discuss a behavior-based robotic navigational task as a *proof of concept* example to the developed theory.

The contributions are:

1. Formulation of a hierarchical supervisory control theory for large-scale FDES.

2. Explore the applicability of the theory in robotic navigational tasks.

1.3 Organization of the thesis

Chapter 1 highlights the focused area of research and outlines the objectives and contributions of the thesis.

Chapter 2 studies the background in the field of supervisory control of FDES. Furthermore, it presents some mobile robot control architectures emphasizing their relative advantages and disadvantages.

Chapter 3 develops an FDES-based supervisory control framework for behavior-based robot navigation and demonstrates its simulation and real-time performances.

Chapter 4 establishes the general decentralized supervisory control theory of FDES. Moreover, employing the proposed architecture, it discusses a tightly-coupled multi robot task execution as an application.

Chapter 5 establishes the modular supervisory control theory of FDES with several related examples.

Chapter 6 establishes the hierarchical supervisory control theory of FDES. In addition, based on the theoretical formulation it presents a proof-of-concept application in the area of behavior-based control of a mobile robot.

Chapter 7 provides the conclusion and future work.

Chapter 2

Background

Since the pioneering work conducted by F.Lin *et al.* [4] on modeling and control of FDES, several notable studies have been performed to develop formal supervisory control theories of FDES. In this chapter, such approaches are briefly discussed while emphasizing their key properties. Moreover, several architectures and DES/ FDES-based techniques for control of mobile robots are examined. Then, a taxonomy for MRS, which is based on the application domains and the behavior coordination schemes, is presented.

2.1 On supervisory control of FDES

Supervisory control was first proposed for asynchronous, event driven and possibly nondeterministic processes having discrete representations to achieve the desired closed-loop system behavior [2,3]. There are several frameworks for modeling discrete event driven processes or systems, such as finite automata [1, 16], Petri nets [1, 17], and their variations [1]. The traditional supervisory control theory of crisp DES focuses on achieving a given specification by restricting the behavior of a system to a subset of its original "uncontrolled behavior". Such a system specification must be able to fulfill two essential requirements:

1. The *safety properties* that prevent the system from reaching certain undesirable states, such as collision of a robot.

2. The *non-blocking properties* that avoid the deadlock states, which terminate the execution of events, as well as the live-lock cycles, which lead to failures in task completion.

We are interested in the *legal* or *admissible* behavior of the controlled system, in which the supervisor never disables a feasible *uncontrollable event* and enables/disables the *controllable events* in order to achieve the desired specification [1]. Furthermore, the presence of *unobservable events* results in generating the same control action (by the *partial-observation supervisor*) for strings, which cannot be differentiated from each other. The theoretical developments are mainly based on the well-established *Ramage-Wonham* supervisory control framework [2, 3].

Supervisory control of FDES is fundamentally studied as an extension of that for crisp DES, when the events and states become "fuzzy" instead of "crisp" assignments. Consequently, these states and events can be represented by fuzzy automata [18,19] or fuzzy Petri nets [20]. When the possibility grades of the fuzzy events and states reach their boundary values (0 or 1), they simply represent the events and states in a crisp DES. Hence, crisp DES can be considered as a subclass of FDES when boundary conditions are applied. Therefore, the performed theoretical developments in the area of supervisory control of FDES must be consistent with the well-established traditional crisp DES formulations. While a comprehensive supervisory control theory of FDES is yet to be set up, two main approaches present in the literature are inspiring and hence deserve attention. They are namely, centralized and decentralized supervisory control of FDES.

2.1.1 Centralized supervisory control of FDES

The centralized supervisory control problem of FDES was first studied by Y. Cao et al. [5]. In this study, the formulation is based on *max-min* automata models [21] and the behavior of the FDES is described by fuzzy languages. The theoretical extensions from crisp DES to FDES are performed in a natural way. However, with this approach the fuzzy events are fully observable and the event controllability is considered crisp (either 0 or 1), which poses some restrictions. The supervisor aims to disable the controllable events to certain degrees in order to achieve the specification.

In [6], D. Qiu presented a formal approach for supervisory control of FDES. In this formulation, each fuzzy event has a degree of controllability but still is fully observable to the supervisor. The controllability and non-blocking controllability theories related to FDES are studied while considering both max-min and max-product automata for modeling FDES. Furthermore, derivation of the *supremal controllable sublanguage* and the *infimal prefix-closed controllable superlanguage* from a given non-controllable fuzzy language are discussed.

Both the above studies are based on event feedback control. However, providing a language specification as a form of possibility distribution of event strings seems to be difficult since the possibility degrees of all prefixes of each string have to be pre-specified. To overcome this, state-based control of FDES was proposed in [22], which uses system state descriptions as specifications. The authors also examined the stabilization of FDES, which refers to the possibility of driving an FDES from its initial state to a prescribed subset of fuzzy states. The key issue with this setting compared to the crisp DES case is that the reachable fuzzy state set of the closed-loop system is generally not a subset of the open-loop system. This is due to the fact that the possibility of occurrence of each fuzzy event is generally not limited to its boundary values.

Notably, [23] presented a more general setting in which each fuzzy event is associated with a degree of controllability and a degree of observability. Furthermore, the authors discuss a controllability and observability theorem of FDES with the partial observation of fuzzy events. Moreover, a computing tree based method is proposed to examine whether the controllability and observability conditions are satisfied, which can further test the existence of supervisors for a given control problem.

Alternatively, an on-line optimal control strategy for decision making in FDES is discussed in [24]. A forward-looking tree is constructed for control synthesis and a *performance index* is computed for each node of the tree. The control objective is specified to maximize the performance index at a finite number of steps. Furthermore, two indices termed as *effectiveness* and *cost* are introduced for each node to solve the *extended optimization* problem (i.e., maximizing the effectiveness for a given cost). This approach is successfully employed in HIV/AIDS treatment planning. Better results can be produced by looking ahead further, but with a higher computational cost. Also, since more fuzzy events are possible at each node, the constructed tree for FDES has more branches than that of crisp DES, which also increases the complexity.

In [7], a general purpose decision making and optimization technology is developed using FDES theory to produce treatment decisions for any given patient. The approach is based on a genetic algorithm-based optimizer and this methodology is successfully applied in the selection of optimal HIV/AIDS treatment regimens. Extension of the method with self-learning capability is presented in [8] including weight adjustments and effectiveness computation for all treatment objectives.

Fault Detection and Diagnosis (FDD) plays a crucial role in increasing the productivity of a system. In [25], FDES literature is further developed by introducing FDD as the generalization of that of crisp DES. Especially, a new class of systems named fuzzy discrete event dynamical systems (FDEDS) is proposed as a non-linear dynamical version of FDES to analyze event-driven systems as non-linear dynamic systems and non-linear system diagnosability is introduced for FDEDS. A fuzzy approach for diagnosability of FDES is presented in [26], which introduces a function to characterize the degree of diagnosability. The necessary and sufficient conditions for diagnosability of FDES are proposed and a method for checking the diagnosability condition is given. The proposed approach is capable of failure diagnosis of both crisp DES and FDES. However, only the centralized diagnosability of FDES is explored in these approaches.

However, for handling ranges of knowledge uncertainties and subjectivity, especially in the field of biomedical applications, the existing theory of FDES is insufficient since it does not support elements in the states and event transition matrices to be fuzzy numbers. Hence, extended FDES (EFDES) is theorized primarily in [27, 28] by incorporating type-2 fuzzy sets [29]. The domain experts can now intuitively and quantitatively express their knowledge by using both type-1 and type-2 fuzzy sets.

2.1.2 Decentralized supervisory control of FDES

Control of a large-scale system using a monolithic, centralized supervisor often poses several issues: such as limited sensing and actuation capabilities, increased complexity of control computation and the problem of the single point of failure. Therefore, to minimize the above issues and increase the availability, the decentralized control, in which the supervision is distributed among a set of local controllers, is established in the literature. These local controllers possess individual sensing and actuation capabilities and cooperatively achieve the desired system-level behavior. Typical examples of decentralized systems include integrated sensor networks, communication and computer networks, complex automated highway systems, etc.

The decentralized supervisory control problem of crisp DES has been extensively studied under the Ramadge-Wonham framework [1, 30–34]. In most cases control actions of the local supervisors are fused using the intersection of locally enabled events, which is now referred to as Conjunctive and Permissive (C&P) architecture [15]. Fusion of control actions using the union of locally enabled events, which is referred to as Disjunctive and Anti-permissive (D&A) architecture, is discussed in [15]. Furthermore, in [15] the authors present a general architecture, which combines both C&P and D&A architectures. Since then, several extensions to the general architecture have been proposed: such as partition of the controllable event set to account for priorities and exclusivities [35], inference-based ambiguity management in decentralized decision making [36], and a multi-decision framework where several existing decentralized control architectures running in parallel and their decisions are combined conjunctively or disjunctively to minimize the information lost [37].

Decentralized supervisory control of FDES was first theorized by Y. Cao et al. [12] and later by F. Liu et al. [13]. In [12], co-observability of fuzzy languages is discussed by generalizing that of crisp languages as discussed in [31]. Furthermore, by incorporating these extensions, necessary and sufficient conditions for the existence of local (fuzzy) supervisors that achieve the fuzzy language specification are derived and a decentralized control of FDES is primarily established. The decentralized control problem of FDES is investigated in [13] based on a more general framework, in which each fuzzy event has a degree of controllability and a degree of observability. Moreover, for verifying whether the controllability and co-observability conditions hold, the authors presented a detailed computing method, which can also test the existence of decentralized supervisors.

The reliability of decentralized supervisory control problem of FDES is studied in [14] with possible failures of some of the local supervisors. A new definition of co-observability is given and the synthesis problem of reliable decentralized supervisory control of FDES is investigated.

In all above studies [12–14] the crisp DES C&P architecture has been extended to FDES. Consequently, the final recommendation is achieved by calculating only the fuzzy-intersection (taking the minimum) of the locally enabled degrees of fuzzy events. A new architecture, that extends the crisp DES D&A architecture to FDES, will combine a different set of information by calculating the fuzzy-union (taking the maximum) of the locally enabled degrees of fuzzy events. Furthermore, a general decentralized supervisory control architecture of FDES, which combines both above architectures, will provide better ambiguity management and information association capabilities in decentralized decision making of FDES.

It is important to note that above literature is based on the decentralized supervisory control of non-communicating controllers. In this research work, the decentralized supervisory control problem of FDES will be addressed without any communication among the local supervisors.

2.2 On control of mobile robotics

The common architectures available for controlling the single robot systems (SRS) can be categorized into four paradigms; namely, hierarchical, reactive, hybrid and behavior-based [38–40]. The hierarchical paradigm represents a top-down approach where all the deliberative plans are integrated to control commands [41]. This approach suffers from slow responsiveness as in every cycle the robot needs to update the world model and do replanning. In the reactive paradigm [42] sense-to-action is tightly-coupled, where sensory information is mapped to motor actions in task execution. This architecture represents a bottom-up approach and is more suitable for operating in dynamic and unstructured environments. Moreover, this method requires minimum computation and world representation. However, a system having no deliberative planning generally results in unreliable decision making. Consequently, the robot is unable to find optimal trajectories or to select the best sense-to-action mapping corresponding to the assigned task [43]. The hybrid paradigm, as shown in [44] integrates deliberation and planning with the reactive control and aims to harness the best of both architectures. The hybrid architecture is composed of three layers; the deliberative layer handles high-level issues such as global path planning, the reactive layer manages low-level control problems such as obstacle avoidance, and the middle layer represents an interface to the deliberative and reactive layers and acts as a coordinator or sequencer, which selects the correct primitive behavior(s) to control the robot [45]. However, interfacing different components poses extra complexity [40]. Moreover, this approach ignores issues related to sensor processing and learning [45]. Behavior-based architectures are composed of sets of independent and concurrently operating modules called behaviors [46]. Each behavior represents a control law, which keeps a set of constraints for achieving and maintaining a certain goal [40]. These systems can integrate both reactive and deliberative components into their behaviors and offer an alternative to hybrid control [40]. These systems lack the centralized state representation, and the network of behaviors maintain the state information at a given time. Behavior-based systems provide excellent real-time performance and can be used to integrate several goal oriented behaviors simultaneously. The behaviors are coordinated by a behavior coordination mechanism, which selects a behavior or a set of behaviors to accomplish a task in an optimized way [39].

The behavior coordination or the action selection of behavior-based mobile robotics can be further classified as either *command arbitration* or *command fusion* [47]. Command arbitration is described as selecting one behavior from a group of competing behaviors, which is simple and effective in most reactive situations such as avoiding an obstacle or moving towards an environmental stimulus. However, due to the behavior suppression in the arbitration method the robot may lose its original planned behavior, which results in an erroneous decision during navigation and leads to instability and starvation [48, 49]. Command fusion mechanisms such as in [50, 51], which combine the recommendations of multiple behaviors, can overcome this problem. However, when competitive behaviors issue conflicting commands the control of the robot yields a local minima, which is a common problem in robotic control [47]. Weighted decision making is introduced in [52] to address this issue where the conflicting commands are weighted according to predefined priorities, and this technique has been successfully employed in behavior-based robot control [53, 54].

2.2.1 Multi robot coordination

A Multi Robot System (MRS) consists of a group of concurrently operating robots, which are interacting with each other and performing a given task collaboratively. With the improved efficiency, availability, robustness and cost-effectiveness, employing a distributed MRS is superior and even essential than having a single centralized robot [55]. Also deploying an MRS instead of an SRS improves the performance and the reliability of the overall system [56]. However, utilization of an MRS over an SRS requires addressing the complexity introduced by multiple interacting robots [40]. The MRS has to be carefully designed in order for each robot to work towards a common goal cooperatively. In order to generate consistent task-directed behaviors for a group of interacting robots, the coordination scheme has to systematically arrange these interactions. Therefore, as expected, scaling the MRS requires employing a more robust, fault-tolerant, simple yet effective coordination scheme.

2.2.2 Distributed architectures for multi robot coordination

A notable architecture, which represents a fully distributed and robust behavior-based control is the ALLIANCE (1998) [57–59]. In this architecture each robot has several sets of behaviors, which are activated by *motivational behaviors*. Each motivational behavior receives information from sensory feedback, inter-robot communication, cross-inhibition from other active behaviors and internal motivations. The fault tolerance is achieved via the use of motivations, which are designed to make the robots perform tasks only if they exhibit their ability. ALLIANCE represents a subsumptive-style command arbitration and hence inherits drawbacks such as instability and starvation [48, 49].

Another architecture that delineates a highly distributed, hybrid and behavior-based approach is the CAMPOUT (Control Architecture for Multi robot Planetary Outposts) (2000), which is designed to achieve cooperative control of heterogeneous robot platforms implemented in planetary rover systems [60, 61]. CAMPOUT depicts a behavior hierarchy implemented on each robot of the group. A behavior coordination mechanism, which supports both behavior arbitration and command fusion selects the best combination of behaviors. CAMPOUT lacks high-level planning capabilities, which results in less optimal task execution.

A formation control task of MRS is shown in [62] using the behavior-based paradigm (1999). The behaviors are implemented as motor schemas and coordination is performed by command fusion. Positional information is transmitted through explicit communication between the peers or from leader to followers and each robot formation position is determined by the perceptual schemas. The effectiveness and portability of the approach is demonstrated by implementing it in two different reactive robotic architectures, namely the Autonomous Robot Architecture (AuRA) [63] and the Unmanned Ground Vehicle (UGV) Demo II architecture [64].

EMERGE (1999) is a distributed, behavior-based architecture, which supports flexible and reliable coordination in a heterogeneous robot group [65]. This consists of a set of high-level strategic behaviors and low-level survival behaviors. Activation of a strategic behavior is based on mathematically-modeled motivations. The low-level behaviors are continually executed with the chosen high-level behavior. However, this approach also employs a priority-based arbitration mechanism and designing the strategic behaviors to represent all possible situations is challenging and tedious.

The Port Arbitrated Behavior paradigm (PAB) is a set of abstractions and techniques for behavior integration in behavior-based systems, which supports port-based messaging, behavior suppression, inhibition and overriding [66]. AYLLU (2000) is an architecture, which extends the standard PAB paradigm to be used in a multi robot domain by providing PAB interactions over IP networks [66]. Here the coordination between robots is achieved by the Broadcast of Local Eligibility (BLE) [67] where behaviors of each robot locally determine the eligibility for performing a task. These eligibilities are then exchanged and compared among the peer behaviors and the chosen behavior claims the task by inhibiting others. This process is termed *cross-inhibition*. *Cross-subsumption*, which is the combination of both cross-inhibition and local subsumption, provides robust, scalable and flexible team cooperation. However, this approach relies on explicit communication and it employs a variety of arbitration methods for behavior coordination.

MURDOCH (2001) is a task allocation system for MRS based on a distributed negotiation protocol [68]. Its communication is based on publish/subscribe messaging. This is addressed by the content, which is suitable when the teams are dynamically arranged for different tasks. One is selected among the capable robots based on the available *metrics* of each. However, this approach uses extensive inter-robot communication and suffers from making locally optimal choices, as greedy algorithms are employed in task scheduling.

Dynamic role assignment of an MRS based on the explicit communication of *utility functions*, which evaluate the task performing capability of each robot respective to their roles, is presented in [69]. The coordination protocol implemented in each robot negotiates with others and exchanges information regarding role assignment and formation selection. The approach is robust to communication failures and can be operated in dynamic and hostile environments. It also facilitates integration of heterogeneous robots having different behavior coordination architectures. However, successful employment of the approach depends on the calibration of utility function coefficients, which is tedious and requires a significant experimental work.

A multiple objective behavior coordination-based approach, which employs command fusion across the group of robots, is presented in [70]. Here the behaviors are designed using fuzzy rule bases and their weights are generated by context-dependent blending [71]. Each robot selects actions which are beneficial to all, and the approach enables achieving multiple goals of multiple robots in parallel. With the increase of group size, information exchange between the members can be critical and hence poses a key problem to solve. Also, writing fuzzy rules to cover the entire search space is cumbersome.

Design of behavior-based controllers for collection tasks is presented in [72]. The robustness is achieved by individually managing the noise and complexity of the environment without inter-robot communication. This mechanism is however, not very acceptable as it does not represent efficient usage of resources. The controllers are then modified to accommodate the inter-robot communication and exchange of status information. Also, a hierarchy is assigned among robots to recognize the rank of each robot. However, the approach is highly task dependent and the controllers have to be redesigned before transforming into different domains. Also, the controllers are based on behavior arbitration.

A layered architecture, which is composed by enabling cross interactions between each level of the three-layered hybrid architecture, is presented in [73]. The lowest layer maintains distributed feedback control loops by enabling sensor data and status information to flow. The executive layer handles decomposition of tasks into subtasks, exchanging the task synchronization information, execution monitoring and fault recovering. Plans are shared and resources are scheduled among the peers at the planning layer, which is developed based on market economy. However, this approach still inherits the limitations of three-layered architectures and the necessity of high band-width and low-latency inter-robot communication adds extra overhead.

In [74], an architecture is proposed to control multiple heterogeneous robots. This consists of three behavioral levels, namely *individual, collective* and *social*. The individual behaviors represent the bottom level and directly control the robot actuators. In contrast with [73], at this level no inter-robot communication is performed. These individual behaviors can be modified by collective behaviors, which represent the second level of the architecture. These are capable of providing group cooperation and collaboration based on the information obtained from inter-robot communication as well as the robots' own sensors. The top level consists of social behaviors, which also receives information from both sources and imposes rules on the collective behaviors. The task assignment is performed using an auction-based strategy.

ABBA (1999) is an architecture for behavior-based agents which achieves cooperative planning by distributing the action selection mechanism throughout the network [75]. Two types of nodes named *Competence Modules* (CMs) and *Feature Detectors* (FDs) are used to represent the behavior of the system. Each CM implements a behavior and FDs deliver information from sensors. The desired performance is achieved by connecting CMs with FDs using preconditions and correction links. As only one CM can be active at any given time, this approach shows an arbitration scheme. Also, the corrections have to be learned during the task execution, which increases the computational cost.

CHARON (2002) is a formal architecture, that facilitates specifying multi agent systems with multiple behaviors and communication protocols in a principled way [76]. It also provides a modular and hierarchical approach to program the behaviors. The *robot-group agent* represents robots that are communicating and exchanging information with each other. Each *robot agent* consists of an *estimator agent* to represent sensors and a *control agent* to switch between the behaviors. The performance is achieved by *parallel composition* of sub agents. Changes of overall behavior of the system is achieved by *sequential composition*. Deliberative and reactive behaviors within each robot are composed by *hierarchical composition*. The approach employs arbitration for action selection.

A decentralized approach to control behavior-based MRS is presented in [77], based on a multiple subsumption architecture, which represents a stimulus response structure. These stimuli carry information and they are exchanged through inter-robot communication. Low-level behaviors of a robot can be suppressed by a higher-level behavior of the same or a different one. The group members coordinate their behaviors based on a priority-based arbitration.

A framework that represents a layered behavior-based architecture is proposed in [78] and it is used in formation control of MRS. The lowest level includes the behaviors, which ensure the safety of the robot, the middle-level consists of safe wandering and the highest level represents the decentralized formation controller. A priority-based arbitration selects actions of group members. The behavior with the highest priority represents message passing through explicit communication between robots, which allows them to switch between the formation and navigation behaviors.

2.2.3 Fuzzy logic-based approaches

Fuzzy Logic (FL) is capable of managing sensor uncertainty and decision vagueness by approximate reasoning. Also it is robust against perturbations and facilitates specification of control commands using linguistic terms [79]. Fuzzy approaches ease the behavior definition and blending when designing and implementing robot control mechanisms, as opposed to the non-fuzzy counterparts. FL-based methods employed in SRS mainly focus on developing behaviors using fuzzy rule bases, utilizing behavior hierarchies and coordination of behaviors using command fusion [71, 80, 81]. These can be easily extended to the MRS domain facilitating robot cooperation and coordination.

In [82] an architecture is presented with a hierarchical organization of heterogeneous robots. The bottom level of the hierarchy represents *scouts*, having a minimum level of autonomy, such as basic navigation and self-preservation. The middle level consists of *coordinator* robots with more deliberative capabilities. The roles of scouts and coordinators can be interchanged. The top layer represents a *supervisor* for *human-in-the-loop* situations. Inter-robot communication is used to exchange map information and mission directives. A fuzzy behavioral hierarchy is employed in the design. The lowest level comprises a collection of primitive behaviors, which operate in a reactive way. The composite behaviors are composed by integrating several primitive behaviors and represent higher level coordination. The approach has been successfully applied to a foraging mission of identifying target objects.

A collection of robots cooperatively tracking a dynamic target is described in [83]. The presented architecture accommodates more than one behavior within a layer by modifying the subsumption architecture. Each behavior in a layer has the same priorities and operates concurrently. Behaviors are implemented in fuzzy rule bases and conclusions are inferred through a fuzzy inference system (FIS). Multiple recommendations are fused and defuzzified to generate the final control command.

2.2.4 crisp DES-based approaches

The crisp DES framework has a sound mathematical foundation and it includes well-established formal methods, which help to analyze the system properties methodically. DES-based modeling and control of MRS has proven successful in the literature.

A finite state automata (FSA)-based modeling of an MRS, which is playing football games, is described in [84]. The environmental dynamics, robot behaviors, and additional restrictions are modeled as separate sub-automata. Controllable and uncontrollable events are identified and the optimal sequence of controllable events for each robot is determined based on minimizing a cost function and the uncertainty of the occurrence of uncontrollable events. The football game automaton is achieved by parallel composition of all sub-automata. The final states correspond to scoring a goal successfully by the team. A cooperative behavior selection policy is utilized to determine the cost function.

Representation and execution of multi robot behaviors using Petri net plans is discussed in [85]. Design of cooperative behaviors is based on joint commitment theory, which is expressed through Petri nets. Explicit communication is used to maintain the synchronization between robots and pass the interrupts in the case of failure. The approach has been successfully employed in the development of cooperative behaviors of robotic soccer teams.

A crisp DES-based supervisory control model for coordination and space sharing of MRS is presented in [86,87]. The approach is capable of formally ensuring collision and deadlock avoidance of the robots, which are concurrently accomplishing their own tasks. Each robot plans its own trajectory independently. Robots dynamically modify their paths and velocity profiles for collision avoidance. Communication between robots is performed to broadcast the local states.

In [88] control of a mobile robot population operating in a discrete environment is modeled in the crisp DES framework. The MRS is modeled as a general FSA and the control of each robot is modeled by automata with smaller dimensions referred to as *navigation automata*. Properties of the main automaton such as blocking, controllability and observability are analyzed and related to those of navigation automata. A decentralized control architecture is employed where the reachability of the global objective of the population is verified by assessing the conditions on navigation automata.

2.2.5 FDES-based approach to control SRS

Huq et al. proposed an FDES-based behavior coordination technique to successfully navigate a mobile robot in an unmodeled environment [10, 89, 90]. In this approach the behaviors are represented as fuzzy automata and the sensory information is used by a fuzzy rule-based system to activate the events, which lead to state transitions. State-based fuzzy observability and controllability measures are also incorporated with final decision making. However, the presented behavior representation leads to state explosion by adding multiple behaviors. Moreover, this approach does not associate formal methods based on the well-established DES framework.

2.2.6 A taxonomy for MRS

The presented distributed architectures of MRS can be classified according to their utilized behavior coordination mechanisms and tabulated as in Table 2.1. Note that the classification of action selection mechanisms that we used here is originally proposed in [47].

Application domain	Behavior coordination mechanism						
	Command arbitration			Command fusion			Not specified
	Priority -based	State -based	Winner -take-all	Voting	Fuzzy	Multiple objective	
Foraging	[57], [72]	[63]	[75]		[82]		[74]
Target tracking	[58], [67], [66]				[83]		
Object manipulation	[60], [61]	[60], [61], [76]			[60], [61]	[60], [61]	[68]
Exploration	[78]	[63], [76], [88]		[64]	[82]		[73]
Soccer games	[65]	[84], [85]					[69]
Target acquisition						[70]	
Robotic assembly							[73]
Localization		[76]					

Table 2.1: A classification of MRS based on behavior coordination

2.2.7 Properties of a better coordination mechanism

The literature suggests several properties that make a better coordination mechanism. They are described as follows.

- **Robustness:** This is realized by having least number of single point failures. A robust coordinator does not hinder the performance of entire system even though there are malfunctioning sub systems. Hence, it supports graceful degradation of the performance.

- **Modularity:** The coordinator should support integration of sub systems in modular manner, which also in turn increases the robustness. The addition of new components must not be cumbersome and the infrastructure must provide meaningful interactions. By integrating new components the performance of the entire system should be improved.

- **Scalability:** The implemented architecture should be general in order to handle more complex problem domains and increased number of components, robots etc. Scaling up the mechanism must also provide less computational overhead to the system. Having a coordinator, which supports modularity, increases the scalability of the system.

- **Uncertainty handling:** Under the presence of noisy sensors and actuators, the internal model of the environment becomes less accurate and hence decrease the performance. A proper uncertainty handling mechanism must address every possible kind of ambiguities efficiently so that the reliable operation of the robot group can be established.

- **Distributed control:** To avoid complexities and single point failures, the control must be distributed over the group members. The inter-robot communication can be employed for information exchange and select the best action for each robot.

- **Supports negotiation:** In multi robot task allocation, the robots must be negotiated and select the best tasks depending on the available information and individual sensing/acting capabilities. The robot with best capabilities should win the task and inhibit other robots. The implemented coordination scheme must support such an approach.

- **Task breakdown:** The coordinator must be able to further decompose the tasks into simple and less complex elementary subtasks. This must be performed in group-level before task allocation, as well as within each robot to subdivide the allocated task. Having such a modular approach simplifies the task accomplishment.

- **Reactive and deliberative:** The employed coordination mechanism should have the ability to respond promptly when unexpected situations of the environment occurs. Furthermore, it should be able to plan for achieving the tasks with efficient use of resources.

- **Having formal framework:** This helps to formally design the system according to the requirements and specifications. Moreover, having such a framework supports analysis of the system performance using formal methods.

- **Knowledge representation:** The utilized mechanism should support representation of knowledge by modeling of world states. As a result, it can select the appropriate actions at a given world state which in turn increases the performance.

- **Weighted decision making:** The employed behavior coordination mechanism within each robot must support the weighted behavior coordination. This satisfies achieving several goals simultaneously and provides more successful task accomplishment.

2.2.8 Proposed approach

An FDES-based approach provides a formal framework for system modeling and control. It helps to capture the deterministic uncertainties of a system and the decentralized control of FDES addresses the control problem of large-scale systems in a less complex and more robust manner. Furthermore, its fuzzy state-based representation describes the world state more accurately, can be used for weighted decision making and supports modularity in behavior integration. Also, its event-based processing is suitable for reactive situations. These resemblances show that behavior coordination of robotic tasks serves as a favorable application area for FDES-based modeling and control.

This research work endeavors to examine the supervisory control of FDES broadly and to apply these developments for behavior coordination of mobile robotics to achieve some of the properties mentioned in 2.2.7.

Chapter 3

Single robot control using FDES

In this chapter, a supervisory control framework that can be effectively used for behavior-based mobile robot navigational tasks is developed. First, some preliminaries of FDES theory are presented with some extensions. Then, the supervisory control theory of FDES is extended and formalized to represent the behavior coordination of SRS. The approach is verified by performing both simulation and real-time experiments on a mobile robot navigating in unmodeled environments.

3.1 Preliminaries

Throughout this thesis, we adopt the following symbols for logical operations.

$\cup$: Set union

$\cap$: Set intersection

$\tilde{\cup}$: Fuzzy-union operator (take the maximum)

$\tilde{\cap}$: Fuzzy-intersection operator (take the minimum or algebraic product)

The variables are defined while following the notations in [1]. Unless otherwise stated, all variables defined in this thesis belong to the class of fuzzy variables. The crisp variables are treated as a special case of the fuzzy variables.

A fuzzy finite automaton is denoted by the quadruple: $G = (Q, \Sigma, \delta, q_0)$, where Q is the set of fuzzy states, Σ is the set of fuzzy events, δ represents the transition mapping, $\delta : Q \times \Sigma \to Q$ and q_0 represents the initial fuzzy state vector [19].

A fuzzy state vector $q(\in Q)$ can be represented as: $q = [\mu_k]_{1\times n}$, where $n = |Q|$, $k \in (1, .., n)$ and $\mu_k(\in [0, 1])$ is the degree of membership of k^{th} fuzzy state in q.

A fuzzy event $\sigma(\in \Sigma)$ can be represented by a matrix as: $\sigma = [\lambda_{i,j}]_{n\times n}$, where $\lambda_{i,j}$ shows the possibility of the system to transit from state i to j and $\lambda_{i,j} \in [0, 1]$.

The transition mapping δ is calculated as: $\delta(q, \sigma) = q \circ \sigma$. Here "$\circ$" represents either Max-Min or Max-Product operation [91]. Generally, these operations can be defined as follows.

Let $A = [a_{i,j}]_{k,p}$ and $B = [b_{i,j}]_{p,m}$ be two matrices. Assume $A \circ B = [c_{i,j}]_{k,m}$.

With Max-Min operation: $c_{i,j} = \max\{\min(a_{i,h}, b_{h,j}) \mid h = 1, ..., p\}$.

With Max-Product operation: $c_{i,j} = \max\{a_{i,h}b_{h,j} \mid h = 1, ..., p\}$.

Assume ε as the null event (or empty string) and Σ^* represents the Kleene-closure of Σ ($\varepsilon \in \Sigma^*$). Clearly, ε can be represented by an identity matrix.

A fuzzy language L, which is generated by a fuzzy automaton, is characterized by the continuous occurrence of fuzzy events. It can be represented using Zadeh's notation [92] as:

$L = \{\frac{\mu_L(x)}{x}\}$, $x \in \Sigma^*$, and $\mu_L(x)(\in [0, 1])$ is the degree of membership of x in L. $L(x)(= \mu_L(x))$ is defined as the possibility of fuzzy string of events x belonging to L. Note that we write $x \in L$ when $\mu_L(x) > 0$.

For example: $L = \frac{0.4}{\alpha\zeta} + \frac{0.3}{\alpha\beta\gamma} + \frac{0.4}{\alpha\beta\delta}$, where $\alpha, \beta, \gamma, \delta, \zeta \in \Sigma$.

Its prefix-closure $\bar{L}$ is also a fuzzy language, which shows how the fuzzy events have evolved.

$\bar{L} = \{\frac{\mu_{\bar{L}}(y)}{y}\}$, $y \in \Sigma^* : \exists t \in \Sigma^*$ such that $yt \in L$ and $\mu_{\bar{L}}(y) \geq \mu_L(yt)$.

For example: $\bar{L} = \frac{1}{\varepsilon} + \frac{0.8}{\alpha} + \frac{0.5}{\alpha\beta} + \frac{0.4}{\alpha\zeta} + \frac{0.3}{\alpha\beta\gamma} + \frac{0.4}{\alpha\beta\delta}$.

A fuzzy sublanguage L_{sub} of L ($L_{\text{sub}} \subseteq L$) is defined as: $\forall s \in \Sigma^*, L_{\text{sub}}(s) \leq L(s)$. Intuitively, $L \subseteq \bar{L}$.

Note that hereafter we specify the fuzzy languages in their prefix-closed forms, which show the way they have evolved over time.

Let $G_1 = (Q_1, \Sigma_1, \delta_1, q_{01})$ and $G_2 = (Q_2, \Sigma_2, \delta_2, q_{02})$ be two fuzzy automata. Their parallel composition, $G_1 \parallel G_2$ makes a new fuzzy automaton as follows [4,6].

$$G_1 \parallel G_2 = (Q_1 \otimes Q_2, \Sigma_1 \cup \Sigma_2, \delta_1 \parallel \delta_2, q_{01} \otimes q_{02}) \tag{3.1}$$

Here, $Q_1 \otimes Q_2 = \{q_1 \otimes q_2 : q_1 \in Q_1, q_2 \in Q_2\}$ and "$\otimes$" operation denotes the tensor product. For two events $\sigma_1 \in \Sigma_1$ and $\sigma_2 \in \Sigma_2$, an event in the combined system $\sigma(\in \Sigma_1 \cup \Sigma_2)$ can be defined as follows [4,6].

$$\sigma = \begin{cases} \sigma_1 \otimes \sigma_2, & \text{if } \sigma \in \Sigma_1 \cap \Sigma_2 \\ \sigma_1 \otimes I_2, & \text{if } \sigma \in \Sigma_1 \setminus \Sigma_2 \\ I_1 \otimes \sigma_2, & \text{if } \sigma \in \Sigma_2 \setminus \Sigma_1 \end{cases} \tag{3.2}$$

where I_1 and I_2 are identity matrices having the order of $|Q_1|$ and $|Q_2|$ respectively. For $q_1 \otimes q_2 \in Q_1 \otimes Q_2$ and $\sigma \in \Sigma_1 \cup \Sigma_2$, the transition mapping of the composition is given as follows.

$$(\delta_1 \parallel \delta_2)(q_1 \otimes q_2, \sigma) = (q_1 \otimes q_2) \circ \sigma \tag{3.3}$$

Fuzzy controllable event set and fuzzy uncontrollable event set are denoted by Σ_c and Σ_{uc} respectively. Also, fuzzy observable event set and fuzzy unobservable event set are denoted by Σ_o and Σ_{uo} respectively. Then, $\Sigma_c \cup \Sigma_{uc} = \Sigma$ and $\Sigma_o \cup \Sigma_{uo} = \Sigma$. For any fuzzy event $\sigma \in \Sigma$, the degree of σ being controllable is denoted by $\Sigma_c(\sigma)$. In the same manner, $\Sigma_{uc}(\sigma)$, $\Sigma_o(\sigma)$, and $\Sigma_{uo}(\sigma)$ have their respective meanings.

As in [4], the partial observability of an event is bounded as follows.

$$\Sigma_o(\sigma) + \Sigma_{uo}(\sigma) = 1 \tag{3.4}$$

Assume the language generated by fuzzy automaton G represented by L_G. This includes all paths that can be followed by δ. By definition L_G is a prefix-closed fuzzy language (i.e., $L_G = \bar{L}_G$). The language *marked* by G represents the fuzzy strings, which end at the "final" or "accepted" states. It is denoted by $L_{G,m}$.

The behavior of an FDES (plant) also can be represented by two fuzzy languages: L is the uncontrolled behavior of the FDES and L_m represents its successfully completed operations. Hence, without loss of generality, assuming $L_G = L$ and $L_{G,m} = L_m$, an FDES can be modeled by a fuzzy automaton G. Therefore, sometimes we will write "FDES G" for convenience.

Then, $L_G(s)$ can be referred to as physical possibility of occurring s, where $s \in \Sigma^*$.

Let S be the supervisor of FDES G. Then S/G denotes "S controlling G" and $L_{S/G}$ is its corresponding fuzzy language. The fuzzy language generated by supervisory control of FDES ($L_{S/G}$) is defined recursively as follows [6].

$$\begin{aligned} &1.L_{S/G}(\varepsilon) = 1 \\ &2.L_{S/G}(s\sigma) = L_{S/G}(s)\tilde{\cap}S_s(\sigma)\tilde{\cap}L_G(s\sigma) \end{aligned} \tag{3.5}$$

where $S_s(\sigma)$ is the degree of fuzzy event σ being enabled by the supervisor S, after observing the fuzzy string s. Here $L_{S/G} \subseteq L_G$ and it is prefixed-closed.

A new fuzzy language $\bar{L}_{S/G,m}$, which is a sublanguage of $\bar{L}_{G,m}$ and contains only those marked fuzzy strings that survived under S/G, can be achieved as: $\bar{L}_{S/G,m} = L_{S/G} \cap \bar{L}_{G,m}$.

The possibility of fuzzy string s ($\in \Sigma^*$) belonging to $\bar{L}_{S/G,m}$ can be defined as follows.

$$\bar{L}_{S/G,m}(s) = L_{S/G}(s) \tilde{\cap} \bar{L}_{G,m}(s) \Rightarrow \bar{L}_{S/G,m}(s) \leq \bar{L}_{G,m}(s) \tag{3.6}$$

The fuzzy language $L_{S/G}$ is called "non-blocking" if it is exactly the same as the prefix-closure of $L_{S/G,m}$.

$$\forall\ s \in \Sigma^* : L_{S/G}(s) = \bar{L}_{S/G,m}(s) \tag{3.7}$$

Assume a prefix-closed fuzzy language specification $\bar{k}(\subseteq L_G)$ is given. Then $\bar{k}$ is said to be $L_{G,m}$-closed if:

$$\bar{k}(s) \leq \bar{L}_{G,m}(s). \tag{3.8}$$

Furthermore, $\bar{k}$ is said to be satisfying the fuzzy controllability condition [23] with respect to L_G and Σ_{uc}, for all $s \in \Sigma^*$ and $\sigma \in \Sigma$ if the following inequality holds.

$$\bar{k}(s) \tilde{\cap} \Sigma_{uc}(\sigma) \tilde{\cap} L_G(s\sigma) \leq \bar{k}(s\sigma) \tag{3.9}$$

This means the possibility of fuzzy string $s\sigma$ belonging to $\bar{k}$ is greater than or equal to the minimum (or product) of the following.

1. Possibility of fuzzy string s belonging to $\bar{k}$.
2. The degree of fuzzy event σ being uncontrollable.
3. The physical possibility of $s\sigma$.

Definition 3.1: The natural projection of σ is defined as follows.

$$P(\sigma) = [\Sigma_{uo}(\sigma)\varepsilon + \Sigma_o(\sigma)\sigma] \tag{3.10}$$

This means that the matrix representing the natural projection of σ can be achieved by multiplying each element of the identity matrix representing ε by $\Sigma_{uo}(\sigma)$ and adding them together with the corresponding elements of the matrix, which is made by multiplying each element of the event matrix representing σ by $\Sigma_o(\sigma)$.

As the unobservability of a fuzzy event σ increases, $P(\sigma)$ reaches ε and the supervisor tends not to observe σ. Similarly, when the observability of σ increases, the supervisor tends to observe σ.

Assume $s = \sigma_1\sigma_2 \ldots \sigma_n$. Let $P(s)$ be the natural projection of s. The following is obtained by considering the natural projection of each fuzzy event individually.

$$P(s) = P(\sigma_1\sigma_2 \ldots \sigma_n) \Rightarrow P(\sigma_1)\, P(\sigma_2) \ldots P(\sigma_n) \tag{3.11}$$

The fuzzy admissibility condition defined in [6] is extended by introducing partial observation supervisory control as follows.

$$\Sigma_{uc}(\sigma) \tilde{\cap} L_G(s\sigma) \leq S_t^P(\sigma) \tag{3.12}$$

where S^P is the partial observation supervisor, $P(s) = t$ and $S_t^P(\sigma)$ is the degree of σ being enabled by S^P after observing t. The following can be derived from the above.

$$L_{S^P/G}(s\sigma) = L_{S^P/G}(s) \tilde{\cap} S_t^P(\sigma) \tilde{\cap} L_G(s\sigma) \tag{3.13}$$

where $L_{S^P/G}$ is the fuzzy language generated by the partial observation supervisor S^P, controlling the FDES G.

Note that $P^{-1}[P(s)]$ denotes the set of all strings, which have the same (or with slight differences) natural projection of s. Then, for $s, s' \in \Sigma^*$, $P^{-1}[P(s)](s')$ defines the degree of $P(s')$ to be seen as same as $P(s)$ (assuming $P^{-1}[P(s)](s) = 1$). This definition identifies the "likelihood" of a (slightly) different fuzzy string to be observed as an already known one.

Extending the fuzzy observability defined in [23] we can derive the following definition for fuzzy observability.

Definition 3.2: Let $\bar{k} \subseteq L_G$, $s'\sigma \in \bar{k}$ and $s \in P^{-1}[P(s')]$. For any $s \in \Sigma^*$ and $\sigma \in \Sigma$, $\bar{k}$ is said to be satisfying the fuzzy observability condition with respect to L_G, P and Σ_c, if the following inequality holds.

$$\bar{k}(s)\tilde{\cap}L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P^{-1}[P(s')](s)\tilde{\cap}\Sigma_c(\sigma) \leq \bar{k}(s\sigma) \tag{3.14}$$

The possibility of fuzzy string $s\sigma$ belonging to $\bar{k}$ is greater than or equal to the minimum (or product) of the following.

1. The possibility of s belonging to $\bar{k}$.
2. The physical possibility of $s\sigma$.
3. The possibility of $s'\sigma$ belonging to $\bar{k}$.
4. The degree of $P(s)$ to be seen as same as $P(s')$.
5. The degree of σ being controllable.

The *Definition 3.2* is important as it adds an extra dimension to the fuzzy observability condition presented in [23].

Assuming $P(s) = t$ and $s'\sigma \in \bar{k}$, following can be derived for partial observation supervisory control.

$$\forall\ \sigma \in \Sigma : S_t^P(\sigma) \geq L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P^{-1}[P(s')](s)\tilde{\cap}\Sigma_c(\sigma) \tag{3.15}$$

The terms have been described under *Definition 3.2*.

Definition 3.3: Combining (3.12) and (3.15), we can define $S_t^P(\sigma)$ as follows.

Let $\mu_1 = \Sigma_{uc}(\sigma)\tilde{\cap}L_G(s\sigma)$ and $\mu_2 = L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P^{-1}[P(s')](s)\tilde{\cap}\Sigma_c(\sigma)$.

For any $\sigma \in \Sigma$:

$$S_t^P(\sigma) = \begin{cases} \mu_1, & \text{if } \mu_1 \geq \mu_2 \text{ and } \mu_1 \geq \bar{k}(s\sigma) \\ \mu_2, & \text{if } \mu_2 > \mu_1 \text{ and } \mu_2 \geq \bar{k}(s\sigma) \\ \bar{k}(s\sigma), & \text{otherwise.} \end{cases} \tag{3.16}$$

This explains the degree of σ being enabled by the partial observation supervisor S^P, according to the degree of σ being uncontrollable and the degree of σ being controllable.

Theorem 3.1: Fuzzy controllability and fuzzy observability theorem.

There exists a non-blocking partial observation supervisor S^P for the FDES G such that $\bar{k}(s) = \bar{L}_{S^P/G,m}(s)$ and $L_{S^P/G}(s) = \bar{k}(s)$, if and only if the following conditions hold.

1. $\bar{k}$ is fuzzy controllable with respect to L_G and Σ_{uc}.
2. $\bar{k}$ is fuzzy observable with respect to L_G, P and Σ_c.
3. $\bar{k}$ is $L_{G,m}$-closed.

Proof: See Appendix.

Note that with the deferences in fuzzy observability conditions discussed in this thesis and in [23], the above theory differs from the one presented in [23].

3.2 Supervisory control of FDES

3.2.1 Problem definition

Assume a scenario where a mobile robot need to navigate in a known environment with some unmodeled obstacles. The navigation objective is to reach a target point in the environment while avoiding any obstacles that can appear in the path. The system has two behaviors, namely "Avoid obstacles"(AO) and "Go to target" (GT). It is desired to have a command fusion type behavior coordination between behaviors [47].

This behavior-based system can be considered as an FDES and it is modeled by a fuzzy automaton as shown in figure 3.1. The behaviors are represented by fuzzy states AO and GT. Two fuzzy events σ_1, σ_2 are defined for transitioning to fuzzy states AO and GT respectively. Here, $Q = \{AO, GT\}$, $\Sigma = \{\sigma_1, \sigma_2\}$ and δ is shown in the figure. Furthermore, we assume the deliberative behavior (i.e., GT) is controlled by fuzzy controllable events and reactive behavior (i.e., AO) is controlled by fuzzy uncontrollable events. This leads to $\sigma_1 \in \Sigma_{uc}$ and $\sigma_2 \in \Sigma_c$.

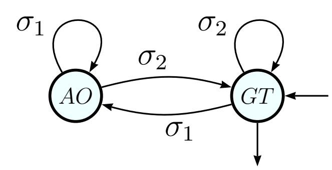


Figure 3.1: Fuzzy automaton modeling two behaviors

Assume these fuzzy events are triggered by the supervisory control based on the sensory perceptions available to the robot. The fuzzy language specification $\bar{k}$ is given according to the initial deliberative planning made with respect to the modeled environment having a clear path by assuming $L_G(s) = 1$, where $s \in \Sigma^*$. This consists of the fuzzy controllable event only as shown below.

$$\bar{k} = \frac{1}{\varepsilon} + \frac{1}{\sigma_2} + \frac{1}{\sigma_2\sigma_2} + ... \tag{3.17}$$

Note that in this case the language specification $\bar{k}$ is crisp as the deliberative planning activates only one controllable event repetitively (i.e., σ_2) with respect to the modeled environment. This is true at the beginning as the robot assumes there are no obstacles in the path.

When the unmodeled obstacles are present in the environment, the physical possibility distribution of the fuzzy events differs from its previous assumption. For example, assume the new physical possibility distribution of the fuzzy strings found to be as follows.

$$L_G = \frac{1}{\varepsilon} + \frac{0.9}{\sigma_1} + \frac{0.8}{\sigma_2} + \frac{0.9}{\sigma_1\sigma_1} + \frac{0.8}{\sigma_1\sigma_2} + \frac{0.8}{\sigma_2\sigma_1} + \frac{0.8}{\sigma_2\sigma_2} \tag{3.18}$$

When the robot comes closer to an unmodeled obstacle, the supervisor enables the feasible fuzzy uncontrollable event (i.e., σ_1) to drive the system to state AO partially and hence avoiding the obstacle. For example, assume the new fuzzy language specification $\bar{k}'$ found to be as follows.

$$\bar{k}' = \frac{1}{\varepsilon} + \frac{0.7}{\sigma_1} + \frac{0.3}{\sigma_2} + \frac{0.7}{\sigma_1\sigma_1} + \frac{0.3}{\sigma_1\sigma_2} + \frac{0.3}{\sigma_2\sigma_1} + \frac{0.3}{\sigma_2\sigma_2} \tag{3.19}$$

The objective now is to achieve $\bar{k}'$ by using the supervisory control of FDES.

3.2.2 Identification of Σ_c and Σ_{uc}

Note that this chapter assumes fuzzy controllable and uncontrollable event sets are mutually disjoint (i.e., $\Sigma_c \cap \Sigma_{uc} = \emptyset$). The relaxation of this assumption can be found in next chapter.

The degree of σ being uncontrollable is defined as follows.

$$\Sigma_{uc}(\sigma) = \begin{cases} X_\sigma \in [0,1], & \text{if } \sigma \in \Sigma_{uc} \\ 0, & \text{if } \sigma \in \Sigma \setminus \Sigma_{uc} \end{cases} \tag{3.20}$$

Here, X_σ is the value obtained from the defuzzification step of a fuzzy rule base, which is constructed for the reactive behavior of the robotic system.

The degree of σ being controllable is defined as follows.

$$\Sigma_c(\sigma) = \begin{cases} L_G(s\sigma)\tilde{\cap}\bar{k}(s\sigma), & \text{if } \sigma \in \Sigma_c,\ s\sigma \in \bar{k} \text{ and } s\sigma \in \bar{k}' \\ L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma), & \text{if } \sigma \in \Sigma_c,\ s\sigma \notin \bar{k},\ s\sigma \in \bar{k}',\ s'\sigma \in \bar{k} \text{ and } |s\sigma| = |s'\sigma| \\ 0, & \text{if } \sigma \in \Sigma \setminus \Sigma_c \end{cases} \tag{3.21}$$

Assume the prefix-closed fuzzy languages $\bar{k}$ and $\bar{k}'$ are as defined in (3.17) and (3.19). The following cases explain (3.21). Note that σ in (3.21) is replaced with σ_2 for following explanation.

Case I: $s\sigma_2 \in \bar{k}$ and $s\sigma_2 \in \bar{k}'$

In this case the degree of σ_2 being controllable is given by the minimum of the possibility of fuzzy event $s\sigma_2$ belonging to $\bar{k}$ and the physical possibility of occurring $s\sigma_2$. Let $s = \sigma_2$ and consider the problem definition in 3.2.1. The fuzzy string $s\sigma_2$ (which is $\sigma_2\sigma_2$) is available in both $\bar{k}$ and $\bar{k}'$. Here, $\Sigma_c(\sigma_2) = min\{0.8,\ 1\} = 0.8$.

Case II: $s\sigma_2 \notin \bar{k}$ and $s\sigma_2 \in \bar{k}'$

In this case there exists another fuzzy string $s'\sigma_2$ in $\bar{k}$ where the sizes of the fuzzy strings $s'\sigma_2$ and $s\sigma_2$ are identical (i.e. both have occurred simultaneously). Then the degree of σ_2 being controllable is given by the minimum of the possibility of fuzzy event $s'\sigma_2$ belonging to $\bar{k}$ and the physical possibility of $s\sigma_2$.

Assume $s = \sigma_1$, $s' = \sigma_2$. The fuzzy string $s\sigma_2$ (which is $\sigma_1\sigma_2$) is available in $\bar{k}'$ but not in $\bar{k}$. However, the fuzzy string $s'\sigma_2$ (which is $\sigma_2\sigma_2$) is available in $\bar{k}$. In this case $\Sigma_c(\sigma_2) = min\{0.8,\ 1\} = 0.8$.

Case III: $\Sigma_c(\sigma_2) = 0$ when $\sigma_2 \in \Sigma \setminus \Sigma_c$

In this case σ_2 is a not a controllable event, thus the operation returns zero.

3.2.3 Supervisor definition

Supervisory control is defined in order to simultaneously enable all feasible fuzzy uncontrollable events and fuzzy controllable events that extend s inside of $\bar{k}'$, with varied possibilities. The fuzzy controllable events, which are enabled by the supervisor, must comply with all other feasible fuzzy uncontrollable events in order to achieve the safe navigation of the robot. To make this possible we define a new set of fuzzy events T_s, where $T_s(\sigma)$ gives the degree of conformity of a fuzzy controllable event σ with all other feasible fuzzy uncontrollable events, given fuzzy string s has occurred in the system.

The degree of a fuzzy event $\sigma(\in \Sigma)$ being enabled by the supervisor S of the FDES G, after the fuzzy string $s(\in \Sigma^*)$ has occurred in the system, is defined as follows.

$$S_s(\sigma) = \begin{cases} \Sigma_{uc}(\sigma)\tilde{\cap}L_G(s\sigma), & \text{if } \sigma \in \Sigma_{uc} \\ \Sigma_c(\sigma)\tilde{\cap}T_s(\sigma), & \text{if } \sigma \in \Sigma_c \end{cases} \tag{3.22}$$

This means that $S_s(\sigma)$ is equal to the degree of σ being uncontrollable together with $s\sigma$ is physically possible, if σ is a fuzzy uncontrollable event. Otherwise, if σ is a fuzzy controllable event then $S_s(\sigma)$ is equal to the degree of σ being controllable together with the conformity of σ with all feasible fuzzy uncontrollable events.

Example 3.1: Consider the problem definition with $\Sigma_{uc}(\sigma_1) = 0.7$ and $T_s(\sigma_2) = 0.3$ and assume these values remain unchanged during the operation. The following supervisor, which is defined as in (3.22) can achieve $\bar{k}'$.

For $\sigma_1 : S_\varepsilon(\sigma_1) = min\{0.7,\ 0.9\} = 0.7$

Using the definition of $L_{S/G}$ in (3.5) $\rightarrow\ \bar{k}'(\sigma_1) = min\{1,\ 0.7,\ 0.9\} = 0.7$

For $\sigma_2 : \Sigma_c(\sigma_2) = min\{0.8,\ 1\} = 0.8$

$S_\varepsilon(\sigma_2) = min\{0.8,\ 0.3\} = 0.3 \rightarrow\ \bar{k}'(\sigma_2) = min\{1,\ 0.3,\ 0.8\} = 0.3$

For $\sigma_1\sigma_1 : S_{\sigma_1}(\sigma_1) = min\{0.7,\ 0.9\} = 0.7 \rightarrow\ \bar{k}'(\sigma_1\sigma_1) = min\{0.7,\ 0.7,\ 0.9\} = 0.7$

For $\sigma_2\sigma_1$: $S_{\sigma_2}(\sigma_1) = min\{0.7,\ 0.8\} = 0.7 \rightarrow\ \bar{k}'(\sigma_2\sigma_1) = min\{0.3,\ 0.7,\ 0.8\} = 0.3$

For $\sigma_1\sigma_2$: $\Sigma_c(\sigma_2) = min\{0.8,\ 1\} = 0.8$

$S_{\sigma_1}(\sigma_2) = min\{0.8,\ 0.3\} = 0.3 \rightarrow\ \bar{k}'(\sigma_1\sigma_2) = min\{0.7,\ 0.3,\ 0.8\} = 0.3$

For $\sigma_2\sigma_2$: $\Sigma_c(\sigma_2) = min\{0.8,\ 1\} = 0.8$

$S_{\sigma_2}(\sigma_2) = min\{0.8,\ 0.3\} = 0.3 \rightarrow\ \bar{k}'(\sigma_2\sigma_2) = min\{0.3,\ 0.3,\ 0.8\} = 0.3$

$$\Rightarrow \frac{1}{\varepsilon} + \frac{0.7}{\sigma_1} + \frac{0.3}{\sigma_2} + \frac{0.7}{\sigma_1\sigma_1} + \frac{0.3}{\sigma_1\sigma_2} + \frac{0.3}{\sigma_2\sigma_1} + \frac{0.3}{\sigma_2\sigma_2}$$

The fuzzy events enabled by the supervisor defined in (3.22) are then used for fuzzy state transition. These fuzzy states represent the weights associated in corresponding behaviors. Hence, this shows a command fusion type behavior coordination scheme.

3.3 Application to single robot control

It should be noted that providing a fuzzy language specification initially is impossible when a robot is navigating in an unmodeled environment where reactive behavior-based control is necessary. Moreover, the language specification is inferred through fuzzy rule bases according to the robot's perception at each decision cycle.

3.3.1 Perception to action mapping

Figure 3.2 describes the perception to action mapping of a physical agent. The **Perception** component is comprised of m sensors used to sense the environment. These sensors extract the information about the subgoals and obstacles. The FDES-based **Coordination** component receives this perceptual information and performs the proposed supervisory control. With fuzzy state transition, the n fuzzy states are weighted accordingly. The weight of a fuzzy state represents the activation level of the corresponding behavior. The **Action** component receives the recommendations of each behavior for the k actuators and implements the vector addition of those recommendations. This generates the final control action for each actuator.

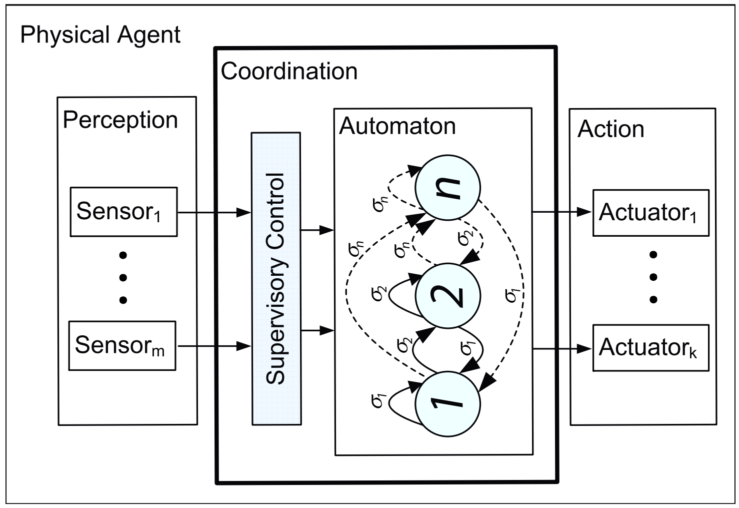


Figure 3.2: The information flow diagram showing perception to action mapping

3.3.2 Behavior coordination of a single robot

Assume a scenario where a mobile robot is moving in an unmodeled environment. The robot has five behaviors: two deliberative behaviors namely, "Follow Route" (FR) and "Go to Target" (GT) and three reactive behaviors namely, "Avoid Obstacle" (AO), "Follow Wall" (FW) and "Avoid Dead ends" (AD). This FDES is modeled by a fuzzy automaton $G = (Q, \Sigma, \delta, q_0)$ as shown in figure 3.3.

Also, G holds the following.

$Q = \{FR, GT, AO, FW, AD\},\ |Q| = 5,$

$\Sigma = \Sigma_c \cup \Sigma_{uc}$, where $\Sigma_c = \{\sigma_{FR}, \sigma_{GT}\}$ and $\Sigma_{uc} = \{\sigma_{AO}, \sigma_{FW}, \sigma_{AD}\}$.

δ is as shown in figure 3.3 and q_0 is the initial fuzzy states representation.

Note that each fuzzy state of the automaton represents the activation level of the corresponding behavior.

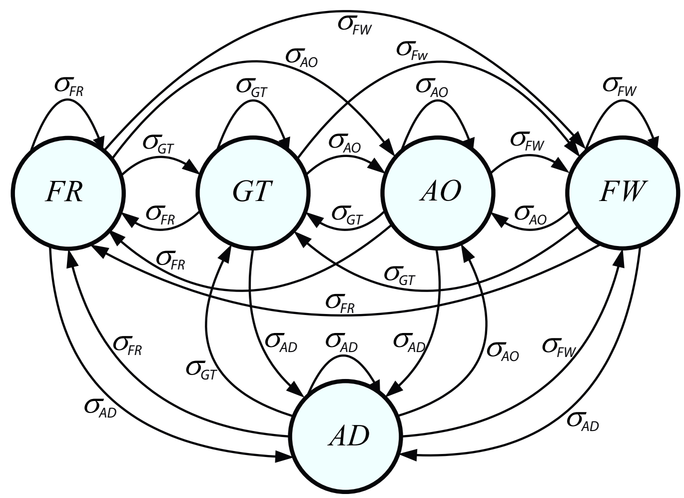


Figure 3.3: Fuzzy automaton G modeling the behaviors of a mobile robot

3.3.3 Behavior modeling

A fuzzy rule base is constructed for each fuzzy event to calculate their degrees of being controllable or uncontrollable. Symmetric triangular membership functions are used to compute the antecedents and consequents of all fuzzy rule bases. To finally obtain a crisp value, the Min-Max-Centroid defuzzification technique is used (i.e., implication: Min, aggregation: Max and defuzzification: Centroid) [79]. Note that for simplicity, from here on, this chapter assumes any fuzzy event σ extending a fuzzy string s is completely physically possible for any time (i.e., $L_G(s\sigma) = 1.0$).

Follow Route behavior (FR): This behavior is used to navigate the robot through waypoints. The fuzzy controllable event, which activates this behavior is σ_{FR}. Figure 3.4 shows the membership functions used to calculate $\Sigma_c(\sigma_{FR})$. Here, the "FR Divergence" is calculated by the absolute angle difference between the current robot heading θ and the direction suggested by the "Follow Route" behavior $\measuredangle\overrightarrow{F.R.}$ (i.e., $|\theta - \measuredangle\overrightarrow{F.R.}|$). Table 3.1 describes the fuzzy rule base designed to calculate the degree of σ_{FR} being controllable.

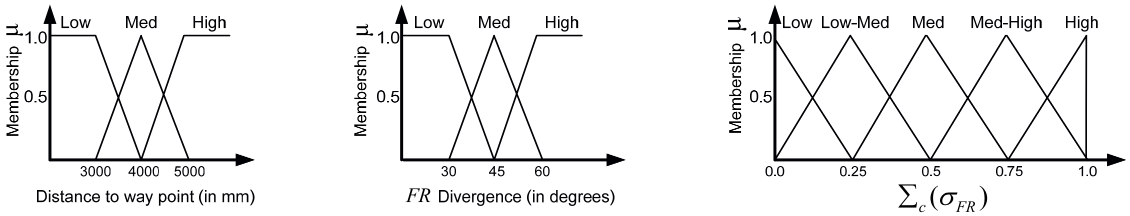


Figure 3.4: Membership functions used to calculate $\Sigma_c(\sigma_{FR})$

Table 3.1: Fuzzy rule base for $\Sigma_c(\sigma_{FR})$

Distance to nearest waypoint	FR Divergence $\lvert\theta - \measuredangle\overrightarrow{F.R.}\rvert$	$\Sigma_c(\sigma_{FR})$
Low	Low	Low
	Medium	Low-Medium
	High	Medium
Medium	Low	Low-Medium
	Medium	Medium
	High	Medium-High
High	Low	Medium
	Medium	Medium-High
	High	High

Table 3.2: Fuzzy rule base for $\Sigma_c(\sigma_{GT})$

Distance to 2nd nearest waypoint	GT Divergence $\lvert\theta - \measuredangle\overrightarrow{G.T.}\rvert$	$\Sigma_c(\sigma_{GT})$
Low	Low	High
	Medium	Medium-High
	High	Medium
Medium	Low	Medium-High
	Medium	Medium
	High	low-Medium
High	Low	Medium
	Medium	Low-Medium
	High	Low

Go to Target behavior (GT): This is used for path optimization. This aims for the next nearest waypoint from the current robot orientation. Table 3.2 describes the fuzzy rule base designed to calculate the degree of the corresponding fuzzy event σ_{GT} being controllable. Here, the "GT Divergence" is calculated by the absolute angle difference between the current robot heading and the direction suggested by the "Go to Target" behavior (i.e., $\lvert\theta - \measuredangle\overrightarrow{G.T.}\rvert$).

Avoid Obstacle behavior (AO): This behavior slides the robot to a direction, which is perpendicular to the line connecting both robot and its nearest obstacle, when the distance from the robot to the obstacle is less than its limit. Otherwise, it becomes inactive. Table 3.3 describes the fuzzy rule base designed to calculate the degree of corresponding fuzzy event σ_{AO} being uncontrollable. Here, the "AO Divergence" is calculated by the absolute angle difference between the current robot heading and the direction suggested by the "Avoid Obstacle" behavior (i.e., $\lvert\theta - \measuredangle\overrightarrow{A.O.}\rvert$).

Table 3.3: Fuzzy rule base for $\Sigma_{uc}(\sigma_{AO})$

Distance to nearest obstacle	AO Divergence $\lvert\theta - \measuredangle\overrightarrow{A.O.}\rvert$	$\Sigma_{uc}(\sigma_{AO})$
Low	Low	Medium
	Medium	Medium-High
	High	High
Medium	Low	Low-Medium
	Medium	Medium
	High	Medium-High
High	Low	Low
	Medium	Low-Medium
	High	Medium

Table 3.4: Fuzzy rule base for $\Sigma_{uc}(\sigma_{FW})$

Distance to nearest obstacle	FW Divergence $\lvert\theta - \measuredangle\overrightarrow{W.F.}\rvert$	$\Sigma_{uc}(\sigma_{FW})$
Low	Low	Medium
	Medium	Medium-High
	High	High
Medium	Low	Low-Medium
	Medium	Medium
	High	Medium-High
High	Low	Low
	Medium	Low-Medium
	High	Medium

Follow Wall behavior (FW): This behavior forces the robot to keep a minimum distance from the obstacles. The direction of this behavior is opposite to the nearest obstacle direction. If the distance is greater than its minimum the behavior becomes inactive. Table 3.4 describes the fuzzy rule base designed to calculate the degree of corresponding fuzzy event σ_{FW} being uncontrollable. Here, the "FW Divergence" is calculated by the absolute angle difference between the current robot heading and the direction suggested by the "Follow Wall" behavior (i.e., $\lvert\theta - \measuredangle\overrightarrow{F.W.}\rvert$).

Avoid Dead ends behavior (AD): This is designed in order to carefully avoid dead end situations. When a dead end is identified on the robot's path, a memory flag is made "High" (i.e. = 1). Then a virtual object is placed for the robot to follow until the robot has moved away from the dead end [93]. Once the dead end is cleared the flag is made "Low" (i.e. = 0). The direction of this behavior is towards the wall. Table 3.5 describes the fuzzy rule base designed to calculate the degree of the corresponding fuzzy event σ_{AD} being uncontrollable. Here, SM represents the value of the memory flag (i.e., 1 or 0).

Table 3.5: Fuzzy rule base for $\Sigma_{uc}(\sigma_{AD})$

SM	$\Sigma_{uc}(\sigma_{FW})$	$\Sigma_{uc}(\sigma_{AD})$
High	Low	High
	Medium	Medium
	High	Low
Low	Low	Low
	Medium	Low
	High	Low

Table 3.6: Fuzzy rule base for $T_s(\sigma_{FR})$

SM	$\Sigma_{uc}(\sigma_{FW})$	$\Sigma_{uc}(\sigma_{AO})$	$T_s(\sigma_{FR})$
Low	Low	Low	High
		Medium	Medium
		High	Low
	Medium	Low	Medium
		Medium	Medium
		High	Low
	High	Low	Low
		Medium	Low
		High	Low
High	Low	Low	Low
		Medium	Low
		High	Low
	Medium	Low	Low
		Medium	Low
		High	Low
	High	Low	Low
		Medium	Low
		High	Low

3.3.4 Supervisor Synthesis

According to (3.22), followings are calculated.

$S_s(\sigma_{FR}) = \Sigma_c(\sigma_{FR})\tilde{\cap}T_s(\sigma_{FR})$, $S_s(\sigma_{GT}) = \Sigma_c(\sigma_{GT})\tilde{\cap}T_s(\sigma_{GT})$

$S_s(\sigma_{AO}) = \Sigma_{uc}(AO)$, $S_s(\sigma_{FW}) = \Sigma_{uc}(FW)$ and $S_s(\sigma_{AD}) = \Sigma_{uc}(AD)$

For example, table 3.6 shows the fuzzy rule base designed to calculate the degree of conformity of σ_{FR} with available fuzzy uncontrollable events. Figure 3.5 shows the membership functions used to calculate $T_s(\sigma_{FR})$.

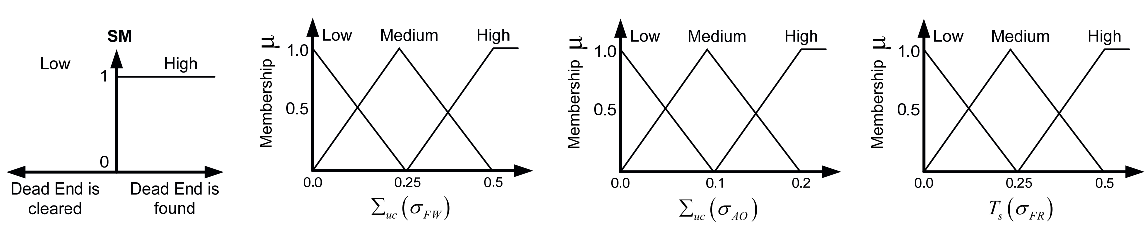


Figure 3.5: Membership functions used to calculate $T_s(\sigma_{FR})$

.

Example 3.2: Assume $\Sigma_c(\sigma_{FR}) = 0.8$, $T_s(\sigma_{FR}) = 0.3$, and SM is "Low". Then $S_s(\sigma_{FR})$ can be calculated according to (3.22) as:

$S_s(\sigma_{FR}) = T_s(\sigma_{FR}) \tilde{\cap} \Sigma_c(\sigma_{FR}) = 0.3 \times 0.8 = 0.24$. (Here, $\tilde{\cap}$ is modeled by the product of two.) Similarly, $S_s(\sigma_{GT})$ also can be calculated.

Assume the fuzzy state vector at time t is denoted by Q_t and the matrix, which represents all fuzzy events at time $t+1$ is denoted by M_{t+1}. The matrix M_{t+1} can be composed as shown below.

$$M_{t+1} = \begin{bmatrix} S_s(\sigma_{FR}) & S_s(\sigma_{GT}) & S_s(\sigma_{AO}) & S_s(\sigma_{FW}) & S_s(\sigma_{AD}) \\ S_s(\sigma_{FR}) & S_s(\sigma_{GT}) & S_s(\sigma_{AO}) & S_s(\sigma_{FW}) & S_s(\sigma_{AD}) \\ S_s(\sigma_{FR}) & S_s(\sigma_{GT}) & S_s(\sigma_{AO}) & S_s(\sigma_{FW}) & S_s(\sigma_{AD}) \\ S_s(\sigma_{FR}) & S_s(\sigma_{GT}) & S_s(\sigma_{AO}) & S_s(\sigma_{FW}) & S_s(\sigma_{AD}) \\ S_s(\sigma_{FR}) & S_s(\sigma_{GT}) & S_s(\sigma_{AO}) & S_s(\sigma_{FW}) & S_s(\sigma_{AD}) \end{bmatrix} \tag{3.23}$$

The following steps are performed recursively for time $t \geq 0$.

1. The next fuzzy state vector Q_{t+1} is calculated as:

   $Q_t \circ M_{t+1} \rightarrow Q_{t+1}$. Here, "$\circ$" represents the Max-Product operation.

2. The sum of the possibility distribution of fuzzy states is then normalized as:

   $\rightarrow \ \forall q_{i,t} \in Q_t, \quad \sum_{i=1}^{|Q|} q_{i,t} = 1$. Here, $q_{i,t}$ is the i^{th} fuzzy state of Q_t.

3. The final coordinated action $\underline{\vec{A}_{t+1}}$ is calculated as:

   $$\underline{\vec{A}_{t+1}} = \sum_{i=1}^{|Q|} q_{i,t+1} \times \vec{\mathbf{a}}_{i,t+1} \tag{3.24}$$

   where $\vec{\mathbf{a}}_{i,t+1}$ is the unit vector representing i^{th} behavior and $q_{i,t+1}$ is its activation level read directly from Q_{t+1}.

Measure of fuzzy state-based controllability

Let $W_{5\times5}$ be the consistency matrix as in [4] where the element $w_{i,j}$ represents the measure of inconsistency between fuzzy states i and j (0 being most consistent and 1 being most inconsistent).

The following properties can be identified from the behaviors.

1. FR and GT are consistent with each other with a higher degree as these behaviors represent goal directed robot motion. Therefore, the assigned measure of inconsistency between these two behaviors is 0.5 (i.e. $w_{1,2} = w_{2,1} = 0.5$).

2. AO and FW are consistent with each other with a moderate degree as these behaviors represent safety, but the directions suggested by these two may be incompatible as the angle difference between these two behaviors is $\frac{\pi}{2}$. This leads to assign $w_{3,4} = w_{4,3} = 0.3$.

3. AD is highly inconsistent with deliberative behaviors. This leads to assign $w_{1,5} = w_{5,1} = 1.0$ and $w_{2,5} = w_{5,2} = 1.0$. Also, this behavior directs the robot to a direction opposite to the one which is suggested by FW. This inconsistency leads to assign $w_{4,5} = w_{5,4} = 1.0$.

4. Deliberative and other reactive behaviors mentioned above are consistent with each other to a lesser degree, as these together represent safe operation and goal directed robot motion, but the direction suggested by these may be fairly contradictory. This leads to assign $w_{1,3} = w_{3,1} = 0.9$ and $w_{1,4} = w_{4,1} = 0.9$. Also $w_{2,3} = w_{3,2} = 0.8$ and $w_{2,4} = w_{4,2} = 0.8$.

5. Directions suggested by AO and AD are inconsistent as they are apart from $\frac{\pi}{2}$. This leads to assign $w_{3,5} = w_{5,3} = 0.7$.

Using the above information W can be constructed.

$$W = \begin{bmatrix} 0.0 & 0.5 & 0.9 & 0.9 & 1.0 \\ 0.5 & 0.0 & 0.8 & 0.8 & 1.0 \\ 0.9 & 0.8 & 0.0 & 0.3 & 0.7 \\ 0.9 & 0.8 & 0.3 & 0.0 & 1.0 \\ 1.0 & 1.0 & 0.7 & 1.0 & 0.0 \end{bmatrix}$$

Note that $q_t \cdot W \cdot q_t^T$ represents the degree of consistency between all fuzzy states as in [4] (0 being most consistent and 1 being least). The measure given by $C_t = (1 - q_t \cdot W \cdot q_t^T)$ represents the consistency degree of each fuzzy state and it is further identified here as fuzzy state-based controllability of the system at time t.

The computational complexity of the approach

A behavior-based system with n behaviors can be effectively modeled by a fuzzy automaton having n fuzzy states and n^2 fuzzy event transitions. The dimensions of the resulting fuzzy state matrix and event matrix will be $(1 \times n)$ and $(n \times n)$, respectively. Hence, the computational complexity of the proposed approach for control computation of a single robot is $\mathcal{O}(n^2)$, neglecting the complexity of the defuzzification step of fuzzy rule bases.

3.4 Implementation of the proposed approach

In addition to the behavior coordination requirements set forth in section IV, certain performance requirements for mobile robot navigation can be identified, such as:

- The robot should navigate to the final goal with more accuracy.
- The total navigation should be completed within a reasonable time.
- The robot should not collide with an obstacle.

- The robot should perform a deadlock and livelock free navigation.

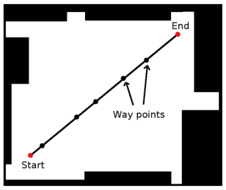
The proposed FDES-based supervisory control approach for behavior coordination is implemented both in simulation and on a real-time physical robot, and is verified as to whether it fulfills the above performance requirements.

3.4.1 Simulation results

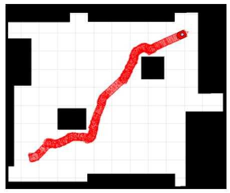
Mobile robot simulations were carried out using MobileSim Version 0.4.0 provided by MobileRobots Inc, with a Pioneer 3 DX robot. A 10m × 12m simulated environment space was used and start and end points were identified. The waypoints were given manually and *dead reckoning* [94] was used to localize the robot since the environment used was relatively small. Several unmodeled obstacles and dead ends were used to examine the performance of the proposed approach. The distances to the obstacles were obtained by using the embedded sonar ring. For testing several simulated environments were considered. Each decision cycle consists of a rotation and a translation command. The rotation was used for angular correction and the translation was used to move the robot to the final decided direction. Robot translation speed was fixed at 50mm per second and the translation cycle was 50ms. Its rotation speed was proportional to the desired heading.

Three behavior coordination schemes were compared with the proposed FDES-based approach to evaluate the performances. These are namely, the unmodulated coordination, the potential field method [95, 96] and a crisp DES-based approach, which represents the behavior arbitration. Throughout the experiments we assumed complete observability for unmodulated, potential field and DES-based coordination schemes ($\Sigma_o(\sigma) = 1$). For proposed FDES-based coordination, partial observability of events was introduced. ($\Sigma_o(\sigma) = 0.8$ represents 80% accuracy of associated sensors). Several simulation tests were carried out for validating the proposed approach.

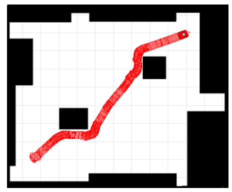
Test I: In an environment with two unmodeled obstacles



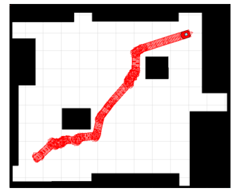
(a) Original map



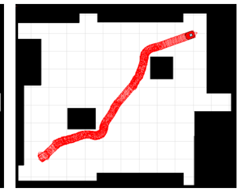
(b) Unmodulated



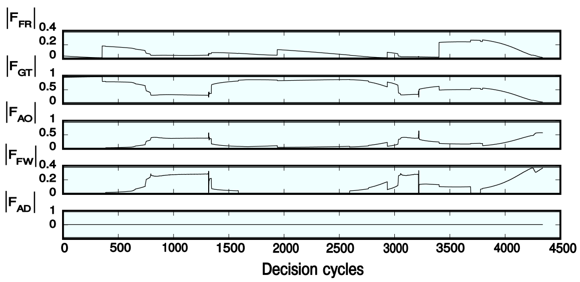
(c) Potential field



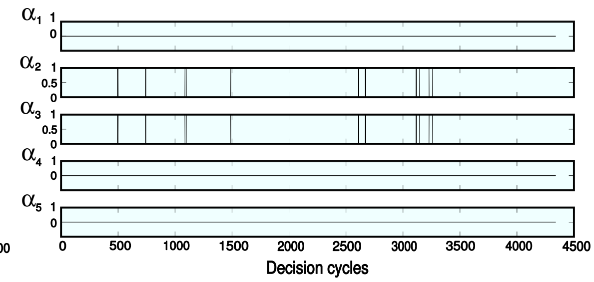
(d) DES-based



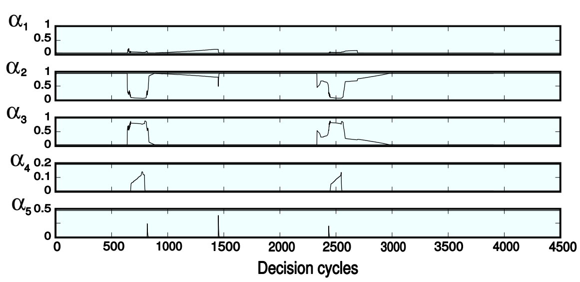
(e) Proposed



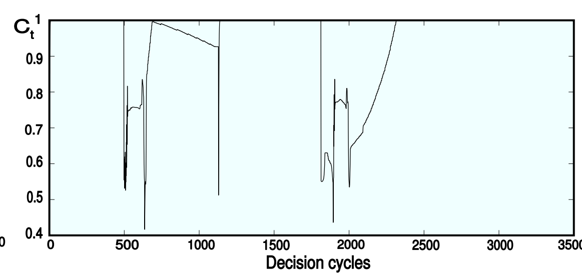
(f) Potential field force magnitudes



(g) Behavior weights of DES-based approach



(h) Behavior weights of proposed approach

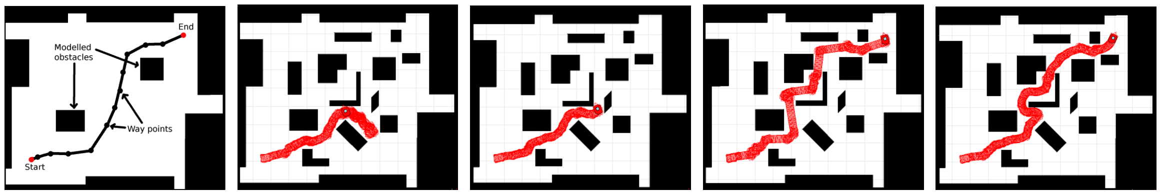


(i) Fuzzy state-based controllability

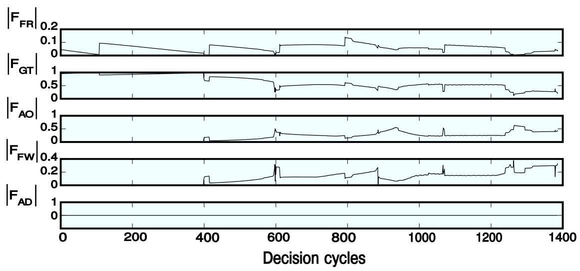
Figure 3.6: Plots related to Test I

Figures 3.6(a)-(e) show the navigation results of a mobile robot using different coordination schemes. The environment has two unmodeled obstacles. In figure 3.6(f), $|F_{FR}|$, $|F_{GT}|$, $|F_{AO}|$, $|F_{FW}|$ and $|F_{AD}|$ represent the normalized magnitudes of virtual attractive and repulsive forces given by FR, GT, AO, FW and AD respectively. In figure 3.6(g) and 3.6(h) $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, α_5 depict the evolution of FR, GT, AO, FW and AD behaviors respectively in crisp DES-based and proposed FDES-based approaches. Figure 3.6(i) shows the evolution of the fuzzy state-based controllability measure of the proposed approach.

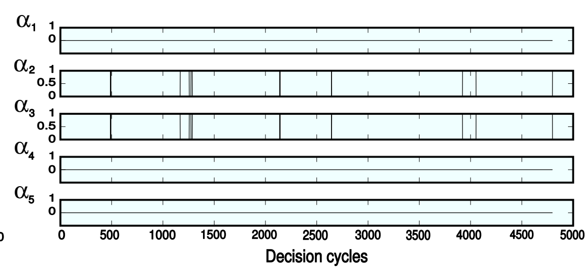
Test II: In a cluttered environment with unmodeled obstacles



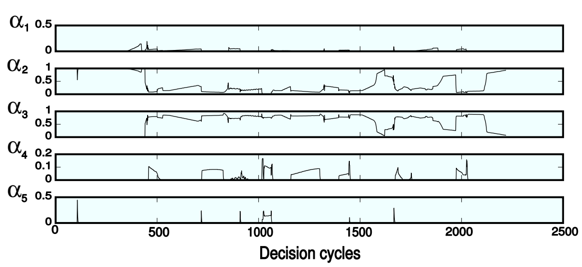
(a) Original map (b) Unmodulated (c) Potential field (d) DES-based (e) Proposed



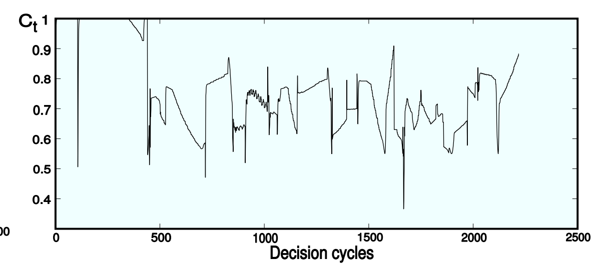
(f) Potential field force magnitudes



(g) Behavior weights of DES-based approach



(h) Behavior weights of proposed approach

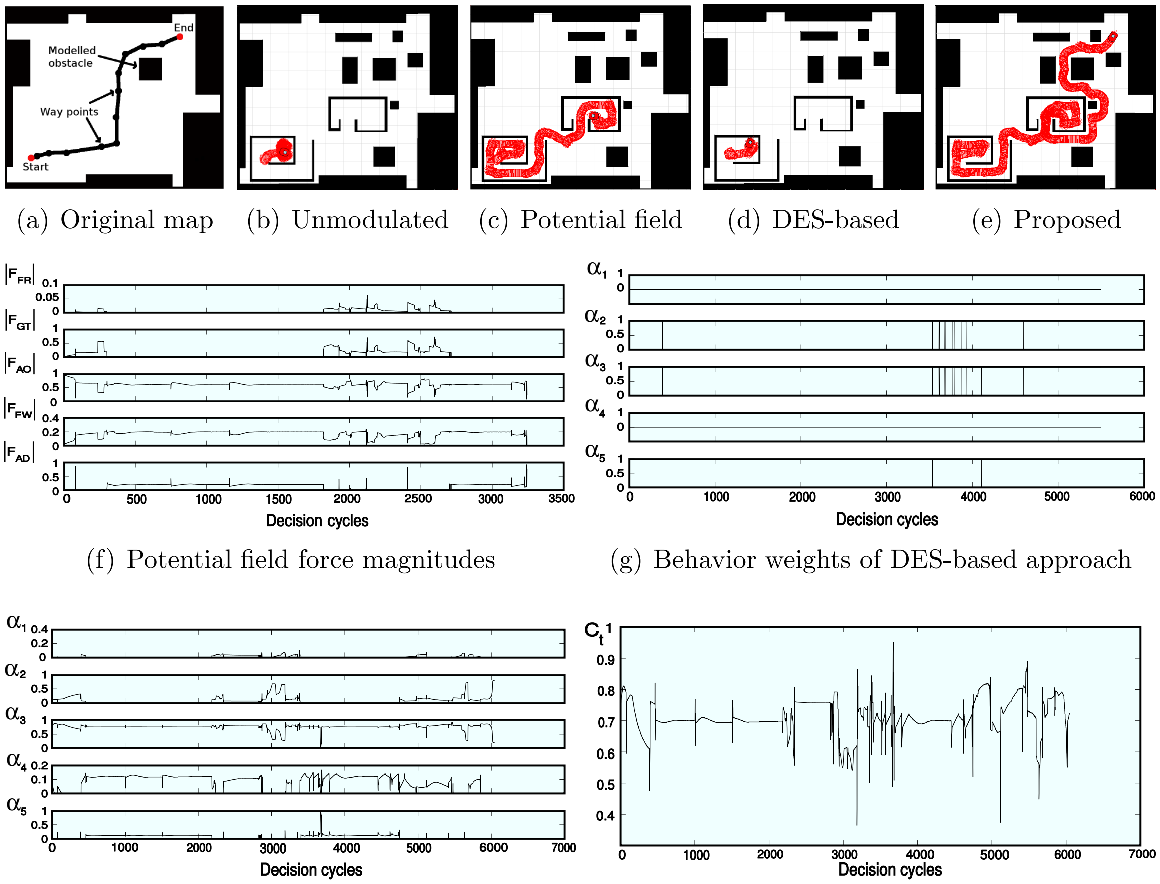


(i) Fuzzy state-based controllability

Figure 3.7: Plots related to Test II

Figures 3.7(a)-(e) show the navigation results of the robot moving in a cluttered environment filled with unmodeled obstacles, produced by using different coordination schemes. The navigation by unmodulated coordination is unsuccessful as the robot collided with an obstacle. Due to its limitations [96], potential field approach is also unsuccessful in this environment. Figure 3.7(f) shows the normalized force magnitudes of potential field approach. Figures 3.7(g) and 3.7(h) show behavior weights of crisp DES and proposed FDES-based behavior coordination schemes. Figure 3.7(i) shows the fuzzy state-based controllability measure of the proposed approach.

Test III: With dead ends and unmodeled obstacles



(a) Original map (b) Unmodulated (c) Potential field (d) DES-based (e) Proposed

(f) Potential field force magnitudes (g) Behavior weights of DES-based approach

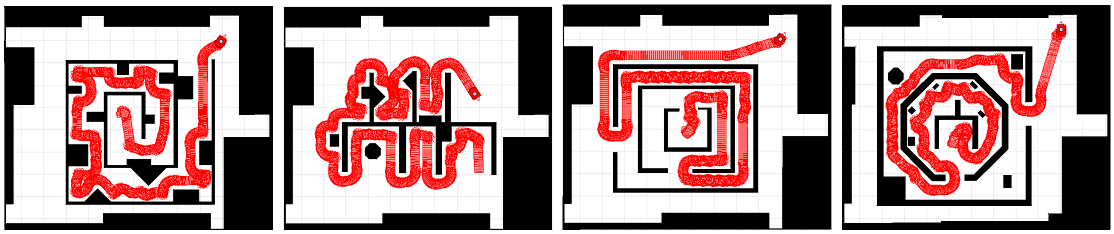
(h) Behavior weights of proposed approach (i) Fuzzy state-based controllability

Figure 3.8: Plots related to Test III

Figures 3.8(a)-(e) show the navigation results in an environment with dead ends. The unmodulated coordination is unsuccessful. Also, both potential field and DES-based approaches introduce the chattering effect and fail to navigate the robot successfully. The proposed FDES-based approach is able to effectively move the robot while avoiding dead ends. Figure 3.8(f) shows the normalized force magnitudes of potential field approach. Figures 3.8(g) and 3.8(h) show behavior weights of crisp DES and proposed FDES-based behavior coordination schemes. Figure 3.8(i) shows the fuzzy state-based controllability measure of the proposed approach.

Test IV: Environments with various unmodeled partitions

Performance of the proposed approach was examined in more complex environments. Figure 3.9 depicts the traveled path of the robot with these environments.



(a) In a spiral like environment (b) In a wall like environment (c) In a maze like environment (d) In a compound environment

Figure 3.9: Navigation in complex environments using proposed approach

3.4.2 Performance evaluation

The following are identified as the metrics for measuring the performance of mobile robot navigational tasks [97, 98].

1. **Total time to goal reach, T_{tot}:** The total execution time to approach the goal is measured. For high performance, it is desirable to have a low execution time.

2. **Path length, L_{tot}:** The total length of the trajectory covered from start to end is measured. Having shorter length is desirable for better performance. The total length from start $(x_0, f(x_0))$ to end $(x_n, f(x_n))$ can be calculated as follows.

$$L_{tot} = \int_{x_0}^{x_n} \left(1 + (f'(x))^2\right)^{\frac{1}{2}} dx$$

where the trajectory is given by $y = f(x)$ in the X-Y plane and $f'(x)$ is the derivative of $f(x)$ with respect to x.

3. **Bending energy, B_E:** The bending energy is a measure of the energy requirement of the movement. It is also useful to evaluate the smoothness of the robot trajectory. Let the curvature ($k(x)$) across the trajectory be defined as follows.

$$k(x) = \frac{f''(x)}{\left(1 + (f'(x))^2\right)^{\frac{3}{2}}}$$

   Then, the average bending energy can be calculated as follows.

$$B_E = \frac{1}{x_n - x_0} \int_{x_0}^{x_n} k^2(x)dx$$

   Having low bending energy is desirable as it increases the smoothness and decreases the energy requirement.

4. **Mean distance to obstacles, D^{Obs}_{Mean}:** The average of the minimum distances between robot (sensor) and the obstacles measured in each execution cycle through entire navigation. A higher value ensures secure navigation.

5. **Minimum distance to obstacles, D^{Obs}_{Min}:** The minimum distance between robot (sensor) and obstacles through the entire navigation. This indicates the risk taken through the entire movement.

6. **Number of collisions, N_{col}:** A collision free operation indicates a safe navigation of the robot.

Based on the above metrics a performance evaluation of the presented behavior coordination mechanisms was performed for the three test environments. The results are shown in Table 3.7.

Table 3.7: Performance evaluation for the three test environments

Test	Coordination scheme	T_{tot} (s)	L_{tot} (m)	B_E	D^{Obs}_{Mean} (m)	D^{Obs}_{Min} (m)	N_{col}
Test I	Unmodulated	244	1.180×10^1	2.453×10^{-2}	4.18×10^{-1}	1.53×10^{-1}	0
	Potential field	238	1.162×10^1	1.209×10^{-2}	5.03×10^{-1}	1.18×10^{-1}	0
	DES-based	242	1.186×10^1	3.073×10^{-1}	4.75×10^{-1}	2.77×10^{-1}	0
	FDES-based	239	1.165×10^1	9.389×10^{-4}	5.04×10^{-1}	2.49×10^{-1}	0
Test II	Unmodulated	Fail	Fail	Fail	Fail	Fail	1
	Potential field	Fail	Fail	Fail	Fail	Fail	1
	DES-based	284	1.351×10^1	5.172×10^0	2.74×10^{-1}	1.01×10^{-1}	0
	FDES-based	282	1.345×10^1	1.791×10^{-1}	3.11×10^{-1}	1.00×10^{-1}	0
Test III	Unmodulated	Fail	Fail	Fail	Fail	Fail	0
	Potential field	Fail	Fail	Fail	Fail	Fail	1
	DES-based	Fail	Fail	Fail	Fail	Fail	1
	FDES-based	630	3.024×10^1	7.482×10^0	2.66×10^{-1}	1.05×10^{-1}	0

The results shown in Table 3.7 indicate that the proposed FDES-based approach was the most successful of all by having less total times and less energy requirements. Moreover, it provided smooth navigations in all cases due to the the command fusion type behavior coordination. The potential field and crisp DES-based approaches failed in the environment having dead ends. Furthermore, the crisp DES-based approach introduced chattering effects in all navigations due to the behavior arbitration and it also had higher energy requirements.

3.4.3 Real-time implementation

The proposed method was implemented in real-time on a physical robot (Pioneer 3 AT platform) with 0.8 partial observability associated with each fuzzy event. Here also, dead reckoning was used to localize the robot.

Test I: The environment with unmodeled obstacles

Figure 3.10 depicts the robot's collision free navigation through unmodeled obstacles while maintaining a better trajectory and moving a shorter distance.

(a) Approaching a subgoal (b) Avoiding the obstacle at front (c) Following the wall (d) Turning at the obstacle (e) Reaching the goal

Figure 3.10: Performance in the real-world with unmodeled obstacles

Test II: The environment with a dead end

Figure 3.11 shows the robot's navigation when a dead end is placed on the waypoints in addition to the unmodeled obstacles. The dead end is avoided smoothly and the final target is achieved.

(a) Approaching a subgoal (b) Turning at the dead end (c) Completing the turn (d) Following the wall (e) Turning at the dead end

(f) Following the wall (g) Completing the turn (h) Following the wall (i) Avoiding the obstacle (j) Reaching the goal

Figure 3.11: Performance in the real-world with a dead end

The accompanying movies depict these implementations. Movie 3.1 shows the navigation in an environment with unmodeled obstacles only (test I). Movie 3.2 shows the navigation in an environment with unmodeled obstacles and a dead end (test II). Some deflections can be seen as the failures of sonar sensors near smooth surfaces.

3.5 Summary

This chapter presented a framework for single robot navigational tasks using supervisory control of FDES. In this approach, a robot is modeled by a fuzzy automaton and its behaviors are represented as fuzzy states. These behaviors are activated through fuzzy event transitions, which are calculated using the sensory information. The fuzzy events are triggered with different weightings, which makes the robot to operate in a command fusion type behavior coordination scheme.

Fuzzy events are incorporated with partial observability to represent the imprecision of the sensors. For analyzing the system in the control theoretical aspect, the fuzzy state-based controllability measure is introduced. In simulation several navigation scenarios are used to evaluate the performances of the proposed method. It is observed that the proposed approach is able to provide successful goal-oriented navigations with smooth trajectories. The approach is also implemented in a physical robot navigating in an environment with unmodeled obstacles and a dead end. The real-time results proved the validity of the proposed method.

The number of behaviors of the system can be increased by adding more fuzzy states to the automaton. Moreover, the approach inherits a formal method for mobile robot behavior coordination from the well-established Ramadge-Wonham framework. The use of formal methods eases the analysis of controllability, observability and stability of a system.

The proposed approach can be also employed in a mobile robot navigating in a slow-moving dynamic environment. However, for better results more behaviors need to be incorporated.

Chapter 4

Decentralized control of FDES

In this chapter, a general architecture for decentralized supervision of FDES is established. Firstly, two different types of decentralized supervisory control architectures of FDES are presented, which fuse the locally-enabled degrees of fuzzy events using the fuzzy-intersection operator and the fuzzy-union operator respectively. Both of these architectures possess limitations in information association. Secondly, to overcome the above drawbacks a general architecture for decentralized supervisory control of FDES is introduced, in which the decisions of local supervisors are fused by using both fuzzy-union and fuzzy-intersection operators. The proposed general architecture is then implemented to control a tightly-coupled multi robot object manipulation task both in simulation and in real-time. A performance evaluation is also performed to quantitatively estimate the validity of the proposed architecture compared to FDES-based centralized and crisp DES-based decentralized approaches.

4.1 The decentralized control architectures of FDES

In addition to the preliminaries set forth in section 3.1, from here onwards we do not distinguish fuzzy controllable events and fuzzy uncontrollable events separately.

Each fuzzy event is associated with a degree of controllability as in [13, 23], which is a more general setting for representing fuzzy events.

$$\Sigma_c(\sigma) + \Sigma_{uc}(\sigma) = 1 \tag{4.1}$$

4.1.1 Conjunctive and Permissive (C&P) architecture

Note that the decentralized supervisory control theories of FDES presented in [12–14] are based on the C&P architecture. However, this reworking aims to offer more straightforward organization of co-observability which facilitates the derivation of the general decentralized supervisory control architecture of FDES.

Figure 4.1 shows the C&P decentralized supervisory control architecture of FDES, which is adapted from its crisp DES version in [15].

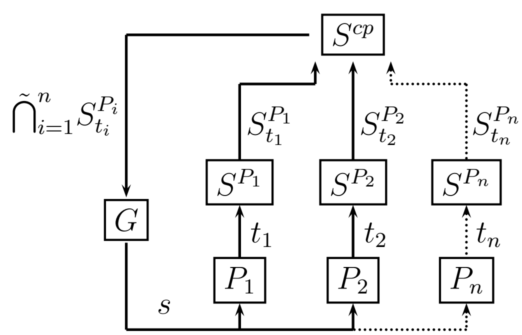


Figure 4.1: C&P decentralized supervisory control architecture of FDES

Each local supervisor S^{P_i}, for $i \in \{1, ..., n\}$, has a different projection of the fuzzy string s, which occurred in the FDES G, at the time of feedback (i.e. $P_i(s) = t_i$). The C&P decentralized supervisor of FDES S^{cp} fuses the degrees of fuzzy events, which are recommended by S^{P_i} using the fuzzy-intersection operator. The final decision of this decentralized supervisor for the system can be computed as follows.

$$S_s^{cp} = \tilde{\bigcap}_{i=1}^{n} S_{t_i}^{P_i} \to \min\{S_{t_1}^{P_1}, ..., S_{t_n}^{P_n}\} \tag{4.2}$$

Under this architecture the default action of S^{P_i} with insufficient information is to completely enable the fuzzy events. (i.e. When S^{P_i} has insufficient information about σ and fuzzy string s has occurred, then $S_{t_i}^{P_i}(\sigma) = 1$; hence, enablement as the default).

Assume Σ_i as the fuzzy event set of S^{P_i} and $\Sigma_{i,c}, \Sigma_{i,o}$ are its fuzzy controllable and observable event sets. Let Σ_{cp} be the fuzzy event set of the C&P decentralized supervisor. Then, $\forall \sigma \in \Sigma_{cp}$:

$$\begin{aligned}
&\Sigma_{cp} = \Sigma_1 \cup ... \cup \Sigma_n \\
&\Sigma_{cp,o}(\sigma) = \Sigma_{1,o}(\sigma)\tilde{\cup}...\tilde{\cup}\Sigma_{n,o}(\sigma) \\
&\Sigma_{cp,c}(\sigma) = \Sigma_{1,c}(\sigma)\tilde{\cup}...\tilde{\cup}\Sigma_{n,c}(\sigma)
\end{aligned} \tag{4.3}$$

where $\Sigma_{cp,o}$ and $\Sigma_{cp,c}$ in (4.3) are the fuzzy observable event set and the fuzzy controllable event set of S^{cp}, respectively. For simplicity, hereafter consider the case in which $n = 2$.

Definition 4.1: Let $\bar{k}$ be a (prefixed-closed) fuzzy sub language over fuzzy event set Σ_{cp}. Assume any fuzzy strings $s, s', s'' \in \bar{k}$ and $s \in P_1^{-1}[P_1(s')]$, $s \in P_2^{-1}[P_2(s'')]$. Then $\bar{k}$ is said to be fuzzy C&P co-observable for any fuzzy string $s \in \Sigma_{cp}^*$, and $\sigma \in \Sigma_{cp}$ with respect to L_G, P_i, and $\Sigma_{i,c}, i = 1, 2$. if the following conditions are satisfied.

$$\begin{aligned}
&\left(\bar{k}(s)\tilde{\cap}L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P_1^{-1}[P_1(s')](s)\tilde{\cap}\Sigma_{1,c}(\sigma) \le \bar{k}(s\sigma)\right), \quad && \text{if } \sigma \in \Sigma_1 \backslash \Sigma_2 \\
&\left(\bar{k}(s)\tilde{\cap}L_G(s\sigma)\tilde{\cap}\bar{k}(s''\sigma)\tilde{\cap}P_2^{-1}[P_2(s'')](s)\tilde{\cap}\Sigma_{2,c}(\sigma) \le \bar{k}(s\sigma)\right), \quad && \text{if } \sigma \in \Sigma_2 \backslash \Sigma_1 \\
&\left(\bar{k}(s)\tilde{\cap}L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P_1^{-1}[P_1(s')](s)\tilde{\cap}\Sigma_{1,c}(\sigma) - \right. && \\
&\left. -\tilde{\cap}\bar{k}(s''\sigma)\tilde{\cap}P_2^{-1}[P_2(s'')](s)\tilde{\cap}\Sigma_{2,c}(\sigma) \le \bar{k}(s\sigma)\right), && \text{if } \sigma \in \Sigma_1 \cap \Sigma_2
\end{aligned} \tag{4.4}$$

The meaning of the above expression can be obtained using terms described in *Definition 3.2*. When the fuzzy events approach crisp events, the fuzzy C&P co-observability is reduced to the crisp version of C&P co-observability defined in [15].

Theorem 4.1: Fuzzy controllability and fuzzy C&P co-observability theorem.

There exists a nonblocking C&P decentralized supervisor of FDES S^{cp} for system G such that $\bar{k} = \bar{L}_{S^{cp}/G,m}$ and $L_{S^{cp}/G} = \bar{k}$ if and only if the following conditions hold:

1. $\bar{k}$ is fuzzy controllable with respect to L_G and $\Sigma_{cp,uc}$.
2. $\bar{k}$ is fuzzy C&P co-observable with respect to L_G, $P_1, ..., P_n$ and $\Sigma_{1,c}, ..., \Sigma_{n,c}$.
3. $\bar{k}$ is $L_{G,m}$-closed.

Proof: See Appendix.

4.1.2 Disjunctive and Anti-permissive (D&A) architecture

This architecture aims to combine the locally-enabled degrees of fuzzy events using an anti-permissive approach. In contrast to the C&P decentralized architecture discussed above, the D&A decentralized architecture is associated with a different set of information.

Figure 4.2 shows the D&A decentralized supervisory control architecture of FDES. Here, the D&A decentralized supervisor of FDES S^{da} fuses the fuzzy events, which are recommended by S^{P_i}, $i \in (1, ..., n)$ using the fuzzy-union operator.

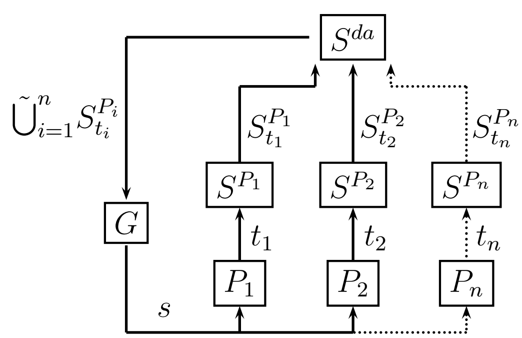


Figure 4.2: D&A decentralized supervisory control architecture of FDES

The final decision of this decentralized supervisor for the system can be computed as follows.

$$S_s^{da} = \tilde{\bigcup}_{i=1}^{n} S_{t_i}^{P_i} \to \max\{S_{t_1}^{P_1}, ..., S_{t_n}^{P_n}\} \tag{4.5}$$

In this architecture the default action of S^{P_i} with insufficient information is to completely disable the fuzzy events. (i.e. $S_{t_i}^{P_i}(\sigma) = 0$; hence, disablement as the default). Let Σ_{da} be the fuzzy event set of the D&A decentralized supervisor. Then, replacing Σ_{cp} with Σ_{da} in (4.3) and assuming $n = 2$ for simplicity, we present the fuzzy D&A co-observability in the following definition.

Definition 4.2: Let $\bar{k}$ be a (prefix-closed) fuzzy sub language over Σ_{da}. Assume any fuzzy strings $s, s', s'' \in \bar{k}$ and $s \in P_1^{-1}[P_1(s')]$, $s \in P_2^{-1}[P_2(s'')]$. Then, $\bar{k}$ is said to be fuzzy D&A co-observable for any fuzzy string $s \in \Sigma_{da}^*$, and $\sigma \in \Sigma_{da}$ with respect to L_G, P_i, and $\Sigma_{i,c}, i = 1, 2.$ if the following conditions are satisfied.

$$\begin{aligned}
&\left(\bar{k}(s)\tilde{\cap}L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P_1^{-1}[P_1(s')](s)\tilde{\cap}\Sigma_{1,c}(\sigma) \le \bar{k}(s\sigma)\right), && \text{if } \sigma \in \Sigma_1 \backslash \Sigma_2 \\
&\left(\bar{k}(s)\tilde{\cap}L_G(s\sigma)\tilde{\cap}\bar{k}(s''\sigma)\tilde{\cap}P_2^{-1}[P_2(s'')](s)\tilde{\cap}\Sigma_{2,c}(\sigma) \le \bar{k}(s\sigma)\right), && \text{if } \sigma \in \Sigma_2 \backslash \Sigma_1 \\
&\left(\bar{k}(s)\tilde{\cap}L_G(s\sigma)\tilde{\cap}\left(\left(\bar{k}(s'\sigma)\tilde{\cap}P_1^{-1}[P_1(s')](s)\tilde{\cap}\Sigma_{1,c}(\sigma)\right) -\right.\right. && \\
&\left.\left.-\tilde{\cup}\left(\bar{k}(s''\sigma)\tilde{\cap}P_2^{-1}[P_2(s'')](s)\tilde{\cap}\Sigma_{2,c}(\sigma)\right)\right) \le \bar{k}(s\sigma)\right), && \text{if } \sigma \in \Sigma_1 \cap \Sigma_2
\end{aligned} \tag{4.6}$$

Theorem 4.2: Fuzzy controllability and fuzzy D&A co-observability theorem.

There exists a nonblocking D&A decentralized supervisor of FDES S^{da} for system G such that $\bar{k} = \bar{L}_{S^{da}/G,m}$ and $L_{S^{da}/G} = \bar{k}$ if and only if the following conditions hold (Note that $\Sigma_{da,c}$ and $\Sigma_{da,uc}$ are the fuzzy controllable event set and the fuzzy uncontrollable event set of this architecture respectively).

1. $\bar{k}$ is fuzzy controllable with respect to L_G and $\Sigma_{da,uc}$.
2. $\bar{k}$ is fuzzy D&A co-observable with respect to L_G, $P_1, ..., P_n$ and $\Sigma_{1,c}, ..., \Sigma_{n,c}$
3. $\bar{k}$ is $L_{G,m}$-closed.

Proof: See Appendix.

4.1.3 General architecture

The key difference between C&P and D&A decentralized supervisory control architectures of FDES lies in their event fusion methods. The fusion method of C&P architecture in (4.2) combines the locally-enabled degrees of fuzzy events using fuzzy-intersection, whereas the fusion method of D&A architecture in (4.5) combines those degrees of fuzzy events using fuzzy-union. Hence incorporating both of these architectures will provide improved information association and ambiguity management in decentralized decision making.

The proposed decentralized supervisory control architecture of FDES is shown in figure 4.3. Here, the general decentralized supervisor S^g receives the recommendations of both S^{cp} and S^{da} and combines them using set union.

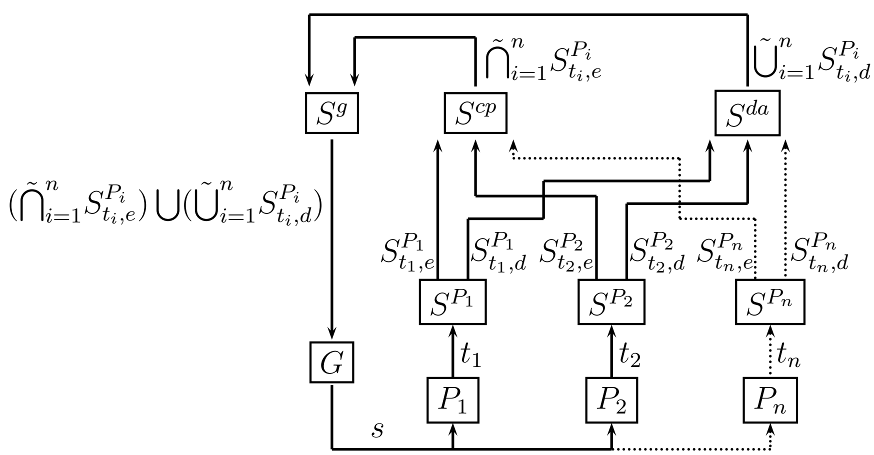


Figure 4.3: General decentralized supervisory control architecture of FDES

Let Σ_g be the set of fuzzy events of the general decentralized supervisor. Then, $\Sigma_g = \Sigma_{cp} \cup \Sigma_{da}$. We assume S^{P_i}, $i \in \{1, ..., n\}$ has a *priori* knowledge of the fusion method of each of its events (either fuzzy-intersection or fuzzy-union). This defines two subsets $\Sigma_{i,e}$ and $\Sigma_{i,d}$ (where $\Sigma_i = \Sigma_{i,e} \cup \Sigma_{i,d}$ and $\Sigma_{i,e} \cap \Sigma_{i,d} = \emptyset$) for S^{P_i}, as described in (4.7).

Note that the decentralized supervisory control framework of crisp DES presented in [36] does not require any *priori* partition of controllable events into permissive or anti-permissive subsets. However, in this architecture each local supervisor needs to compute a grade of ambiguity for their (local) decisions. This ambiguity-grade depends on the self-ambiguity as well as the ambiguities of others and hence a communication link needs to be maintained among supervisors. In this thesis, we study the decentralized supervisory control of FDES with non-communicating local controllers and they only need to report their decisions to the global (decentralized) supervisor.

For all $\sigma,\ \sigma \in \Sigma_i$:

$$\begin{aligned} &\sigma \in \Sigma_{i,e}, \quad \text{if the default is enablement: fuse using (4.2).} \\ &\sigma \in \Sigma_{i,d}, \quad \text{if the default is disablement: fuse using (4.5).} \end{aligned} \tag{4.7}$$

The final decision of the general decentralized supervisor S^g is computed by combining (4.2) and (4.5) as follows.

$$S_s^g = \left(\tilde{\bigcap}_{i=1}^{n} S_{t_i,e}^{P_i}\right) \bigcup \left(\tilde{\bigcup}_{i=1}^{n} S_{t_i,d}^{P_i}\right) \rightarrow \min\{S_{t_1,e}^{P_1}, ..., S_{t_n,e}^{P_n}\} + \max\{S_{t_1,d}^{P_1}, ..., S_{t_n,d}^{P_n}\} \tag{4.8}$$

Let $\Sigma_{i,e,c}$ and $\Sigma_{i,d,c}$ be the subsets of fuzzy controllable events of $\Sigma_{i,e}$ and $\Sigma_{i,d}$ respectively. For each $S^{P_i}, i \in \{1, ..., n\}$: $\Sigma_{i,e} = \Sigma_i \cap \Sigma_{cp}$ and $\Sigma_{i,d} = \Sigma_i \cap \Sigma_{da}$.

Using (4.4) and (4.6) and assuming $n = 2$ for simplicity, we now define a general form of fuzzy co-observability.

Definition 4.3: Let $\bar{k}$ be a fuzzy sub language over fuzzy event set Σ_g. Assume any fuzzy strings $s, s', s'' \in \bar{k}$ and $s \in P_1^{-1}[P_1(s')]$, $s \in P_2^{-1}[P_2(s'')]$. Then, $\bar{k}$ is said to be fuzzy co-observable for any fuzzy string $s \in \Sigma_g^*$, and $\sigma \in \Sigma_g$ with respect to L_G,

P_i, $\Sigma_{i,e,c}$ and $\Sigma_{i,d,c}$ where $i = 1, 2.$, if the following conditions are satisfied:

$$\begin{array}{l} 1.\ \bar{k} \text{ is fuzzy C\&P co-observable with respect to } L_G,\ P_i,\ \Sigma_{i,e,c}. \\ 2.\ \bar{k} \text{ is fuzzy D\&A co-observable with respect to } L_G,\ P_i,\ \Sigma_{i,d,c}. \end{array} \tag{4.9}$$

This general form of fuzzy co-observability leads to theorize the existence result of the general decentralized control architecture of FDES as follows.

Theorem 4.3: Fuzzy controllability and general fuzzy co-observability theorem.

There exists a nonblocking fuzzy generalized decentralized supervisor S^g for system G such that $\bar{k} = \bar{L}_{S^g/G,m}$ and $L_{S^g/G} = \bar{k}$ if and only if the following conditions hold:

1. $\bar{k}$ is fuzzy controllable with respect to L_G and $\Sigma_{g,uc}$.

2. $\bar{k}$ is fuzzy co-observable with respect to L_G, P_i, $\Sigma_{i,e,c}$ and $\Sigma_{i,d,c}$ where $i \in \{1, ..., n\}$.

3. $\bar{k}$ is $L_{G,m}$-closed.

Proof: See Appendix.

Note that in this thesis, *Theorem 4.3* is also referred to as the general decentralized supervisory control theory of FDES.

The following properties can be observed.

1. In general, fuzzy languages generated by the C&P decentralized supervisor of FDES and the D&A decentralized supervisor of FDES are not the same: $L_{S^{cp}/G}(s) \neq L_{S^{da}/G}(s)$.

2. In general, the possibility of a fuzzy string belonging to the fuzzy language generated by the C&P decentralized supervisor of FDES, or the D&A decentralized supervisor of FDES, is not larger than its possibility of belonging to the fuzzy language generated by the general decentralized supervisor of FDES: $L_{S^{cp}/G}(s) \tilde{\cup} L_{S^{da}/G}(s) \leq L_{S^g/G}(s)$.

4.2 Application to decentralized control of MRS

A multi robot object manipulation task is performed using the proposed general architecture of FDES. Assume a robot team is comprised of two autonomous mobile robots (i.e $n = 2$). A beam of 1.0m length represents the object to be transported to the goal position. It is assumed that both ends of the beam are hinged at the center points of the robots facilitating free rotation around the contact points. Hence the distance between the center points is fixed with the beam length. Also, assume non-skid motion of robot wheels. Figure 4.4 shows the above scenario, in which two robots collaboratively transport a beam by following way-points.

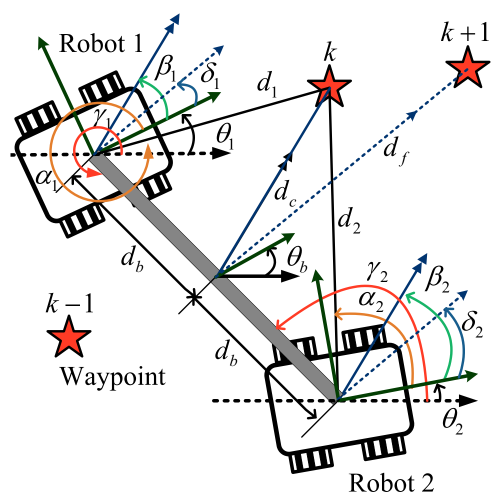


Figure 4.4: Two robots moving a beam

In this figure, θ_i is the i^{th} robot heading with respect to the horizontal axis, α_i is the angle between i^{th} robot heading and the current (k^{th}) way-point and d_i is the distance between i^{th} robot center point and the way-point. The angle between the i^{th} robot heading and the line connecting the beam center and the current way-point is given by β_i. In the same manner, δ_i is this angle when calculated for the next ($k + 1^{\text{th}}$) way-point. The angle of the beam with respect to the horizontal axis measured from i^{th} robot is given by γ_i. (Note that γ_i can be measured by using an angle sensor installed on i^{th} robot and θ_i.) θ_b is the heading of the beam center with respect to the horizontal axis and half of the beam length is given by d_b. The distance to the current way-point from the beam center is d_c and d_f is the distance to the next way-point from the beam center. It is desirable to follow the way-points by the beam center and control the robots accordingly. Based on the above assumptions the following can be derived for robot i.

$$\beta_i = \tan^{-1} \frac{d_b \sin(\theta_i - \gamma_i) + d_i \sin \alpha_i}{d_i \cos \alpha_i - d_b \cos(\theta_i - \gamma_i)} \tag{4.10}$$

$$d_c = \frac{d_b \sin(\theta_i + \alpha_i - \gamma_i)}{\sin(\beta_i - \alpha_i)} \tag{4.11}$$

Similar expressions can be derived for δ_i and d_f. Therefore, each robot has sufficient information for decentralized decision making for the MRS based on its sensory perceptions. Note that based on the above assumptions $\theta_1 \approx \theta_2 \approx \theta_b$. For simplicity, let us consider the case in which $\gamma_i = \frac{\pi}{2}$.

4.2.1 Modelling of the MRS

To control the MRS a behavior-based approach is used. Four behaviors are defined, namely two deliberative behaviors ("Follow Route" and "Go to Target") and two reactive behaviors ("Avoid Obstacle" and "Follow Wall"). To model the MRS according to the i^{th} robot's perception a fuzzy automaton G_i is used.

$$G_i = (Q_i, \Sigma_i, \delta_i, q_{0,i}) \tag{4.12}$$

The fuzzy state vector Q_i represents the behaviors of the MRS according to the i^{th} robot's perception as:

$$Q_i = \{fr_i, gt_i, ao_i, fw_i\} \tag{4.13}$$

where fr_i, gt_i, ao_i and fw_i represent Follow Route, Go to Target, Avoid Obstacle and Follow Wall behaviors of MRS respectively according to the perception of the i^{th} robot.

The fuzzy event set Σ_i includes the fuzzy events, which change the activation levels of the above behaviors.

$$\Sigma_i = \{\sigma_{fr,i}, \sigma_{gt,i}, \sigma_{ao,i}, \sigma_{fw,i}\} \tag{4.14}$$

Also, δ_i and $q_{0,i}$ hold their usual definitions. Figure 4.5 depicts G_i.

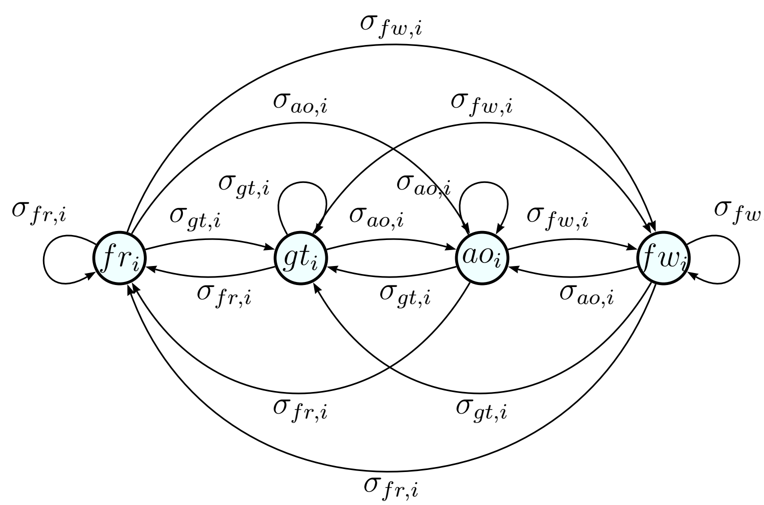


Figure 4.5: Fuzzy automaton G_i

4.2.2 Behavior modeling

As in section 3.3.3, a fuzzy rule base is constructed for each fuzzy event in (4.14) to calculate its possibility of being enabled by the i^{th} supervisor (robot). The following assumptions are made for simplicity.

1. Each fuzzy string is completely physically possible (i.e., $L_{G_i}(s_i\sigma_i) = 1.0$).

2. Each fuzzy event is fully controllable for its local supervisors (i.e.$\Sigma_{i,c}(\sigma_i) = 1.0$).

3. Each fuzzy event is fully observable to the general decentralized supervisor (i.e., $\Sigma_{g,o}(\sigma_i) = 1.0$).

These fuzzy rule bases are represented in Tables 4.1 - 4.4. In 4.1, $S_{t_i}^{P_i}(\sigma_{ao,i})$ is the possibility of the fuzzy event $\sigma_{ao,i}$ being enabled by i^{th} local fuzzy supervisor S^{P_i} after observing the fuzzy string s with projection P_i where $P_i(s) = t_i$. Likewise, $S_{t_i}^{P_i}(\sigma_{fw,i})$, $S_{t_i}^{P_i}(\sigma_{fr,i})$ and $S_{t_i}^{P_i}(\sigma_{gt,i})$ have respective meanings.

Figure 4.6 shows the membership functions used to calculate $S_{t_i}^{P_i}(\sigma_{fr,i})$. The symmetric membership functions and values of figure 4.6 have been determined experimentally to have successful navigational results.

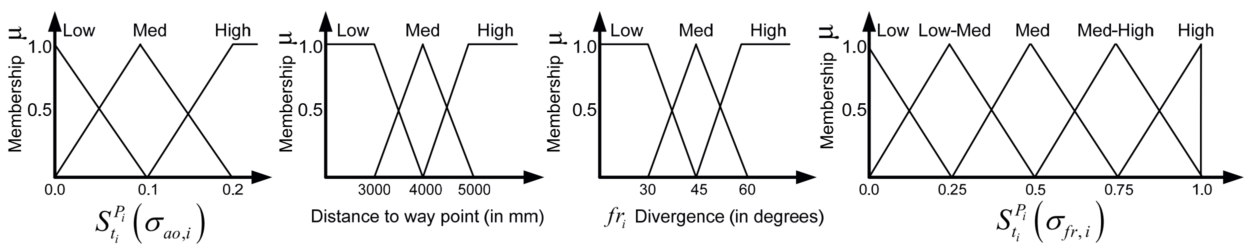


Figure 4.6: Membership functions used to calculate $S_{t_i}^{P_i}(\sigma_{fr,i})$ for i^{th} robot

Table 4.1: Fuzzy rule base for $S_{t_i}^{P_i}(\sigma_{ao,i})$

Distance to nearest obstacle	$\lvert\theta_i - \measuredangle\overrightarrow{ao_i}\rvert$	$S_{t_i}^{P_i}(\sigma_{ao,i})$
Low	Low	Medium
	Medium	Medium-High
	High	High
Medium	Low	Low-Medium
	Medium	Medium
	High	Medium-High
High	Low	Low
	Medium	Low-Medium
	High	Medium

Table 4.2: Fuzzy rule base for $S_{t_i}^{P_i}(\sigma_{fw,i})$

Distance to nearest obstacle	$\lvert\theta_i - \measuredangle\overrightarrow{fw_i}\rvert$	$S_{t_i}^{P_i}(\sigma_{fw,i})$
Low	Low	Medium
	Medium	Medium-High
	High	High
Medium	Low	Low-Medium
	Medium	Medium
	High	Medium-High
High	Low	Low
	Medium	Low-Medium
	High	Medium

Table 4.3: Fuzzy rule base for $S_{t_i}^{P_i}(\sigma_{fr,i})$

$S_{t_i}^{P_i}(\sigma_{ao,i})$	d_c	$\lvert\theta_i - \measuredangle\overrightarrow{fr_i}\rvert$ ($\lvert\beta_i\rvert$)	$S_{t_i}^{P_i}(\sigma_{fr,i})$
Low	Low	Low	Low
		Medium	Low-Medium
		High	Medium
	Medium	Low	Low-Medium
		Medium	Medium
		High	Medium-High
	High	Low	Medium
		Medium	Medium-High
		High	High
Medium	Low	Low	Low
		Medium	Low-Medium
		High	Medium
	Medium	Low	Low-Medium
		Medium	Medium
		High	Medium
	High	Low	Medium
		Medium	Medium
		High	Medium
High	Low	Low	Low
		Medium	Low
		High	Low
	Medium	Low	Low
		Medium	Low
		High	Low
	High	Low	Low
		Medium	Low
		High	Low

Table 4.4: Fuzzy rule base for $S_{t_i}^{P_i}(\sigma_{gt,i})$

$S_{t_i}^{P_i}(\sigma_{ao,i})$	d_f	$\lvert\theta_i - \measuredangle\overrightarrow{gt_i}\rvert$ ($\lvert\delta_i\rvert$)	$S_{t_i}^{P_i}(\sigma_{gt,i})$
Low	Low	Low	High
		Medium	Medium-High
		High	Medium
	Medium	Low	Medium-High
		Medium	Medium
		High	Low-Medium
	High	Low	Medium
		Medium	Low-Medium
		High	Low
Medium	Low	Low	Medium
		Medium	Medium
		High	Medium
	Medium	Low	Medium
		Medium	Medium
		High	Low-Medium
	High	Low	Medium
		Medium	Low-Medium
		High	Low
High	Low	Low	Low
		Medium	Low
		High	Low
	Medium	Low	Low
		Medium	Low
		High	Low
	High	Low	Low
		Medium	Low
		High	Low

Each local supervisor (robot) enables the fuzzy events according to its own perception. Assume σ_{fr}, σ_{gt}, σ_{ao}, $\sigma_{fw} \in \Sigma_g$ are the fuzzy events enabled by the general decentralized supervisor of FDES S^g for the MRS. According to (4.8), these events can be calculated as follows.

$$\begin{aligned} S_s^g(\sigma_{fr}) &= \tilde{\bigcap}_{i=1}^{2} S_{t_i}^{P_i}(\sigma_{fr,i}) \\ S_s^g(\sigma_{gt}) &= \tilde{\bigcap}_{i=1}^{2} S_{t_i}^{P_i}(\sigma_{gt,i}) \\ S_s^g(\sigma_{ao}) &= \tilde{\bigcup}_{i=1}^{2} S_{t_i}^{P_i}(\sigma_{ao,i}) \\ S_s^g(\sigma_{fw}) &= \tilde{\bigcup}_{i=1}^{2} S_{t_i}^{P_i}(\sigma_{fw,i}) \end{aligned} \tag{4.15}$$

As in (3.23), a matrix M_{t+1} can be constructed to represent all fuzzy events which are enabled by S^g for time $t+1$ after observing fuzzy string s.

$$M_{t+1} = \begin{bmatrix} S_s^g(\sigma_{fr}) & S_s^g(\sigma_{gt}) & S_s^g(\sigma_{ao}) & S_s^g(\sigma_{fw}) \\ S_s^g(\sigma_{fr}) & S_s^g(\sigma_{gt}) & S^{g_s}(\sigma_{ao}) & S_s^g(\sigma_{fw}) \\ S_s^g(\sigma_{fr}) & S_s^g(\sigma_{gt}) & S_s^g(\sigma_{ao}) & S_s^g(\sigma_{fw}) \\ S_s^g(\sigma_{fr}) & S_s^g(\sigma_{gt}) & S_s^g(\sigma_{ao}) & S_s^g(\sigma_{fw}) \end{bmatrix} \tag{4.16}$$

Then, this matrix is applied to each robot of the MRS. The final behavior activation levels of each robot can be calculated as in (3.24).

4.2.3 Implementation

Mobile robot simulations were performed using MobileSim (Version 0.4.0) provided by ActivMedia Robotics, based on Pioneer 3 DX robots. A 12m $\times$ 20m simulated environment space was used and start and goal points were identified. The waypoints were assigned manually corresponding to the center point of the beam. The robots were localized using dead reckoning.

Distance to the obstacles from each robot was measured by using their embedded sonar rings. For testing, several simulated environments with unmodeled obstacles were considered. Each decision cycle consists of a rotation and a translation command. Rotation was used for angular correction and translation was used to move the robot in the final chosen direction. Robot translation speed was fixed at 50mm per second and the translation cycle was 50ms. Its rotation speed was proportional to the desired heading. For simplicity, it is assumed that an error-free perfect communication channel exists between the local supervisors and the decentralized supervisor. The performance of the proposed general decentralized architecture of FDES in modeling and control of an MRS was compared with the centralized FDES-based architecture and general decentralized architecture of crisp DES.

Centralized control architecture based on FDES

Since the behavior of the MRS according to the i^{th} robot's perception is modeled by a fuzzy automaton G_i, the MRS can be completely modeled by a fuzzy automaton $G_{sys} = (Q_{sys}, \Sigma_{sys}, \delta_{sys}, q_{0,sys})$ where $G_{sys} = G_1||...||G_n$ (in this application n=2). Note that in this case the fuzzy event matrices corresponding to each model G_i are constructed individually according to the available local information. The calculation of fuzzy events and states of the composed system model G_{sys} ($= G_1||G_2$) is performed as in (3.2). Since both G_1 and G_2 consist of four fuzzy states to model behaviors individually, the total number of states of G_{sys} will be sixteen. Assume the fuzzy state vector of G_{sys} is shown as follows.

$$\begin{aligned} Q_{sys} = Q_1 \otimes Q_2 = \quad & (fr_1/fr_2, fr_1/gt_2, fr_1/ao_2, fr_1/fw_2, - \\ & gt_1/fr_2, gt_1/gt_2, gt_1/ao_2, gt_1/fw_2, - \\ & ao_1/fr_2, ao_1/gt_2, ao_1/ao_2, ao_1/fw_2, - \\ & fw_1/fr_2, fw_1/gt_2, fw_1/ao_2, fw_1/fw_2) \end{aligned} \tag{4.17}$$

Then, the weights of the behaviors controlling the MRS are calculated as follows.

$$\begin{aligned} &\text{Weight of Follow Route} = fr_1/fr_2 \\ &\text{Weight of Go to Target} = gt_1/gt_2 \\ &\text{Weight of Avoid Obstacle} = \max(fr_1/ao_2, gt_1/ao_2, ao_1/fr_2, ao_1/gt_2, ao_1/ao_2) \\ &\text{Weight of Follow Wall} = \max(fr_1/fw_2, gt_1/fw_2, fw_1/fr_2, fw_1/gt_2, fw_1/fw_2) \end{aligned} \tag{4.18}$$

The control commands of each robot of the MRS are calculated according to (4.18). This leads to a command fusion type of behavior coordination in the MRS.

General decentralized control architecture based on crisp DES

In this case both robots in the MRS are controlled by an approach based on the general decentralized control architecture presented in [15]. As DES-based modeling represents a system in only one state at any given time, the MRS will be controlled by only one behavior accordingly. This leads to behavior arbitration type of behavior coordination in the MRS.

Proposed general decentralized control architecture based on FDES

In this case both robots are controlled by an approach based on proposed general decentralized control architecture of FDES. This also leads to command fusion type behavior coordination in the MRS.

4.2.4 The computational complexity of the proposed approach

A mobile robot controlled by n number of behaviors can be effectively modeled by a fuzzy automaton with n number of fuzzy states and n^2 number of fuzzy event transitions. Hence, the dimensions of the resulting fuzzy state matrix and event matrix are $(1 \times n)$ and $(n \times n)$, respectively. Assume an MRS with m number of robots. If the MRS is modeled either by using C&P decentralized control architecture of FDES or by D&A decentralized control architecture of FDES alone, the computational complexity of the system will be $\mathcal{O}(m \times n^2)$ neglecting the complexity of the defuzzification step of the fuzzy rule bases. Since the general decentralized control architecture of FDES represents both C&P and D&A decentralized architectures of FDES, in the worst case the computational complexity can be calculated as $\mathcal{O}(2m \times n^2)$. Hence, the proposed general decentralized architecture of FDES has higher computational complexity compared to C&P or D&A decentralized control architectures alone.

The centralized architecture of FDES consists of parallel composition of m number of modules (i.e. $G_{sys} = G_1||...||G_m$). Hence, the resulting fuzzy state matrix and event matrix are $(1 \times n^m)$ and $(n^m \times n^m)$ in dimensions, respectively. As a result, the complexity of the composed system will be $\mathcal{O}(n^{2m})$. Hence the computational demand of the proposed general decentralized architecture of FDES is less compared to that of the centralized architecture of FDES.

With the increased number of concurrently acting robots in the MRS, the centralized architecture rapidly increases the computational complexity. Moreover, the communication and resource allocation issues of the MRS also demand higher computing power. This problem of increased demand of computing power in centralized systems can be solved by adopting a decentralized control approach. However, as the decentralized control approaches do not incorporate all the information available in the MRS, the solutions may often end up in local minima.

4.2.5 Simulation results

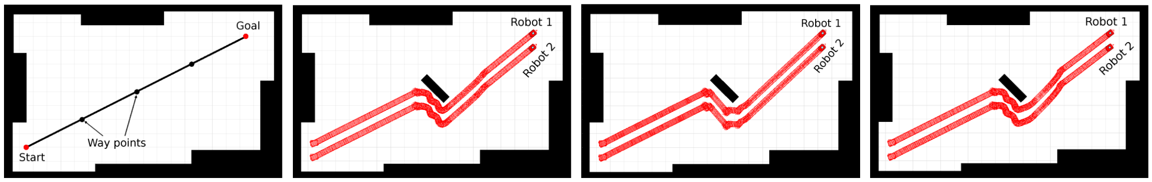
Test I: Environment with one unmodeled obstacle

Figure 4.7 shows the navigation results of the MRS moving in an environment with one unmodeled obstacle, by employing different supervisory control architectures.

In figure 4.8(a), $\alpha_1, \alpha_2, \alpha_3$ and α_4 depict the evolution of Follow Route, Go to Target, Avoid Obstacles, and Follow Wall behaviors respectively when the MRS is controlled by the centralized FDES-based control architecture. Figure 4.8(b) shows the heading of both robots in each decision cycle.

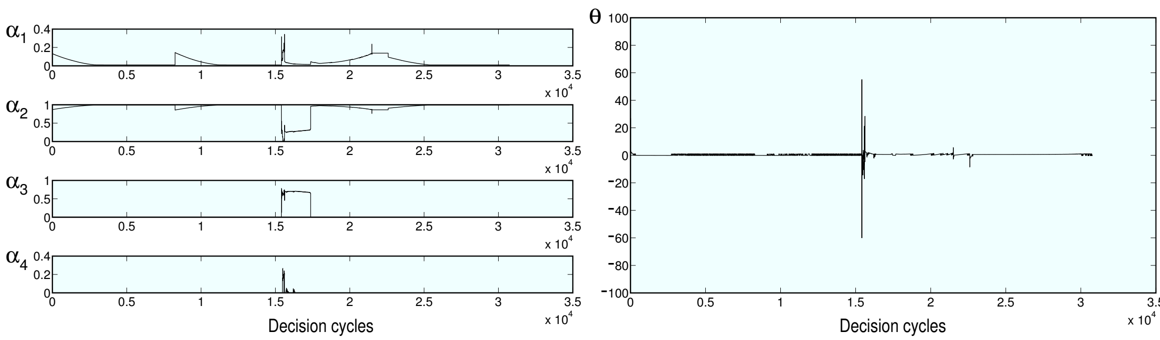
Figure 4.9(a) shows the evolution of behaviors when the MRS is controlled by the general decentralized crisp DES-based control architecture. Figure 4.9(b) shows the heading of both robots at each decision cycle.

Figure 4.10(a) shows the evolution of behaviors when the MRS is controlled by the proposed general decentralized FDES-based control architecture. Figure 4.10(b) shows the heading of both robots in each decision cycle.



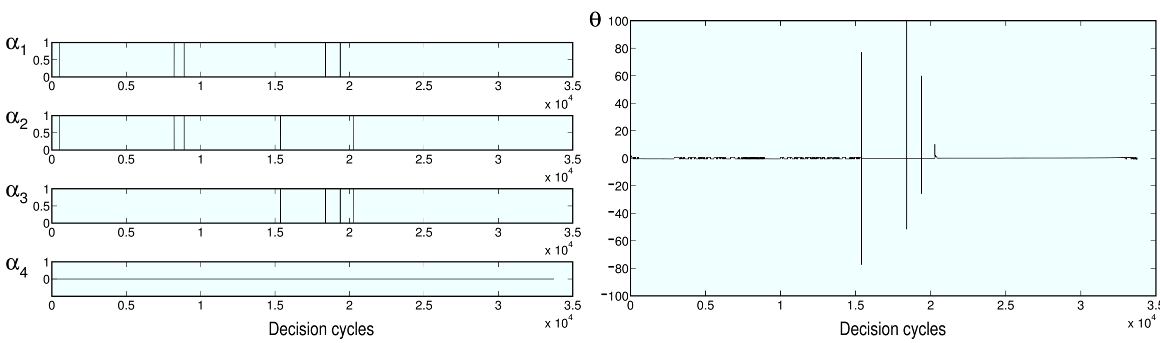
(a) Original map (b) Centralized FDES (c) Decentralized DES (d) Decentralized FDES

Figure 4.7: Navigation results for Test I



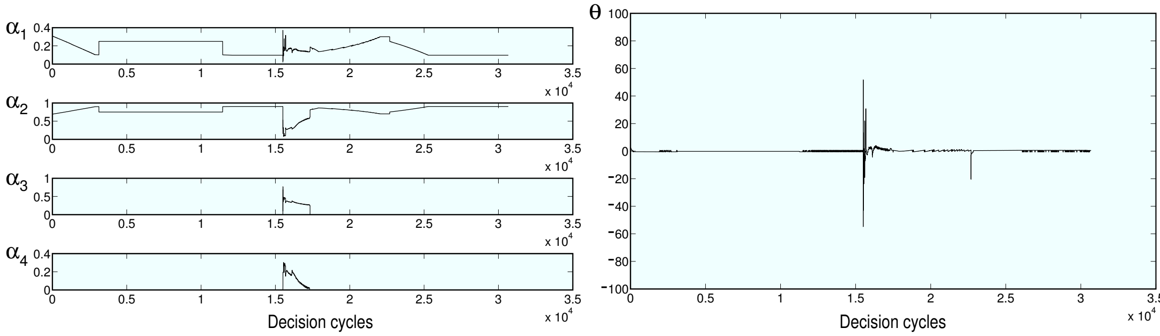
(a) Behavior weights (b) Heading of both robots

Figure 4.8: Plots related to centralized FDES-based control for Test I



(a) Behavior weights (b) Heading of both robots

Figure 4.9: Plots related to decentralized crisp DES-based control for Test I



(a) Behavior weights   (b) Heading of both robots

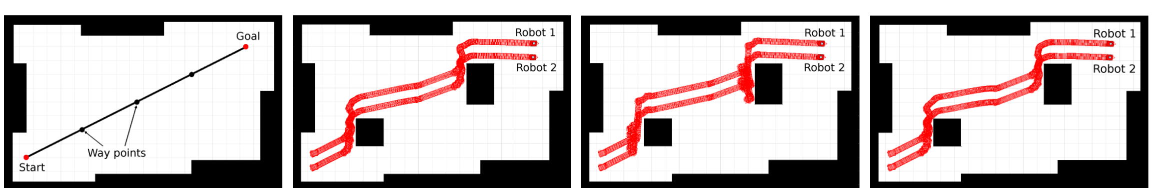
Figure 4.10: Plots related to decentralized FDES-based control for Test I

Test II: Environment with two unmodeled obstacles

Figure 4.11 shows the navigation scenarios of the MRS moving in an environment with two unmodeled obstacles in the path. Figure 4.12(a) shows the evolution of behaviors when the MRS is controlled by the centralized FDES-based control architecture. Figure 4.12(b) shows the heading of both robots in each decision cycle.

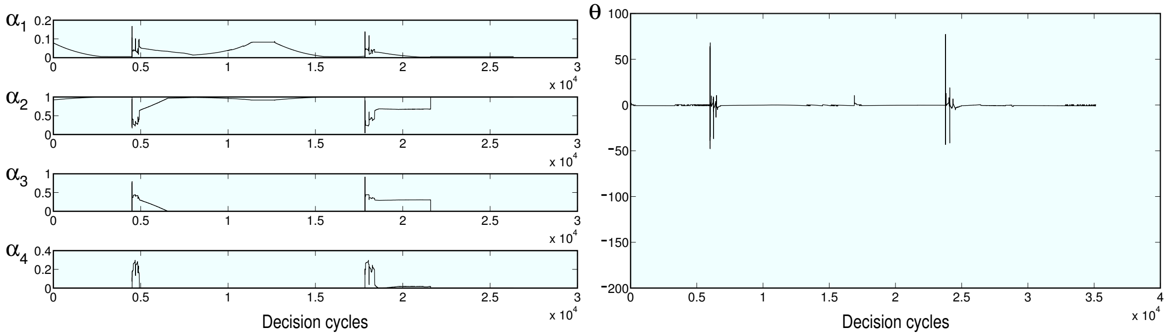
Figure 4.13(a) shows the evolution of behaviors when the MRS is controlled by the general decentralized crisp DES-based control architecture. Figure 4.13(b) shows the heading of both robots in each decision cycle.

Figure 4.14(a) shows the evolution of behaviors when the MRS is controlled by the proposed general decentralized FDES-based control architecture. Figure 4.14(b) shows the heading of both robots in each decision cycle.



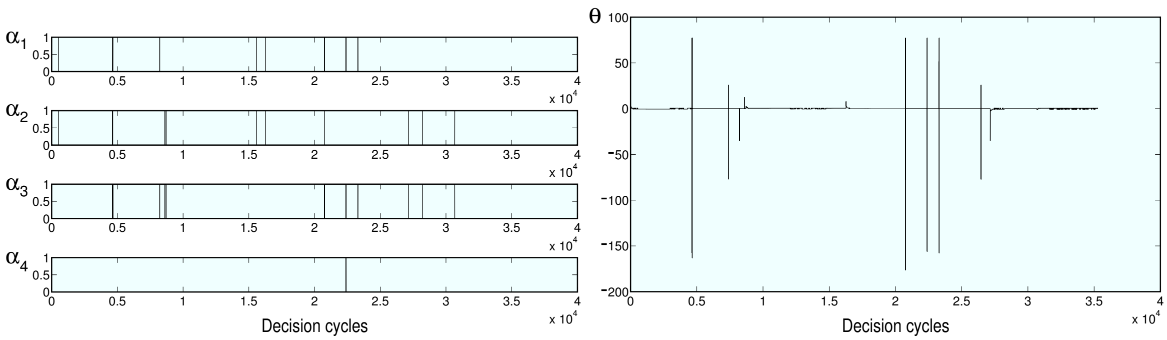
(a) Original map   (b) Centralized FDES   (c) Decentralized DES   (d) Decentralized FDES

Figure 4.11: Navigation results for Test II



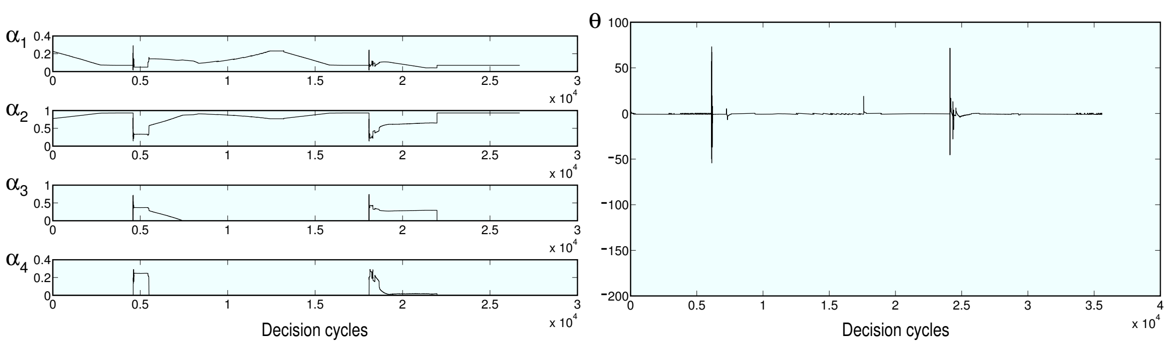
(a) Behavior weights (b) Heading of both robots

Figure 4.12: Plots related to centralized FDES-based control for Test II



(a) Behavior weights (b) Heading of both robots

Figure 4.13: Plots related to decentralized crisp DES-based control for Test II



(a) Behavior weights (b) Heading of both robots

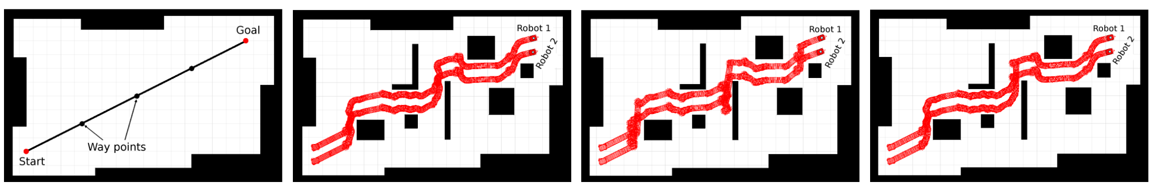
Figure 4.14: Plots related to decentralized FDES-based control for Test II

Case III: Cluttered environment with several unmodeled obstacles

Figure 4.15 shows the navigation scenarios of the MRS moving in an environment with two unmodeled obstacles in the path. Figure 4.16(a) shows the evolution of behaviors when the MRS is controlled by the centralized FDES-based control architecture. Figure 4.16(b) shows the heading of both robots in each decision cycle.

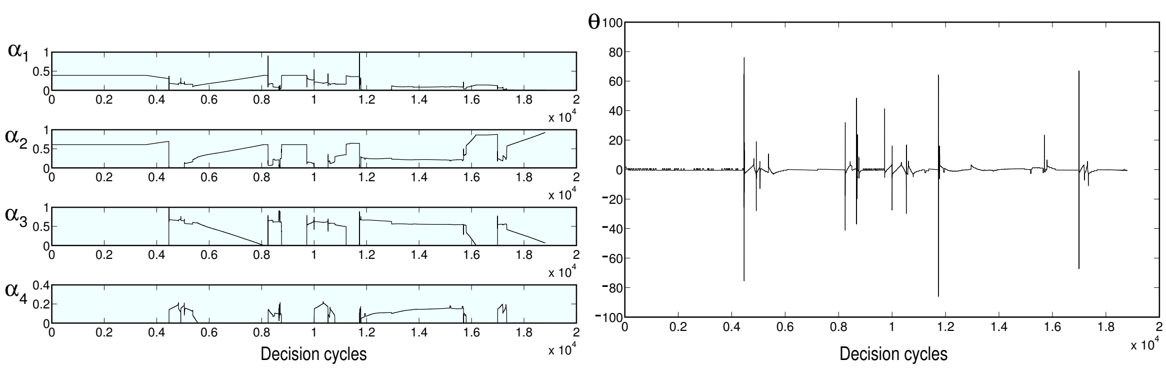
Figure 4.17(a) shows the evolution of behaviors when the MRS is controlled by the general decentralized crisp DES-based control architecture. Figure 4.17(b) shows the heading of both robots in each decision cycle.

Figure 4.18(a) shows the evolution of behaviors when the MRS is controlled by the proposed general decentralized FDES-based control architecture. Figure 4.18(b) shows the heading of both robots in each decision cycle.



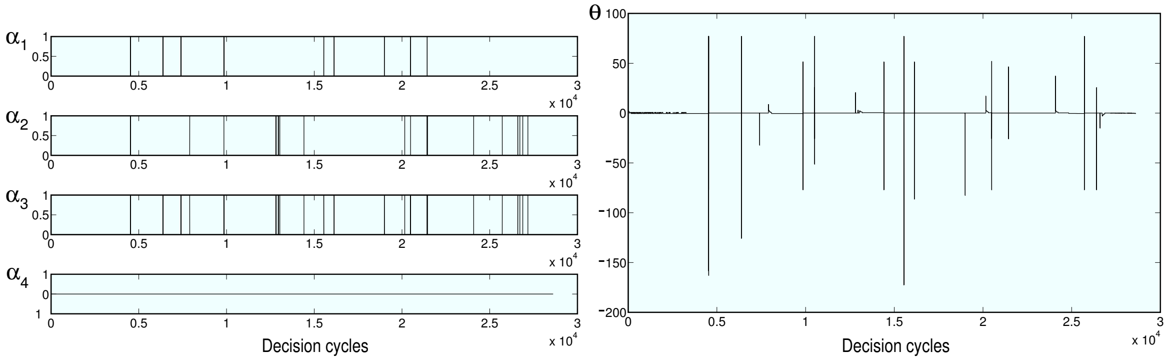
(a) Original map (b) Centralized FDES (c) Decentralized DES (d) Decentralized FDES

Figure 4.15: Navigation results for Test III



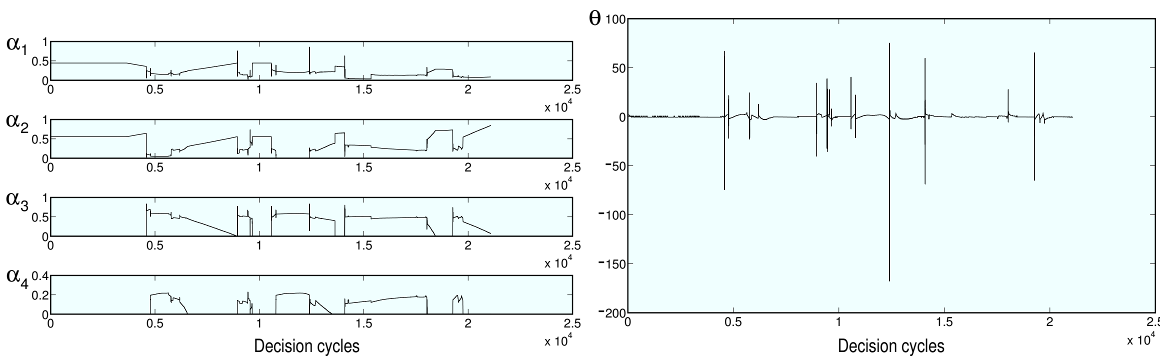
(a) Behavior weights (b) Heading of both robots

Figure 4.16: Plots related to centralized FDES-based control for Test III



(a) Behavior weights (b) Heading of both robots

Figure 4.17: Plots related to decentralized crisp DES-based control for Test III



(a) Behavior weights (b) Heading of both robots

Figure 4.18: Plots related to decentralized FDES-based control for Test III

4.2.6 Performance evaluation

The performance of the proposed architecture is compared to the centralized FDES and decentralized crisp DES-based approaches using the following metrics [97, 98]:

1. **Total time to goal reach,** T_{tot}**:** For high performance, it is desirable to have a low execution time.

2. **Total path length,** L_{tot}**:** Having shorter length is desirable for better performance.

3. **Bending energy,** B_E**:** Having low bending energy is desirable as it increases

the smoothness and decreases the energy requirement.

4. **Mean distance to obstacles from i^{th} Robot, $D_{Mean,i}$:** The average of the minimum distances between i^{th} robot and the obstacles measured in each execution cycle through the entire navigation. A higher value ensures secure navigation.

5. **Minimum distance to obstacles from i^{th} Robot, $D_{Min,i}$:** The minimum distance between i^{th} robot and obstacles through the entire navigation. This indicates the risk taken through the entire movement.

6. **Number of collisions, N_{col}:** A collision free operation indicates a safe navigation of the robot.

Based on the above metrics a performance evaluation was performed and the results are shown in Table 4.5.

Table 4.5: Performance evaluation of the presented architectures

Case	Architecture type	T_{tot} (s)	L_{tot} $\times 10^1$ (m)	B_E $\times 10^{-4}$	$D_{Mean,1}$ $\times 10^{-1}$(m)	$D_{Min,1}$ $\times 10^{-1}$(m)	$D_{Mean,2}$ $\times 10^{-1}$(m)	$D_{Min,2}$ $\times 10^{-1}$(m)	N_{col}
I	Centralized FDES	376	1.95	5.73	3.15	2.18	≥ 7	≥ 7	0
	Decentralized DES	382	1.96	8.14	5.29	3.98	≥ 7	≥ 7	0
	Decentralized FDES	380	1.95	4.98	4.0	2.41	≥ 7	≥ 7	0
II	Centralized FDES	399	2.05	3.77	5.25	3.42	4.33	1.74	0
	Decentralized DES	477	2.40	9.49	2.54	1.66	4.09	2.37	0
	Decentralized FDES	400	2.06	3.90	5.1	3.39	4.12	1.00	0
III	Centralized FDES	417	2.14	5.85	3.46	2.07	4.92	2.3	0
	Decentralized DES	455	2.30	59.40	3.40	1.16	3.79	1.00	0
	Decentralized FDES	420	2.13	4.47	3.25	2.01	4.38	2.26	0

Based on the Table 4.5, all employed architectures were successful. The centralized FDES architecture outperformed others by having a less total time, but it needs a higher computing power. The decentralized control architecture based on crisp

DES was the slowest approach and had the highest energy requirement. This is due to the chattering effects it introduced near to the obstacles. The proposed FDES-based decentralized control architecture produced approximate results compared with the centralized FDES architecture, although the former has a less computational complexity.

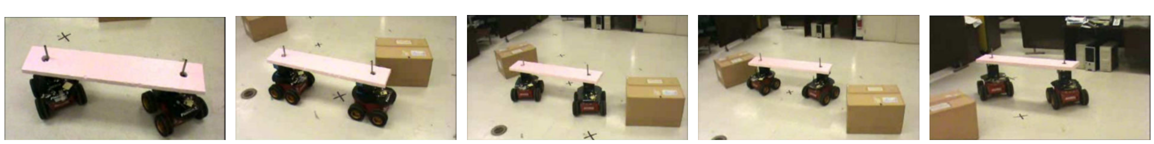
4.2.7 Real-time implementation

The proposed architecture is implemented in real-time on an MRS having two physical robots (Pioneer 3 AT) moving in an environment with unmodeled obstacles. The object to transport is a light styrofoam board having the dimensions of 150cm $\times$ 30cm $\times$ 5cm, which is hinged to both robots at the ends. Here also, dead reckoning was used to localize the robots. The results are shown in figures 4.19 and 4.20.

The accompanying movies depict the real-time implementation. Movies 4.1 and 4.2 show the navigations of MRS in environments with one and two unmodeled obstacles respectively.

(a) Route following (b) Avoiding obstacle (c) Wall following (d) Turning at obstacle (e) Approaching goal

Figure 4.19: Real-time navigation: In an environment with 1 obstacle



(a) Route following (b) Avoiding obstacle (c) Wall following (d) Avoiding obstacle (e) Approaching goal

Figure 4.20: Real-time navigation: In an environment with 2 obstacles

4.3 Summary

This chapter discussed the decentralized supervisory control of FDES in a general framework. Mainly, the decentralized supervisory control architectures, which represent "fusion by intersection" and "fusion by union" were extended to the fuzzy domain. Then, combining both of these architectures, a general form of decentralized supervisory control architecture was established for FDES. Furthermore, both event controllability and event observability were considered as fuzzy.

The proposed architecture was implemented in both simulation and real-time in a tightly-coupled multi robot object manipulation task. The application represents a decentralized MRS where local decisions are made by each robot and the global decision is computed by a host computer. Due to sensor limitations each robot is aware of its local environment only and makes suboptimal decisions, which must be fused to achieve a globally optimal decision. Hence, the task is a good candidate for modeling as a decentralized FDES.

A performance evaluation was also carried out and the proposed method was compared with two other approaches. The proposed method gave successful results with less computational demand.

Chapter 5

Decentralized modular control of FDES

In this chapter, a supervisory control theory is established for concurrently operating multiple interacting FDES modules. This helps to resolve the horizontal control complexities present in large-scale systems that are modeled as FDES. Firstly, some important definitions are provided to extend the crisp DES modular control theory to FDES. Secondly, incorporating those definitions, a decentralized modular supervisory control theory for concurrent FDES is established.

5.1 Modular and concurrent FDES

Assume a group of n modules concurrently processing and distributed over an area where each module has different sites and no communication is allowed between modules. The local behavior of the i^{th} module is modeled by a fuzzy automaton G_i where $i \in \{1, ..., n\}$. The composite plant represents the global behavior of all modules and is modeled by the parallel composition of all fuzzy automata (i.e., $G_1||...||G_n$). Figure 5.1 shows the fuzzy automata G_1 and G_2, which model the local behaviors of two such modules.

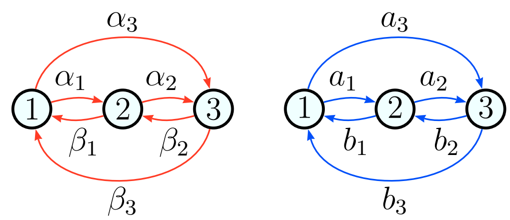


Figure 5.1: Fuzzy automata G_1 and G_2

Figure 5.2 shows the corresponding parallel composition used to model the behavior of the composite plant where $n = 2$.

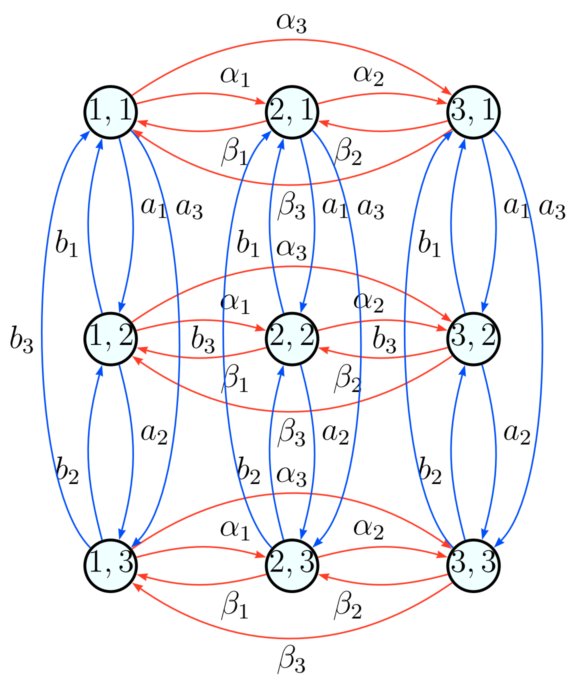


Figure 5.2: Fuzzy automaton $G_1 \parallel G_2$

As the centralized supervisory control of the composite system increases the complexity of supervisor synthesis, decentralized modular supervisory control is preferred. Let Σ_i be the set of fuzzy events of G_i and Σ be the set of fuzzy events of $G_1||...||G_n$. It is known that $\Sigma = \Sigma_1 \cup ... \cup \Sigma_n$. Note that the natural projection is defined here as $P_i : \Sigma^* \rightarrow \Sigma_i^*$ (i.e., each local supervisor sees its own set of fuzzy events).

By extending the decentralized modular control for crisp DES discussed in [99,100] to FDES, we can derive the fuzzy language generated by the composite plant as a representation of the fuzzy languages generated by each local plant.

$$L_{G_1||...||G_n} = P_1^{-1}(L_{G_1}) \cap ... \cap P_n^{-1}(L_{G_n}) \tag{5.1}$$

Note that the inverse projection of a fuzzy language L (i.e., $P^{-1}(L)$) is a set of fuzzy languages that the natural projection of each of them is L (respective to P). Let $P^{-1}(L)(s)$ be the possibility of the fuzzy string s belongs to the inverse projection of L. Then, (5.1) implies following equality for any $s \in \Sigma^*$.

$$L_{G_1||...||G_n}(s) = P_1^{-1}(L_{G_1})(s)\tilde{\cap}...\tilde{\cap}P_n^{-1}(L_{G_n})(s) \tag{5.2}$$

The possibility of a fuzzy string s belongs to the fuzzy language generated by the composite plant is equal to the minimum of the possibilities of s belong to the inverse projections of each fuzzy language.

5.1.1 Separability of a fuzzy language

Extending the separability of the languages in crisp DES presented in [100, 101], we can provide the following definition for the separability of fuzzy languages in FDES.

Definition 5.1: Let $\Sigma = \Sigma_1 \cup ... \cup \Sigma_n$. Then a fuzzy language $\bar{k}(\subseteq \Sigma^*)$ is said to be *separable* with respect to $\Sigma_1, ..., \Sigma_n$, if there exists a set of fuzzy languages $\bar{k}_i(\subseteq \Sigma_i^*)$ where $i \in \{1, ..., n\}$, for any $s \in \Sigma^*$, such that:

$$\bar{k}(s) = P_1^{-1}(\bar{k}_1)(s)\tilde{\cap}...\tilde{\cap}P_n^{-1}(\bar{k}_n)(s). \tag{5.3}$$

This says that if a fuzzy language is separable then its prefix-closure can be decomposed to several prefix-closed sub languages. Note that in DES separability of a crisp language does not necessarily imply that its prefix-closure is also decomposable to several prefix-closed sub languages (refer to Lemma 7 in [100]). But in FDES, as it is required to specify a fuzzy language in its prefix-closure form, the notion of separability is described using prefix-closed fuzzy languages.

Example 5.1: Let $n = 2$, $\Sigma_1 = \{\alpha, \beta\}$, $\Sigma_2 = \{a, b\}$. Assume the followings.

$\bar{k}_1 = \frac{1}{\varepsilon} + \frac{0.7}{\alpha} + \frac{0.5}{\beta}$

$\bar{k}_2 = \frac{1}{\varepsilon} + \frac{0.3}{a} + \frac{0.6}{b}$

$\bar{k} = \frac{1}{\varepsilon} + \frac{0.7}{\alpha} + \frac{0.5}{\beta} + \frac{0.3}{a} + \frac{0.6}{b} + \frac{0.3}{\alpha a} + \frac{0.3}{a\alpha} + \frac{0.3}{\beta a} + \frac{0.3}{a\beta} + \frac{0.6}{\alpha b} + \frac{0.6}{b\alpha} + \frac{0.5}{\beta b} + \frac{0.5}{b\beta}$

We can verify that the language k is separable with respect to $\{\Sigma_1, \Sigma_2\}$. The distributed local knowledge of $\bar{k}$ at i^{th} module is represented by $P_i(\bar{k})$. The following statements are straightforward.

$P_1(\bar{k}) = \frac{1}{\varepsilon} + \frac{0.7}{\alpha} + \frac{0.5}{\beta} + \frac{0.3}{\alpha} + \frac{0.3}{\beta} + \frac{0.6}{\alpha}$

$P_2(\bar{k}) = \frac{1}{\varepsilon} + \frac{0.3}{a} + \frac{0.6}{b} + \frac{0.5}{b}$

Assume $P_i(\bar{k})(s)_{max}$ represents the highest degree of the possibility of s belonging to $P_i(\bar{k})$. For example:

$P_1(\bar{k})(\alpha)_{max} = 0.7$, $P_1(\bar{k})(\beta)_{max} = 0.5$, $P_2(\bar{k})(a)_{max} = 0.3$, $P_2(\bar{k})(b)_{max} = 0.6$.

Note that $\bar{k}_i = P_i(\bar{k})$ when the condition $\bar{k}_i(s) = P_i(\bar{k})(s)_{max}$ holds. Then separability indicates that the global fuzzy language specification $\bar{k}$ can be recovered by combining all local fuzzy language specifications given by $\bar{k}_1 ... \bar{k}_n$ corresponding to FDES modules $G_1 ... G_n$, when the condition $\bar{k}_i(s) = P_i(\bar{k})(s)_{max}$ holds.

Definition 5.2: Consider $\bar{k}_i \subseteq \Sigma_i^*$. Then a fuzzy language $\bar{k}$ is said to be *separably-controllable-observable* with respect to $\cup_i^n \Sigma_i$ if the following conditions are satisfied.

1. $\bar{k}$ is separable with respect to $\Sigma_1, ..., \Sigma_n$.
2. $\bar{k}_i$ is fuzzy controllable.
3. $\bar{k}_i$ is fuzzy observable.

Assume a global fuzzy language specification $\bar{k}$ for composite plant $G_1||...||G_n$ is given and we want to verify whether there exists a set of modular partial observation supervisors $S_1^P, ..., S_n^P$ for concurrent modules modeled by $G_1, ..., G_n$, such that $L_{(S_1^P/G_1)||...||(S_n^P/G_n)}(s) = \bar{k}(s)$. *Theorem 5.1* is presented for this purpose.

5.1.2 A Modular supervisory control theory of FDES

Theorem 5.1: Modular partial observation supervisory control theorem for FDES.

Assume a system having concurrently operating FDES modules $G_1, ..., G_n$ with local fuzzy event sets $\Sigma_1, ..., \Sigma_n$ with respective local projections $P_1, ..., P_n$ and a set of local fuzzy language specifications $\bar{k}_i$ ($\bar{k}_i \subseteq \Sigma_i^*$). Let $\Sigma = \Sigma_1 \cup ... \cup \Sigma_n$ and the global fuzzy language specification be $\bar{k}$ ($\bar{k} \subseteq \Sigma^*$, $k \neq \emptyset$). Also assume the fuzzy events in Σ_i are partially observable to S_i^P.

There exists a set of modular partial observation supervisors $S_1^P, ..., S_n^P$ such that for any $s \in \Sigma^*$: $L_{(S_1^P/G_1)||...||(S_n^P/G_n)}(s) = \bar{k}(s)$ and $\bar{L}_{(S_1^P/G_1^P)||...||(S_n^P/G_n),m}(s) = \bar{k}(s)$, if and only if the following conditions hold.

1. $\bar{k}$ is separably-controllable-observable.
2. $\bar{k}_i$ is $L_{G_i,m}$-closed where $i \in \{1, ..., n\}$.

Proof: See appendix.

5.2 Summary

In this chapter, we have discussed the decentralized modular control of concurrent FDES. The notion of *separability* was introduced for fuzzy languages and the concept of separably-controllable-observability was also introduced as an existence condition for modular decentralized fuzzy supervisors. Associating those definitions, a modular supervisory control theory of FDES was established.

Chapter 6

Hierarchical control of FDES

In this chapter, a hierarchical supervisory control theory of FDES is established. This helps to resolve the vertical control complexities present in large-scale FDES. The hierarchical supervisory control architecture consists of multi-level supervisors assigned to the detailed low-level and the abstract high-level models of the plant. The system requirements are specified at the high-level and implemented at the low-level. Some important definitions and algorithms are introduced to preserve the hierarchical consistency between low-level and high-level models. Finally, using the established hierarchical supervisory control theory of FDES, a behavior-based mobile robot navigation example is discussed.

6.1 The hierarchical control architecture of FDES

Figure 6.1 shows the two-level hierarchical supervisory control architecture of FDES, which is adopted from its crisp DES version in [102]. The low-level FDES G_{lo} is supervised by the low-level controller S_{lo}. Assume G_{hi} as a high-level FDES, which represents the abstract simplified specification of G_{lo}. Control is virtually exercised on G_{hi} by the high-level supervisor S_{hi} through the channel Con_{hi}. Corresponding commands are generated and passed to S_{lo} from S_{hi} through the command channel Com_{hi-lo} to control G_{lo}. The low-level control is exerted on G_{lo} through the channel Con_{lo} and S_{lo} is informed about the results via the channel Inf_{lo}. The high-level abstraction G_{hi} is entirely driven by G_{lo} and updated via the information channel Inf_{lo-hi} whenever necessary. Finally, the summary of the implemented control actions is reported to S_{hi} via Inf_{hi}.

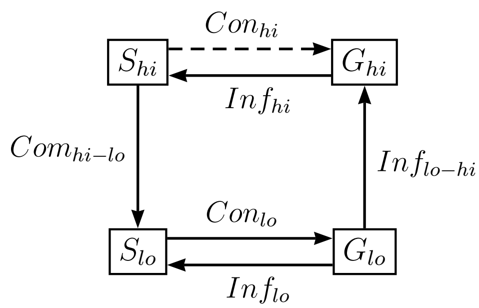


Figure 6.1: A two-level hierarchical supervisory control system

Assume Σ and T are as the sets of fuzzy events of G_{lo} and G_{hi} respectively. Let the fuzzy languages generated from G_{lo} (i.e., behavior of G_{lo}) and G_{hi} be L_{lo} and L_{hi} respectively. To model the low-level to high-level information flow Inf_{lo-hi}, we can define the *prefix-preserving* map $\theta : L_{lo} \to T^*$ with the following properties as in crisp DES [102].

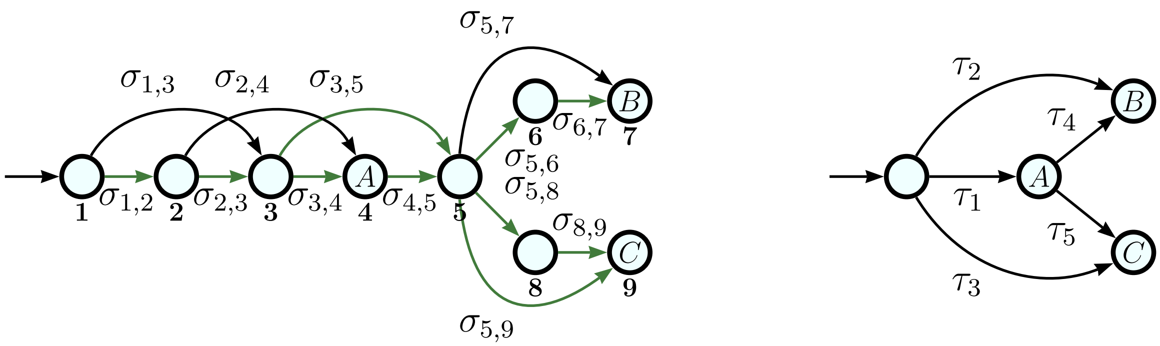
1. $\theta(\varepsilon) = \varepsilon$

2. For all $s \in L_{lo}$ and $\sigma \in \Sigma$:

$$\theta(s\sigma) = \begin{cases} \text{either } \theta(s) \\ \text{or } \theta(s)\,\tau, \text{ for some } \tau \in T. \end{cases}$$

Consider a low-level FDES G_{lo} and its high-level FDES G_{hi}. Assume their corresponding fuzzy automata G_{lo} and G_{hi} are as shown in figure 6.2(a) and 6.2(b) respectively.

A fuzzy state in G_{lo}, which generates an alphabet, is defined as a *vocal node*. Otherwise, it is a *silent node*. The fuzzy states 4, 7, and 9 of G_{lo} shown in figure 6.2(a) are vocal nodes and the others are silent. There exist multiple paths in G_{lo} to achieve the fuzzy states 4 (A), 7 (B) and 9 (C). As a result, we can generate several possible high-level fuzzy events to indicate the transitions to the above fuzzy states as shown in figure 6.2(b).



(a) Fuzzy automaton G_{lo} (b) Fuzzy automaton G_{hi}

Figure 6.2: Fuzzy automata models of low-level and high-level FDES

6.1.1 Computation of *main-path*

Each high-level fuzzy event corresponds to an occurrence of a vocal node, and in the low-level system this occurs through a set of fuzzy strings. For example in figure 6.2(a) the high-level fuzzy event τ_1 can be generated from three possible low-level fuzzy strings: $\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}$, $\sigma_{1,2}\sigma_{2,4}$ or $\sigma_{1,3}\sigma_{3,4}$. Hence, the following definition is provided.

Definition 6.1: Define the *main-path*, $L_{lo}^{MP} \subseteq L_{lo}$, of G_{lo}, such that it consists of low-level fuzzy strings that achieve each high-level fuzzy string $t \in L_{hi}$ with the above mapping θ according to **Algorithm 1**.

Each low-level fuzzy string in L_{lo}^{MP} contributes to generate the high-level specification with the mapping θ. Assume $\theta(L_{lo})$ has g number of high-level fuzzy strings.

```
input  : Σ, L_lo, θ
output: L_lo^MP
for ∀ t ∈ θ(L_lo) do
    Let θ^{-1}(t) = s_1, ..., s_n. Define M_s^t as an empty set: M_s^t = ∅.
    for i ∈ {1, ..., n} do
        Let s_i = σ_{i,1}...σ_{i,m} // m depends on |s_i|.
        Define L_{lo,uc} such that L_{lo,uc}(s_i) = min{Σ_uc(σ_{i,1}), ..., Σ_uc(σ_{i,m})}.
        If L_{lo,uc}(s_i) = 0 : M_s^t ← s_i.
    end
    Select all s_k, L_{lo,uc}(s_k) > 0, (1 ≤ k ≤ n) such that ∀ j ∈ {1, ..., n}:
    L_{lo,uc}(s_k) ∩̃ L_lo(s_k) ≥ L_{lo,uc}(s_j) ∩̃ L_lo(s_j).
    M_s^t ← s_k
    Let |M_s^t| = p, p ≥ 1 // M_s^t has p number of strings.
    Reassign strings in M_s^t such that ∀ l ∈ {1, ..., p}: s_l ∈ M_s^t.
    Select s_q, (1 ≤ q ≤ p) such that ∀ l ∈ {1, ..., p}: L_lo(s_q) ≥ L_lo(s_l).
    L_lo^MP ← s_q
end
```

Algorithm 1: The computation of main path L_{lo}^{MP}

Then, the computational complexity of **Algorithm 1** will be $\mathcal{O}(g \times n \times m)$.

Example 6.1: Consider G_{lo} in figure 6.2(a). Assume the degrees of the low-level fuzzy events being uncontrollable are as follows.

$\Sigma_{uc}(\sigma_{1,2}) = 0.2$,$\Sigma_{uc}(\sigma_{2,3}) = 0.3$, $\Sigma_{uc}(\sigma_{3,4}) = 0.4$, $\Sigma_{uc}(\sigma_{4,5}) = 0.1$, $\Sigma_{uc}(\sigma_{5,6}) = 0.3$, $\Sigma_{uc}(\sigma_{6,7}) = 0.5$, $\Sigma_{uc}(\sigma_{5,8}) = 0.1$, $\Sigma_{uc}(\sigma_{8,9}) = 0.4$, $\Sigma_{uc}(\sigma_{1,3}) = 0.1$, $\Sigma_{uc}(\sigma_{2,4}) = 0.3$, $\Sigma_{uc}(\sigma_{3,5}) = 0.4$, $\Sigma_{uc}(\sigma_{5,7}) = 0.4$, and $\Sigma_{uc}(\sigma_{5,9}) = 0.3$.

Let the physical possibilities of the low-level fuzzy strings be given as follows.

$L_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}) = 0.9$, $L_{lo}(\sigma_{1,2}\sigma_{2,4}) = 0.5$, $L_{lo}(\sigma_{1,3}\sigma_{3,4}) = 0.7$. $L_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,5}\sigma_{5,6}\sigma_{6,7}) = 0.8$, $L_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,5}\sigma_{5,7}) = 0.6$, $L_{lo}(\sigma_{1,3}\sigma_{3,5}\sigma_{5,6}\sigma_{6,7}) = 0.4$, $L_{lo}(\sigma_{1,3}\sigma_{3,5}\sigma_{5,7}) = 0.5$. $L_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,5}\sigma_{5,8}\sigma_{8,9}) = 0.7$, $L_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,5}\sigma_{5,9}) = 0.4$, $L_{lo}(\sigma_{1,3}\sigma_{3,5}\sigma_{5,8}\sigma_{8,9}) = 0.8$, $L_{lo}(\sigma_{1,3}\sigma_{3,5}\sigma_{5,9}) = 0.7$. $L_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,6}\sigma_{6,7}) = 0.7$, $L_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,7}) = 0.6$, $L_{lo}(\sigma_{1,2}\sigma_{2,4}\sigma_{4,5}\sigma_{5,6}\sigma_{6,7}) = 0.4$, $L_{lo}(\sigma_{1,2}\sigma_{2,4}\sigma_{4,5}\sigma_{5,7}) = 0.3$, $L_{lo}(\sigma_{1,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,6}\sigma_{6,7}) = 0.5$, $L_{lo}(\sigma_{1,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,7}) = 0.4$. $L_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,8}\sigma_{8,9}) = 0.8$, $L_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,9}) = 0.6$, $L_{lo}(\sigma_{1,2}\sigma_{2,4}\sigma_{4,5}\sigma_{5,8}\sigma_{8,9}) = 0.5$, $L_{lo}(\sigma_{1,2}\sigma_{2,4}\sigma_{4,5}\sigma_{5,9}) = 0.4$, $L_{lo}(\sigma_{1,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,8}\sigma_{8,9}) =$

0.5, and $L_{lo}(\sigma_{1,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,9}) = 0.6$.

Note that in practical situations these possibilities represent constraints in the physical systems. The main-path, L_{lo}^{MP} is computed for G_{lo} using **Algorithm 1**. This is colored green in the figure 6.2(a) and given as follows.

$$L_{lo}^{MP} = \{(\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}), (\sigma_{1,2}\sigma_{2,3}\sigma_{3,5}\sigma_{5,6}\sigma_{6,7}), (\sigma_{1,2}\sigma_{2,3}\sigma_{3,5}\sigma_{5,9}), (\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,6}\sigma_{6,7}),$$
$$(\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,8}\sigma_{8,9})\}$$

The physical possibility of a high-level fuzzy string t can be defined according to that of low-level fuzzy strings s in L_{lo}^{MP}, which generate t with the mapping θ.

$L_{hi}(\varepsilon) = 1$

$L_{hi}(t) = L_{lo}(s)$, such that $s \in L_{lo}^{MP}$ and $\theta(s) = t$.

Example 6.2: Refer to G_{lo} shown in Figure 6.2(a). Let τ_1, τ_2, τ_3, $\tau_1\tau_4$, $\tau_1\tau_5 \in L_{hi}$. With the above definition, the calculated physical possibilities of high-level fuzzy strings are:

$L_{hi}(\tau_1) = 0.9$, $L_{hi}(\tau_2) = 0.8$, $L_{hi}(\tau_3) = 0.4$, $L_{hi}(\tau_1\tau_4) = 0.7$, and $L_{hi}(\tau_1\tau_5) = 0.8$.

6.1.2 The ***output-control-consistency*** of FDES

As each fuzzy event in G_{lo} is generally associated with a degree of being uncontrollable, $\tau \in T$ also possesses this property in the high-level. This leads to define the *output-control-consistency* condition for FDES as in *Definition 6.2.*

Definition 6.2: Consider a hierarchical system having a low-level FDES G_{lo} and its high-level abstraction G_{hi}. Assume the set of fuzzy events of G_{lo} and G_{hi} are Σ and T respectively. Let the fuzzy languages generated by G_{lo} and G_{hi} be L_{lo} and L_{hi} respectively. Assume the high-level fuzzy event τ is generated by the low-level fuzzy string s through the main-path with mapping θ (i.e., $\theta(s) = \tau$) where s is located between two consecutive vocal nodes. Let $s = \sigma_1 ... \sigma_n$. Then, G_{lo} is said to be *output-control-consistent* if the following condition holds.

$$T_{uc}(\tau) = \min\{\Sigma_{uc}(\sigma_1), ..., \Sigma_{uc}(\sigma_n)\} \tag{6.1}$$

Example 6.3: Refer to G_{lo} in figure 6.2(a). Assume G_{lo} is output-control-consistent. Then, the degrees of the high-level fuzzy events $\tau_i \in T$ $(i = 1, 2, 3, 4, 5)$ being uncontrollable can be calculated as follows.

$T_{uc}(\tau_1) = \min\{\Sigma_{uc}(\sigma_{1,2}), \Sigma_{uc}(\sigma_{2,3}), \Sigma_{uc}(\sigma_{3,4})\} = 0.2$

$T_{uc}(\tau_2) = \min\{\Sigma_{uc}(\sigma_{1,2}), \Sigma_{uc}(\sigma_{2,3}), \Sigma_{uc}(\sigma_{3,5}), \Sigma_{uc}(\sigma_{5,6}), \Sigma_{uc}(\sigma_{6,7})\} = 0.2$

$T_{uc}(\tau_3) = \min\{\Sigma_{uc}(\sigma_{1,2}), \Sigma_{uc}(\sigma_{2,3}), \Sigma_{uc}(\sigma_{3,5}), \Sigma_{uc}(\sigma_{5,9})\} = 0.2$

$T_{uc}(\tau_4) = \min\{\Sigma_{uc}(\sigma_{4,5}), \Sigma_{uc}(\sigma_{5,6}), \Sigma_{uc}(\sigma_{6,7})\} = 0.1$

$T_{uc}(\tau_5) = \min\{\Sigma_{uc}(\sigma_{4,5}), \Sigma_{uc}(\sigma_{5,8}), \Sigma_{uc}(\sigma_{8,9})\} = 0.1$

Note that because of these degrees of being uncontrollable, any high-level fuzzy event in G_{hi} cannot be completely disabled by low-level supervisory control of G_{lo}. Only partial disablement of high-level fuzzy events is possible.

6.1.3 Calculation of a controllable sub language

Assume G_{lo} is output-control-consistent. Let the high-level specification language of G_{hi} be $\bar{k}_{hi}$ $(\bar{k}_{hi} \subseteq L_{hi})$, which is prefix-closed and fuzzy controllable with respect to L_{hi}. The inverse map $\theta^{-1}(\bar{k}_{hi})$ generates the corresponding prefix-closed low-level fuzzy specification language $\bar{k}_{lo} \subseteq L_{lo}$ of G_{lo}.

As in crisp DES, in FDES also the derived low-level specification $\bar{k}_{lo}$ for G_{lo} is not necessarily fuzzy controllable. Let $\theta^{-1}(\tau)_m$ represent the language of marked low-level fuzzy strings, which generate τ in high-level with the mapping θ. A language m_{lo}, which consists of all such low-level fuzzy strings can be constructed as follows.

Let $t \in \bar{k}_{hi}$, $\forall\ \tau \in T$, where $t\tau \in \bar{k}_{hi}$: $\theta^{-1}(\tau)_m \subseteq m_{lo}$.

Assume $\bar{k}_{lo}^{\uparrow}$ represents the supremal fuzzy controllable prefix-closed sub language of $\bar{k}_{lo}$. The computation of $\bar{k}_{lo}^{\uparrow}$ can be performed using the following steps.

1. If $\bar{k}_{lo}$ contains a fuzzy string $s = \sigma_1..\sigma_k$, such that for some i ($1 \leq i \leq k$), $\Sigma_c(\sigma_i) = 1$ and $s \in (L_{lo} \backslash \bar{k}_{lo})/\Sigma_{uc}^*$ (refer to the quotient operation in [1]) then any string in $\bar{k}_{lo}$, which contains s as a prefix must be removed by complete disablement of σ_i (the same as in crisp DES).

2. With the remaining fuzzy strings, $\bar{k}_{lo}^{\uparrow}$ can be achieved using **Algorithm 2**.

input : $T, L_{hi}, \bar{k}_{hi}, \Sigma, L_{lo}, m_{lo}$
output: $\bar{k}_{lo}^{\uparrow}$

for $\forall\ \tau \in T$ **do**
  Let $t \in \bar{k}_{hi}$ and $\eta = T_{uc}(\tau) \tilde{\cap} L_{hi}(t\tau)$. Compute the degree of τ being enabled by S_{hi} as follows:

$$S_{hi,t}(\tau) = \begin{cases} \eta, & \text{if } \eta \geq \bar{k}_{hi}(t\tau) \\ \bar{k}_{hi}(t\tau), & \text{otherwise.} \end{cases}$$

  Assign $m_{lo}\left(\theta^{-1}(\tau)_m\right) = S_{hi,t}(\tau)$.
end
Let $|m_{lo}| = n$ `//` m_{lo} `has` n `number of strings.`
Let $r \in L_{lo}^{MP}$ such that $\theta(r) = t$ `//` r `is in the main path.`
for $s_i \in m_{lo}, i \in \{1, ..., n\}$ **do**
  Let $s_i = \sigma_0 ... \sigma_k$, $\sigma_0 = \varepsilon$, and $\bar{k}_{lo}^{\uparrow}(\varepsilon) = 1$.
  for $\sigma_j, j \in (1, ..., k)$ **do**
    Let $\zeta = \Sigma_{uc}(\sigma_j) \tilde{\cap} L_{lo}(r\sigma_j)$.
    Calculate the feasibility of σ_j being available in s_i, $F_{s_i}(\sigma_j)$ as follows:

$$F_{s_i}(\sigma_j) = \begin{cases} \zeta, & \text{if } \zeta \geq m_{lo}(s_i) \\ m_{lo}(s_i), & \text{otherwise.} \end{cases}$$

    $\bar{k}_{lo}^{\uparrow}(r\sigma_j) = \bar{k}_{lo}^{\uparrow}(r\sigma_{j-1}) \tilde{\cap} F_{s_i}(\sigma_j) \tilde{\cap} L_{lo}(r\sigma_j)$
  end
end

Algorithm 2: The computation of $\bar{k}_{lo}^{\uparrow}$

Assume T has h number of high-level fuzzy events. Then, the computational complexity of **Algorithm 2** can be calculated as $\mathcal{O}(h + n \times k)$.

Example 6.4: Assume we want to enforce the occurrence of high-level fuzzy event $\tau_1 \in \bar{k}_{hi}$ in figure 6.2(b) while keeping the possibilities of the other high-level fuzzy events (or strings) of occurring at their minimum levels, which are equal to their degrees of being uncontrollable. There exist three different low-level fuzzy string paths for achieving τ_1.

$\theta^{-1}(\tau_1) = \{\varepsilon, \sigma_{1,2}, \sigma_{1,2}\sigma_{2,3}, \sigma_{1,2}\sigma_{2,3}\sigma_{3,4}, \sigma_{1,2}\sigma_{2,4}, \sigma_{1,3}, \sigma_{1,3}\sigma_{3,4}\}$

$\theta^{-1}(\tau_1)_m = \{\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}, \sigma_{1,2}\sigma_{2,4}, \sigma_{1,3}\sigma_{3,4}\} \subseteq m_{lo}$

Let $L_{lo}(\sigma_{1,2}) = L_{lo}(\sigma_{1,2}\sigma_{2,3}) = L_{lo}(\sigma_{1,3}) = 1$. Note that $L_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}) = 0.9$, $L_{lo}(\sigma_{1,2}\sigma_{2,4}) = 0.5$, and $L_{lo}(\sigma_{1,3}\sigma_{3,4}) = 0.7$. Let $\bar{k}_{hi}(\tau_1) = 0.7 \rightarrow S_{hi,\varepsilon}(\tau_1) = 0.7$.

Then, $m_{lo}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}, \sigma_{1,2}\sigma_{2,4}, \sigma_{1,3}\sigma_{3,4}) = 0.7$ implies that:

$k_{lo}^{\uparrow}(\sigma_{1,2}) = k_{lo}^{\uparrow}(\sigma_{1,2}\sigma_{2,3}) = \bar{k}_{lo}^{\uparrow}(\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}) = 0.7$, $k_{lo}^{\uparrow}(\sigma_{1,2}\sigma_{2,4}) = 0.5$, and $k_{lo}^{\uparrow}(\sigma_{1,3}) = k_{lo}^{\uparrow}(\sigma_{1,3}\sigma_{3,4}) = 0.7$.

It is true that $\bar{k}_{lo}^{\uparrow} \subseteq \bar{k}_{lo}$ and $\bar{k}_{lo}^{\uparrow}(s) \leq \bar{k}_{lo}(s)$. As a result, $\theta(\bar{k}_{lo}^{\uparrow}) \subseteq \bar{k}_{hi}$ and $\theta(\bar{k}_{lo}^{\uparrow})(t) \leq \bar{k}_{hi}(t)$. Hence, the same as in the crisp DES case, only a sub set of the high-level specification can be achieved by low-level supervisory control of FDES.

6.1.4 The *strictly-output-control-consistency* of FDES

Assume the high-level fuzzy string $\tau_1\tau_5$ is desired to occur in G_{hi}, whereas $\tau_1\tau_4$ is undesired. In an attempt to increase the possibility of the occurrence of $\tau_1\tau_5$ indeed requires increasing the possibility of the low-level fuzzy event $\sigma_{4,5}$ of being enabled to a necessary extent. As a result, the possibilities of low-level fuzzy strings $\sigma_{4,5}\sigma_{5,7}$ and $\sigma_{4,5}\sigma_{5,6}\sigma_{6,7}$ occurring in G_{lo} will also be increased. Consequently, the possibility of $\tau_1\tau_4$ occurring in G_{hi} will be increased. This leads to define the *strictly-output-control-consistency* of FDES.

Definition 6.3: Consider a hierarchical system having a low-level FDES G_{lo} and its high-level abstraction G_{hi}. If G_{lo} is output-control consistent, and if increasing the possibility of the occurrence of a desired high-level fuzzy event (by controlling its corresponding low-level fuzzy string) does not increase the possibility of the occurrence of an undesired high-level fuzzy event (more than its degree of being uncontrollable), then G_{lo} is said to be *strictly-output-control-consistent*.

The property of strictly-output-control-consistency of FDES implies that the low-level model allows the enabling/disabling of each high-level fuzzy event individually according to its degree of being controllable. Consider the following example.

Example 6.5: Consider *Example 6.3* and figure 6.2. Assume the desired and undesired high-level fuzzy strings are $\tau_1\tau_5$ and $\tau_1\tau_4$ respectively. Here G_{lo} is not strictly-output-control-consistent as it increases the possibility of the occurrence of $\tau_1\tau_4$ in an attempt to increase that of $\tau_1\tau_5$. A cure is to break down G_{lo} and assign vocal nodes to preserve the strictly-output-control-consistency. Let us assign the state 5 of G_{lo} in figure 6.2(a) as a vocal node "X". This modifies the high-level abstraction G_{hi} to G'_{hi} as shown in figure 6.3. The new high-level fuzzy event and state are colored brown in the figure.

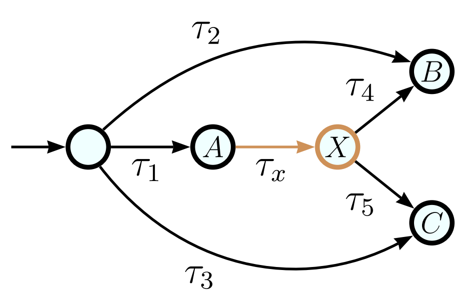


Figure 6.3: Modified fuzzy automaton G'_{hi}

As a result, L_{hi} has more high-level strings: $L_{hi} = \{\tau_1, \tau_2, \tau_3, \tau_1\tau_x, \tau_1\tau_x\tau_4, \tau_1\tau_x\tau_5\}$.

When the low-level FDES satisfies the strictly-output-control-consistency condition we say the low-level FDES is *hierarchically consistent* with its high-level abstraction. This means that the high-level specification can be precisely achieved by the low-level supervisory control. Note that in this case $\theta(\bar{k}_{lo}^{\uparrow}) = \bar{k}_{hi}$ and $\theta(\bar{k}_{lo}^{\uparrow})(t) = \bar{k}_{hi}(t)$ for $t \in T^*$. If the low-level model is not strictly output control consistent, then $\theta(\bar{k}_{lo}^{\uparrow})(t) \neq \bar{k}_{hi}(t)$.

6.1.5 Dealing with the unobservability of low-level fuzzy events

When unobservability is associated with the low-level fuzzy events of G_{lo}, the high-level specification language $\bar{k}_{hi}$ of G_{hi} has to incorporate the resulting high-level fuzzy events and strings with their relevant degrees. This leads to extend the *H-observability* of crisp DES in [103] to FDES as follows.

Definition 6.4: Consider a hierarchical system having a low-level FDES G_{lo}, with its fuzzy controllable and fuzzy observable specification $\bar{k}'_{lo} \subseteq L_{lo}$, and a high-level FDES G_{hi}, with its specification language $\bar{k}_{hi} \subseteq L_{hi}$, Let $s, s' \in \bar{k}'_{lo}$ where $s' \in P^{-1}[P(s)]$. Also, $\theta(s) = t, \theta(s') = t'$, where $t, t' \in \bar{k}_{hi}$. Then, the high-level specification language $\bar{k}_{hi}$ is said to be *H-fuzzy observable* with respect to $\bar{k}'_{lo}$ and G_{hi} for all $\tau \in T$, if the following inequality holds.

$$\bar{k}_{hi}(t\tau) \cap \bar{k}_{hi}(t') \cap L_{hi}(t'\tau) \cap T_c(\tau) \cap P^{-1}[P(s)](s') \leq \bar{k}_{hi}(t'\tau) \tag{6.2}$$

Intuitively, this means that the possibility of fuzzy string $t'\tau$ belongs to prefix-closed high-level specification $\bar{k}_{hi}$ is greater than or equal to the minimum of the possibility of $t\tau$ belongs to $\bar{k}_{hi}$ and the possibility of t' belongs to $\bar{k}_{hi}$, together with the physical possibility of $t'\tau$ occurring in high-level, the degree of the high-level fuzzy event τ being controllable and the possibility of $P(s')$ to be seen as the same as $P(s)$.

Remark 6.1: Extending from the crisp DES version to FDES, let G_{lo} be output-control consistent, the high-level specification language be $\bar{k}_{hi} \subseteq L_{hi}$ and the low-level specification be $\bar{k}_{lo}$ where $\bar{k}_{lo} = \theta^{-1}(\bar{k}_{hi})$. If $\bar{k}_{lo}$ is fuzzy controllable with respect to L_{lo}, then $\bar{k}_{hi}$ is also fuzzy controllable with respect to L_{hi}.

6.1.6 A Hierarchical supervisory control theory of FDES

The following theorem is presented for verifying the existence of a low-level supervisor to achieve high-level specification.

Theorem 6.1: Hierarchical supervisory control theory of FDES with partial observation.

Assume a hierarchical system having a low-level FDES G_{lo} and its high-level abstraction G_{hi}. Let G_{lo} be output-control-consistent with respect to G_{hi}. Assume $\bar{k}_{hi}$ as the prefix-closed fuzzy controllable high-level specification of G_{hi}. Furthermore, let $\bar{k}_{lo} = \theta^{-1}(\bar{k}_{hi})$ be the corresponding prefix-closed low-level fuzzy specification language of G_{lo} and $\bar{k}_{lo}^{\uparrow}$ represent the supremal prefix-closed fuzzy sub language of $\bar{k}_{lo}$, which is fuzzy controllable. Then, under the foregoing assumptions there exists a low-level supervisor S_{lo} for G_{lo} such that $L_{S_{lo}/G_{lo}}(s) = \bar{k}_{lo}^{\uparrow}(s)$ and $\theta(\bar{k}_{lo}^{\uparrow}) = \bar{k}_{hi}$, if the following conditions hold.

1. $\bar{k}_{lo}^{\uparrow}$ is $L_{lo,m}$-closed.
2. G_{lo} is strictly-output-control-consistent.
3. $\bar{k}_{lo}^{\uparrow}$ is fuzzy observable with respect to L_{lo}, P and Σ_c.
4. $\bar{k}_{hi}$ is H-fuzzy observable with respect to $\bar{k}_{lo}^{\uparrow}$ and G_{hi}.

Proof: See appendix.

6.2 A mobile robot navigation example

We discuss a behavior-based mobile robot navigation example as a proof-of-concept application to the presented hierarchical supervisory control theory of FDES. Consider the environment shown in figure 6.4 where a mobile robot is required to perform certain tasks.

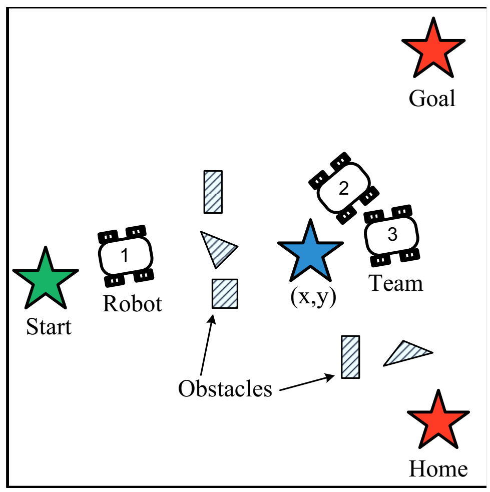


Figure 6.4: The environment for mobile robot navigation

The task breakdown can be summarized as:

Task 1: Robot 1 goes from Start to (x,y) avoiding the obstacles. After completing the task 1, robot 1 can perform either of the following.

Task 2: At (x,y), robot 1 joins with its team members (2 and 3) and they navigate to Goal while keeping formation.

Task 3: From (x,y), robot 1 goes to Home avoiding obstacles.

A fuzzy behavior-based approach is adopted for implementing these sub tasks [82]. Task 1 is performed by using two behaviors: namely *Go-to-xy (Gxy)* and *Avoid-Obstacles (AO)*. Similarly, for task 2: *Go-to-Goal (GG)* and *Keep-Formation (KF)*, and for task 3: *Go-to-Home (GH)* and *Avoid-Obstacles (AO)* are assigned. A behavior is modeled by a simple fuzzy automaton. The behavior weightings are represented by the states of the fuzzy automaton [10]. Figure 6.5 shows the modeling of AO.

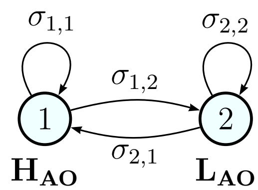


Figure 6.5: Fuzzy automaton modeling AO

The fuzzy states 1 and 2 represent the *high* ($\mathbf{H_{AO}}$) and *low* ($\mathbf{L_{AO}}$) states of behavior weightings. The fuzzy events, which are composed using sensory perceptions, model the transition possibilities of these behavior weightings.

By combining the environmental information and above behavior collections, a detailed fuzzy automaton G_{lo} can be constructed for modeling the robot tasks, which collectively represent the low-level FDES. This is shown in figure 6.6. This model facilitates smooth transition between behaviors without making the robot unstable.

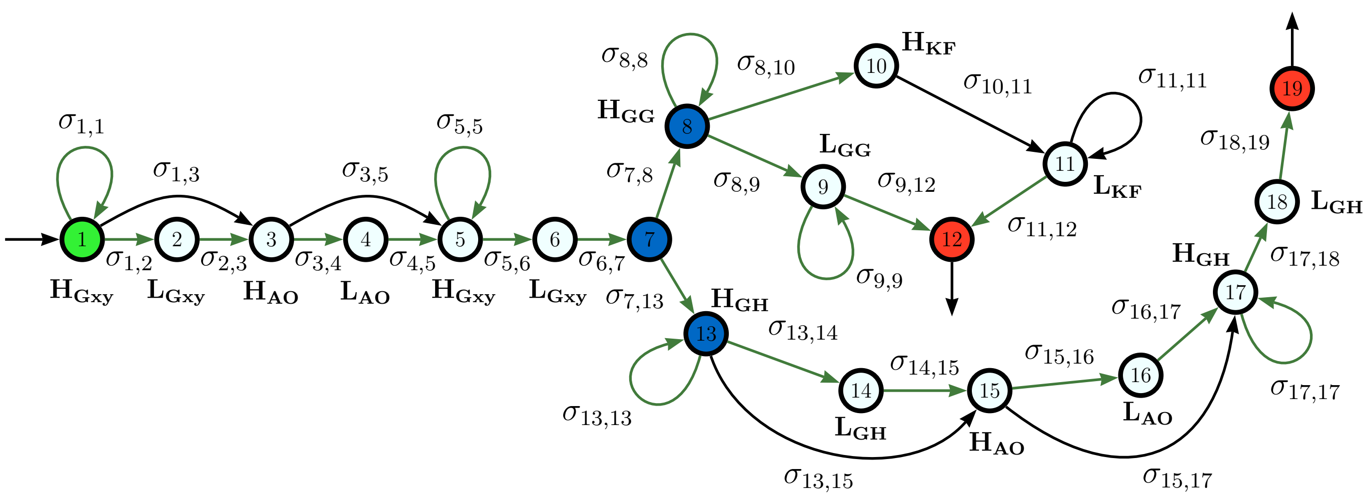


Figure 6.6: Fuzzy automaton G_{lo} modeling robot tasks

The fuzzy state 1 represents the *high* level of *Gxy* behavior ($\mathbf{H_{Gxy}}$) and fuzzy state 2 represents the *low* level of *Gxy* behavior ($\mathbf{L_{Gxy}}$) and so on. State 7 represents the completion of the task 1 (robot arriving to (x,y)). States 12 and 19 represent completion of task 2 and task 3 respectively.

Physical possibilities of fuzzy strings of G_{lo} are assigned in such a way that they reduce the instabilities caused by behavior transitions.

$L_{lo}(\sigma_{1,1}) = 1$, $L_{lo}(\sigma_{1,1}\sigma_{1,3}) = 0.5$, $L_{lo}(\sigma_{1,1}\sigma_{1,3}\sigma_{3,5}) = 0.3$, $L_{lo}(\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,5}) = 0.5$,

$L_{lo}(\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,8}\sigma_{8,8}\sigma_{8,10}) = 0.5$,

$L_{lo}(\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,13}\sigma_{13,13}\sigma_{13,15}) = 0.5$,

$L_{lo}(\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,13}\sigma_{13,13}\sigma_{13,15}\sigma_{15,17}) = 0.3$,

and $L_{lo}(\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,13}\sigma_{13,13}\sigma_{13,14}\sigma_{14,15}\sigma_{15,17}) = 0.5$.

For navigational safety, some degree of uncontrollability is assigned to each low-level fuzzy event, which controls the AO behavior.

$\Sigma_{uc}(\sigma_{1,1}) = 1$, $\Sigma_{uc}(\sigma_{1,2}) = 0.5$,$\Sigma_{uc}(\sigma_{2,3}) = 1$, $\Sigma_{uc}(\sigma_{1,3}) = 0.5$, $\Sigma_{uc}(\sigma_{3,4}) = 0.6$, $\Sigma_{uc}(\sigma_{4,5}) = 1$, $\Sigma_{uc}(\sigma_{3,5}) = 0.4$, $\Sigma_{uc}(\sigma_{5,5}) = 1$, $\Sigma_{uc}(\sigma_{5,6}) = 1$, $\Sigma_{uc}(\sigma_{6,7}) = 1$, $\Sigma_{uc}(\sigma_{7,8}) = 0$, $\Sigma_{uc}(\sigma_{8,8}) = 1$, $\Sigma_{uc}(\sigma_{8,9}) = 0.5$, $\Sigma_{uc}(\sigma_{9,9}) = 1$, $\Sigma_{uc}(\sigma_{9,12}) = 1$, $\Sigma_{uc}(\sigma_{8,10}) = 0.5$, $\Sigma_{uc}(\sigma_{10,11}) = 1$, $\Sigma_{uc}(\sigma_{11,11}) = 1$, $\Sigma_{uc}(\sigma_{11,12}) = 1$ $\Sigma_{uc}(\sigma_{7,13}) = 0$, $\Sigma_{uc}(\sigma_{13,13}) = 1$, $\Sigma_{uc}(\sigma_{13,14}) = 0.7$,$\Sigma_{uc}(\sigma_{14,15}) = 1$, $\Sigma_{uc}(\sigma_{13,15}) = 0.3$, $\Sigma_{uc}(\sigma_{15,16}) = 0.6$, $\Sigma_{uc}(\sigma_{16,17}) = 1$, $\Sigma_{uc}(\sigma_{15,17}) = 0.4$, $\Sigma_{uc}(\sigma_{17,17}) = 1$, $\Sigma_{uc}(\sigma_{17,18}) = 1$, and $\Sigma_{uc}(\sigma_{18,19}) = 1$.

Note that for this example if not specifically mentioned the physical possibility of any continuation of a fuzzy string is the same as that of the fuzzy string itself ($L_{lo}(s\sigma) = L_{lo}(s)$). Note that $\sigma_{7,8}$ and $\sigma_{7,13}$ act as *links* to task 2 and 3. For simplicity, assume each fuzzy event is completely observable.

Figure 6.7 shows the high-level abstraction G_{hi} of G_{lo}. The high-level fuzzy events τ_1, τ_3 and τ_5 model tasks 1, 2 and 3 respectively. Also, τ_2 and τ_4 represent low-level fuzzy events $\sigma_{7,8}$ and $\sigma_{7,13}$, which are completely controllable.

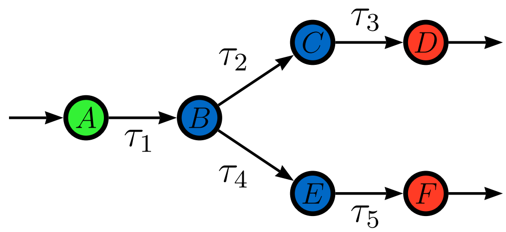


Figure 6.7: Fuzzy automaton G_{hi}

The following steps are performed in this example.

1. The main-path computation. According to **Algorithm 1** the main-path is computed for G_{lo} and colored green in figure 6.6.

2. Calculation of physical possibilities of high-level fuzzy strings using main-path. The results are shown as follows.

   $L_{hi}(\tau_1) = 1$, $L_{hi}(\tau_1\tau_2) = 1$, $L_{hi}(\tau_1\tau_2\tau_3) = 1$, $L_{hi}(\tau_1\tau_4) = 1$, and $L_{hi}(\tau_1\tau_4\tau_5) = 1$.

3. Assuming G_{lo} is output-control-consistent, calculation of the degrees of the high-level fuzzy events being uncontrollable. The results are shown as follows.

   $T_{uc}(\tau_1) = 0.5$, $T_{uc}(\tau_2) = 0$, $T_{uc}(\tau_3) = 0.5$, $T_{uc}(\tau_4) = 0$, and $T_{uc}(\tau_5) = 0.6$.

Assume the robot needs to perform tasks 1 and 2, which represent $\tau_1\tau_2\tau_3$ in high-level (The ABCD branch of G_{hi}). Let the controllable high-level specification be given as follows.

$$\bar{k}_{hi} = \frac{1}{\varepsilon} + \frac{1}{\tau_1} + \frac{1}{\tau_1\tau_2} + \frac{1}{\tau_1\tau_2\tau_3}$$

Based on this the supremal fuzzy controllable low-level specification $\bar{k}_{lo}^{\uparrow}$ is generated using **Algorithm 2** and considering all possible continuations of the low-level fuzzy strings.

$$
\begin{aligned}
\bar{k}_{lo}^{\uparrow} = {} & \frac{1}{\varepsilon} + \frac{1}{\sigma_{1,1}} + \frac{1}{\sigma_{1,1}\sigma_{1,2}} + \ldots + \frac{1}{\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,8}\sigma_{8,8}\sigma_{8,9}\sigma_{9,9}\sigma_{9,12}} + \\
& \frac{0.5}{\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,8}\sigma_{8,8}\sigma_{8,10}} + \ldots + \frac{0.5}{\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,8}\sigma_{8,8}\sigma_{8,10}\sigma_{10,11}\sigma_{11,11}\sigma_{11,12}} \\
& + \frac{0.5}{\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,5}} + \ldots + \frac{0.5}{\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,8}\sigma_{8,8}\sigma_{8,9}\sigma_{9,9}\sigma_{9,12}} + \ldots + \\
& \frac{0.5}{\sigma_{1,1}\sigma_{1,2}\sigma_{2,3}\sigma_{3,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,8}\sigma_{8,8}\sigma_{8,10}\sigma_{10,11}\sigma_{11,11}\sigma_{11,12}} + \frac{0.5}{\sigma_{1,1}\sigma_{1,3}} + \ldots + \\
& \frac{0.5}{\sigma_{1,1}\sigma_{1,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,8}\sigma_{8,8}\sigma_{8,9}\sigma_{9,9}\sigma_{9,12}} + \ldots + \frac{0.5}{\sigma_{1,1}\sigma_{1,3}\sigma_{3,4}\sigma_{4,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,8}\sigma_{8,8}\sigma_{8,10}\sigma_{10,11}\sigma_{11,11}\sigma_{11,12}} + \\
& \frac{0.3}{\sigma_{1,1}\sigma_{1,3}\sigma_{3,5}} + \ldots + \frac{0.3}{\sigma_{1,1}\sigma_{1,3}\sigma_{3,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,8}\sigma_{8,8}\sigma_{8,9}\sigma_{9,9}\sigma_{9,12}} + \ldots + \frac{0.3}{\sigma_{1,1}\sigma_{1,3}\sigma_{3,5}\sigma_{5,5}\sigma_{5,6}\sigma_{6,7}\sigma_{7,8}\sigma_{8,8}\sigma_{8,10}\sigma_{10,11}\sigma_{11,11}\sigma_{11,12}}
\end{aligned}
$$

Now a low-level supervisor S_{lo} can be assigned to achieve $\bar{k}_{lo}^{\uparrow}$ in G_{lo}, which leads to command fusion type behavior coordination in the robot [47].

6.3 Summary

In this chapter, we have established the hierarchical supervisory control theory of FDES. Some properties of crisp DES such as strictly-output-control-consistency and H-observability are extended and redefined for FDES to maintain the hierarchical consistency between low-level and high-level modules. Furthermore, algorithms are provided to compute the necessary fuzzy language specifications and an application of behavior-based mobile navigation is discussed using the developed theory.

Chapter 7

Conclusion

The objective of this thesis is to address the supervisory control problem of FDES broadly, while investigating several possible applications related to the autonomous navigation of mobile robots. In this chapter, a summary of the research is given and the key contributions are presented. Some future research directions are also discussed at the end.

7.1 Discussion

This research explores the supervisory control problem of FDES in three main architectures, namely decentralized, modular and hierarchical. Furthermore, it develops a novel FDES-based supervisory control framework for modeling and control of behavior-based robots.

In Chapter 3, we proposed a supervisory control framework based on FDES for behavior-based control of SRS. Due to the necessity of responding to reactive situations in a navigational task, an event-driven model is preferred. The vagueness of representation stems from the uncertainties of states and the inaccuracies of sensory information. Hence, behavior-based robotics serve as a favorable application domain for FDES-based system modeling and control. Identification of fuzzy controllable and uncontrollable events as incidents, which trigger deliberative and reactive behaviors respectively, is the key concept behind the proposed framework. The approach shows a modular scheme where behaviors can be integrated effectively. This methodology outperforms the approach presented in [10] by having reduced computational complexity. The simulation and experimental results demonstrated the performance of the proposed method compared to some of the existing approaches.

In Chapter 4, we established a general decentralized supervisory control architecture for large-scale and distributed FDES models. The previous work on decentralized supervisory control of FDES concerned events fusion based on the fuzzy-intersection operator only and this may lead to a suboptimal global decision. Therefore, first we proposed a new approach, which fuses the fuzzy events using the fuzzy-union operator. Then, by combining both above operators, a general form of decentralized supervision of FDES was established for better information association and optimal decision making. The general architecture was implemented on an MRS having two mobile robots. The objective was to perform a tightly-coupled object manipulation task in an environment with unmodeled obstacles. Both simulation and experimental results demonstrated the success of the approach. In the real-time implementation, a perfect communication was assumed between the robots and the host computer. Moreover, a performance evaluation was carried out to compare the architecture with centralized FDES and decentralized crisp DES-based architectures. The results showed that the proposed approach was successful.

In Chapter 5, we developed a decentralized supervisory control theory for concurrently operating FDES modules. Each module is controlled by a supervisor, which does not communicate, representing a practical constraint in a physical multi-module system. Global supervision is distributed among the local supervisors to resolve the horizontal complexities of control computation.

In Chapter 6, we formulated a hierarchical supervisory control theory for large-scale and complex FDES models. The hierarchical control architecture consists of multi-level supervisors assigned to detailed low-level and abstract high-level models of the plant to resolve the vertical control complexities. The system requirements are specified at high-level and they are achieved at low-level while preserving the consistency. Due to the nature of possibility distributions in FDES, the low-level fuzzy languages cannot be derived from the high-level specification by a straightforward approach as in crisp DES control. To overcome this, algorithms are provided to calculate the low-level specification. Finally, using the established hierarchical supervisory control theory of FDES, a behavior-based mobile robot navigation is discussed as a proof-of-concept application.

7.1.1 Research limitations

Several key problems can be identified related to this research work. Mainly, specifying system requirements as a form of a fuzzy language is difficult and tedious. This requires obtaining the exact possibility distributions of fuzzy strings, which is troublesome and depends on the environment. Especially, in reactive behavior-based systems, which require abrupt changes in actions, providing such a pre-defined language specification seems impossible. An alternative and more convenient approach to the string representation in event-based control of FDES is the state representation in state-based control of FDES [22]. However, in this approach a set of pre-specified fuzzy states need to be provided, which is infeasible in applications such as behavior-based robotics.

Also, for optimal control problems of FDES, constructing a forward-looking tree is tedious as the FDES tree inherits more branches compared to its crisp-DES counterpart. Furthermore, the underlying fuzzy rule bases of each fuzzy event should be optimized for better performances.

7.2 Research contributions

The key contributions of this research can be summarized as follows.

1. **Development of an FDES-based supervisory control framework for modeling and control of behavior-based systems**

   Such an approach supports a formal evaluation of a system, which is affected by deterministic uncertainties in its states and events representation. Moreover, the proposed method eases the behavior integration and it has less computational complexity compared to the previous FDES-based approach. The framework was implemented both in simulation and in real-time on a mobile robot moving in unmodeled environments, and its performance was compared with other approaches.

2. **Establishment of a general decentralized supervisory control theory of FDES and its application on behavior-based coordination in MRS**

   Firstly, a new architecture, which fuses the events using the fuzzy-union operator was proposed to capture a new set of information in decentralized decision making. Secondly, incorporating this and existing approaches, a general form of decentralized supervision was established for better information association. The general architecture was implemented both in simulation and in real-time on an MRS and its performance was evaluated.

3. **Establishment of a decentralized modular supervisory control theory of concurrent FDES**

   For concurrently operating and multiple interacting FDES modules, a modular supervisory control theory was proposed to decrease the control complexity in the horizontal direction.

4. **Establishment of a hierarchical supervisory control theory of FDES**

   For large-scale complex FDES, a hierarchical supervisory control theory was developed to reduce the vertical control complexities, while maintaining the consistency between models. An example of robot navigation was discussed to show the applicability of the approach to physical systems.

7.2.1 Publications resulting from the research

This research work led to the following technical papers.

- Journal papers:

  1. Awantha Jayasiri, George K. I. Mann and Raymond G. Gosine, "Generalizing the Decentralized Control of Fuzzy Discrete Event Systems", in press as a regular paper in *IEEE Transactions on Fuzzy Systems*.
  2. Awantha Jayasiri, George K. I. Mann and Raymond G. Gosine, "Modular Supervisory Control and Hierarchical Supervisory Control of Fuzzy Discrete Event Systems", in press as a regular paper in *IEEE Transactions on Automation Science and Engineering*.
  3. Awantha Jayasiri, George K. I. Mann and Raymond G. Gosine, "Behavior Coordination of Mobile Robotics Using Supervisory Control of Fuzzy Discrete Event Systems", as a regular paper in *IEEE Transactions on*

Systems, Man, and Cybernetics, Part B: Cybernetics, vol.41, no.5, pp. 1224-1238, 2011.

- Conference papers:

  1. Awantha Jayasiri, George K. I. Mann and Raymond G. Gosine, "A general Architecture for Decentralized Supervisory Control of Fuzzy Discrete Event Systems," accepted in *American Control Conference, ACC*, 2012.
  2. Awantha Jayasiri, George K. I. Mann and Raymond G. Gosine, "Hierarchical supervisory control of fuzzy discrete event systems", in *Proceedings of the American Control Conference, ACC*, 2011, pp. 4490-4495.
  3. Awantha Jayasiri, George K. I. Mann and Raymond G. Gosine, "Tightly-coupled multi robot coordination using decentralized supervisory control of Fuzzy Discrete Event Systems", in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2011, pp. 3358-3363.
  4. Awantha Jayasiri, George K. I. Mann and Raymond G. Gosine, "Decentralized modular control of concurrent Fuzzy Discrete Event Systems", in *Proceedings of the American Control Conference, ACC*, 2010, pages 3359-3364.
  5. Awantha Jayasiri, George K. I. Mann and Raymond G. Gosine, "Mobile robot behavior coordination using supervisory control of Fuzzy Discrete Event Systems", in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2009, pp. 690-695.
  6. Awantha Jayasiri, George K. I. Mann and Raymond G. Gosine, "Mobile robot navigation in unknown environments based on supervisory control of partially-observed Fuzzy Discrete Event Systems", in *Proceedings of the International Conference on Advanced Robotics, ICAR*, 2009, pp. 1-6.

7. Awantha Jayasiri, George K. I. Mann and Raymond G. Gosine, "Supervisory control of Fuzzy Discrete Event Systems and its application to mobile robot navigation", in *Proceedings of the Canadian Conference on Electrical and Computer Engineering, CCECE*, 2009, pp. 1147-1151.

7.3 Future directions

Two main future research potentials can be identified based on this work.

- We believe the presented theoretical contributions in this research together with the previous work in the literature have discussed the supervisory control of FDES primarily. However, a comprehensive study of the supervisory control of FDES covering all the areas parallel to those of crisp DES still needs to be performed. Some interesting research areas such as distributed supervisory control of FDES and hierarchical control of decentralized FDES are yet to be established and their applicabilities for real-world problems need to be investigated.
- In contrast to the FDES-based modeling, which can represent a system with its deterministic uncertainties, Probabilistic Discrete Event Systems (PDES) model the stochastic behavior of a system having non-deterministic uncertainties [104]. Therefore, a system can be modeled with a complete representation of its uncertainties by an architecture, which utilizes both FDES and PDES. The supervisory control of such a combined framework will be challenging, yet interesting, research.

Appendix: Proofs of theorems

***Theorem 3.1* proof**

Note that we consider the general case where both (3.4) and (4.1) holds (i.e., both event observability and event controllability are fuzzy).

$\Longrightarrow$ Consider the following.

1. The base case string of length = 0. By definition $L_{S^P/G}(\varepsilon)$=1, $\bar{k}(\varepsilon)$=1. Assume the condition is true for fuzzy string s and $|s| \leq n$. Then $L_{S^P/G}(s) = \bar{k}(s)$.

2. Consider σ as a non-null fuzzy event, so that $|s\sigma| = n + 1$. From (3.13) We know that: $L_{S^P/G}(s\sigma) = L_{S^P/G}(s)\tilde{\cap}S_t^P(\sigma)\tilde{\cap}L_G(s\sigma)$.

$$\Rightarrow L_{S^P/G}(s\sigma) = \bar{k}(s)\tilde{\cap}S_t^P(\sigma)\tilde{\cap}L_G(s\sigma) \tag{A}$$

$\Rightarrow L_{S^P/G}(s\sigma) = \bar{k}(s)\tilde{\cap}S_t^P(\sigma)\tilde{\cap}L_G(s\sigma)$

Assume $s'\sigma \in \bar{k}$, $\sigma \in \Sigma$ and $P(s) = t$.

Consider the first scenario in (3.16) where $S_t^P(\sigma) = \Sigma_{uc}(\sigma)\tilde{\cap}L_G(s\sigma)$.

Substituting with (A) and fuzzy controllability condition in (3.9) leads to:

$\Rightarrow L_{S^P/G}(s\sigma) = \bar{k}(s)\tilde{\cap}\Sigma_{uc}(\sigma)\tilde{\cap}L_G(s\sigma) \leq \bar{k}(s\sigma)$.

Consider the second scenario in (3.16)

where $S_t^P(\sigma) = L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P^{-1}\left[P(s')\right](s)\tilde{\cap}\Sigma_c(\sigma)$.

Substituting with (A) and fuzzy observability condition in (3.14) leads to: $\Rightarrow$ $L_{S^P/G}(s\sigma) = \bar{k}(s)\tilde{\cap}L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P^{-1}\left[P(s')\right](s)\tilde{\cap}\Sigma_c(\sigma) \leq \bar{k}(s\sigma)$.

Consider the third scenario in (3.16) where $S_t^P(\sigma) = \bar{k}(s\sigma)$.

$\Rightarrow L_{S^P/G}(s\sigma) = \bar{k}(s)\tilde{\cap}\bar{k}(s\sigma)\tilde{\cap}L_G(s\sigma) \leq \bar{k}(s\sigma)$

We have shown $L_{S^P/G}(s\sigma) \leq \bar{k}(s\sigma)$.

If $\Sigma_{uc}(\sigma)\tilde{\cap}L_G(s\sigma) \geq \bar{k}(s\sigma)$ then according to (3.16), $S_t^P(\sigma) = \Sigma_{uc}(\sigma)\tilde{\cap}L_G(s\sigma)$. Substituting with $\bar{k}(s\sigma) \leq \bar{k}(s)$ and $\bar{k}(s\sigma) \leq L_G(s\sigma)$: $\bar{k}(s\sigma) \leq \bar{k}(s)\tilde{\cap}\Sigma_{uc}(\sigma)\tilde{\cap}L_G(s\sigma)$.

$\Rightarrow \bar{k}(s\sigma) \leq L_{S^P/G}(s)\tilde{\cap}S_t^P(\sigma)\tilde{\cap}L_G(s\sigma) \Rightarrow \bar{k}(s\sigma) \leq L_{S^P/G}(s\sigma)$

If $L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P^{-1}\left[P(s')\right](s)\tilde{\cap}\Sigma_c(\sigma) \geq \bar{k}(s\sigma)$, Substituting with $\bar{k}(s\sigma) \leq \bar{k}(s)$ and $\bar{k}(s\sigma) \leq L_G(s\sigma)$: $\bar{k}(s\sigma) \leq \bar{k}(s)L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P^{-1}\left[P(s')\right](s)\tilde{\cap}\Sigma_c(\sigma)$.
$\Rightarrow \bar{k}(s\sigma) \leq L_{S^P/G}(s)\tilde{\cap}S_t^P(\sigma)\tilde{\cap}L_G(s\sigma) \Rightarrow \bar{k}(s\sigma) \leq L_{S^P/G}(s\sigma)$

If $S_t^P(\sigma) = \bar{k}(s\sigma)$ then with $\bar{k}(s\sigma) \leq \bar{k}(s)$ and $\bar{k}(s\sigma) \leq L_G(s\sigma)$: $\bar{k}(s\sigma) \leq \bar{k}(s)\tilde{\cap}S_t^P(\sigma)\tilde{\cap}L_G(s\sigma)$. $\Rightarrow \bar{k}(s\sigma) \leq L_{S^P/G}(s)\tilde{\cap}S_t^P(\sigma)\tilde{\cap}L_G(s\sigma) \Rightarrow \bar{k}(s\sigma) \leq L_{S^P/G}(s\sigma)$
Therefore, we have proved that $\bar{k}(s\sigma) = L_{S^P/G}(s\sigma)$ holds for any $s \in \Sigma^*$ and $\sigma \in \Sigma$ where $|s \cdot \sigma| = n + 1$. This completes the proof of the induction step.

$\Longleftarrow$ Assume $L_{S^P/G}(s\sigma) = \bar{k}(s\sigma)$.

1. Proof of fuzzy controllability condition holds.

   We know $L_{S^P/G}(s\sigma) = L_{S^P/G}(s)\tilde{\cap}S_t^P(\sigma)\tilde{\cap}L_G(s\sigma)$. With fuzzy admissibility condition in (3.12) this leads to: $L_{S^P/G}(s\sigma) \geq L_{S^P/G}(s)\tilde{\cap}\Sigma_{uc}(\sigma)\tilde{\cap}L_G(s\sigma)$. Also we know that $L_{S^P/G}(s\sigma) = \bar{k}(s\sigma)$ and $L_{S^P/G}(s) = \bar{k}(s)$.

   $\Rightarrow \bar{k}(s\sigma) \geq \bar{k}(s)\tilde{\cap}\Sigma_{uc}(\sigma)\tilde{\cap}L_G(s\sigma)$

   The fuzzy controllability condition holds.

2. Proof of fuzzy observability condition also holds:

   Let $L_{S^P/G}(s)\tilde{\cap}S_t^P(\sigma)\tilde{\cap}L_G(s\sigma) = L_{S^P/G}(s\sigma)$.

   From (3.15): $S_t^P(\sigma) \geq L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P^{-1}\left[P(s')\right](s)\tilde{\cap}\Sigma_c(\sigma)$. $\Rightarrow$

   $L_{S^P/G}(s)\tilde{\cap}L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P^{-1}\left[P(s')\right](s)\tilde{\cap}\Sigma_c(\sigma) \leq L_{S^P/G}(s\sigma)$

   With $L_{S^P/G}(s\sigma) = \bar{k}(s\sigma)$ and $L_{S^P/G}(s) = \bar{k}(s), \Rightarrow$

   $\bar{k}(s)\tilde{\cap}L_G(s\sigma)\tilde{\cap}\bar{k}(s'\sigma)\tilde{\cap}P^{-1}\left[P(s')\right](s)\tilde{\cap}\Sigma_c(\sigma) \leq \bar{k}(s\sigma)$.

   The fuzzy observability condition also holds.

3. To prove $L_{G,m}$- closure also holds, let us extend the definition of $L_{S/G,m}(s)$ for partial observation scenario:

   $\bar{L}_{S^P/G,m}(s) = L_{S^P/G}(s)\tilde{\cap}\bar{L}_{G,m}(s) \Rightarrow \bar{L}_{S^P/G,m}(s) \leq \bar{L}_{G,m}(s) \Rightarrow \bar{k}(s) \leq \bar{L}_{G,m}(s)$.

   Then, $L_{G,m}$- closure condition also holds. This completes the proof.

***Theorem 4.1* proof**:

$\Longrightarrow$ Proving that the fuzzy controllability and $L_{G,m}$- closure conditions hold, is the same as in the proof of *Theorem 3.1.* Only the proof that fuzzy C&P co-observability holds is described below.

Assume $\bar{k}$ is not fuzzy C&P co-observable with respect to L_G, $P_1, ..., P_n$ and $\Sigma_{1,c}, ..., \Sigma_{n,c}$. This implies when $s, s', s'\sigma \in \bar{k}$, $\Sigma_{cp,c}(\sigma) > 0$ and $\exists\ h \in \{1, ..., n\}$ such that $s \in P_h^{-1}\left[P_h(s')\right]$ but $s\sigma \in L_G \backslash \bar{k}$.

Assume the special case where $\forall\ i \in \{1, ..., n\}$, $P_i(s) = P_i(s') = t_i$.

$$\Rightarrow \sigma \notin S_s^{cp} \Rightarrow \sigma \notin \bigcap_{i=1}^{n} S_{t_i}^{P_i} \Rightarrow \exists\ j \in \{1, ..., n\} \text{ such that } \sigma \notin S_{t_j}^{P_j} \text{ and } \sigma \in \Sigma_{j,c}. \quad \text{(A)}$$

Also, with above assumptions:

$$s'\sigma \in \bar{k} \Rightarrow \sigma \in S_{s'}^{cp} \Rightarrow \sigma \in \bigcap_{i=1}^{n} S_{t_i}^{P_i} \Rightarrow \forall\ i \in \{1, ..., n\} \text{ such that } \sigma \in S_{t_i}^{P_i} \text{ and } \sigma \in \Sigma_{i,c} \tag{B}$$

Note that (A) and (B) leads to a contradiction. Then $\bar{k}$ must be fuzzy C&P co-observable with respect to L_G, $P_1, ..., P_n$ and $\Sigma_{1,c}, ..., \Sigma_{n,c}$.

$\Longleftarrow$ The proof of $\bar{k} = \bar{L}_{S^{cp}/G,m}$ and $L_{S^{cp}/G} = \bar{k}$ while k holds fuzzy controllability, fuzzy C&P co-observability and $L_{G,m}$-closure is same as in proof of *Theorem 3.1*. The modifications are S_t^P changes to S_s^{cp} and fuzzy observability changes to fuzzy C&P co-observability. This completes the proof.

***Theorem 4.2* proof**:

$\Longrightarrow$ Proving fuzzy controllability and $L_{G,m}$- closure conditions are hold, are same as in proof of *Theorem 3.1*. Only the proof of fuzzy D&A co-observability is also hold is described below.

Assume $\bar{k}$ is not fuzzy D&A co-observable with respect to L_G, $P_1, ..., P_n$ and $\Sigma_{1,c}, ..., \Sigma_{n,c}$. This implies when $s, s', s\sigma \in \bar{k}$, $\Sigma_{da,c}(\sigma) > 0$ and $\forall\ i \in \{1, ..., n\}$ such that $s' \in P_i^{-1}[P_i(s)]$ but $s'\sigma \in L_G \backslash \bar{k}$.

Assume the special case where $\forall\ i \in \{1, ..., n\}$, $P_i(s) = P_i(s') = t_i$.

$$\Rightarrow \sigma \notin S_{s'}^{da} \Rightarrow \sigma \notin \bigcup_{i=1}^{n} S_{t_i}^{P_i} \Rightarrow \forall\ i \in \{1, ..., n\} \text{ such that } \sigma \notin S_{t_i}^{P_i} \text{ and } \sigma \in \Sigma_{i,c}. \tag{C}$$

Also, with above assumptions:

$$s\sigma \in \bar{k} \Rightarrow \sigma \in S_s^{da} \Rightarrow \sigma \in \bigcup_{i=1}^{n} S_{t_i}^{P_i} \Rightarrow \exists\ k \in \{1, ..., n\} \text{ such that } \sigma \in S_{t_k}^{P_k} \text{ and } \sigma \in \Sigma_{k,c}. \tag{D}$$

Note that (C) and (D) leads to a contradiction. Then $\bar{k}$ must be fuzzy D&A co-observable with respect to L_G, $P_1, ..., P_n$ and $\Sigma_{1,c}, ..., \Sigma_{n,c}$.

$\Longleftarrow$ The proof of $\bar{k} = \bar{L}_{S^{da}/G,m}$ and $L_{S^{da}/G} = \bar{k}$ while $\bar{k}$ holds fuzzy controllability, fuzzy D&A co-observability and $L_{G,m}$-closure is same as in proof of *Theorem 3.1.* The modifications are S_t^P changes to S_s^{da} and fuzzy observability changes to fuzzy D&A co-observability. This completes the proof.

***Theorem 4.3* proof**:

$\Longrightarrow$ Proving fuzzy controllability, fuzzy co-observability and $L_{G,m}$- closure conditions are hold, are same as in proof of *Theorem 3.1, Theorem 4.1* and *Theorem 4.2.* The only modification is fuzzy co-observability in generalized architecture represents both fuzzy C&P and fuzzy D&A co-observabilities as mentioned in *Definition 4.3.*

$\Longleftarrow$ The proof of $\bar{k} = \bar{L}_{S^g/G,m}$ and $L_{S^g/G} = \bar{k}$ while $\bar{k}$ holds fuzzy controllability, fuzzy co-observability and $L_{G,m}$-closure is same as in proof of *Theorem 3.1.* The modifications are S_t^P changes to S_s^g, which represents both S_s^{cp} and S_s^{da}, and fuzzy observability changes to fuzzy co-observability. This completes the proof.

***Theorem 5.1* proof**:

$\Longrightarrow$ Assume there exists a set of modular partial observation supervisors $\{S_1^P, ..., S_n^P\}$ such that $\bar{L}_{(S_1^P/G_1)||...||(S_n^P/G_n),m}(s) = \bar{k}(s)$ and $L_{(S_1^P/G_1)||...||(S_n^P/G_n)}(s) = \bar{k}(s)$.

Define $\bar{k}_i(s) = L_{S_i^P/G_i}(s)$ and $\bar{k}_i(s) = \bar{L}_{S_i^P/G_i,m}(s)$.

Then $\bar{k}(s) = P_1^{-1}(\bar{k}_1)(s)\tilde{\cap}...\tilde{\cap}P_n^{-1}(\bar{k}_n)(s)$. $\Rightarrow$ k is *separable.*

Since $L_{S_i^P/G_i}$ is the language generated by partial observation supervisor S_i^P for the plant G_i, it is fuzzy controllable, fuzzy observable and $L_{G_i,m}$-closed. So k_i also fuzzy controllable, fuzzy observable and $L_{G_i,m}$-closed according to $Theorem$ 3.1, because $\bar{k}_i(s) = L_{S_i^P/G_i}(s)$ and $\bar{k}_i(s) = \bar{L}_{S_i^P/G_i,m}(s)$. $\Rightarrow$

1. k is *separably-controllable-observable.*

2. k_i is $L_{G_i,m}$-closed.

$\Longleftarrow$ Assume above two conditions are hold. According to *Theorem 3.1* fuzzy controllability, fuzzy observability and $L_{G_i,m}$-closure imply that there exist a set of modular partial observation supervisors $S_1^P, ...S_n^P$ such that $\bar{k}_i(s) = \bar{L}_{S_i^P/G_i,m}(s)$ and $\bar{k}_i(s) = L_{S_i^P/G_i}(s)$.

The separability condition of fuzzy language k with respect to $\{P_1, ..., P_n\}$ implies there exists a set of fuzzy languages $\bar{k}_i \subseteq \Sigma_i^*$ such that, $\bar{k}(s) = P_1^{-1}(\bar{k}_1)(s)\tilde{\cap}...\tilde{\cap}P_n^{-1}(\bar{k}_n)(s)$. By substituting $\bar{k}_i(s) = L_{S_i^P/G_i}(s)$, $\bar{k}(s) = P_1^{-1}(L_{S_1^P/G_1})(s)\tilde{\cap}...\tilde{\cap}P_n^{-1}(L_{S_n^P/G_n})(s)$.

$\Rightarrow \bar{k}(s) = L_{(S_1/G_1)||...||(S_n/G_n)}(s)$ and $\bar{k}(s) = \bar{L}_{(S_1/G_1)||...||(S_n/G_n),m}(s)$.

This completes the proof.

***Theorem 6.1* proof**:

By definition $\bar{k}_{lo}^{\uparrow}$ is fuzzy controllable. Conditions 1 and 3 of *Theorem 6.1* say that $\bar{k}_{lo}^{\uparrow}$ is $L_{lo,m}$-closed and fuzzy observable.

$\Rightarrow$ By the *Theorem* 3.1, there exist a low-level supervisor S_{lo} for G_{lo} such that $L_{S_{lo}/G_{lo}}(s) = \bar{k}_{lo}^{\uparrow}(s)$. Assume $\theta(\bar{k}_{lo}^{\uparrow}) \neq \bar{k}_{hi}$ and write $\theta(\bar{k}_{lo}^{\uparrow}) = W$ for simplicity.

$\rightarrow W(t'\tau) > \bar{k}_{hi}(t'\tau)$ or $W(t'\tau) < \bar{k}_{hi}(t'\tau)$

1. Consider the first case where $W(t'\tau) > \bar{k}_{hi}(t'\tau)$. Since $\bar{k}_{lo}^{\uparrow}$ is fuzzy observable, $\exists\ s_1, s_2, s_3, s_4 \in \Sigma^*$ and $t, t' \in T^*$ such that $s_1s_2 \in L_{lo}$ and $s_3s_4 \in L_{lo}$. Also $\theta(s_1) = t, \theta(s_1s_2) = t\tau$, and $\theta(s_3) = t', \theta(s_3s_4) = t'\tau$ with $P^{-1}\left[P(s_1)\right](s_3) > 0$ and $s_1, s_3 \in \bar{k}_{lo}^{\uparrow}$.

Furthermore, W is fuzzy controllable (Refer Remark 6.1). Therefore, $W(t'\tau)$ can be calculated with above information as follows:

Let $\psi_1 = W(t')\tilde{\cap}L_{hi}(t'\tau)\tilde{\cap}T_{uc}(\tau)$ and

$\psi_2 = W(t')\tilde{\cap}L_{hi}(t'\tau)\tilde{\cap}W(t\tau)\tilde{\cap}T_c(\tau)\tilde{\cap}P^{-1}\left[P(s_1)\right](s_3)$.

Then, $W(t'\tau) = \max\{\psi_1, \psi_2\}$. Now assume the special case in which $W(t'\tau) = \psi_2$ then, $W(t')\tilde{\cap}L_{hi}(t'\tau)\tilde{\cap}W(t\tau)\tilde{\cap}T_c(\tau)\tilde{\cap}P^{-1}\left[P(s_1)\right](s_3) > \bar{k}_{hi}(t'\tau)$.

But G_{lo} is strictly-output-control-consistent. Hence, $W(t') = \bar{k}_{hi}(t')$ and $W(t\tau) = \bar{k}_{hi}(t\tau) \rightarrow \bar{k}_{hi}(t')\tilde{\cap}L_{hi}(t'\tau)\tilde{\cap}\bar{k}_{hi}(t\tau)\tilde{\cap}T_c(\tau)\tilde{\cap}P^{-1}\left[P(s_1)\right](s_3) > \bar{k}_{hi}(t'\tau)$.

$\Rightarrow \bar{k}_{hi}$ is not H-fuzzy observable with respect to $\bar{k}'_{lo}$.

2. Consider the second case where $W(t'\tau) < \bar{k}_{hi}(t'\tau)$ Assume the special scenario

where $W(t'\tau) = 0$ provided $T_{uc}(\tau) = 0$ (i.e. τ is completely controllable) and $t' \in W$.

$$\bar{k}_{hi}(t'\tau) > 0 \rightarrow t'\tau \in \bar{k}_{hi} \tag{A}$$

In this case also since $\bar{k}_{lo}^{\uparrow}$ is fuzzy observable, $\exists\ s_1, s_2, s_3, s_4 \in \Sigma^*$ and $t, t' \in T^*$ such that $s_1s_2 \in L_{lo}$ and $s_3s_4 \in L_{lo}$. Also $\theta(s_1) = t, \theta(s_1s_2) = t\tau$, and $\theta(s_3) = t', \theta(s_3s_4) = t'\tau$ with $P^{-1}(P(s_1))(s_3) > 0$ and $s_1, s_3 \in \bar{k}_{lo}^{\uparrow}$. Furthermore, since $T_{uc}(\tau) = 0 \rightarrow W(t'\tau) = W(t')\tilde{\cap}L_{hi}(t'\tau)\tilde{\cap}W(t\tau)\tilde{\cap}T_c(\tau)\tilde{\cap}P^{-1}[P(s_1)](s_3)$ (i.e. ψ_2 above). Since, $W(t'\tau) = 0$ and $t' \in W \rightarrow W(t\tau) = 0$.

Since $\bar{k}_{hi}$ is H-fuzzy observable with respect to $\bar{k}_{lo}^{\uparrow}$ and G_{hi} and $T_{uc}(\tau) = 0$, consider the special case in which, $\bar{k}_{hi}(t\tau)\tilde{\cap}\bar{k}_{hi}(t')\tilde{\cap}L_{hi}(t'\tau)\tilde{\cap}T_c(\tau)\tilde{\cap}P^{-1}[P(s)](s') = \bar{k}_{hi}(t'\tau)$ But G_{lo} is strictly-output-control-consistent. Hence, $\bar{k}_{hi}(t') = W(t')$ and $\bar{k}_{hi}(t\tau) = W(t\tau) \rightarrow W(t\tau)\tilde{\cap}W(t')\tilde{\cap}L_{hi}(t'\tau)\tilde{\cap}T_c(\tau)\tilde{\cap}P^{-1}[P(s)](s') = \bar{k}_{hi}(t'\tau)$.

With $W(t\tau) = 0 \rightarrow \bar{k}_{hi}(t'\tau) = 0$,

$$\bar{k}_{hi}(t'\tau) = 0 \rightarrow t'\tau \notin \bar{k}_{hi} \tag{B}$$

Note that (A) and (B) leads to a contradiction.

$\Rightarrow$ With 1 and 2 we have proved that $\theta(\bar{k}_{lo}^{\uparrow}) = \bar{k}_{hi}$. This completes the proof.

Bibliography

[1] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Springer, 2008.

[2] R. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[3] ——, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[4] F. Lin and H. Ying, "Modeling and control of fuzzy discrete event systems," *IEEE Transactions of Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, pp. 408–415, 2002.

[5] Y. Cao and M. Ying, "Supervisory control of fuzzy discrete event systems," *IEEE Transaction of Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 2, pp. 366–371, 2005.

[6] D. Qiu, "Supervisory control of fuzzy discrete event systems: A formal approach," *IEEE Transactions of Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 1, pp. 72–88, 2005.

[7] H. Ying, F. Lin, R. MacArthur, J. Cohn, D. Barth-Jones, H. Ye, and L. Crane, "A fuzzy discrete event system approach to determining optimal HIV/AIDS treatment regimens," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 4, pp. 663–676, 2006.

[8] ——, "A self-learning fuzzy discrete event system for HIV/AIDS treatment regimen selection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 4, pp. 966–979, 2007.

[9] D. Li, W. Lan, H. Zhou, and S. Shao, "Control of fuzzy discrete event systems and its application to air conditioning system," *International Journal of Modelling, Identification and Control*, vol. 8, no. 2, pp. 122 –129, 2009.

[10] R. Huq, G. K. I. Mann, and R. R. Gosine, "Behavior modulation technique in mobile robotics using fuzzy discrete event system," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 903–916, 2006.

[11] A. Jayasiri, G. K. I. Mann, and R. R. Gosine, "Behavior coordination of mobile robotics using supervisory control of fuzzy discrete event systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 5, pp. 1224–1238, 2011.

[12] 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.

[13] F. Liu and D. Qiu, "Decentralized supervisory control of fuzzy discrete events systems," *European Journal of Control*, vol. 14, no. 3, pp. 234–243, 2008.

[14] F. Wang, Z. Feng, and P. Jiang, "Reliable decentralized supervisory control of fuzzy discrete event systems," *Fuzzy Sets and Systems*, vol. 161, no. 12, pp. 1657 – 1668, 2010.

[15] T. S. Yoo and S. Lafortune, "A general architecture for decentralized supervisory control of discrete-event systems," *Discrete Event Dynamic Systems, Theory and Applications*, vol. 12, no. 3, pp. 335–377, 2002.

[16] R. Kumar and V. K. Garg, *Modeling and Control of Logical Discrete Event Systems*. Boston, MA: Kluwer, 1995.

[17] L. E. Holloway, B. H. Krogh, and A. Giua, "A survey of Petri net methods for controlled discrete event systems," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 7, no. 2, pp. 151–190, 1997.

[18] D. W. Qiu, "Characterizations of fuzzy finite automata," *Fuzzy Sets and Systems*, vol. 141, no. 3, pp. 391–414, 2004.

[19] A. Kandel and S. C. Lee, *Fuzzy Switching and Automata: Theory and Applications*. Crane Russak, 1979.

[20] C. G. Looney, "Fuzzy Petri nets for rule-based decision making," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 178 –183, 1988.

[21] E. S. Santos, "Max-min automata," *Information and Control*, vol. 13, pp. 363–377, 1968.

[22] Y. Cao, M. Ying, and G. Chen, "State-based control of fuzzy discrete-event systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 2, pp. 410–424, 2007.

[23] 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, 2009.

[24] F. Lin, H. Ying, R. D. MacArthur, J. A. Cohn, D. B. Jones, and L. R. Crane, “Decision making in fuzzy discrete event systems,” *Information Sciences*, vol. 177, no. 18, pp. 3749–3763, 2007.

[25] E. Kilic, “Diagnosability of fuzzy discrete event systems,” *Information Sciences*, vol. 178, no. 3, pp. 858–870, 2008.

[26] 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.

[27] 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, no. 2, pp. 316–328, 2009.

[28] K. Wongsopanakul, “An extended fuzzy discrete event system for HIV/AIDS treatment regimen selection,” Ph.D. dissertation, Department of Electrical and Computer Engineering, Wayne State University, Detroit, Michigan, USA, 2011.

[29] J. M. Mendel, “Advances in type-2 fuzzy sets and systems,” *Informatin Sciences*, vol. 177, no. 1, pp. 84–110, 2007.

[30] F. Lin and W. M. Wonham, “Decentralized supervisory control of discrete-event systems,” *Information Sciences*, vol. 44, no. 3, pp. 199–224, 1988.

[31] K. Rudie and W. M. Wonham, “Think globally, act locally: Decentralized supervisory control,” *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1692–1708, 1992.

[32] P. Kozak and W. M. Wonham, "Fully decentralized solutions of supervisory control problems," *IEEE Transactions on Automatic Control*, vol. 40, no. 12, pp. 2094–2097, 1995.

[33] S. Takai and T. Ushio, "Reliable decentralized supervisory control of discrete event systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 30, no. 5, pp. 661–667, 2000.

[34] ——, "A modified normality condition for decentralized supervisory control of discrete event systems," *Automatica*, vol. 38, no. 1, pp. 185–189, 2002.

[35] W. Qiu, R. Kumar, and V. Chandra, "Decentralized control of discrete event systems using prioritized composition with exclusion," *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2425 –2430, 2008.

[36] R. Kumar and S. Takai, "Inference-based ambiguity management in decentralized decision-making: Decentralized control of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 10, pp. 1783–1794, 2007.

[37] H. Chakib and A. Khoumsi, "Multi-decision supervisory control: Parallel decentralized architectures cooperating for controlling discrete event systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2608 –2622, 2011.

[38] M. Matarić, "Behavior-based control: Examples from navigation, learning, and group behavior," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 323–336, 1997.

[39] R. R. Murphy, *Introduction to AI Robotics.* MIT Press, 2000.

[40] C. V. Jones and M. J. Mataric, "Behavior-based coordination in multi-robot systems," in *Autonomous Mobile Robots: Sensing, Control, Decision-Making, and Applications.* Marcel Dekker Inc., 2005, pp. 549–569.

[41] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Vision and navigation for the carnegie mellon navlab," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 362–373, 1988.

[42] R. Brooks, "Intelligence without representation," *Artificial Intelligence*, vol. 47, no. 1-3, pp. 139–159, 1991.

[43] J. Rosenblatt and J. A. Hendler, "Architectures for mobile robot control," *Advances in Computers*, vol. 48, pp. 315–353, 1999.

[44] R. Simmons, "Structured control for autonomous robots," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 1, pp. 34–43, 1994.

[45] E. Gat, "Three-layer architectures," in *Artificial intelligence and mobile robots: Case Studies of Successful Robot Systems.* MIT Press, 1998, pp. 195–210.

[46] R. Arkin, *Behavior-Based Robotics.* MIT Press, 1998.

[47] P. Pirjanian, "Behavior coordination mechanisms - state-of-the-art," Institute of Robotics and Intelligent Systems, School of Engineering, University of Southern California, Tech. Rep., 1999.

[48] D. Payton, J. K. Rosenblatt, and D. Keirsey, "Plan guided reaction," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 6, pp. 1370–1382, 1990.

[49] M. K. Sahota, "Action selection in robotics in dynamic environments through inter-behavior bidding," in *Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior, SAB*, 1994, pp. 138–142.

[50] J. Rosenblatt, "DAMN: Distributed architecture for mobile navigation," Ph.D. dissertation, Carnegie Mellon University Robotics Institute, Pittsburgh, PA, USA, 1997.

[51] C. K. Tan, S. H. M. Amin, R. Mamat, and J. K. Tar, "Using voting technique in mobile robot behavior coordination for goal-directed navigation," *Jurnal Teknologi*, vol. 36(D), pp. 55–70, 2002.

[52] A. Saffiotti, K. Konolige, and E. H. Ruspini, "A multivalued logic approach to integrating planning and control," *Artificial Intelligence*, vol. 76, no. 1-2, pp. 481–526, 1995.

[53] P. Vadakkepat, O. C. Miin, X. Peng, and T. H. Lee, "Fuzzy behavior-based control of mobile robots," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 559–565, 2004.

[54] D. Shi, E. G. C. Jr., and D. Dunlap, "Robot navigation in cluttered 3-D environments using preference-based fuzzy behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 6, pp. 1486–1499, 2007.

[55] S. Verret, "Current state of the art in multirobot systems," Defence R&D Canada - Suffield, Tech. Rep., 2005.

[56] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot systems: a classification focused on coordination," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 5, pp. 2015–2028, 2004.

[57] L. Parker, "Alliance: an architecture for fault tolerant multirobot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.

[58] ——, “Cooperative robotics for multi-target observation,” *Intelligent Automation and Soft Computing*, vol. 5, no. 1, pp. 5–19, 1999.

[59] ——, “On the design of behavior-based multi-robot teams,” *Journal of Advanced Robotics*, vol. 10, no. 6, pp. 547–578, 1996.

[60] T. Huntsberger, P. Pirjanian, A. Trebi-Ollennu, H. Das Nayar, H. Aghazarian, A. Ganino, M. Garrett, S. Joshi, and P. Schenker, “CAMPOUT: A control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 33, no. 5, pp. 550–559, 2003.

[61] T. L. Huntsberger, A. Trebi-Ollennu, H. Aghazarian, P. S. Schenker, P. Pirjanian, and H. D. Nayar, “Distributed control of multi-robot systems engaged in tightly coupled tasks,” *Autonomous Robots*, vol. 17, no. 1, pp. 79–92, 2004.

[62] T. Balch and R. C. Arkin, “Behavior-based formation control for multi-robot teams,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[63] R. C. Arkin and T. Balch, “AuRA: Principles and practice in review,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 175–189, 1997.

[64] W. Chun and T. Jochem, “Unmanned ground vehcle demo II: Demonstration A,” *Unmanned Systems Magazine*, vol. 12, no. 1, pp. 14–20, 1994.

[65] K. T. Song and C. C. Tang, “A distributed architecture for effective multirobot cooperation in team competitions,” in *Proceedings of the 25th Annual Conference of the IEEE Industrial Electronics Society, IECON*, vol. 2, 1999, pp. 547–552.

[66] B. B. Werger, "Ayllu: Distributed port-arbitrated behavior-based control," in *Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems, DARS*, vol. 4, 2000, pp. 25–34.

[67] B. B. Werger and M. J. Mataric, "Broadcast of local eligibility for multi-target observation," in *Proceedings of the 5th International Symposium on Distributed Autonomous Robotic Systems, DARS*, vol. 4, 2000, pp. 347–356.

[68] B. P. Gerkey and M. J. Mataric, "Principled communication for dynamic multi-robot task allocation," in *Proceedings of the 7th International Symposium on Experimental Robotics, ISER*, 2001, pp. 353–362.

[69] L. Iocchi, D. Nardi, M. Piaggio, and A. Sgorbissa, "Distributed coordination in heterogeneous multi-robot systems," *Autonomous Robots*, vol. 15, no. 2, pp. 155–168, 2003.

[70] P. Pirjanian and M. Mataric, "Multi-robot target acquisition using multiple objective behavior coordination," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2000, pp. 2696–2702.

[71] A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation: a catalogue raisonnè," IRIDIA, Universitè Libre de Bruxelles, 50 av. F. Rossevelt, CP 194/6, B-1050, Tech. Rep., 1997.

[72] D. Goldberg and M. Mataric, "Design and evaluation of robust behavior-based controllers for distributed multi-robot collection tasks," in *Robot Teams: From Diversity to Polymorphism*. AK Peters, 2002, pp. 315–344.

[73] R. Simmons, T. Smith, M. B. Dias, D. Goldberg, D. Hershberger, A. Stentz, and R. Zlot, "A layered architecture for coordination of mobile robots," in

Multi-Robot Systems: From Swarms to Intelligent Automata: Proceedings from the NRL Workshop on Multi-Robot Systems, 2002, pp. 103–112.

[74] P. Garcia, P. Caamao, F. Bellas, and R. J. Duro, "A behavior based architecture with auction-based task assignment for multi-robot industrial applications," in *LNCS / Bioinspired Applications in Artificial and Natural Computation*. Springer-Verlag, 2009, pp. 372–381.

[75] D. Jung and A. Zelinsky, "An architecture for distributed cooperative planning in a behaviour-based multi-robot system," *Robotics and autonomous systems*, vol. 26, no. 2-3, pp. 149–174, 1999.

[76] R. Fierro, A. Das, J. Spletzer, J. Esposito, V. Kumar, J. P. Ostrowski, G. Pappas, C. J. Taylor, Y. Hur, R. Alur, I. Lee, G. Grudic, and B. Southall, "A framework and architecture for multi-robot coordination," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 977–995, 2002.

[77] K. H. Lee and J. H. Kim, "Multi-robot cooperation-based mobile printer system," *Robotics and Autonomous Systems*, vol. 54, no. 3, pp. 193–204, 2006.

[78] C. Kuppan, M. Singaperumal, and T. Nagarajan, "Layered framework for formation control of multiple mobile robots - A state-based approach," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics, ROBIO*, 2009, pp. 2067–2072.

[79] J. Yen and R. Langari, *Fuzzy Logic: Intelligence, Control and Information.* Pearson Education, 2003.

[80] A. Saffiotti, "Fuzzy logic in autonomous robotics: behavior coordination," in *Proceedings of the 6th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, vol. 1, 1997, pp. 573–578.

[81] A. Saffiotti, E. H. Ruspini, and K. Konolige, "Blending reactivity and goal-directedness in a fuzzy controller," in *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems, FUZZ-IEEE*, vol. 1, 1993, pp. 134–139.

[82] E. Tunstel, M. A. A. de Oliveira, and S. Berman, "Fuzzy behavior hierarchies for multi-robot control," *International Journal of Intelligent Systems*, vol. 17, no. 5, pp. 449–470, 2002.

[83] D. Pham and M. H. Awadalla, "Fuzzy-logic-based behaviour coordination in a multi-robot system," in *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 218, no. 6, 2004, pp. 583–598.

[84] B. Damas and P. Lima, "Stochastic discrete event model of a multi-robot team playing an adversarial game," in *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, IAV*, 2004.

[85] P. F. Palamara, V. A. Ziparo, L. Iocchi, D. Nardi, and P. Lima, "Teamwork design based on Petri net plans," in *LNCS/RoboCup 2008: Robot Soccer World Cup XII.* Springer-Verlag, 2009, pp. 200–211.

[86] E. Roszkowska, "DES-based coordination of space-sharing mobile robots," in *LNCS / Computer Aided Systems Theory EUROCAST.* Springer-Verlag, 2007, vol. 4739, pp. 1041–1048.

[87] ——, "Decentralized motion-coordination policy for cooperative mobile robots," in *Proceedings of the 9th International Workshop on Discrete Event Systems, WODES*, 2008, pp. 364–369.

[88] F. Melo, P. Lima, and M. Ribeiro, "Event-driven modelling and control of a mobile robot population," in *Proceedings of the 8th Conference on Intelligent Autonomous Systems, IAS*, 2004, pp. 237–244.

[89] R. Huq, "Behavior-based robotics control using fuzzy discrete event system," Ph.D. dissertation, Faculty of Engineering and Applied Science, Memorial Univeristy of Newfoundland, St. John's, Newfoundland, Canada, 2006.

[90] R. Huq, G. K. I. Mann, and R. R. Gosine, "Distributed fuzzy discrete event system for robotic sensory information processing," *Special Issue of Expert System on Advances in Intelligent Information Processing*, vol. 23, no. 5, pp. 273–289, 2006.

[91] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications.* Prentice Hall PTR, 1995.

[92] E. T. Lee and L. A. Zadeh, "Note on fuzzy languages," *Information Sciences*, vol. 1, no. 4, pp. 421 – 434, 1969.

[93] A. Zhu and S. X. Yang, "Neurofuzzy-based approach to mobile robot navigation in unknown environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 4, pp. 610–621, 2007.

[94] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots.* MIT Press, 2004.

[95] R. C. Arkin, "Motor schema-based mobile robot navigation," *International Journal of Robotic Research*, vol. 8, no. 4, pp. 92–112, 1989.

[96] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings of the IEEE Conference on Robotics and Automation, ICRA*, 1991, pp. 1398–1404.

[97] N. Muñoz, J. Valencia, and N. Londoño, "Evaluation of navigation of an autonomous mobile robot," in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems, PerMIS*, 2007, pp. 15–21.

[98] D. Calisi and D. Nardi, "Performance evaluation of pure-motion tasks for mobile robots with respect to world models," *Autonomous Robots*, vol. 27, no. 4, pp. 465 – 481, 2009.

[99] C. Zhou, R. Kumar, and R. S. Sreenivas, "Decentralized modular control of concurrent discrete event systems," in *Proceedings of the 46th IEEE Conference on Decision and Control, CDC*, 2007, pp. 5918–5923.

[100] K. Rohloff and S. Lafortune, "The verification and control of interacting similar discrete-event systems," *SIAM Journal on Control and Optimization*, vol. 45, no. 2, pp. 634 – 667, 2006.

[101] Y. Willner and M. Heymann, "Supervisory control of concurrent discrete-event systems," *International Journal of Control*, vol. 54, no. 5, pp. 1143–1169, 1991.

[102] K. C. Wong and W. M. Wonham, "Hierarchical control of discrete-event systems," *Discrete Event Dynamic Systems*, vol. 6, no. 3, pp. 241–273, 1996.

[103] S. G. Kim, K. H. Cho, and J. T. Lim, "Hierarchical supervisory control of discrete event systems based on h-observability," *IEE Proceedings Control Theory and Applications*, vol. 150, no. 2, pp. 179–182, 2003.

[104] V. Pantelic, S. M. Postma, and M. Lawford, "Probabilistic supervisory control of probabilistic discrete event systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 2013 –2018, 2009.