

COOPERATIVE CONTROL AND FAULT RECOVERY FOR
NETWORK OF HETEROGENEOUS AUTONOMOUS
UNDERWATER VEHICLES

MARIA ENAYAT

A THESIS
IN
THE DEPARTMENT
OF
ELECTRICAL AND COMPUTER ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF APPLIED SCIENCE IN ELECTRICAL ENGINEERING
CONCORDIA UNIVERSITY
MONTRÉAL, QUÉBEC, CANADA

APRIL 2015

© MARIA ENAYAT, 2015

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Maria Enayat**

Entitled: **Cooperative Control and Fault Recovery for Network
of Heterogeneous Autonomous Underwater Vehicles**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____	Chair
Dr. M. Zahangir Kabir	
_____	Examiner
Dr. Luis Rodrigues	
_____	Examiner
Dr. Youmin Zhang	
_____	Supervisor
Professor Khashayar Khorasani	

Approved by _____
Chair of Department or Graduate Program Director

Dean, Faculty of Engineering and Computer Science

Abstract

COOPERATIVE CONTROL AND FAULT RECOVERY FOR NETWORK OF HETEROGENEOUS
AUTONOMOUS UNDERWATER VEHICLES

Maria Enayat

The purpose of this thesis is to develop cooperative recovery control schemes for a team of heterogeneous autonomous underwater vehicles (AUV). The objective is to have the network of autonomous underwater vehicles follow a desired trajectory while agents maintain a desired formation. It is assumed that the model parameters associated with each vehicle is different although the order of the vehicles are the same.

Three cooperative control schemes based on dynamic surface control (DSC) technique are developed. First, a DSC-based centralized scheme is presented in which there is a central controller that has access to information of all agents at the same time and designs the optimal solution for this cooperative problem. This scheme is used as a benchmark to evaluate the performance of other schemes developed in this thesis.

Second, a DSC-based decentralized scheme is presented in which each agent designs its controller based on only its information and the information of its desired trajectory. In this scheme, there is no information exchange among the agents in the team. This scheme is also developed for the purpose of comparative studies.

Third, two different semi-decentralized or distributed schemes for the network of heterogeneous autonomous underwater vehicles are proposed. These schemes are a synthesis of a consensus-based algorithm and the dynamic surface control technique with the difference that in one of them the desired trajectories of agents are used in the consensus algorithm while in the other the actual states of the agents are used. In the former scheme, the agents communicate their desired relative distances with the agents within their set of nearest neighbors and each agent determines its own control trajectory. In this semi-decentralized scheme, the velocity measurements of the virtual leader and all the followers are not required to reach the consensus formation. However, in the latter, agents communicate their relative distances and velocities with the agents within their set of nearest neighbors. In both semi-decentralized schemes only a subset of agents has access to information of a virtual leader. The comparative studies between these two semi-decentralized schemes are provided which show the superiority of the former semi-decentralized scheme over latter.

Furthermore, to evaluate the efficiency of the proposed DSC-based semi-decentralized scheme with consensus algorithm using desired trajectories, a comparative study is performed between this scheme and three cooperative schemes of model-dependent coordinated tracking algorithm, namely the centralized, decentralized, and semi-decentralized schemes.

Given that the dynamics of autonomous underwater vehicles are inevitably subjected to system faults, and in particular the actuator faults, to improve the performance of the network

of agents, active fault-tolerant control strategies corresponding to the three developed schemes are also designed to recover the team from the loss-of-effectiveness in the actuators and to ensure that the closed-loop signals remain bounded and the team of heterogeneous autonomous underwater vehicles satisfy the overall design specifications and requirements.

The results of this research can potentially be used in various marine applications such as underwater oil and gas pipeline inspection and repairing, monitoring oil and gas pipelines, detecting and preventing any oil and gas leakages. However, the applications of the proposed cooperative control and its fault-tolerant scheme are not limited to underwater formation path-tracking and can be applied to any other multi-vehicle systems that are characterized by Euler–Lagrange equations.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Professor Khorasani for the continuous support of my research, for his patience, motivation, enthusiasm, and immense knowledge. This thesis could not be accomplished without his invaluable and technical guidance.

In my daily work I have been blessed with a friendly and cheerful group of fellow students. I would like to thank my fellows in Systems and Control Laboratory: Amin Salar, Esmail Alizadeh, Mahsa Khoshab, Niloofar Ashtiani, and Niusha Modabbernia, for the stimulating discussions, for the sleepless nights we were working together, and for all the fun we have had in these years. My sincere thanks also go to Hossein Khodadadi and Mohammad Khosravi for their help and the thoughtful discussions we have had.

Last but not the least, I would like to thank my loved ones in particular my parents, my sister and my brother for their unconditional love and inspiration and for supporting me spiritually throughout my life. None of this would have been possible without their encouragement, and to them I dedicate this thesis.

Contents

Abstract	iii
List of Figures	xi
List of Tables	xvii
List of Acronyms	xxiii
List of Notations	xxii
1 Introduction	1
1.1 Introduction	1
1.2 Literature Review	2
1.2.1 Control of Homogeneous and Heterogeneous Multi-agent Systems	2
1.2.2 Dynamic Surface Control (DSC)	5
1.2.3 Control of a Single Marine Vehicle	6
1.2.4 Control of a Group of Marine Vehicles	9
1.2.5 Fault-tolerant Control of Underwater Vehicles	14
1.3 Statement of the Problem and Thesis Contributions	17
1.4 Thesis Outline	19
2 Background Information	21
2.1 Preliminaries of Nonlinear Systems	22
2.2 Dynamics and Modeling of Underwater Vehicles	23
2.2.1 Coordinate Frames	23
2.2.2 Full Order Model of Underwater Vehicles	26

2.2.2.1	Kinematics	26
2.2.2.2	Dynamics (Kinetics)	26
2.2.3	Reduced Order Model of Underwater Vehicles	27
2.2.4	Equations of Motion of Underwater Vehicles with Uncertainties	29
2.3	Dynamic Surface Control	31
2.3.1	Introduction	31
2.3.2	Design Procedure for a Class of Nonlinear Systems	32
2.3.3	Dynamic Surface Control of an Autonomous Underwater Vehicle	36
2.4	Cooperative Multi-agent Systems	38
2.4.1	Preliminaries of Graph Theory	42
2.5	Fault-tolerant Control Systems	44
2.6	Model-dependent Coordinated Tracking Algorithm for Networked Euler-Lagrange Systems	47
2.7	Conclusion	49
3	DSC-based Cooperative Schemes for Multiple Autonomous Underwater Vehicles	50
3.1	Introduction	51
3.2	Design Specification	53
3.3	DSC-based Centralized Control Scheme	58
3.4	DSC-based Decentralized Control Scheme	60
3.4.1	Stability Analysis	61
3.5	DSC-based Semi-Decentralized Control Scheme with Consensus Algorithm Using States of Agents	62
3.5.1	Stability Analysis	63

3.6	DSC-based Semi-Decentralized Control Scheme with Consensus Algorithm Using Desired Trajectories	66
3.6.1	Stability Analysis	67
3.7	Simulation Results	68
3.7.1	Using Actual States vs. Using Desired Trajectories in Consensus Algo- rithm for DSC-based Semi-decentralized Control Scheme	70
3.7.2	Quantitative Comparison of DSC-based Centralized, Decentralized, and Semi-decentralized Schemes	78
3.7.3	DSC-based Semi-decentralized Scheme vs. Cooperative Schemes Based on Model-dependent Coordinated Tracking Algorithm in Fault-free Situ- ation	87
3.7.4	Simulation Scenarios for Semi-decentralized DSC-based Scheme	98
3.7.4.1	Scenario 3.1: Different Group Size	98
3.7.4.2	Scenario 3.2: Different Network Topologies	108
3.7.4.3	Scenario 3.3: Different Reference Trajectories	114
3.7.4.4	Scenario 3.4: Different Initial Values	120
3.8	Conclusion	128
4	DSC-based Cooperative Fault-tolerant Control Schemes	130
4.1	Fault-tolerant Control for an AUV Using DSC Technique	131
4.2	Cooperative FTC Scheme for Multiple AUVs Using DSC Technique	136
4.2.1	Fault-tolerant DSC-based Centralized Control Scheme	136
4.2.2	Fault-tolerant DSC-based Decentralized Control Scheme	137
4.2.2.1	Stability Analysis	138
4.2.3	Fault-tolerant DSC-based Semi-Decentralized Control Scheme with Con- sensus Algorithm Using Desired Trajectories	140
4.2.3.1	Stability Analysis	141

4.3	Simulation Results	143
4.3.1	Semi-decentralized DSC-based Scheme in Faulty Situation Without Compensation Control	144
4.3.2	Quantitative Comparison of DSC-based Fault-tolerant Centralized, Decentralized, and Semi-decentralized Schemes	149
4.3.3	Simulation Scenarios for Semi-decentralized Fault-tolerant Scheme	160
4.3.3.1	Scenario 4.1: Number of Faulty Agents	160
4.3.3.2	Scenario 4.2: Time Delay in Fault Detection	176
4.3.3.3	Scenario 4.3: Error in Estimation of the Fault Severity	177
4.3.3.4	Scenario 4.4: Error in the Fault Isolation Decision	190
4.4	Conclusion	202
5	Conclusions and Future Work	204
	References	209

List of Figures

2.1	Body-fixed and inertial reference frames of an AUV [112].	24
2.2	Block diagram of the first-order filter.	35
2.3	Types of actuator faults [120].	45
2.4	Classification of fault tolerant control systems.	46
3.1	From left to right, centralized, decentralized, and semi-decentralized cooperative schemes, where the white and dark boxes represent agents and controllers, respectively [122].	51
3.2	Network topology for a group of six vehicles with a virtual leader.	69
3.3	Error signals of agents #1 and #2 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm. .	72
3.4	Error signals of agents #3 and #4 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm. .	73
3.5	Error signals of agents #5 and #6 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm. .	73
3.6	Control efforts of agent #1 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.	74
3.7	Control efforts of agent #2 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.	74
3.8	Control efforts of agent #3 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.	75

3.9	Control efforts of agent #4 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.	75
3.10	Control efforts of agent #5 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.	76
3.11	Control efforts of agent #6 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.	76
3.12	Error signals of agents #1 and #2 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.	80
3.13	Error signals of agents #3 and #4 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.	80
3.14	Error signals of agents #5 and #6 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.	81
3.15	Control effort of agent #1 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.	81
3.16	Control effort of agent #2 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.	82
3.17	Control effort of agent #3 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.	82
3.18	Control effort of agent #4 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.	83
3.19	Control effort of agent #5 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.	83
3.20	Control effort of agent #6 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.	84
3.21	Position trajectories of agents #1 and #2 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.	94

3.22	Zoomed version of position trajectories of agents #1 and #2 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated schemes.	94
3.23	Position trajectories of agents #3 and #4 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.	95
3.24	Zoomed version of position trajectories of agents #3 and #4 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated schemes.	95
3.25	Position trajectories of agents #5 and #6 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.	96
3.26	Zoomed version of position trajectories of agents #5 and #6 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated schemes.	96
3.27	Control input signals of agents #1 and #2 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.	97
3.28	Control input signals of agents #3 and #4 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.	97
3.29	Control input signals of agents #5 and #6 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.	98
3.30	Error signals of agents #1, #2, and #3 in fault-free situation for scenario 3.1, case 1.	101

3.31 Error signals of agents #1, #2, and #4 in fault-free situation for scenario 3.1, case 2.	101
3.32 Error signals of agents #2, #3, and #5 in fault-free situation for scenario 3.1, case 3.	102
3.33 Error signals of agents #2, #4, and #6 in fault-free situation for scenario 3.1, case 4.	102
3.34 Error signals of agents #3, #5, and #7 in fault-free situation for scenario 3.1, case 5.	103
3.35 Error signals of agent #1 and #2 in fault-free situation for the centralized scheme and three semi-decentralized cases of scenario 3.2.	110
3.36 Error signals of agent #3 and #4 in fault-free situation for the centralized scheme and three semi-decentralized cases of scenario 3.2.	111
3.37 Error signals of agent #5 and #6 in fault-free situation for the centralized scheme and three semi-decentralized cases of scenario 3.2.	111
3.38 Error signals of agents #1, #2, and #3 in fault-free situation for the semi- decentralized and centralized schemes for scenario 3.3, case 1.	117
3.39 Error signals of agents #4, #5, and #6 in fault-free situation for the semi- decentralized and centralized schemes for scenario 3.3, case 1.	117
3.40 Error signals of agents #1, #2, and #3 in fault-free situation for the semi- decentralized and centralized schemes for scenario 3.3, case 2.	118
3.41 Error signals of agents #4, #5, and #6 in fault-free situation for the semi- decentralized and centralized schemes for scenario 3.3, case 2.	118
3.42 Error signals of agents #1, #2, and #3 in fault-free situation for the semi- decentralized and centralized schemes for scenario 3.3, case 3.	119
3.43 Error signals of agents #4, #5, and #6 in fault-free situation for the semi- decentralized and centralized schemes for scenario 3.3, case 3.	119

3.44	Error signals of agents #1, #2, and #3 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 1.	123
3.45	Error signals of agents #4, #5, and #6 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 1.	123
3.46	Error signals of agents #1, #2, and #3 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 2.	124
3.47	Error signals of agents #4, #5, and #6 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 2.	124
3.48	Error signals of agents #1, #2, and #3 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 3.	125
3.49	Error signals of agents #4, #5, and #6 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 3.	125
4.1	Block diagram of an active fault-tolerant control scheme of a system.	132
4.2	Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 1.	146
4.3	Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 2.	146
4.4	Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 3.	147
4.5	Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 4.	147
4.6	Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 5.	148
4.7	Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 6.	148
4.8	Tracking error signals of agents #1 and #2 in faulty situation for all fault-tolerant DSC-based cooperative schemes.	151
4.9	Tracking error signals of agents #3 and #4 in faulty situation for all fault-tolerant DSC-based cooperative schemes.	151
4.10	Tracking error signals of agents #5 and #6 in faulty situation for all fault-tolerant DSC-based cooperative schemes.	152
4.11	Control input signals of agents #1 and #2 in faulty situation for all fault-tolerant DSC-based cooperative schemes.	152

4.12	Control input signals of agents #3 and #4 in faulty situation for all fault-tolerant DSC-based cooperative schemes.	153
4.13	Control input signals of agents #5 and #6 in faulty situation for all fault-tolerant DSC-based cooperative schemes.	153
4.14	Tracking error signals of agents #1, #2, and #3 for scenario 4.1, case 1.	170
4.15	Tracking error signals of agents #4, #5, and #6 for scenario 4.1, case 1.	170
4.16	Tracking error signals of agents #1, #2, and #3 for scenario 4.1, case 2.	171
4.17	Tracking error signals of agents #4, #5, and #6 for scenario 4.1, case 2.	171
4.18	Tracking error signals of agents #1, #2, and #3 for scenario 4.1, case 3.	172
4.19	Tracking error signals of agents #4, #5, and #6 for scenario 4.1, case 3.	172
4.20	Tracking error signals of agents #1 and #2 for scenario 4.1, case 4.	173
4.21	Tracking error signals of agents #3 and #4 for scenario 4.1, case 4.	173
4.22	Tracking error signals of agents #1 and #2 for scenario 4.1, case 5.	174
4.23	Tracking error signals of agents #3 and #4 for scenario 4.1, case 5.	174
4.24	Tracking error signals of agents #1 and #2 for scenario 4.1, case 6.	175
4.25	Tracking error signals of agents #3 and #4 for scenario 4.1, case 6.	175
4.26	Tracking error signals of faulty agents for scenario 4.3, case 1.	187
4.27	Tracking error signals of faulty agents for scenario 4.3, case 2.	187
4.28	Tracking error signals of faulty agents for scenario 4.3, case 3.	188
4.29	Tracking error signals of faulty agents for scenario 4.3, case 4.	188
4.30	Tracking error signals of faulty agents for scenario 4.3, case 5.	189
4.31	Tracking error signals of faulty agents for scenario 4.3, case 6.	189
4.32	Tracking error signals of agents #1, #2, and #3 for scenario 4.4, case 1.	198
4.33	Tracking error signals of agents #4, #5, and #6 for scenario 4.4, case 1.	198
4.34	Tracking error signals of agents #1, #2, and #3 for scenario 4.4, case 2.	199
4.35	Tracking error signals of agents #4, #5, and #6 for scenario 4.4, case 2.	199
4.36	Tracking error signals of agents #1, #2, and #3 for scenario 4.4, case 3.	200
4.37	Tracking error signals of agents #4, #5, and #6 for scenario 4.4, case 3.	200
4.38	Tracking error signals of all agents for scenario 4.4, case 4.	201
4.39	Tracking error signals of all agents for scenario 4.4, case 5.	201

List of Tables

2.1	Motion components used for marine vehicles	25
2.2	Types of actuator faults [120]	44
3.1	Initial conditions	70
3.2	Team level RMS analysis of maximum error, steady state error, and maximum absolute control effort for semi-decentralized schemes using desired trajectories and states of agents in consensus algorithm	72
3.3	Quantitative analysis of maximum errors, steady state errors, and maximum absolute control efforts for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm	77
3.4	Team level RMS analysis of rise time, settling time, steady state error, maximum overshoot, and maximum and minimum control effort for all schemes of Section 3.7.2 in fault-free situation	79
3.5	Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #1	84
3.6	Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #2	85

3.7	Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #3	85
3.8	Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #4	86
3.9	Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #5	86
3.10	Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #6	87
3.11	Team level RMS analysis of settling time, maximum error, and maximum absolute control effort of DSC-based semi-decentralized and model-dependent coordinated tracking schemes in healthy situation	90
3.12	Quantitative analysis of settling time of the DSC-based semi-decentralized scheme and all cooperative schemes based on model-dependent coordinated tracking algorithm in fault-free situation for all agents	91
3.13	Quantitative analysis of maximum error of the DSC-based semi-decentralized scheme and all cooperative schemes based on model-dependent coordinated tracking algorithm in fault-free situation for all agents	92
3.14	Quantitative analysis of maximum absolute control effort of the DSC-based semi-decentralized scheme and all cooperative schemes based on model-dependent coordinated tracking algorithm in fault-free situation for all agents	93
3.15	Different number of agents in the group for scenario 3.1	99

3.16	Team level RMS analysis of maximum error and steady state error for semi-decentralized and centralized schemes in fault-free situation for scenario 3.1 . . .	100
3.17	Response characteristics of the semi-decentralized and centralized schemes for scenario 3.1, case 1	103
3.18	Response characteristics of the semi-decentralized and centralized schemes for scenario 3.1, case 2	104
3.19	Response characteristics of the semi-decentralized and centralized schemes for scenario 3.1, case 3	105
3.20	Response characteristics of the semi-decentralized and centralized schemes for scenario 3.1, case 4	106
3.21	Response characteristics of the semi-decentralized and centralized schemes for scenario 3.1, case 5	107
3.22	Different network topology for scenario 3.2	109
3.23	Team level RMS analysis of maximum error and steady state error for semi-decentralized and centralized schemes in fault-free situation for scenario 3.2 . . .	110
3.24	Maximum errors of all cases of the semi-decentralized and centralized schemes for scenario 3.2	112
3.25	Steady state errors of all cases of the semi-decentralized and centralized schemes for scenario 3.2	113
3.26	Different reference trajectories for scenario 3.3	114
3.27	Maximum errors of all cases of the semi-decentralized and centralized schemes for scenario 3.3	115
3.28	Steady state errors of all cases of the semi-decentralized and centralized schemes for scenario 3.3	116
3.29	Team level RMS analysis of maximum error and steady state error for semi-decentralized and centralized schemes in fault-free situation for scenario 3.3 . . .	120
3.30	Different Initial conditions for scenario 3.4	122

3.31	Maximum errors of all cases of the semi-decentralized and centralized schemes for scenario 3.4	126
3.32	Steady state errors of all cases of the semi-decentralized and centralized schemes for scenario 3.4	127
3.33	Team level RMS analysis of maximum error and steady state error for semi- decentralized and centralized schemes in fault-free situation for scenario 3.4 . . .	128
4.1	Cases of faulty situation without compensation control	145
4.2	Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #1	154
4.3	Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #2	155
4.4	Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #3	156
4.5	Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #4	157
4.6	Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #5	158
4.7	Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #6	159
4.8	Team level RMS analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for Section 4.3.2	160
4.9	Cases of scenario 4.1	162
4.10	Team level RMS analysis of maximum error after t_f (m) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.1	163
4.11	Team level RMS analysis of time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.1 . . .	163

4.12	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.1 case 1	164
4.13	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.1 case 2	165
4.14	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.1 case 3	166
4.15	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.1 case 4	167
4.16	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.1 case 5	168
4.17	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.1 case 6	169
4.18	Time delay in detection	177
4.19	Different error in estimation for scenario 4.3	179
4.20	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.3 case 1	180
4.21	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.3 case 2	181
4.22	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.3 case 3	182
4.23	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.3 case 4	183
4.24	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.3 case 5	184
4.25	Response characteristics of the fault-tolerant semi-decentralized and centralized schemes for scenario 4.3 case 6	185

4.26	Team level RMS analysis of maximum error after t_f (m) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.3	186
4.27	Team level RMS analysis of time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.3	186
4.28	Different error in isolation for scenario 4.4	191
4.29	Quantitative analysis of the semi-decentralized and centralized schemes in faulty situation for scenario 4.4 case 1	192
4.30	Quantitative analysis of the semi-decentralized and centralized schemes in faulty situation for scenario 4.4 case 2	193
4.31	Quantitative analysis of the semi-decentralized and centralized schemes in faulty situation for scenario 4.4 case 3	194
4.32	Quantitative analysis of the semi-decentralized and centralized schemes in faulty situation for scenario 4.4 case 4	195
4.33	Quantitative analysis of the semi-decentralized and centralized schemes in faulty situation for scenario 4.4 case 5	196
4.34	Team level RMS analysis of maximum error after t_f (m) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.4	197
4.35	Team level RMS analysis of time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.4	197

List of Acronyms

AFTCS	Active Fault-Tolerant Control Systems.
AI	Artificial Intelligence.
AUV	Autonomous Underwater Vehicles.
DOF	Degrees of Freedom.
DSC	Dynamic Surface Control.
EKF	Extended Kalman Filter.
FDI	Fault Detection and Isolation.
FTCS	Fault-Tolerant Control Systems.
HJI	Hamilton–Jacobi–Isaacs.

LOE	Loss-of-Effectiveness.
LS	Least Squares.
MIMO	Mmulti-input Multi-output.
MPC	Model Predictive Control.
MSS	Multiple Sliding Surface.
PFTCS	Passive Fault-Tolerant Control Systems.
RMS	root mean squares.
ROV	Remotely Operated Vehicle.
SGA	Successive Galerkin Approximation.
UGV	Unmanned Ground Vehicle.

List of Notations

η	Position and orientation vector with coordinates in the earth-fixed frame
ν	Linear and angular velocity vector with coordinates in the body-fixed frame
τ	Forces and moments acting on the vehicle in the body-fixed frame
$R(\eta)$	transformation matrix
M	Inertia matrix
$C(\nu)$	Matrix of Coriolis and centripetal terms
$D(\nu)$	Damping matrix
$\Delta f(\nu)$	Vector of unknown nonlinear uncertainties
ρ	Upper bound of $\Delta f(\nu)$
S_1	First error surface
S_2	Second error surface
γ	Filter time constant
λ_n	Gain matrix

ε	Nonlinear damping constant
V_G	Node set,
E_G	Set of edges
A_G	Associated adjacency matrix
a_{ij}	Entries of associated adjacency matrix
N_j	Set of neighbors of node j
D	In-degree matrix
L	Laplacian matrix
I_m	$m \times m$ identity matrix
$\mathbf{0}_3$	$m \times m$ zero matrix
x_{1d}	Desired position
σ^{ij}	Desired formation positioning between i^{th} and j^{th} vehicles
$\ \cdot\ $	Euclidean norm of a vector
\otimes	Kronecker product
F	Actual effectiveness coefficient matrix
f_j	Actual loss of control effectiveness faults of j^{th} agent
\hat{F}	Estimated effectiveness coefficient matrix
\hat{f}_j	Estimated loss of control effectiveness faults of j^{th} agent
τ_F	Fault-tolerant control input
τ_C	Compensation control input
τ_H	Healthy controller

Chapter 1

Introduction

1.1 Introduction

In recent years, cooperative networks of underwater vehicles have largely attracted attention due to their extensive application and efficiency to increase performance. A network of vehicles can do more tasks in the same amount of time in comparison to an individual vehicle working alone and it can also distribute the computational required load of tasks. Moreover, such systems can benefit from increased success in a mission in case one vehicle in the group becomes disabled while performing tasks since other vehicles could rapidly compensate the loss, instead of a total mission failure. That is why in this thesis the problem of cooperative control for a network of multiple agents via nonlinear control methods in healthy and faulty situations is considered.

1.2 Literature Review

This section reviews the literature on heterogeneous and homogeneous multi-agent systems, single and cooperative control schemes of marine vehicles, dynamic surface control technique, and fault-tolerant control methods for the application of underwater vehicles.

1.2.1 Control of Homogeneous and Heterogeneous Multi-agent Systems

In the literature, most of the works conducted in the area of consensus and formation problem of multi-agent systems mainly try to stabilize homogeneous systems. Homogeneous systems are systems of multiple agents in which all agents have the same internal architecture. Consensus problems for homogeneous nonlinear systems have been addressed in [1–6], to name a few. In case of the formation problem of the homogeneous nonlinear systems, [7–10] have addressed various techniques and methodologies.

On the other hand, there are another type of systems known as heterogeneous systems in which the agents might vary in different aspects such as ability, structure, or functionality. The cooperative control of heterogeneous multi-agent systems is one of the open areas of research and in comparison to homogeneous systems, it has been the topic of fairly few works. Up to now, some works have been conducted for different types of heterogeneous multi-agent systems under various constraints. In this part, some of the most interesting recent works are introduced.

In the literature, works have been conducted for the consensus problem of heterogeneous uncertain linear multi-agent systems such as [11] and [12]. The consensus problem of a heterogeneous multi-agent system containing agents with first-order and second-order integrator models is considered in [13] in which the velocities of second-order integrator agents are decided by the control input of first-order integrator agent. The reference [14] solved the same problem

as in [13] problem by applying the graph theory and the Lyapunov direct method, in which the second-order integrator agents cannot get the velocity measurements form feedback.

The finite-time consensus problem for heterogeneous multi-agent systems consist of first-order and second-order agents is addressed in [15] using a novel continuous nonlinear distributed consensus protocol. In [16], the consensus algorithm of multi-agent second-order systems with non-symmetric interconnection and heterogeneous delays is studied, where the generalized Nyquist criterion is applied.

The reference [17] considered the leader-follower consensus problem of heterogeneous multi-agent systems by taking to account a fuzzy disturbance observer with an adaptive control method based on the Lyapunov stability theory to compensate the observation error which is caused by the discrepancy between the unknown factor and the estimated values.

The reference [18] deals with the problem of consensus control for a multi-agent system with heterogeneous nonlinear subsystem dynamics. In this paper, the objective is that outputs of the subsystems follow a desired trajectory which is a function of an exosystem state. By taking advantages of the internal model design strategy, a consensus control design which uses the relative outputs is proposed to ensure that the outputs of all the subsystems converge to the predefined desired output trajectory.

In [19] the consensus problem is conceived for a class of nonlinear and heterogeneous systems. It is assumed that the topology of the communication network has the possibility to change in an arbitrary and intermittent manner. A matrix-theoretical approach is applied to find the necessary and sufficient condition of cooperative controllability, and then this condition is used to search for cooperative control Lyapunov function for linear cooperative systems.

Authors in [20] studied second-order consensus problem of heterogeneous nonlinear multi-

agent systems with time-varying delays by introducing novel decentralized adaptive strategies to both the coupling strengths and the feedback gains, based on the Lyapunov stability theory. In another work in this domain, the network consensus problem for a multi-agent system consists of agents with heterogeneous fractional-order nonlinear dynamics that can be split into several sub-groups based on their dynamics and equilibriums is investigated in [21].

In [22], H_∞ almost output synchronization of multi-agent systems with linear, right-invertible, and introspective agents with non-identical dynamics in exposure of external disturbances and under directed interconnection structures is addressed. The reference [23] addressed finite-time consensus for heterogeneous multi-agent systems composed of agents with mixed orders over fixed and switching topology. In this paper, the design of the finite-time consensus protocol is based on graph theory, matrix theory, and LaSalle's invariance principle.

Output synchronization and regulation problem of a network of heterogeneous introspective discrete-time right-invertible agents is studied in [24]. The reference [25] considered the high-order consensus problem for heterogeneous multi-agent systems with unknown communication delays. In this paper, the model of the agents is considered as transfer function in Laplace domain.

Authors of [26] considered control design for distributed heterogeneous systems in which the controller is designed to obtain and keep the distributed spatial structure of the nominal system. In another study, a navigation and stabilization layout for 3-degrees of freedom (DOF) formation of heterogeneous UAVs and unmanned ground vehicles (UGV) acting under a hawk-eye like relative localization is investigated in [27]. In this paper, a novel model predictive control (MPC) based method is applied for formation keeping in a leader-follower constellation into a desired target area.

Among all works conducted in this area, a few papers considered heterogeneous cases of the synchronization problem. Particularly, [28] and [29] studied the output synchronization problem of heterogeneous agents under nonlinear approaches. Recent results are mostly limited to heterogeneous linear dynamical systems as in [30–32]. The authors of [30] used an internal model approach to address the linear output synchronization of heterogeneous agents. The same problem is investigated in [32] considering the agents with uncertainties.

1.2.2 Dynamic Surface Control (DSC)

Dynamic surface control technique has been used to control several systems including automated cars [33], high maneuver missiles [34], flexible-joint robots [35], DC motor servomechanism [36], quadrotors [37], and four-state model of bicycles [38]. This technique has been also applied on marine vehicles such as remotely operated underwater vehicles [39], ships [40, 41], marine shaft system [42], overactuated ocean surface vessels [43], etc.

An adaptive dynamic surface control based method for a class of multi-input multi-output (MIMO) nonlinear systems with lock-in-place actuator faults and uncertainties is presented in [44]. Another adaptive scheme of dynamic surface control technique is presented in [45] and [46] for a class of time-delay nonlinear systems. In [47], adaptive DSC technique for air speed and flight path angle control are addressed for the application of the longitudinal dynamics of a flexible hypersonic flight vehicle. In another study, the neural network based adaptive dynamic surface control for the problem of trajectory tracking for a fully-actuated autonomous underwater vehicle is investigated in [48].

In [49], dynamic surface sliding control and hybrid systems are combined for the problem of dynamic positioning of ocean vehicles. Also, in [50–52] the leader-follower cooperative control algorithms based on neural networks and DSC technique are presented for the problem of

formation and target tracking problems of a group of multiple agents.

1.2.3 Control of a Single Marine Vehicle

For decades, the problem of controlling under-actuated marine vehicles has attracted many attentions. First attempts of tracking control of under-actuated marine vehicles using nonlinear models are presented in [53, 54] where, under the assumption that the forward velocity is positive, global exponential position tracking is given using controllers derived from feedback linearization and backstepping. Since only two degrees of freedom is considered, full state tracking control is not obtained. A generalization of [54] to consist different types of forces is presented in [55] where vectorial backstepping method is employed.

The reference [56] addressed a control method using linearization and high-gain control method that cause the global exponential stability of the position trajectories. Also, full state tracking controller for under-actuated marine vehicle is addressed in [57] in order to achieve global exponential practical stability which means an arbitrarily small neighborhood of the reference trajectory is globally exponentially stable.

The reference [58] used a coordinate transformation to transform the model into a triangular form and designed a controller using the recursive method of integrator backstepping for local exponential stability of full-state tracking. These results can be extended to semi-global exponential stability for specified conditions.

In [59], a tracking controller which is in the form of full-state is addressed for under-actuated marine vehicles with diagonal inertial and damping matrices using the theory of cascaded systems that causes globally exponentially stability in case that the reference yaw velocity is being constantly stimulated which is not always applicable in practice.

Based on [59], the reference [60] studied global asymptotic tracking of under-actuated marine

vehicles using Lyapunov and passivity methods which allows a precise construction of Lyapunov functions and comforts designs which are robust and adaptive.

Same as [59], in order to achieve exponential tracking, The reference [58–60] proposed a controller which require a desired yaw velocity that is constantly exciting. [61] proposed a global tracking control method to satisfy global K –exponential stability of the tracking error dynamics under less strict persistently conditions than [59] which is capable to reach global K –exponential convergence for a desired straight-line path.

In [62], the stimulation limitations applied in [58–60] is omitted by being replaced with nonzero desired yaw velocity. Also, the reference trajectory can be a considered as a curve including straight line. Using this solution, the reference [63] presented the first universal time-varying controller which simultaneously dissolve stabilization and tracking problems. In this paper, the nonlinear damping terms are neglected and system has a diagonal matrices.

A high-gain continuous time-varying controller is provided in [64] to obtain globally uniformly ultimately bounded regulation and tracking for under-actuated marine vehicles. In this paper, similar to [63], regulation problem is treated as a subclass of tracking problem. In [65], an adaptive controllers to approximate uncertain hydrodynamic parameters is addressed. Also, a robust adaptive controller for path-tracking problem of under-actuated marine vehicles is proposed by [66] in which matrices are assumed to be diagonal.

The reference [67] propose a path-following controller for under-actuated marine vehicles. The objective is to stabilize the under-actuated sway-yaw dynamics using only one controller while the vehicle has a invariant speed. In another work, a robust adaptive control is addressed in [68] where it is designed based on parametric uncertainties and errors of state measurements.

Authors of [69] and [70] proposed path-following controllers for two degrees of freedom non-

linear models with diagonal matrices with this assumption that the forward speed is invariant. The reference [69] utilized the sliding model method while [70] utilized feedback linearization with collaboration of the line-of-sight guidance system to reach a straight-line path-following controller that obtains globally asymptotically stability of the error dynamics. In this paper, it is proven that the guidance parameters have a great effect on the stability of the closed-loop system, and the guidance parameters should be designed based on the velocity of the vehicle and the model parameters.

In [71], backstepping method and Serret-Frenet frame are used for the problem of two dimension path-following of straight lines and circles in presence of constant ocean currents to define the error dynamics. Using this method, the local convergence is achieved. To tackle irrotational ocean currents, a current estimator is proposed. The reference [72] considered the same problem with three dimension path-following assumption.

Inspired by [71], the reference [73] proposed a novel controller for path-following of three degrees of freedom under-actuated marine vehicle using the Serret-Frenet frame to find the dynamics of geometric error. The novelty of this work is that the assumption which indicates that initial position error must be smaller than the smallest radius of curvature present in the path is relaxed in comparison to [74] which primarily proposed the same method for path-following of ground robots. This restriction is resolved by regulating the velocity of the Serret-Frenet frame which moves on the path. An extension of this work is provided in [75] where robust adaptive scheme is adopted.

An adaptive switching supervisory control method is used in [76] in order to solve the problem of global boundedness and convergence of the position tracking and path-following error to an arbitrarily neighborhood of the origin to overcome the large and abrupt model

parametric uncertainties.

Many works conducted in this area have considered diagonal matrices in model of the system which is a valid assumption that the marine vehicle possesses three planes of symmetry or off-diagonal element are negligible in comparison to the diagonal ones. However, in reality, many marine vehicles have port-starboard symmetry, but do not have fore-aft symmetry. Therefore, this assumption of diagonal matrices is not always realistic. In this case, [77] considered a two degrees of freedom model of under-actuated marine vehicles with constant velocity where mass and linear damping matrices are not considered as diagonal. Under certain conditions on system parameters, the globally asymptotically and locally exponentially stabilization of the error system at the origin is proved using a straight-line path-following controller. The same problem for three degrees of freedom model of marine vehicles is addressed in [78].

Authors of [79] used the vectorial backstepping method to solve the problem of a two-dimensional guidance-based straight-line path-following controller for three degrees of freedom model of marine vehicles with nonzero off-diagonal terms in the system matrices. To cope with under-actuation, a dynamical term is added to the controller which increases the order of the closed-loop system. In this paper, the convergence to the path is not proved formally despite of the fact that it guarantees global ultimate boundedness of the sway velocity.

1.2.4 Control of a Group of Marine Vehicles

Besides all the works conducted on the topic of controlling the individual marine vehicles, the problem of cooperative control of a group of marine systems is also addressed in the literature, as will be discussed in this section. Cooperative control is to control a group of individual agents in a way that they accomplish one or several objectives by cooperating with each other inside the group. Most of the applications of cooperative control of groups of unmanned underwater

vehicles contain formation control and motion coordination. Recently, formation control of marine vehicles has attracted lots of attention among many others. The objective of formation control is to design a controller that makes the agents of a group move in a desired geometric shape, for instance circular shape. One of the approaches to solve this problem is the leader-follower structure in which one or some agents are assigned as leaders which send informations to other agents which are known as followers. This section mostly focuses on the topic of cooperative formation control of multiple agents based on leader-follower structure.

One of the well-known applications of AUVs is to employ large numbers of AUVs to perform dangerous tasks such as mine-sweeping. In [80], authors addressed a leader-follower formation control algorithm which can be applied to up to three dimensional formations and it consists of both trajectory and formation control algorithms. The proposed algorithm is robust and the only acoustic communication that it requires is an intermittent broadcast from the leader vehicle.

In [81], in order to cope with the problem of the formation tracking of cooperative control of multiple AUVs, a variable structure control law is presented to keep the AUVs track along the desired trajectory by minimizing the cross track error which is calculated from the line-of-sight angle. Furthermore, in this paper, a mathematical model for desired formation based on leader-follower scheme is established and the relative formation control method based on feedback linearization is addressed.

The reference [82] presented a leader-follower formation control for autonomous underwater vehicles. First, the relative equations of two AUVs containing one leader and one follower are modeled. Then the relative velocity vector is projected on two directions, along connection line between them which is perpendicular to it. Considering the disturbance in underwater

environment, the system equations of relative movement is established, and then using feedback linearization the system equations are converted into linear equations with a disturbance. At the end, the feed forward and feedback optimal control laws of the obtained linear system is designed to compensate the effect of disturbance.

A new method based on the Jacobi shape theory and geometric reduction for formation control of AUVs is addressed in [83]. For the horizontal motion of each AUV, a 3-DOF dynamic model that has control inputs over surge force and yaw moment is considered. The horizontal dynamics of AUVs are given as dynamics for three cases as formation motion, formation shape, and vehicle orientation by using the Jacobi transform. In this study, when additional symmetries in vehicle design occurs, the system decouples. Therefore, controllers of all three cases can be designed individually.

In [84], the successive Galerkin approximation (SGA) method has been applied to the nonlinear formation control for a class of multiple AUVs which have the model of four-input driftless nonlinear chained systems. Since SGA approach is developed for time-invariant nonlinear control systems, to make it applicable to the fundamentally time-varying formation control problems, a nonlinear change of coordinates and feedback has been presented in first step. Afterward, the nonlinear optimal and robust controls are synthesized by dissolving the associated Hamilton–Jacobi–Isaacs (HJI) equation with the aid of SGA algorithm.

Authors of [85] addressed a feasible navigation and control approach for a formation of a heterogeneous group of AUVs and an autonomous surface craft. The proposed specific heterogeneous solution was specified by its environment perception capabilities and directly related to the developed navigation system. The proposed approach provides the low cost implementation of a network of autonomous underwater vehicles coordination control for many oceanographic

missions.

The reference [86] addressed a formation control method for a network of AUVs using region control concept. The desired formation is defined by the shape of the selected area where all AUVs are required to stay inside. In order that the members of the group do not collide, minimum distance between robots is defined in advance. Using this region control method, it has been shown that a simple PD controller along gravity compensation controller can provide a good performance for a group of multiple 6-DOF autonomous underwater vehicles moving in a desired formation.

A time-varying, smooth feedback control law that gives asymptotic convergence to the origin of the formation error dynamics for a nonlinear formation-keeping control of multiple nonholonomic autonomous underwater vehicles is proposed in [87]. The proposed formation-keeping control law is based on a nonlinear coordinate change and the Lyapunov direct design scheme. Furthermore, a continuous, time-varying feedback control law with asymptotic stability is proposed by using the integrator backstepping technique to settle the follower dynamically to its desired relative docking position and orientation with respect to the leader which is the second contribution of this study.

In [88], authors investigated the leader-follower formation control of under-actuated AUVs. By using position measurements from the leader, a virtual vehicle is designed such that its trajectory converges to the reference trajectory of the follower. Also, for the follower a position tracking control is proposed so that it tracks the virtual vehicle using the combination of Lyapunov and backstepping methods.

The reference [89] has presented an observer-based robust finite-time consensus control scheme for leader-follower multi-agent systems using multiple-surface sliding mode observer

which ensure finite-time consensus under the condition that only the agents in the set of neighbors of the leader have access to its data and these agents might not have a directed path to all other ones.

A practical robust finite-time consensus tracking control for multi-agent systems via terminal sliding-mode surface is addressed in [90]. In this study, it is shown that for leader-follower multi-agent systems with second-order dynamic model, it is possible to achieve global finite-time consensus on terminal sliding-mode surface using switching control laws.

In [91] authors proposed nonlinear coordination control schemes for formation control of a team of under-actuated marine vehicles. In this paper, authors divided the problem into two phases as path-following of a single vehicle and coordinating the path parameter. In this paper, backstepping and Lyapunov method synthesis is used to obtain the path-following algorithm for each AUV, and then synchronize the path parameter of each AUV in order to perform the formation mission.

For the problem of coordination control in leader-follower multi-agent systems with uncertainties, the reference [92] presented a new adaptive backstepping sliding mode control method. The combination of these two methods has provided various advantages such as systematical and convenient controller design procedure, robustness to external disturbance and system uncertainties, which cannot be provided by either of them individually.

In [93], a new control algorithm based on potential function and behavior rules to effectively control the formation of a multiple AUV system under uncertain environment with the obstacle avoidance is introduced. In order to achieve the formation control tracking the target effectively while avoiding obstacle, a new distributed control algorithm is designed with the proper selection of potential functions concerning with objects, obstacles, and the structure of

the formation.

The reference [94] considered the problem of leader-follower formation control for a group of AUVs using adaptive control laws for spatial motions. The objective in this study is to make the leader to track a desired trajectory, and make the followers to keep a predefined distance with respect to the configuration of leader in three dimensional spaces. In contrast of previous studies on formation control of multiple AUVs, hydrodynamic parameter uncertainties of the AUVs is considered in the formation control law. To deal with such uncertainties, an adaptive control law based on inverse dynamics of the plant is developed.

Authors of [94] presented an adaptive distributed control for a group of AUVs in [95]. In this paper, same as [94], the hydrodynamic parameter uncertainties of the vehicles are incorporated into the formation control law. To tackle these uncertainties, an adaptive control law mainly based on inverse dynamics of the plant is developed. Moreover, a distributed controller using adaptive control technique and standard control methods in a two loop design approach is presented in order to cope with communication constraints caused by limited bandwidth in underwater environment.

Based on the works stated above, controlling a group of AUVs with uncertainties has been the topic of fairly few works. Therefore, considering these systems can be a motivating topic for the literature.

1.2.5 Fault-tolerant Control of Underwater Vehicles

With developments in control systems, dependability, reliability, and safety have become important issues to consider. If a system fails during operation, there could be catastrophic consequences. Hence, fault-tolerant control systems have become more significant and essential than before. There are several works in the literature that addressed this topic. However, to

the best of the knowledge of the author, the active fault-tolerant control issue for autonomous underwater vehicles has not been fully investigated, which remains challenging and motivating topic in the literature.

In [96], a sliding mode controller is presented for the trajectory control for a type of marine vehicles namely remotely operated vehicles (ROV). Also, a new approach for thrust allocation that is based on minimizing the largest individual component of the thrust manifold is addressed in this study. The reference [97] proposed an actuator fault-tolerant robust control scheme for underwater vehicles to solve the tracking problem for vehicle positions where a sliding mode control law is developed using the available position measurements and the velocity estimates provided by the observer.

The cooperative fault-tolerant decentralized model predictive control of a group of autonomous underwater vehicles is addressed in [98]. In this paper, each vehicle broadcasts its position, its future behavior, and its actuator/sensor fault situation to its neighbors using underwater communication channels. Then, each vehicle defines its desired formation to keep and plans its future actions depending on its local information and the information that it has received from neighbors.

In reference [99], a fault diagnosis and fault accommodation scheme for underwater vehicles are presented. In the fault diagnosis unit, improved cerebellar model articulation controller neural network is used to realize the fault identification and in the fault accommodation unit, a method of direct calculations of moment is used for the control allocation problem.

The fault identification and fault-tolerant control scheme of unmanned underwater vehicles is presented in [100] in which the identification is based on a neural network and the reconstruction algorithm is based on weighted pseudo-inverse in order to find the solution of the

control allocation problem. A novel approach to fault diagnosis and fault-tolerant control of actuators of a class of open-frame unmanned underwater vehicles is investigated in [101]. This paper focuses on multi-uncertain abrupt time-variant faults situation. The fault accommodation module uses a control algorithm based on weighted pseudo-inverse to reduce the error.

The H_∞ robust fault-tolerant controller is presented in [102] in order to improve the security and reliability of an autonomous underwater vehicle considering different actuator failures. The reference [103] presented fault-tolerant control methods for a hovering AUV with four horizontal and two vertical actuators. This work deals with the fault-tolerant control for the case when maximally three actuators are fully malfunctioned.

The reference [104] proposed an actuator fault-tolerant control scheme, consist of the usual modules performing detection, isolation, accommodation, designed for a class of nonlinear systems, and then applied to an underwater remotely operated vehicle used for inspection purposes. Detection is in charge of a residual generation module, while a sliding-mode-based approach has been used both for the ROV control and the fault isolation, after the application of an input decoupling nonlinear state transformation to the model of the ROV.

In [105], authors describe the design and implementation of a fault-tolerant control system for Omni-directional intelligent navigator, a 6-DOF autonomous underwater vehicle. In this paper, for the fault accommodation process, three methods have been considered which are anticipated fault accommodation, pseudo-inverse, and artificial intelligence methods. For the anticipated fault accommodation method, all possible faults and their possible solutions are specified in advanced, and then they are included in the control system. The pseudo-inverse approach tries to preserve the product of the input vector and the input matrix in the case of a fault on input signals. The artificial intelligence (AI) method collects data online and learns,

and its performance depends on the complexity of the intelligence program.

A actuator fault diagnosis and accommodation system for open-frame underwater vehicles is introduced in [106]. The fault accommodation unit uses information provided by the fault diagnosis module to accommodate faults and performs an appropriate control reallocation. This module uses weighted pseudo-inverse to find the solution of the control allocation problem, which minimize a control energy cost function which is used as the optimization criteria.

1.3 Statement of the Problem and Thesis Contributions

In this thesis, the problems of control and fault recovery of multiple heterogeneous autonomous underwater vehicles with uncertainties are addressed using dynamic surface control technique. The main objective is to develop cooperative control strategies so that the group of autonomous underwater vehicles with uncertainties follow the desired trajectory while agents keep a desired formation.

The cooperative control strategy proposed in this thesis is a semi-decentralized scheme in which the group of agents have the leader-follower structure where the information of the leader which is assumed to be virtual is known only to a subset of agents. The virtual leader does not receive any information from the followers. However, the followers communicate their desired relative positions with other agents in their set of neighbors in order to reach consensus. The proposed semi-decentralized scheme consists of a consensus-based algorithm combined with the dynamic surface control technique. In order to evaluate the performance of the proposed semi-decentralized scheme, three other cooperative schemes namely centralized, decentralized, and semi-decentralized control scheme with consensus algorithm using states of agents are developed as well.

In the second part of this research, the fault recovery task is accomplished to overcome the loss-of-effectiveness fault in the actuators of the agents in a group of autonomous underwater vehicles. Since the dynamics of autonomous underwater vehicles, particularly the actuators, are inevitably subjected to all types of system faults, to enhance the reliability of performance of these vehicles, the active fault-tolerant control scheme of the proposed semi-decentralized DSC-based approach is designed. To analyze the performance of the proposed fault-tolerant semi-decentralized DSC-based scheme, the DSC-based fault-tolerant centralized and decentralized schemes are developed too.

The contributions of the work developed in this thesis to solve the aforementioned problems are detailed as follows:

- A novel semi-decentralized control strategy consisting of a consensus-based algorithm and dynamic surface control technique is introduced to solve the problem of path-tracking and formation keeping of a group of heterogeneous autonomous underwater vehicles with uncertainties. In this scheme, agents communicate their desired relative distances with the agents within their set of nearest neighbors to reach consensus and it is assumed that there is a virtual leader that only a subset of agents has access to its information.
- Another novel semi-decentralized control scheme which consists of a consensus algorithm using their relative distances and velocities and dynamic surface control technique is also introduced to solve the same problem. In this scheme, agents communicate with their nearest neighbors to reach consensus and there is a virtual leader that only a subset of agents has access to its information. The comparative studies between these two semi-decentralized schemes are provided which show the superiority of the former semi-decentralized scheme over the latter.

- Inspired by [107], the active fault-tolerant control layout of the proposed semi-decentralized DSC-based scheme is developed which can recover the system from loss-of-effectiveness faults occurred in the actuators of autonomous underwater vehicles in a cooperative network.
- For the same problem, the centralized DSC-based control approach and its active fault-tolerant scheme are presented as benchmarks in order to respectively be compared to the proposed semi-decentralized approach and its fault-tolerant scheme to evaluate the performance of the proposed semi-decentralized scheme.
- The decentralized DSC-based control approach and its active fault-tolerant scheme are also illustrated to be respectively compared to the proposed semi-decentralized approach and its fault-tolerant scheme in order to show the advantages and improvements of the performance of the proposed semi-decentralized scheme.
- Three cooperative schemes, namely centralized, decentralized, and semi-decentralized schemes, based on model-dependent coordinated tracking algorithm presented in [108] for a class of mechanical systems known as Lagrangian systems are provided and their performances are compared to the first proposed semi-decentralized DSC-based scheme in the simulation studies and the improvements of the proposed semi-decentralized DSC-based scheme are highlighted.

1.4 Thesis Outline

The structure of this thesis is as follows.

- **Chapter 1** includes the literature review on topics of heterogeneous and homogeneous multi-

agent systems, single and cooperative control schemes of marine vehicles, dynamic surface control technique, and fault-tolerant control methods for the application of underwater vehicles. Also, the statement of problem and thesis contributions are given in this chapter.

- **Chapter 2** presents the background information on the topics that are used in this thesis. These information include preliminaries on nonlinear systems, dynamics and modeling of underwater vehicles containing full order and reduced order models, concepts of dynamic surface control and its complete design procedure, the individual path-tracking DSC-based control of one underwater vehicle, introduction and preliminaries on multi-agent systems and fault and fault-tolerant control systems, and finally introduction of the cooperative model-dependent coordinated tracking algorithm for Lagrangian systems.
- **Chapter 3** introduces four cooperative control strategies, i.e. the centralized, decentralized, and two novel semi-decentralized schemes, to control a group of heterogeneous autonomous underwater vehicles with uncertainties based on dynamic surface control technique. In order to investigate the performance of the presented control strategies, the simulation results of several scenarios and comparative studies are also represented.
- **Chapter 4** starts with designing the DSC-based fault-tolerant control for an autonomous underwater vehicle follows by the fault-tolerant DSC-based control schemes of all cooperative control strategies introduced in Chapter 3. To evaluate the performance of the proposed fault-tolerant semi-decentralized DSC-based control strategy, simulation results of various scenarios and comparative studies are represented as well.
- **Chapter 5** addresses concluding remarks and discusses some potential future works.

Chapter 2

Background Information

In this chapter, a review on some basic concepts related to the work conducted in this thesis is provided. The organization of this chapter is as follows. First, some necessary preliminaries and definitions on the topic of nonlinear control systems is given in Section 2.1. Second, dynamics and modeling of underwater vehicles containing full order and reduced order models are presented in Section 2.2. Next, dynamic surface control is introduced in Section 2.3 which contains the complete design procedure and the individual path-tracking DSC-based control of one underwater vehicle. Then, in Section 2.4, an introduction on multi-agent systems and some essential preliminaries on graph theory are addressed followed by some preliminaries on the topic of fault and fault-tolerant control systems are given in Section 2.5. Finally, in Section 2.6, a model-dependent coordinated tracking algorithm is represented which will be used for comparative studies in the next chapters.

2.1 Preliminaries of Nonlinear Systems

This section reviews fundamental concepts and theorems of nonlinear systems that are crucial in the developments of this thesis. These concepts are mainly extracted from [109].

Definition 1. Stability in the sense of Lyapunov. Consider the autonomous nonlinear system $\dot{x} = f(x)$ where $f : D \rightarrow \mathbb{R}^n$ and without loss of generality assume that $x^* = 0$ is the equilibrium point of the system. The equilibrium point is said to be stable if $\forall \varepsilon > 0, \exists \delta > 0$ such that

$$\|x(0)\| < \delta \implies \|x(t)\| < \varepsilon \quad , \quad \forall t \geq 0$$

Definition 2. Asymptotic stability. For the same system mentioned above, The equilibrium point is said to be asymptotically stable if it is stable and δ can be selected in such a way that

$$\|x(0)\| < \delta \implies \lim_{t \rightarrow \infty} x(t) = 0$$

Definition 3. Boundedness. The solutions of aforementioned system are uniformly bounded if there exists a positive constant γ so that for every $\alpha \in (0, \gamma)$, there exists $\beta(\alpha) > 0$ such that

$$\|x(0)\| \leq \alpha \implies \|x\| \leq \beta$$

Theorem 1. Lyapunov stability theory. For aforementioned system with $x^* = 0$ as the equilibrium point and $D \subset \mathbb{R}^n$ as a set containing $x^* = 0$, let $V : D \rightarrow \mathbb{R}$ be a continuously differentiable function so that $V(0) = 0$ and $V(x) > 0$ in $D - \{0\}$.

- If $\dot{V}(x) \leq 0$ in D , then $x^* = 0$ is stable.
- If $\dot{V}(x) < 0$ in $D - \{0\}$, then $x^* = 0$ is asymptotically stable.

2.2 Dynamics and Modeling of Underwater Vehicles

In order to design, simulate and develop control systems for underwater vehicles, a dynamic model must be identified first. Accurate modeling of underwater vehicles is essential for autonomous control. However, the modeling and control of underwater vehicles is complicated since it includes many nonlinearities and modeling uncertainties. Many hydrodynamic and inertial nonlinearities are present due to coupling between degrees of freedom. For example, currents usually exist in the underwater environments which become coupled with the direction of motion [110].

The modeling of different underwater vehicles has been investigated in many papers and studies. Although the physical characteristics and consequently the parameters of different AUVs vary in each case, the main kinematics and kinetics of almost all of them are indeed the same. For each underwater vehicle, the parameters of the chosen model are identified based on least squares (LS) and extended Kalman filter (EKF) techniques as two different steps.

This section presents the modeling process of an underwater vehicle. This is important for control purposes in order to derive a successful model-based controller. In this thesis, the model derivation is mainly based on [111].

2.2.1 Coordinate Frames

Since in this thesis, the path-tracking problem is considered, a coordinate system must be developed for the problem in order to relate the absolute position and orientation errors to local states variables that are capable of being controlled directly by the system. To derive the equations of motion for a marine vehicle it is necessary to define two coordinate frames as:

- Earth-fixed coordinate frame $\{U\}$ composed by the orthonormal axes (X_U, Y_U, Z_U)

- Body-fixed coordinate frame $\{B\}$ composed by the axes (X_B, Y_B, Z_B)

The body-fixed coordinate frame (X_B, Y_B, Z_B) is the moving coordinate frame and it is fixed to the vehicle and its axes coincide with the principal axes of inertia and as shown in Figure 2.1, they are defined as follows:

- X_B : the longitudinal axis (directed from the stern to fore);
- Y_B : the transversal axis (directed from port to starboard);
- Z_B : the normal axis (directed from top to bottom).

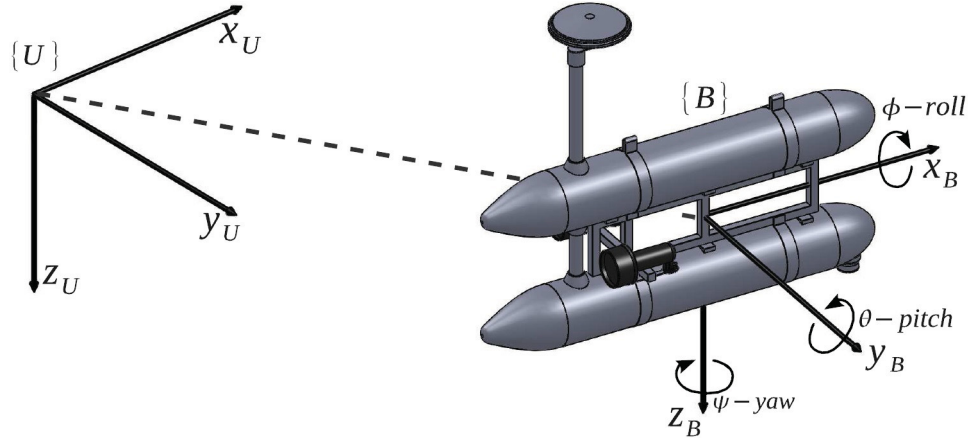


Figure 2.1: Body-fixed and inertial reference frames of an AUV [112].

To simplify the equations of the model, the origin of the body-fixed frame is normally chosen to coincide with the center of mass of the vehicle. The motion control of $\{B\}$ which corresponds to the motion of the vehicle is described relative to the inertial frame $\{U\}$.

In general, six independent coordinates are necessary to specify the evolution of the position and orientation (6-DOF), three position coordinates (x, y, z) , and using three Euler orientation angles (ϕ, θ, ψ) . These six motion components are defined as surge, sway, heave, roll, pitch, and yaw, which can be written based on the SNAME 1 notation as in Table 2.1 or in a generalized

position and velocity vector form adopted from [111] as

$$\eta = [\eta_1^T, \eta_2^T]^T \quad (2.2.1)$$

$$\nu = [\nu_1^T, \nu_2^T]^T \quad (2.2.2)$$

$$\tau = [\tau_1^T, \tau_2^T]^T \quad (2.2.3)$$

where

$\eta_1 = [x \ y \ z]^T$ is the position of the origin of $\{B\}$ expressed in $\{U\}$,

$\eta_2 = [\phi \ \theta \ \psi]^T$ is the orientation of $\{B\}$ with respect to $\{U\}$,

$\nu_1 = [u \ v \ w]^T$ is the linear velocity of the origin of $\{B\}$ relative to $\{U\}$,

$\nu_2 = [p \ q \ r]^T$ is the angular velocity of $\{B\}$ relative to $\{U\}$,

$\tau_1 = [X \ Y \ Z]^T$ is the actuating forces expressed in $\{B\}$,

$\tau_2 = [K \ M \ N]^T$ is the actuating moments expressed in $\{B\}$.

Table 2.1: Motion components used for marine vehicles

DOF	Description	Linear and Angular Velocity	Position and Euler Angles	External Forces
1	motion in the x-direction (surge)	u	x	X
2	motion in the y-direction (sway)	v	y	Y
3	motion in the z-direction (heave)	w	z	Z
4	rotation about the x-axis (roll)	p	ϕ	K
5	rotation about the y-axis (pitch)	q	θ	M
6	rotation about the z-axis (yaw)	r	ψ	N

2.2.2 Full Order Model of Underwater Vehicles

2.2.2.1 Kinematics

The kinematic equation which describes the relation between the body-fixed velocity vector and the position vector η in the north-east-down (NED) coordinate frame can be expressed as:

$$\dot{\eta} = R(\eta_2)\nu \quad (2.2.4)$$

In equation (2.2.4), $R(\eta_2)$ is the transformation matrix described by

$$R(\eta_2) = \begin{bmatrix} R_1(\eta_2) & 0 \\ 0 & R_2(\eta_2) \end{bmatrix} \quad (2.2.5)$$

where the rotation matrix $R_1(\eta_2)$ is given by

$$R_1(\eta_2) = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

and the angular velocity transformation $R_2(\eta_2)$ is defined as

$$R_2(\eta_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}$$

Note that $R_2(\eta_2)$ is singular for $\theta = \pm \left(\frac{\pi}{2}\right)$. In the aforementioned equations, $c(\cdot)$, $s(\cdot)$, and $t(\cdot)$ denote $\cos(\cdot)$, $\sin(\cdot)$, and $\tan(\cdot)$ respectively. In practice $\theta = \pm \left(\frac{\pi}{2}\right)$ would not happen do to the physics of motions of underwater vehicles. Therefore, it would not impose a problem for the control problem.

2.2.2.2 Dynamics (Kinetics)

The full order nonlinear dynamic equations of motion of a marine vehicle is expressed in the form of:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + \Delta f(\nu) = \tau \quad (2.2.6)$$

where the vectors and matrices of equation (2.2.6) are described as

- $M \in \mathbb{R}^{6 \times 6}$: inertia matrix (including added mass)
- $C(\nu) \in \mathbb{R}^{6 \times 6}$: matrix of Coriolis and centripetal terms (including added mass)
- $D(\nu) \in \mathbb{R}^{6 \times 6}$: damping matrix
- $\tau \in \mathbb{R}^6$: vector of control inputs
- $\Delta f(\nu) \in \mathbb{R}^6$: vector of unknown nonlinear uncertainties such that each of its element is bounded by a C^1 function $\rho_i(\nu)$, i.e. $|\Delta f_i(\nu)| \leq \rho_i(\nu)$ for $i = 1, \dots, 6$.

The kinetics, described by equation (2.2.6), are derived from rigid-body dynamics and hydrodynamic forces and moments. The procedure of deriving the dynamics of marine vehicles is given comprehensively by [111].

In this model, the uncertainties which are due to the lack of knowledge of the fundamental true physics of the system are taken into account. This uncertainty which is known as structural uncertainty or model inadequacy, relies on how accurately a mathematical model describes the system in a real-life situation, considering the fact that models are almost approximations to reality [113].

2.2.3 Reduced Order Model of Underwater Vehicles

In the literature, based on the application, the dynamic equations of motion of an underwater vehicle might be considered only in the horizontal plane with respect to the body fixed frame. In this model, it is assumed that the roll, pitch, and heave are close to zero and their dynamics are negligible. Thus, the motion is described by the surge, sway, and yaw dynamics and it can be obtained since independent control forces and moments are simultaneously available in all

degrees of freedom. Therefore, the reduced order model of an autonomous underwater vehicle for the horizontal motion is described as

$$\begin{aligned} M\dot{\nu} + C(\nu)\nu + D\nu + \Delta f(\nu) &= \tau \\ \dot{\eta} &= R(\psi)\nu \end{aligned} \tag{2.2.7}$$

where M , $C(\nu)$, and D belong to $\mathbb{R}^{3 \times 3}$, $\nu = [u \ v \ r]^T$ is the vector of linear and angular velocities, $\Delta f(\nu) \in \mathbb{R}^3$ is the vector of unknown nonlinear uncertainties, $\eta = [x \ y \ \psi]^T$ is the vector of positions and orientation in the inertial frame, and $\tau = [\tau_1 \ \tau_2 \ \tau_3]^T$ where τ_1, τ_2 , and τ_3 are the forces and moment that act on the surge, sway and yaw dynamics, respectively.

For 3-DOF horizontal motion, the rotation matrix is reduced to one principal rotation about the z -axis. Therefore, $R(\psi)$ is given by

$$R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2.8}$$

In this model, it has been assumed that the marine craft has xz -plane of symmetry and has homogeneous mass distribution. In other words, the center of gravity coincides with the center of added mass. It is reasonable to have such of symmetry in case that the marine vehicle has port or starboard symmetry. Coincidence of the center of gravity and the center of added mass results in simplified M and $C(\nu)$. Furthermore, it results in the fact that surge is decoupled from sway and yaw. This assumption results in final system matrices as

$$M = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{23} & m_{33} \end{bmatrix} \tag{2.2.9}$$

$$C(\nu) = \begin{bmatrix} 0 & 0 & -(m_{22}v + m_{23}r) \\ 0 & 0 & m_{11}u \\ m_{22}v + m_{23}r & -m_{11}u & 0 \end{bmatrix} \tag{2.2.10}$$

$$D = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix} \quad (2.2.11)$$

where $M = M_{RB} + M_A$, $C = C_{RB} + C_A$, and $D = D_{ln} + D_n$ with following components:

- The rigid-body mass and inertia matrix, and also the added mass matrix are found as

$$M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \quad ; \quad M_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix}$$

- The rigid-body and the added mass Coriolis and centripetal matrices are given by

$$C_{RB} = \begin{bmatrix} 0 & 0 & -m(v + x_g r) \\ 0 & 0 & mu \\ m(v + x_g r) & -mu & 0 \end{bmatrix}$$

$$C_A = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v - Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u & 0 \end{bmatrix}$$

- The linear and nonlinear damping matrices are modeled as

$$D_{ln} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix}$$

$$D_n = \begin{bmatrix} -X_{|u|u}|u| & 0 & 0 \\ 0 & -Y_{|v|v}|v| - Y_{|r|v}|r| & -Y_{|r|v} \\ 0 & -N_{|v|v}|v| - N_{|r|v}|r| & -N_{|r|v}|r| - N_{|r|r}|r| \end{bmatrix}$$

where damping in surge is coupled from sway and yaw. For more information on each element in these matrices, the reader is referred to [111].

2.2.4 Equations of Motion of Underwater Vehicles with Uncertainties

In this section, a different representation of the model of underwater vehicles is presented. As mentioned before, the equations of motion can be represented in both body-fixed and earth-fixed reference frames. The equations of motion in the body-fixed frame is the same as equation

(2.2.7). To eliminate the ν and $\dot{\nu}$ from equation (2.2.7), the equations of motion in the earth-fixed reference frame can be obtained by applying the following kinematic transformations with the assumption that $R(\psi)$ is a non-singular matrix,

$$\begin{aligned} \dot{\eta} &= R(\eta)\nu & \iff & \nu = R^{-1}(\eta)\dot{\eta} \\ \ddot{\eta} &= R(\eta)\dot{\nu} + \dot{R}(\eta)\nu & \iff & \dot{\nu} = R^{-1}(\eta) \left[\ddot{\eta} - \dot{R}(\eta)R^{-1}(\eta)\dot{\eta} \right] \end{aligned} \quad (2.2.12)$$

which yields to

$$M_\eta(\eta)\ddot{\eta} + C_\eta(\eta, \dot{\eta})\dot{\eta} + D_\eta(\eta)\dot{\eta} + \Delta f'_\eta(\eta) = \tau_\eta \quad (2.2.13)$$

where

$$\begin{aligned} M_\eta(\eta) &= R^{-T}(\eta)MR^{-1}(\eta) \\ C_\eta(\eta, \dot{\eta}) &= R^{-T}(\eta) \left[C(\eta) - MR^{-1}(\eta)\dot{R}(\eta) \right] R^{-1}(\eta) \\ D_\eta(\eta) &= R^{-T}(\eta)DR^{-1}(\eta) \\ \tau_\eta &= R^{-T}(\eta)\tau \\ \Delta f'_\eta(\eta) &= R^{-T}(\eta)\Delta f(\eta) \end{aligned} \quad (2.2.14)$$

To obtain the state space representation in controllable canonical form of underwater vehicles from equation (2.2.13), a change of variables as $z_1 = \eta$ and $z_2 = \dot{\eta}$ is used. Therefore, equation (2.2.13) can be rewritten in controllable canonical form as

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = u + \Delta f_\eta(z_1) \end{cases} \quad (2.2.15)$$

where

$$u = M_\eta^{-1} [\tau_\eta - C_\eta(z_1, z_2)z_2 - D_\eta(z_1)z_2] \quad (2.2.16)$$

and

$$\Delta f_\eta(z_1) = M_\eta^{-1}\Delta f'_\eta(z_1) \quad (2.2.17)$$

and $|\Delta f_{\eta(i)}(z_1)| < \rho_{\eta(i)}$ for $i = 1, 2, 3$.

Remark 2.1. One of the properties of the earth-fixed vector representation given in [111] is that $M_\eta = M_\eta^T > 0$. Consequently, based on the properties of the positive definite matrices, M_η^{-1} is also positive definite.

2.3 Dynamic Surface Control

2.3.1 Introduction

Due to the inability of feedback linearization method to overcome uncertainties, recently much attention has been given to Lyapunov-based control design techniques, such as backstepping approach and sliding mode control method.

Sliding mode control is a standard approach to cope with the parametric and modeling uncertainties of a nonlinear system. In this method, Lyapunov stability is applied to keep the nonlinear system under control. This approach is a method which transformed a higher-order system into an ordinary differential function. Sliding mode control requires an advanced mathematical background than other control techniques. Alternative techniques which avoid this problem include integrator backstepping and multiple sliding surface (MSS) [114].

Backstepping method designs a controller recursively by taking some state variables as virtual controls and using them as intermediate control laws during each stage of controller design. This method avoids wasteful cancellation of nonlinear terms that happens with feedback linearization. In fact, it can even add additional nonlinear terms to improve transient performance of the system. There is a class of strict-feedback form systems that connects a series of integrators to the input of a system with a known feedback-stabilizing control law. Thus, the stabilizing approach is known as integrator backstepping. However, this method leads to an explosion of terms that follow as the iteration steps increase and there is a need to differentiate the initial functions many times.

The multiple sliding surface, which is an alternative control design method, was developed independently of the integrator backstepping method, although they are mathematically very

similar to each other. This control method has the same problem as integrator backstepping in the case that it leads to explosion of terms and it needs to bound the uncertainties.

In order to avoid these drawbacks, a robust nonlinear control technique named dynamic surface control has been developed to reduce the complexity of integrator backstepping and sliding-mode controls. The DSC technique is basically composed of multiple sliding surface control and a series of first-order filters which aims to compensate the failure caused by explosion of terms. This method provides a significant performance in the presence of uncertainties.

Since in this thesis the model of autonomous underwater vehicles with uncertainties is considered as in equation (2.2.15), because of the significant ability of dynamic surface control technique to control these type of systems, this method has been selected to control the group of autonomous underwater vehicles.

2.3.2 Design Procedure for a Class of Nonlinear Systems

Based on [115], consider the following class of nonlinear systems of order n with uncertainties

$$\begin{cases} \dot{x}_i = x_{i+1} + f_i(x_1, \dots, x_i) + \Delta f_i(x_1, \dots, x_i) \\ \dot{x}_n = u + f_n(x_1, \dots, x_n) + \Delta f_n(x_1, \dots, x_n) \\ y = x_1 \end{cases} \quad (2.3.1)$$

for $1 < i < n-1$ where x_i s, u , and y are the states, input, and output of the system respectively, and the following assumptions are considered.

Assumption 2.1. $f_i : D \rightarrow \mathbb{R}$ is a smooth nonlinear function with $f_i(0) = 0$ and the column stack vector $f(x) = [f_1, \dots, f_n]^T$ and $\partial f(x)/\partial x$ are continuous on $D \in \mathbb{R}^n$ and $f(x)$ is locally Lipschitz in D .

Assumption 2.2. The uncertainty $\Delta f_i(x_1, \dots, x_i)$ is locally Lipschitz on \mathbb{R}^n .

Assumption 2.3. There exists a C^1 function $\rho_i(x_1, \dots, x_i)$ such that

$$|\Delta f_i(x_1, \dots, x_i)| \leq \rho_i(x_1, \dots, x_i) \quad ; \quad \rho_i(0, \dots, 0) = 0$$

where $\rho_i(x_1, \dots, x_i)$ is not required to be globally Lipschitz in their arguments.

Assumption 2.4. $\Delta f_i(x_1, \dots, x_i)$ is continuous on $\{x_1, \dots, x_i\}$ to guarantee the existence of solutions. In other words, each component function of unknown but locally Lipschitz nonlinear uncertainty $\Delta f(x)$ is bounded by a known class C^1 function, $\rho(x)$.

Objective. The objective is to select the control laws such that the system is stable and x_1 remains in an arbitrarily small boundary around a desired trajectory, i.e. x_{1d} where x_{1d} is the feasible output trajectory.

The standard design procedure for the dynamic surface control, which stabilizes the Lipschitz nonlinear system, is given in [115]. In this method, at each step of design, a feedback controller strengthened by nonlinear damping terms to counteract modeling errors is designed. In other words, the upper bound of uncertainty $\Delta f_i(x_1, \dots, x_i)$ is involved in the controller signal. Based on this, the first step is to define the first error surface as

$$s_1 := x_1 - x_{1d} \quad (2.3.2)$$

Differentiating S_1 with respect to time yields

$$\dot{s}_1 = x_2 + f_1(x_1) + \Delta f_1(x_1) - \dot{x}_{1d} \quad (2.3.3)$$

At this point, a design variable \bar{x}_2 is defined as

$$\bar{x}_2 = -f_1(x_1) + \dot{x}_{1d} - \lambda_1 s_1 - \frac{s_1 \rho_1^2}{2\varepsilon} \quad (2.3.4)$$

where λ_1 is the controller gain which is an arbitrary positive constants, ε is an arbitrary positive constant, and $\frac{s_1 \rho_1^2}{2\varepsilon}$ is the nonlinear damping term which makes x_1 remains in an arbitrarily small boundary around x_{1d} after some time in the presence of the locally Lipschitz uncertainty, $\Delta f_1(x_1)$. In this approach, x_2 is considered as the forcing term for the first dynamic surface in

the sense that if $x_2 = \bar{x}_2$, proper choice of λ_1 leads to $\dot{s}_1 \leq 0$. Therefore, the second step is to force $x_2 \rightarrow \bar{x}_2$ by defining the second error surface as

$$s_2 := x_2 - \bar{x}_2 \quad (2.3.5)$$

Differentiating s_2 with respect to time results in

$$\dot{s}_2 = x_3 + f_2(x_1, x_2) + \Delta f_2(x_1, x_2) - \dot{\bar{x}}_2 \quad (2.3.6)$$

Since in computing the derivative of \bar{x}_2 , the unknown term $\Delta f_1(x_1)$ shows up, it leads to the problem of "explosion of terms" which is caused by the repeated differentiations of virtual controllers that leads to a complicated algorithm with heavy computational burden. In this problem, with increasing the order of the system, the complexity of the controller will increase severely too [116].

To overcome this problem and also to be able to implement the controller in practice, a new state variable as \bar{x}_{2f} is introduced and \bar{x}_2 is passed through a first-order low-pass filter with the filter time constant γ_2 to obtain \bar{x}_{2f} as

$$\gamma_2 \dot{\bar{x}}_{2f} + \bar{x}_{2f} = \bar{x}_2 \quad (2.3.7)$$

with $\bar{x}_{2f}(0) = \bar{x}_2(0)$. Considering the transfer function of this filter

$$\frac{\bar{X}_{2f}}{\bar{X}_2} = \frac{1}{\gamma_2 s + 1}$$

it can be derived that by choosing a small filter time constant $\bar{X}_{2f} \approx \bar{X}_2$, and since $\bar{x}_{2f}(0) = \bar{x}_2(0)$ then $\bar{x}_{2f} \approx \bar{x}_2$. As shown in Figure 2.2, we have access to $\dot{\bar{x}}_{2f}$ without facing the problem of explosion of terms.

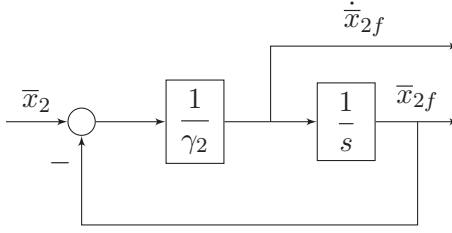


Figure 2.2: Block diagram of the first-order filter.

Having an equivalent expression for \bar{x}_2 , one is able to define an appropriate design variable \bar{x}_3 to force $x_2 \rightarrow \bar{x}_2$ as

$$\bar{x}_3 = -f_2(x_1, x_2) - \frac{s_2 \rho_2^2}{2\varepsilon} - \lambda_2 s_2 + \dot{\bar{x}}_{2f} \quad (2.3.8)$$

The same procedure should be conducted for all x_i s where $i = 3 : n - 1$. For the last state, we have

$$s_n := x_n - \bar{x}_n \quad (2.3.9)$$

and

$$\dot{s}_n = u + f_n(x_1, \dots, x_n) + \Delta f_n(x_1, \dots, x_n) - \dot{\bar{x}}_n \quad (2.3.10)$$

Using a low-pass filter to find an equivalent expression for $\dot{\bar{x}}_n$ we have

$$\gamma_n \dot{\bar{x}}_{nf} + \bar{x}_{nf} = \bar{x}_n \quad (2.3.11)$$

As the last step, the control signal is defined as:

$$u = -f_n(x_1, \dots, x_n) - \frac{s_n \rho_n^2}{2\varepsilon} - \lambda_n s_n + \dot{\bar{x}}_{nf} \quad (2.3.12)$$

which can satisfy the control objective that the system is stable and x_1 remain in an arbitrarily small boundary around the desired trajectory.

2.3.3 Dynamic Surface Control of an Autonomous Underwater Vehicle

In this section, designing a controller to solve the path-tracking problem of an individual underwater vehicle is addressed by employing DSC technique. To be able to use this technique, it is required to use earth-fixed reference model of the underwater vehicles in controllable canonical form given in equation (2.2.15) where $z_1 = \eta \in \mathbb{R}^3$ and $z_2 = \dot{\eta} \in \mathbb{R}^3$. In this problem, the objective is that the position of the underwater vehicle, i.e. z_1 , remains in an arbitrarily small boundary around the desired trajectory, i.e. $z_{1d} \in \mathbb{R}^3$.

As the first step, the first vector of error surfaces as $S_1 = [s_1 \ s_2 \ s_3]^T$ is defined as

$$S_1 := z_1 - z_{1d} \quad (2.3.13)$$

By differentiating S_1 , we have

$$\dot{S}_1 = z_2 - \dot{z}_{1d} \quad (2.3.14)$$

Choosing

$$\bar{z}_2 = \dot{z}_{1d} - \lambda_1 S_1 \quad (2.3.15)$$

with $\lambda_1 \in \mathbb{R}^{3 \times 3}$ as a positive definite diagonal gain matrix leads to $\dot{S}_1 = -\lambda_1 S_1$ which indicates that S_1 is asymptotically stable if $z_2 \rightarrow \bar{z}_2$. Therefore, the next step is to force $z_2 \rightarrow \bar{z}_2$. Thus, the second vector of error surfaces as $S_2 = [s_4 \ s_5 \ s_6]^T$ is defined as

$$S_2 := z_2 - \bar{z}_2 \quad (2.3.16)$$

The second differential equation of the system is used to obtain a control input so that $z_2 \rightarrow \bar{z}_2$.

Therefore,

$$\dot{S}_2 = u + \Delta f_\eta(z_1) - \dot{\bar{z}}_2 \quad (2.3.17)$$

where $\Delta f_\eta(z_1) = [\Delta f_{\eta(1)}(z_1) \ \Delta f_{\eta(2)}(z_1) \ \Delta f_{\eta(3)}(z_1)]^T$ and

$$\dot{\bar{z}}_2 = \ddot{z}_{1d} - \lambda_1 (z_2 - \dot{z}_{1d}) \quad (2.3.18)$$

One can choose u which is the required auxiliary controller given in equation (2.2.16) as:

$$u = \ddot{z}_{1d} - \lambda_1 (z_2 - \dot{z}_{1d}) - \lambda_2 S_2 - \frac{1}{2\varepsilon}(S_2 \circ P) \quad (2.3.19)$$

where $\frac{S_2 \circ P}{2\varepsilon}$ is the nonlinear damping term with \circ as the Hadamard product or also known as the entry-wise product of matrices and $P = \rho_\eta \circ \rho_\eta$ with $\rho_\eta = [\rho_{\eta(1)} \ \rho_{\eta(2)} \ \rho_{\eta(3)}]^T$, $\lambda_2 \in \mathbb{R}^{3 \times 3}$ is a positive definite diagonal gain matrix, and ε is a positive tuning parameter. Based on this design procedure, the closed-loop error dynamics is

$$\begin{cases} \dot{S}_1 = z_2 - \dot{z}_{1d} \\ \dot{S}_2 = u + \Delta f_\eta(z_1) - \ddot{z}_{1d} + \lambda_1 [z_2 - \dot{z}_{1d}] \end{cases} \quad (2.3.20)$$

This equation (2.3.20) can be rewritten in terms of S_1 and S_2 as

$$\begin{cases} \dot{S}_1 = -\lambda_1 S_1 + S_2 \\ \dot{S}_2 = -\lambda_2 S_2 - \frac{1}{2\varepsilon}(S_2 \circ P) + \Delta f_\eta(S_1 + z_{1d}) \end{cases} \quad (2.3.21)$$

To guarantee the stability of the closed-loop system, the Lyapunov function candidate is selected as

$$V = \frac{1}{2} (S_1^T S_1 + S_2^T S_2) \quad (2.3.22)$$

Derivative of V along the trajectories of equation (2.3.20) is given by

$$\begin{aligned} \dot{V} &= S_1^T \dot{S}_1 + S_2^T \dot{S}_2 \\ &= S_1^T (z_2 - \dot{z}_{1d}) + S_2^T (u + \Delta f_\eta(S_1 + z_{1d}) - \ddot{z}_2) \end{aligned} \quad (2.3.23)$$

Applying equations (2.3.15) and (2.3.19) leads to

$$\dot{V} = -S_1^T \lambda_1 S_1 + S_1^T S_2 - S_2^T \lambda_2 S_2 + S_2^T \Delta f_\eta(S_1 + z_{1d}) - \frac{S_2^T (S_2 \circ P)}{2\varepsilon} \quad (2.3.24)$$

Using Young's inequality which indicates

$$ab \leq \frac{a^2}{2\varepsilon} + \frac{\varepsilon b^2}{2}$$

for each row of S_2^T , by choosing $a = S_{2(i)}\rho_{\eta(i)}$ and $b = 1$ and considering the fact that $|\Delta f_{\eta(i)}(S_1 + z_{1d})| < \rho_{\eta(i)}$, it can be shown that

$$S_{2(i)}\Delta f_{\eta(i)}(S_1 + z_{1d}) \leq |S_{2(i)}|\rho_{\eta(i)} \leq \frac{S_{2(i)}^2\rho_{\eta(i)}^2}{2\varepsilon} + \frac{\varepsilon}{2} ; \quad i = 1, 2, 3$$

where $S_{2(i)}$ denotes the i^{th} entry of vector S_2^T . On the other hand, we have

$$S_2^T \Delta f(S_1 + z_{1d}) = S_{2(1)}^T \Delta f_{\eta(1)}(S_1 + z_{1d}) + S_{2(2)}^T \Delta f_{\eta(2)}(S_1 + z_{1d}) + S_{2(3)}^T \Delta f_{\eta(3)}(S_1 + z_{1d})$$

Therefore,

$$\begin{aligned} S_2^T \Delta f(S_1 + z_{1d}) &\leq \frac{S_{2(1)}^2\rho_{\eta(1)}^2 + S_{2(2)}^2\rho_{\eta(2)}^2 + S_{2(3)}^2\rho_{\eta(3)}^2}{2\varepsilon} + 3\frac{\varepsilon}{2} \\ &\leq \frac{S_2^T(S_2 \circ P)}{2\varepsilon} + 3\frac{\varepsilon}{2} \end{aligned}$$

Based on these results, it can be shown that

$$\dot{V} \leq -S_1^T \lambda_1 S_1 - S_2^T \lambda_2 S_2 + S_1^T S_2 + 3\frac{\varepsilon}{2} \quad (2.3.25)$$

Since we have quadratic forms of S_1 and S_2 , for the choices of sufficiently large λ_1 and λ_2 , it can be obtained that $-S_1^T \lambda_1 S_1 - S_2^T \lambda_2 S_2 + S_1^T S_2 \leq 0$. Also, proper choice of ε leads to a negative semi-definite \dot{V} in $\mathbb{S} = \left\{ (S_1, S_2) \mid -S_1^T \lambda_1 S_1 - S_2^T \lambda_2 S_2 + S_1^T S_2 + 3\frac{\varepsilon}{2} \leq 0 \right\}$ and the results are local. $\dot{V} \leq 0$ indicates that the closed-loop system is stable and z_1 and z_2 are bounded.

Remark 2.2. In equation (2.3.24), $S_1^T S_2$ appears since $z_2 - \bar{z}_2 \neq 0$ in \dot{S}_1 .

2.4 Cooperative Multi-agent Systems

In multi-agent systems, agents are the computational entities that operate and decide based on some tasks or goals. In spite of the fact that in many cases agents can operate individually in order to tackle a specific problem, sometimes it happens that a system consists of various agents

has to be designed to solve a more complex problem. Therefore, a multi-agent system can be described as a set of agents that are capable to communicate with each other and sometimes with their environment in order to cope with the problems which are beyond the capabilities and knowledge of an individual agent.

As various motivations for the increasing interest in studying multi-agent systems, we can mention the capability of multi-agent systems in coping with the problems in which several objectives are involved. In this case, a centralized controller is not practically implementable because of large computational cost and complexity. Furthermore, another significant motivation of developing multi-agent systems is to enhance performance in sense of computational efficiency, flexibility, reliability, extensibility, robustness, and responsiveness to name a few.

Based on the internal model of the particular individual agents forming a multi-agent system, agents may be classified as homogeneous structure or heterogeneous structure.

- **Homogeneous:** In a homogeneous architecture, all agents forming the multi-agent system have the same internal architecture. Internal architecture refers to the states, local goals, sensor capabilities, inference mechanism, and possible actions. The differences between the agents are their physical locations and the part of the environment where the action is conducted and each agent receives an input from there [117].
- **Heterogeneous:** In a heterogeneous architecture, the agents may differ in various aspects such as capability, structure, dynamic model, and functionality. Based on the characteristics of the environment and the location of a particular agent, the actions chosen by an agent might differ from another agent located in a different location but it will have the same functionality. Heterogeneous architecture helps to make modeling applications much closer to real world. Each agent can have different local goals that may contradict

the objective of other agents.

Furthermore, a team of multiple agents might have a global goal in common, or each of them might be assigned to pursue a part of a mission individually while fulfilling another common goal. In this thesis, the particular characteristic is that all agents must coordinate their actions. The importance of coordination arises due to the fact that agents have to avoid conflicts and also to improve global efficiency. When a group of dynamic agents share information or tasks to accomplish a common objective, we call the multi-agent systems a cooperative multi-agent system. In this case, the need for cooperative control rises. Two important problems in cooperative control of multi-agent systems are consensus and formation control that their applications have vastly emerged recently. The properties of these two cooperative controls can be defined as follows:

- **Consensus control:** The control objective of this problem is that a group of agents reach consensus on the values of interest. This requirement comes from the fact that in order for agents to coordinate their behaviors, they need to exploit some shared knowledge such as directions, velocities, etc.
- **Formation control:** In this problem, a network of agents aim to either reach and keep a formation, or reconfigure from one formation to another. The control of the multi-agent systems is considerably simplified when a mission is performed by means of a formation. Another benefits of formation control include increased robustness, flexibility, and success.

To solve these cooperative control problems of multi-agent systems, several approaches are introduced in the literature. Among these approaches, the most common ones are as follows:

- **Leader-follower:** The leader-following architecture consists of a leader in the group of

agents while the other agents follow that leader and receive some or all of its information such as position, orientation, velocity, etc. This approach has the advantage of simplicity in such a way that the internal stability of the formation is implied by stability of the individual vehicles. However, it greatly depends on the leader in reaching the control objectives. This dependence on a single vehicle is a disadvantage that may cause the single point of failure. Also, the lack of feedback from followers to the leader may cause instability in the entire group.

- **Behavioral methods:** The behavioral scheme prescribes a set of desired behaviors for each agent in the team. Possible behaviors include path and neighbor tracking, obstacle avoidance, formation keeping, etc. In formation control, multiple objectives should be satisfied. Therefore, from the behavioral scheme, it is expected that a control law that meets the control objectives obtains from weighting the relative importance of each behavior. This scheme motivates a cooperative implementation in which feedback to the formation is available due to the fact that an agent performs based on its neighbors. In case that the behavioral rules are given as algorithms, this scheme is mathematically difficult to analyze since the team behavior is not clear and characteristics like stability cannot generally be guaranteed.
- **Virtual structures:** In the virtual structure scheme, the entire formation is behaved as a solid virtual structure and operates as a single rigid body. The control input for a single agent is defined by deriving the dynamic model of the virtual structure and then translate its motion into a desirable motion for each agent. This scheme simplifies prescribing a coordinated behavior for the group, while formation keeping is guaranteed by the scheme. The disadvantage of this scheme is that if the formation has to keep a

unique virtual structure at all times, the potential applications are limited.

In this thesis, heterogeneous multi-agent systems are considered. The heterogeneity of the vehicles are assumed to be in their internal architecture where there are the same number of states for each vehicle but the matrices of the model of vehicles vary. Also, as will be discussed later on Chapter 3, for the semi-decentralized scheme, the leader-follower approach with a virtual leader is considered to solve the problem of formation path-tracking of a group of heterogeneous underwater vehicles using a consensus algorithm.

2.4.1 Preliminaries of Graph Theory

The communication network of a multi-agent cooperative system can mathematically be modeled by a graph. In a graph, each vehicle is considered as a node. A graph with a nonempty finite set of n nodes is usually expressed as $G = (V_G, E_G, A_G)$ where $V_G = v_1, v_2, \dots, v_n$ represents the node set, $E_G \subset V_G \times V_G$ is the set of edges, and $A_G = [a_{ij}] \in \mathbb{R}^{n \times n}$ is the associated adjacency matrix. In this thesis, the graph is assumed to be time invariant, i.e., A_G is constant.

An edge from node j to node i is denoted by (v_j, v_i) , which means that node i receives information from node j and vice versa. a_{ij} is the weight of (v_j, v_i) , and $a_{ij} > 0$ if $(v_j, v_i) \in E_G$, otherwise $a_{ij} = 0$. Node i is called a neighbor of node j if $(v_j, v_i) \in E_G$ and the set of neighbors of node j is denoted by $N_j = \{i \mid (v_j, v_i) \in E_G\}$. The in-degree matrix is defined as $D = \text{diag}\{d_i\} \in \mathbb{R}^{n \times n}$ with $d_i = \sum_{j \in N_i} a_{ij}$. Moreover, the Laplacian matrix is defined as $L = D - A_G$.

In this thesis, it is assumed that in addition to the n followers, there exists a leader, labeled as agent 0. Graph \bar{G} is the corresponding directed graph for agents 0 to n (i.e. the leader and all followers). The communications between the leader and followers are shown with a_{i0} which is a positive constant if the leader is a neighbor of agent i and $a_{i0} = 0$ otherwise.

Lemma 2.1. ([108], Lemma 1.1.) Let L be the Laplacian matrix associated with the undirected graph G of order p . Then for the undirected graph G , L has at least one zero eigenvalue and all its nonzero eigenvalues are positive. Furthermore, L has a simple zero eigenvalue and all other eigenvalues are positive if and only if G is connected.

Lemma 2.2. ([118], Lemma 2.10.) Suppose that $z = [z_1^T, \dots, z_p^T]^T$ with $z_i \in \mathbb{R}^m$. Let $A \in \mathbb{R}^{(p \times p)}$ and $L \in \mathbb{R}^{(p \times p)}$ be, respectively, the adjacency matrix and the Laplacian matrix associated with the undirected graph G . Then the following five conditions are equivalent.

1. L has a simple zero eigenvalue with an associated eigenvector $\mathbf{1}_p$ and all other eigenvalues are positive;
2. $(L \otimes I_m)z = 0$ if and only if $z_1 = \dots = z_p$;
3. Consensus is reached for the closed-loop system $\dot{z} = -(L \otimes I_m)z$ or equivalently $\dot{z} = \sum_{j=1}^p a_{ij}(z_i - z_j)$, where a_{ij} is the $(i, j)^{th}$ entry of A . That is, for all $z_i(0)$ and all $i, j = 1, \dots, p$, $\|z_i(t) - z_j(t)\| \rightarrow 0$ as $t \rightarrow \infty$;
4. The directed graph G has a directed spanning tree;
5. The rank of L is $p - 1$.

Lemma 2.3. ([108], Lemma 1.6.) Let G be a graph for p followers, labeled as agents or followers 1 to p . Let $A = [a_{ij}] \in \mathbb{R}^{p \times p}$ and $L \in \mathbb{R}^{p \times p}$ be, respectively, the adjacency matrix and the Laplacian matrix associated with G . Suppose that in addition to the p followers, there exists a leader, labeled as agent 0. Let \bar{G} be the corresponding directed graph for agents 0 to p (i.e. the leader and all followers). Defining $H = L + \text{diag}\{a_{10}, \dots, a_{p0}\}$ in which a_{i0} is a positive constant if the leader is a neighbor of agent i and $a_{i0} = 0$ otherwise. H is symmetric positive definite if and only if the leader has directed paths to all followers.

2.5 Fault-tolerant Control Systems

Similar to model uncertainties, faults and failures can change the behavior of a system. Fault in a dynamical system is a deviation of the system structure parameters such as actuators, sensors or physical structures from the nominal conditions. Every dynamical control system may be subject to faults. In underwater vehicles, actuators are one of the most common and important sources of faults [119]. Therefore, it is crucial to develop accurate fault diagnosis and recovery methods for actuators. Various types of actuator faults can be mathematically formulated as in Table 2.2.

Table 2.2: Types of actuator faults [120]

Type of fault	Description
No fault	$u_a^i(t) = u_c^i(t)$
Loss of effectiveness (LOE)	$u_a^i(t) = k_i(t)u_c^i(t) \quad 0 < \epsilon_i \leq k_i(t) < 1, \quad \forall t \geq t_{F_i}$
Float	$u_a^i(t) = 0 \quad \forall t \leq t_{F_i}$
Lock-in-place (LIP)	$u_a^i(t) = u_c^i(t_{F_i}) \quad \forall t \geq t_{F_i}$
Hard-over (HO)	$u_a^i(t) = u_{i \min} \vee u_{i \max} \quad \forall t \leq t_{F_i}$

In this table, the actuator input and output of the i^{th} actuator are represented as $u_c^i(t)$ and $u_a^i(t)$ respectively, the time that fault is injected to the i^{th} actuator is denoted by t_{F_i} , the actuator effectiveness coefficient of the i^{th} actuator is defined as $k_i(t) \in [\epsilon_i, 1]$ where $\epsilon_i > 0$ is the minimum effectiveness, and $u_{i \min}$ and $u_{i \max}$ are respectively the lower and upper limits on the actuation level of the i^{th} actuator. Also, Figure 2.3 displays the effect of these faults on an actuator output signal.

There are two types of fault-tolerant control systems (FTCS) as passive and active fault-tolerant control systems. In the case of passive fault-tolerant control systems (PFTCS), a fixed

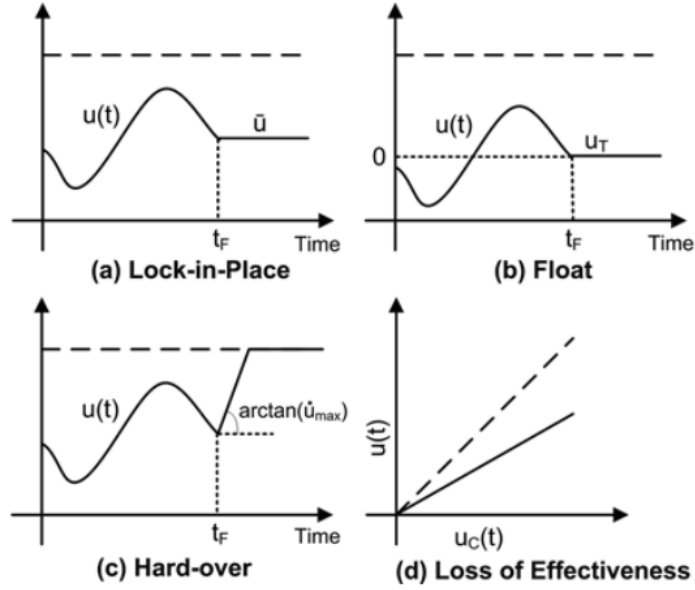


Figure 2.3: Types of actuator faults [120].

controller is proposed to tolerate only a limited predetermined faults throughout the control process. The very limited fault tolerance capability is the major drawback of this approach. On the other hand, active fault-tolerant control systems (AFTCS) rely on the fault diagnosis mechanism to detect, isolate, and identify the faults in real time, and then a reconfiguration mechanism is synthesized to reconfigure the controllers according to the online fault diagnosis information. Generally speaking, active fault-tolerant control is less conservative than the passive one and has been increasingly the main methodology in the field of FTCS design.

There are two main approaches in active fault-tolerant control systems to redesigning or recovering the controller to become fault-tolerant. These approaches are fault accommodation and control reconfiguration. Fault accommodation is to adapt the parameters of the controller to the dynamical properties of the faulty system. In this recovery approach, the input and output signals of the system used in the control loop remain the same as for the fault-free case. If fault accommodation does not perform sufficiently or is impossible, the complete control loop

has to be reconfigured. Reconfiguration consists of selection of a new control configuration where alternative input and output signals are used [121].

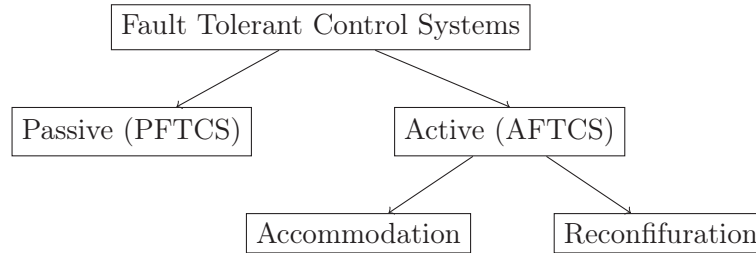


Figure 2.4: Classification of fault tolerant control systems.

Since the principal task of AFTCS is the online reconfiguration of the controller, fault detection and isolation (FDI) module plays an important role in configuration. The FDI module constantly monitors the performance of a system in order to detect faults in the system and estimates their severity and also to identify the cause, time, and location of the fault which is known as identification. However, this is unrealistic in practical systems to expect that the FDI module provides the exact information of the fault occurred in a system. Therefore, it is crucial to consider errors in detection, isolation, and identification while dealing with the AFTC systems. The errors in the FDI module consist of delay in fault detection, error in estimation of the fault severity, and error in isolation the faulty agents. In order to consider the realistic situation, in simulation section of Chapter 4 in which we deal with faulty situations, we assumed that all of these errors have occurred in the FDI module with various levels of severity.

Among studies conducted in the literature, the model-based fault-tolerant control systems for underwater vehicles have been the topic of fairly few papers. Moreover, as presented in Section 1.2.5, most of the works have considered the actuator redundancy of underwater vehicles. On the other hand, like most of the mechanical systems, loss-of-effectiveness (LOE) in

actuators of underwater vehicles is highly probable. Therefore, in order to overcome the lack of this topic in literature, in this thesis we focused on LOE faults in actuators of underwater vehicles.

2.6 Model-dependent Coordinated Tracking Algorithm for Networked Euler-Lagrange Systems

In this section, adopted from [108], the model-dependent coordinated tracking algorithm for a class of mechanical systems known as Lagrangian systems is presented. The motivation of presenting this method is to comparing it to the method proposed in this thesis to evaluate its performance.

In this method, the objective is to drive a group of agents modeled by Euler–Lagrange equations to reach desired relative formation with local interaction while they track a leader. Therefore, a coordinated tracking algorithm is presented in which there is a leader for networked Lagrangian systems under the constraints that the leader is a neighbor of only a subset of the followers and the followers have only local interactions.

Consider a group of n agents with Euler–Lagrange equations given by

$$M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) = \tau_i \quad (2.6.1)$$

for $i = 1, \dots, n$ where $q_i \in \mathbb{R}^p$ is the vector of generalized coordinates, $M_i(q_i) \in \mathbb{R}^{p \times p}$ is the symmetric positive-definite inertia matrix, $C_i(q_i, \dot{q}_i) \in \mathbb{R}^{p \times p}$ is the matrix of Coriolis and centrifugal torques, $g_i(q_i) \in \mathbb{R}^p$ is the vector of gravitational torques, and $\tau_i \in \mathbb{R}^p$ is the vector of torques produced by the actuators associated with the i^{th} agent.

The model-dependent coordinated tracking algorithm for equation (2.6.1) is proposed as

$$\tau_i = \tau_{i1} + \tau_{i2} + \tau_{i3}, \quad (2.6.2a)$$

$$\tau_{i1} = - \sum_{j=0}^n a_{ij} ((q_i - \sigma_i) - (q_j - \sigma_j)), \quad (2.6.2b)$$

$$\tau_{i2} = - \sum_{j=1}^n c_{ij} [(\dot{q}_i - \hat{v}_i) - (\dot{q}_j - \hat{v}_j)] - c_{i0} (\dot{q}_i - \hat{v}_i), \quad (2.6.2c)$$

$$\tau_{i3} = M_i(q_i)\dot{v}_i + C_i(q_i, \dot{q}_i)\hat{v}_i + g_i(q_i), \quad (2.6.2d)$$

in which

$$\dot{v}_i = - \sum_{j=1}^n b_{ij} (\hat{v}_i - \hat{v}_j) - b_{i0} (\hat{v}_i - \dot{q}_0) \quad (2.6.3)$$

where $i = 1, \dots, n$, \hat{v}_i is the i^{th} follower's estimate of the vector of generalized coordinate derivatives of the leader, σ_i is the desired formation positioning, and for $i, j = 1, \dots, n$ it is defined that:

- a_{ij} is the $(i, j)th$ entry of the adjacency matrix $A \in \mathbb{R}^{n \times n}$ associated with the graph $G_A = (V, E_A)$ characterizing the interaction among the n followers for q_i , $a_{i0} > 0$ if in \bar{G}_A the leader is a neighbor of the follower and $a_{i0} = 0$ otherwise.
- b_{ij} is the $(i, j)th$ entry of the adjacency matrix $B \in \mathbb{R}^{n \times n}$ associated with the graph $G_B = (V, E_B)$ characterizing the interaction among the n followers for \hat{v}_i , $b_{i0} > 0$ if in \bar{G}_B the leader is a neighbor of the follower and $a_{i0} = 0$ otherwise.
- c_{ij} is the $(i, j)th$ entry of the adjacency matrix $C \in \mathbb{R}^{n \times n}$ associated with the graph $G_C = (V, E_C)$ characterizing the interaction among the n followers for $\dot{q}_i - \hat{v}_i$, $c_{i0} > 0$ if in \bar{G}_C the leader is a neighbor of the follower and $c_{i0} = 0$ otherwise.

Here \bar{G}_A , \bar{G}_B , and \bar{G}_C are the directed graph characterizing the interaction among the leader and the followers corresponding to G_A , G_B , and G_C , respectively.

In this controller, equation (2.6.2b) is used to drive the vector of generalized coordinates of the i^{th} follower to track those of the followers and the leader who are its neighbors, equation (2.6.2c) is used to drive the vector of generalized coordinate derivatives of the i^{th} follower to track \hat{v}_i , equation (2.6.2d) is the compute-torque control with compensation, and equation (2.6.3) is used to estimate the vector of generalized coordinate derivatives of the leader.

Remark 2.3. In practice, having various graphs to characterize the interaction among agents provides more flexibility in choosing different sensor sets for each agent.

Remark 2.4. Using controller (2.6.2) for cooperative control of Lagrangian systems without uncertainties leads to asymptotic stability of the group. However, in case that uncertainties are added to system, the asymptotically stability proof of this controller given in [108] is not valid anymore due to the fact that this controller has a model-dependent scheme.

2.7 Conclusion

In this chapter, a review of the background materials related to the work conducted in this thesis is provided. First, fundamental concepts and theorems of nonlinear systems are presented. Then, the full order and reduced order dynamics and modeling of underwater vehicles adapted from [111] are represented. In the next part, dynamic surface control method followed by its complete design procedure is introduced. Also, the individual path-tracking control design based on dynamic surface control technique for an underwater vehicle and its stability analysis are presented. The essential of the topic of cooperative multi-agent systems and the fault-tolerant control systems are discussed afterward. Finally, in the last part of this chapter, the model-dependent coordinated tracking algorithm is presented which will be used for comparative studies.

Chapter 3

DSC-based Cooperative Schemes for Multiple Autonomous Underwater Vehicles

In this chapter, cooperative control strategies to control a group of heterogeneous AUVs based on dynamic surface control technique are proposed. First, in Section 3.1 an introduction on definition, pros, and cons of different cooperative control schemes is presented. The design specification of all cooperative schemes are presented in Section 3.2. In Sections 3.3, 3.4, 3.5, and 3.6, the DSC-based centralized, decentralized, semi-decentralized scheme with consensus algorithm using desired trajectories, and semi-decentralized scheme with consensus algorithm using actual states for a group of heterogeneous AUVs are respectively introduced. For the decentralized and both semi-decentralized schemes the stability analysis are presented as well. For the centralized scheme, since a concatenated system is controlled, the stability analysis is the same as controlling one vehicle given in Section 2.3.3. In Section 3.7, simulation results

of various scenarios and comparative studies are represented to analyze the performance of proposed control strategy.

3.1 Introduction

In this chapter, the goal is to develop cooperative control schemes in order to accomplish the objective which is the heterogeneous network of autonomous underwater vehicles with uncertainties tracks a desired path while agents keep a desired formation where the relative positioning of the vehicles should maintain in a desired distance. The novelty of the developed cooperative control schemes in this chapter is in combining DSC technique with a consensus algorithm to fulfill the cooperative objectives while making advantage from the capability of DSC technique to control nonlinear systems with uncertainties.

Generally, the cooperative control of multi-agent systems can be accomplished by three different distribution control schemes as shown in Figure 3.1, namely, centralized, decentralized, and semi-decentralized cooperative schemes.

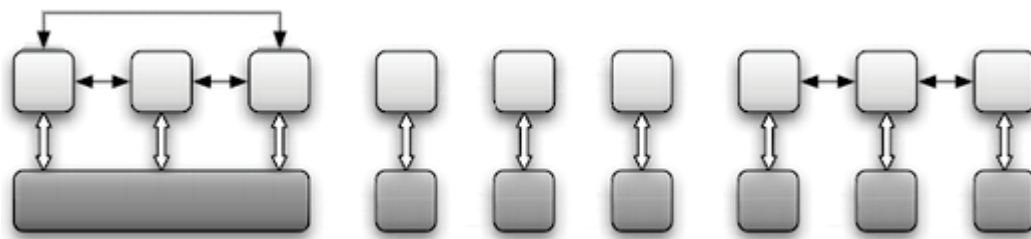


Figure 3.1: From left to right, centralized, decentralized, and semi-decentralized cooperative schemes, where the white and dark boxes represent agents and controllers, respectively [122].

In centralized scheme, the common reference trajectory is implemented at a central location and broadcasts the control input to every agent in the group. For this scheme, the simple

conceptual framework is its main advantage. In this scheme, there is one system and all agents use the same controller which is obtained based on the information of all agents in the group. Since the central system uses the global information of all agents, it provides the optimal solution for the cooperative problem. In this scheme, all agents are considered to be neighbors who send and receive all data in the group. Therefore, one of the cons of this scheme is that because of having a large scale system with many inputs, states, and outputs, the problem can be costly if the system model is huge and many agents are involved. This implementation scheme results in a single point of failure and is not scalable well to a large number of agents. The centralized control scheme might be also unappealing for economical and implementation reasons.

Decentralized control scheme is a natural remedy to drawbacks of centralized scheme where a local copy of the reference command is available for each agent. In case of leader-follower approach, the decentralized scheme refers to the structure in which all followers receive data only from the command center. If each vehicle implements the same cooperation algorithm, it is expected that the decentralized scheme achieves the same cooperation as the centralized scheme. In decentralized control scheme, local computation of control variables and also simplicity of the design are the pros of this scheme. However, absence of communication between agents limits the achievable performance. Moreover, if one of the agents loses its communications with the command center, there is no chance to refine it and bring it back to the group.

Finally, for semi-decentralized control scheme, there are local communications between agents and one can tune the trade-off between communication burden and performance. This scheme is a middle ground between centralized and decentralized schemes. It has almost the efficiency of centralized scheme, while there is no need to solve a huge problem which is imprac-

tical in reality. Still, there are some costs to gain such a performance such as complex design methods and slightly deviated transient response. The difference between this scheme and the centralized scheme is that the centralized controller is divided into several simpler sub-systems performing based on local information which causes lower computational and communication requirements. Furthermore, in this scheme, there are challenges due to network non-idealities such as delays, packet drops, etc.

3.2 Design Specification

In this section, the design specifications of all cooperative schemes considered in this thesis are determined in order to highlight the differences among these schemes.

In order to be able to benefit from dynamic surface control method to solve the cooperative control of a group of heterogeneous autonomous underwater vehicles with uncertainties, the leader-follower structure is selected. The strength of the leader-follower structure is that group behavior is directed by specifying the behavior of a single quantity which is the leader. The weakness of this structure is that having one of the agents as the leader has the problem of a single point of failure; if the leader fails, the entire group will fail. To solve this problem, instead of assigning one of the vehicles as the leader, a virtual leader as a moving reference point can be used. This feature adds robustness to the failure of an individual vehicle. Also, in the underwater environment that the communication lost are very probable, it is more practical to have a virtual leader instead of assigning one of the agents as the leader, since if the real leader loses the efficiency of its communication systems then the entire group will fail. Thus, in all of these schemes, the position states of the virtual leader, also known as the command reference, which is labeled as agent "0" is denoted as $z_{1d}^{(0)} = [x^{(0)} \ y^{(0)} \ \psi^{(0)}]^T$. The communications

specifications and the design specifications of the desired trajectory in each cooperative scheme presented in this thesis are as follows:

- **Centralized scheme:** In this scheme, it is assumed that there is a central command location which defines the control input of the entire group using the position states of all agents and broadcasts it to every agent in the network. Also, it is assumed that there is a virtual leader which sends its positioning data to some of the agents of the group. However, it does not receive any data from any follower. In this scheme, to obtain the desired trajectory of each agent, a consensus-formation tracking algorithm introduced in [123] is used as

$$\dot{z}_{1d} \triangleq \begin{bmatrix} -\xi_1 \left(\sum_{j \in N_i} (z_{1d}^{1j} - \sigma^{1j}) + a_{i0}(z_{1d}^{10} - \sigma^{10}) \right) - \xi_2 \operatorname{sgn} \left(\sum_{j \in N_i} (z_{1d}^{1j} - \sigma^{1j}) + a_{i0}(z_{1d}^{10} - \sigma^{10}) \right) \\ \vdots \\ -\xi_1 \left(\sum_{j \in N_i} (z_{1d}^{nj} - \sigma^{nj}) + a_{n0}(z_{1d}^{n0} - \sigma^{n0}) \right) - \xi_2 \operatorname{sgn} \left(\sum_{j \in N_i} (z_{1d}^{nj} - \sigma^{nj}) + a_{n0}(z_{1d}^{n0} - \sigma^{n0}) \right) \end{bmatrix} \quad (3.2.1)$$

in which $z_{1d} = [z_{1d}^{(1)} \dots z_{1d}^{(n)}]^T \in \mathbb{R}^{(m \times n) \times 1}$ is the column stack vector of desired trajectories, ξ_1 and ξ_2 are positive constants, z_{1d}^{ij} is the relative desired position of i^{th} vehicle with respect to j^{th} vehicle, σ^{ij} is the desired formation positioning between i^{th} and j^{th} vehicles, $a_{i0} > 0$ if agent i receives data from the virtual leader otherwise $a_{i0} = 0$, z_{1d}^{i0} is the relative desired position of i^{th} vehicle with respect to the virtual leader, σ^{i0} is the desired formation positioning between i^{th} and the virtual leader, and $\|\cdot\|$ denotes the Euclidean norm of a vector.

Since the aim of using this consensus algorithm is to produce the desired trajectories of each agent, we need an algorithm with a finite-time consensus instead of just asymptotic consensus. In the aforementioned consensus-formation tracking algorithm, since there

is a moving reference with a time varying position and velocity, the signum function is required to reach a finite-time consensus as proved in Theorem 3.1 of [123]. However, since signum function is a discontinuous function, it is not differentiable and it may cause chattering problem in actuators. To avoid this problem, an estimation of this function will be used as follows

$$\dot{z}_{1d} = \begin{bmatrix} -\xi_1 \left(\sum_{j \in N_i} (z_{1d}^{1j} - \sigma^{1j}) + a_{10}(z_{1d}^{10} - \sigma^{10}) \right) - \xi_2 \frac{\sum_{j \in N_i} (z_{1d}^{1j} - \sigma^{1j}) + a_{10}(z_{1d}^{10} - \sigma^{10})}{\left\| \sum_{j \in N_i} (z_{1d}^{1j} - \sigma^{1j}) + a_{10}(z_{1d}^{10} - \sigma^{10}) \right\| + \epsilon} \\ \vdots \\ -\xi_1 \left(\sum_{j \in N_i} (z_{1d}^{nj} - \sigma^{nj}) + a_{n0}(z_{1d}^{n0} - \sigma^{n0}) \right) - \xi_2 \frac{\sum_{j \in N_i} (z_{1d}^{nj} - \sigma^{nj}) + a_{n0}(z_{1d}^{n0} - \sigma^{n0})}{\left\| \sum_{j \in N_i} (z_{1d}^{nj} - \sigma^{nj}) + a_{n0}(z_{1d}^{n0} - \sigma^{n0}) \right\| + \epsilon} \end{bmatrix} \quad (3.2.2)$$

where ϵ is a positive constant which is added to prevent having zero in denominator.

Theorem 3.1. ([123], Theorem 5.1.) Assume that the fixed undirected graph \bar{G} is connected and the virtual leader has directed paths to all vehicles 1 to n at each time instant. Also, assume that the leader has a time-vary position $z^{(0)}$ and velocity $\dot{z}^{(0)}$ for which it is assumed that $|\dot{z}^{(0)}| < \Gamma_l$. With using controller

$$u^{(i)} = -\xi_1 \left(\sum_{j \in N_i} (z^{ij} - \sigma^{ij}) + a_{i0}(z^{i0} - \sigma^{i0}) \right) - \xi_2 \frac{\sum_{j \in N_i} (z^{ij} - \sigma^{ij}) + a_{i0}(z^{i0} - \sigma^{i0})}{\left\| \sum_{j \in N_i} (z^{ij} - \sigma^{ij}) + a_{i0}(z^{i0} - \sigma^{i0}) \right\| + \epsilon}$$

for a system with single-integrator dynamics as

$$\dot{z}^{(i)} = u^{(i)}$$

if $\xi_2 > \Gamma_l$, then $|z^{(i)}(t) - z^{(j)}(t)| \rightarrow \sigma^{ij}$ in finite time for $i, j = 0, \dots, n$.

- **Decentralized scheme:** In this scheme, it is assumed that all of the agents directly receive the positioning data of the virtual leader and each agent rules its own controller only based on its position information and the positioning data of the command reference.

Therefore, the desired positioning for each vehicle obtains individually using a predefined positioning based on a desired distance and orientation from the command reference. One can generate the desired positioning of each agent, such that it is located by the distance $\beta^{(i)}$ and the orientation $\alpha^{(i)}$ with respect to the common reference trajectory as

$$z_{1d}^{(i)} \triangleq \begin{bmatrix} x_d^{(i)} \\ y_d^{(i)} \\ \psi_d^{(i)} \end{bmatrix} = \begin{bmatrix} x^{(0)} + \beta^{(i)}(\cos(\psi^{(0)} + \alpha^{(i)})) \\ y^{(0)} + \beta^{(i)}(\sin(\psi^{(0)} + \alpha^{(i)})) \\ \psi^{(0)} \end{bmatrix} \quad (3.2.3)$$

- **Semi-decentralized scheme:**

1. **With consensus algorithm using states of agents:** In this semi-decentralized scheme, there is a virtual leader whose position information is only known to a subset of agents and it does not receive any information from followers. In this scheme, agents communicate their relative position and velocity states with agents within their set of neighbors and each agent determines its own control input. Therefore, the desired trajectories of each vehicle is a function of positions of agents and are obtained based on a consensus-based formation tracking algorithm. Considering Lemma 2.2, the dynamics of the desired trajectories is given by

$$\dot{z}_{1d}^{(i)} \triangleq \dot{z}_1^{(i)} = -\alpha_1 \left(\sum_{j \in N_i} (z_1^{ij} - \sigma^{ij}) + a_{i0}(z_1^{i0} - \sigma^{i0}) \right) \quad (3.2.4)$$

in which z_1^{ij} is the relative position of i^{th} vehicle with respect to the j^{th} vehicle, σ^{ij} is the desired formation positioning between i^{th} and j^{th} vehicles, $a_{i0} > 0$ if agent i receives data from the virtual leader otherwise $a_{i0} = 0$, z_1^{i0} is the relative position of i^{th} vehicle with respect to the virtual leader, σ^{i0} is the desired formation positioning between i^{th} and the virtual leader. The derivative of (3.2.4) is given by

$$\ddot{z}_{1d}^{(i)} \triangleq \ddot{z}_2^{(i)} = -\alpha_2 \left(a_{i0} \dot{z}_2^{i0} + \sum_{j \in N_i} \dot{z}_2^{ij} \right) \quad (3.2.5)$$

which is required due to the structure of dynamic surface control technique as will be explained in Section 3.5. In equation (3.2.5), $z_2^{ij} = z_2^i - z_2^j$ is the relative velocity of the i^{th} vehicle with respect to the j^{th} vehicle, $a_{i0} > 0$ if agent i receives data from the virtual leader otherwise $a_{i0} = 0$, and $z_2^{i0} = z_2^i - z_2^0$ is the relative velocity of i^{th} vehicle with respect to the virtual leader.

2. **With consensus algorithm using desired trajectories:** In this type of semi-decentralized scheme, instead of the position and velocity states, agents communicate their desired relative positions with agents within their set of neighbors and each agent determines its own control input. To define the desired trajectories of each vehicle, a consensus-based formation tracking algorithm based on the leader-follower approach is used. In this scheme, the desired trajectory of each agent is obtained using a consensus algorithm as

$$\dot{z}_{1d}^{(i)} = -\xi_1 \left(\sum_{j \in N_i} (z_{1d}^{ij} - \sigma^{ij}) + a_{i0}(z_{1d}^{i0} - \sigma^{i0}) \right) - \xi_2 \frac{\sum_{j \in N_i} (z_{1d}^{ij} - \sigma^{ij}) + a_{i0}(z_{1d}^{i0} - \sigma^{i0})}{\left\| \sum_{j \in N_i} (z_{1d}^{ij} - \sigma^{ij}) + a_{i0}(z_{1d}^{i0} - \sigma^{i0}) \right\| + \epsilon} \quad (3.2.6)$$

where ξ_1 , ξ_2 , and ϵ are positive constants, and for all $i, j = 0, \dots, n$ it is assumed that z_{1d}^{ij} is the relative desired position of i^{th} vehicle with respect to j^{th} vehicle, σ^{ij} is the desired formation positioning between i^{th} and j^{th} vehicles, N_i is the set of neighbors of i^{th} agent, $a_{i0} > 0$ if agent i receives data from the virtual leader otherwise $a_{i0} = 0$, and $\|\cdot\|$ denotes the Euclidean norm of a vector.

Remark 3.1. The advantage of using the second semi-decentralized scheme is that, first, the size of the transmitted data is halved since in the former semi-decentralized scheme $z_1^{ij} \in \mathbb{R}^3$ and $z_2^{ij} \in \mathbb{R}^3$ were transmitted between agents but in the latter scheme only $z_{1d}^{ij} \in \mathbb{R}^3$ is transmitted. In addition, $z_{1d}^{(i)}$ s are the data calculated by the agents,

and agents do not need to obtain data from displacement and velocity sensors which are subject to errors and faults.

3.3 DSC-based Centralized Control Scheme

In this section, the centralized control scheme based on DSC technique for formation-tracking of multiple heterogeneous underwater vehicles is presented. The objective for each vehicle is to maintain a desired formation with respect to other vehicles while all of them track a desired predefined path.

In this scheme, there is a central controller that has global information of the entire team and determines the local inputs of each agent and communicates these inputs to agents by imposing stringent communication requirements. Since the central controller has access to all information at the same time by using a concatenated model of the system, it has an optimal solution. Therefore, this control scheme is a benchmark case that provides a point of reference against which the other cooperative control schemes will be compared. However, in this scheme, because of solving a global problem of large size, the size of computations is high. Based on equations (2.2.16) and (2.3.19), the centralized controller based on DSC technique is given by

$$\tau = M_{con}(\underline{\eta}) \left(\dot{\underline{\bar{z}}}_2 - \lambda_2 \underline{S}_2 - \frac{1}{2\varepsilon} (\underline{S}_2 \circ \underline{P}) \right) + C_{con}(\underline{\eta}, \dot{\underline{\eta}}) \dot{\underline{\eta}} + D_{con}(\underline{\eta}) \dot{\underline{\eta}} \quad (3.3.1)$$

with

$$\underline{\bar{z}}_2 = -\lambda_1 \underline{S}_1 + \dot{\underline{z}}_{1d} \quad (3.3.2)$$

in which $\dot{\underline{z}}_{1d}$ is given by equation (3.2.2), and differentiating $\underline{\bar{z}}_2$ yields

$$\dot{\underline{\bar{z}}}_2 = -\lambda_1 (\underline{z}_2 - \dot{\underline{z}}_{1d}) + \ddot{\underline{z}}_{1d} \quad (3.3.3)$$

In this scheme, the concatenated model of the dynamic equations of all agents is used. Considering n as the number of vehicles, m as the number of states of each vehicles, and \otimes as the Kronecker product, the vectors and matrices of equation (3.3.1) are defined as follows:

- $M_{con} = I_{n \times n} \otimes M \in \mathbb{R}^{(m \times n) \times (m \times n)}$ with $M = [M_\eta^{(1)} \dots M_\eta^{(n)}]^T$;
- $C_{con} = I_{n \times n} \otimes C \in \mathbb{R}^{(m \times n) \times (m \times n)}$ with $C = [C_\eta^{(1)} \dots C_\eta^{(n)}]^T$;
- $D_{con} = I_{n \times n} \otimes D \in \mathbb{R}^{(m \times n) \times (m \times n)}$ with $D = [D_\eta^{(1)} \dots D_\eta^{(n)}]^T$;
- $\underline{\lambda}_1 = \text{diag}(\lambda_1^{(1)}, \dots, \lambda_1^{(n)}) \in \mathbb{R}^{(m \times n) \times (m \times n)}$ and $\underline{\lambda}_2 = \text{diag}(\lambda_2^{(1)}, \dots, \lambda_2^{(n)}) \in \mathbb{R}^{(m \times n) \times (m \times n)}$ are the concatenated matrices of the controller gains;
- $\underline{\eta} = [\eta^{(1)} \dots \eta^{(n)}]^T \in \mathbb{R}^{(m \times n) \times 1}$ is the column stack vectors of positions of all agents;
- $\underline{\tau} = [\tau^{(1)} \dots \tau^{(n)}]^T \in \mathbb{R}^{(m \times n) \times 1}$ is the column stack vectors of control input of all agents;
- $\underline{P} = \underline{\rho}_\eta \circ \underline{\rho}_\eta$ with $\underline{\rho}_\eta = [\rho_\eta^{(1)} \dots \rho_\eta^{(n)}]^T \in \mathbb{R}^{(m \times n) \times 1}$ is the column stack vector of element-wised squared upper bounds of $\Delta f_\eta^{(i)}(z_1)$ of all agents;
- $\underline{S}_1 = [(z_1^{(1)} - z_{1d}^{(1)}) \dots (z_1^{(n)} - z_{1d}^{(n)})]^T \in \mathbb{R}^{(m \times n) \times 1}$ is the column stack vector of the first surface errors;
- $\underline{S}_2 = [(z_2^{(1)} - \bar{z}_2^{(1)}) \dots (z_2^{(n)} - \bar{z}_2^{(n)})]^T \in \mathbb{R}^{(m \times n) \times 1}$ is the column stack vector of the second surface errors;
- $\underline{\bar{z}}_2 = [\bar{z}_2^{(1)} \dots \bar{z}_2^{(n)}]^T \in \mathbb{R}^{(m \times n) \times 1}$ is the column stack vector of design variables.

Remark 3.2. In this centralized scheme, the interactions between agents comes through the controller which appears in \dot{z}_{1d} of equation (3.2.2).

Remark 3.3. Since in this scheme we control a single concatenated system, the stability analysis is the same as controlling one vehicle given in Section 2.3.3.

3.4 DSC-based Decentralized Control Scheme

This scheme is a distribution scheme where the agents do not share any information with each other and each agent has access to the information of the reference trajectory and decide only based on its own information and the desired reference. In this case, the objective for each individual agent is to follow the desired trajectory while keeping a desired distance from this trajectory in order to fulfill the formation objective. Since there is no communications in this scheme, this scheme cannot provide an optimum performance. In this scheme, the model of each agent is represented as

$$M_\eta^{(i)}\ddot{\eta}^{(i)} + C_\eta^{(i)}(\eta^{(i)}, \dot{\eta}^{(i)})\dot{\eta}^{(i)} + D_\eta^{(i)}(\eta^{(i)})\dot{\eta}^{(i)} + \Delta f_\eta^{(i)}(\eta^{(i)}) = \tau_\eta^{(i)}$$

with $i = 1, \dots, n$ where n is the number of agents. As explained in Section 2.2.4, this equation can be rewritten in controllable canonical form as

$$\begin{cases} \dot{z}_1^{(i)} = z_2^{(i)} \\ \dot{z}_2^{(i)} = u^{(i)} + \Delta f_\eta^{(i)}(z_1^{(i)}) \end{cases} \quad (3.4.1)$$

with the auxiliary control law as

$$u^{(i)} = M_\eta^{-1(i)} \left[\tau_\eta^{(i)} - C_\eta^{(i)}(z_1^{(i)}, z_2^{(i)})z_2^{(i)} - D_\eta^{(i)}(z_1^{(i)})z_2^{(i)} \right] \quad (3.4.2)$$

Using DSC method, the controller of each vehicle is defined by itself as

$$\tau_\eta^{(i)} = M_\eta^{(i)} \left(\frac{\dot{z}_2^{(i)}}{\bar{z}_2^{(i)}} - \lambda_2^{(i)} S_2^{(i)} - \frac{S_2^{(i)} \circ P^{(i)}}{2\varepsilon} \right) + C_\eta^{(i)}(z_1^{(i)}, z_2^{(i)})z_2^{(i)} + D_\eta^{(i)}(z_1^{(i)})z_2^{(i)} \quad (3.4.3)$$

with $S_2^{(i)} = z_2^{(i)} - \bar{z}_2^{(i)}$, $P^{(i)} = \rho_\eta^{(i)} \circ \rho_\eta^{(i)}$ where $\rho_\eta^{(i)} = [\rho_{\eta(1)}^{(i)} \ \rho_{\eta(2)}^{(i)} \ \rho_{\eta(3)}^{(i)}]^T$, and

$$\bar{z}_2^{(i)} = \dot{z}_{1d}^{(i)} - \lambda_1^{(i)} (z_1^{(i)} - z_{1d}^{(i)}) \quad (3.4.4)$$

and

$$\frac{\dot{z}_2^{(i)}}{\bar{z}_2^{(i)}} = -\lambda_1^{(i)} (z_2^{(i)} - \dot{z}_{1d}^{(i)}) + \ddot{z}_{1d}^{(i)} \quad (3.4.5)$$

in which the trajectory $z_{1d}^{(i)}$ is obtained from equation (3.2.3).

3.4.1 Stability Analysis

In order to theoretically show the stability of each agent in a decentralized scheme, consider $S_1^{(i)} = z_1^{(i)} - z_{1d}^{(i)}$ and $S_2^{(i)} = z_2^{(i)} - \bar{z}_2^{(i)}$ as the first and second surface errors of i^{th} agent respectively. The closed-loop error dynamics is

$$\begin{cases} \dot{S}_1^{(i)} = z_2^{(i)} - \dot{z}_{1d}^{(i)} \\ \dot{S}_2^{(i)} = u^{(i)} + \Delta f_\eta^{(i)}(S_1^{(i)} + z_{1d}^{(i)}) - \left(\ddot{z}_{1d}^{(i)} - \lambda_1^{(i)} (z_2^{(i)} - \dot{z}_{1d}^{(i)}) \right) \end{cases} \quad (3.4.6)$$

for $1 \leq i \leq n$ where n is the number of agents. To guarantee the stability of the closed-loop system, the Lyapunov function candidate is selected as

$$V^{(i)} = \frac{1}{2} \left(S_1^{T(i)} S_1^{(i)} + S_2^{T(i)} S_2^{(i)} \right) \quad (3.4.7)$$

Differentiating V along the trajectories of equation (3.4.6) is given by

$$\dot{V}^{(i)} = S_1^{T(i)} \left(z_2^{(i)} - \dot{z}_{1d}^{(i)} \right) + S_2^{T(i)} \left(u^{(i)} + \Delta f_\eta^{(i)}(z_1^{(i)}) - \dot{z}_2^{(i)} \right) \quad (3.4.8)$$

Applying equation (3.4.4) and $u^{(i)} = \dot{z}_2^{(i)} - \lambda_2^{(i)} S_2^{(i)} - \frac{S_2^{(i)} \circ P^{(i)}}{2\varepsilon}$ as the auxiliary part of the equation (3.4.3) leads to

$$\dot{V}^{(i)} = -S_1^{T(i)} \lambda_1^{(i)} S_1^{(i)} + S_1^{T(i)} S_2^{(i)} - S_2^{T(i)} \lambda_2^{(i)} S_2^{(i)} + S_2^{T(i)} \Delta f_\eta^{(i)}(z_1^{(i)}) - \frac{S_2^{T(i)} (S_2^{(i)} \circ P^{(i)})}{2\varepsilon} \quad (3.4.9)$$

Same as Section 2.3.3, using Young's inequality, it can be shown that

$$\dot{V}^{(i)} \leq -S_1^{T(i)} \lambda_1^{(i)} S_1^{(i)} - S_2^{T(i)} \lambda_2^{(i)} S_2^{(i)} + S_1^{T(i)} S_2^{(i)} + 3\frac{\varepsilon}{2}$$

Proper choice of $\lambda_1^{(i)}$, $\lambda_2^{(i)}$, and ε leads to $\dot{V}^{(i)} \leq 0$ in $\mathbb{S} = \{(S_1, S_2) | -S_1^{T(i)} \lambda_1^{(i)} S_1^{(i)} - S_2^{T(i)} \lambda_2^{(i)} S_2^{(i)} + S_1^{T(i)} S_2^{(i)} + 3\frac{\varepsilon}{2} \leq 0\}$ which indicates that the results are local. Based in these results, each agent is stable and $z_1^{(i)}$ and $z_2^{(i)}$ are bounded.

In the next step of stability proof, the stability of the entire group should be taken into account. Since in the decentralized scheme there is no communication between agents, to show

the stability of the group, it is required to show that the relative distance between each two arbitrary agents remains bounded. To this end, the relative distance between two arbitrary agents is defined as

$$\begin{aligned}
\|z^{ij}\| &= \|z_1^{(i)} - z_1^{(j)}\| \\
&= \|z_1^{(i)} - z_{1d}^{(i)} - z_1^{(j)} + z_{1d}^{(i)}\| \\
&= \|z_1^{(i)} - z_{1d}^{(i)} - z_1^{(j)} + z_{1d}^{(j)} + r^{ij}\| \\
&\leq \|z_1^{(i)} - z_{1d}^{(i)}\| + \|z_1^{(j)} - z_{1d}^{(j)}\| + \|r^{ij}\|
\end{aligned}$$

Since it has been shown that each agent is individually stable and its trajectories are bounded, i.e. $\|z_1^{(k)} - z_{1d}^{(k)}\| \leq \epsilon$ where $\epsilon > 0$ for all $k = 1, \dots, n$, and also $\|r^{ij}\|$ is a positive constant value defined in the desired formation design, it can be concluded that

$$\begin{aligned}
\|z^{ij}\| &\leq 2\epsilon + \|r^{ij}\| \\
\|z^{ij}\| &\leq \bar{\epsilon}
\end{aligned} \tag{3.4.10}$$

where $\bar{\epsilon} > 0$. This indicates that the relative distance between each two arbitrary agents will remain bounded.

3.5 DSC-based Semi-Decentralized Control Scheme with Consensus Algorithm Using States of Agents

To overcome the drawback of the centralized scheme, which is being costly for economical and implementation reasons, and the drawback of the decentralized scheme, which is the inability of group to refine the agent which loses its communications with the command center, a novel semi-decentralized control scheme based on DSC technique combined with a consensus-based cooperative algorithm on the position and velocity states is proposed for the problem

of formation path-tracking of a group of heterogeneous autonomous underwater vehicles. In this formation-consensus problem, the objective is to maintain a proper relative positioning among the vehicles while some of them have access to information of the virtual leader. The assumptions in this section are as follows.

Assumption 3.1. Graph \overline{G} is a fixed graph and it is connected which results in a positive semi-definite Lagrangian matrix associated with it.

Assumption 3.2. At least one agent in the group is connected to the virtual leader.

Assumption 3.3. The leader has a time-vary position $z_{1d}^{(0)}$ and velocity $\dot{z}_{1d}^{(0)}$ for which it is assumed that $|\dot{z}_{1d}^{(0)}| < \Gamma_l$.

For the group of underwater vehicles with the model represented in equations (3.4.1), based on the design specifications presented in Section 3.2, the controller of each agent is designed as

$$\tau_\eta^{(i)} = M_\eta^{(i)} u^{(i)} + C_\eta^{(i)}(z_1^{(i)}, z_2^{(i)}) z_2^{(i)} + D_\eta^{(i)}(z_1^{(i)}) z_2^{(i)} \quad (3.5.1)$$

where

$$u^{(i)} = -\lambda_2^{(i)} S_2^{(i)} - \frac{S_2^{(i)} \circ P^{(i)}}{2\varepsilon} - \alpha_2 \left(\sum_{j \in N_i} (z_2^{ij}) + a_{i0}(z_2^{i0}) \right) - \lambda_1^{(i)} \left(z_2^{(i)} + \alpha_1 \left(\sum_{j \in N_i} (z_1^{ij} - \sigma^{ij}) + a_{i0}(z_1^{i0} - \sigma^{i0}) \right) \right) \quad (3.5.2)$$

with $P^{(i)} = \rho_\eta^{(i)} \circ \rho_\eta^{(i)}$ where $\rho_\eta^{(i)} = [\rho_{\eta(1)}^{(i)} \ \rho_{\eta(2)}^{(i)} \ \rho_{\eta(3)}^{(i)}]^T$ and

$$S_2^{(i)} = z_2^{(i)} - \left(-\alpha_1 \left(\sum_{j \in N_i} (z_1^{ij} - \sigma^{ij}) + a_{i0}(z_1^{i0} - \sigma^{i0}) \right) - \lambda_1^{(i)} \left(z_1^{(i)} - z_{1d}^{(i)} \right) \right) \quad (3.5.3)$$

3.5.1 Stability Analysis

Theorem 3.2. Suppose that Assumptions 3.1, 3.2, and 3.3 are valid. By using controller given by equation (3.5.1) for a group of agents with dynamics given by equations (2.2.15) and

(2.2.16), with proper choices of $\lambda_1^{(i)}$, $\lambda_2^{(i)}$, and ε , the team of agents is stable and all $z_1^{(i)}$ and $z_2^{(i)}$ are bounded.

Proof. Without loss of generality, assume that all $\lambda_1^{(i)}$ s are equal, and the same for $\lambda_2^{(i)}$ s. Considering $\underline{S}_1 \triangleq [S_1^{T(1)}, \dots, S_1^{T(n)}]^T$ and $\underline{S}_2 \triangleq [S_2^{T(1)}, \dots, S_2^{T(n)}]^T$ as the column stack vectors of the first and second error surfaces with $S_1^{(i)}, S_2^{(i)} \in \mathbb{R}^3$, we can write the closed-loop error dynamics of the group of agents as

$$\begin{cases} \dot{\underline{z}}_1 = \underline{z}_2 + \alpha_1 (M \otimes I_3) \underline{z}_1 \\ \dot{\underline{z}}_2 = \underline{u} + \underline{\Delta f}_\eta(\underline{S}_1 + \underline{z}_{1d}) + \underbrace{\alpha_2 (M \otimes I_3) \underline{z}_2 + (\lambda_1 \otimes I_n) (\underline{z}_2 + \alpha_1 (M \otimes I_3) \underline{z}_1)}_{\dot{\underline{z}}_2} \\ \dot{\underline{z}}_{1d} = -\alpha_1 (M \otimes I_3) \underline{z}_1 \end{cases} \quad (3.5.4)$$

where I_3 is the 3×3 identity matrix, $\underline{z}_1 \triangleq [z_1^{T(1)}, \dots, z_1^{T(n)}]^T$, $\underline{z}_2 \triangleq [z_2^{T(1)}, \dots, z_2^{T(n)}]^T$, $\underline{u} \triangleq [u^{T(1)}, \dots, u^{T(n)}]^T$, $\underline{\Delta f}_\eta(z_1) \triangleq [\Delta f_\eta(z_1)^{T(1)}, \dots, \Delta f_\eta(z_1)^{T(n)}]^T$, $\underline{z}_{1d} \triangleq [z_{1d}^{T(1)}, \dots, z_{1d}^{T(n)}]^T$ with $z_1^{(i)}, z_2^{(i)}, u^{(i)}, \Delta f_\eta(z_1)^{(i)}, z_{1d}^{(i)} \in \mathbb{R}^3$ and $M \triangleq L + \text{diag}\{a_{i0}, \dots, a_{n0}\}$ in which L is the Laplacian matrix and a_{i0} for is a positive constant if the leader is a neighbor of agent i and $a_{i0} = 0$ otherwise.

It is worth noting that $z_{1d}^{(i)}$ before was a function of desired trajectories, i.e. $z_{1d}^{(i)} = f(z_{1d}^{(0)}, \dots, z_{1d}^{(n)})$ for $i = 0, \dots, n$. However, in this scheme, dynamics of $z_{1d}^{(i)}$ depends on the actual states, i.e. $z_{1d}^{(i)} = f(z_1^{(0)}, \dots, z_1^{(n)})$ for $i = 0, \dots, n$. Therefore, their dynamics should be taken into account for the stability analysis as done in the Lyapunov function analysis.

To theoretically show the stability of the semi-decentralized control scheme for the multi-agent team, in the candidate Lyapunov function, all states of all agents should be taken into account. Therefore, the candidate Lyapunov function is chosen as

$$V = \frac{1}{2} (\underline{S}_1^T \underline{S}_1 + \underline{S}_2^T \underline{S}_2 + \underline{z}_{1d}^T (M \otimes I_3) \underline{z}_{1d}) \quad (3.5.5)$$

Differentiating V along the trajectories of the closed-loop system given by equation (3.5.4) yields

$$\begin{aligned}
\dot{V} &= \underline{S}_1^T (z_2 + \alpha_1 (M \otimes I_3) z_1) \\
&\quad + \underline{S}_2^T \left(\underline{u} + \underline{\Delta f}_\eta(z_1) + \alpha_2 (M \otimes I_3) z_2 + (\lambda_1 \otimes I_n) (z_2 + \alpha_1 (M \otimes I_3) z_1) \right) \\
&\quad + \underline{z}_{1d}^T (M \otimes I_3) \dot{z}_{1d}
\end{aligned} \tag{3.5.6}$$

Based on Lemma 2.2, for a closed-loop system as $\dot{z} = -\alpha(M \otimes I_m)z$ the consensus is reached, since $(M \otimes I_3)$ is a positive semi-definite matrix. As in Section 3.2, since the dynamics of the desired trajectory in equation (3.5.4) is defined as $\dot{z}_{1d} \triangleq \dot{z}_1 = -\alpha_1(M \otimes I_3)z_1$, we can rewrite $\underline{z}_{1d}^T (M \otimes I_3) \dot{z}_{1d}$ as $-\alpha_1 \underline{z}_1^T (M \otimes I_3)^2 z_1$. Using this inference and applying the controller $\underline{u} = -\lambda_2 \underline{S}_2 - \frac{1}{2\varepsilon} (\underline{S}_2 \circ \underline{P}) - \alpha_2 (M \otimes I_3) z_2 - \lambda_1 (z_2 + \alpha_1 (M \otimes I_3) z_1)$ and $\underline{z}_2 = -\lambda_1 \underline{S}_1 - \alpha_1 (M \otimes I_3) z_1$ leads to

$$\begin{aligned}
\dot{V} &= -\underline{S}_1^T (\lambda_1 \otimes I_n) \underline{S}_1 + \underline{S}_1^T \underline{S}_2 - \underline{S}_2^T (\lambda_2 \otimes I_n) \underline{S}_2 + \underline{S}_2^T \underline{\Delta f}_\eta(z_1) \\
&\quad - \frac{1}{2\varepsilon} \underline{S}_2^T (\underline{S}_2 \circ \underline{P}) - \alpha_1 \underline{z}_1^T (M \otimes I_3)^2 z_1
\end{aligned} \tag{3.5.7}$$

As in Section 2.3.3, using Young's inequality, it can be shown that

$$\dot{V} \leq -\underline{S}_1^T (\lambda_1 \otimes I_n) \underline{S}_1 + \underline{S}_1^T \underline{S}_2 - \underline{S}_2^T (\lambda_2 \otimes I_n) \underline{S}_2 - \alpha_1 \underline{z}_1^T (M \otimes I_3)^2 z_1 + 3n \frac{\varepsilon}{2} \tag{3.5.8}$$

As illustrated in Section 2.3.3, the proper choice of λ_1 , λ_2 , and ε leads to $\dot{V} \leq 0$ in $\mathbb{S} = \{(\underline{S}_1, \underline{S}_2) \mid -\underline{S}_1^T (\lambda_1 \otimes I_n) \underline{S}_1 + \underline{S}_1^T \underline{S}_2 - \underline{S}_2^T (\lambda_2 \otimes I_n) \underline{S}_2 - \alpha_1 \underline{z}_1^T (M \otimes I_3)^2 z_1 + 3n \frac{\varepsilon}{2} \leq 0\}$ which indicates the system is locally stable and all $z_1^{(i)}$ and $z_2^{(i)}$ are bounded. \blacksquare

3.6 DSC-based Semi-Decentralized Control Scheme with Consensus Algorithm Using Desired Trajectories

In this section, another novel DSC-based semi-decentralized control scheme combined with a consensus-based cooperative algorithm is proposed for the problem of formation path-tracking of a group of heterogeneous autonomous underwater vehicles. The design specifications of this scheme is presented in Section 3.2 and the objective and all assumptions of Section 3.5 are considered. The difference of this semi-decentralized scheme with the one in Section 3.5 is that here instead of the position and velocity states, agents communicate their desired relative positions with their neighbors to determines their desired relative positions.

In this scheme, for the agents with the model represented in equations (3.4.1) and (3.4.2), the controller of each agent is designed as

$$\tau_\eta^{(i)} = M_\eta^{(i)} \left(\dot{\bar{z}}_2^{(i)} - \lambda_2^{(i)} S_2^{(i)} - \frac{S_2^{(i)} \circ P^{(i)}}{2\varepsilon} \right) + C_\eta^{(i)}(z_1^{(i)}, z_2^{(i)})z_2^{(i)} + D_\eta^{(i)}(z_1^{(i)})z_2^{(i)} \quad (3.6.1)$$

with $S_2^{(i)} = z_2^{(i)} - \bar{z}_2^{(i)}$, $P^{(i)} = \rho_\eta^{(i)} \circ \rho_\eta^{(i)}$ where $\rho_\eta^{(i)} = [\rho_{\eta(1)}^{(i)} \ \rho_{\eta(2)}^{(i)} \ \rho_{\eta(3)}^{(i)}]^T$, and

$$\bar{z}_2^{(i)} = \dot{z}_{1d}^{(i)} - \lambda_1^{(i)} (z_1^{(i)} - z_{1d}^{(i)}) \quad (3.6.2)$$

and

$$\dot{\bar{z}}_2^{(i)} = -\lambda_1^{(i)} (z_2^{(i)} - \dot{z}_{1d}^{(i)}) + \ddot{z}_{1d}^{(i)} \quad (3.6.3)$$

in which the trajectory $z_{1d}^{(i)}$ is obtained from

$$\dot{z}_{1d}^{(i)} = -\xi_1 \left(\sum_{j \in N_i} (z_{1d}^{ij} - \sigma^{ij}) + a_{i0}(z_{1d}^{i0} - \sigma^{i0}) \right) - \xi_2 \frac{\sum_{j \in N_i} (z_{1d}^{ij} - \sigma^{ij}) + a_{i0}(z_{1d}^{i0} - \sigma^{i0})}{\left\| \sum_{j \in N_i} (z_{1d}^{ij} - \sigma^{ij}) + a_{i0}(z_{1d}^{i0} - \sigma^{i0}) \right\| + \epsilon}$$

as explained in Section 3.2.

3.6.1 Stability Analysis

Theorem 3.3. Suppose that Assumptions 3.1, 3.2, and 3.3 are valid, and based on Theorem 3.1. $|z_{1d}^{(i)} - z_{1d}^{(j)}| \rightarrow \sigma^{ij}$ as $t \rightarrow \infty$. By using controller given by equation (3.6.1) for a group of agents with dynamics given by equations (2.2.15) and (2.2.16), with proper choices of $\lambda_1^{(i)}$, $\lambda_2^{(i)}$, and ε , the team of agents is stable and all $z_1^{(i)}$ and $z_2^{(i)}$ are bounded.

Proof. Without loss of generality, assume that all $\lambda_1^{(i)}$ s are equal, and the same for $\lambda_2^{(i)}$ s. Considering $\underline{S}_1 \triangleq [S_1^{T(1)}, \dots, S_1^{T(n)}]^T$ and $\underline{S}_2 \triangleq [S_2^{T(1)}, \dots, S_2^{T(n)}]^T$ as the column stack vectors of the first and second error surfaces with $S_1^{(i)}, S_2^{(i)} \in \mathbb{R}^3$, for the closed-loop error dynamics of the group of agents given by

$$\begin{cases} \dot{\underline{S}}_1 = \underline{z}_2 - \dot{\underline{z}}_{1d} \\ \dot{\underline{S}}_2 = \underline{u} + \underline{\Delta f}_\eta(\underline{z}_1) - (\ddot{\underline{z}}_{1d} - (\lambda_1 \otimes I_n)(\underline{z}_2 - \dot{\underline{z}}_{1d})) \end{cases} \quad (3.6.4)$$

where I_3 is the 3×3 identity matrix, $\underline{z}_1 \triangleq [z_1^{T(1)}, \dots, z_1^{T(n)}]^T$, $\underline{z}_2 \triangleq [z_2^{T(1)}, \dots, z_2^{T(n)}]^T$, $\underline{u} \triangleq [u^{T(1)}, \dots, u^{T(n)}]^T$, $\underline{\Delta f}_\eta(\underline{z}_1) \triangleq [\Delta f_\eta(z_1)^{T(1)}, \dots, \Delta f_\eta(z_1)^{T(n)}]^T$, $\dot{\underline{z}}_{1d} \triangleq [\dot{z}_{1d}^{T(1)}, \dots, \dot{z}_{1d}^{T(n)}]^T$, and $\ddot{\underline{z}}_{1d} \triangleq [\ddot{z}_{1d}^{T(1)}, \dots, \ddot{z}_{1d}^{T(n)}]^T$ with $z_1^{(i)}, z_2^{(i)}, u^{(i)}, \Delta f_\eta(z_1)^{(i)}, \dot{z}_{1d}^{(i)}, \ddot{z}_{1d}^{(i)} \in \mathbb{R}^3$, the candidate Lyapunov function is chosen as

$$V = \frac{1}{2} (\underline{S}_1^T \underline{S}_1 + \underline{S}_2^T \underline{S}_2) \quad (3.6.5)$$

Differentiating V along the trajectories of the closed-loop system given by equation (3.6.4) leads to

$$\dot{V} = \underline{S}_1^T (\underline{z}_2 - \dot{\underline{z}}_{1d}) + \underline{S}_2^T \left(\underline{u} + \underline{\Delta f}_\eta(\underline{z}_1) - (\ddot{\underline{z}}_{1d} - (\lambda_1 \otimes I_n)(\underline{z}_2 - \dot{\underline{z}}_{1d})) \right) \quad (3.6.6)$$

Applying $\underline{u} = -(\lambda_2 \otimes I_n) \underline{S}_2 - \frac{1}{2\varepsilon} (\underline{S}_2 \circ \underline{P}) + (\ddot{\underline{z}}_{1d} - (\lambda_1 \otimes I_n)(\underline{z}_2 - \dot{\underline{z}}_{1d}))$ and $\bar{\underline{z}}_2 = -(\lambda_1 \otimes I_n) \underline{S}_1 + \dot{\underline{z}}_{1d}$ leads to

$$\dot{V} = -\underline{S}_1^T (\lambda_1 \otimes I_n) \underline{S}_1 + \underline{S}_1^T \underline{S}_2 - \underline{S}_2^T (\lambda_2 \otimes I_n) \underline{S}_2 + \underline{S}_2^T \underline{\Delta f}_\eta(\underline{z}_1) - \frac{1}{2\varepsilon} \underline{S}_2^T (\underline{S}_2 \circ \underline{P}) \quad (3.6.7)$$

As in Section 2.3.3, using Young's inequality, it can be shown that

$$\dot{V} \leq -\underline{S}_1^T (\lambda_1 \otimes I_n) \underline{S}_1 + \underline{S}_1^T \underline{S}_2 - \underline{S}_2^T (\lambda_2 \otimes I_n) \underline{S}_2 + 3n\frac{\varepsilon}{2} \quad (3.6.8)$$

As explained in Section 2.3.3, the proper choice of λ_1 , λ_2 , and ε leads to $\dot{V} \leq 0$ in $\mathbb{S} = \{(S_1, S_2) | -\underline{S}_1^T (\lambda_1 \otimes I_n) \underline{S}_1 + \underline{S}_1^T \underline{S}_2 - \underline{S}_2^T (\lambda_2 \otimes I_n) \underline{S}_2 + 3n\frac{\varepsilon}{2} \leq 0\}$ which indicates the system is locally stable and all $z_1^{(i)}$ and $z_2^{(i)}$ are bounded. ■

3.7 Simulation Results

In this section, to analyze and evaluate the effectiveness of the performance of the cooperative control strategies addressed previously in this chapter, various scenarios have been conducted as well as comparative simulations between different schemes. Moreover, it has been assumed that the communication of data between agents in all schemes is ideal with zero time delay and loss. The scenarios defined in this section vary in the number of agents, communication topologies, initial values, and reference trajectories.

In all scenarios, two sets of model parameters taken from [124] and [125] are used. For the first model of AUV, which is used for agents #1, #2, and #3, the system matrices are

$$M_1 = \begin{bmatrix} 80.026 & 0 & 0 \\ 0 & 80.041 & 0.0139 \\ 0 & 0.0139 & 10.011 \end{bmatrix} ; \quad D_1 = \begin{bmatrix} 0.72 & 0 & 0 \\ 0 & 0.8896 & 7.25 \\ 0 & 0.0313 & 1.9 \end{bmatrix}$$

$$C_1(\eta) = \begin{bmatrix} 0 & 0 & -(80.041v + 0.0139r) \\ 0 & 0 & 80.026u \\ 80.041v + 0.0139r & -80.026u & 0 \end{bmatrix}$$

and for the second model, which is used for agents #4, #5, #6, and #7, the system matrices are

$$M_2 = \begin{bmatrix} 25.8 & 0 & 0 \\ 0 & 33.8 & 1.0115 \\ 0 & 1.0115 & 2.76 \end{bmatrix} ; \quad D_2 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 7 & 0.1 \\ 0 & 0.1 & 0.5 \end{bmatrix}$$

$$C_2(\eta) = \begin{bmatrix} 0 & 0 & -(33.8v + 1.0115r) \\ 0 & 0 & 25.8u \\ 33.8v + 1.0115r & -25.8u & 0 \end{bmatrix}$$

For all vehicles the uncertainties are considered as

$$\Delta f_\eta^{(i)}(\eta) = \begin{bmatrix} \psi^{(i)} \sin(\psi^{(i)}) \\ \psi^{(i)} \cos(\psi^{(i)}) \\ 0 \end{bmatrix}$$

which is locally Lipschitz and state-dependent nonlinearity such that $\Delta f_\eta(0) = 0$ and also it satisfies

$$|\psi^{(i)} \sin(\psi^{(i)})| \leq |\psi^{(i)}| \quad ; \quad |\psi^{(i)} \cos(\psi^{(i)})| \leq |\psi^{(i)}|$$

In the simulations tests, the desired formation positioning for all schemes and the network topology for the semi-decentralized scheme is shown in Figure 3.2 where the desired relative distances between neighbors are considered as $(v_2, v_3) = (v_5, v_6) = 6$ and $(v_1, v_2) = (v_3, v_4) = (v_4, v_5) = (v_6, v_1) = 3\sqrt{2}$, unless we specify a different network topology for a certain part.

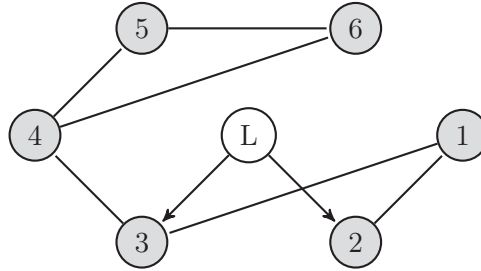


Figure 3.2: Network topology for a group of six vehicles with a virtual leader.

In addition, the initial conditions for each agent is given in Table 3.1. The controller parameters are considered as $\varepsilon = 10$, $\xi_1 = 2$, $\xi_2 = 4$, $\lambda_1 = 1.8$, and $\lambda_2 = 2.9$ for all schemes unless they are assigned different values for a certain simulation test. It is worth mentioning that these parameters are chosen based on a trade-off between performance and affordability of control gains.

Table 3.1: Initial conditions

Vehicle	Position and Euler Angles	Linear and Angular Velocity
1	$\eta(0) = [0.3 \ 1 \ \frac{\pi}{3}]^T$	$\nu(0) = [0.1 \ 0.04 \ 0.4]^T$
2	$\eta(0) = [0.4 \ 1.2 \ \frac{\pi}{6}]^T$	$\nu(0) = [0.02 \ 0.12 \ 0.03]^T$
3	$\eta(0) = [0.3 \ 0.1 \ \frac{\pi}{3}]^T$	$\nu(0) = [0.05 \ 0.02 \ 0.15]^T$
4	$\eta(0) = [-1 \ -0.5 \ \frac{\pi}{2}]^T$	$\nu(0) = [0.1 \ 0.3 \ 0.04]^T$
5	$\eta(0) = [-0.1 \ -0.2 \ \frac{2\pi}{3}]^T$	$\nu(0) = [0.3 \ 0.01 \ 0.4]^T$
6	$\eta(0) = [-0.2 \ 0.5 \ \frac{\pi}{4}]^T$	$\nu(0) = [0.4 \ 0.04 \ 0.6]^T$

3.7.1 Using Actual States vs. Using Desired Trajectories in Consensus Algorithm for DSC-based Semi-decentralized Control Scheme

The purpose of this part is to compare the effect of using states of agents and the desired trajectories in consensus algorithm for DSC-based semi-decentralized control scheme. For the simulations conducted in this part, the reference command is considered as

$$\eta_{ref} = \left[9 \cos\left(\frac{t}{9}\right) \ \frac{t}{3} \ \frac{\pi}{6} \right]^T$$

For the semi-decentralized control scheme with consensus algorithm using states of agents the control parameters for all agents are assigned as $\alpha_1 = \alpha_2 = 7$, $\lambda_1 = 6$, and $\lambda_2 = 11$, and for the semi-decentralized control scheme with consensus algorithm using the desired trajectories the control parameters for all agents are assigned as $\lambda_1 = 1.8$, and $\lambda_2 = 2.9$. The initial values, network topology, model parameters, and uncertainties are considered as given in Section 3.7.

Tracking error trajectories of position and orientation of each agent for both types of semi-

decentralized schemes are presented in Figures 3.3 to 3.5. Based on these simulation results, in overall, for the transient time, the tracking errors for the case that the states of agents are used in consensus algorithm are mostly lower than the case that the desired trajectories are used. However, in steady state time the results are reversed and the errors for the case that the states of agents are used in consensus algorithm are higher than the case that the desired trajectories are used.

In Figures 3.6 to 3.11, the control input signals for both cases are represented with their zoomed version. For all agents, the absolute maximum control effort costs in the case that the states of agents are used in consensus algorithm are much higher than the other case which is a draw back for this scheme. The reason of the high cost functions is that to reach acceptable results it is required to increase the control gains.

To quantitatively analyze to performance of both semi-decentralized schemes, the response characteristics such as the maximum errors, steady state errors, and the maximum absolute control efforts are summarized in Table 3.3 separately for each agent. For the team level analysis of both schemes, the root mean squares (RMS) of the response characteristics given in Table 3.3 are presented in Table 3.2.

Based on the results of this part, although the performance of both methods are closed to each other, the individual and team performances of the semi-decentralized scheme in which the desired trajectories are used in the consensus algorithm is superior to the one that the states of agents are used in consensus algorithm. On the other hand, using the desired trajectories in the consensus algorithm has this advantage that in this scheme the size of data that need to be transmitted among agents is half of the other case. Because of all these advantages, henceforth the semi-decentralized scheme in which the desired trajectories are used in the

consensus algorithm is chosen in the rest of this thesis.

Table 3.2: Team level RMS analysis of maximum error, steady state error, and maximum absolute control effort for semi-decentralized schemes using desired trajectories and states of agents in consensus algorithm

	RMS Maximum error (m)	RMS e_{ss} (sec)	RMS Max Abs. C.E. ($N.m$)
Desired trajectory	5.7926	0.0888	77.0162
States of agents	5.2777	0.2436	173.8437

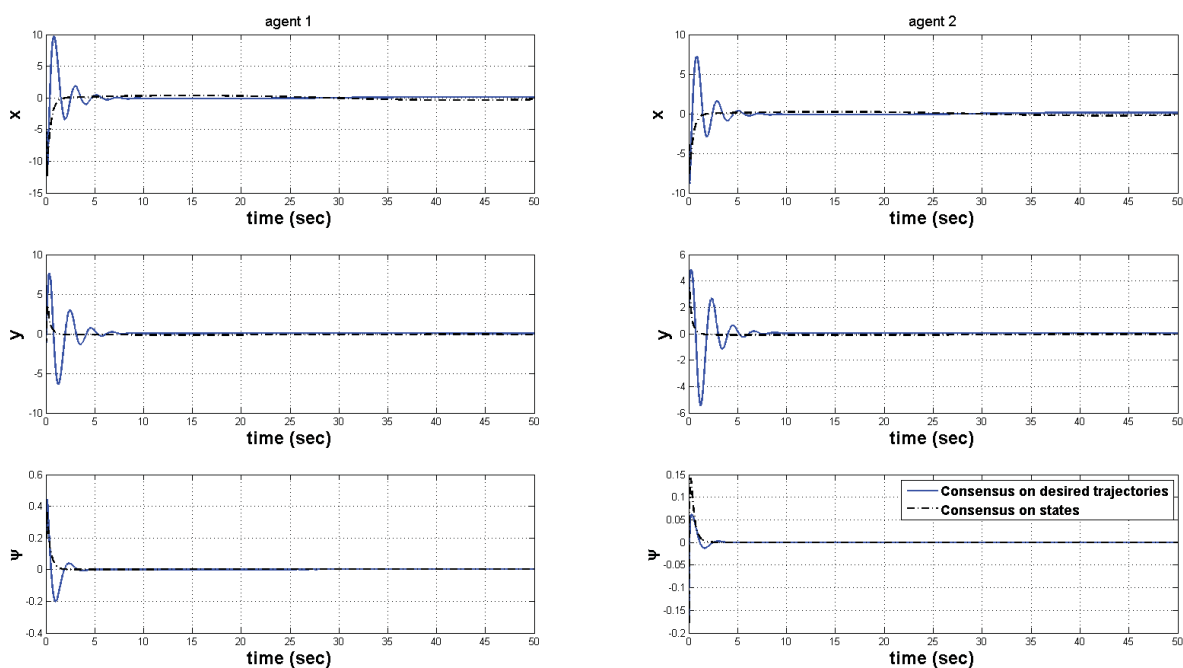


Figure 3.3: Error signals of agents #1 and #2 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.

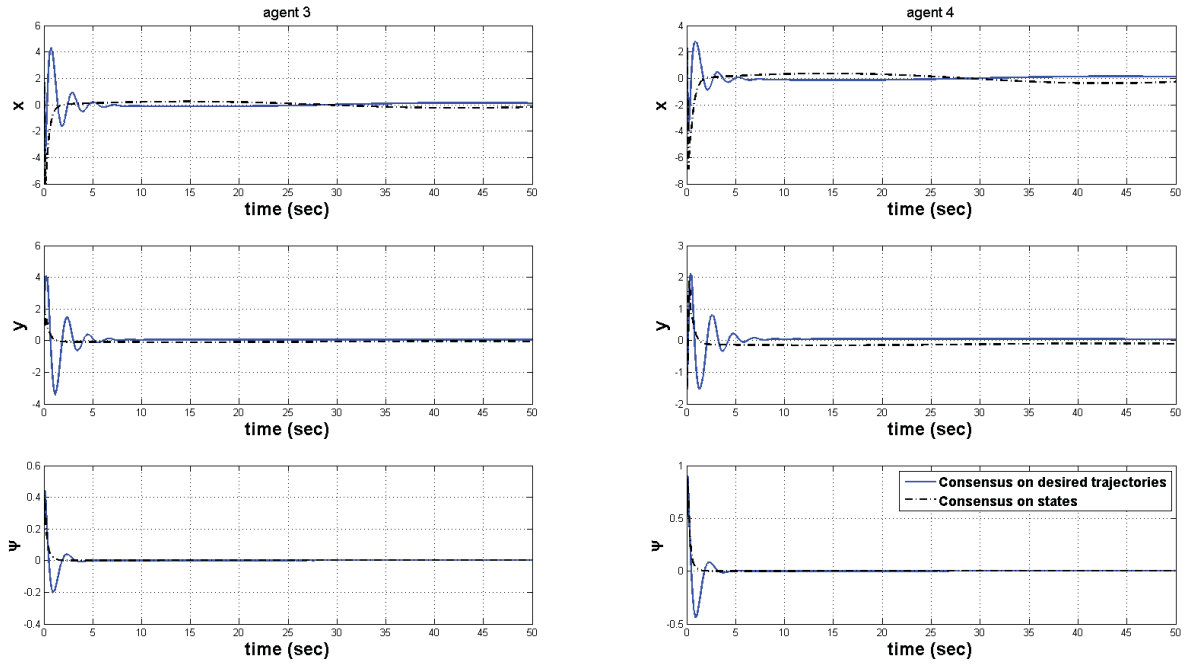


Figure 3.4: Error signals of agents #3 and #4 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.

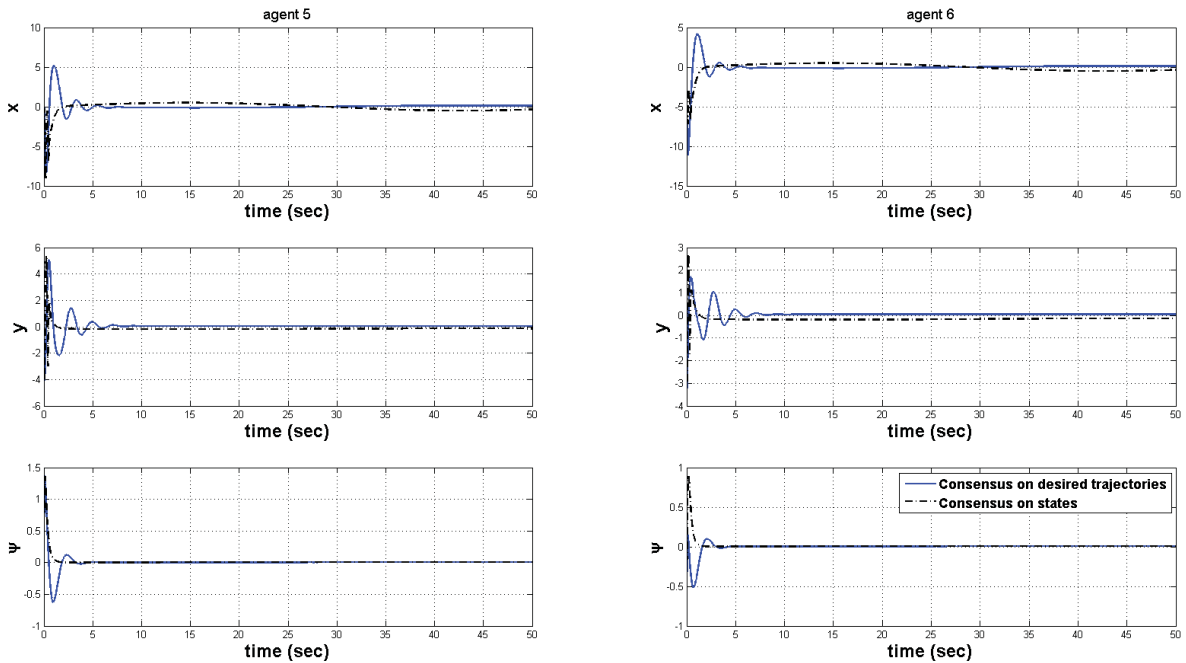


Figure 3.5: Error signals of agents #5 and #6 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.

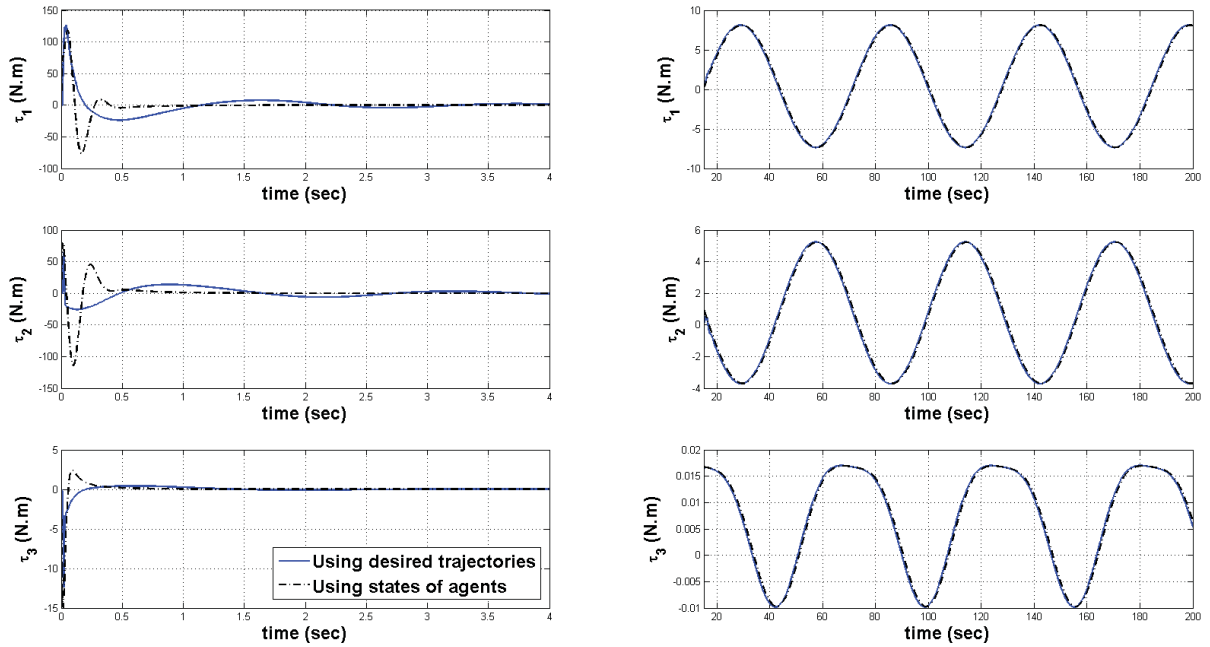


Figure 3.6: Control efforts of agent #1 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.

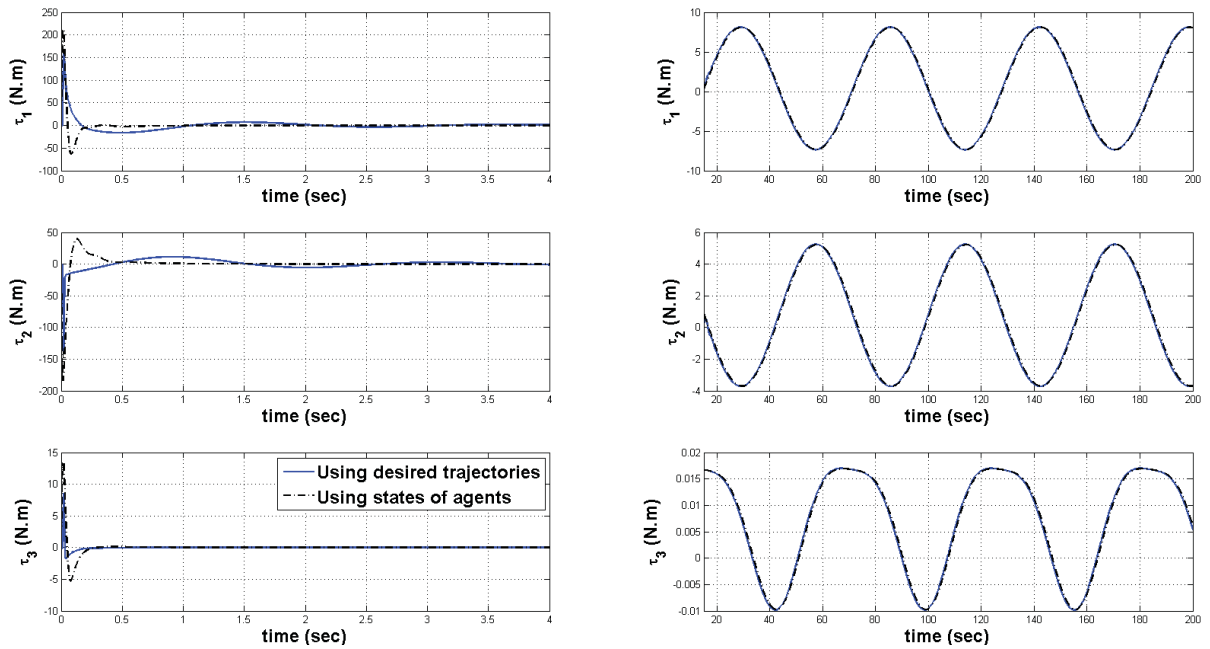


Figure 3.7: Control efforts of agent #2 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.

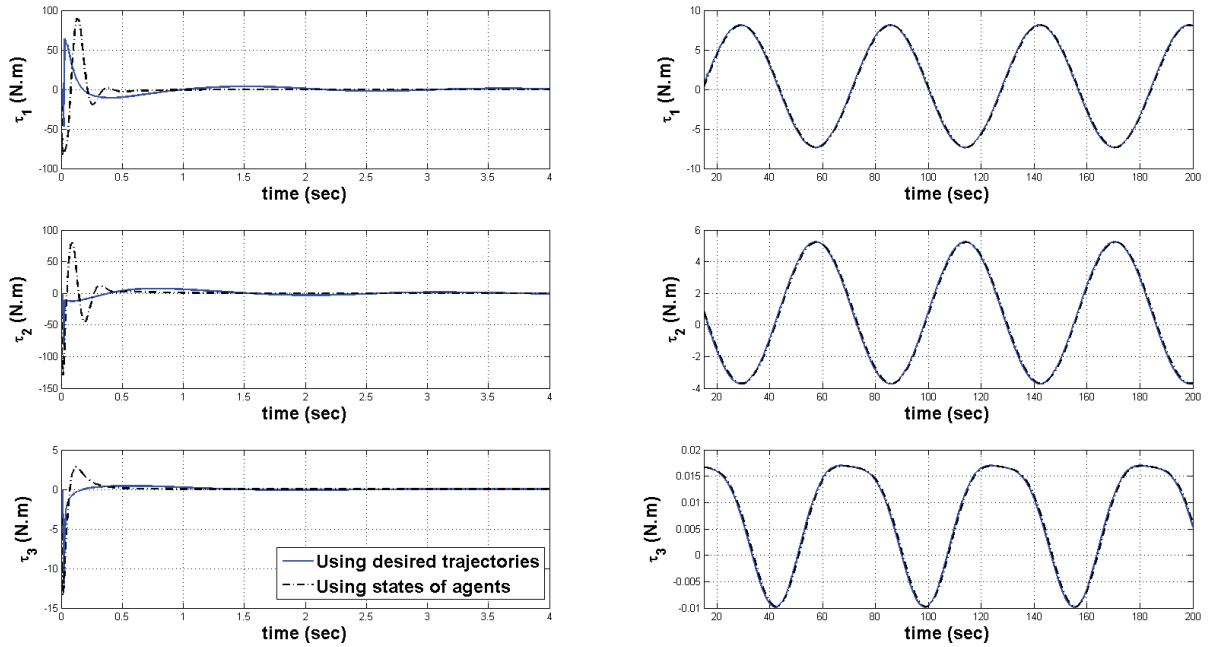


Figure 3.8: Control efforts of agent #3 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.

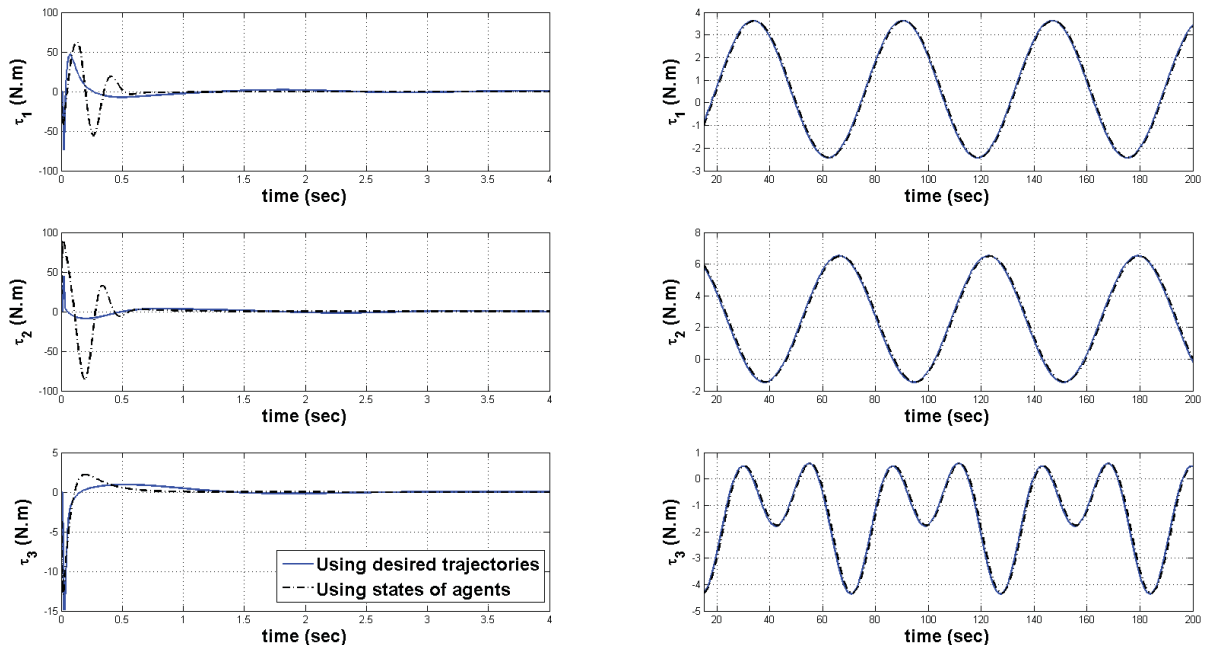


Figure 3.9: Control efforts of agent #4 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.

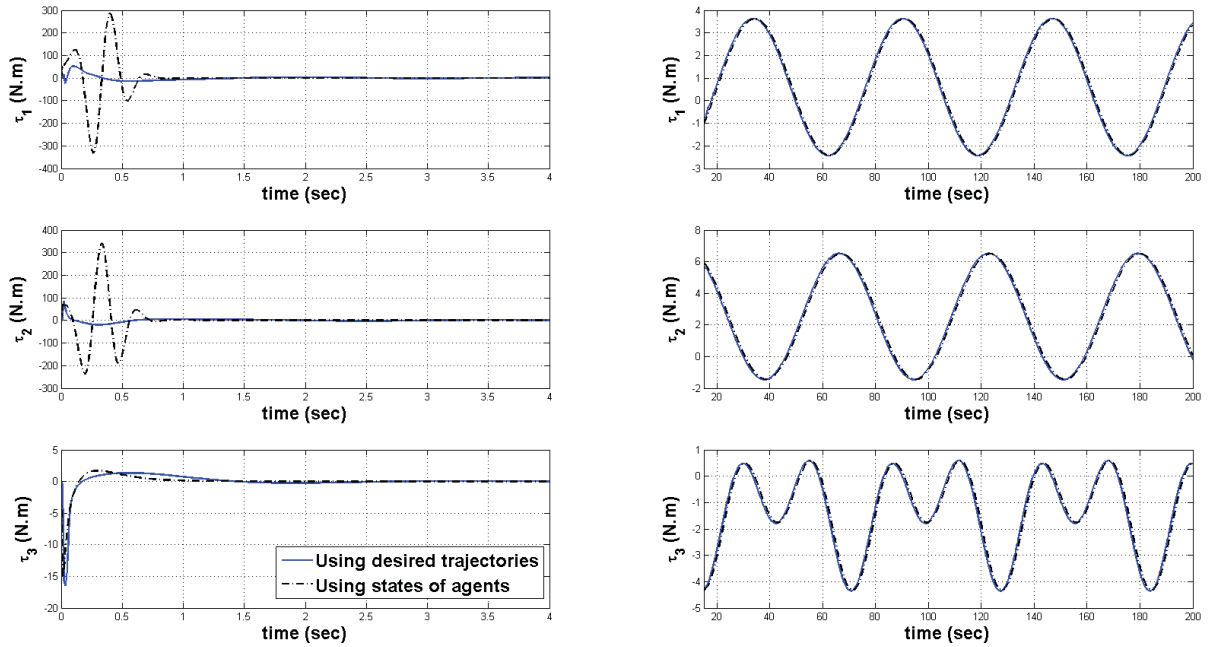


Figure 3.10: Control efforts of agent #5 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.

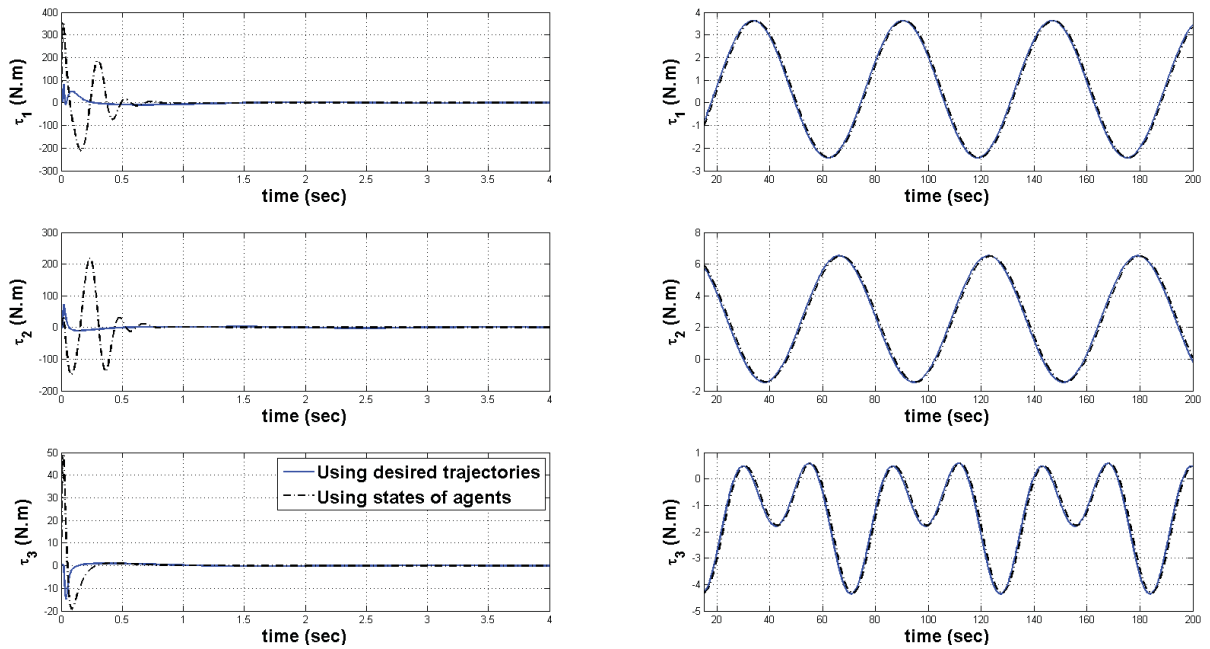


Figure 3.11: Control efforts of agent #6 in fault-free situation for semi-decentralized schemes using states of agents and desired trajectories in consensus algorithm.

Table 3.3: Quantitative analysis of maximum errors, steady state errors, and maximum absolute control efforts for semi-decentralized schemes using states of agents and desired trajectories in

consensus algorithm

		Consensus algorithm on	Maximum error (m)	Steady state error (m)	Max abs. C.E. ($N.m$)
agent 1	x	Desired trajectory	12.2243	0.144	126.28
		States of agents	12.259	0.374	118.62
	y	Desired trajectory	7.6578	0.0543	64.82
		States of agents	3.5972	0.15	114.61
	ψ	Desired trajectory	0.4424	1.2×10^{-6}	12.33
		States of agents	0.3704	2.1×10^{-7}	14.91
agent 2	x	Desired trajectory	8.8321	0.1437	161.13
		States of agents	7.5687	0.2495	210.57
	y	Desired trajectory	5.4592	0.0543	139.82
		States of agents	3.4681	0.1084	183.3
	ψ	Desired trajectory	0.178	1.2×10^{-6}	8.62
		States of agents	0.1807	1.4×10^{-7}	13.4
agent 3	x	Desired trajectory	4.3052	0.1435	64.16
		States of agents	5.9491	0.2509	89.47
	y	Desired trajectory	4.0818	0.0543	80.81
		States of agents	2.1817	0.1089	128.7
	ψ	Desired trajectory	0.4435	1.2×10^{-6}	11.98
		States of agents	0.4116	1.4×10^{-7}	13.31
agent 4	x	Desired trajectory	3.9081	0.1445	73.41
		States of agents	6.8667	0.3784	62.75
	y	Desired trajectory	2.1221	0.0531	44.81
		States of agents	1.8809	0.1515	88.93
	ψ	Desired trajectory	0.898	7.88×10^{-4}	14.82
		States of agents	0.879	4.48×10^{-5}	12.92
agent 5	x	Desired trajectory	8.3599	0.1448	52.49
		States of agents	9.23	0.5062	332.6
	y	Desired trajectory	5.0708	0.0531	85.51
		States of agents	5.3997	0.1942	340.51
	ψ	Desired trajectory	1.3727	7.882×10^{-4}	16.39
		States of agents	1.3571	8.95×10^{-5}	14.89
agent 6	x	Desired trajectory	11.1146	0.1447	82.19
		States of agents	7.1569	0.509	353.64
	y	Desired trajectory	3.2104	0.0531	71.87
		States of agents	2.9488	0.1952	217.86
	ψ	Desired trajectory	0.759	7.87×10^{-4}	14.72
		States of agents	0.8892	8.95×10^{-5}	48.67

3.7.2 Quantitative Comparison of DSC-based Centralized, Decentralized, and Semi-decentralized Schemes

In order to be able to quantitatively compare the semi-decentralized, decentralized, and centralized DSC-based schemes, a scenario with zero initial conditions for all six states of all agents and a constant reference trajectory as

$$\eta_{ref} = \left[5 \ 5 \ \frac{\pi}{6} \right]^T$$

are considered. In this part, the network topology, desired formation, model of agents, and their uncertainties are assumed to be the same as given in Section 3.7. The error signals are represented in Figures 3.12 to 3.14 and the control input signals of all agents with their zoomed versions are represented in Figures 3.15 to 3.20. In addition, the response characteristics such as settling time, steady state error, maximum error, and maximum absolute control efforts are quantitatively summarized in Tables 3.5 to 3.10.

Based on the analytical results given in Tables 3.5 to 3.10, in the centralized scheme the settling times all position and orientation states of all agents are always lower in the centralized scheme, then in the decentralized scheme, and at last in semi-decentralized scheme.

Based on obtained results from analyzing steady state errors, it can be seen that for all three schemes the e_{ss} are very close and always bounded. To be more precise, in x and y states of all agents the steady state errors are lower in the centralized scheme, then in the decentralized scheme, and finally in the semi-decentralized scheme, but in ψ orientation the semi-decentralized scheme has the lowest e_{ss} , followed by the decentralized and then the centralized schemes.

Also, the maximum error in all position states of the centralized scheme has the lowest error among all schemes, and after that the semi-decentralized scheme has lower maximum errors than the decentralized scheme. For the maximum absolute control efforts (C.E.), the

centralized scheme leads to a notably higher control cost in the transient time in comparison to the other schemes. After that, the semi-decentralized scheme has the higher control effort. However, after the transient time, all schemes reach the same constant final amount for their control efforts.

The team level comparison of all schemes are presented in Table 3.4. For team level analysis, the root mean square of all characteristics given in Tables 3.5 to 3.10 are obtained for each scheme. In this table, all results explained above are confirmed.

Based on all these results, the centralized scheme has the best and optimal performance in comparison to other two schemes but with the cost of the high control effort. The performance of the decentralized scheme is pretty good too. However, because of the absence of communication between agents, in case one agent loses its communications with the command center, there is no chance to refine it and bring it back to the group, which is a major drawback to using this scheme in sensible missions. Finally, the performance of the semi-decentralized scheme is very close to the centralized scheme while it does not impose stringent communication requirements as in the centralized scheme and does not have the lack of communication as in the decentralized scheme, which makes it more applicable in practice.

Table 3.4: Team level RMS analysis of rise time, settling time, steady state error, maximum overshoot, and maximum and minimum control effort for all schemes of Section 3.7.2 in fault-free situation

	RMS <i>t_s (sec)</i>	RMS <i>e_{ss} (sec)</i>	RMS Max error (m)	RMS Max Abs. C.E. <i>(N.m)</i>
Semi-decentralized	6.752	0.0092	4.3096	73.3513
Centralized	3.665	0.001	4.0315	113.41
Decentralized	5.827	0.0044	4.7363	5.392

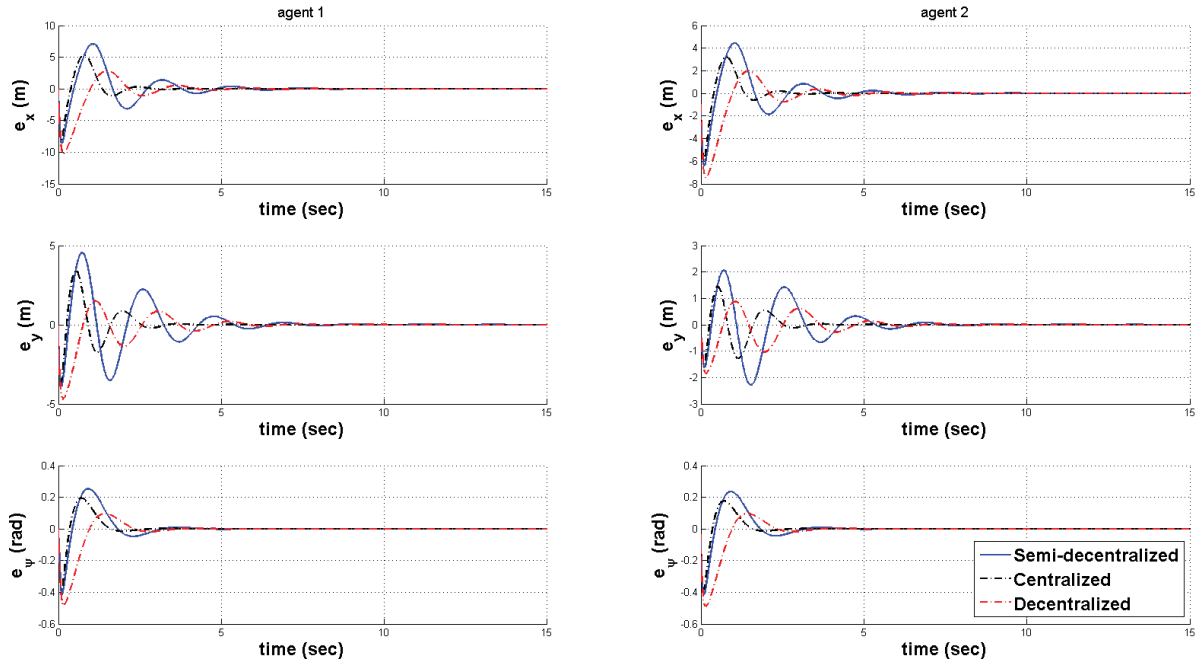


Figure 3.12: Error signals of agents #1 and #2 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.

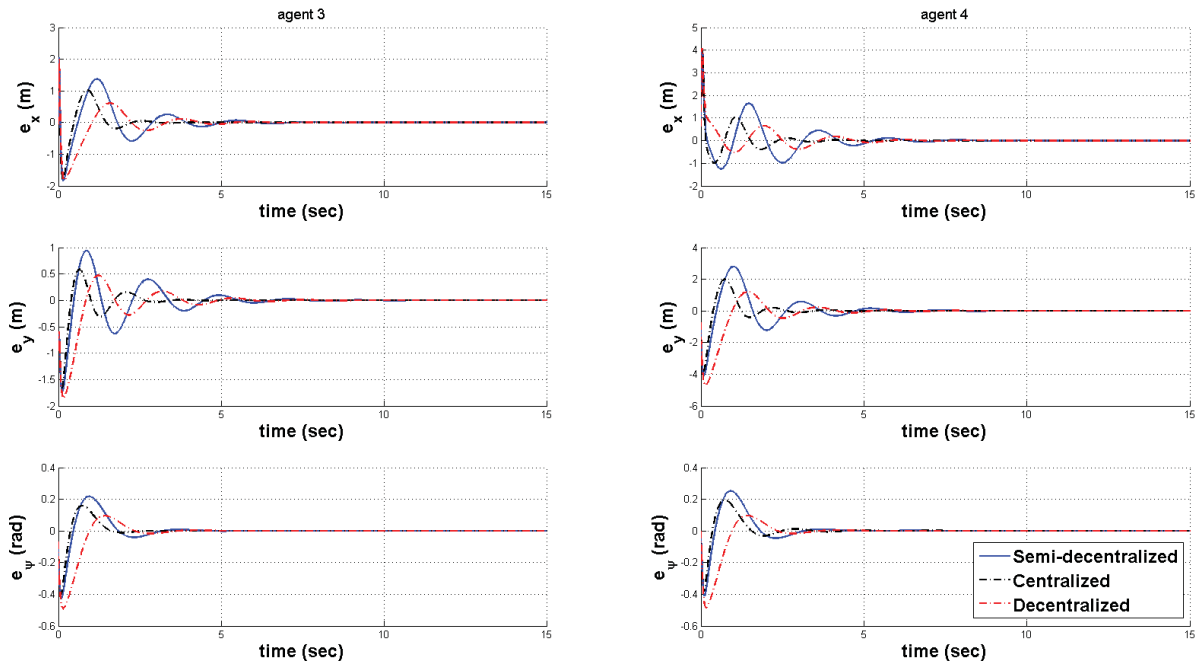


Figure 3.13: Error signals of agents #3 and #4 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.

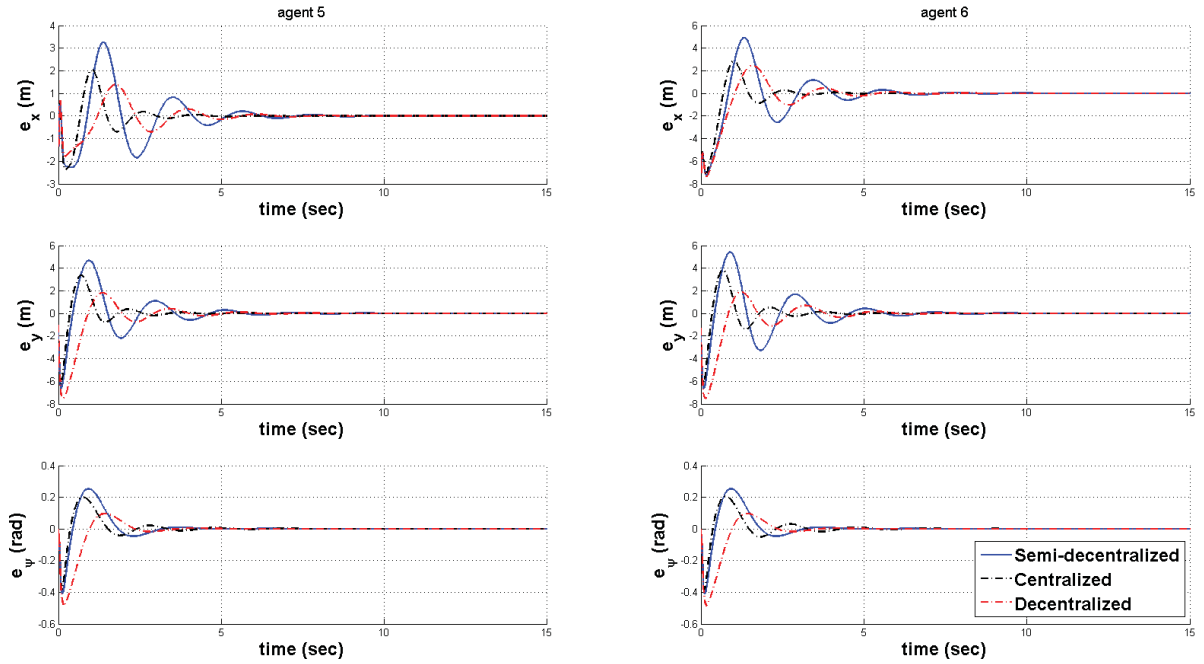


Figure 3.14: Error signals of agents #5 and #6 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.

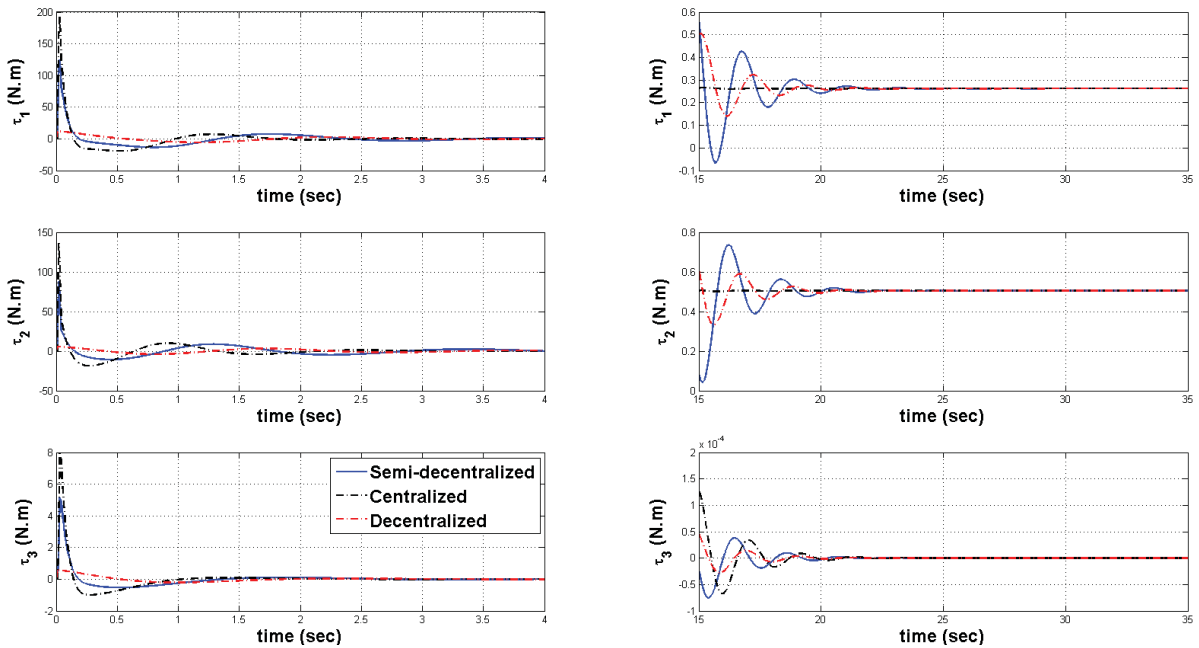


Figure 3.15: Control effort of agent #1 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.

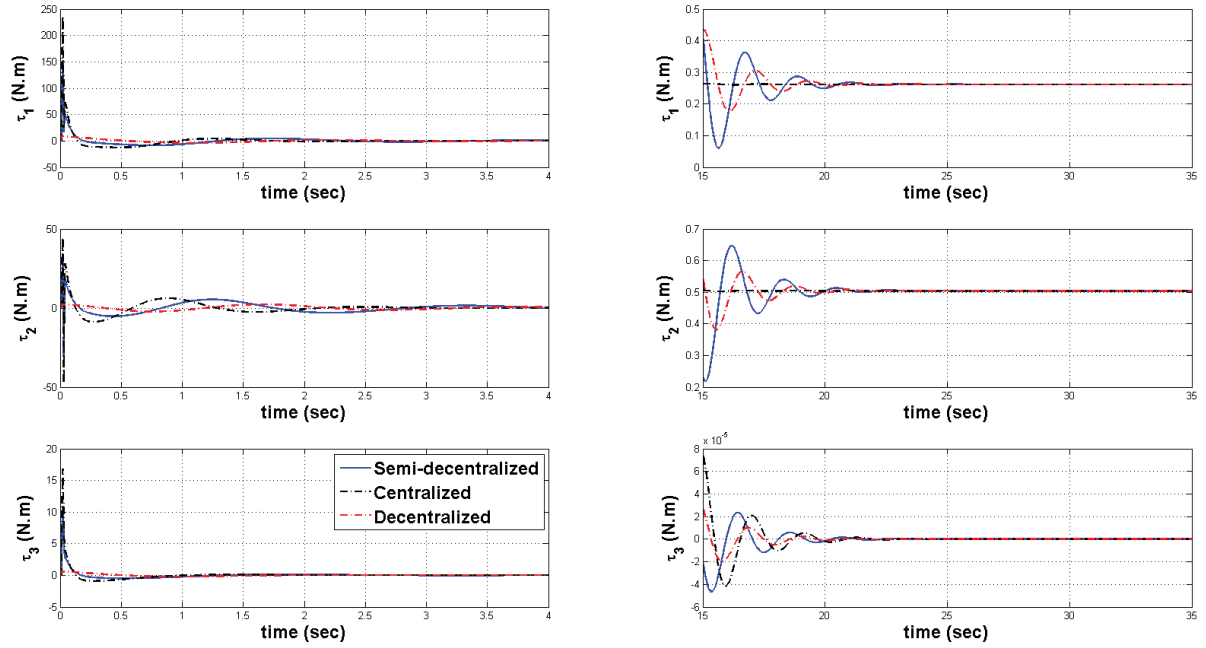


Figure 3.16: Control effort of agent #2 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.

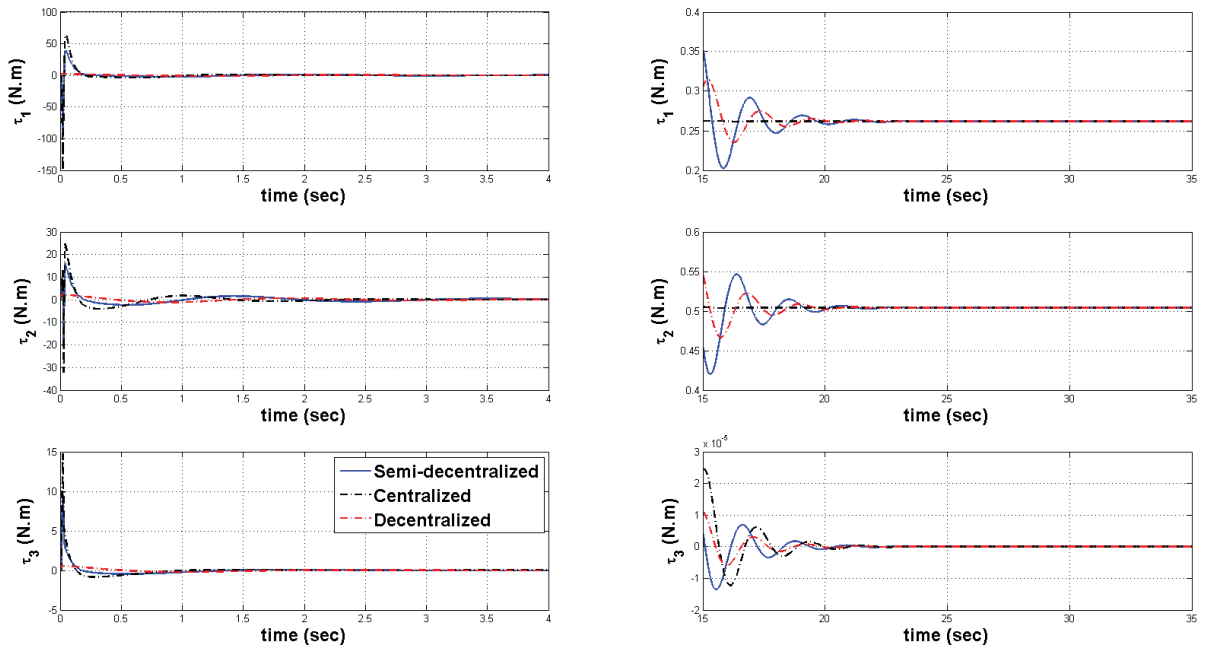


Figure 3.17: Control effort of agent #3 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.

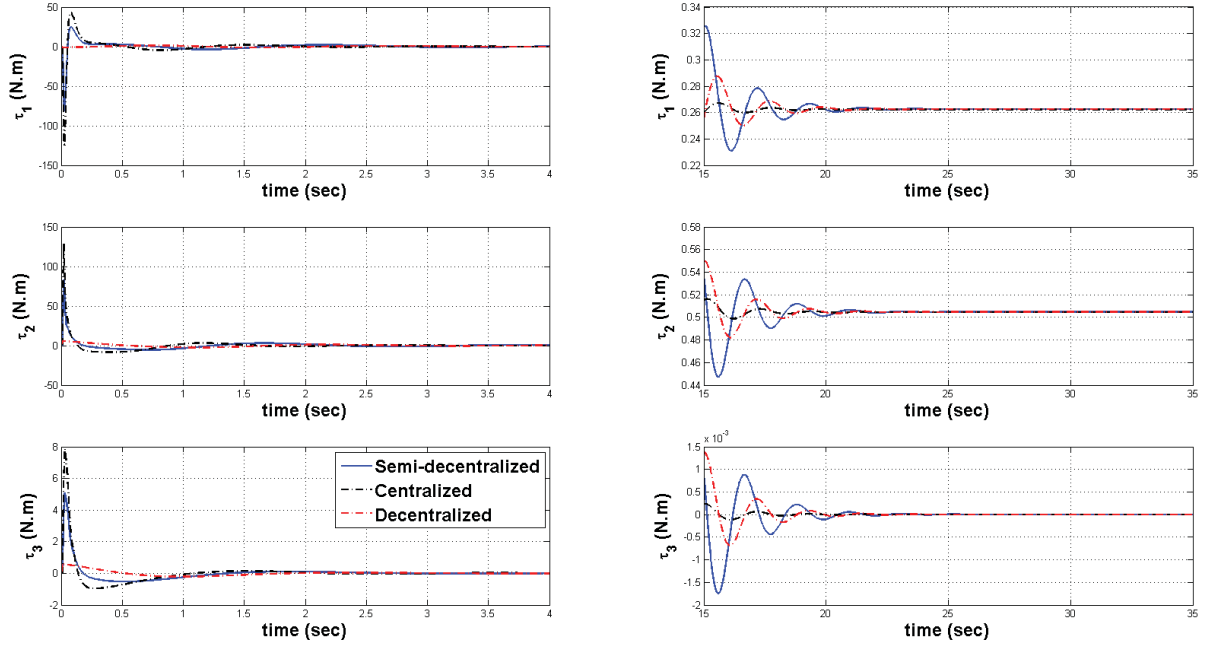


Figure 3.18: Control effort of agent #4 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.

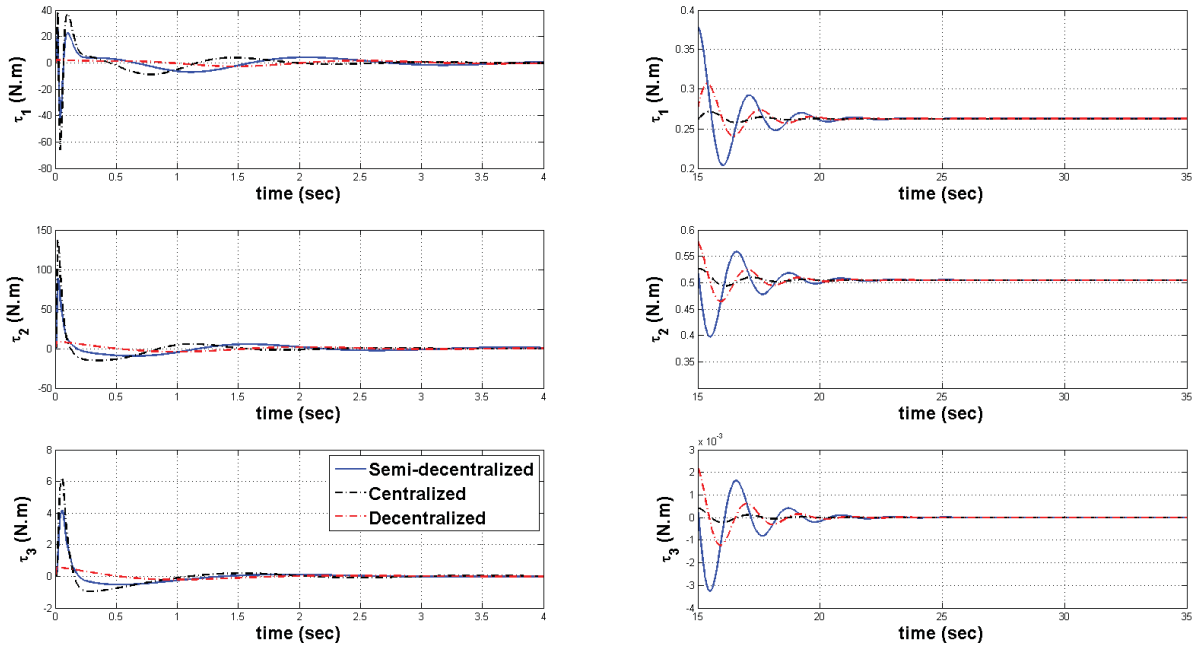


Figure 3.19: Control effort of agent #5 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.

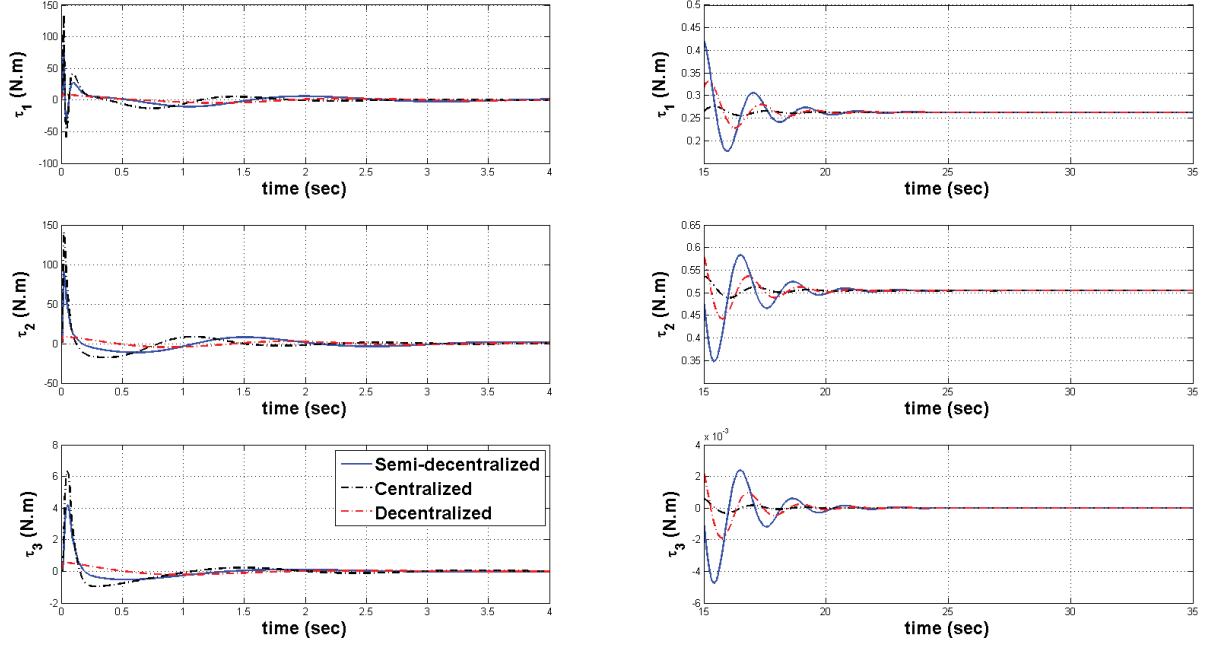


Figure 3.20: Control effort of agent #6 in fault-free situation for the semi-decentralized, centralized, and decentralized DSC-based schemes.

Table 3.5: Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #1

		Settling time (<i>sec</i>)	Steady state error (<i>m</i>)	Maximum error (<i>m</i>)	Max abs. C.E. (<i>N.m</i>)
x	Semi-decentralized	8.83	0.0122	8.6172	124.75
	Centralized	4.25	3.39×10^{-4}	8.0385	192.21
	Decentralized	7.26	0.0082	10.229	11.87
y	Semi-decentralized	9.13	0.0174	4.5421	88.32
	Centralized	3.95	6.34×10^{-4}	3.684	136.5
	Decentralized	7.69	0.0071	4.6737	5.39
ψ	Semi-decentralized	2.69	2.1×10^{-6}	0.4108	5.14
	Centralized	1.39	2.44×10^{-5}	0.3827	7.93
	Decentralized	2.06	4.65×10^{-6}	0.4848	0.565

Table 3.6: Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #2

		Settling time (<i>sec</i>)	Steady state error (<i>m</i>)	Maximum error (<i>m</i>)	Max abs. C.E. (<i>N.m</i>)
<i>x</i>	Semi-decentralized	7.77	0.0077	6.4149	153.19
	Centralized	3.56	0.0003	6.0229	236.91
	Decentralized	7.12	0.0057	7.4729	8.6328
<i>y</i>	Semi-decentralized	8.21	0.111	2.2916	30.604
	Centralized	3.76	0.0006	1.5288	47.38
	Decentralized	7.48	0.0053	1.8504	2.2681
<i>ψ</i>	Semi-decentralized	2.66	1.95×10^{-6}	0.4165	10.87
	Centralized	1.37	1.46×10^{-5}	0.3915	16.808
	Decentralized	2.05	4.6×10^{-6}	0.4902	0.565

Table 3.7: Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #3

		Settling time (<i>sec</i>)	Steady state error (<i>m</i>)	Maximum error (<i>m</i>)	Max abs. C.E. (<i>N.m</i>)
<i>x</i>	Semi-decentralized	6.71	0.0028	2.0446	97.02
	Centralized	2.8	0.0003	2.0446	149.93
	Decentralized	5.21	0.0014	2.0446	2.1582
<i>y</i>	Semi-decentralized	6.32	0.0039	1.7374	20.84
	Centralized	3.15	0.0006	1.6716	32.15
	Decentralized	5.63	0.0022	1.8558	2.1582
<i>ψ</i>	Semi-decentralized	2.65	2.3×10^{-6}	0.4232	9.56
	Centralized	1.37	5.7×10^{-6}	0.3998	14.79
	Decentralized	2.05	4.7×10^{-6}	0.4907	0.565

Table 3.8: Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #4

		Settling time (<i>sec</i>)	Steady state error (<i>m</i>)	Maximum error (<i>m</i>)	Max abs. C.E. (<i>N.m</i>)
<i>x</i>	Semi-decentralized	7.18	0.0056	4.0255	80.65
	Centralized	4.34	0.0011	4.0033	124.27
	Decentralized	6.51	0.0028	4.0642	1.41
<i>y</i>	Semi-decentralized	7.56	0.0071	4.0785	82.77
	Centralized	4.13	0.0017	3.8703	128.05
	Decentralized	6.93	0.0041	4.6936	5.39
<i>ψ</i>	Semi-decentralized	2.68	7.8×10^{-4}	0.4106	5.1
	Centralized	2.01	5.52×10^{-4}	0.3833	7.87
	Decentralized	2.06	7.9×10^{-4}	0.4862	0.565

Table 3.9: Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #5

		Settling time (<i>sec</i>)	Steady state error (<i>m</i>)	Maximum error (<i>m</i>)	Max abs. C.E. (<i>N.m</i>)
<i>x</i>	Semi-decentralized	8.12	0.0115	3.2509	42.39
	Centralized	4.51	0.0012	2.3536	66.04
	Decentralized	7.41	0.0039	1.7864	2.77
<i>y</i>	Semi-decentralized	8.57	0.0114	6.6034	89.07
	Centralized	5.01	0.002	6.3116	137.82
	Decentralized	7.01	0.0056	7.5128	8.632
<i>ψ</i>	Semi-decentralized	2.69	7.8×10^{-4}	0.4098	4.15
	Centralized	2.11	5.9×10^{-4}	0.382	6.23
	Decentralized	2.06	7.9×10^{-4}	0.4815	0.565

Table 3.10: Quantitative analysis of settling time, steady state error, maximum error, and maximum absolute control effort for all schemes in fault-free situation for agent #6

		Settling time (<i>sec</i>)	Steady state error (<i>m</i>)	Maximum error (<i>m</i>)	Max abs. C.E. (<i>N.m</i>)
<i>x</i>	Semi-decentralized	9.03	0.0162	7.3101	86.68
	Centralized	5.31	0.0014	7.1183	134.94
	Decentralized	7.35	0.6385	7.3615	8.63
<i>y</i>	Semi-decentralized	8.61	0.0157	6.6586	91.26
	Centralized	5.71	0.0022	6.3892	141.26
	Decentralized	7.75	0.0062	7.5253	8.63
<i>ψ</i>	Semi-decentralized	2.66	7.8×10^{-4}	0.4104	4.18
	Centralized	2.7	6.3×10^{-4}	0.3822	6.33
	Decentralized	2.06	7.9×10^{-4}	0.4826	0.565

3.7.3 DSC-based Semi-decentralized Scheme vs. Cooperative Schemes Based on Model-dependent Coordinated Tracking Algorithm in Fault-free Situation

The next comparative study is performed between the DSC-based semi-decentralized scheme and three cooperative schemes of model-dependent coordinated tracking algorithm, namely the centralized, decentralized, and semi-decentralized schemes.

For the simulation performed in this section, desired relative distances, model parameters and uncertainties, and the initial conditions for all for schemes are considered the same as Section 3.7. In this part, the reference trajectory is considered as

$$\eta_{ref} = \left[15 \sin \left(\frac{t}{8} \right) \left(\frac{t}{3} \right) \left(\frac{\pi}{6} \right) \right]^T$$

The rest of the simulation conditions for each scheme are as follows:

- For the DSC-based semi-decentralized scheme, the network topology is given in Figure 3.2 and the controller parameters of the DSC-based semi-decentralized scheme for all agents are considered as $\lambda_1 = 1.8$ and $\lambda_2 = 2.9$.
- For the centralized model-dependent coordinated tracking scheme, the concatenated model containing dynamic model of n agents with p states is derived as

$$M_{con}(\underline{q})\ddot{\underline{q}} + C_{con}(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + g_{con}(\underline{q}) = \underline{\tau} \quad (3.7.1)$$

where $\underline{q} \in \mathbb{R}^{(n \times p)}$ is the column stack vector of generalized coordinates, $M_{con}(\underline{q}) \in \mathbb{R}^{(n \times p) \times (n \times p)}$ is the concatenated symmetric positive-definite inertia matrix, $C_{con}(\underline{q}, \dot{\underline{q}}) \in \mathbb{R}^{(n \times p) \times (n \times p)}$ is the concatenated matrix of Coriolis and centrifugal torques, $g_{con}(\underline{q}) \in \mathbb{R}^{(n \times p)}$ is the column stack vector of gravitational torques, and $\underline{\tau} \in \mathbb{R}^{(n \times p)}$ is the column stack vector of torques. In this scheme, it is assumed that the topology of \bar{G}_A , \bar{G}_B , and \bar{G}_C are the same as DSC-based semi-decentralized scheme given in Figure 3.2 with $a_{ij} = 15$, $b_{ij} = 30$, and $c_{ij} = 30$ for all $i = 1, \dots, n$ and $j = 0, 1, \dots, n$.

- For the decentralized scheme, consider

$$G = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}$$

which indicates that all followers are only connected to the leader, and do not communicate with other followers. Using this graph, It is assumed that $\bar{G}_A = 15 \times G$, $\bar{G}_B = 30 \times G$, and $\bar{G}_C = 30 \times G$.

- In the semi-decentralized model-dependent coordinated tracking scheme, the network topology of \bar{G}_A , \bar{G}_B , and \bar{G}_C are considered the same as DSC-based semi-decentralized

scheme given in Figure 3.2 but their weights are different as $a_{ij} = 30$, $b_{ij} = 60$, and $c_{ij} = 60$ for $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, n\}$ in case that agent i receive data from j .

The reason that in all cooperative schemes based on model-dependent coordinated tracking algorithm the controller gains, i.e. a_{ij} s, b_{ij} s, and c_{ij} s, are considered high is that this method is not designed for systems with uncertainties. Thus, this is a way to make it more robust.

Figures 3.21 to 3.26 represent the position and orientation trajectories of each agent tracking the desired reference for all aforementioned schemes with their zoom version in the transient time. As displayed in these figures, although the control parameters of all cooperative schemes of model-dependent coordinated tracking algorithm are chosen high, the formation-tracking performances of these schemes are not as accurate as the DSC-based semi-decentralized scheme especially in ψ orientation. Furthermore, in the cooperatives schemes based on model-dependent coordinated tracking algorithm, it takes almost three times more for all agents to reach the desired trajectories of the virtual leader in comparison to the DSC-based semi-decentralized. However, among the cooperatives schemes based on model-dependent coordinated tracking algorithm the centralized scheme has the best performance while the semi-decentralized scheme has the most degraded performance. The degraded performances of the cooperatives schemes based on model-dependent coordinated tracking algorithm are due to their inability to deal with uncertainties in the model of the agents.

Also, the control input signals of the DSC-based semi-decentralized scheme and the cooperatives schemes based on model-dependent coordinated tracking algorithm are represented in Figures 3.27 to 3.29. For all agents, the control efforts for the cooperatives schemes based on model-dependent coordinated tracking algorithm are notably higher than the ones in the DSC-based semi-decentralized scheme.

The quantitative comparison of the results of all schemes of this section are given in Tables 3.12 to 3.14 for the settling time, maximum position error, and maximum absolute control effort of each agent. In addition, for the team level analysis, the root mean squares of the settling time, maximum position error, and maximum absolute control effort, and maximum control effort of each schemes of this section are given in Table 3.11. As shown in this table, the DSC-based semi-decentralized scheme has the lowest RMS settling time and RMS maximum absolute control effort in comparison to all cooperative schemes based on model-dependent coordinated tracking algorithm, but the RMS maximum position error of the DSC-based semi-decentralized scheme is slightly higher than all cooperative schemes based on model-dependent coordinated tracking algorithm. These results indicate that this scheme provides a remarkably better performance by having a faster response with lower control costs.

Table 3.11: Team level RMS analysis of settling time, maximum error, and maximum absolute control effort of DSC-based semi-decentralized and model-dependent coordinated tracking schemes in healthy situation

	RMS <i>t_s (sec)</i>	RMS Max error (N.m)	RMS Max C.E. (N.m)
Semi-decentralized DSC-based	5.0692	4.026	174.67
Semi-decentralized MD-based	31.2902	3.8269	252.19
Decentralized MD-based	28.5209	3.8609	205.92
Centralized MD-based	58.0724	3.3646	389.04

Table 3.12: Quantitative analysis of settling time of the DSC-based semi-decentralized scheme and all cooperative schemes based on model-dependent coordinated tracking algorithm in fault-free situation for all agents

		Settling time (<i>sec</i>)			
		Semi-decentralized DSC-based	Semi-decentralized MD-based	Decentralized MD-based	Centralized MD-based
agent 1	x	3.48	13.65	12.08	4.18
	y	5.33	37.32	27.72	12.9
	ψ	2.78	58.87	58.87	22.3
agent 2	x	3.91	13.1	19.73	6.94
	y	7.52	37.45	36.71	15.52
	ψ	1.63	46.97	65.11	21.72
agent 3	x	3.11	13.19	18.92	8.33
	y	4.46	10.79	20.3	5.62
	ψ	2.71	40.21	42.27	22.98
agent 4	x	7.94	14.67	11.35	8.71
	y	5.62	12.2	13.16	7.48
	ψ	3.83	48.67	12.37	17.34
agent 5	x	7.08	14.23	12.94	10.09
	y	6.27	12.84	14.06	6.4
	ψ	2.92	40.8	14.04	15.77
agent 6	x	6.76	15.15	13.91	8.3
	y	6.29	12.53	14.19	6.44
	ψ	3.22	41.71	17.8	17.84

Table 3.13: Quantitative analysis of maximum error of the DSC-based semi-decentralized scheme and all cooperative schemes based on model-dependent coordinated tracking algorithm in fault-free situation for all agents

		Maximum error (m)			
		Semi-decentralized DSC-based	Semi-decentralized MD-based	Decentralized MD-based	Centralized MD-based
agent 1	x	4.6453	5.7031	5.6637	5.5341
	y	2.9473	2.1644	2.1644	2.1644
	ψ	0.4526	0.7419	0.8218	0.5554
agent 2	x	5.1656	5.5428	5.5811	5.4621
	y	4.3113	4.7438	4.7798	4.7341
	ψ	0.1765	1.1697	1.8688	1.037
agent 3	x	5.634	5.6342	5.6342	5.6342
	y	3.5608	3.5652	3.5643	3.5645
	ψ	0.45191	1.0353	1.512	0.5548
agent 4	x	6.3158	6.3907	6.388	6.4161
	y	4.2771	1.3925	1.3925	1.3925
	ψ	0.9199	1.1105	1.2732	1.0523
agent 5	x	4.877	3.9786	3.9798	4.0055
	y	6.0352	3.2838	3.2836	3.281
	ψ	1.4062	2.0908	2.3369	1.5436
agent 6	x	6.5453	6.5449	6.5449	6.5449
	y	2.5925	2.6076	2.6077	2.6045
	ψ	0.759	1.0465	0.759	0.75904

Table 3.14: Quantitative analysis of maximum absolute control effort of the DSC-based semi-decentralized scheme and all cooperative schemes based on model-dependent coordinated tracking algorithm in fault-free situation for all agents

		Maximum absolute control effort ($N.m$)			
		Semi-decentralized DSC-based	Semi-decentralized MD-based	Decentralized MD-based	Centralized MD-based
agent 1	x	121.72	268.53	230.19	511.92
	y	247.46	233.57	392.65	589.36
	ψ	9.3175	25.786	20.188	46.513
agent 2	x	227.39	464.84	230.04	547.24
	y	393.48	581.77	436.48	821.67
	ψ	6.6804	17.573	5.9534	55.221
agent 3	x	433.43	438.44	223.44	522.39
	y	235.24	438.78	376.9	587.57
	ψ	8.8328	34.579	21.113	49.899
agent 4	x	85.033	83.409	69.773	144.94
	y	33.316	153.01	173.1	274.76
	ψ	24.382	11.259	15.26	25.913
agent 5	x	49.698	54.759	71.414	157.38
	y	71.473	97.196	193.85	367.35
	ψ	26.492	23.995	23.214	81.115
agent 6	x	47.813	105.13	75.799	235.76
	y	60.399	128.88	188.12	435.72
	ψ	5.8925	74.215	19.82	74.414

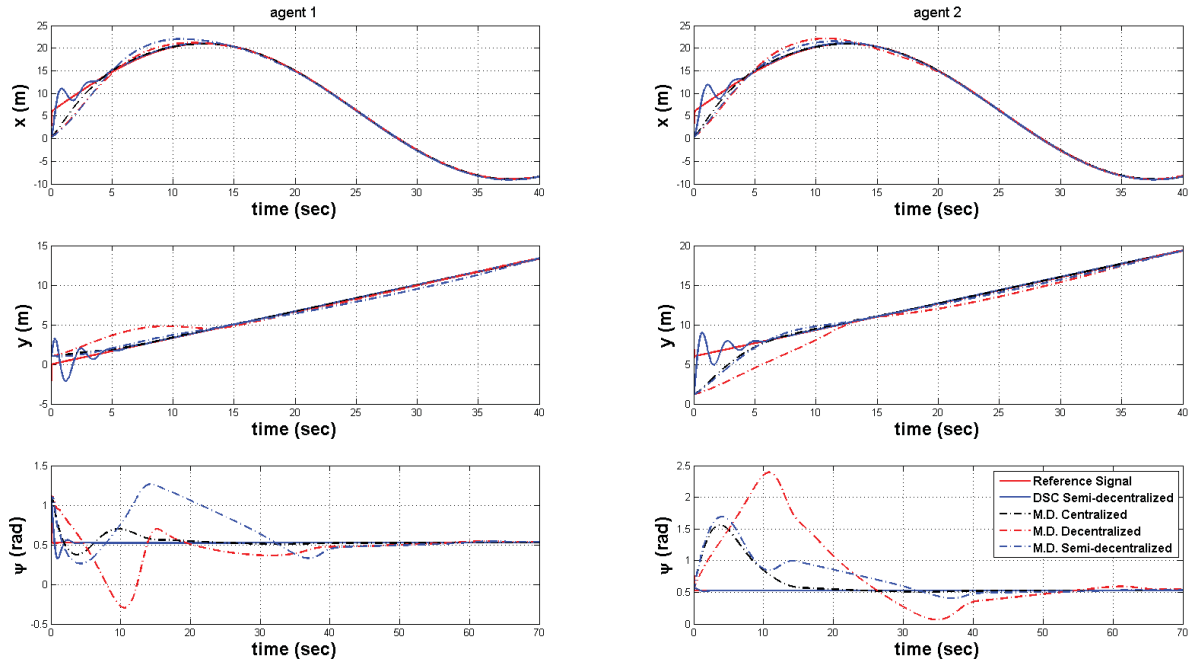


Figure 3.21: Position trajectories of agents #1 and #2 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.

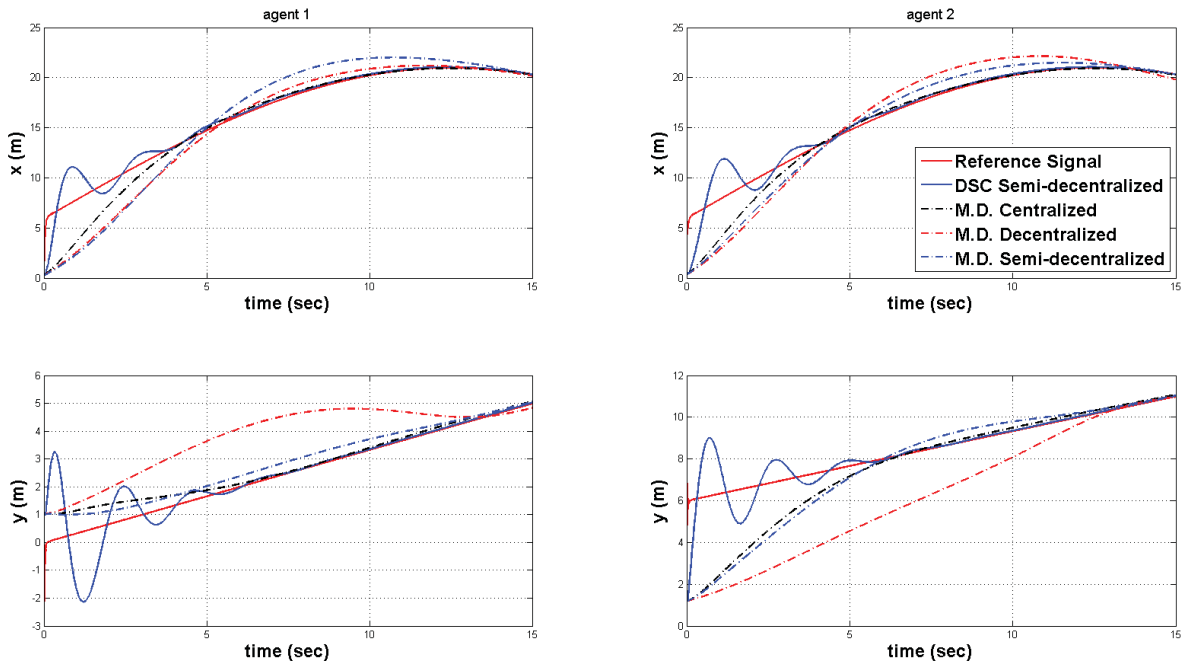


Figure 3.22: Zoomed version of position trajectories of agents #1 and #2 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated schemes.

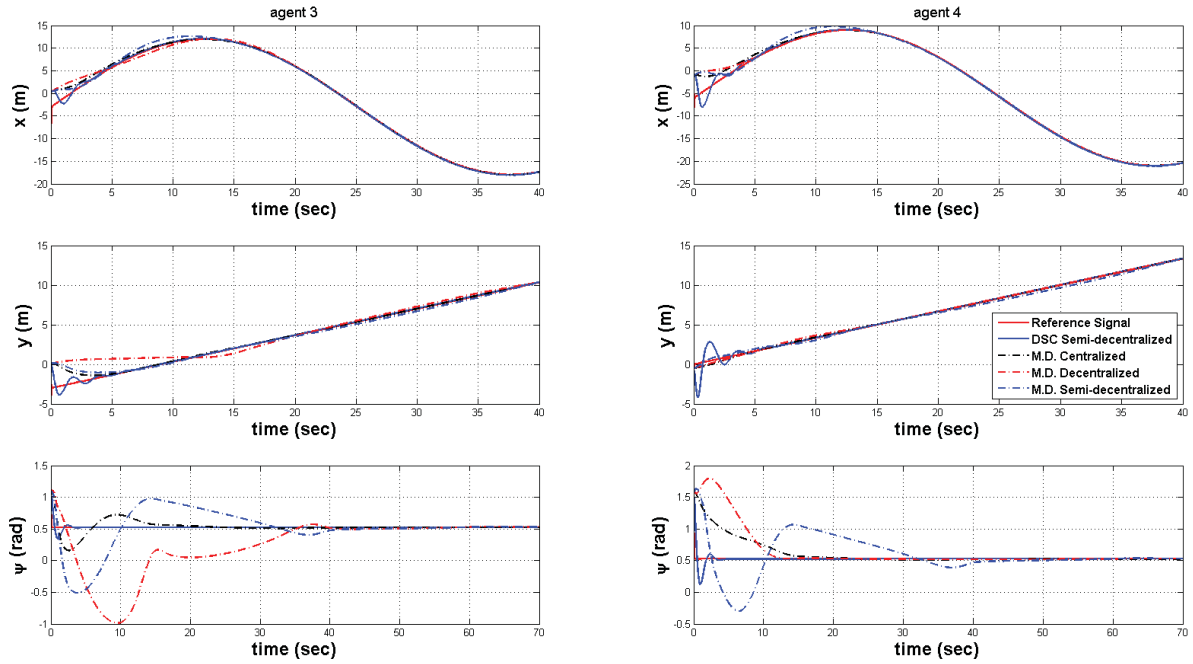


Figure 3.23: Position trajectories of agents #3 and #4 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.

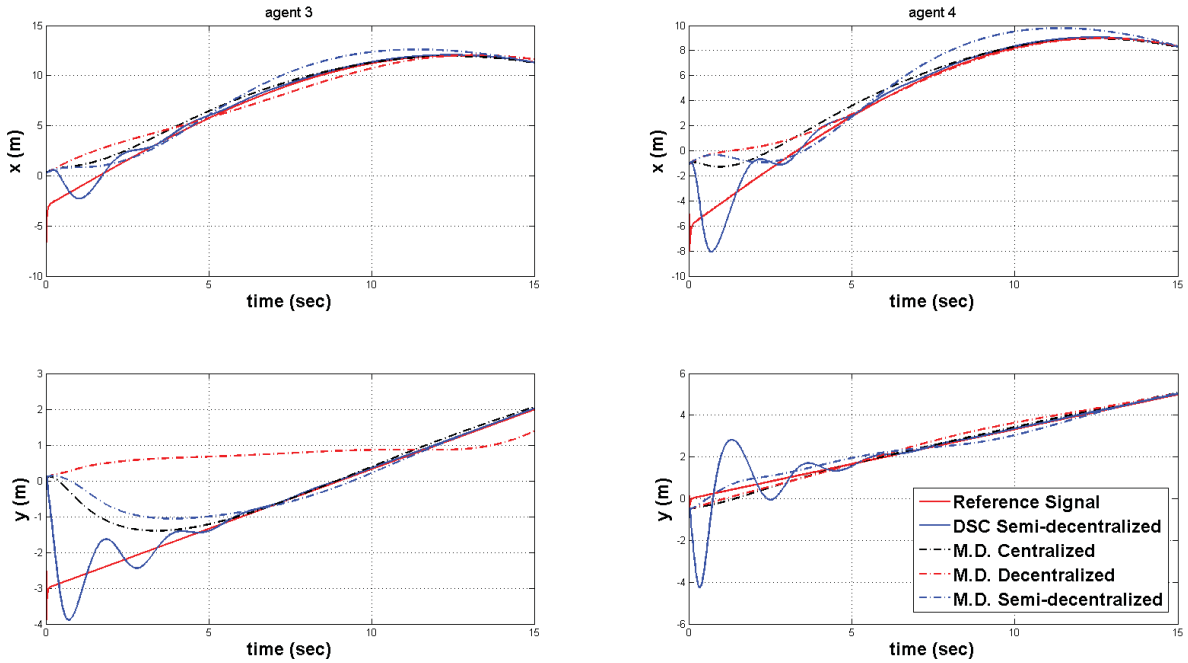


Figure 3.24: Zoomed version of position trajectories of agents #3 and #4 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated schemes.

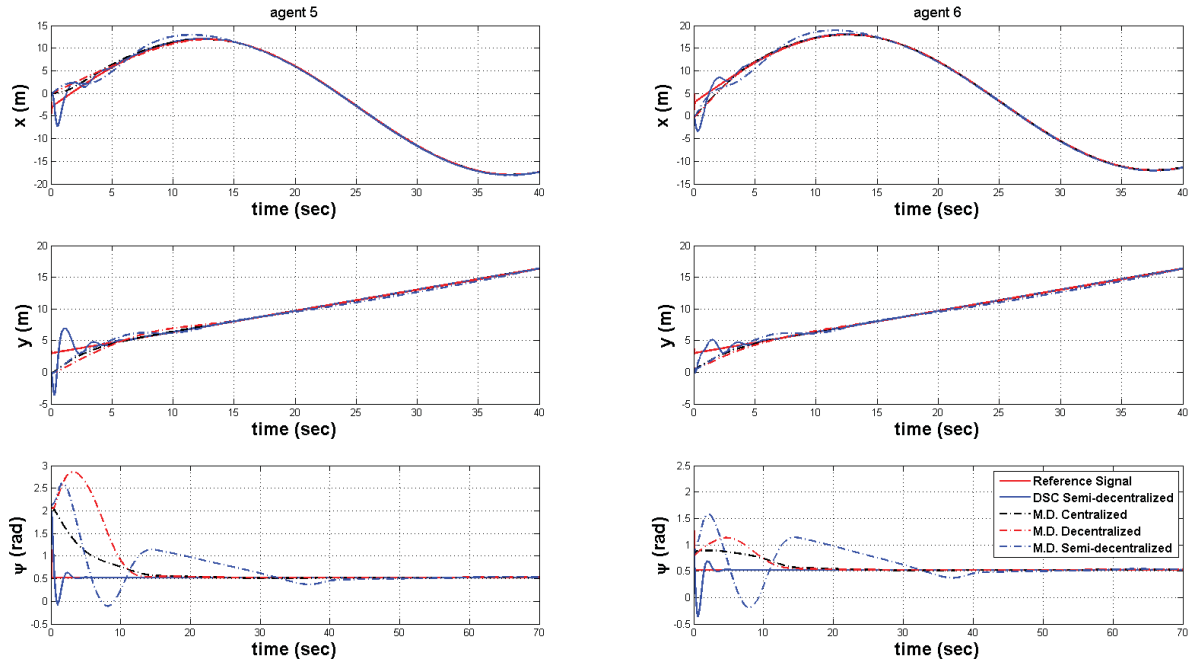


Figure 3.25: Position trajectories of agents #5 and #6 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.

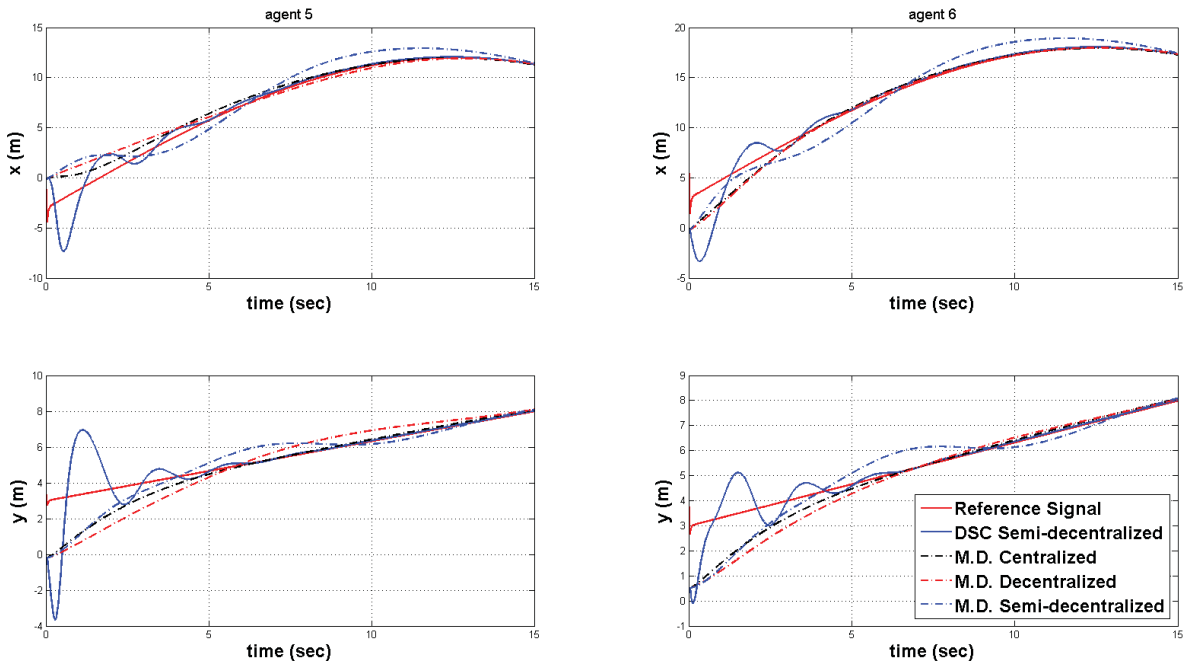


Figure 3.26: Zoomed version of position trajectories of agents #5 and #6 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated schemes.

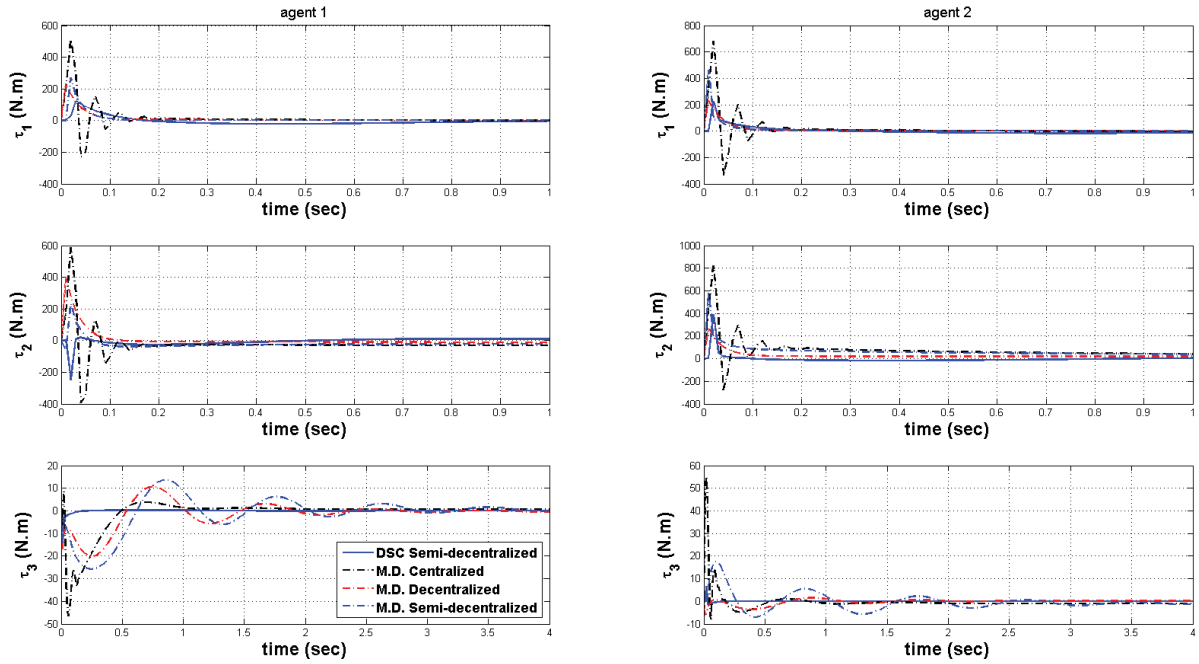


Figure 3.27: Control input signals of agents #1 and #2 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.

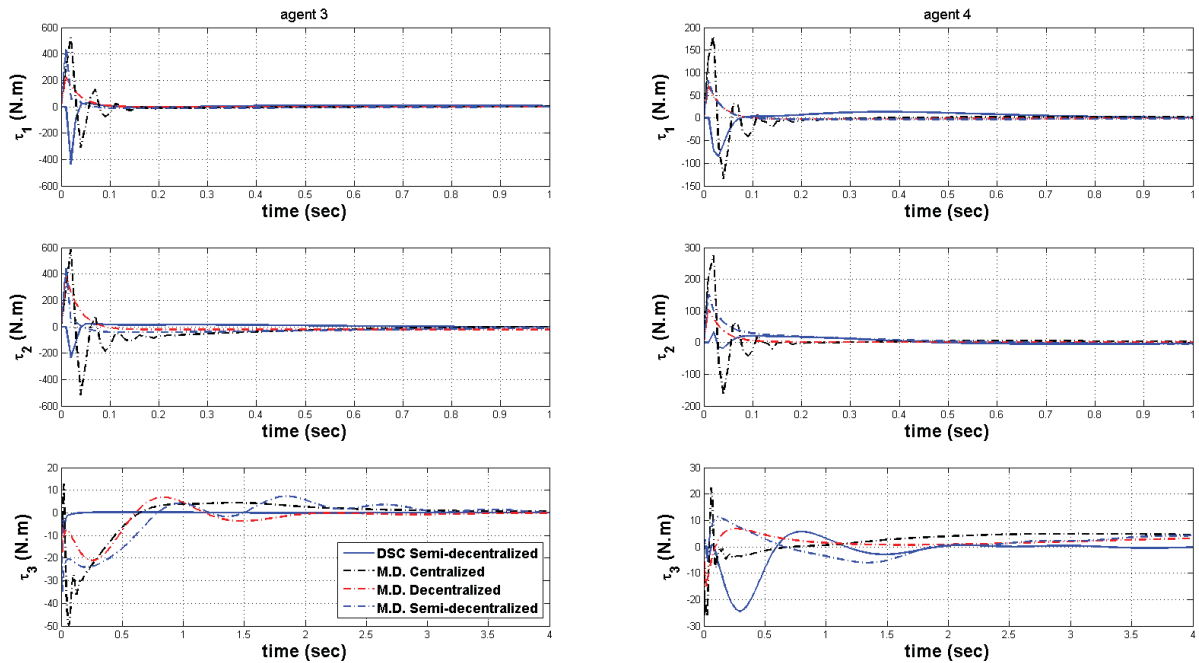


Figure 3.28: Control input signals of agents #3 and #4 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.

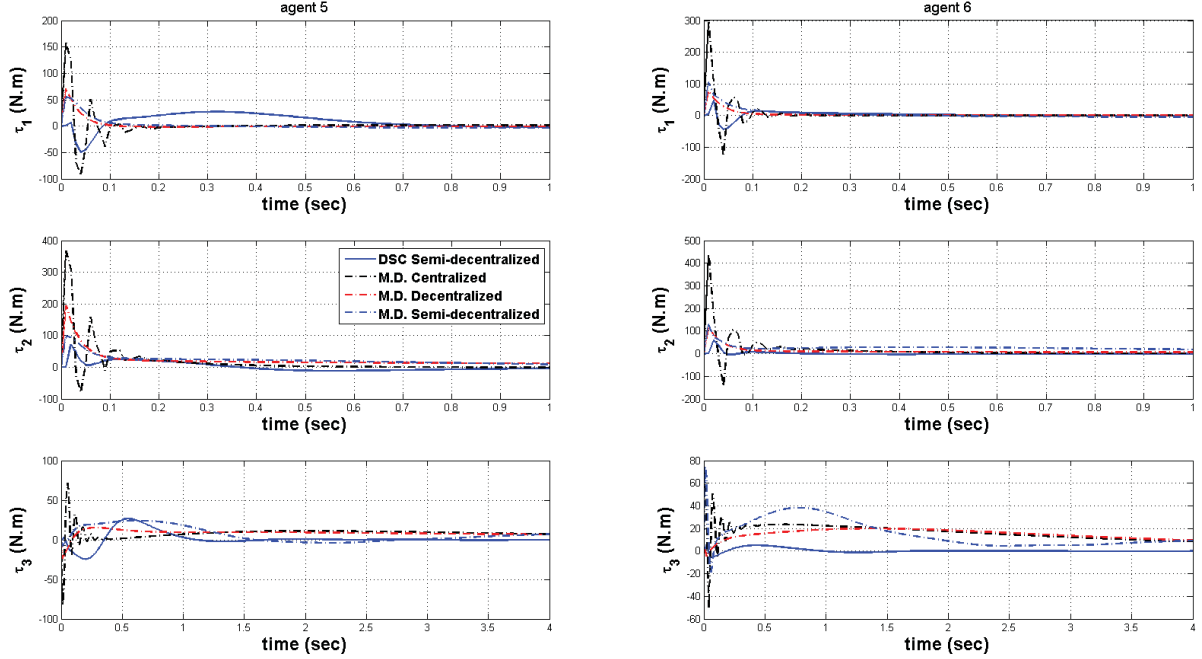


Figure 3.29: Control input signals of agents #5 and #6 for semi-decentralized DSC-based and centralized, decentralized, and semi-centralized model-dependent coordinated tracking schemes.

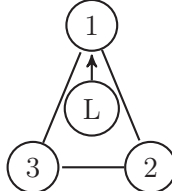
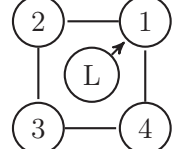
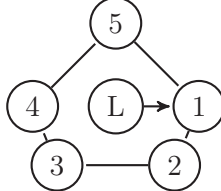
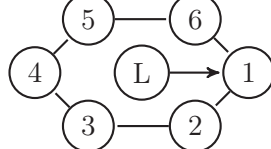
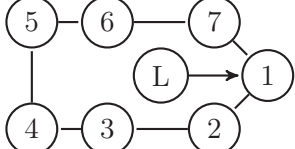
3.7.4 Simulation Scenarios for Semi-decentralized DSC-based Scheme

In this section, to investigate the ability of the proposed semi-decentralized DSC-based scheme in various situations and conditions, four different scenarios are simulated considering several conditions and possibilities in a group of multiple AUVs.

3.7.4.1 Scenario 3.1: Different Group Size

In this scenario, we aim to analyze the ability of the proposed semi-decentralized DSC-based scheme to fulfill the objectives in the different network topologies with different number of vehicles involved in the group. One feature that all the considered networks in this scenario have in common is that their connection topologies are a ring and only one of the agents is directly connected to the virtual leader. To this end, five cases are defined as in Table 3.15.

Table 3.15: Different number of agents in the group for scenario 3.1

Case	Number of agents	Label of graph	Graph
1	3	G_1	
2	4	G_2	
3	5	G_3	
4	6	G_4	
5	7	G_5	

For the simulation performed in this section, desired relative distances, model parameters and uncertainties, and controller parameters are considered the same as Section 3.7, but the initial conditions of all agents are set to zero. Also, the reference trajectory is considered as

$$\eta_{ref} = \left[15 \sin\left(\frac{t}{8}\right) \left(\frac{t}{3}\right) 0 \right]^T$$

For each case, in Figures 3.30 to 3.34, the error signals of x , y , and ψ states for three randomly chosen agents are presented for both semi-decentralized and centralized schemes. In this section, the centralized scheme is used as a benchmark to evaluate the results of the proposed semi-decentralized scheme.

Also, the maximum and steady state errors of all agents in each case for both semi-decentralized and centralized schemes are tabulated in Tables 3.17 to 3.21.

Based on these results it can be seen that the maximum errors of agents in the semi-decentralized scheme for all cases are higher than the ones in the centralized scheme. However, these differences are minor. From the analysis of steady state error, it can be seen that the e_{ss} in surge and sway are lower in the centralized scheme while in yaw, it is lower in the semi-decentralized scheme, no matter how many agents are involved in the mission. Based on these results, it can be concluded that the proposed semi-decentralized DSC-based scheme has significant performance independent of the numbers of agents in the network.

Table 3.16: Team level RMS analysis of maximum error and steady state error for semi-decentralized and centralized schemes in fault-free situation for scenario 3.1

	RMS maximum error (m)		RMS steady state error (m)	
	Semi-decentralized	Centralized	Semi-decentralized	Centralized
Case 1	3.270822	3.086608	0.153036	0.136372
Case 2	3.581606	3.275826	0.153113	0.136424
Case 3	3.317131	2.925715	0.153276	0.136481
Case 4	2.926747	2.411658	0.153398	0.074393
Case 5	3.339930	3.312630	0.150701	0.150435

The team level analysis of the semi-decentralized and centralized schemes are presented in Table 3.16. For team level analysis, the root mean square of maximum error and steady state error are obtained for each scheme. In this table, the RMS maximum error and RMS steady

state error of the centralized scheme are slightly lower than the semi-decentralized scheme.

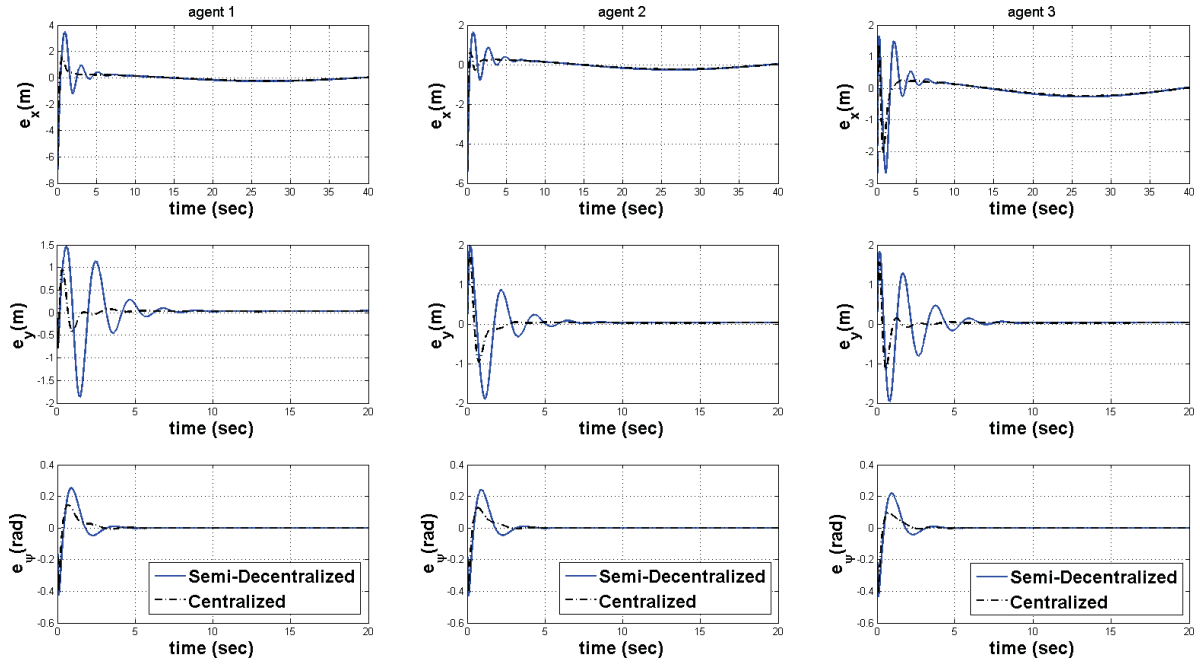


Figure 3.30: Error signals of agents #1, #2, and #3 in fault-free situation for scenario 3.1, case 1.

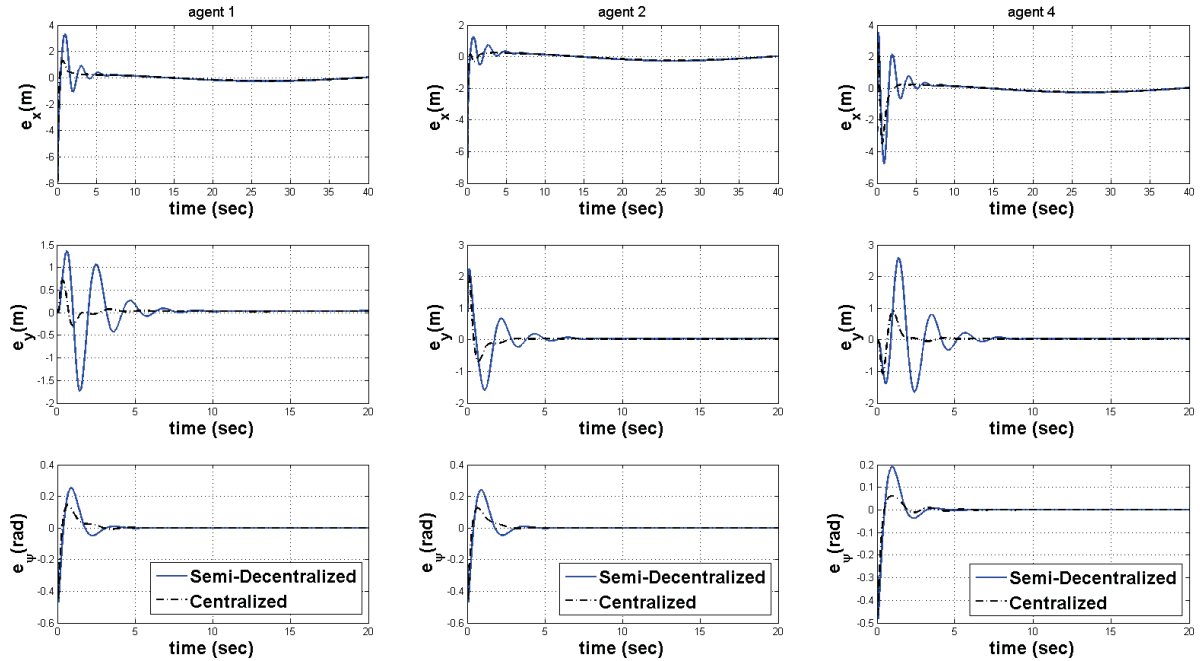


Figure 3.31: Error signals of agents #1, #2, and #4 in fault-free situation for scenario 3.1, case 2.

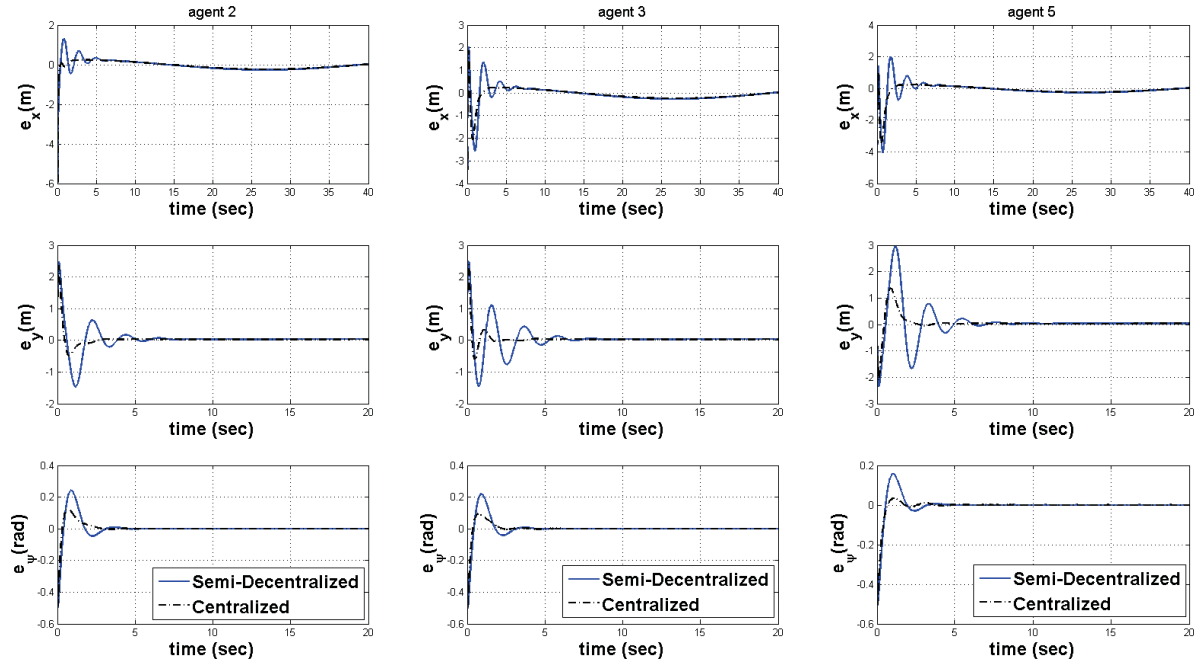


Figure 3.32: Error signals of agents #2, #3, and #5 in fault-free situation for scenario 3.1, case 3.

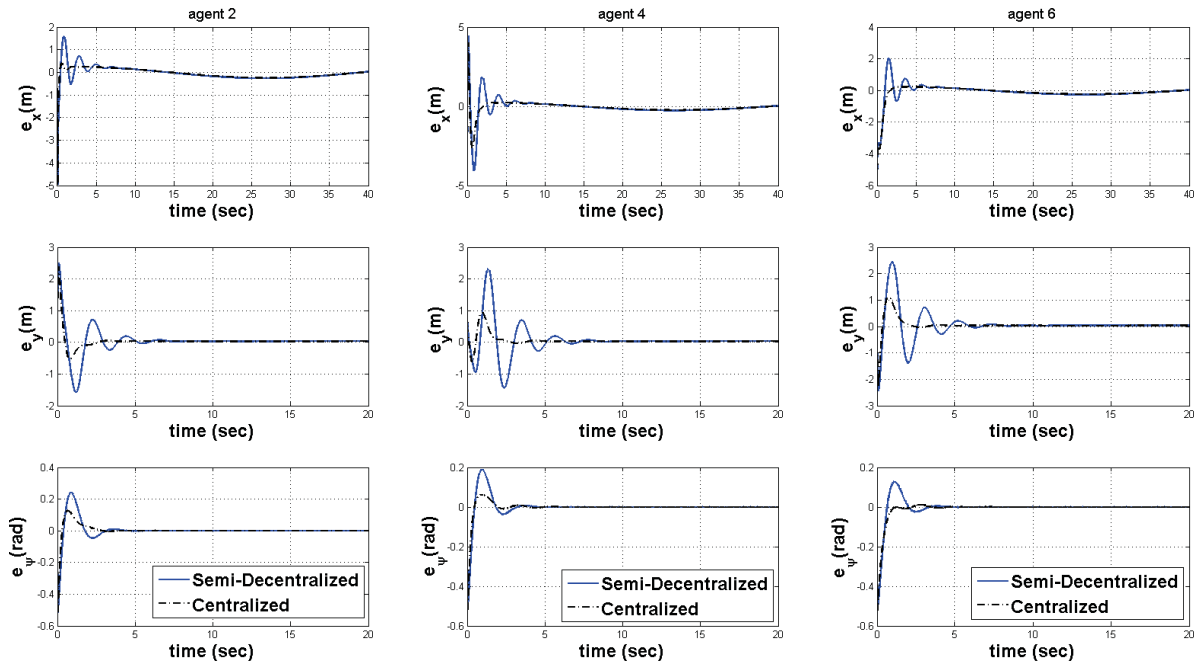


Figure 3.33: Error signals of agents #2, #4, and #6 in fault-free situation for scenario 3.1, case 4.

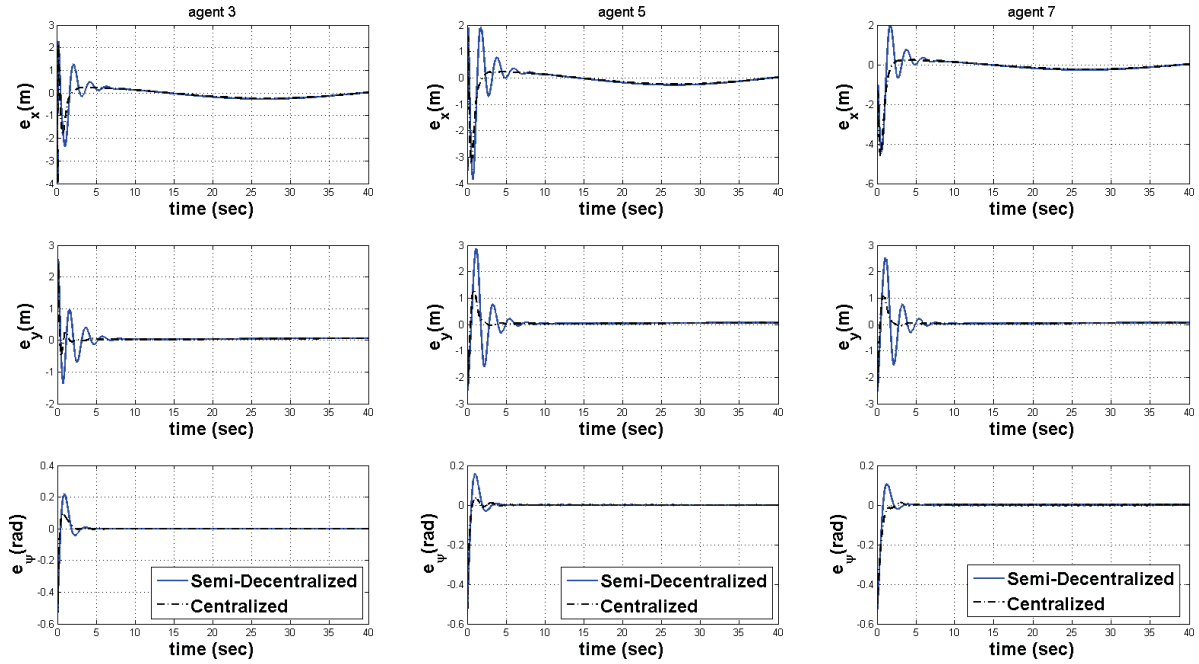


Figure 3.34: Error signals of agents #3, #5, and #7 in fault-free situation for scenario 3.1, case 5.

Table 3.17: Response characteristics of the semi-decentralized and centralized schemes for scenario

3.1, case 1

			Maximum error (m)		Steady state error (m)	
			Semi-decentralized	Centralized	Semi-decentralized	Centralized
Case 1	agent 1	x	6.936	6.624	0.2633	0.2345
		y	1.87	0.9525	0.03027	0.02803
		ψ	0.4239	0.3899	2.2×10^{-4}	2.241×10^{-4}
	agent 2	x	5.401	5.273	0.2634	0.2347
		y	1.98	1.764	0.03031	0.02804
		ψ	0.4281	0.3952	2.204×10^{-4}	2.214×10^{-4}
	agent 3	x	2.677	2.672	0.2633	0.2344
		y	1.967	1.559	0.03019	0.02805
		ψ	0.4358	0.4065	6.52×10^{-4}	6.55×10^{-4}

Table 3.18: Response characteristics of the semi-decentralized and centralized schemes for scenario

3.1, case 2

			Maximum error (m)		Steady state error (m)	
			Semi-decentralized	Centralized	Semi-decentralized	Centralized
Case 2	agent 1	x	7.849	7.753	0.2634	0.2346
		y	1.742	0.7237	0.03103	0.02802
		ψ	0.4665	0.4458	9.97×10^{-9}	6.54×10^{-6}
	agent 2	x	6.405	6.344	0.2634	0.2346
		y	2.24	2.054	0.03105	0.02804
		ψ	0.4695	0.4492	9.8×10^{-9}	5.42×10^{-6}
	agent 3	x	2.815	2.3	0.2633	0.2346
		y	2.22	2.024	0.03081	0.02805
		ψ	0.4745	0.4565	9.51×10^{-9}	1.57×10^{-6}
	agent 4	x	4.774	3.507	0.2635	0.2347
		y	2.591	1.082	0.03052	0.02805
		ψ	0.4813	0.4574	9.08×10^{-9}	1.51×10^{-6}

Table 3.19: Response characteristics of the semi-decentralized and centralized schemes for scenario

3.1, case 3

			Maximum error (m)		Steady state error (m)	
			Semi-decentralized	Centralized	Semi-decentralized	Centralized
Case 3	agent 1	x	7.34	6.76	0.2642	0.2347
		y	1.819	0.806	0.03155	0.02805
		ψ	0.4955	0.4855	1.85×10^{-4}	1.95×10^{-4}
	agent 2	x	5.928	5.928	0.264	0.2346
		y	2.476	2.356	0.03154	0.02806
		ψ	0.4968	0.4878	1.405×10^{-6}	6.97×10^{-6}
	agent 3	x	2.574	2.063	0.2635	0.2347
		y	2.497	2.388	0.03128	0.02808
		ψ	0.4991	0.4914	4.06×10^{-4}	4.13×10^{-4}
	agent 4	x	4.564	3.475	0.2635	0.2348
		y	2.517	1.048	0.03096	0.02809
		ψ	0.5023	0.4963	1.37×10^{-6}	1.59×10^{-5}
	agent 5	x	4.074	3.538	0.263	0.2348
		y	2.953	2.12	0.03075	0.02808
		ψ	0.507	0.5023	1.96×10^{-4}	2.12×10^{-4}

Table 3.20: Response characteristics of the semi-decentralized and centralized schemes for scenario

3.1, case 4

			Maximum error (m)		Steady state error (m)	
			Semi-decentralized	Centralized	Semi-decentralized	Centralized
Case 4	agent 1	x	6.406	5.4102	0.2635	0.1259
		y	1.945	0.9316	0.0315	0.0295
		ψ	0.5132	0.5101	1.03×10^{-3}	8.73×10^{-4}
	agent 2	x	4.935	3.078	0.2636	0.1259
		y	2.505	2.378	0.0391	0.0293
		ψ	0.5138	0.5117	5.28×10^{-4}	6.9×10^{-6}
	agent 3	x	2.646	2.098	0.2636	0.1257
		y	2.527	2.411	0.0367	0.0288
		ψ	0.514	0.5136	9.25×10^{-9}	9.8×10^{-5}
	agent 4	x	4.31	4.002	0.2635	0.1254
		y	2.315	0.6327	0.03093	0.0284
		ψ	0.516	0.51	5.28×10^{-4}	5.57×10^{-4}
	agent 5	x	3.5	2.839	0.2638	0.1252
		y	2.764	2.232	0.03017	0.0288
		ψ	0.5186	0.516	1.36×10^{-3}	5.18×10^{-4}
	agent 6	x	3.732	3.631	0.2636	0.1252
		y	2.449	2.232	0.03018	0.0291
		ψ	0.5215	0.5214	1.5×10^{-3}	4.39×10^{-4}

Table 3.21: Response characteristics of the semi-decentralized and centralized schemes for scenario

3.1, case 5

			Maximum error (m)		Steady state error (m)	
			Semi-decentralized	Centralized	Semi-decentralized	Centralized
Case 5	agent 1	x	6.929	6.929	0.263	0.234
		y	1.929	0.928	0.0313	0.027
		ψ	0.524	0.524	1.873×10^{-4}	1.95×10^{-4}
	agent 2	x	5.968	5.968	0.263	0.234
		y	2.545	2.421	0.0315	0.02801
		ψ	0.524	5.178	1.876×10^{-4}	1.963×10^{-4}
	agent 3	x	3.949	3.949	0.263	0.234
		y	2.567	2.45	0.0312	0.02802
		ψ	0.523	0.418	1.84×10^{-4}	1.94×10^{-4}
	agent 4	x	4.574	4.176	0.2635	0.2374
		y	2.39	0.9873	0.0309	0.02804
		ψ	0.523	0.523	2.03×10^{-4}	2.37×10^{-4}
	agent 5	x	3.848	3.497	0.2636	0.2348
		y	2.86	2.32	0.0307	0.02804
		ψ	0.5225	0.5219	6.41×10^{-4}	6.52×10^{-4}
	agent 6	x	5.69	5.69	0.233	0.2362
		y	2.52	2.328	0.0306	0.0803
		ψ	0.5235	0.512	2.15×10^{-4}	2.21×10^{-4}
	agent 7	x	4.645	4.386	0.2636	0.234
		y	2.531	2.35	0.0308	0.2804
		ψ	0.525	0.52	2.18×10^{-4}	2.19×10^{-4}

3.7.4.2 Scenario 3.2: Different Network Topologies

In this section, to evaluate the effect of network topologies on the proposed semi-decentralized DSC-based control scheme, for a group of six autonomous underwater vehicles, three different networks have been considered as in Table 3.22. In the first case, a simple ring topology is defined in which there is only one agent directly connected to the virtual leader. In the second case, there are two agents directly connected to the virtual leader, namely the first layer followers, and they are connected to all other agents, namely the second layer followers which do not communicate with each other. And finally, the third case is the same as the second one with this difference that the second layer followers receive data from all agents no matter they are first or second layer agents.

For the simulation performed in this section, desired relative distances, model parameters and uncertainties, and controller parameters are considered the same as Section 3.7. In this part, the initial conditions of all agents are set to zero and the reference trajectory is considered as

$$\eta_{ref} = \left[15 \sin\left(\frac{t}{8}\right) \left(\frac{t}{3}\right) \left(\frac{\pi}{6}\right) \right]^T$$

The error trajectories of transient time of all agents for all cases plus the centralized scheme are represented in Figures 3.35, 3.36, and 3.37. In addition, in Tables 3.24 and 3.25, the maximum errors and steady state errors of all agents in all cases are compared to the ones from the centralized scheme which are the best possible response of the problem.

Based on these simulation results, for agent #1 which is a first layer follower in all three cases, the communication topology does not have a significant effect on its performance. However, for other agents which are second layer followers, it can be seen that for the first case the error converges to zero slower than second and third cases. However, the maximum errors in y

and ψ states are lower in the first case than the other two cases. In x position the maximum errors vary from agent to agent which is the effect of desired position of each agent in the given formation. On the other hand, for all agents, the maximum errors and steady state errors of all agents from the centralized scheme are slightly lower in comparison to all three cases of the semi-decentralized scheme. However, these differences are minor.

Table 3.22: Different network topology for scenario 3.2

Case	Graph
1	
2	
3	

To summarize, in the first case where the communication is minimum, the entire group has a close performance to other cases with more communication among agents, and this is one of the

advantages of the proposed semi-decentralized DSC-based control scheme since for underwater vehicles it is better to keep the communication among agents as minimum as possible because of the environmental situations which may cause communication loss among agents.

Table 3.23: Team level RMS analysis of maximum error and steady state error for semi-decentralized and centralized schemes in fault-free situation for scenario 3.2

	Semi-decentralized scheme			Centralized scheme
	Case 1	Case 2	Case 3	
RMS maximum error (m)	3.0061	3.3871	3.5465	2.411
RMS steady state error (m)	0.131866	0.133067	0.136544	0.128204

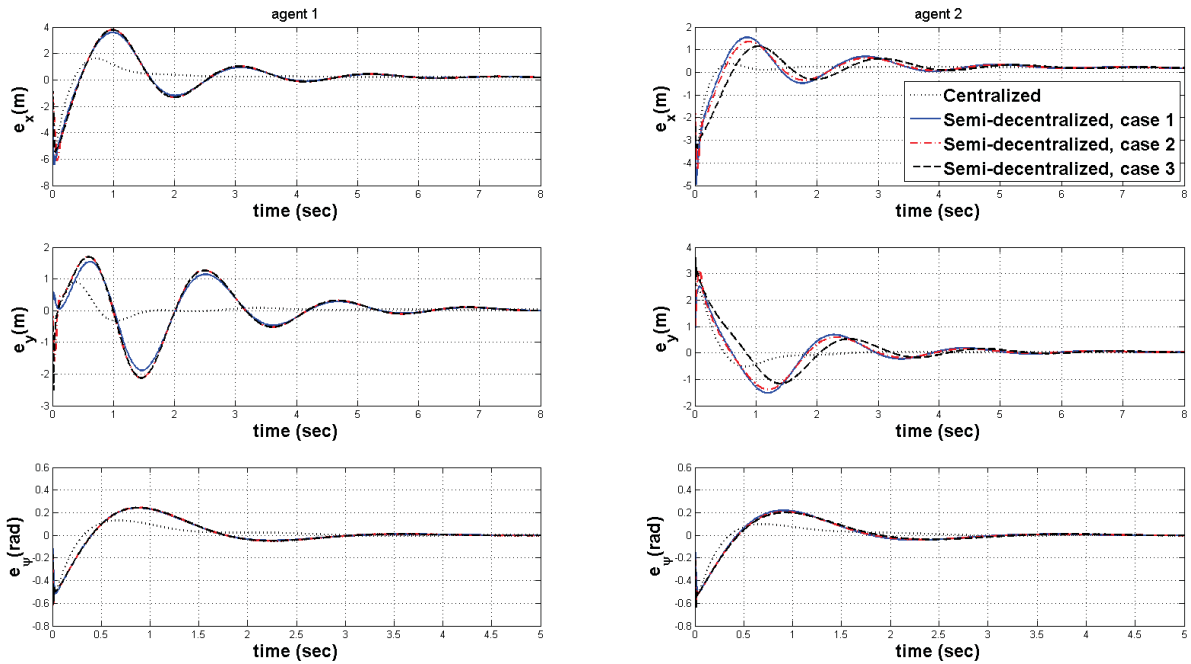


Figure 3.35: Error signals of agent #1 and #2 in fault-free situation for the centralized scheme and three semi-decentralized cases of scenario 3.2.

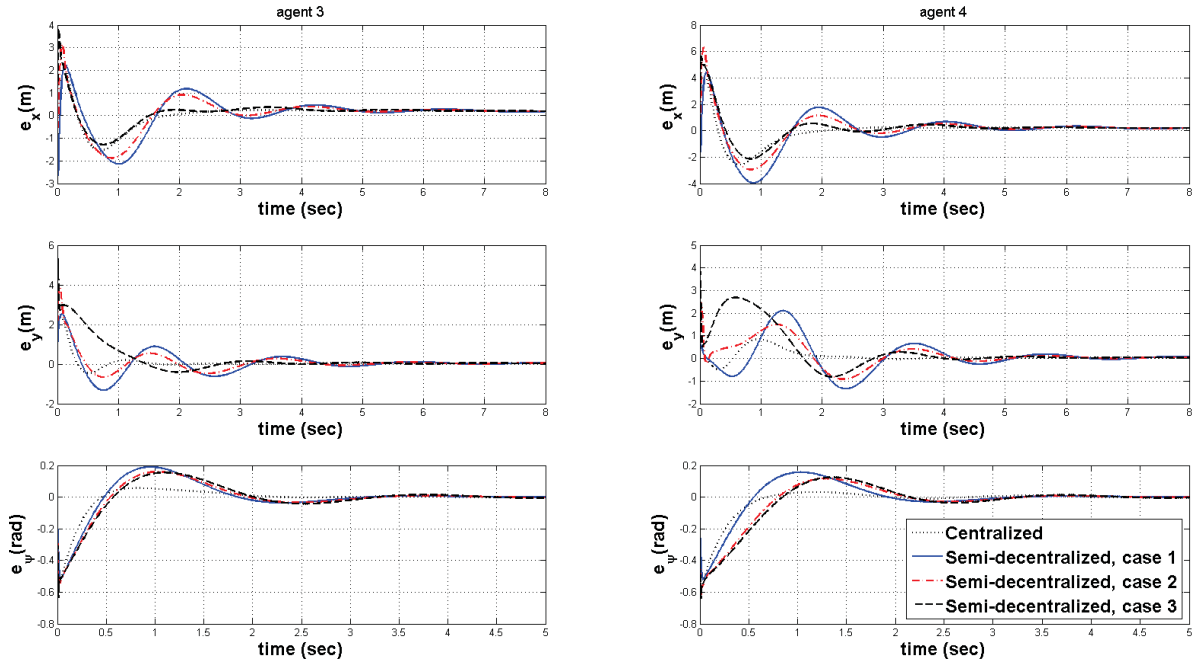


Figure 3.36: Error signals of agent #3 and #4 in fault-free situation for the centralized scheme and three semi-decentralized cases of scenario 3.2.

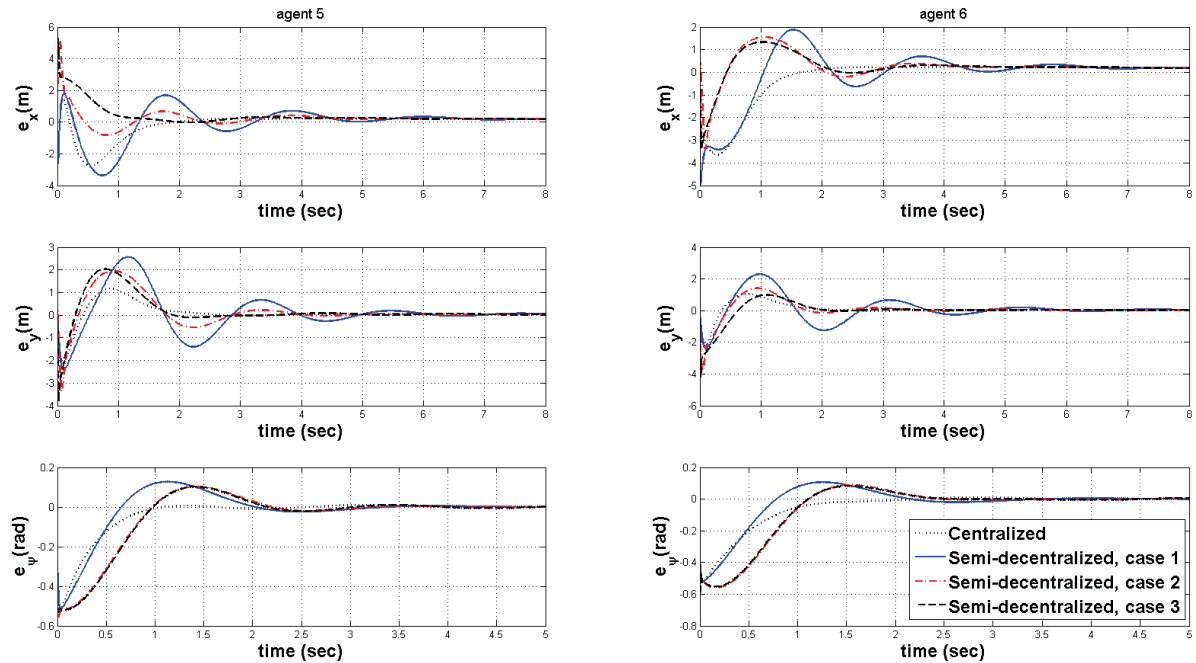


Figure 3.37: Error signals of agent #5 and #6 in fault-free situation for the centralized scheme and three semi-decentralized cases of scenario 3.2.

Table 3.24: Maximum errors of all cases of the semi-decentralized and centralized schemes for scenario

3.2

		Maximum error (m)			
		Semi-decentralized scheme			Centralized scheme
		Case 1	Case 2	Case 3	
agent 1	x	6.4064	6.0994	5.4352	5.4102
	y	1.8902	2.1365	2.5095	0.9316
	ψ	0.5169	0.6124	0.6108	0.5101
agent 2	x	4.9352	4.2352	3.3395	3.078
	y	2.5054	3.1359	3.6139	2.378
	ψ	0.5178	0.613	0.6381	0.5117
agent 3	x	2.6463	3.1473	3.812	2.098
	y	2.5267	3.9358	5.3095	2.411
	ψ	0.519	0.6046	0.6361	0.5136
agent 4	x	4.4313	6.3223	5.6556	4.002
	y	2.1059	2.4827	4.1525	0.6327
	ψ	0.5203	0.6186	0.6371	0.51
agent 5	x	3.3716	5.0609	5.3251	2.839
	y	2.5631	3.3902	3.7813	2.232
	ψ	0.5218	0.5561	0.5478	0.516
agent 6	x	4.9492	3.3885	3.4037	3.631
	y	2.4291	3.8173	4.2188	2.232
	ψ	0.5233	0.6883	0.6846	0.5214

Table 3.25: Steady state errors of all cases of the semi-decentralized and centralized schemes for scenario 3.2

		Steady state error (m)			
		Semi-decentralized scheme			Centralized scheme
		Case 1	Case 2	Case 3	
agent 1	x	0.132	0.1328	0.1329	0.1259
	y	0.0221	0.0224	0.0231	0.0295
	ψ	5.1×10^{-6}	1.5×10^{-3}	1.6×10^{-3}	8.73×10^{-4}
agent 2	x	0.2401	0.243	0.2434	0.2347
	y	0.0291	0.0293	0.0291	0.02778
	ψ	1.25×10^{-4}	9×10^{-4}	10^{-3}	5.43×10^{-4}
agent 3	x	0.237	0.2395	0.2398	0.2347
	y	0.0277	0.0337	0.0338	0.02276
	ψ	2.97×10^{-4}	10^{-4}	4×10^{-4}	4.95×10^{-6}
agent 4	x	0.239	0.2392	0.2397	0.2347
	y	0.03	0.0349	0.03495	0.02831
	ψ	4.51×10^{-3}	9×10^{-4}	4×10^{-4}	5.33×10^{-4}
agent 5	x	0.2444	0.2476	0.2439	0.2348
	y	0.0319	0.0396	0.0282	0.02766
	ψ	3.5×10^{-3}	1.7×10^{-3}	1.2×10^{-3}	5.18×10^{-4}
agent 6	x	0.2445	0.2445	0.2449	0.2348
	y	0.031	0.0322	0.0324	0.02757
	ψ	2.3×10^{-3}	2.1×10^{-3}	2×10^{-3}	4.39×10^{-4}

3.7.4.3 Scenario 3.3: Different Reference Trajectories

To analyze the performance of the proposed semi-decentralized DSC-based control scheme for various reference trajectories, three different reference trajectories have been considered as in Table 3.26. For the simulations conducted in this scenario, for a group of six autonomous underwater vehicles as in Figure 3.2, desired relative distances, initial conditions, model parameters and uncertainties, and controller parameters are considered the same as Section 3.7.

The error signals of all agents in the semi-decentralized and centralized schemes for all cases defined in Table 3.26 are given in Figures 3.38 to 3.43. Furthermore, the quantitative comparison of maximum errors and steady state errors for the semi-decentralized and centralized schemes in each case are given in Tables 3.27 and 3.28 respectively. Based on obtained results, independent of the desired reference trajectories, the error trajectories of the semi-decentralized scheme are very closed to the ones from the centralized scheme. However, for all agents in each case, the maximum errors and steady state errors are slightly lower in the centralized scheme in comparison to the semi-decentralized scheme.

Table 3.26: Different reference trajectories for scenario 3.3

Case	Reference
1	$\eta_{ref} = \left[15 \sin\left(\frac{t}{8}\right) \quad \frac{t}{3} \quad 0 \right]^T$
2	$\eta_{ref} = \left[2t \quad \sin\left(\frac{t}{5}\right) + 2 \quad 0.5 \right]^T$
3	$\eta_{ref} = \left[\cos\left(\frac{t}{2}\right) \quad \left(\frac{t}{2}\right) \quad 0.5 \cos\left(\frac{t}{10}\right) \right]^T$

The team level comparison of these two schemes for each case are presented in Table 3.29 by giving the root mean square of the maximum errors and steady state errors. In this table,

the results explained above are confirmed.

Table 3.27: Maximum errors of all cases of the semi-decentralized and centralized schemes for scenario

3.3

		Maximum error (m)					
		Semi-decentralized			Centralized		
		Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
agent 1	x	5.4973	5.9247	6.2606	5.2213	5.7895	5.977
	y	3.358	3.9028	4.3883	3.2918	3.6951	4.3867
	ψ	1.0163	0.622	0.622	0.9352	0.5821	0.5823
agent 2	x	2.7953	2.7953	3.6519	2.7937	2.7937	3.6504
	y	3.6434	2.4622	3.7871	3.2876	2.4003	3.524
	ψ	0.579	0.1408	0.1408	0.5412	0.1248	0.1249
agent 3	x	3.481	3.3895	2.6893	3.4852	3.3412	2.6607
	y	2.4164	0.8903	2.3568	2.3664	0.7331	2.3568
	ψ	0.9533	0.5291	0.5291	0.9069	0.5026	0.5027
agent 4	x	4.7675	4.6639	3.8652	4.7129	4.5634	3.9154
	y	2.5045	4.4422	2.7871	2.7368	4.6183	2.6722
	ψ	1.3887	0.9508	0.9508	1.3158	0.8959	0.8959
agent 5	x	4.2765	6.4937	5.5003	4.2509	6.0377	5.465
	y	5.3705	7.552	5.5251	5.6417	7.9106	5.8686
	ψ	1.8687	1.4125	1.4125	1.7711	1.3239	1.3239
agent 6	x	4.3663	4.6074	5.2639	4.3648	4.5924	5.0247
	y	2.7335	4.6378	2.8196	2.6019	4.4986	2.6839
	ψ	0.781	0.2977	0.2978	0.7375	0.2554	0.2554

Table 3.28: Steady state errors of all cases of the semi-decentralized and centralized schemes for scenario 3.3

		Steady state error (m)					
		Semi-decentralized			Centralized		
		Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
agent 1	x	0.2641	0.2816	0.06	0.2347	0.2537	0.0596
	y	0.0475	0.0265	0.0104	0.0427	0.0229	0.0082
	ψ	2×10^{-7}	4.3×10^{-4}	6.8×10^{-3}	5.9×10^{-8}	4.2×10^{-4}	6×10^{-3}
agent 2	x	0.2641	0.2813	0.058	0.2347	0.2535	0.0581
	y	0.0474	0.0262	0.0117	0.02778	0.0227	0.0095
	ψ	9.2×10^{-8}	4.6×10^{-4}	7.2×10^{-3}	4.4×10^{-9}	4.5×10^{-4}	6.4×10^{-3}
agent 3	x	0.2642	0.2812	0.0587	0.2347	0.2534	0.0585
	y	0.047	0.0266	0.0112	0.02276	0.023	0.0091
	ψ	1.8×10^{-8}	2.4×10^{-5}	6.6×10^{-3}	9.1×10^{-9}	2.3×10^{-5}	5.8×10^{-3}
agent 4	x	0.2642	0.2813	0.0585	0.2347	0.2535	0.0583
	y	0.0472	0.0267	0.0108	0.02831	0.023	0.0087
	ψ	2.8×10^{-8}	4.1×10^{-4}	6.3×10^{-3}	4.7×10^{-9}	4×10^{-4}	5.5×10^{-3}
agent 5	x	0.2641	0.2815	0.0583	0.2348	0.2536	0.058
	y	0.0473	0.0268	0.011	0.02766	0.023	0.0089
	ψ	3.7×10^{-8}	8.4×10^{-4}	6.1×10^{-3}	1.3×10^{-9}	8.3×10^{-4}	5.4×10^{-3}
agent 6	x	0.2641	0.2815	0.0588	0.2348	0.2536	0.0586
	y	0.0473	0.0268	0.0112	0.02757	0.0231	0.009
	ψ	1.1×10^{-7}	4.2×10^{-4}	6.6×10^{-3}	7.7×10^{-9}	4×10^{-4}	5.9×10^{-3}

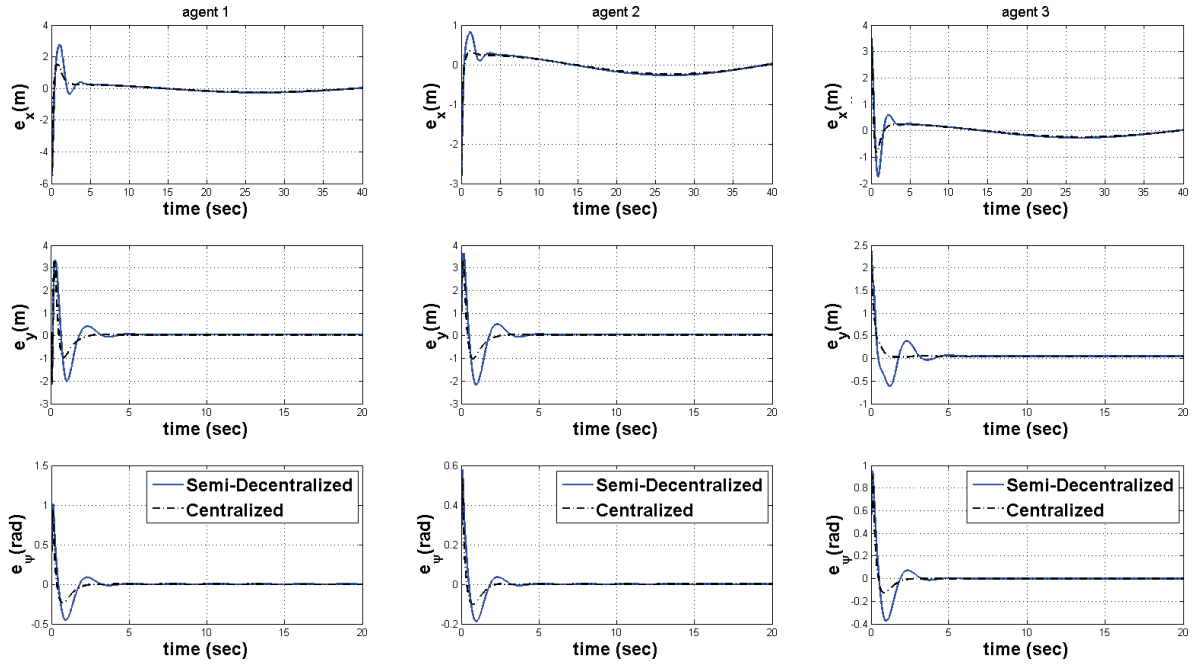


Figure 3.38: Error signals of agents #1, #2, and #3 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.3, case 1.

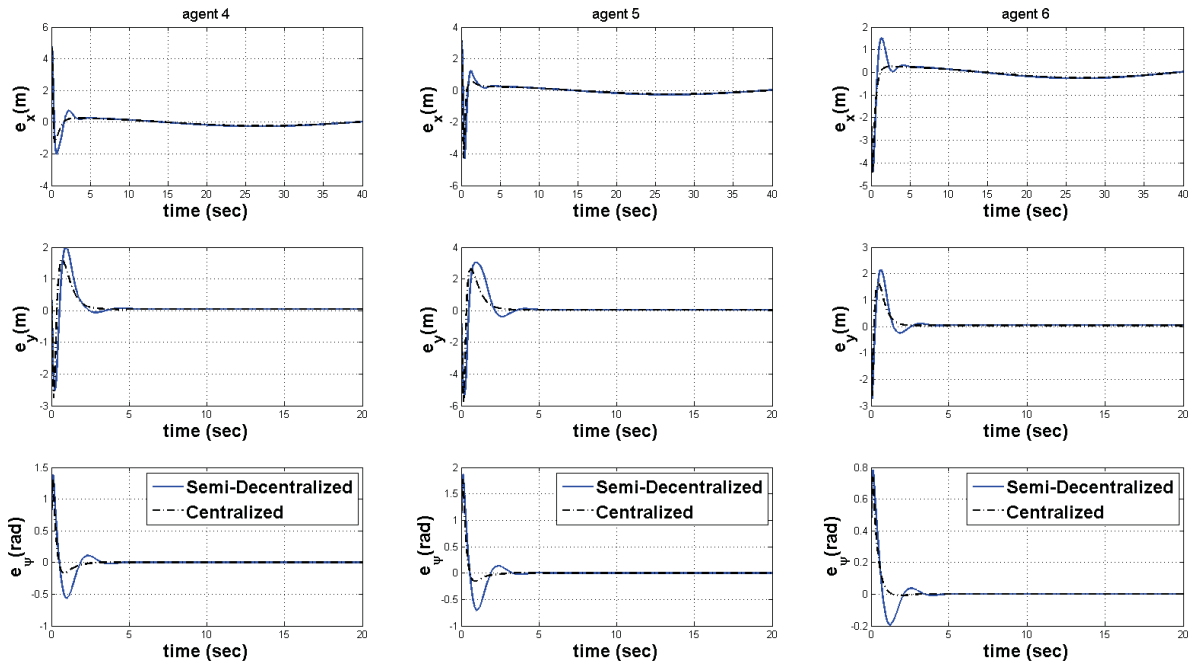


Figure 3.39: Error signals of agents #4, #5, and #6 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.3, case 1.

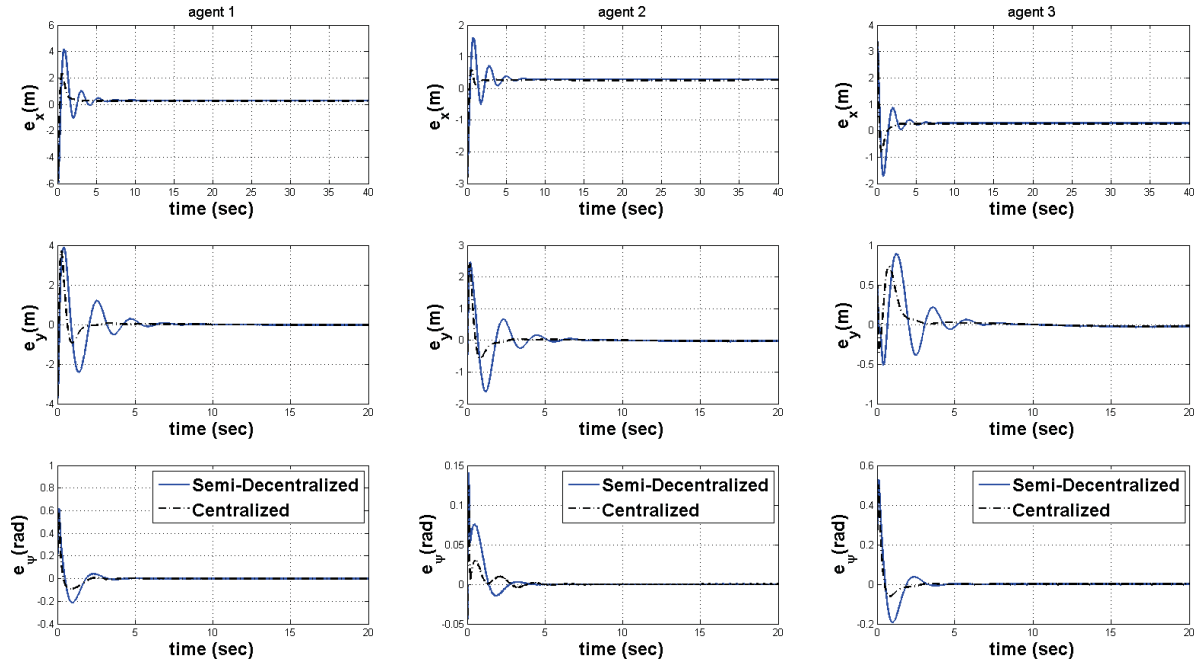


Figure 3.40: Error signals of agents #1, #2, and #3 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.3, case 2.

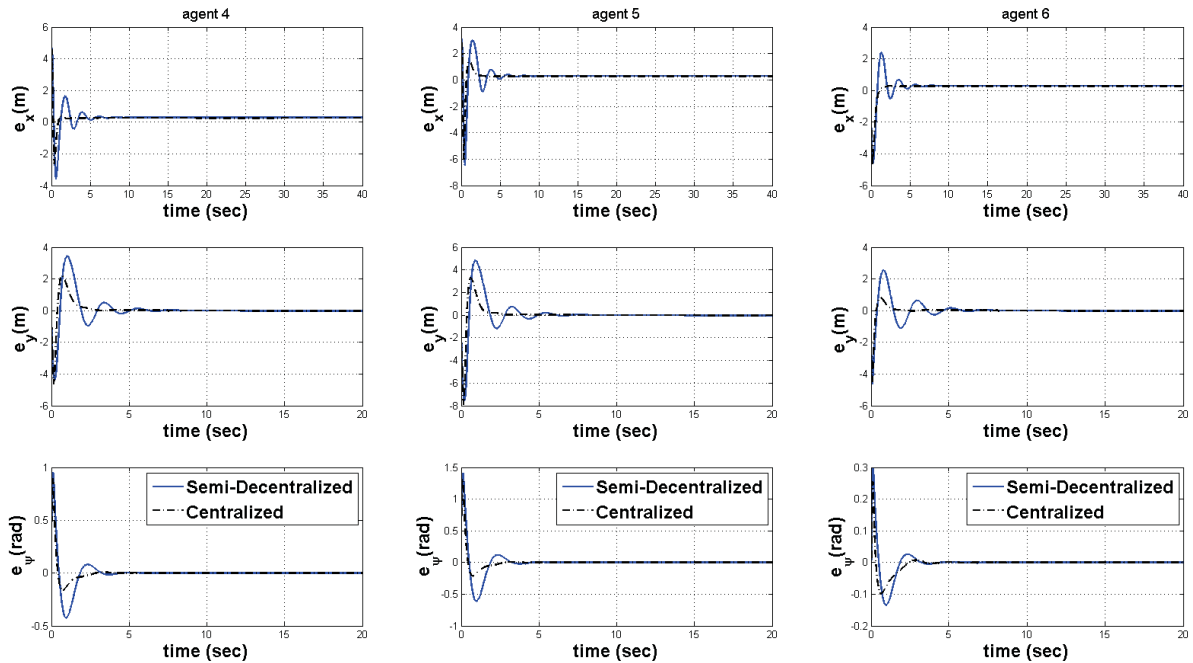


Figure 3.41: Error signals of agents #4, #5, and #6 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.3, case 2.

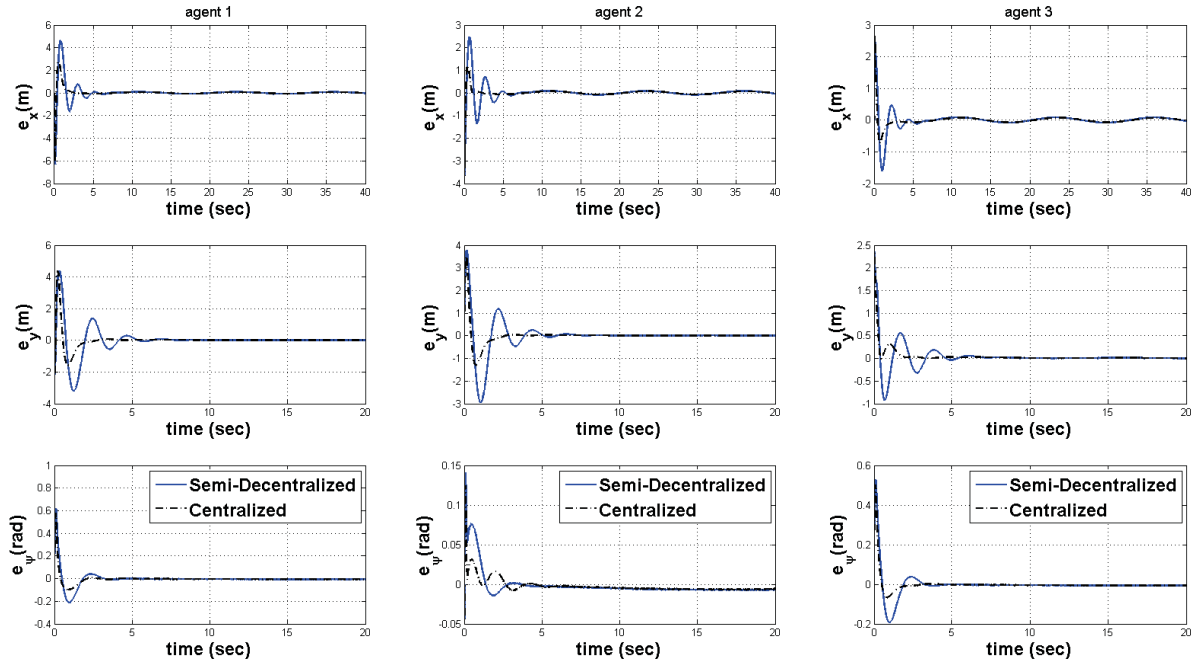


Figure 3.42: Error signals of agents #1, #2, and #3 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.3, case 3.

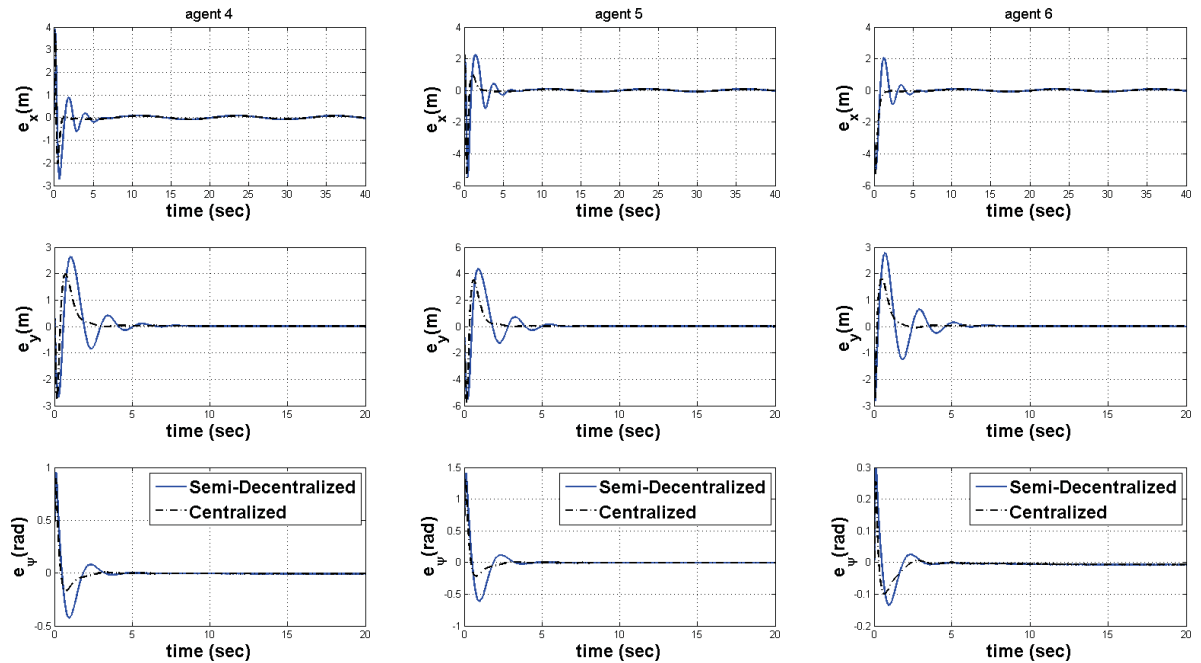


Figure 3.43: Error signals of agents #4, #5, and #6 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.3, case 3.

Table 3.29: Team level RMS analysis of maximum error and steady state error for semi-decentralized and centralized schemes in fault-free situation for scenario 3.3

	Semi-decentralized			Centralized		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
RMS maximum error (m)	3.26187	3.82672	3.50981	3.2256	3.78725	3.46251
RMS steady state error (m)	0.15492	0.16319	0.0347	0.13663	0.14698	0.03434

3.7.4.4 Scenario 3.4: Different Initial Values

In this scenario, we aim to analyze the ability of the proposed semi-decentralized DSC-based scheme to fulfill the objectives which are formation keeping and path-tracking with subject to initial values of each vehicle. To this purpose, several simulation tests have been conducted and the following results are obtained:

- The initial values of linear and angular velocities, i.e. u , v , and r should be less than 20 (m/s).
- The initial values of ψ , yaw orientation, should be less than 130° or 0.73π (rad).
- For surge and sway positions, i.e. x and y respectively, there is a numerical limitation of 1000 (m) which is an unrealistic situation in practice. Therefore, taking into account the underwater data transmit limitations in practical situation, this limit will be much less than this value in real missions.

Also, to provide some sample of results, besides the initial values given in Table 3.1, two more cases of different initial values are considered as in Table 3.30. In the first case, initial

values of x , y , and all velocities given in Table 3.1 are increased ten times, and in the second case, initial values of x and y are increased 50 times and for velocities, they are increased 15 times.

In the simulation tests, a group of six autonomous underwater vehicles in graph G_4 of Table 3.15 is considered for which the desired relative distances, model parameters and uncertainties, and controller parameters are considered the same as Section 3.7. Also, the reference trajectory is considered as

$$\eta_{ref} = \left[15 \sin\left(\frac{t}{8}\right) \left(\frac{t}{3}\right) \left(\frac{\pi}{6}\right) \right]^T$$

In this part, all cases have been simulated for the semi-decentralized and centralized schemes where the centralized scheme is used as a criterion to evaluate the performance of the semi-decentralized scheme. The maximum errors and the steady state errors of all agents in each case for both schemes are tabulated in Tables 3.31 and 3.32 respectively in order to compare two schemes quantitatively. Also, Figures 3.44 to 3.49 represent the results of all agents for all cases given in Table 3.30.

Based on these results, it is concluded that for different initial values, the proposed semi-decentralized DSC-based scheme has the ability to fulfill the objectives of path-tracking and formation keeping. However, if the initial values of positions of agents have been chosen closer to the positions of the reference, there would have been less transient errors. Also, based on the quantitative comparison of the maximum errors and the steady state errors given in Tables 3.31 and 3.32, it can be seen that these errors are slightly higher in the semi-decentralized scheme in comparison to the centralized schemes. The team level comparison of the semi-decentralized and centralized schemes for all cases are presented in Table 3.33 by presenting the root mean square of the maximum errors and the steady state errors. In this table, all results given above

are confirmed.

Table 3.30: Different Initial conditions for scenario 3.4

Case	Agent	Position and Euler Angles	Linear and Angular Velocity
1	1	$\eta(0) = [0.3 \ 1 \ \frac{\pi}{3}]^T$	$\nu(0) = [0.1 \ 0.04 \ 0.4]^T$
	2	$\eta(0) = [0.4 \ 1.2 \ \frac{\pi}{6}]^T$	$\nu(0) = [0.02 \ 0.12 \ 0.03]^T$
	3	$\eta(0) = [0.3 \ 0.1 \ \frac{\pi}{3}]^T$	$\nu(0) = [0.05 \ 0.02 \ 0.15]^T$
	4	$\eta(0) = [-1 \ -0.5 \ \frac{\pi}{2}]^T$	$\nu(0) = [0.1 \ 0.3 \ 0.04]^T$
	5	$\eta(0) = [-0.1 \ -0.2 \ \frac{2\pi}{3}]^T$	$\nu(0) = [0.3 \ 0.01 \ 0.4]^T$
	6	$\eta(0) = [-0.2 \ 0.5 \ \frac{\pi}{4}]^T$	$\nu(0) = [0.4 \ 0.04 \ 0.6]^T$
2	1	$\eta(0) = [3 \ 10 \ \frac{\pi}{3}]^T$	$\nu(0) = [1 \ 0.4 \ 4]^T$
	2	$\eta(0) = [4 \ 12 \ \frac{\pi}{6}]^T$	$\nu(0) = [0.2 \ 1.2 \ 0.3]^T$
	3	$\eta(0) = [3 \ 1 \ \frac{\pi}{3}]^T$	$\nu(0) = [0.5 \ 0.2 \ 1.5]^T$
	4	$\eta(0) = [-10 \ -5 \ \frac{\pi}{2}]^T$	$\nu(0) = [1 \ 3 \ 0.4]^T$
	5	$\eta(0) = [-1 \ -2 \ \frac{2\pi}{3}]^T$	$\nu(0) = [3 \ 0.1 \ 4]^T$
	6	$\eta(0) = [-2 \ 5 \ \frac{\pi}{4}]^T$	$\nu(0) = [4 \ 0.4 \ 6]^T$
3	1	$\eta(0) = [15 \ 50 \ \frac{\pi}{3}]^T$	$\nu(0) = [1.5 \ 0.6 \ 6]^T$
	2	$\eta(0) = [20 \ 60 \ \frac{\pi}{6}]^T$	$\nu(0) = [0.3 \ 1.8 \ 0.45]^T$
	3	$\eta(0) = [15 \ 5 \ \frac{\pi}{3}]^T$	$\nu(0) = [0.75 \ 0.3 \ 2.25]^T$
	4	$\eta(0) = [-50 \ -25 \ \frac{\pi}{2}]^T$	$\nu(0) = [1.5 \ 4.5 \ 0.6]^T$
	5	$\eta(0) = [-5 \ -10 \ \frac{2\pi}{3}]^T$	$\nu(0) = [4.5 \ 0.15 \ 6]^T$
	6	$\eta(0) = [-10 \ 25 \ \frac{\pi}{4}]^T$	$\nu(0) = [6 \ 0.6 \ 9]^T$

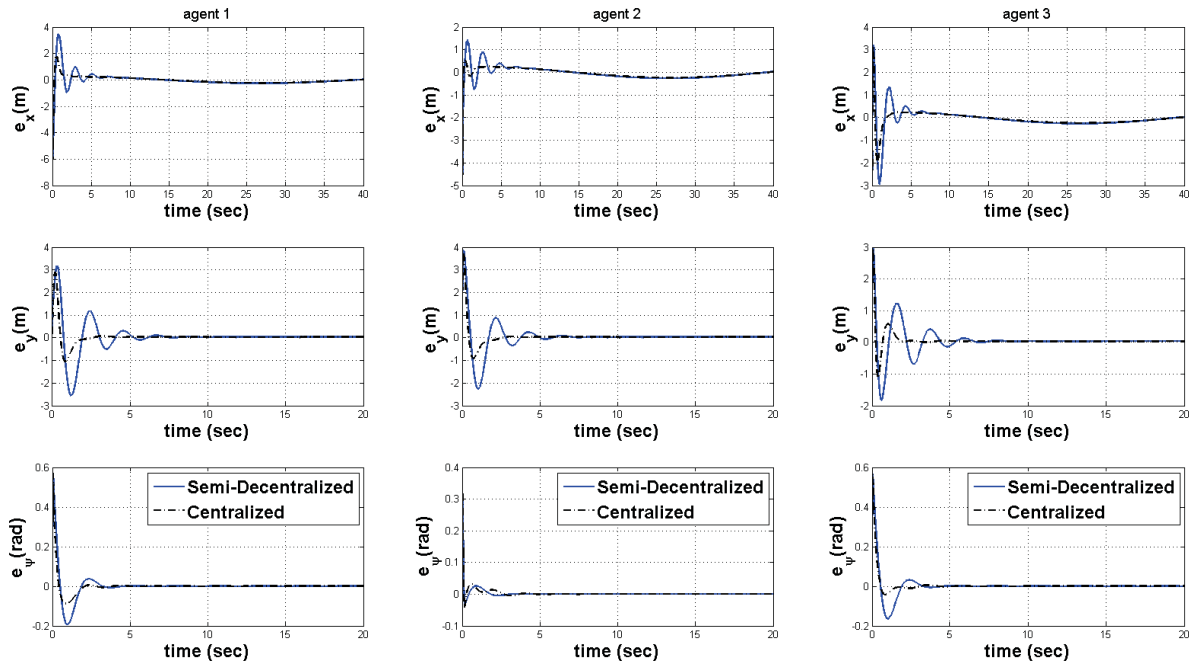


Figure 3.44: Error signals of agents #1, #2, and #3 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 1.

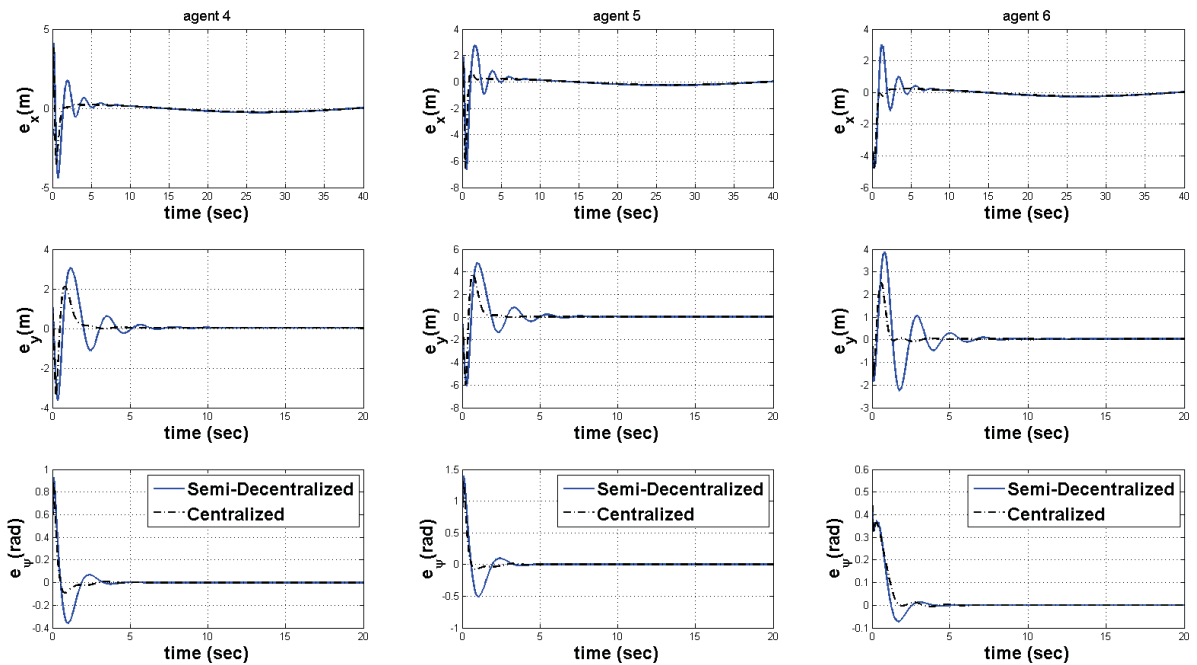


Figure 3.45: Error signals of agents #4, #5, and #6 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 1.

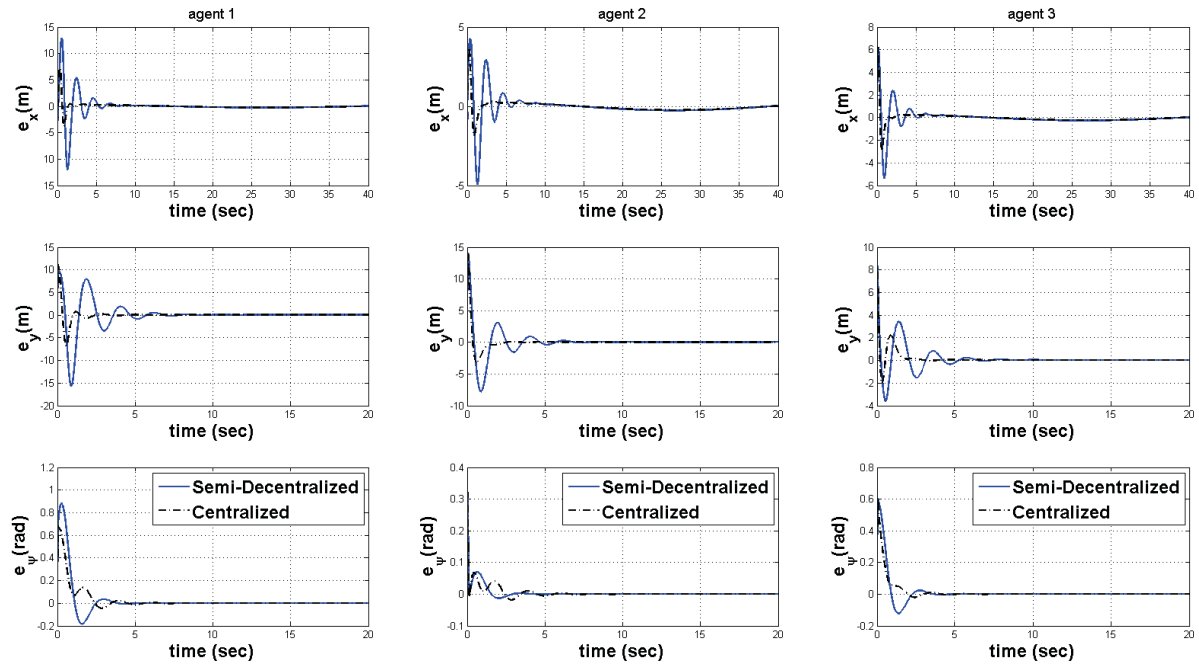


Figure 3.46: Error signals of agents #1, #2, and #3 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 2.

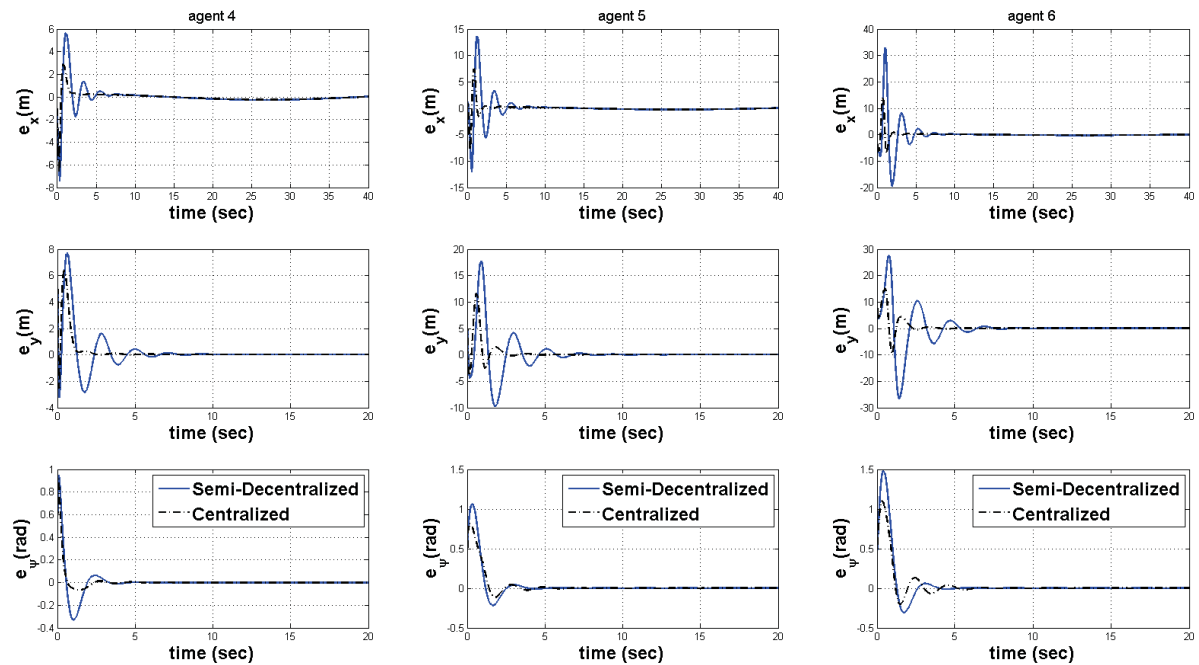


Figure 3.47: Error signals of agents #4, #5, and #6 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 2.

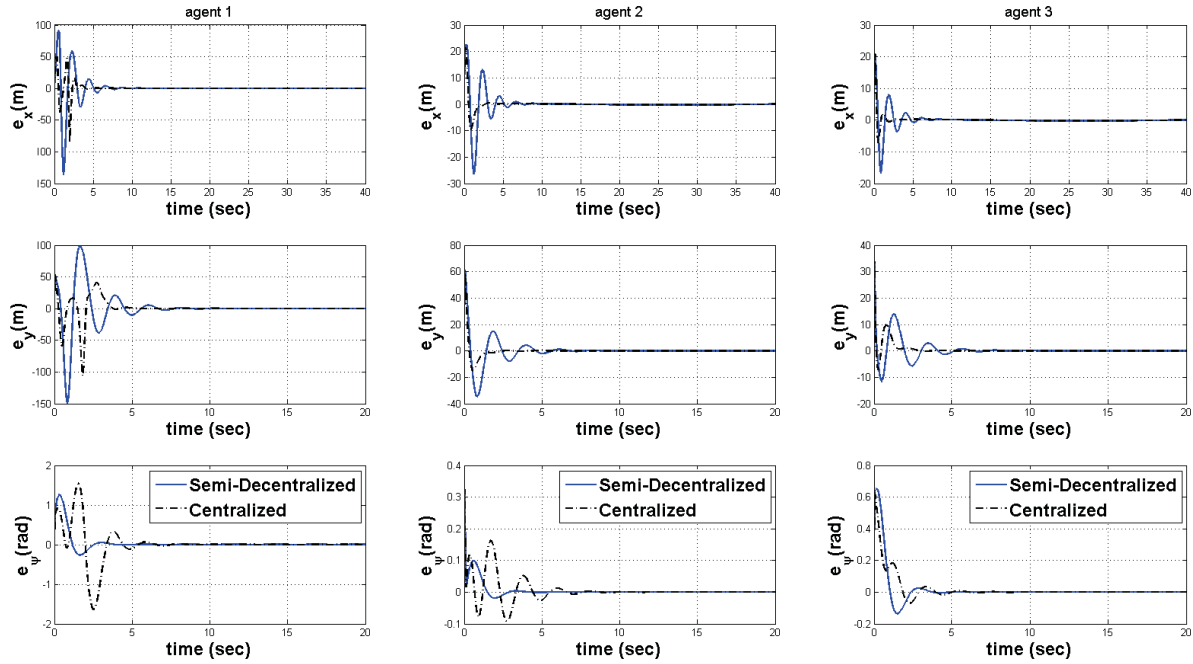


Figure 3.48: Error signals of agents #1, #2, and #3 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 3.

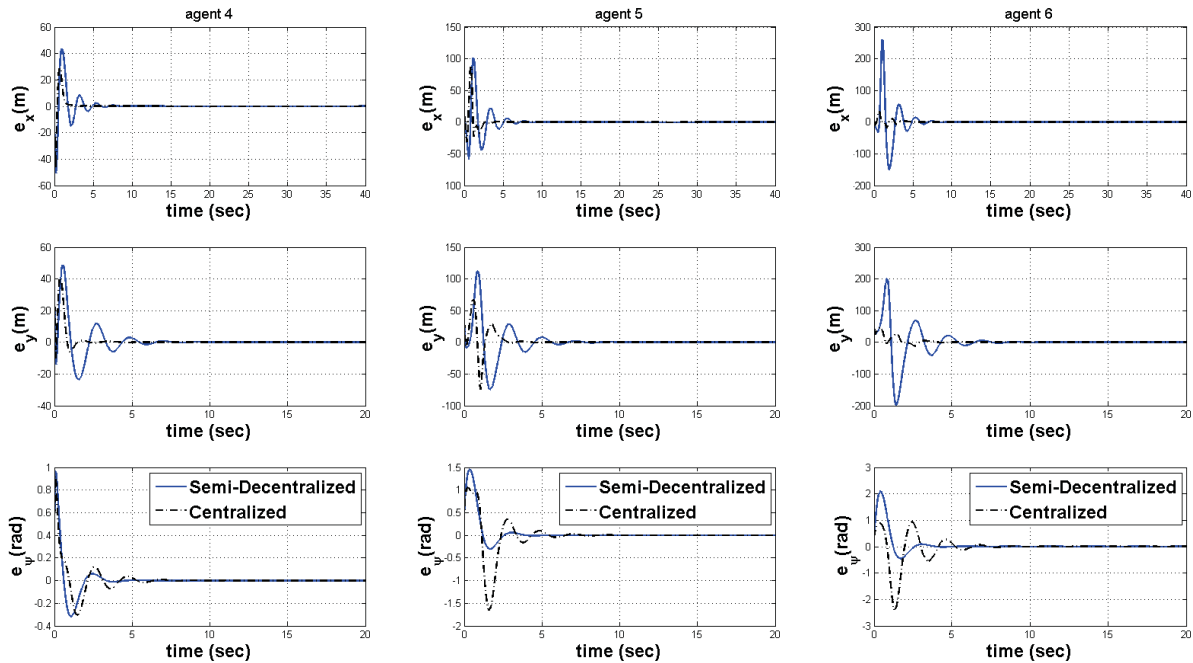


Figure 3.49: Error signals of agents #4, #5, and #6 in fault-free situation for the semi-decentralized and centralized schemes for scenario 3.4, case 3.

Table 3.31: Maximum errors of all cases of the semi-decentralized and centralized schemes for scenario

3.4

		Maximum error (m)					
		Semi-decentralized			Centralized		
		Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
agent 1	x	6.0303	12.85	136.13	6.0302	7.3295	84.8554
	y	3.1664	15.7285	148.91	2.8938	11.1428	106.12
	ψ	0.5699	0.8841	1.2618	0.5698	0.6846	1.6377
agent 2	x	4.5107	4.9729	26.4967	4.5107	3.6751	21.4856
	y	3.8562	14.1276	61.1683	3.7728	13.9976	61.1667
	ψ	0.3162	0.3216	0.3246	0.3162	0.3216	0.3246
agent 3	x	3.1963	6.1557	19.9086	3.1809	6.2239	21.0057
	y	2.9593	8.3256	33.6843	2.9697	8.3256	33.6843
	ψ	0.5656	0.6057	0.6538	0.5656	0.6053	0.6274
agent 4	x	4.3837	7.4233	50.5686	3.8289	6.7011	47.1881
	y	3.6193	7.7393	48.7915	3.4177	6.383	40.7407
	ψ	0.9273	0.9506	0.9637	0.8979	0.9148	0.9242
agent 5	x	6.6683	13.682	101.31	6.523	7.9998	88.21
	y	6.0536	17.7442	112.62	5.933	11.6272	74.3655
	ψ	1.3929	1.0603	1.4547	1.3247	0.8207	1.6538
agent 6	x	4.7766	32.7922	259.88	4.7766	5.272	35.4111
	y	3.872	27.532	201.4	2.5337	10.0477	46.1016
	ψ	0.4392	1.4789	2.0849	0.4392	0.6817	2.3933

Table 3.32: Steady state errors of all cases of the semi-decentralized and centralized schemes for scenario 3.4

		Steady state error (m)					
		Semi-decentralized			Centralized		
		Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
agent 1	x	0.2634	0.2634	0.2634	0.2346	0.2346	0.2346
	y	0.0315	0.0328	0.0406	0.0284	0.0283	0.0276
	ψ	8.1×10^{-9}	8.6×10^{-9}	9.5×10^{-9}	9.5×10^{-6}	1.7×10^{-5}	2.76×10^{-5}
agent 2	x	0.2634	0.2634	0.2634	0.2346	0.2346	0.2346
	y	0.0316	0.032	0.0332	0.0284	0.0283	0.0275
	ψ	1.1×10^{-9}	1.2×10^{-9}	1.3×10^{-9}	8.8×10^{-6}	1.1×10^{-5}	2.75×10^{-5}
agent 3	x	0.2634	0.2634	0.2634	0.2346	0.2346	0.2346
	y	0.0313	0.0312	0.0297	0.0283	0.0284	0.0277
	ψ	7.8×10^{-9}	8×10^{-9}	8.1×10^{-9}	5.3×10^{-6}	2.6×10^{-6}	2.77×10^{-5}
agent 4	x	0.2634	0.2634	0.2635	0.2346	0.2346	0.2347
	y	0.031	0.0308	0.0293	0.0283	0.0283	0.0276
	ψ	1.1×10^{-9}	1.6×10^{-9}	1.8×10^{-9}	2.3×10^{-6}	4×10^{-6}	2.76×10^{-5}
agent 5	x	0.2635	0.2635	0.2635	0.2347	0.2347	0.2347
	y	0.0308	0.0296	0.0225	0.0283	0.0283	0.0278
	ψ	2.7×10^{-8}	7.4×10^{-9}	9×10^{-9}	3.3×10^{-7}	5.2×10^{-6}	2.78×10^{-5}
agent 6	x	0.2635	0.2635	0.2636	0.2347	0.2347	0.2347
	y	0.031	0.0314	0.0279	0.0283	0.0282	0.027
	ψ	1.8×10^{-9}	4.2×10^{-9}	10^{-8}	1.9×10^{-6}	2.6×10^{-6}	2.7×10^{-5}

Table 3.33: Team level RMS analysis of maximum error and steady state error for semi-decentralized and centralized schemes in fault-free situation for scenario 3.4

	Semi-decentralized			Centralized		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
RMS maximum error (m)	3.76796	13.3331	100.74	3.61386	7.11756	49.9056
RMS steady state error (m)	0.153156	0.153164	0.153163	0.136449	0.136447	0.13640

3.8 Conclusion

In this chapter, four cooperative control strategies namely the centralized, decentralized, and two novel semi-decentralized schemes are proposed based on dynamic surface control method to solve the problem of formation path-tracking of a group of heterogeneous autonomous underwater vehicles with locally Lipschitz uncertainties. The heterogeneity considered for the vehicles is assumed to be in their internal model. It is assumed that for each agent there are the same number of states while the model of each vehicle might vary. For the proposed semi-decentralized schemes, the leader-follower structure is chosen in which the information of the virtual leader is only known to a subset of agents.

To show the stability of the group in these cooperative strategies, the stability analysis for the decentralized and both semi-decentralized schemes are presented. As demonstrated in this chapter, each distribution scheme has its own pros and cons. For the problem considered in this thesis, based on the simulation results, it can be concluded that the semi-decentralized scheme with consensus algorithm using desired trajectories enhances the formation-tracking performance in both formation keeping and path-tracking problems in comparison to the decentralized

scheme where there is no communications between agents. Moreover, the semi-decentralized scheme with consensus algorithm using desired trajectories has a fairly similar response to the centralized scheme while there is no need to solve the problem of high dimension with significant computational complexities. Although the centralized scheme has the optimal solution, since it is impractical to be implemented in many cases, the semi-decentralized scheme is the best solution for the given problem since it does not require strict communication constraints.

To evaluate the performance of the semi-decentralized scheme with consensus algorithm using desired trajectories, simulation results of various scenarios and comparative studies are represented. These scenarios vary in the number of agents, network topologies, initial values, and reference trajectories. From the comparative study between the semi-decentralized scheme with consensus algorithm using desired trajectories and the centralized, decentralized, the semi-decentralized with consensus algorithm using actual states, and three cooperative schemes, i.e. centralized, decentralized, and semi-decentralized schemes, based on model-dependent coordinated tracking algorithm, the efficient performance of the proposed DSC-based scheme is evident. One of the advantages of the proposed DSC-based semi-decentralized control scheme over the model-dependent coordinated tracking algorithm is that in the proposed DSC-based semi-decentralized control scheme only the desired positions of agents are communicated to reach consensus while in the model-dependent coordinated tracking algorithm, agents need to communicate both their position and velocity states to fulfill the objectives.

In addition, from the simulation scenarios of this chapter, it can be concluded that the proposed semi-decentralized scheme has the significant performance independent of the number of agents in the group, the topology of the network, and the reference trajectory. In the simulation results, the effect and boundary of the initial values of agents have been also illustrated.

Chapter 4

DSC-based Cooperative Fault-tolerant Control Schemes

In this chapter, active fault-tolerant scheme of cooperative control strategies presented in Chapter 3 is addressed. In the first part, the fault-tolerant DSC-based control scheme of one autonomous underwater vehicle is presented in Section 4.1. In Sections 4.2, the cooperative fault-tolerant DSC-based control strategies for a group of heterogeneous AUVs are presented with their stability analysis. Finally, in Section 4.3, the simulation results of various scenarios and comparative studies between the proposed fault-tolerant semi-decentralized, centralized, and decentralized schemes are represented in order to evaluate the performance of the proposed fault-tolerant semi-decentralized DSC-based control strategy.

4.1 Fault-tolerant Control for an AUV Using DSC Technique

In this thesis, the DSC technique is used to overcome the problem of controlling systems with model uncertainties. In this chapter, the objective is to control a system not only with model uncertainties, but also with faults. Since AUV dynamics are inevitably subjected to all kinds of system fault, to improve the reliability of these vehicles, the fault-tolerant control technique must be considered when designing a control system for AUVs. In this chapter, the goal is to design an active fault-tolerant system for a group of networked AUVs based on DSC technique in order to recover system from the possible loss-of-effectiveness actuator faults and to ensure that the closed-loop signals are bounded and the group of heterogeneous AUVs still satisfies an acceptable performance for both formation and individual behaviors by tracking the desired trajectory. The active fault-tolerant system uses the information provided by the FDI module such as the time that faults happen in the system, the fault severity, and the actuator in which fault occurred. These information provided by the FDI module might contain errors which should be taken into account.

To the best of the knowledge of the author, the topic of active fault-tolerant control for AUVs has not been completely addressed in the literature, which remains challenging and motivating to do this study.

In this chapter, the control input τ_H for healthy situations, i.e. actuator fault-free case, which was designed in the previous chapter based on DSC technique is used. Once the FDI module sends the information about the existence of an actuator fault at time t_c , a compensation control input τ_C will be activated and added to the healthy control input τ_H to reduce the effects of the actuator fault. It is worth mentioning that t_c might be different than t_f , the time that faults happens, which is because of the error in detection by the FDI module. Figure 4.1 shows

the configuration of the active fault-tolerant control scheme for a given system.

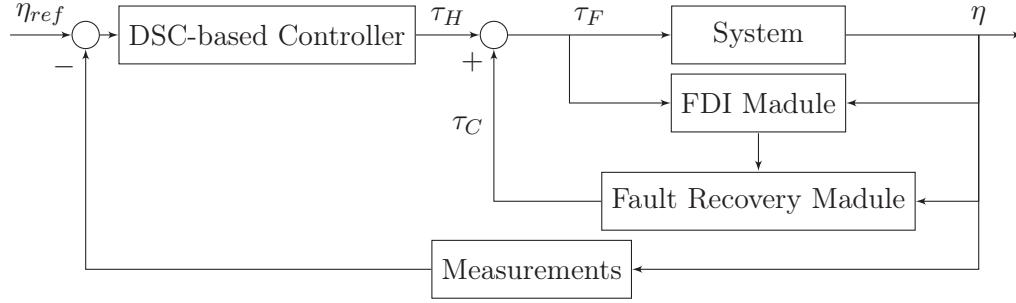


Figure 4.1: Block diagram of an active fault-tolerant control scheme of a system.

When the FDI module sends the information about the existence of actuator faults, in order to formulate the fault-tolerant control problem, the faulty dynamics of an AUV must be established. Using the state space representation in controllable canonical form of an AUV given in equation (2.2.15), the dynamics of an autonomous underwater vehicle under LOE actuator fault is given by:

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = M_\eta^{-1}((I - F)\tau_F - C_\eta(z_1, z_2)z_2 - D_\eta(z_1)z_2) + \Delta f_\eta(z_1) \end{cases} \quad (4.1.1)$$

where τ_F is the fault-tolerant control input, $F = \text{diag}\{f_j\}$ is the actual effectiveness coefficient matrix of all actuators for $j = \{1, 2, 3\}$ with $f_j \in [0, 1)$ as the percentage of the actual loss of control effectiveness faults. In reality, the information of the faults that the FDI module provides is an estimation of the actual faults occurred in the system. Therefore, the information that is used in control reconfiguration scheme is $\hat{F} = \text{diag}\{\hat{f}_j\}$ for $j = \{1, 2, 3\}$ where $\hat{f}_j \in [0, 1)$ is the percentage of the estimated loss of control effectiveness faults provided by the FDI module and it is assumed that $0 \leq |f_j - \hat{f}_j| < 1$ and consequently $\mathbf{0}_3 \leq \|F - \hat{F}\| < I_3$ where I_3 and $\mathbf{0}_3$ are the 3×3 identity and zero matrices, respectively.

As shown in the previous chapter, the healthy controller in equation (3.4.3) can achieve the

stability of the system and tracking of a desired reference for an AUV in actuator fault-free case. In this section, this result is extended to deal with the fault-tolerant control problem of an AUV, and then it will be extended for cooperative schemes. In this case, a compensation control input τ_C will be designed on the basis of the healthy controller τ_H to compensate for the effects of actuator faults. Therefore, the fault-tolerant control input τ_F of the faulty system in equation (4.1.1) consists of two parts as

$$\tau_F = \tau_H + \tau_C \quad (4.1.2)$$

For the fault-tolerant controller design, the first step of design procedure which is defining the first dynamic surface is the same as design of the healthy control input τ_H . The second step of designing the compensation control input is based on the idea of [107]. To start the design procedure, the first step is to define the first vector of error surfaces as

$$S_1 := z_1 - z_{1d} \quad (4.1.3)$$

Differentiating S_1 along the trajectories of the closed-loop system using the fault-tolerant control input given in equation (4.1.2) yields

$$\dot{S}_1 = z_2 - \dot{z}_{1d} \quad (4.1.4)$$

Choosing \bar{z}_2 as

$$\bar{z}_2 = \dot{z}_{1d} - \lambda_1 S_1 \quad (4.1.5)$$

where λ_1 is an arbitrary positive definite gain matrix, leads to $\dot{S}_1 = -\lambda_1 S_1$ which indicates that $S_1 \rightarrow 0$ in case that $z_2 \rightarrow \bar{z}_2$. Therefore, the second step is to force $z_2 \rightarrow \bar{z}_2$ by defining the second vector of error surfaces as

$$S_2 := z_2 - \bar{z}_2 \quad (4.1.6)$$

Differentiating S_2 results in

$$\begin{aligned}
\dot{S}_2 &= M_\eta^{-1} ((I - F)\tau_F - C_\eta(z_1, z_2)z_2 - D_\eta(z_1)z_2) + \Delta f_\eta(z_1) - \dot{\bar{z}}_2 \\
&= M_\eta^{-1} ((I - F)\tau_H + (I - F)\tau_C - C_\eta(z_1, z_2)z_2 - D_\eta(z_1)z_2) + \Delta f_\eta(z_1) - \dot{\bar{z}}_2 \\
&= M_\eta^{-1} (\tau_H - C_\eta(z_1, z_2)z_2 - D_\eta(z_1)z_2) + \Delta f_\eta(z_1) - \dot{\bar{z}}_2 + M_\eta^{-1} (-F\tau_H + (I - F)\tau_C)
\end{aligned} \tag{4.1.7}$$

where

$$\dot{\bar{z}}_2 = \ddot{z}_{1d} - \lambda_1 (z_2 - \dot{z}_{1d}) \tag{4.1.8}$$

Based on equations (4.1.4) and (4.1.7), the closed-loop error dynamics is given by

$$\begin{cases} \dot{S}_1 = z_2 - \dot{z}_{1d} \\ \dot{S}_2 = M_\eta^{-1} (\tau_H - C_\eta(z_1, z_2)z_2 - D_\eta(z_1)z_2) + \Delta f_\eta(z_1) \\ \quad - \dot{\bar{z}}_2 + M_\eta^{-1} (-F\tau_H + (I - F)\tau_C) \end{cases} \tag{4.1.9}$$

To guarantee the stability of the closed-loop system, the Lyapunov function candidate is chosen as

$$V = \frac{1}{2} (S_1^T S_1 + S_2^T S_2) \tag{4.1.10}$$

Differentiating V along the trajectories of equation (4.1.9) is given by

$$\begin{aligned}
\dot{V} &= S_1^T \dot{S}_1 + S_2^T \dot{S}_2 \\
&= S_1^T [z_2 - \dot{z}_{1d}] + S_2^T [M_\eta^{-1} (\tau_H - C_\eta(z_1, z_2)z_2 - D_\eta(z_1)z_2) \\
&\quad + \Delta f_\eta(z_1) - \dot{\bar{z}}_2 + M_\eta^{-1} (-F\tau_H + (I - F)\tau_C)]
\end{aligned} \tag{4.1.11}$$

Considering equation (4.1.8) and τ_H from equation (2.2.16) as

$$\tau_H = M_\eta \left(\ddot{z}_{1d} - \lambda_1 (z_2 - \dot{z}_{1d}) - \lambda_2 S_2 - \frac{1}{2\varepsilon} (S_2 \circ P) \right) + C_\eta(z_1, z_2)z_2 + D_\eta(z_1)z_2$$

leads to

$$\begin{aligned}\dot{V} = & -S_1^T \lambda_1 S_1 + S_1^T S_2 - S_2^T \lambda_2 S_2 + S_2^T \Delta f_\eta(z_1) \\ & - \frac{S_2^T (S_2 \circ P)}{2\varepsilon} + S_2^T M_\eta^{-1} (-F\tau_H + (I - F)\tau_C)\end{aligned}\quad (4.1.12)$$

Using Young's inequality leads to

$$\dot{V} \leq -S_1^T \lambda_1 S_1 + S_1^T S_2 - S_2^T \lambda_2 S_2 + 3\frac{\varepsilon}{2} + S_2^T M_\eta^{-1} (-F\tau_H + (I - F)\tau_C)\quad (4.1.13)$$

Defining the compensation control input as

$$\tau_C := -\frac{S_2}{\|S_2\| + \delta_2} \left(\|\tau_H\| + \frac{\delta_1 \|\tau_H\|}{\varsigma} \right)\quad (4.1.14)$$

where $\varsigma = [1 - \max\{\hat{f}_j\}] \neq 0$ for $j = \{1, 2, 3\}$ and δ_1 and δ_2 are positive constant scalars,

leads to

$$\begin{aligned}\dot{V} \leq & -S_1^T \lambda_1 S_1 + S_1^T S_2 - S_2^T \lambda_2 S_2 + 3\frac{\varepsilon}{2} - S_2^T M_\eta^{-1} F\tau_H \\ & - S_2^T M_\eta^{-1} (I - F) \frac{S_2}{\|S_2\| + \delta_2} \left(\|\tau_H\| + \frac{\delta_1 \|\tau_H\|}{\varsigma} \right)\end{aligned}\quad (4.1.15)$$

Using Cauchy-Schwarz inequality leads to

$$\begin{aligned}\dot{V} \leq & -S_1^T \lambda_1 S_1 + S_1^T S_2 - S_2^T \lambda_2 S_2 + 3\frac{\varepsilon}{2} - \|S_2^T\| \|M_\eta^{-1} F\| \|\tau_H\| \\ & - \frac{(I - F) \left(1 + \frac{\delta_1}{1 - \max\{\hat{f}_j\}} \right)}{\|S_2\| + \delta_2} \|\tau_H\| S_2^T M_\eta^{-1} S_2\end{aligned}\quad (4.1.16)$$

$\frac{(I - F) \left(1 + \frac{\delta_1}{1 - \max\{\hat{f}_j\}} \right)}{\|S_2\| + \delta_2}$ is a positive quantity since F , $I - F$, $1 - \max\{\hat{f}_j\}$, and δ_1 are

positive. Also, M_η^{-1} is positive definite as discussed in Section 2.2.4. Therefore, with choices

of large enough λ_1 and λ_2 and proper choice of ε it can be obtained that \dot{V} is negative semi-

definite for a region in (S_1, S_2) which shows that the system is locally stable and boundedness

of its trajectories.

Remark 4.1. Based on this proof, it can be seen that with proper choices of λ_1 , λ_2 , and δ_1 , the difference between actual and estimated faults does not theoretically affect the boundedness of the trajectories of the system.

4.2 Cooperative FTC Scheme for Multiple AUVs Using DSC

Technique

In this section, cooperative fault-tolerant control schemes based on dynamic surface control technique for underwater vehicles subject to LOE faults are addressed. Having the reconfiguration procedure given in previous section, the DSC-based fault-tolerant centralized, decentralized, and semi-decentralized schemes are addressed in this section.

4.2.1 Fault-tolerant DSC-based Centralized Control Scheme

Considering the fully connected centralized architecture described in Section 3.3, the recovery control law of the DSC-based centralized control scheme for a group of n underwater vehicles with m states is formulated as

$$\mathcal{I}_F = \mathcal{I}_H + \mathcal{I}_C$$

where $\mathcal{I}_H \in \mathbb{R}^{(m \times n) \times 1}$ is given by equation (3.3.1) and $\mathcal{I}_C \in \mathbb{R}^{(m \times n) \times 1}$ is defined as

$$\mathcal{I}_C = -\frac{\underline{S}_2}{\|\underline{S}_2\| + \delta_2} \left(\|\mathcal{I}_H\| + \frac{\delta_1 \|\mathcal{I}_H\|}{\bar{\varsigma}} \right)$$

in which δ_1 and δ_2 are positive constant quantities and

- $\bar{\varsigma} = \min_{1 \leq i \leq n} \{\varsigma^{(i)}\} = \min_{1 \leq i \leq n} \left\{ 1 - \max_{1 \leq j \leq m} \{\hat{f}_j^{(i)}\} \right\} \neq 0$;
- $\underline{S}_2 = \left[(z_2^{(1)} - \bar{z}_2^{(1)}) \dots (z_2^{(n)} - \bar{z}_2^{(n)}) \right]^T \in \mathbb{R}^{(m \times n) \times 1}$ is the column stack vector of the

second surface errors with

$$\bar{z}_2 = -\lambda_1 \underline{S}_1 + \dot{z}_{1d} \quad (4.2.1)$$

with \dot{z}_{1d} given by equation (3.2.1) and

- $\underline{S}_1 = \left[(z_1^{(1)} - z_{1d}^{(1)}) \dots (z_1^{(n)} - z_{1d}^{(n)}) \right]^T \in \mathbb{R}^{(m \times n) \times 1}$ is the column stack vector of the first surface errors.

Applying this centralized control law to control a group of heterogeneous autonomous underwater vehicles in which agents are subject to LOE actuator faults leads to stability of the system where all $z_1^{(i)}$ s and $z_2^{(i)}$ s are bounded for $1 \leq i \leq n$ and they are within an arbitrarily small boundary around the desired trajectory..

Remark 4.2. Since in this scheme, the corresponding dynamics system models of all agents are concatenated in one high dimensions model, overall it can be seen as one agent. Therefore, the stability analysis of the system under the DSC-based fault-tolerant centralized control scheme is the same as in Section 4.1.

4.2.2 Fault-tolerant DSC-based Decentralized Control Scheme

Based on the results given in Section 4.1 and the decentralized architecture given in Section 3.4, the DSC-based fault-tolerant decentralized control scheme for a group of n agents will be formulated as follows:

$$\tau_F^{(i)} = \tau_H^{(i)} + \tau_C^{(i)} \quad (4.2.2)$$

where

$$\tau_C^{(i)} = -\frac{S_2^{(i)}}{\|S_2^{(i)}\| + \delta_2^{(i)}} \left(\|\tau_H^{(i)}\| + \frac{\delta_1^{(i)} \|\tau_H^{(i)}\|}{\varsigma^{(i)}} \right) \quad (4.2.3)$$

in which $\delta_1^{(i)}$ and $\delta_2^{(i)}$ are positive constant quantities, $\varsigma^{(i)} = [1 - \max\{\hat{f}_j^{(i)}\}] \neq 0$ for $j = \{1, 2, 3\}$, and

$$\tau_H^{(i)} = M_\eta^{(i)} \left(\dot{\bar{z}}_2^{(i)} - \lambda_2^{(i)} S_2^{(i)} - \frac{S_2^{(i)} \circ P^{(i)}}{2\varepsilon} \right) + C_\eta^{(i)}(z_1^{(i)}, z_2^{(i)}) z_2^{(i)} + D_\eta^{(i)}(z_1^{(i)}) z_2^{(i)} \quad (4.2.4)$$

where

$$S_2^{(i)} = z_2^{(i)} - \bar{z}_2^{(i)}$$

with $\bar{z}_2^{(i)}$ defined the same as equation (3.4.4) as

$$\bar{z}_2^{(i)} = \dot{z}_{1d}^{(i)} - \lambda_1^{(i)} S_1^{(i)}$$

with its derivative as

$$\dot{\bar{z}}_2^{(i)} = \ddot{z}_{1d}^{(i)} - \lambda_1^{(i)} \left(z_2^{(i)} - \dot{z}_{1d}^{(i)} \right) \quad (4.2.5)$$

where z_{1d} given by equation (3.2.3).

Using the control input given by equation (4.2.2) for a team of heterogeneous AUVs where agents are subject to LOE actuator faults leads to stability of the system where z_1 and z_2 of all agents are bounded and all z_{1s} are within an arbitrarily small neighborhood around the desired trajectory. The stability analysis of the agents under this fault-tolerant decentralized control scheme is given as follows.

4.2.2.1 Stability Analysis

In this section, the stability of the i^{th} agent in the decentralized fault-tolerant control scheme is shown. The closed-loop error dynamics is

$$\begin{cases} \dot{S}_1^{(i)} = z_2^{(i)} - \dot{z}_{1d}^{(i)} \\ \dot{S}_2^{(i)} = M_\eta^{-1(i)} \left(\tau_H^{(i)} - C_\eta^{(i)}(z_1^{(i)}, z_2^{(i)}) z_2^{(i)} - D_\eta^{(i)}(z_1^{(i)}) z_2^{(i)} \right) + \Delta f_\eta^{(i)}(x_1^{(i)}) \\ \quad - \dot{\bar{z}}_2^{(i)} + M_\eta^{-1(i)} \left(-F^{(i)} \tau_H^{(i)} + (I - F^{(i)}) \tau_C^{(i)} \right) \end{cases} \quad (4.2.6)$$

where n is the number of agents. To guarantee the stability of the closed-loop system, the Lyapunov function candidate is selected as

$$V^{(i)} = \frac{1}{2} \left(S_1^{T(i)} S_1^{(i)} + S_2^{T(i)} S_2^{(i)} \right) \quad (4.2.7)$$

Derivative of V along the trajectories of equation (4.2.6) is given by

$$\begin{aligned} \dot{V}^{(i)} &= S_1^{T(i)} \dot{S}_1^{(i)} + S_2^{T(i)} \dot{S}_2^{(i)} \\ &= S_1^{T(i)} \left[z_2^{(i)} - \dot{z}_{1d}^{(i)} \right] + S_2^{T(i)} \left[M_\eta^{-1(i)} \left(\tau_H^{(i)} - C_\eta^{(i)}(z_1^{(i)}, z_2^{(i)}) z_2^{(i)} - D_\eta^{(i)}(z_1^{(i)}) z_2^{(i)} \right) \right. \\ &\quad \left. + \Delta f_\eta^{(i)}(z_1^{(i)}) - \dot{z}_2^{(i)} + M_\eta^{-1(i)} \left(-F^{(i)} \tau_H^{(i)} + (I - F^{(i)}) \tau_C^{(i)} \right) \right] \end{aligned} \quad (4.2.8)$$

Using equations (4.2.4) and (4.2.5) leads to

$$\begin{aligned} \dot{V}^{(i)} &= -S_1^{T(i)} \lambda_1^{(i)} S_1^{(i)} - S_2^{T(i)} \lambda_2^{(i)} S_2^{(i)} + S_1^{T(i)} S_2^{(i)} + S_2^{T(i)} \Delta f_\eta^{(i)}(z_1^{(i)}) \\ &\quad - \frac{S_2^{T(i)} (S_2^{(i)} \circ P^{(i)})}{2\varepsilon} + S_2^{T(i)} M_\eta^{-1(i)} \left(-F^{(i)} \tau_H^{(i)} + (I - F^{(i)}) \tau_C^{(i)} \right) \end{aligned} \quad (4.2.9)$$

Using Young's inequality leads to

$$\dot{V}^{(i)} \leq -S_1^{T(i)} \lambda_1^{(i)} S_1^{(i)} - S_2^{T(i)} \lambda_2^{(i)} S_2^{(i)} + S_1^{T(i)} S_2^{(i)} + 3\frac{\varepsilon}{2} + S_2^{T(i)} M_\eta^{-1(i)} \left(-F^{(i)} \tau_H^{(i)} + (I - F^{(i)}) \tau_C^{(i)} \right) \quad (4.2.10)$$

Using equation (4.2.3) yields

$$\begin{aligned} \dot{V}^{(i)} &\leq -S_1^{T(i)} \lambda_1^{(i)} S_1^{(i)} - S_2^{T(i)} \lambda_2^{(i)} S_2^{(i)} + S_1^{T(i)} S_2^{(i)} + 3\frac{\varepsilon}{2} - S_2^{T(i)} M_\eta^{-1(i)} F^{(i)} \tau_H^{(i)} \\ &\quad - S_2^{T(i)} M_\eta^{-1(i)} (I - F^{(i)}) \frac{S_2^{(i)}}{\|S_2^{(i)}\| + \delta_2^{(i)}} \left(\|\tau_H^{(i)}\| + \frac{\delta_1^{(i)} \|\tau_H^{(i)}\|}{\varsigma^{(i)}} \right) \\ &\leq -S_1^{T(i)} \lambda_1^{(i)} S_1^{(i)} - S_2^{T(i)} \lambda_2^{(i)} S_2^{(i)} + S_1^{T(i)} S_2^{(i)} + 3\frac{\varepsilon}{2} - \|S_2^{T(i)}\| \|M_\eta^{-1(i)} F^{(i)}\| \|\tau_H^{(i)}\| \\ &\quad - \frac{(I - F^{(i)}) \left(1 + \frac{\delta_1^{(i)}}{1 - \max\{\hat{f}_j^{(i)}\}} \right)}{\|S_2^{(i)}\| + \delta_2^{(i)}} \|\tau_H^{(i)}\| \|S_2^{T(i)} M_\eta^{-1(i)} S_2^{(i)}\| \end{aligned} \quad (4.2.11)$$

Since $I - F^{(i)}$, $1 - \max\{\hat{f}_j^{(i)}\}$, and $\delta_1^{(i)}$ are positive quantities, the choices of sufficiently large $\lambda_1^{(i)}$ and $\lambda_2^{(i)}$, and the proper choice of ε lead to negative semi-definite $\dot{V}^{(i)}$ in

$$\mathbb{S} = \{(S_1, S_2) \mid -S_1^{T(i)} \lambda_1^{(i)} S_1^{(i)} - S_2^{T(i)} \lambda_2^{(i)} S_2^{(i)} + S_1^{T(i)} S_2^{(i)} + 3\frac{\varepsilon}{2} - \|S_2^{T(i)}\| \|M_\eta^{-1(i)} F^{(i)}\| \|\tau_H^{(i)}\| \\ - \frac{(I - F^{(i)}) \left(1 + \frac{\delta_1^{(i)}}{1 - \max\{\hat{f}_j^{(i)}\}}\right)}{\|S_2^{(i)}\| + \delta_2^{(i)}} \|\tau_H^{(i)}\| S_2^{T(i)} M_\eta^{-1(i)} S_2^{(i)} \leq 0\}$$

which shows the locally stability of each agent in the decentralized scheme and also the boundedness of its trajectories. Also, the stability of the entire group can be shown in the same way as in Section 3.4.1.

4.2.3 Fault-tolerant DSC-based Semi-Decentralized Control Scheme with Consensus Algorithm Using Desired Trajectories

In the semi-decentralized fault-tolerant control scheme, the objective is that a group of AUVs track a desired trajectory and remain a desired formation positioning while there are LOE faults in one or more agents. In this cooperative scheme, agents will communicate only their relative positions with the agents in their set of neighbors in order to reach consensus and a subset of them receive data from the virtual leader. All assumptions of Section 3.5 apply here as well. The proposed semi-decentralized fault-tolerant control scheme is given as

$$\tau_F^{(i)} = \tau_H^{(i)} - \underbrace{\frac{S_2^{(i)}}{\|S_2^{(i)}\| + \delta_2^{(i)}} \left(\|\tau_H^{(i)}\| + \frac{\delta_1^{(i)} \|\tau_H^{(i)}\|}{\varsigma^{(i)}} \right)}_{\tau_C^{(i)}} \quad (4.2.12)$$

where $\tau_C^{(i)}$ is the compensation control input of i^{th} agent, $S_2^{(i)} = z_2^{(i)} - \bar{z}_2^{(i)}$ is the vector of second error surfaces, $\varsigma^{(i)} = [1 - \max\{\hat{f}_j^{(i)}\}] \neq 0$ for $j = \{1, 2, 3\}$, and $\tau_H^{(i)}$ is the healthy control input given by

$$\tau_\eta^{(i)} = M_\eta^{(i)} \left(\dot{\bar{z}}_2^{(i)} - \lambda_2^{(i)} S_2^{(i)} - \frac{S_2^{(i)} \circ P^{(i)}}{2\varepsilon} \right) + C_\eta^{(i)}(z_1^{(i)}, z_2^{(i)}) z_2^{(i)} + D_\eta^{(i)}(z_1^{(i)}) z_2^{(i)} \quad (4.2.13)$$

in which $\dot{z}_2^{(i)}$ is given by

$$\dot{z}_2^{(i)} = -\lambda_1^{(i)} \left(z_2^{(i)} - \dot{z}_{1d}^{(i)} \right) + \ddot{z}_{1d}^{(i)} \quad (4.2.14)$$

where the trajectory $z_{1d}^{(i)}$ is obtained from

$$\dot{z}_{1d}^{(i)} = -\xi_1 \left(\sum_{j \in N_i} (z_{1d}^{ij} - \sigma^{ij}) + a_{i0}(z_{1d}^{i0} - \sigma^{i0}) \right) - \xi_2 \frac{\sum_{j \in N_i} (z_{1d}^{ij} - \sigma^{ij}) + a_{i0}(z_{1d}^{i0} - \sigma^{i0})}{\left\| \sum_{j \in N_i} (z_{1d}^{ij} - \sigma^{ij}) + a_{i0}(z_{1d}^{i0} - \sigma^{i0}) \right\| + \epsilon}$$

as explained in Section 3.2.

Using the control signal defined by equation (4.2.12) for a network of heterogeneous autonomous underwater vehicles in which LOE actuator faults occurred in one or some of vehicles leads to stability of the system which indicates that $z_1^{(i)}$ s and $z_2^{(i)}$ s are bounded for $1 \leq i \leq n$ and $z_1^{(i)}$ s are within an arbitrarily small neighborhood around the desired trajectory. The stability analysis of the system under the fault-tolerant semi-decentralized control scheme is given as follows.

4.2.3.1 Stability Analysis

Theorem 4.1. Suppose that Assumptions 3.1, 3.2, and 3.3 given in section 3.5 are valid, and based on Theorem 3.1. $|z_{1d}^{(i)} - z_{1d}^{(j)}| \rightarrow \sigma^{ij}$ as $t \rightarrow \infty$. By using controller given by equation (4.2.12) for a group of agents subjected to LOE faults with dynamics given by equation (4.1.1), with proper choices of $\lambda_1^{(i)}$, $\lambda_2^{(i)}$, and ε , the group of agents is stable and all $z_1^{(i)}$ and $z_2^{(i)}$ are bounded.

Proof. Without loss of generality, assume that all $\lambda_1^{(i)}$ s are equal, and the same for $\lambda_2^{(i)}$ s, $\delta_1^{(i)}$ s, $\varsigma^{(i)}$ s, and $\delta_2^{(i)}$ s. Considering $\underline{S}_1 \triangleq [S_1^{T(1)}, \dots, S_1^{T(n)}]^T$ and $\underline{S}_2 \triangleq [S_2^{T(1)}, \dots, S_2^{T(n)}]^T$ as the column stack vectors of the first and second error surfaces with $S_1^{(i)}, S_2^{(i)} \in \mathbb{R}^3$, to show the

stability of the closed-loop error dynamics of the group of agents given by

$$\begin{cases} \dot{\underline{z}}_1 = \underline{z}_2 - \dot{\underline{z}}_{1d} \\ \dot{\underline{z}}_2 = (M_\eta)_{con}^{-1} (\underline{\tau}_H - (C_\eta)_{con}(\underline{z}_1, \underline{z}_2)\underline{z}_2 - (D_\eta)_{con}(\underline{z}_1)\underline{z}_2) \\ \quad + \underline{\Delta f}_\eta(\underline{z}_1) - \dot{\underline{z}}_2 + (M_\eta)_{con}^{-1} (-\underline{F}\underline{\tau}_H + (I - \underline{F})\underline{\tau}_C) \end{cases} \quad (4.2.15)$$

where $(M_\eta)_{con}$, $(C_\eta)_{con}$, and $(D_\eta)_{con}$ are defined in Section 3.3, $\underline{z}_1 \triangleq [z_1^{T(1)}, \dots, z_1^{T(n)}]^T$, $\underline{z}_2 \triangleq [z_2^{T(1)}, \dots, z_2^{T(n)}]^T$, $\underline{\tau}_H \triangleq [\tau_H^{T(1)}, \dots, \tau_H^{T(n)}]^T$, $\underline{\tau}_C \triangleq [\tau_C^{T(1)}, \dots, \tau_C^{T(n)}]^T$, $\underline{\Delta f}_\eta(\underline{z}_1) \triangleq [\Delta f_\eta(z_1)^{T(1)}, \dots, \Delta f_\eta(z_1)^{T(n)}]^T$, $\underline{F} \triangleq [F^{T(1)}, \dots, F^{T(n)}]^T$, and $\dot{\underline{z}}_2 \triangleq [\dot{z}_2^{T(1)}, \dots, \dot{z}_2^{T(n)}]$ with $z_1^{(i)}, z_2^{(i)}, \tau_H^{(i)}, \tau_C^{(i)}, \Delta f_\eta(z_1)^{(i)}, F^{(i)}, \dot{z}_2^{(i)} \in \mathbb{R}^3$, using the fault-tolerant semi-decentralized control scheme for the multi-agent team, the candidate Lyapunov function is chosen as

$$V = \frac{1}{2} (\underline{S}_1^T \underline{S}_1 + \underline{S}_2^T \underline{S}_2) \quad (4.2.16)$$

Taking derivative of V along the trajectories of equation (4.2.15) is given by

$$\begin{aligned} \dot{V} = & \underline{S}_1^T (\underline{z}_2 - \dot{\underline{z}}_{1d}) + \underline{S}_2^T [(M_\eta)_{con}^{-1} (\underline{\tau}_H - (C_\eta)_{con} \underline{z}_2 - (D_\eta)_{con} \underline{z}_2) \\ & + \underline{\Delta f}_\eta(\underline{z}_1) - \dot{\underline{z}}_2 + (M_\eta)_{con}^{-1} (-\underline{F}\underline{\tau}_H + (I - \underline{F})\underline{\tau}_C)] \end{aligned} \quad (4.2.17)$$

Applying $\dot{\underline{z}}_2 = -\lambda_1 (\underline{z}_2 - \dot{\underline{z}}_{1d}) + \dot{\underline{z}}_{1d}$ and equation (4.2.13) leads to

$$\begin{aligned} \dot{V} = & -\underline{S}_1^T (\lambda_1 \otimes I_n) \underline{S}_1 + \underline{S}_1^T \underline{S}_2 \\ & - \underline{S}_2^T (\lambda_2 \otimes I_n) \underline{S}_2 + \underline{S}_2^T \underline{\Delta f}_\eta(\underline{z}_1) - \frac{\underline{S}_2^T (\underline{S}_2 \circ \underline{P})}{2\varepsilon} \\ & + \underline{S}_2^T (M_\eta)_{con}^{-1} (-\underline{F}\underline{\tau}_H + (I - \underline{F})\underline{\tau}_C) \end{aligned} \quad (4.2.18)$$

Same as Section 2.3.3, using Young's inequality, it can be shown that

$$\begin{aligned} \dot{V} \leq & -\underline{S}_1^T (\lambda_1 \otimes I_n) \underline{S}_1 - \underline{S}_2^T (\lambda_2 \otimes I_n) \underline{S}_2 + \underline{S}_1 \underline{S}_2 + 3n \frac{\varepsilon}{2} \\ & + \underline{S}_2^T (M_\eta)_{con}^{-1} (-\underline{F}\underline{\tau}_H + (I - \underline{F})\underline{\tau}_C) \end{aligned} \quad (4.2.19)$$

Using equation (4.2.3) yields to

$$\begin{aligned}
\dot{V} &\leq -\|\underline{S}_1\|^2(\lambda_1 \otimes I_n) - \|\underline{S}_2\|^2(\lambda_2 \otimes I_n) + \|\underline{S}_1\|\|\underline{S}_2\| - \underline{S}_2^T (M_\eta)_{con}^{-1} \underline{F} \tau_H \\
&\quad - \underline{S}_2^T (M_\eta)_{con}^{-1} (I - \underline{F}) \frac{\underline{S}_2}{\|\underline{S}_2\| + \delta_2} \left(\|\tau_H\| + \frac{\delta_1 \|\tau_H\|}{\varsigma} \right) \\
&\leq -\|\underline{S}_1\|^2(\lambda_1 \otimes I_n) - \|\underline{S}_2\|^2(\lambda_2 \otimes I_n) + \|\underline{S}_1\|\|\underline{S}_2\| - \|\underline{S}_2^T\| \|(M_\eta)_{con}^{-1} \underline{F}\| \|\tau_H\| \\
&\quad - \frac{(I - \underline{F}) \left(1 + \frac{\delta_1}{1 - \max\{\hat{f}_j\}} \right)}{\|\underline{S}_2\| + \delta_2} \|\tau_H\| \|\underline{S}_2^T (M_\eta)_{con}^{-1} \underline{S}_2\|
\end{aligned} \tag{4.2.20}$$

Since $I - F$, $1 - \max\{\hat{f}_j^{(i)}\}$, and δ_1 are positive values, proper choices of λ_1 , λ_2 , and ε lead to

$\dot{V} \leq 0$ in

$$\begin{aligned}
\mathbb{S} = \{ &(S_1, S_2) \mid -\|\underline{S}_1\|^2(\lambda_1 \otimes I_n) - \|\underline{S}_2\|^2(\lambda_2 \otimes I_n) + \|\underline{S}_1\|\|\underline{S}_2\| - \|\underline{S}_2^T\| \|(M_\eta)_{con}^{-1} \underline{F}\| \|\tau_H\| \\
&- \frac{(I - \underline{F}) \left(1 + \frac{\delta_1}{1 - \max\{\hat{f}_j\}} \right)}{\|\underline{S}_2\| + \delta_2} \|\tau_H\| \|\underline{S}_2^T (M_\eta)_{con}^{-1} \underline{S}_2\| \leq 0 \}
\end{aligned}$$

which shows locally stability of the system and indicates that all $z_1^{(i)}$ and $z_2^{(i)}$ are bounded. ■

4.3 Simulation Results

In this section, to evaluate the performance of the proposed DSC-based semi-decentralized fault-tolerant control scheme to control the group of autonomous underwater vehicles, several simulations are presented. Before applying the recovery part of the proposed controller, the performance of a group of AUVs under faulty situation has been simulated only with considering τ_H , i.e. there is no compensation control to compensate injected fault. After that, to see the effect of τ_C , several scenarios have been defined. In the first part, the proposed semi-decentralized scheme is compared to the centralized and then the decentralized schemes. Furthermore, to show the capability of the proposed control scheme to reach the objectives which are formation

keeping and path-tracking when there are LOE faults in one or more agents in the multi-agent group under different scenarios with various conditions, several scenarios are defined and the results are analyzed.

In all scenarios defined in this section, two sets of model parameters for the group of heterogeneous autonomous underwater vehicles are considered as presented in Section 3.7. The goal for each agent in all scenarios is to keep the desired formation while tracking the reference trajectory which is defined as

$$\eta_{ref} = \left[15 \sin\left(\frac{t}{8}\right) \left(\frac{t}{3}\right) 0 \right]^T$$

In Sections 4.3.1 and 4.3.2, network topology, desired relative distances, initial conditions, model uncertainties, and parameters of τ_H are considered the same as Section 3.7. In addition, the control parameters of compensation control input are considered as $\delta_1 = 4$ and $\delta_2 = 1.5$. Moreover, in order to show the effectiveness of the proposed semi-decentralized scheme in real situation, in these sections, the detection time delay of five second is considered in simulation tests, which means the compensation control input will be activated five second after the time that fault is happened in the system, i.e. $t_c = 25 \text{ sec}$.

Because of the inability of the decentralized scheme to overcome the isolation errors, this condition is not considered in comparative sections. However, to analyze the performance of the semi-decentralized scheme, this condition will be considered later.

4.3.1 Semi-decentralized DSC-based Scheme in Faulty Situation Without Compensation Control

One of the advantages of dynamic surface control technique is its robustness to faults with low severities which can be quite compensated using only τ_H . Also, for the faults with middle

range severity, as will be shown later, it can keep the group stable in which the trajectories are just bounded. In other words, the τ_H controller acts like a passive fault-tolerant control system that can only compensate a limited range of predetermined faults. Therefore, the proposed cooperative active fault-tolerant scheme can be used to compensate the severe faults. This explains our choices of high fault severities in all defined scenarios. The main aim of this part is to show that why fault reconfiguration has been selected instead of a simpler solution which is fault accommodation.

Table 4.1: Cases of faulty situation without compensation control

Case	Controller parameters	Faulty agents
1	$\lambda_1 = 1.8 , \lambda_2 = 2.9$	agent 1: 40% agent 3: 50% agent 5: 60%
2	$\lambda_1 = 4 , \lambda_2 = 6$	agent 1: 40% agent 3: 50% agent 5: 60%
3	$\lambda_1 = 12 , \lambda_2 = 20$	agent 1: 40% agent 3: 50% agent 5: 60%
4	$\lambda_1 = 1.8 , \lambda_2 = 2.9$	agent 1: 75% agent 3: 80% agent 5: 85%
5	$\lambda_1 = 5 , \lambda_2 = 9$	agent 1: 75% agent 3: 80% agent 5: 85%
6	$\lambda_1 = 12 , \lambda_2 = 20$	agent 1: 75% agent 3: 80% agent 5: 85%

To show the robustness of this technique, several simulations have been conducted and six cases as indicated in Table 4.1 are presented as samples. In these cases, for two sets of fault severities, three sets of control parameters are considered. The first set of fault severities is considered in middle range, while the second set is considered in high range severities. In

these simulations, desired relative distances, model parameters and uncertainties, reference trajectory, and initial conditions for each agent are considered as in Section 3.7.3.

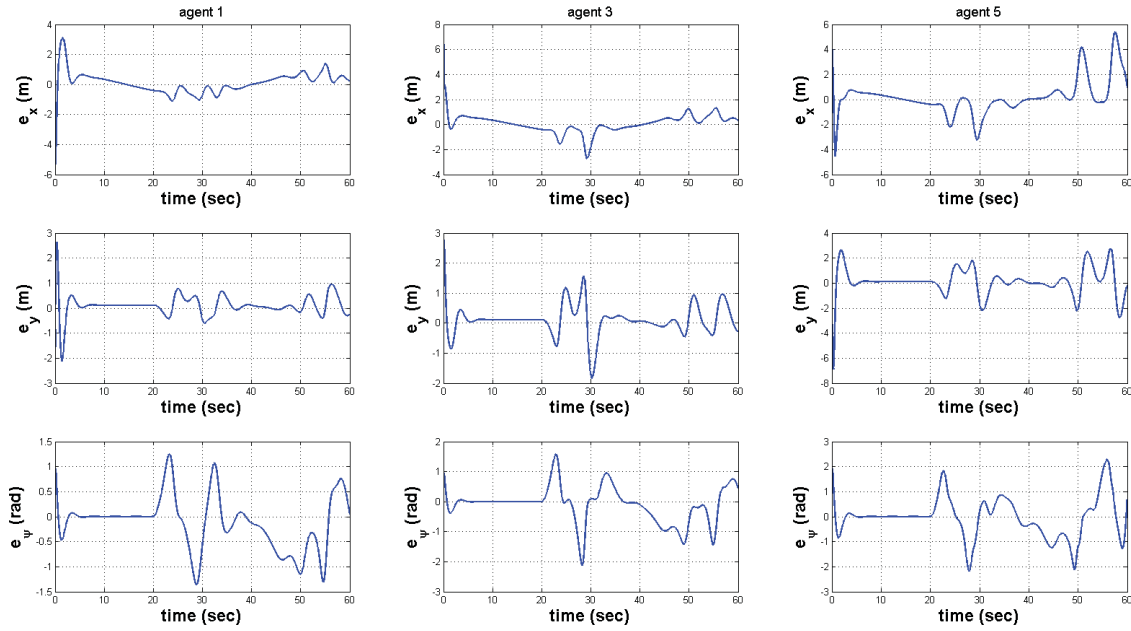


Figure 4.2: Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 1.

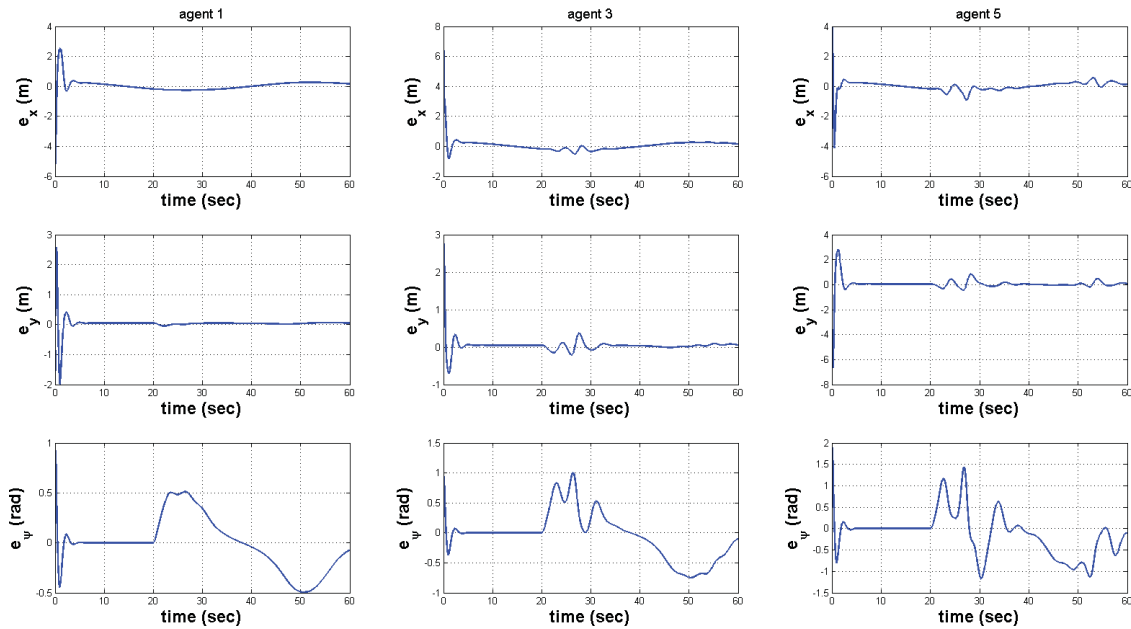


Figure 4.3: Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 2.

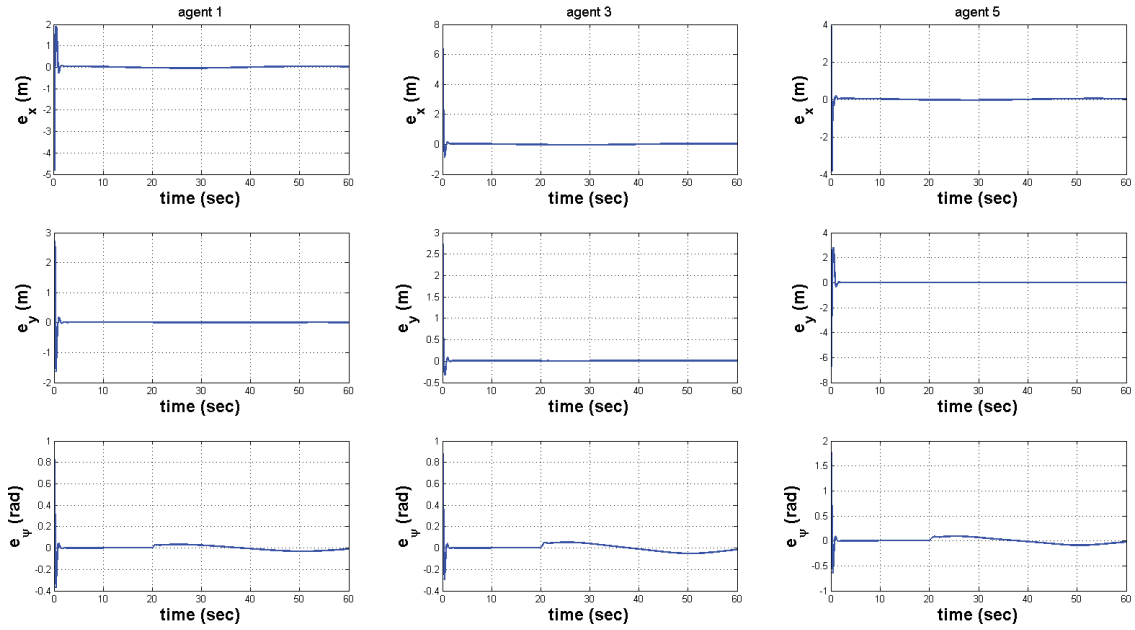


Figure 4.4: Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 3.

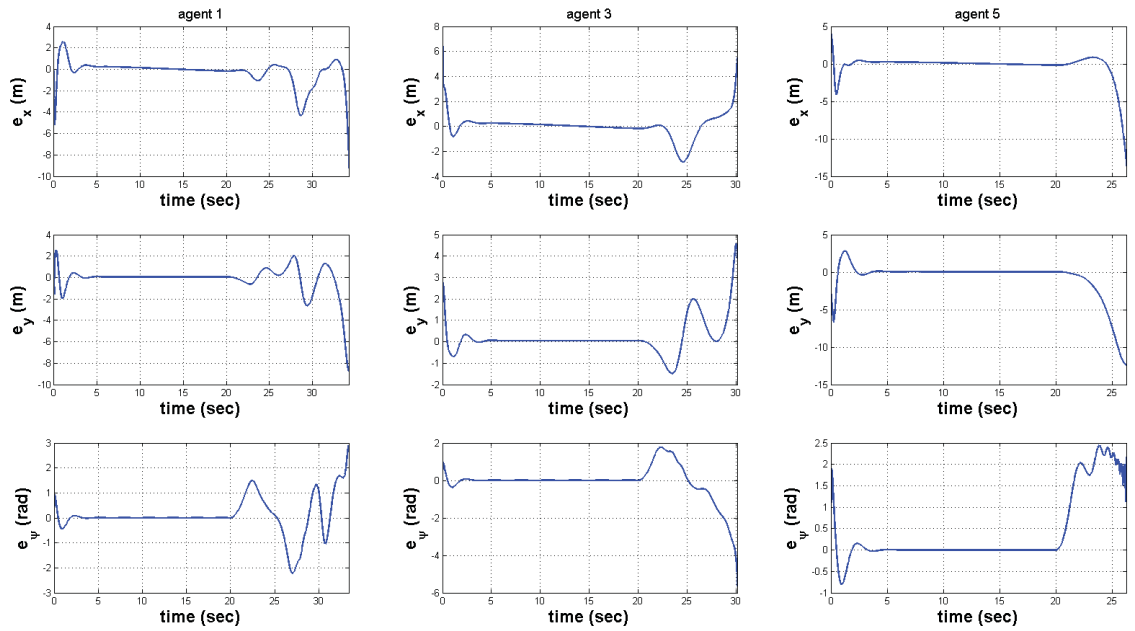


Figure 4.5: Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 4.

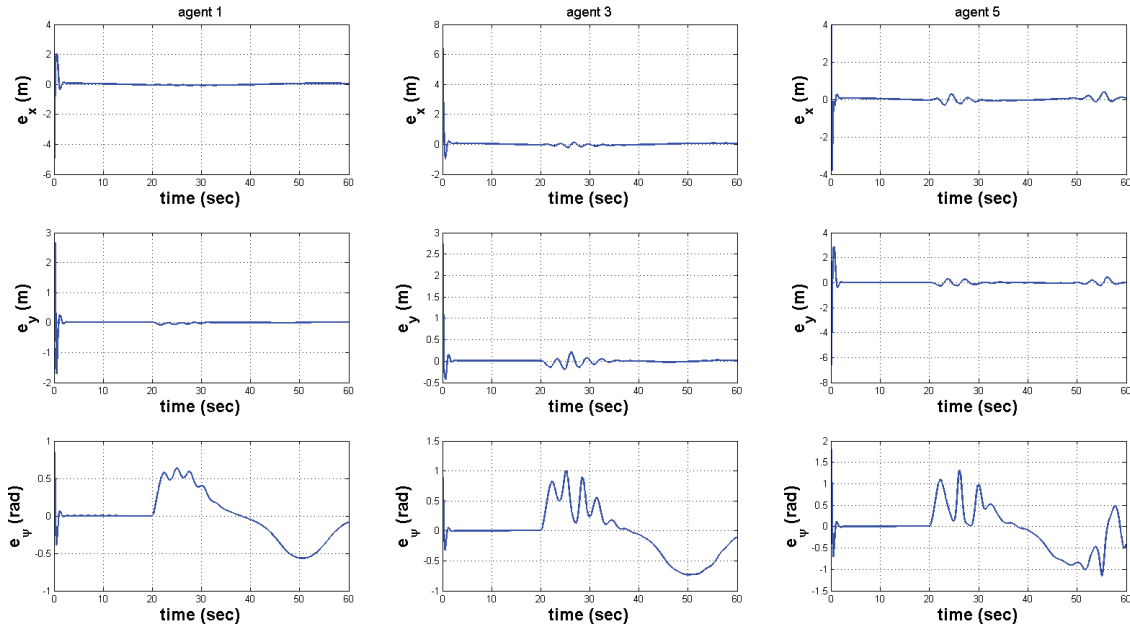


Figure 4.6: Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 5.

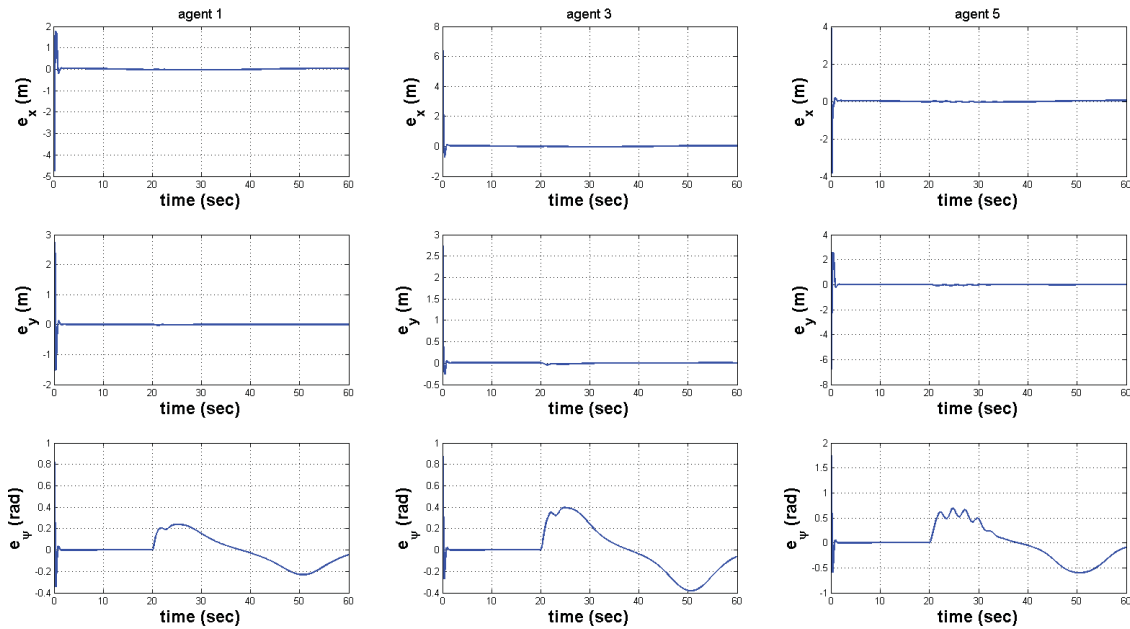


Figure 4.7: Tracking error signals of agent #1, #3, and #5 for Section 4.3.1, case 6.

In Figures 4.2 to 4.7, the simulation results of the error trajectories of the semi-decentralized scheme for faulty case without any compensation control are presented. From these simulations,

it can be seen that for middle range severity faults, large values of λ_1 and λ_2 lead to low steady state errors as in Case 3. For low values of λ_1 and λ_2 as in Case 1, although the responses are not satisfactory, at least it do not lead to instability. For the high severity faults, low values of λ_1 and λ_2 , as in Case 4, can lead to instability of agents. However, in case of choosing λ_1 and λ_2 large enough, although the designed healthy control input τ_H cannot guarantee the low steady state errors in trajectories of the faulty agents in the team, the trajectories are still bounded and the group is stable. Although by increasing the healthy controller parameters the results are improved, this solution is not an appropriate way to deal with faults in the system since the affordability of control gains is not satisfied. Therefore, in this thesis the fault reconfiguration method has been selected instead of the fault accommodation approach.

4.3.2 Quantitative Comparison of DSC-based Fault-tolerant Centralized, Decentralized, and Semi-decentralized Schemes

In this section, the proposed DSC-based fault-tolerant semi-decentralized scheme is compared to the fault-tolerant decentralized and centralized schemes to evaluate the efficiency of this scheme. The initializations and assumptions of this section are given in Section 4.3. The reference trajectory is assumed to be

$$\eta_{ref} = \left[15 \sin\left(\frac{t}{8}\right) \left(\frac{t}{3}\right) 0 \right]^T$$

Also, it is assumed that the LOE fault occurs at $t_f = 20 \text{ sec}$ in agent #1 with severity of 60%, in agent #4 with severity of 65%, and in agent #5 with severity of 60%. In these simulation tests, the detection time delay of five second is considered.

In Figures 4.8 to 4.10 the tracking errors of each agent for all schemes are presented. In Figures 4.11 to 4.13 the control input signals of the all cooperative schemes are presented. for

all agents, the control effort cost In addition, the response characteristics of each agent for all schemes are summarized in Tables 4.2 to 4.2.

Based on these results, the maximum errors of all position states of each agent after the time that faults are injected to the system are mostly less in the centralized scheme in comparison to the decentralized and semi-decentralized schemes. These errors in the decentralized and semi-decentralized schemes are partly closed to each other. However, these are lower in the semi-decentralized scheme. Also, after activation of τ_C , the time that it takes for each faulty agent to recover is less in the centralized scheme in comparison to the semi-decentralized scheme, and it is lower in the semi-decentralized scheme in comparison to the decentralized scheme. For the faulty agents, in the decentralized scheme, the maximum control efforts after t_f are higher with more oscillation in comparison to other schemes. However, in fault-free agents, the centralized scheme has the highest maximum control efforts after t_f .

The team level comparison of all schemes are presented in Table 4.8. For team level analysis, the root mean square of all characteristics given in Tables 4.2 to 4.7 are presented. In this table, the results shows that the centralized scheme has lowest RMS in maximum error and for the time that it takes to $e_{ss} \rightarrow 0$ after activation of τ_C . After that, the semi-decentralized scheme has lower RMS in maximum error and for the time that it takes to $e_{ss} \rightarrow 0$ after activation of τ_C in comparison to the decentralized scheme. The team level maximum control effort after the time that faults occurred in the team has the highest value in the decentralized scheme, followed by semi-decentralized and then the centralized scheme.

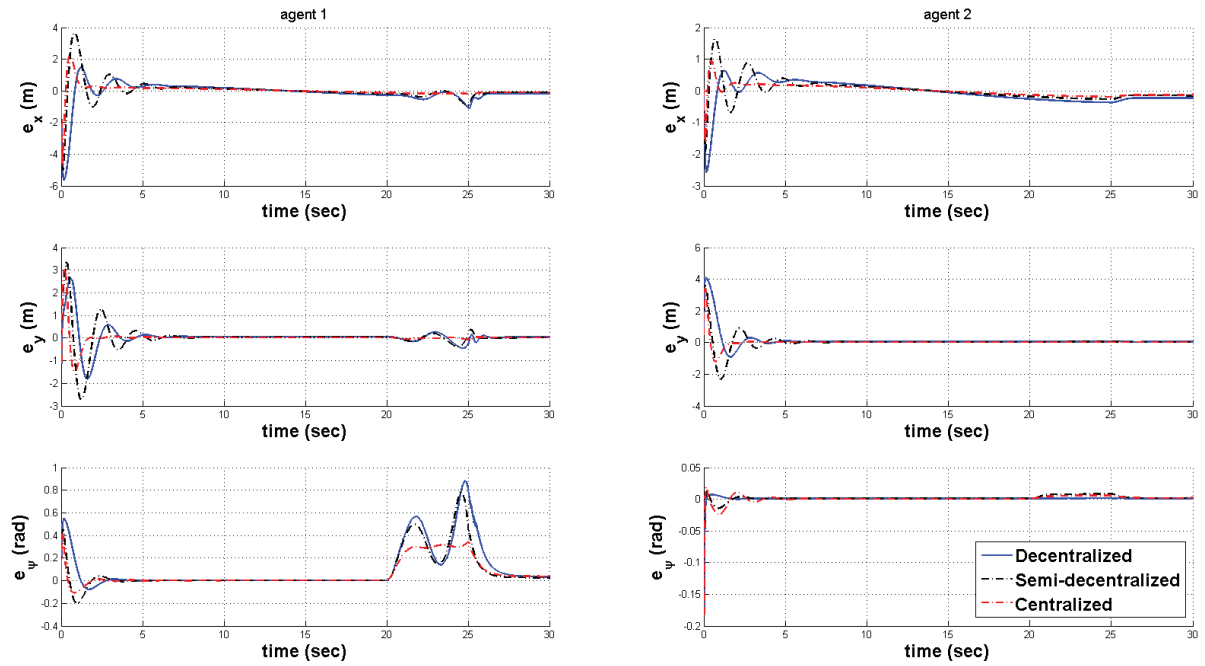


Figure 4.8: Tracking error signals of agents #1 and #2 in faulty situation for all fault-tolerant DSC-based cooperative schemes.

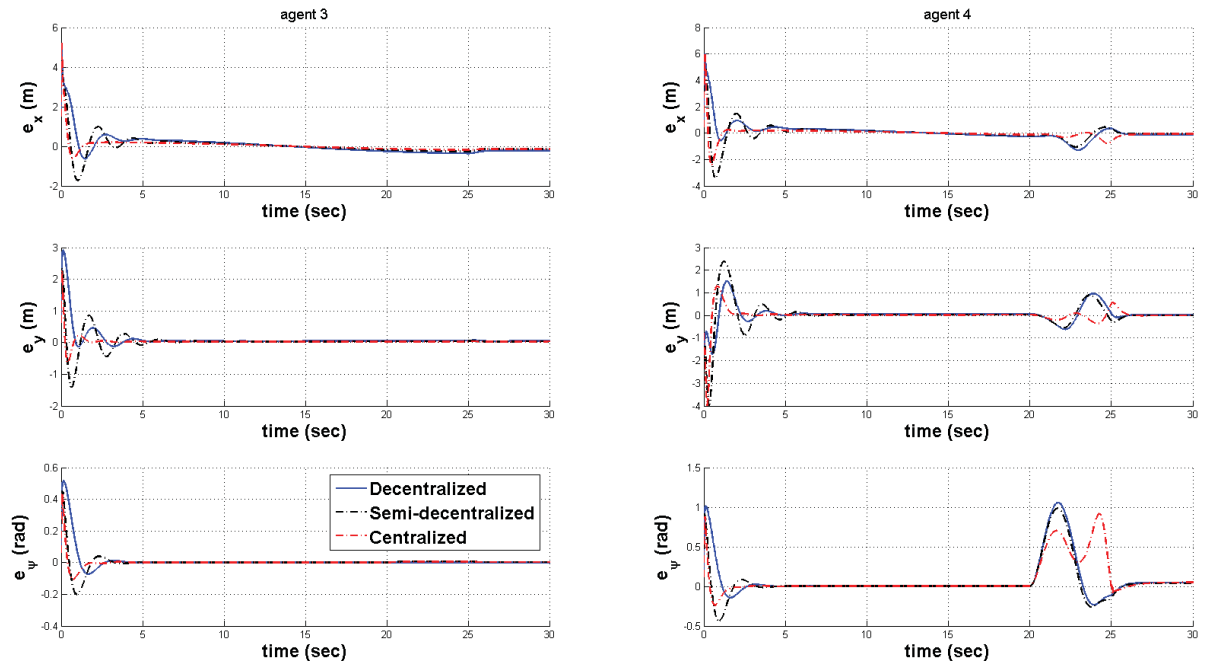


Figure 4.9: Tracking error signals of agents #3 and #4 in faulty situation for all fault-tolerant DSC-based cooperative schemes.

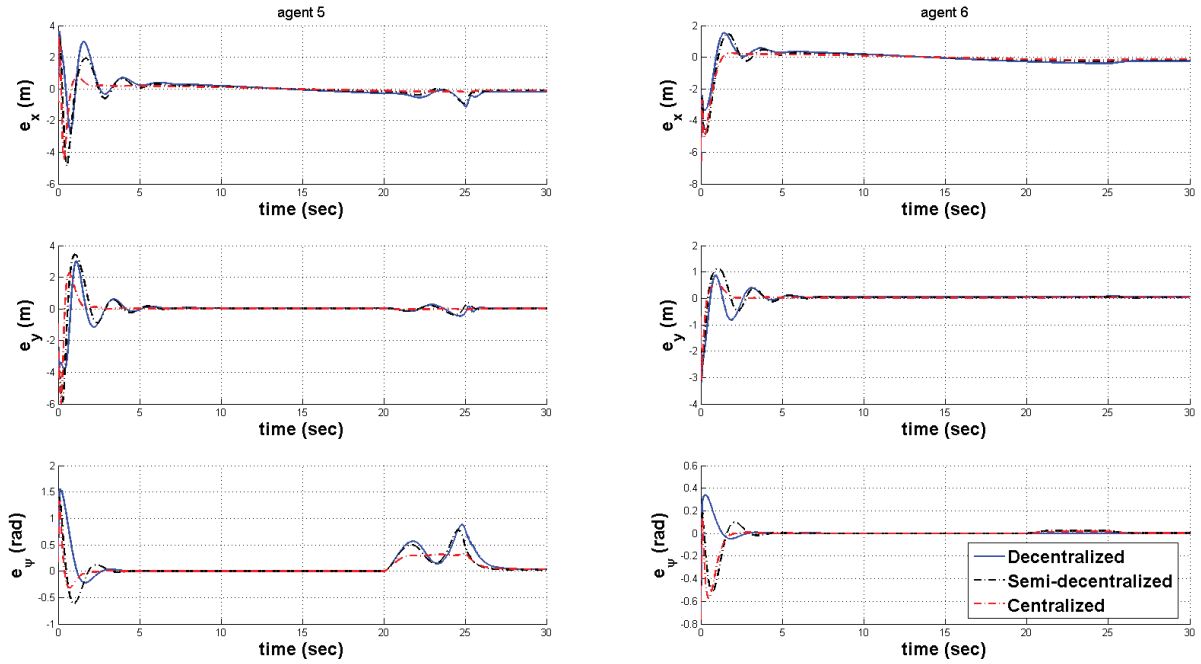


Figure 4.10: Tracking error signals of agents #5 and #6 in faulty situation for all fault-tolerant DSC-based cooperative schemes.

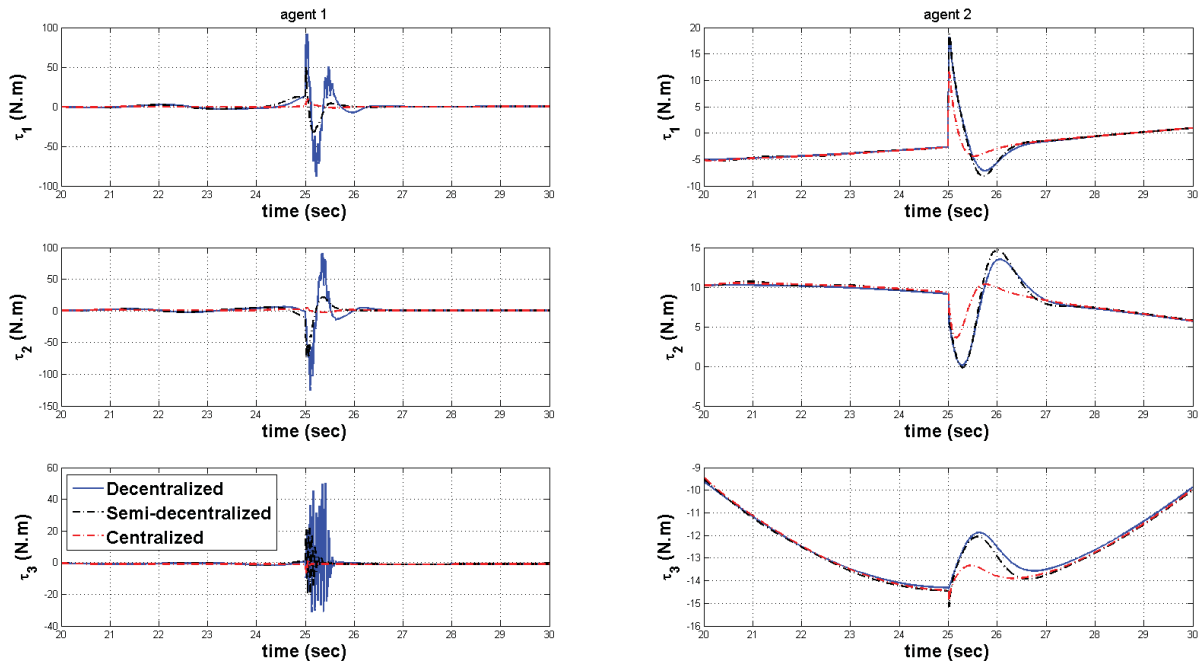


Figure 4.11: Control input signals of agents #1 and #2 in faulty situation for all fault-tolerant DSC-based cooperative schemes.

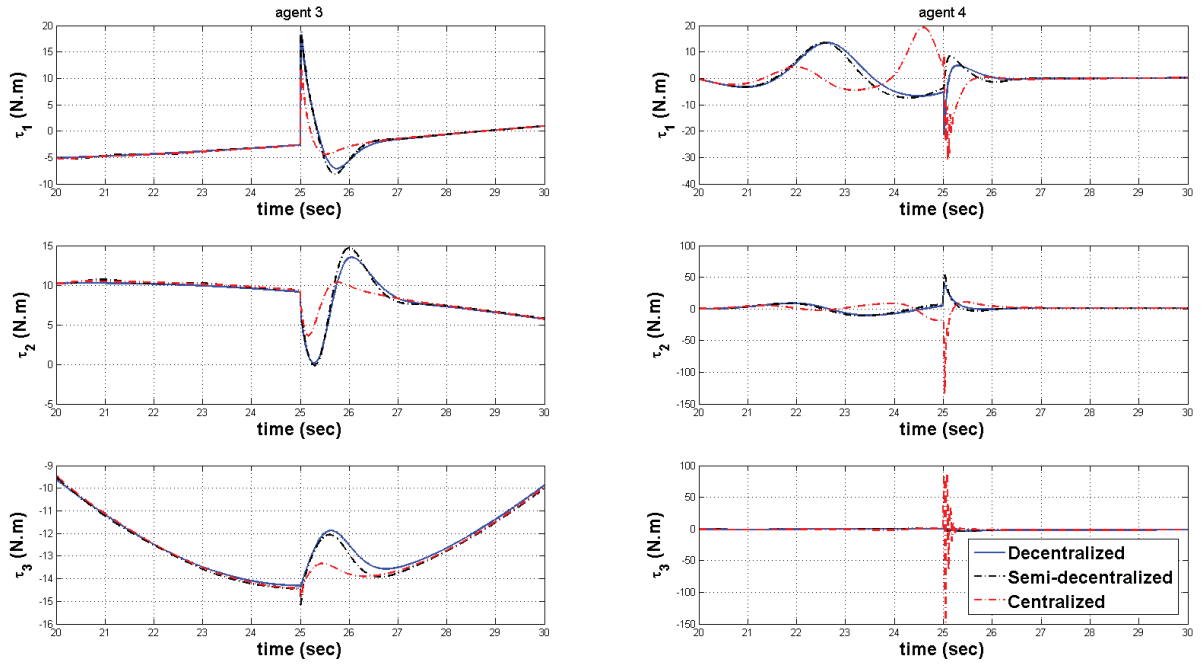


Figure 4.12: Control input signals of agents #3 and #4 in faulty situation for all fault-tolerant DSC-based cooperative schemes.

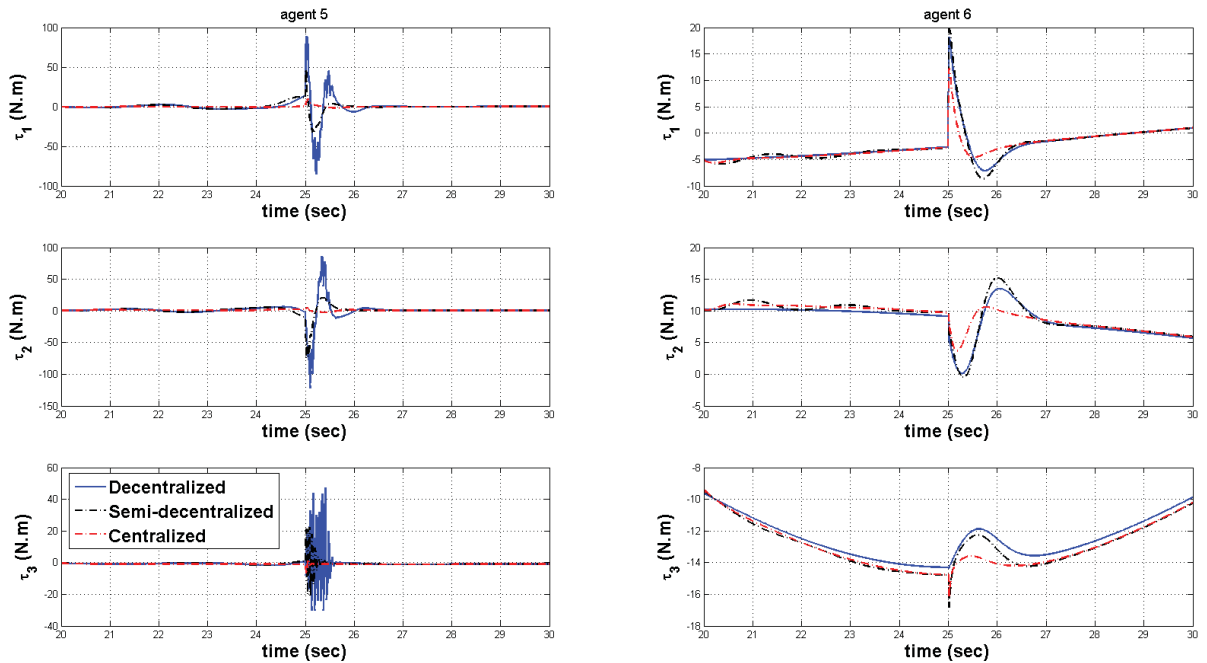


Figure 4.13: Control input signals of agents #5 and #6 in faulty situation for all fault-tolerant DSC-based cooperative schemes.

Table 4.2: Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #1

		Semi-decentralized	Centralized	Decentralized
x	Maximum error after t_f (m)	0.9271	0.2037	1.1167
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.15	0.85	1.17
	Max. Abs. control effort after t_f ($N.m$)	49.873	9.5238	91.442
y	Maximum error after t_f (m)	0.3891	0.0697	0.4591
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.66	0.39	1.32
	Max. Abs. control effort after t_f ($N.m$)	74.1	4.645	125.08
ψ	Maximum error after t_f (m)	0.7794	0.3458	0.8806
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.89	0.97	2.47
	Max. Abs. control effort after t_f ($N.m$)	21.96	7.2115	50.095

Table 4.3: Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #2

		Semi-decentralized	Centralized	Decentralized
x	Maximum error after t_f (m)	0.2647	0.1897	0.3692
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.88	0.54	1.28
	Max. Abs. control effort after t_f ($N.m$)	18.745	11.546	18.263
y	Maximum error after t_f (m)	0.0511	0.0337	0.06939
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0	0	0
	Max. Abs. control effort after t_f ($N.m$)	14.693	10.515	13.51
ψ	Maximum error after t_f (m)	0.0084	0.00608	0.0014
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.66	0.48	0
	Max. Abs. control effort after t_f ($N.m$)	15.13	14.873	14.364

Table 4.4: Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #3

		Semi-decentralized	Centralized	Decentralized
x	Maximum error after t_f (m)	0.2647	0.1897	0.3663
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.8	0.72	1.15
	Max. Abs. control effort after t_f ($N.m$)	18.746	11.546	18.263
y	Maximum error after t_f (m)	0.051	0.0337	0.0688
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0	0	0
	Max. Abs. control effort after t_f ($N.m$)	14.692	10.515	13.51
ψ	Maximum error after t_f (m)	0.0084	0.00608	0.00141
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.42	0.28	0
	Max. Abs. control effort after t_f ($N.m$)	15.13	14.873	14.364

Table 4.5: Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #4

		Semi-decentralized	Centralized	Decentralized
x	Maximum error after t_f (m)	1.0608	0.7854	1.2981
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.52	0.92	0.92
	Max. Abs. control effort after t_f ($N.m$)	13.371	30.777	21.405
y	Maximum error after t_f (m)	0.8912	0.5712	0.9598
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.94	1.11	0.94
	Max. Abs. control effort after t_f ($N.m$)	54.156	133.36	39.413
ψ	Maximum error after t_f (m)	0.9855	0.91543	1.0571
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.16	1.77	1.16
	Max. Abs. control effort after t_f ($N.m$)	3.8559	146.54	3.3996

Table 4.6: Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #5

		Semi-decentralized	Centralized	Decentralized
x	Maximum error after t_f (m)	0.9236	0.2047	1.1324
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.48	0.82	2.24
	Max. Abs. control effort after t_f ($N.m$)	47.6	9.5545	88.662
y	Maximum error after t_f (m)	0.3875	0.07	0.4562
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.65	0.33	1.42
	Max. Abs. control effort after t_f ($N.m$)	74.581	4.572	122.14
ψ	Maximum error after t_f (m)	0.7779	0.3469	0.87974
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.87	0.97	1.59
	Max. Abs. control effort after t_f ($N.m$)	22.403	7.192	47.271

Table 4.7: Quantitative analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for agent #6

		Semi-decentralized	Centralized	Decentralized
x	Maximum error after t_f (m)	0.2639	0.1895	0.39863
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.86	0.73	1.22
	Max. Abs. control effort after t_f ($N.m$)	19.848	12.283	18.263
y	Maximum error after t_f (m)	0.0518	0.03329	0.07462
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0	0	0
	Max. Abs. control effort after t_f ($N.m$)	15.188	11.064	13.51
ψ	Maximum error after t_f (m)	0.0244	0.0175	0.0014
	Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.32	0.17	0
	Max. Abs. control effort after t_f ($N.m$)	16.845	16.123	14.364

Table 4.8: Team level RMS analysis of all fault-tolerant DSC-based cooperative schemes in faulty situation for Section 4.3.2

	Semi-decentralized	Decentralized	Centralized
RMS maximum error after t_f (m)	0.5933	0.6938	0.4058
RMS time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.7919	1.2087	0.7612
RMS max. Abs. control effort after t_f ($N.m$)	48.303	55.638	35.291

4.3.3 Simulation Scenarios for Semi-decentralized Fault-tolerant Scheme

To evaluate the capability of the proposed semi-decentralized fault-tolerant scheme in different situations, in this section, several scenarios are simulated considering various conditions and possibilities in group of multi-agents followed by different conditions in the FDI module.

4.3.3.1 Scenario 4.1: Number of Faulty Agents

In this scenario, we aim to analyze the ability of the proposed fault-tolerant semi-decentralized scheme to fulfill the objectives which are formation keeping and path-tracking in the different network topologies where there are different numbers of faulty vehicles among the group. To this end, six different cases are defined as in Table 4.9.

In this scenario, for two different networks, three situations are considered. In the first situation, randomly some vehicles are assumed to be faulty. In the second situation, all of the vehicles are faulty except the ones that are directly connected to the virtual leader. And finally, in the third situation, only the vehicles which are directly connected to the virtual leader are assumed to be faulty and the rest of the vehicles are fault-free.

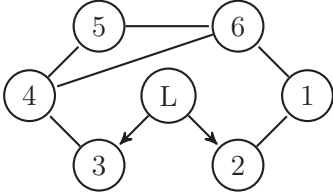
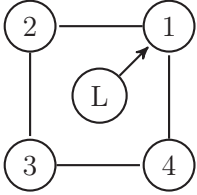
From Table 4.9 there are some differences between graph G_6 and G_7 . In G_6 , there are six agents and two of them are directly connected to the virtual leader and they do not receive information from other vehicles. In G_7 , there are four agents and there is only one agent that is directly connected to the virtual leader and it receives information from other vehicles. In all cases of this scenario the LOE fault occurs in $t_f = 20 \text{ sec}$ and the fault detection time delay of five second is considered in simulation tests, i.e. $t_c = 25 \text{ sec}$. The control parameters of compensation control input are considered as $\delta_1 = 4$ and $\delta_2 = 1.5$ which are chosen based on a trade-off between performance and affordability of control gains.

To analyze the performance of the proposed semi-decentralized fault-tolerant control scheme, the error signals of the semi-decentralized and centralized schemes for each case are presented in Figures 4.14 to 4.25. In addition, in order to be able to evaluate the performance of the proposed fault-tolerant semi-decentralized scheme by comparing it to the centralized fault-tolerant scheme, in Tables 4.12 to 4.17, the maximum errors after injection of fault and the time elapsed to obtain almost zero steady state error after activation of compensation control are given for all agents in each case for both semi-decentralized and centralized schemes. It is worth mentioning that the time elapsed to $e_{ss} \approx 0$ after activation of τ_C are given only for faulty agents. From these results it can be seen that:

- For all agents, the maximum errors after the time that fault is injected to the system are quite lower in the centralized scheme in comparison to the proposed semi-decentralized fault-tolerant scheme.
- The time elapsed to obtain $e_{ss} \approx 0$ after the time that τ_C is activated varies from case to case. However, the semi-decentralized scheme mostly had better performance on damping the errors caused by injected fault specially in yaw.

- The proposed fault-tolerant semi-decentralized scheme can recover multi-agent systems in faulty situations no matter how many agents and which agents are faulty.
- The performance of the proposed fault-tolerant semi-decentralized DSC-based scheme does not depend on the topology of the network of agents.
- When faulty agents are the ones which are directly connected to the virtual leader, their effect on fault-free agents is more than the case that all vehicles are faulty except the ones directly connected to the virtual leader.
- The y position has less influence from LOE fault in comparison to x and ψ states.

Table 4.9: Cases of scenario 4.1

Graph	Case	Faulty agents
G_6 : 	1	agent 1: 85% agent 3: 80% agent 6: 75%
	2	agent 1: 85% agent 4: 70% agent 5: 80% agent 6: 75%
	3	agent 2: 80% agent 3: 85%
G_7 : 	4	agent 1: 85% agent 3: 80%
	5	agent 2: 85% agent 3: 80% agent 4: 70%
	6	agent 1: 80%

For the team level analysis, the root mean square of the maximum error after t_f and the time to $e_{ss} \rightarrow 0$ after activation of τ_c for all cases are respectively presented in Tables 4.10 and 4.11. Form the results given in this table, it can be seen that the semi-decentralized scheme

has a better performance for the time that it takes to $e_{ss} \rightarrow 0$ after activation of τ_c in team level in comparison to the centralized scheme. However, the team level PRMS maximum error of the centralized scheme is lower than the semi-decentralized scheme.

Table 4.10: Team level RMS analysis of maximum error after t_f (m) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.1

	RMS maximum error	
	Semi-decentralized	Centralized
Case 1	1.84713	0.88988
Case 2	1.90726	0.94245
Case 3	1.74442	0.81087
Case 4	2.12467	0.98668
Case 5	2.20716	1.05649
Case 6	1.06889	0.56608

Table 4.11: Team level RMS analysis of time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.1

	RMS time to $e_{ss} \rightarrow 0$	
	Semi-decentralized	Centralized
Case 1	1.50946	2.48253
Case 2	1.71574	2.32858
Case 3	1.38457	1.88866
Case 4	1.54955	1.59223
Case 5	1.44669	1.53809
Case 6	1.38161	1.23046

Table 4.12: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes

for scenario 4.1 case 1

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	5.3104	1.8819
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.54	1.85
	y	Maximum error after t_f (m)	2.9473	1.2321
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.65	1.29
	ψ	Maximum error after t_f (m)	1.9339	1.6995
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.16	1.38
agent 2	x	Maximum error after t_f (m)	0.2633	0.235
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0479	0.0431
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0165	0.0181
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 3	x	Maximum error after t_f (m)	2.8806	0.9159
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.08	0.26
	y	Maximum error after t_f (m)	1.484	0.6876
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.96	0.57
	ψ	Maximum error after t_f (m)	1.7415	1.54
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.65	2.69
agent 4	x	Maximum error after t_f (m)	0.2622	0.2351
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0475	0.0428
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0357	0.0109
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 5	x	Maximum error after t_f (m)	0.2622	0.2351
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0477	0.0428
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0364	0.0313
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 6	x	Maximum error after t_f (m)	1.6548	0.5658
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.89	0.42
	y	Maximum error after t_f (m)	1.3019	0.4544
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.18	2.82
	ψ	Maximum error after t_f (m)	1.6138	1.3869
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.46	5.72

Table 4.13: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes

for scenario 4.1 case 2

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	5.3104	1.8819
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.54	1.85
	y	Maximum error after t_f (m)	2.9473	1.2321
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.78	1.75
	ψ	Maximum error after t_f (m)	1.9339	1.6995
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.22	1.43
agent 2	x	Maximum error after t_f (m)	0.2621	0.2351
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0478	0.0431
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0359	0.0179
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 3	x	Maximum error after t_f (m)	0.2622	0.2351
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0476	0.0428
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.035	0.0027
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 4	x	Maximum error after t_f (m)	1.0804	0.3995
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.87	0.69
	y	Maximum error after t_f (m)	0.887	0.3267
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.96	2.77
	ψ	Maximum error after t_f (m)	1.4717	1.2308
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.22	3.91
agent 5	x	Maximum error after t_f (m)	2.882	0.9172
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.82	1.23
	y	Maximum error after t_f (m)	1.4846	0.6884
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.96	0.54
	ψ	Maximum error after t_f (m)	1.7433	1.5424
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.62	3.96
agent 6	x	Maximum error after t_f (m)	1.6548	0.5658
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.27	0.83
	y	Maximum error after t_f (m)	1.3019	0.4544
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.92	1.57
	ψ	Maximum error after t_f (m)	1.6138	1.3869
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.82	3.53

Table 4.14: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes

for scenario 4.1 case 3

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	0.2621	0.235
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0477	0.043
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0394	0.052
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 2	x	Maximum error after t_f (m)	2.8814	0.9155
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.1	0.28
	y	Maximum error after t_f (m)	1.4843	0.6876
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.96	0.54
	ψ	Maximum error after t_f (m)	1.742	1.5403
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.55	2.89
agent 3	x	Maximum error after t_f (m)	5.3432	1.8781
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.56	2.91
	y	Maximum error after t_f (m)	2.9429	1.2297
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.75	1.51
	ψ	Maximum error after t_f (m)	1.9314	1.6964
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.89	1.39
agent 4	x	Maximum error after t_f (m)	0.2622	0.2329
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0475	0.0428
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0357	0.0313
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 5	x	Maximum error after t_f (m)	0.2609	0.2343
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0475	0.0428
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0632	0.0352
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 6	x	Maximum error after t_f (m)	0.2609	0.2343
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0475	0.0428
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0637	0.057
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—

Table 4.15: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes
for scenario 4.1 case 4

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	5.3067	1.8744
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.6	1.91
	y	Maximum error after t_f (m)	2.9413	1.2282
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.61	1.3
	ψ	Maximum error after t_f (m)	1.9311	1.6955
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.95	1.44
agent 2	x	Maximum error after t_f (m)	0.262	0.2349
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0475	0.0427
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0349	0.0027
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 3	x	Maximum error after t_f (m)	2.8743	0.9127
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.09	0.21
	y	Maximum error after t_f (m)	1.4816	0.6861
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.96	0.55
	ψ	Maximum error after t_f (m)	1.7402	1.5388
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.65	2.73
agent 4	x	Maximum error after t_f (m)	0.2624	0.235
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0474	0.0429
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0359	0.0152
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—

Table 4.16: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes

for scenario 4.1 case 5

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	0.262	0.2349
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0475	0.0427
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0354	0.0053
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 2	x	Maximum error after t_f (m)	5.3312	1.8734
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.57	1.92
	y	Maximum error after t_f (m)	2.9411	1.228
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.62	1.31
	ψ	Maximum error after t_f (m)	1.9306	1.695
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.89	1.4
agent 3	x	Maximum error after t_f (m)	2.8743	0.9127
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.15	0.18
	y	Maximum error after t_f (m)	1.4816	0.6861
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.96	0.59
	ψ	Maximum error after t_f (m)	1.7402	1.5388
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.75	2.04
agent 4	x	Maximum error after t_f (m)	1.0769	0.3977
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.95	0.68
	y	Maximum error after t_f (m)	0.8841	0.3256
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.77	0.21
	ψ	Maximum error after t_f (m)	1.4701	1.2291
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.23	2.98

Table 4.17: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes
for scenario 4.1 case 6

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	2.8756	0.9133
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.15	0.18
	y	Maximum error after t_f (m)	1.4821	0.6864
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.96	0.59
	ψ	Maximum error after t_f (m)	1.7405	1.5391
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.75	2.04
agent 2	x	Maximum error after t_f (m)	0.262	0.2349
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0475	0.0427
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0349	0.0027
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 3	x	Maximum error after t_f (m)	0.262	0.2349
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0471	0.0427
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0354	0.0061
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
agent 4	x	Maximum error after t_f (m)	0.2601	0.235
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	y	Maximum error after t_f (m)	0.0475	0.0427
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—
	ψ	Maximum error after t_f (m)	0.0359	0.0152
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	—

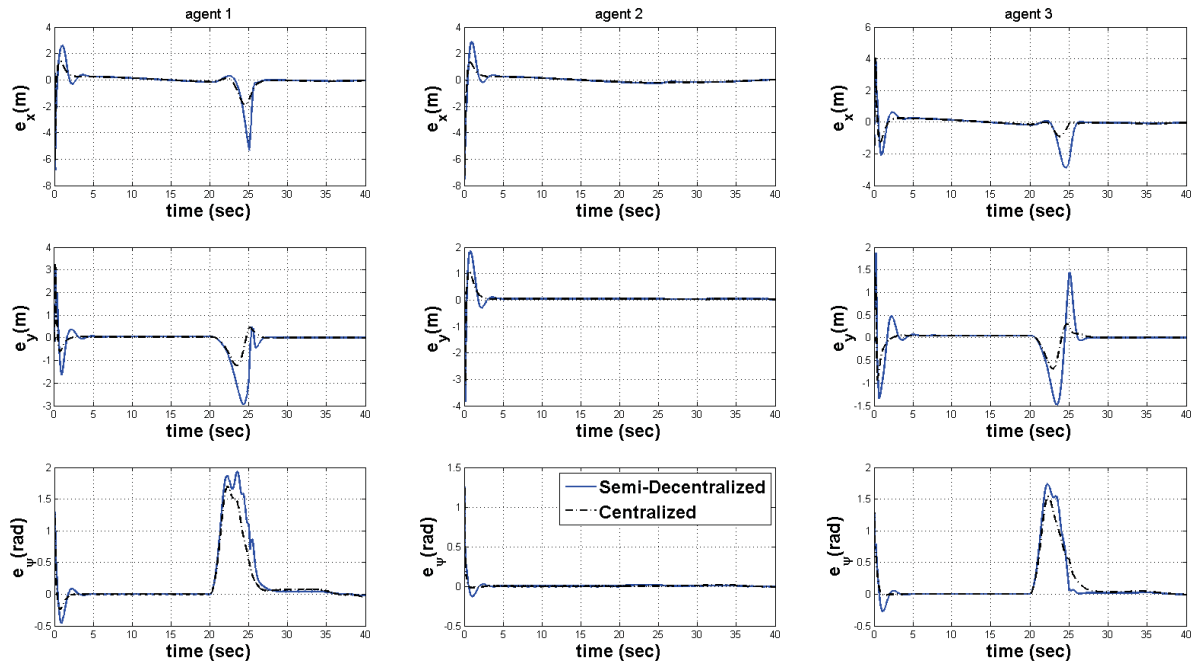


Figure 4.14: Tracking error signals of agents #1, #2, and #3 for scenario 4.1, case 1.

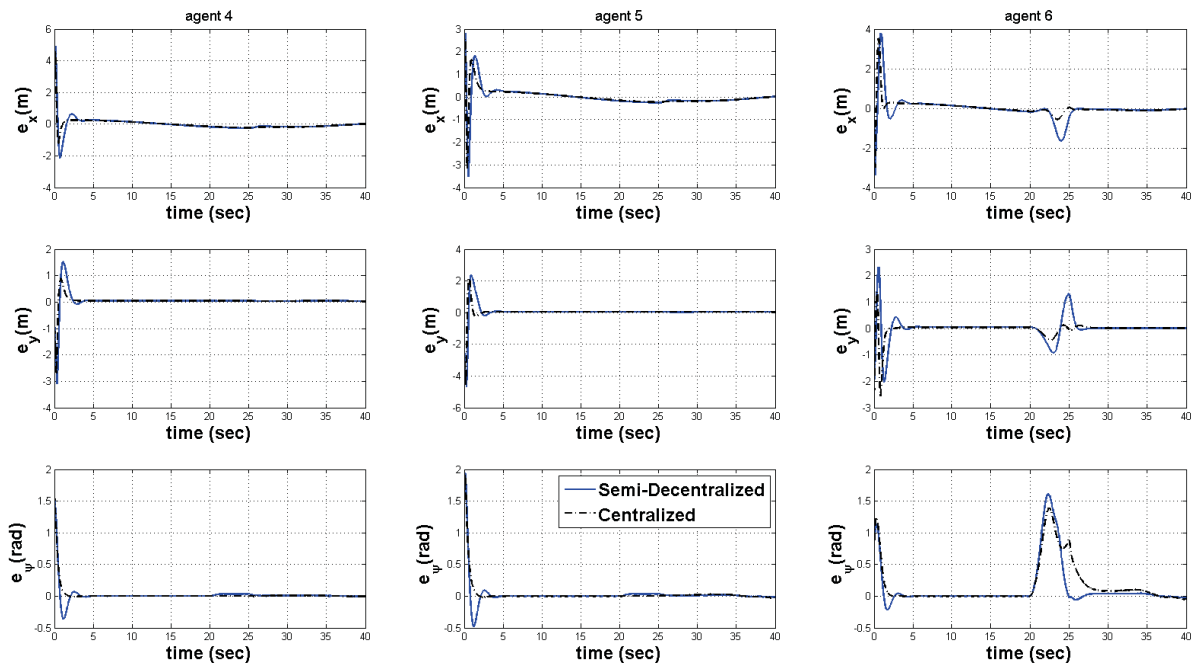


Figure 4.15: Tracking error signals of agents #4, #5, and #6 for scenario 4.1, case 1.

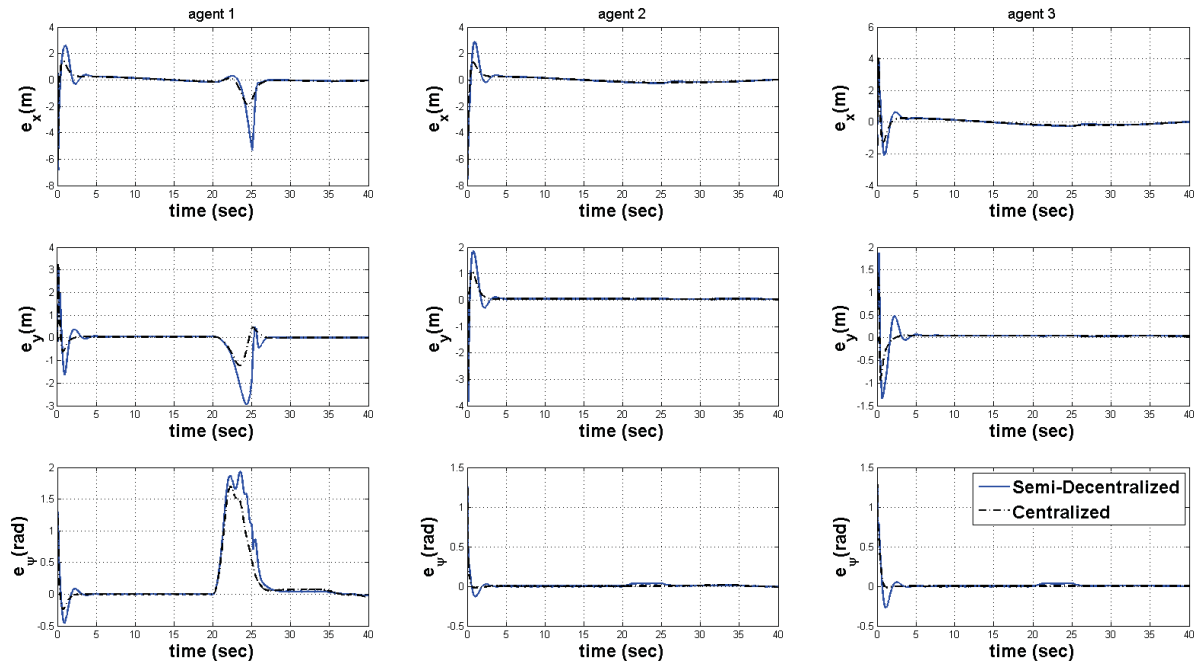


Figure 4.16: Tracking error signals of agents #1, #2, and #3 for scenario 4.1, case 2.

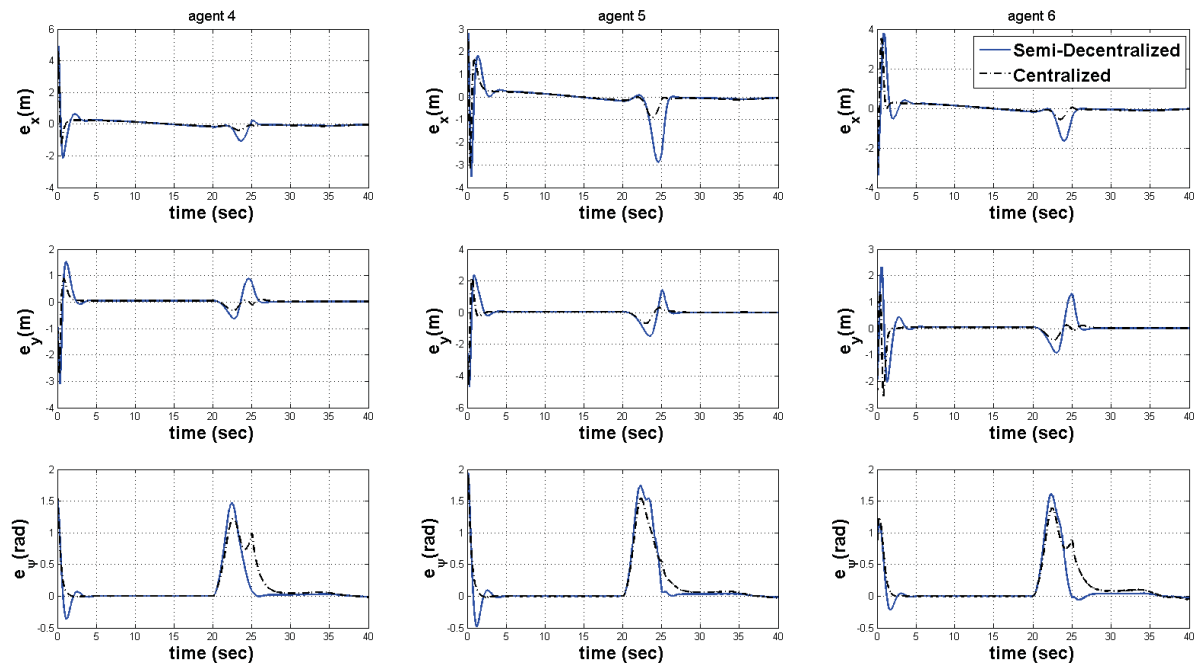


Figure 4.17: Tracking error signals of agents #4, #5, and #6 for scenario 4.1, case 2.

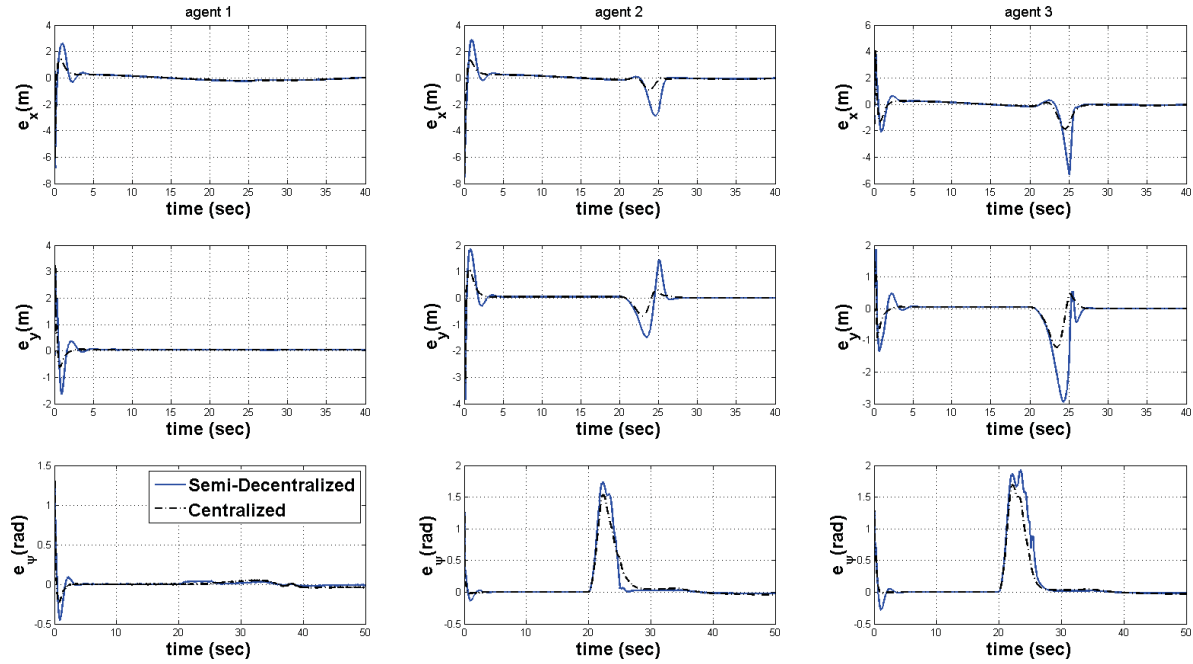


Figure 4.18: Tracking error signals of agents #1, #2, and #3 for scenario 4.1, case 3.

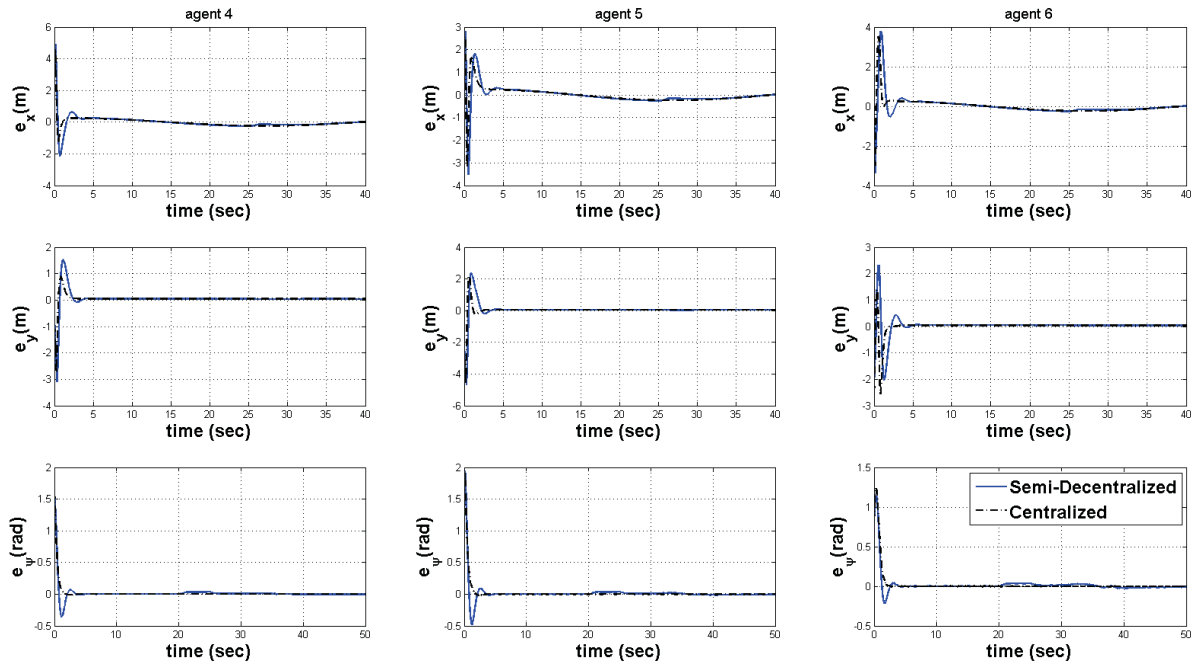


Figure 4.19: Tracking error signals of agents #4, #5, and #6 for scenario 4.1, case 3.

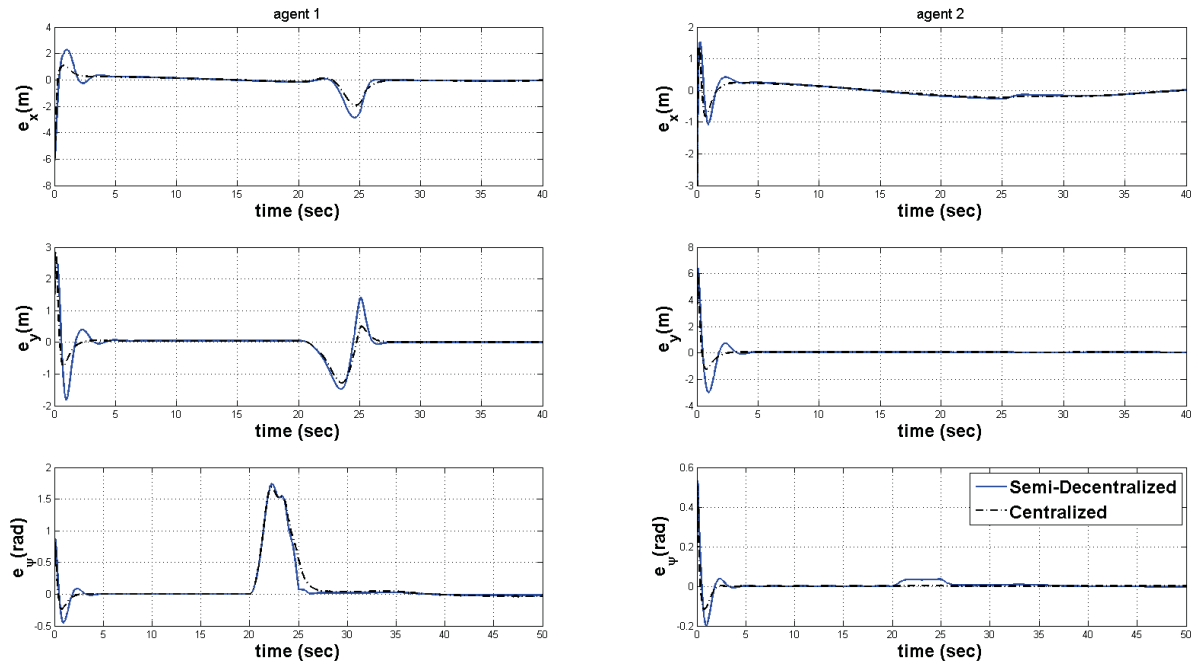


Figure 4.20: Tracking error signals of agents #1 and #2 for scenario 4.1, case 4.

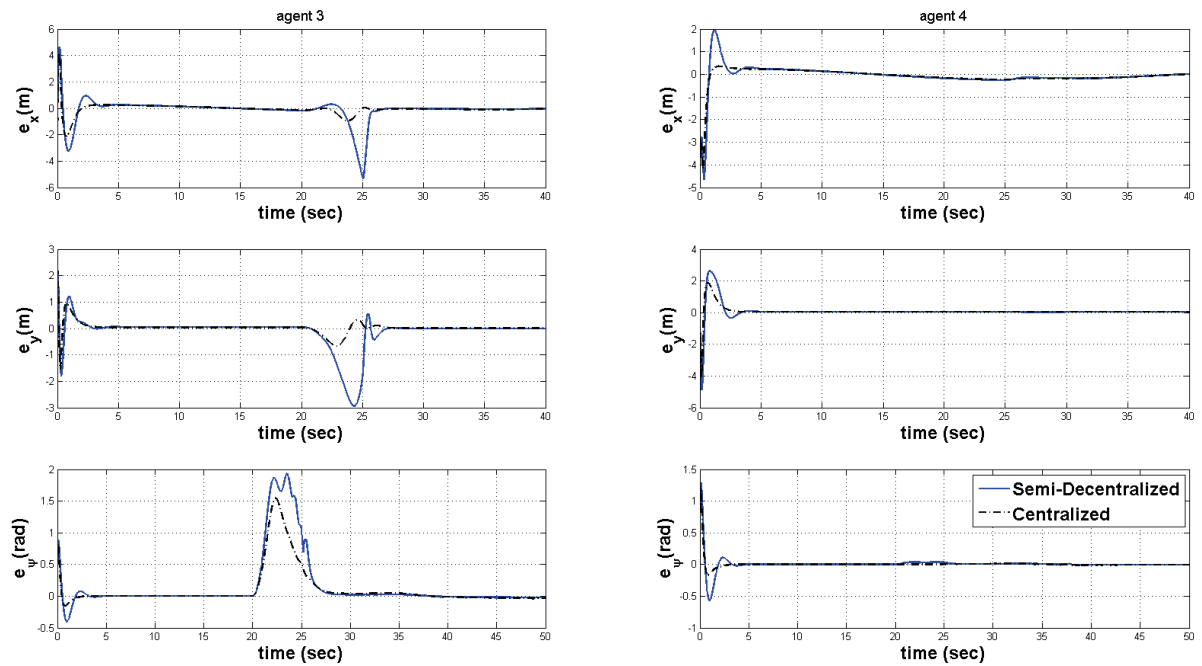


Figure 4.21: Tracking error signals of agents #3 and #4 for scenario 4.1, case 4.

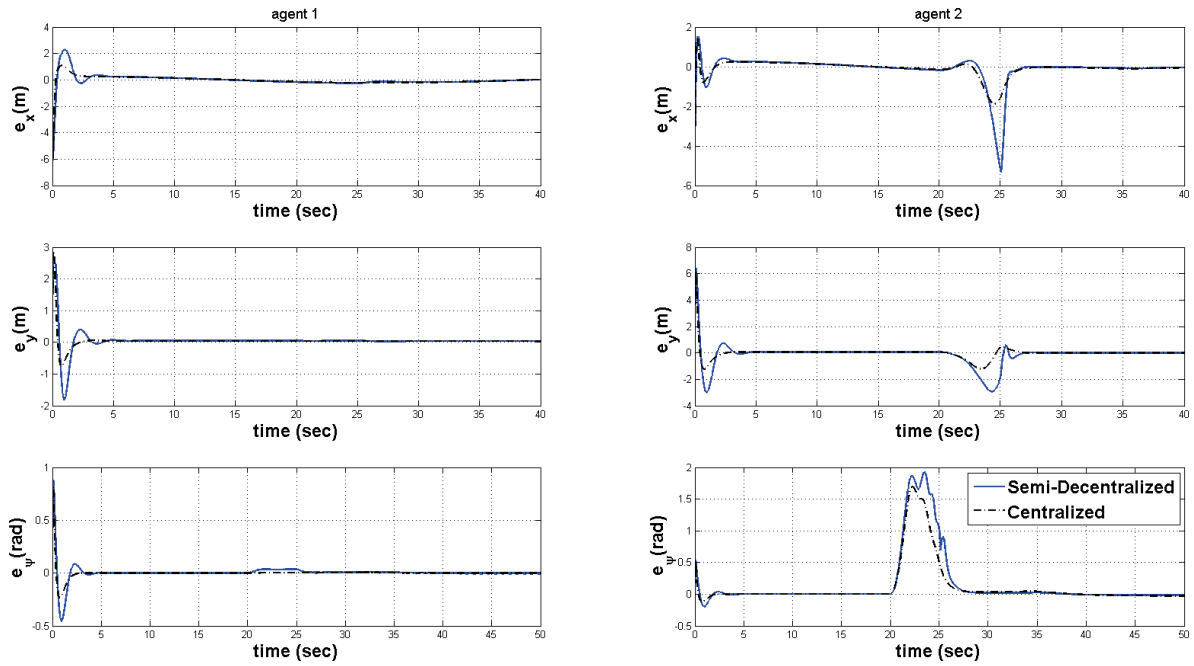


Figure 4.22: Tracking error signals of agents #1 and #2 for scenario 4.1, case 5.

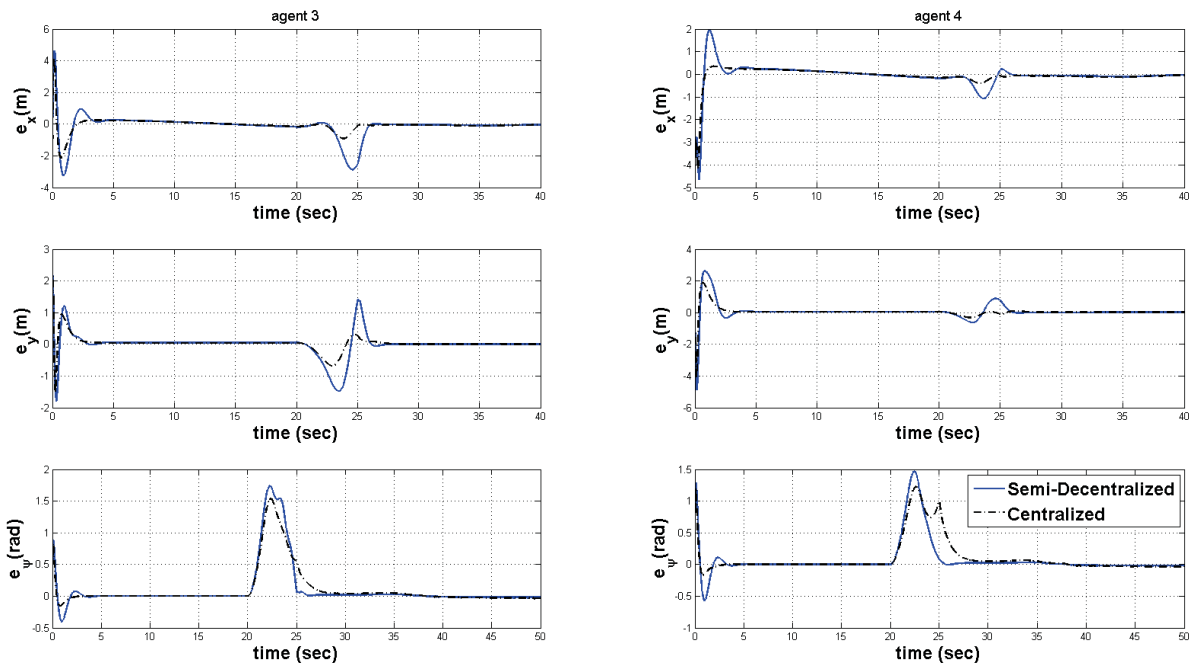


Figure 4.23: Tracking error signals of agents #3 and #4 for scenario 4.1, case 5.

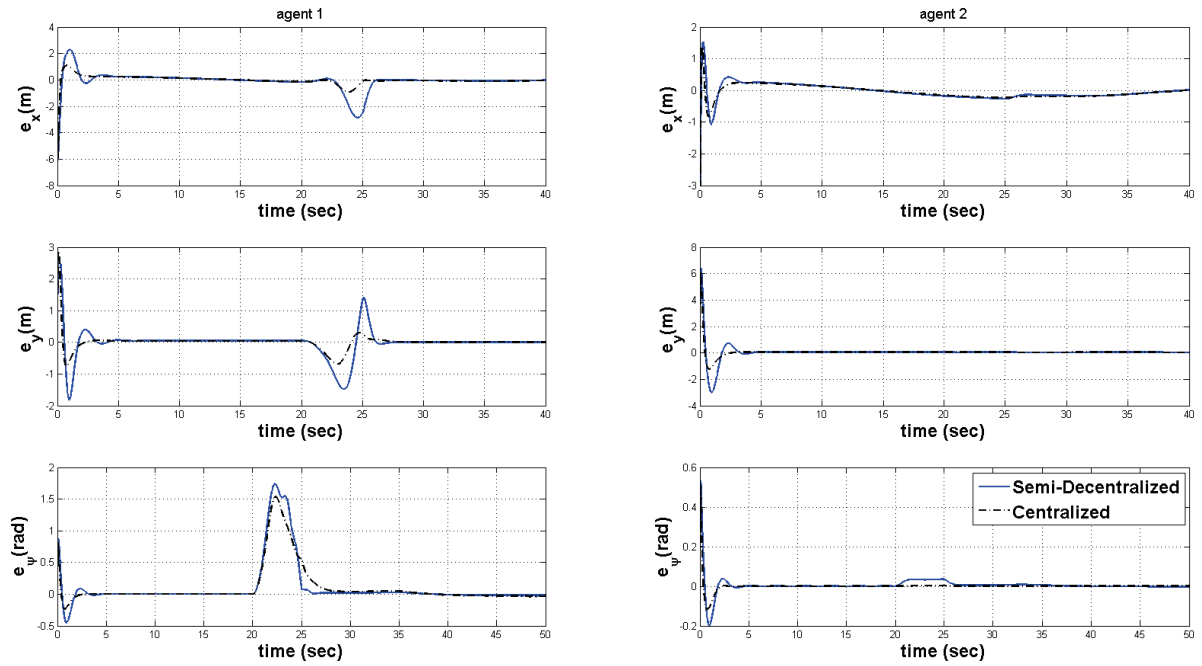


Figure 4.24: Tracking error signals of agents #1 and #2 for scenario 4.1, case 6.

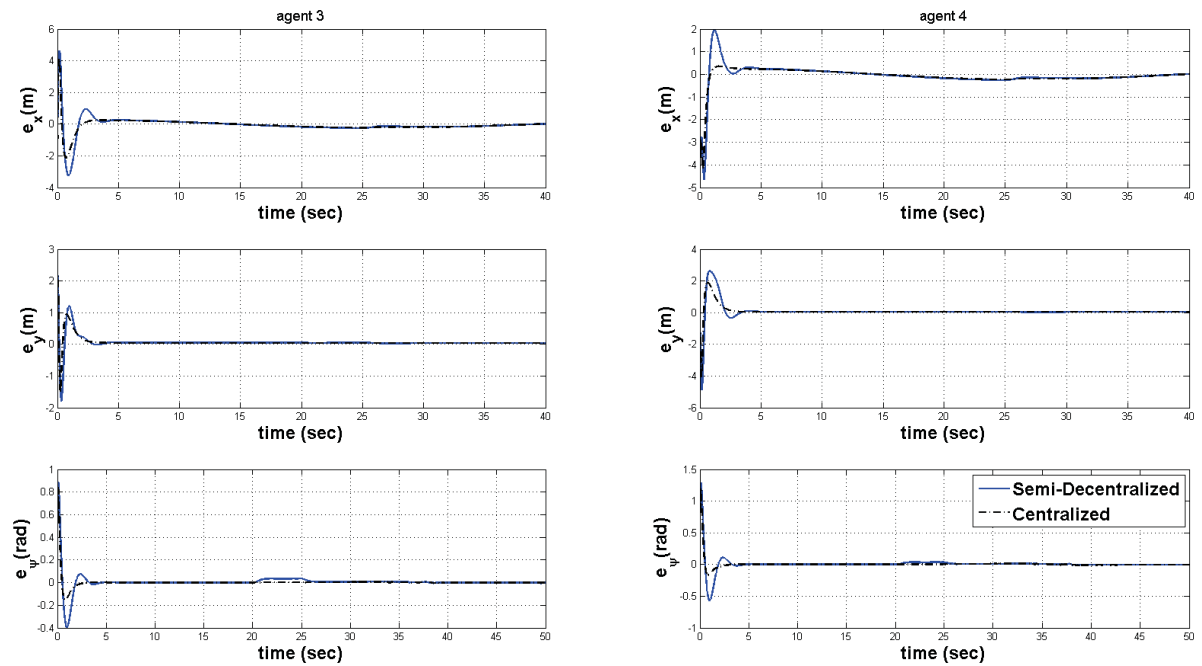


Figure 4.25: Tracking error signals of agents #3 and #4 for scenario 4.1, case 6.

4.3.3.2 Scenario 4.2: Time Delay in Fault Detection

As explained in Section 4.3.1, one of the privileges of using DSC technique to control the group of AUV systems with LOE faults is its robustness to low magnitude faults which can be quite compensated with using only τ_H . Also, for the faults middle range severities, τ_H can keep the group stable although the tracking performance is not ensured. Therefore, even after long delays, compensation control is still capable to recover the faulty vehicles.

However, for severe faults with the range of higher than 80%, the proposed fault-tolerant scheme can only compensate the fault after a limited amount of delay between the time that fault is injected to the multi-agent system and the time that compensation control is activated. The maximum delays for different percentage ranges of LOE faults that τ_C can recover the group of multi-agent underwater vehicles are given in Table 4.18.

These results are obtained based on several simulation tests on G_6 and G_7 from Table 4.9 in which different number of agents were considered faulty with different fault severities. In all of these tests, the control parameters are considered as $\delta_1 = 4$ and $\delta_2 = 1.5$ which are selected based on a trade-off between performance and affordability of control gains. For instance, it has been observed that in both graphs if one or some of the agents have the LOE fault of 90%, the maximum detection delay that after it compensation control can still provide the recovery for the group is 2 *sec*.

It is worth noting that it is possible to extend the tolerated detection time delay by reducing the control parameters. However, it will degrade the performance of the recovery in a way that the recovered system would have higher steady state error, higher maximum error, and also it would take more time for the controller to compensate the injected fault.

Table 4.18: Time delay in detection

	Fault Severity	Maximum Delay
Fault Percentage Range	95%	up to 1 sec
	90%	up to 2 sec
	85%	up to 9 sec
	80%	No limit

4.3.3.3 Scenario 4.3: Error in Estimation of the Fault Severity

In this scenario, it is assumed that the estimation of the fault severity provided by the FDI module has error and is not equal to the actual fault. To test the performance of the proposed fault-tolerant semi-decentralized scheme in tackling this problem, six cases for three different networks have been defined in Table 4.19. In the 1st, 3rd, and 5th cases, it is assumed that the actual fault is underestimated while in the 2nd, 4th, and 6th cases, it has been assumed that the estimated fault is higher than the actual fault. To simulate these scenarios, the detection delay of 5 sec is considered and the control parameters of τ_C are considered as $\delta_1 = 4$ and $\delta_2 = 1.5$. Rest of the assumptions such as desired relative distances, model parameters and uncertainties, initial conditions, reference trajectory, and healthy controller parameters are considered the same as in Section 3.7.

The simulation results of all six cases of this scenario are presented in Figures 4.26 to 4.31 in which only the faulty agents of each case are depicted. The quantitative results of the maximum errors after injection of fault and the time elapsed to obtain almost zero steady state error after activation of compensation control are represented in Tables 4.20 to 4.25 for faulty agents of each case for both semi-decentralized and centralized schemes.

Based on the results and the comparative studies in this scenario, the following outcomes are obtained:

- The proposed fault-tolerant semi-decentralized DSC-based scheme can handle up to 50% error in estimation when it is overestimated. For underestimated case it is able to handle up to 70% of error in estimation.
- When the fault severity is underestimated, as in 1st, 3rd, and 5th cases, there is an error in ψ orientation of the faulty agents after the time that compensation control is activated and this error is higher than the one in 2nd, 4th, and 6th cases where the fault severity is overestimated.
- For all agents, the maximum error after injecting the fault to the system is lower in the centralized scheme in comparison to the proposed semi-decentralized fault-tolerant scheme.
- The time elapsed to obtain $e_{ss} \approx 0$ after the activation of τ_C is mostly lower in the centralized scheme than the proposed semi-decentralized fault-tolerant scheme for all agents in x and y positions. However, in ψ orientation, the semi-decentralized scheme had better performance on damping the error caused by injected fault.
- In comparison to x and ψ states, y position has less influence from LOE fault in actuators.
- The performance of the proposed fault-tolerant semi-decentralized DSC-based scheme does not depend on the topology of the network of agents in case that the FDI module has estimation error.

The reason of the significant performance of the proposed semi-decentralized fault-tolerant DSC-based control scheme in presence of error in estimation of the severity of fault provided

by the FDI module is that in equation of τ_C given in equation (4.1.14), the only part that depends on the estimated fault is $\varsigma = 1 - \max\{\hat{f}_i\}$. Therefore, error in the fault severity has not a severe effect on the faulty systems.

Table 4.19: Different error in estimation for scenario 4.3

Case	Actual Faulty	Estimated Fault	Graph
1	agent 1: 75% agent 3: 65% agent 6: 70%	agent 1: 50% agent 3: 35% agent 6: 35%	G_6
2	agent 1: 40% agent 3: 45% agent 6: 50%	agent 1: 85% agent 3: 85% agent 6: 85%	G_6
3	agent 2: 40% agent 3: 80% agent 4: 80%	agent 2: fault-free agent 3: 35% agent 4: 30%	G_7
4	agent 2: 45% agent 3: fault-free agent 4: 35%	agent 2: 70% agent 3: 25% agent 4: 15%	G_7
5	agent 1: 70% agent 3: 80% agent 5: 75%	agent 1: 15% agent 3: 20% agent 5: 10%	G_3
6	agent 1: 50% agent 3: 40% agent 5: 65%	agent 1: 65% agent 3: 50% agent 5: 70%	G_3

The team level comparison of both schemes for all cases are presented in Tables 4.26 and 4.27. For team level analysis, the root mean square of both characteristics given in Tables 4.20 to 4.25 are obtained for each scheme. In this table, the results given above are confirmed.

Table 4.20: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes

for scenario 4.3 case 1

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	1.6548	0.2534
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.86	0.38
	y	Maximum error after t_f (m)	1.3021	0.3468
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.55	0.43
	ψ	Maximum error after t_f (m)	1.6136	1.0663
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	1.64
agent 3	x	Maximum error after t_f (m)	0.7575	0.1317
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.68	0.49
	y	Maximum error after t_f (m)	0.6235	0.1039
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.92	0.54
	ψ	Maximum error after t_f (m)	1.3196	0.7042
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.82	1.33
agent 6	x	Maximum error after t_f (m)	1.0815	0.1648
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.13	0.6
	y	Maximum error after t_f (m)	0.8878	0.1605
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.65	1.36
	ψ	Maximum error after t_f (m)	1.4734	0.9054
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	—	1.49

Table 4.21: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes

for scenario 4.3 case 2

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	0.2533	0.212
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.04	1.02
	y	Maximum error after t_f (m)	0.0523	0.0396
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.13	0.6
	ψ	Maximum error after t_f (m)	0.5027	0.3777
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.26	1.65
agent 3	x	Maximum error after t_f (m)	0.2794	0.2135
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.07	1.07
	y	Maximum error after t_f (m)	0.0944	0.0413
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.83	0.41
	ψ	Maximum error after t_f (m)	0.6596	0.5126
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.54	1.92
agent 6	x	Maximum error after t_f (m)	0.3284	0.2199
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1	1
	y	Maximum error after t_f (m)	0.1534	0.0735
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.75	0.61
	ψ	Maximum error after t_f (m)	0.828	0.6449
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.44	2.22

Table 4.22: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes

for scenario 4.3 case 3

			Semi-decentralized	Centralized
agent 2	x	Maximum error after t_f (m)	0.2526	0.1255
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	y	Maximum error after t_f (m)	0.0523	0.0233
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.4995	0.1588
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.48	0.65
agent 3	x	Maximum error after t_f (m)	2.8743	0.4775
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.96	1.38
	y	Maximum error after t_f (m)	1.5166	0.3758
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.24	1.77
	ψ	Maximum error after t_f (m)	1.7402	1.2604
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.83	1.56
agent 4	x	Maximum error after t_f (m)	2.8749	0.4777
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.02	1.44
	y	Maximum error after t_f (m)	1.5252	0.3785
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.33	1.83
	ψ	Maximum error after t_f (m)	1.7408	1.2608
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.79	1.69

Table 4.23: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes

for scenario 4.3 case 4

			Semi-decentralized	Centralized
agent 2	x	Maximum error after t_f (m)	0.2785	0.1252
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.49	0.45
	y	Maximum error after t_f (m)	0.0938	0.0233
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.98	0.32
	ψ	Maximum error after t_f (m)	0.6581	0.2283
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.44	0.76
agent 3	x	Maximum error after t_f (m)	0.2641	0.1295
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.16	0.52
	y	Maximum error after t_f (m)	0.0475	0.0233
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.54	≈ 0
	ψ	Maximum error after t_f (m)	0.0051	0.0076
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
agent 4	x	Maximum error after t_f (m)	0.2513	0.1255
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.91	0.68
	y	Maximum error after t_f (m)	0.0696	0.0233
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.29	≈ 0
	ψ	Maximum error after t_f (m)	0.3607	0.113
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.25	0.42

Table 4.24: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes

for scenario 4.3 case 5

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	1.0786	0.8855
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.67	0.31
	y	Maximum error after t_f (m)	1.4707	0.3983
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.87	1.06
	ψ	Maximum error after t_f (m)	0.326	1.2295
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.36	3.01
agent 3	x	Maximum error after t_f (m)	2.8763	0.9138
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	4.12	0.27
	y	Maximum error after t_f (m)	1.5457	0.6866
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.87	0.78
	ψ	Maximum error after t_f (m)	1.7405	1.5391
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	10.78	2.67
agent 5	x	Maximum error after t_f (m)	1.6505	0.5628
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.37	0.86
	y	Maximum error after t_f (m)	1.2993	0.4526
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.45	2.11
	ψ	Maximum error after t_f (m)	1.6115	1.3838
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	10.89	4.52

Table 4.25: Response characteristics of the fault-tolerant semi-decentralized and centralized schemes
for scenario 4.3 case 6

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	0.3276	0.2399
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.15	1.15
	y	Maximum error after t_f (m)	0.1529	0.0833
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.75	0.59
	ψ	Maximum error after t_f (m)	0.8257	0.689
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.4	2.39
agent 3	x	Maximum error after t_f (m)	0.2529	0.2298
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.03	2
	y	Maximum error after t_f (m)	0.0575	0.0427
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.36	1.01
	ψ	Maximum error after t_f (m)	0.5002	0.4231
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.32	1.98
agent 5	x	Maximum error after t_f (m)	0.7559	0.3163
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.8	0.69
	y	Maximum error after t_f (m)	0.6221	0.2402
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.92	0.45
	ψ	Maximum error after t_f (m)	1.3197	1.0813
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.85	2.91

Table 4.26: Team level RMS analysis of maximum error after t_f (m) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.3

	RMS maximum error	
	Semi-decentralized	Centralized
Case 1	1.24142	0.54956
Case 2	0.42929	0.32810
Case 3	1.74889	0.66336
Case 4	0.49687	0.17587
Case 5	1.63752	0.97560
Case 6	0.65234	0.48302

Table 4.27: Team level RMS analysis of time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.3

	RMS time to $e_{ss} \rightarrow 0$	
	Semi-decentralized	Centralized
Case 1	1.14159	1.04079
Case 2	1.26699	1.30906
Case 3	1.76764	1.51996
Case 4	1.05865	0.54645
Case 5	5.57540	2.20383
Case 6	0.65234	0.48302

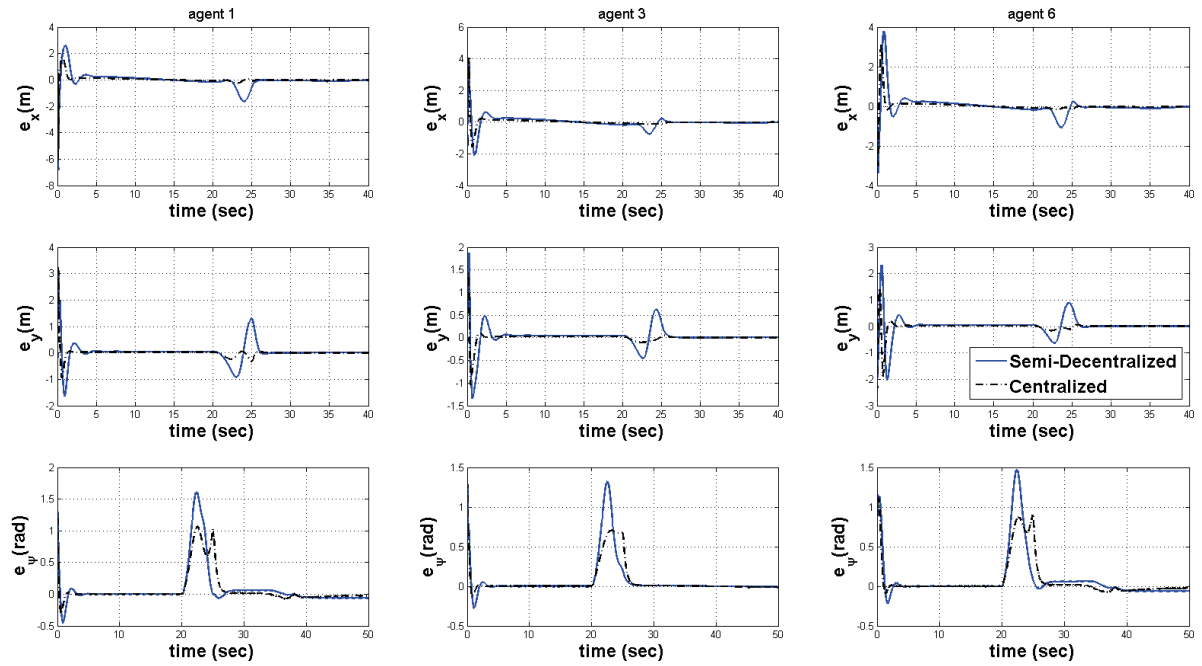


Figure 4.26: Tracking error signals of faulty agents for scenario 4.3, case 1.

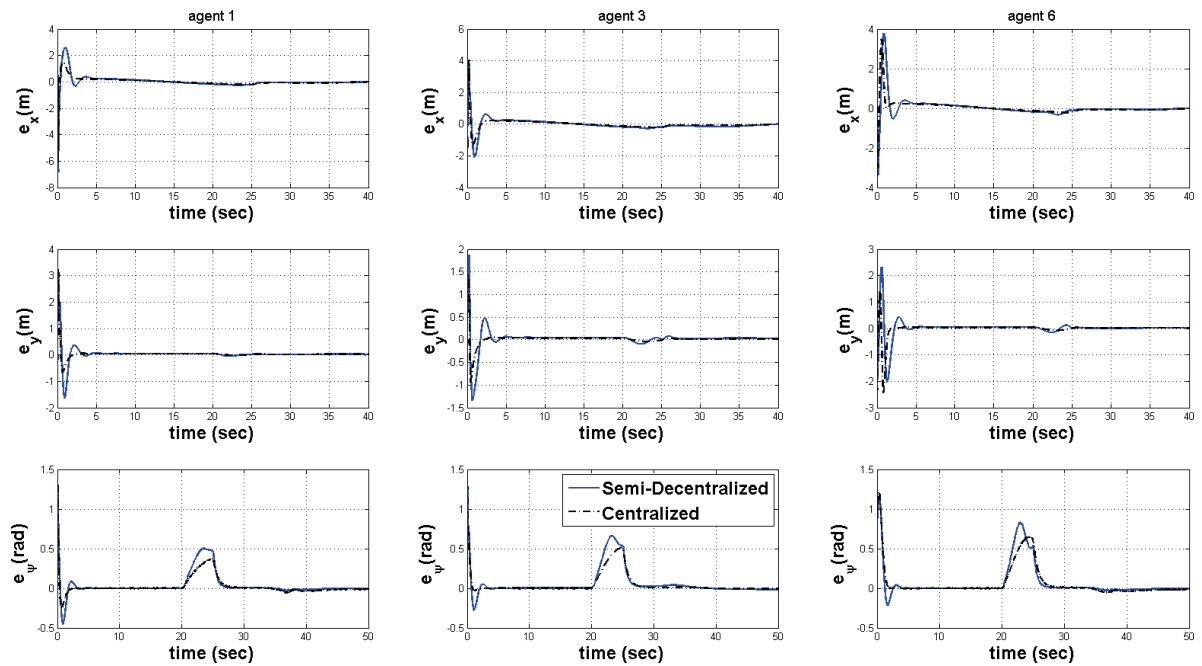


Figure 4.27: Tracking error signals of faulty agents for scenario 4.3, case 2.

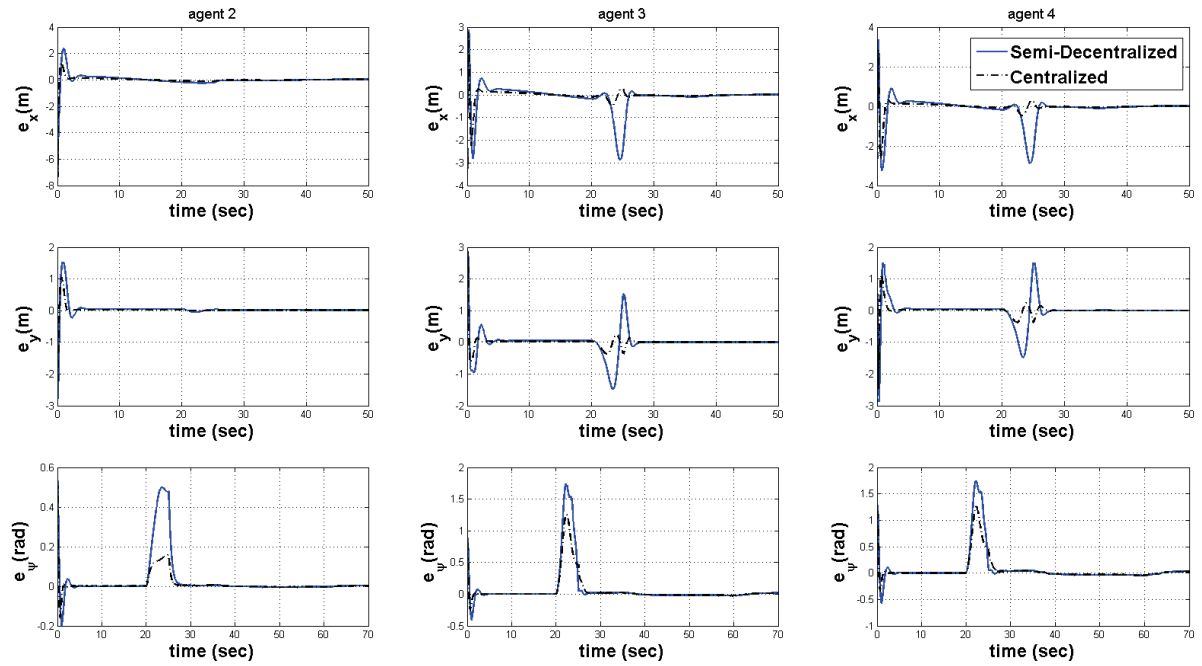


Figure 4.28: Tracking error signals of faulty agents for scenario 4.3, case 3.

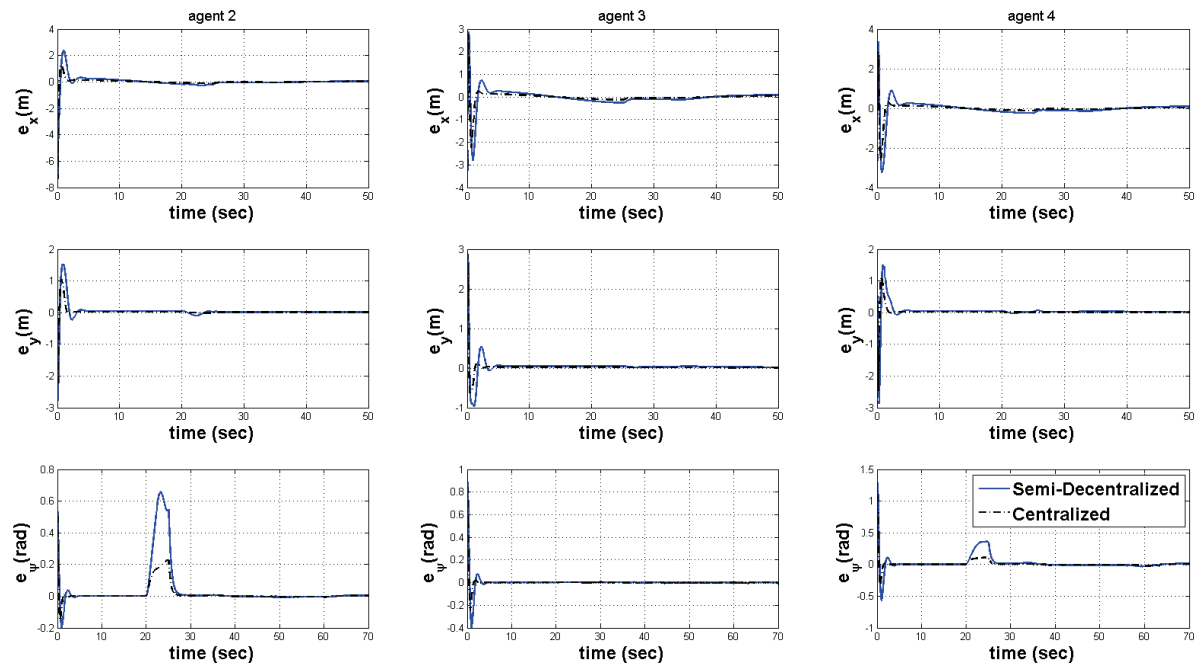


Figure 4.29: Tracking error signals of faulty agents for scenario 4.3, case 4.

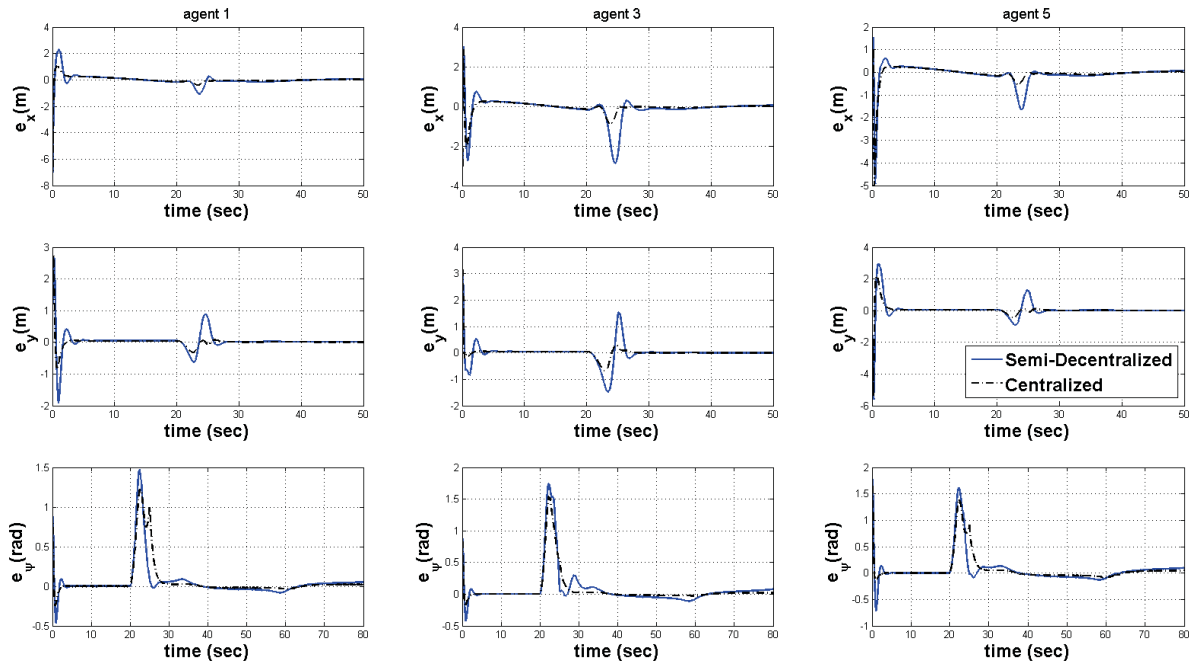


Figure 4.30: Tracking error signals of faulty agents for scenario 4.3, case 5.

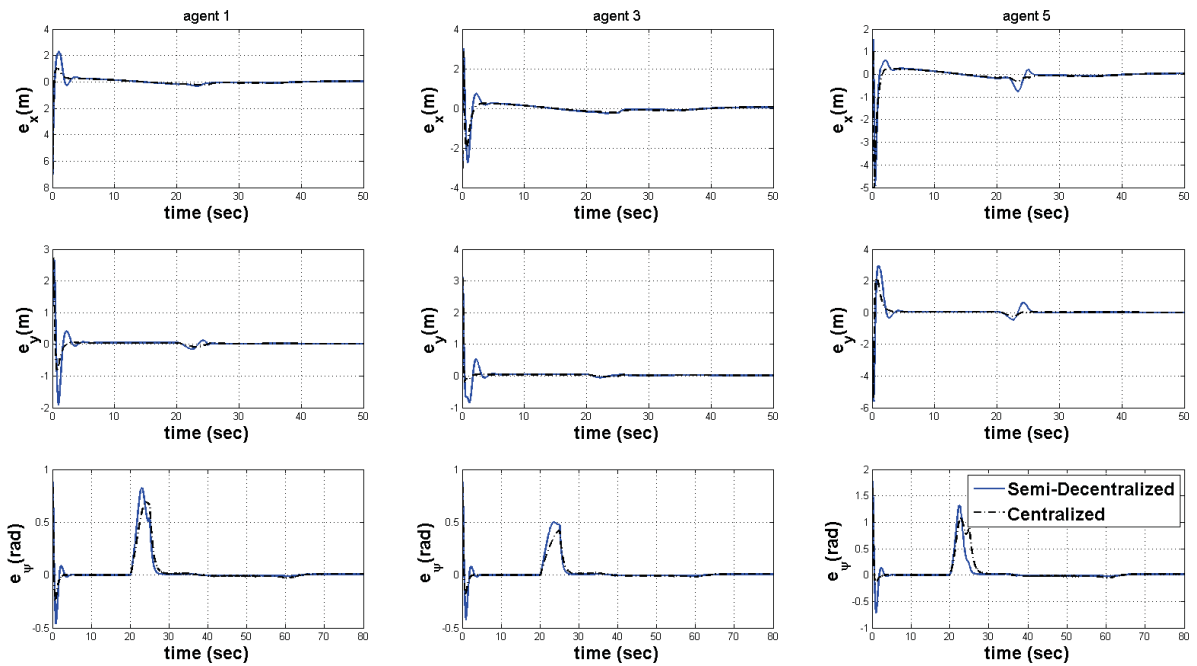


Figure 4.31: Tracking error signals of faulty agents for scenario 4.3, case 6.

4.3.3.4 Scenario 4.4: Error in the Fault Isolation Decision

As the last scenario, the error in the fault isolation decision is considered. It is assumed that the FDI module mistakenly diagnosed a healthy agent as a faulty one or vice versa. To investigate the performance of the proposed fault-tolerant semi-decentralized scheme in such situations, for two different graphs, different conditions are assumed as in Table 4.28. As the first case, for graph G_6 , it is assumed that three agents are faulty but the FDI module diagnosed one of them incorrectly. Second and third cases are quite the same as first one with this difference that instead of one agent, respectively two and three agents are diagnosed by mistake. The same scenarios have been defined for graph G_7 in forth and fifth cases.

The simulation results for all cases of this scenario for both semi-decentralized and centralized schemes are presented in Figures 4.32 to 4.39 where fault is injected to actual faulty agents from Table 4.28 at $t_f = 20 \text{ sec}$ but after a delay of 5 sec the FDI module diagnosed estimated faulty agents incorrectly. Also, the quantitative comparison between these two schemes are represented in Tables 4.29 to 4.33 to evaluate the performance of the proposed semi-decentralized fault-tolerant DSC-based scheme.

From the simulation results of this scenario, it can be observed that because of the robustness of the proposed semi-decentralized DSC-based control in healthy situation against low severity faults, the fault-tolerant semi-decentralized DSC-based control can tackle the error in isolation just in the same way as delay in fault detection scenario. Nevertheless, the steady state errors in ψ orientation of some cases depending on fault severity did not converge to zero as fast as other scenarios. However, this error is close to zero.

In addition, in this scenario, the only limitation is that if a fault-free agent is diagnosed as a faulty agent with the faults in the range of higher than 50%, the system will be unstable. It

is worth noting that the same result was obtained in scenario 4.3 given in Section 4.3.3.3.

From case #1, #2, and #4, it can be seen that in the semi-decentralized scheme because of the fact that agents receive information from their set of neighbors, existence of fault can slightly influence the performance in x direction of agents which are neither faulty nor incorrectly diagnosed as faulty agents.

Table 4.28: Different error in isolation for scenario 4.4

Case	Actual faulty agents	Estimated faulty agents	Graph
1	agent 1: 85% agent 2: 75% agent 6: 50%	agent 1: 85% agent 2: 75% agent 4: 50%	G_6
2	agent 1: 80% agent 2: 45% agent 6: 50%	agent 1: 80% agent 3: 45% agent 4: 50%	G_6
3	agent 1: 40% agent 2: 48% agent 6: 50%	agent 5: 40% agent 3: 48% agent 4: 50%	G_6
4	agent 3: 75% agent 4: 35%	agent 3: 75% agent 1: 35%	G_7
5	agent 3: 50% agent 4: 45%	agent 2: 50% agent 1: 45%	G_7

Table 4.29: Quantitative analysis of the semi-decentralized and centralized schemes in faulty situation
for scenario 4.4 case 1

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	5.3072	1.0935
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.87	0.75
	y	Maximum error after t_f (m)	2.9473	0.6948
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.85	0.93
	ψ	Maximum error after t_f (m)	1.9339	1.4659
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	2.52	0.22
agent 2	x	Maximum error after t_f (m)	1.6533	0.2525
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.18	0.46
	y	Maximum error after t_f (m)	1.3012	0.3522
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.49	0.5
	ψ	Maximum error after t_f (m)	1.6119	1.0638
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.78	0.92
agent 4	x	Maximum error after t_f (m)	0.2642	0.1295
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.79	0.42
	y	Maximum error after t_f (m)	0.0475	0.0234
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.034	0.0324
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
agent 6	x	Maximum error after t_f (m)	0.3284	0.1275
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.95	0.78
	y	Maximum error after t_f (m)	0.1534	0.0323
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.93	0.51
	ψ	Maximum error after t_f (m)	0.828	0.3306
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.9	1.17

Table 4.30: Quantitative analysis of the semi-decentralized and centralized schemes in faulty situation

for scenario 4.4 case 2

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	2.884	0.4812
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.31	0.41
	y	Maximum error after t_f (m)	1.4855	0.3768
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.03	0.98
	ψ	Maximum error after t_f (m)	1.7437	1.2649
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.76	1.5
agent 2	x	Maximum error after t_f (m)	0.2793	0.1253
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.37	0.96
	y	Maximum error after t_f (m)	0.0947	0.0236
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.94	0.49
	ψ	Maximum error after t_f (m)	0.6603	0.2294
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.43	0.76
agent 3	x	Maximum error after t_f (m)	0.2642	0.1296
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.93	0.47
	y	Maximum error after t_f (m)	0.0476	0.0234
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.0223	0.0217
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
agent 4	x	Maximum error after t_f (m)	0.2642	0.1295
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.15	0.59
	y	Maximum error after t_f (m)	0.0475	0.0235
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.034	0.0324
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
agent 6	x	Maximum error after t_f (m)	0.3284	0.1275
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.96	0.7
	y	Maximum error after t_f (m)	0.1534	0.0323
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.95	0.56
	ψ	Maximum error after t_f (m)	0.828	0.3306
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.92	1.27

Table 4.31: Quantitative analysis of the semi-decentralized and centralized schemes in faulty situation

for scenario 4.4 case 3

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	0.2533	0.1256
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.15	0.75
	y	Maximum error after t_f (m)	0.0658	0.0235
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.5027	0.1611
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.42	0.7
agent 2	x	Maximum error after t_f (m)	0.305	0.1263
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	y	Maximum error after t_f (m)	0.1279	0.0261
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.75	0.53
	ψ	Maximum error after t_f (m)	0.7594	0.2854
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.44	0.86
agent 3	x	Maximum error after t_f (m)	0.2641	0.1296
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.86	0.8
	y	Maximum error after t_f (m)	0.0476	0.0234
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.0273	0.0219
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
agent 4	x	Maximum error after t_f (m)	0.2642	0.1295
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.08	0.62
	y	Maximum error after t_f (m)	0.0473	0.0234
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.034	0.0324
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
agent 5	x	Maximum error after t_f (m)	0.2641	0.1296
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.92	0.63
	y	Maximum error after t_f (m)	0.0475	0.0234
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.0343	0.0512
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
agent 6	x	Maximum error after t_f (m)	0.3284	0.1275
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.91	0.71
	y	Maximum error after t_f (m)	0.1534	0.0323
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.89	0.52
	ψ	Maximum error after t_f (m)	0.828	0.3306
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.89	1.26

Table 4.32: Quantitative analysis of the semi-decentralized and centralized schemes in faulty situation
for scenario 4.4 case 4

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	0.264	0.1295
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.98	0.49
	y	Maximum error after t_f (m)	0.0475	0.0233
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.0147	0.0114
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
agent 3	x	Maximum error after t_f (m)	1.6482	0.2513
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.39	0.94
	y	Maximum error after t_f (m)	1.2979	0.3496
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.66	0.68
	ψ	Maximum error after t_f (m)	1.61	1.0621
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.71	2.05
agent 4	x	Maximum error after t_f (m)	0.2513	0.1255
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.18	0.91
	y	Maximum error after t_f (m)	0.0524	0.0233
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.46	≈ 0
	ψ	Maximum error after t_f (m)	0.3607	0.113
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.17	0.45

Table 4.33: Quantitative analysis of the semi-decentralized and centralized schemes in faulty situation
for scenario 4.4 case 5

			Semi-decentralized	Centralized
agent 1	x	Maximum error after t_f (m)	0.264	0.1295
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.26	0.75
	y	Maximum error after t_f (m)	0.0475	0.0233
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.0034	0.0052
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
agent 2	x	Maximum error after t_f (m)	0.2641	0.1295
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.16	0.63
	y	Maximum error after t_f (m)	0.0475	0.0233
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
	ψ	Maximum error after t_f (m)	0.0036	0.0013
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	≈ 0	≈ 0
agent 3	x	Maximum error after t_f (m)	0.3269	0.1273
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.74	0.71
	y	Maximum error after t_f (m)	0.1525	0.0359
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.83	0.64
	ψ	Maximum error after t_f (m)	0.8245	0.3286
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.86	1.18
agent 4	x	Maximum error after t_f (m)	0.2787	0.1252
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	0.97	0.51
	y	Maximum error after t_f (m)	0.0939	0.0252
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.94	0.42
	ψ	Maximum error after t_f (m)	0.6592	0.2292
		Time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec)	1.56	0.9

The team level comparison of both schemes for all cases are presented in Tables 4.34 and 4.35. For team level analysis, the root mean square of the maximum error after t_f and the time to $e_{ss} \rightarrow 0$ after activation of τ_c are given for each scheme. In this table, all results explained above are confirmed.

Table 4.34: Team level RMS analysis of maximum error after t_f (m) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.4

	RMS maximum error	
	Semi-decentralized	Centralized
Case 1	2.01055	0.66406
Case 2	1.00167	0.38342
Case 3	0.33692	0.133981
Case 4	0.89823	0.38872
Case 5	0.35058	0.13813

Table 4.35: Team level RMS analysis of time to $e_{ss} \rightarrow 0$ after activation of τ_c (sec) for the semi-decentralized and centralized schemes in faulty situation for Scenario 4.4

	RMS time to $e_{ss} \rightarrow 0$	
	Semi-decentralized	Centralized
Case 1	1.53271	0.72094
Case 2	1.30516	0.85829
Case 3	1.17981	0.76527
Case 4	1.38693	1.06608
Case 5	1.47594	0.75099

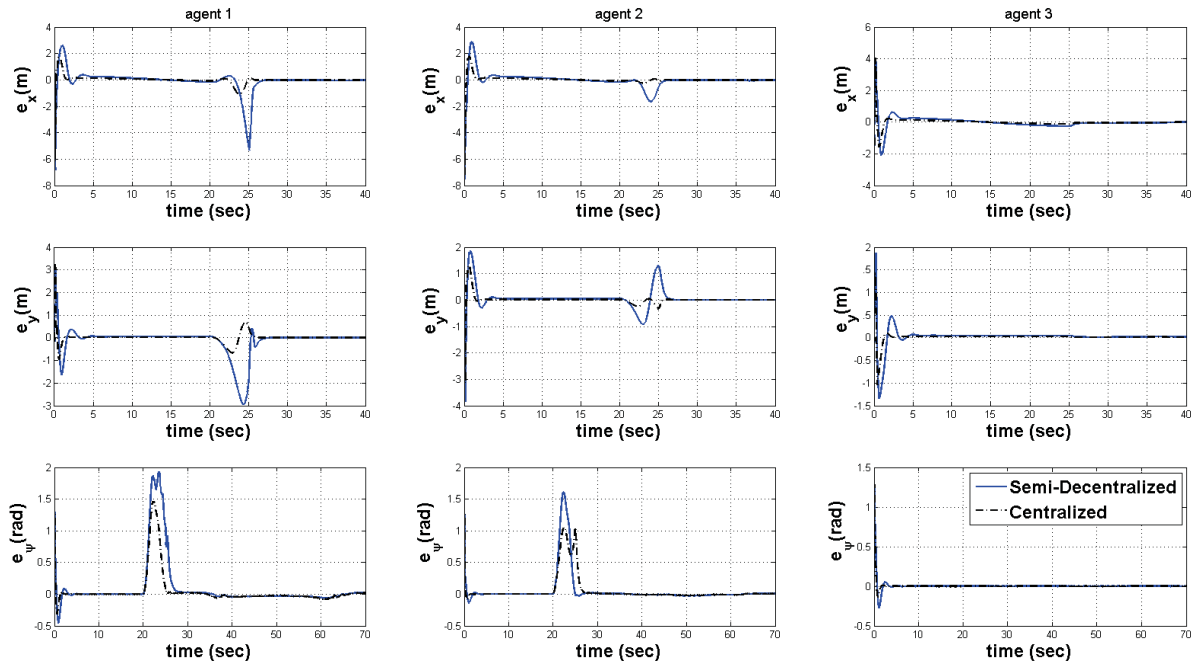


Figure 4.32: Tracking error signals of agents #1, #2, and #3 for scenario 4.4, case 1.

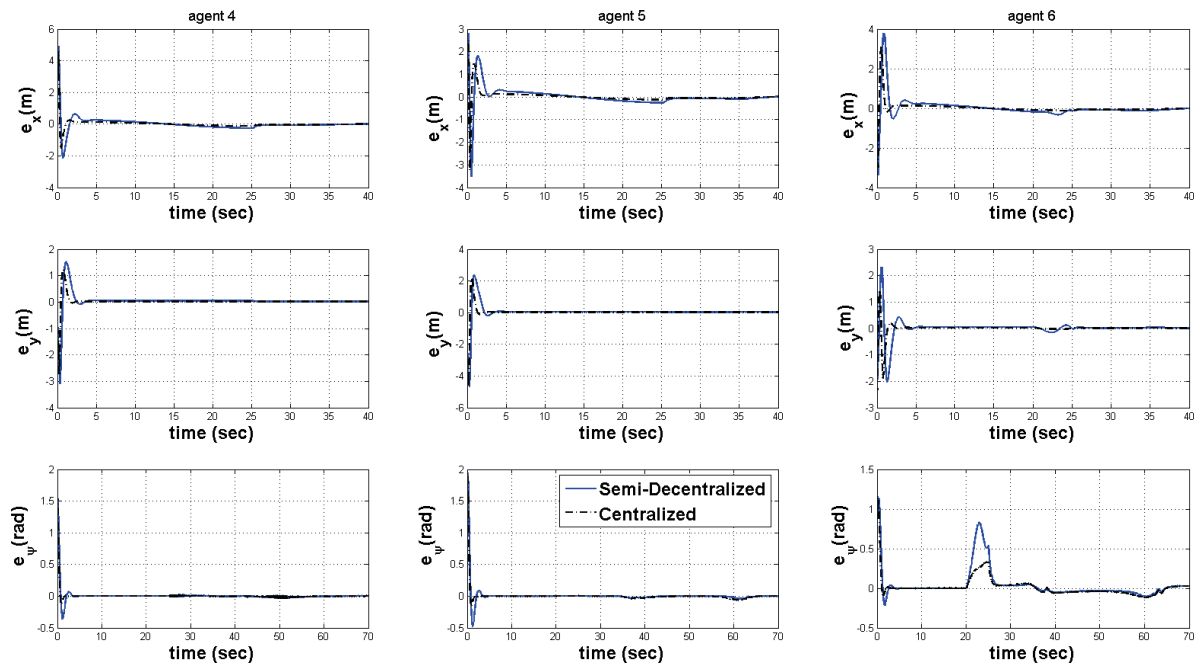


Figure 4.33: Tracking error signals of agents #4, #5, and #6 for scenario 4.4, case 1.

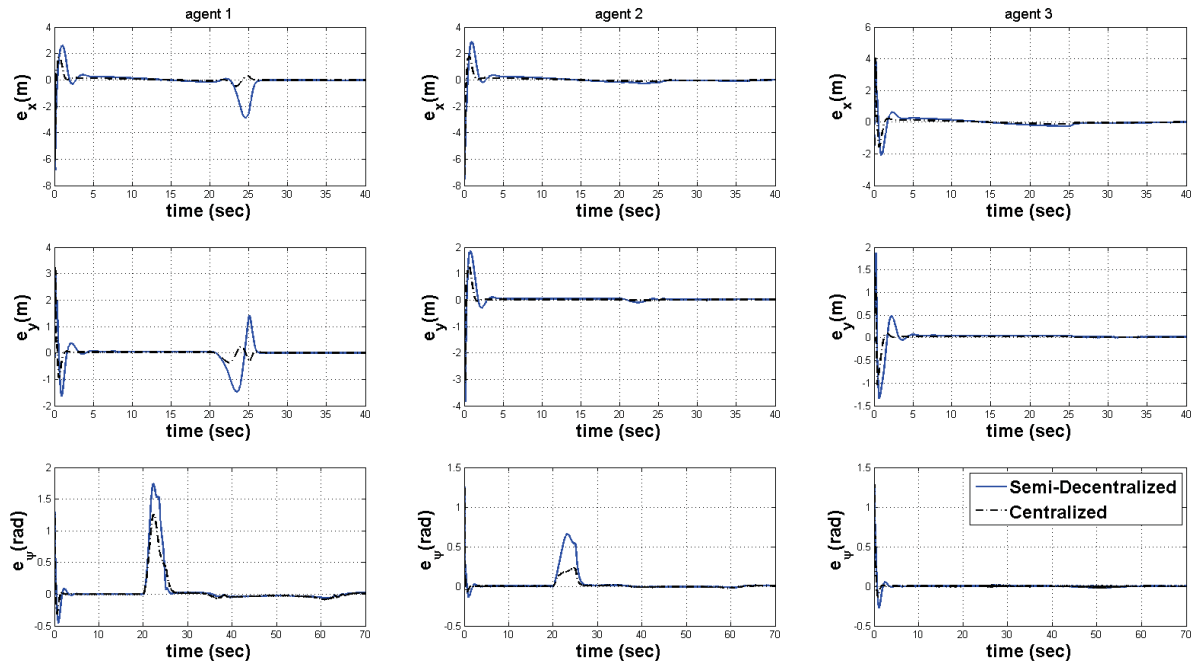


Figure 4.34: Tracking error signals of agents #1, #2, and #3 for scenario 4.4, case 2.

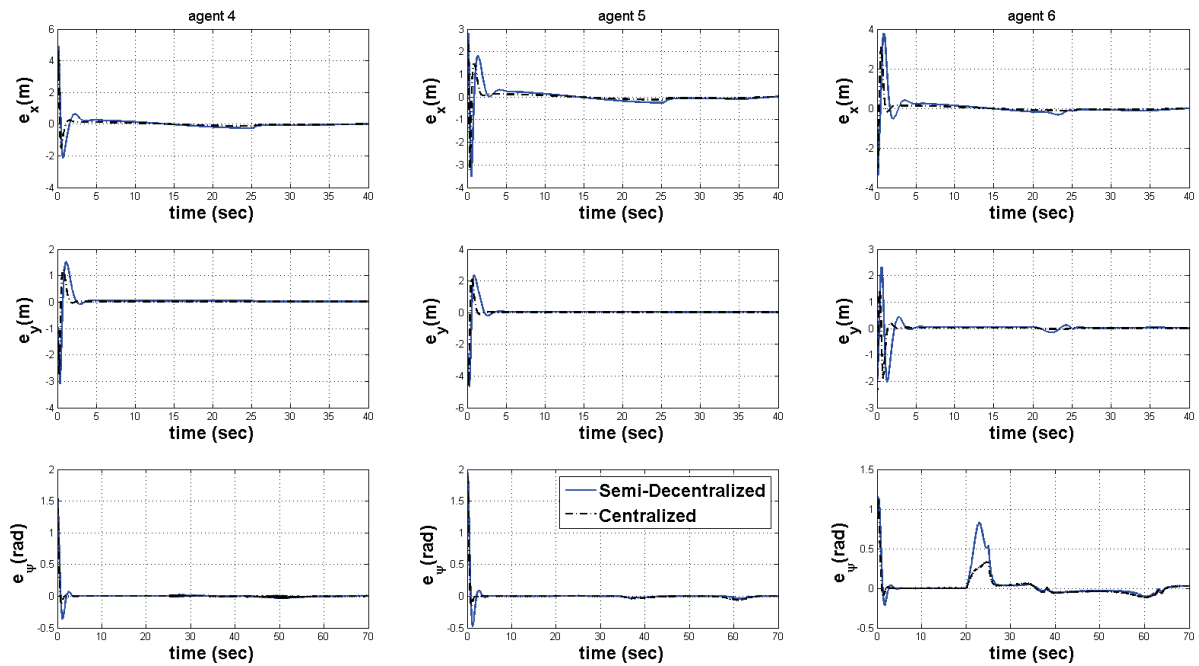


Figure 4.35: Tracking error signals of agents #4, #5, and #6 for scenario 4.4, case 2.

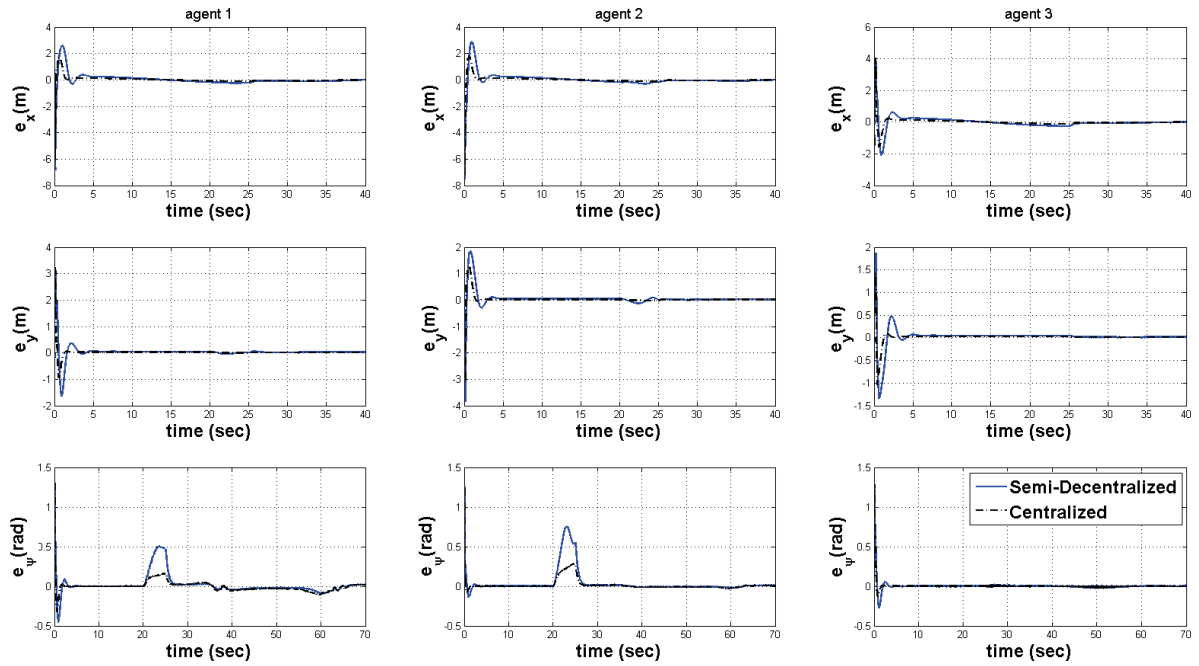


Figure 4.36: Tracking error signals of agents #1, #2, and #3 for scenario 4.4, case 3.

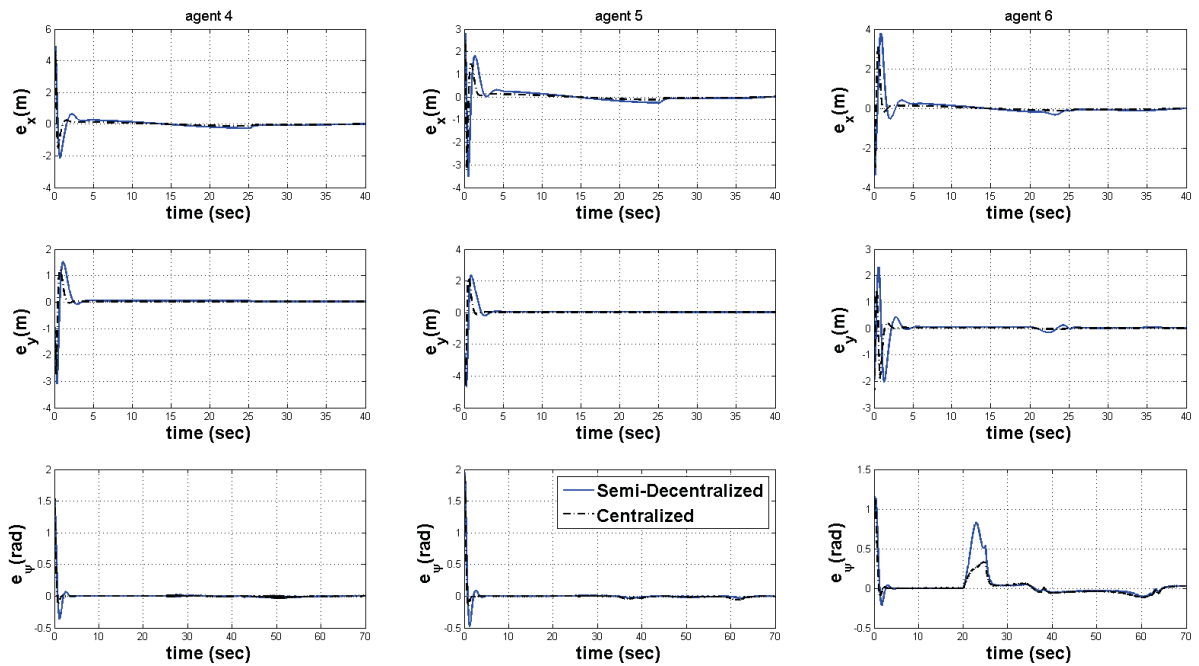


Figure 4.37: Tracking error signals of agents #4, #5, and #6 for scenario 4.4, case 3.

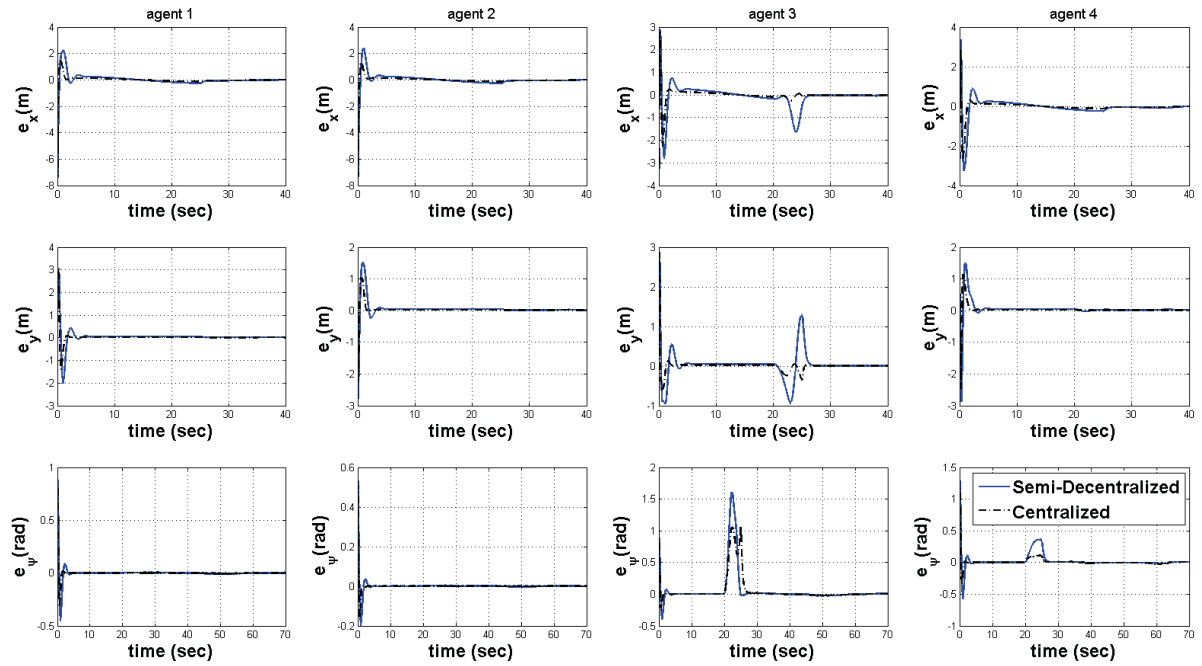


Figure 4.38: Tracking error signals of all agents for scenario 4.4, case 4.

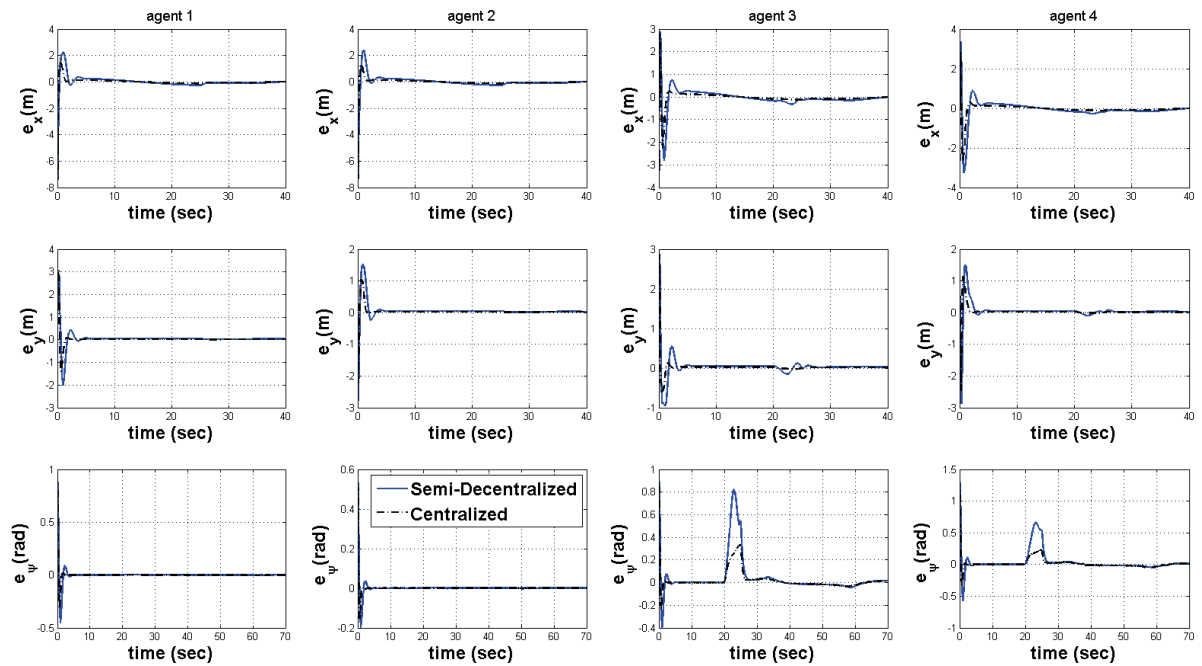


Figure 4.39: Tracking error signals of all agents for scenario 4.4, case 5.

4.4 Conclusion

In this chapter, the active fault-tolerant DSC-based centralized, decentralized, and semi-decentralized schemes are addressed to tackle the LOE fault in underwater vehicles' actuators which are one of the most common and important sources of faults in underwater vehicles due to exposure to seawater. The aim of developing these active fault-tolerant control schemes are to guarantee the boundedness of closed-loop signals and to ensure that the group of heterogeneous AUVs track a desired trajectory while keeping a desired formation. Since the performance of active fault-tolerant control schemes relies on the information provided by the fault detection and isolation modules, it is important to consider errors in detection, isolation, and identification when dealing with these systems.

In order to investigate the performance of the proposed fault-tolerant semi-decentralized DSC-based control strategy, the simulation results of various scenarios and comparative studies are demonstrated. In the simulation part, beside evaluating the performance of the semi-decentralized scheme, the effect of four different conditions are analyzed as well. These conditions are the effect of the number of faulty agents in a given network, delay in fault detection, error in estimation of fault severity in the agents, and error in isolating the faulty agents from fault-free ones. Based on the results obtained from the simulation studies, it can be attained that the proposed fault-tolerant semi-decentralized DSC-based scheme has the ability to overcome the LOE fault in underwater vehicles' actuators up to 95% severity depending on the situations no matter how many of agents are faulty.

However, this performance might be degraded for some conditions caused by the FDI module. These conditions impose some limitations for the recovery process. For instance, in Scenario 4.2 in which the effect of delay in fault detection is investigated, for the nominal control param-

eters selected based on a trade-off between performance and affordability of control gains, in case that one or more agents have the LOE fault with 95% severity, the fault-tolerant scheme can recover the system only if there is a maximum 1 *sec* delay in detection of fault. For the LOE faults with 90% and 85% severities, the limitation in maximum tolerable delay in detection are 2 *sec* and 9 *sec*, respectively. For the LOE faults with 80% magnitude and less, there are no limitation in maximum tolerable delay in fault detection.

Also, based on the results of Scenario 4.3, the limitations that the error in estimation of fault severity impose to the system are that the proposed fault-tolerant semi-decentralized DSC-based scheme can handle up to 50% error in fault estimation when the fault severity is over-estimated. For under-estimated cases, it is able to handle up to 70% error in fault estimation.

Based on the results of Scenario 4.4, the only limitation that the error in isolation imposes to the recovery system is that the fault-tolerant semi-decentralized DSC-based control can recover the injected fault only if the fault-free agent which is incorrectly diagnosed as a faulty agent associates with the faults with severity lower than 50%.

Chapter 5

Conclusions and Future Work

In this thesis, the problems of cooperative control and fault recovery of a network of heterogeneous autonomous underwater vehicles are addressed using four different cooperative schemes based on dynamic surface control. DSC technique is employed in this thesis due to its great capability in controlling nonlinear systems with uncertainties.

First, the DSC-based centralized control scheme is represented as a benchmark since in this scheme there is a central controller that has global information of all agents which enables it to provide the optimal solution for the cooperative problem. However, these optimal results are obtained at the cost of strict communication constraints due to the fact that all agents send and receive information to and from all other agents in the group. Despite the best performance, the centralized control is unappealing for economical and implementation reasons in case that the system model is huge and many agents are involved in the mission.

The second scheme that is developed in this thesis is the DSC-based decentralized control

scheme which is opposite to the centralized scheme. In this scheme, each agent determines its own controller based only on its own information and the information of its desired path while there is no communication among agents of the group. Although decentralization makes it very simple to implement the system, the absence of communication between agents notably degrades the performance of all agents and in case one of the agents loses its performance efficiency, there will be no chance to recover it.

The third proposed scheme, i.e. the DSC-based semi-decentralized scheme consists of a consensus algorithm using the desired trajectories, is a cooperative scheme which is a middle ground between the centralized and decentralized schemes in the sense that it does not have strict communication constraints and still has outstanding performance. The semi-decentralized scheme is similar to the centralized controller which is divided into several simpler sub-systems that causes lower computational and communication requirements. It also resolves the problem of a single point of failure in the system. The simulation results and comparative studies for all three developed cooperative schemes indicate the superior performance of the semi-decentralized scheme.

The fourth developed cooperative scheme is the semi-decentralized control scheme which consists of dynamic surface control technique and a consensus algorithm that uses relative distances and velocities of agents. The development of this scheme was motivated by showing the superiority of the semi-decentralized scheme with consensus algorithm using the desired trajectories based on two main reasons. First, in the semi-decentralized scheme with consensus algorithm using the desired trajectories only desired positions belonging to \mathbb{R}^3 space are transmitted while in the semi-decentralized scheme with consensus algorithm uses relative distances and velocities of agents, $z_1^{ij} \in \mathbb{R}^3$ and $z_2^{ij} \in \mathbb{R}^3$ are transmitted between agents. Therefore, the

size of the transmitted data is halved in the former scheme in comparison to the latter one. Second, the desired positions, i.e. $z_{1d}^{(i)}$ s, are the data calculated by the agents, and agents in the semi-decentralized scheme with consensus algorithm using the desired trajectories do not need to obtain data from displacement and velocity sensors which are subject to errors and faults.

To analyze the performance of proposed semi-decentralized control strategy, three cooperative schemes of model-dependent coordinated tracking algorithm presented in [108], namely the centralized, decentralized, and semi-decentralized schemes were compared to the DSC-based semi-decentralized scheme and the results show the superior performance of the proposed semi-decentralized scheme with consensus algorithm using the desired trajectories. Based on this comparative study the advantages of our proposed schemes are evident. In contrast to the model-dependent coordinated tracking algorithm, in the proposed DSC-based semi-decentralized scheme there is no need to exchange the velocities of agents and the virtual leader. Another disadvantage of using the model-dependent coordinated tracking algorithm for the problem given in this thesis is that since this method does not have the ability to overcome the uncertainties in the system, the only way to make it more robust is to use high gain controllers. However, even by using high gains one cannot reach the same performance of the proposed semi-decentralized scheme based on DSC technique.

Moreover, in order to cope with the problem of LOE faults in actuators of autonomous underwater vehicles, the DSC-based active fault-tolerant control layout of all three aforementioned cooperative schemes are developed in such a way that once the FDI module detects the existence of a fault in an actuator, a compensation control input will be activated to eliminate the effects of the injected fault. The simulation studies with various conditions and situations indicate that the recovered system by the active fault-tolerant DSC-based semi-decentralized

scheme meets the design specifications even if the performance of the FDI module is not ideal. Non-idealities of the FDI module that are considered in the simulation studies consist of the effects of the number of faulty agents in a network, delays in fault detection, errors in estimation of the fault severity, and isolation errors to distinguish the faulty agents from fault-free ones.

The research addressed in this thesis can supply the basis for future research in the field of cooperative control and fault accommodation of multi-agent systems. A number of potential future works that can be considered as extensions to this research are suggested as follows:

- In this thesis, the reduced model of autonomous underwater vehicles is considered. However, considering six degrees of freedom might improve the performance of the entire group especially in real missions.
- Beside the centralized, decentralized, and semi-decentralized schemes introduced in this thesis, the distributed cooperative scheme could be taken into consideration as well. In distributed scheme, agents communicate the information of their controller in addition to their absolute or relative states which improves the performance of the entire group.
- The heterogeneity considered in this thesis appears in the matrices of the model of agents, nevertheless they have the same dimensions. Considering heterogeneous agents with various degrees of freedom is a challenging topic and it provides the possibility of implementing the proposed control system on the groups of agents that are physically and naturally different or perform in diverse environments simultaneously.
- In this thesis, the stability of a heterogeneous multi-agent system controlled by DSC-based technique is shown. A potential future study on this topic is to find the size of the bound within which the stability of the system is provided, and compare it with the ones

obtained for backstepping and sliding mode control techniques.

- The proposed fault-tolerant scheme can be developed to address other types of faults in actuators such as lock in place, float, and hard over.
- Another possible extension to this study is to consider and accommodate the sensor faults in the network of agents using the proposed cooperative fault-tolerant schemes.

Bibliography

- [1] G. Wen, Z. Duan, W. Yu, and G. Chen, “Consensus of multi-agent systems with nonlinear dynamics and sampled-data information: a delayed-input approach,” *International Journal of Robust and Nonlinear Control*, vol. 23, no. 6, pp. 602–619, 2013.
- [2] Z. M. Li, Y. M. Gao, and Y. F. Niu, “An adaptive cooperative control method for nonlinear system based on directed graph,” *Applied Mechanics and Materials*, vol. 328, pp. 67–71, 2013.
- [3] H. Du, Y. He, and Y. Cheng, “Finite-time cooperative tracking control for a class of second-order nonlinear multi-agent systems,” *Kybernetika*, vol. 49, no. 4, pp. 507–523, 2013.
- [4] H. Li, X. Liao, X. Lei, T. Huang, and W. Zhu, “Second-order consensus seeking in multi-agent systems with nonlinear dynamics over random switching directed networks,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 6, pp. 1595–1607, 2013.
- [5] X. Li, Z. Ma, C. Li, and J. Cao, “Third-order leader-following consensus in a nonlinear

- multiagent network via impulsive control,” in *Abstract and Applied Analysis*, vol. 2013, Hindawi Publishing Corporation, 2013.
- [6] W. Yu, W. Ren, W. X. Zheng, G. Chen, and J. Lü, “Distributed control gains design for consensus in multi-agent systems with second-order nonlinear dynamics,” *Automatica*, vol. 49, no. 7, pp. 2107–2115, 2013.
- [7] W. Li, Z. Chen, and Z. Liu, “Formation control for nonlinear multi-agent systems by robust output regulation,” *Neurocomputing*, vol. 140, pp. 114–120, 2014.
- [8] W. Li, Z. Chen, and Z. Liu, “Leader-following formation control for second-order multiagent systems with time-varying delay and nonlinear dynamics,” *Nonlinear Dynamics*, vol. 72, no. 4, pp. 803–812, 2013.
- [9] D. Meng, Y. Jia, J. Du, and J. Zhang, “High-precision formation control of nonlinear multi-agent systems with switching topologies: A learning approach,” *International Journal of Robust and Nonlinear Control*, 2014.
- [10] V. Gazi and B. Fidan, “Adaptive formation control and target tracking in a class of multi-agent systems: Formation maneuvers,” in *IEEE 13th International Conference on Control, Automation and Systems (ICCAS)*, pp. 78–85, 2013.
- [11] H. Kim, H. Shim, and J. H. Seo, “Output consensus of heterogeneous uncertain linear multi-agent systems,” *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 200–206, 2011.
- [12] G. Rodrigues de Campos, L. Brinon Arranz, A. Seuret, and S.-I. Niculescu, “On the consensus of heterogeneous multi-agent systems: A decoupling approach,” in *Estimation and Control of Networked Systems*, vol. 3, pp. 246–251, 2012.
- [13] Y. Zheng, Y. Zhu, and L. Wang, “Consensus of heterogeneous multi-agent systems,” *IET*

- Control Theory & Applications*, vol. 5, no. 16, pp. 1881–1888, 2011.
- [14] Y. Zheng and L. Wang, “Consensus of heterogeneous multi-agent systems without velocity measurements,” *International Journal of Control*, vol. 85, no. 7, pp. 906–914, 2012.
- [15] Z. Ya-Kun, G. Xin-Ping, and L. Xiao-Yuan, “Finite-time consensus of heterogeneous multi-agent systems,” *Chinese Physics B*, vol. 22, no. 3, p. 038901, 2013.
- [16] H.-Y. Yang, F.-C. Wang, and S.-Y. Zhang, “Consensus of second-order multi-agent systems with nonsymmetric interconnection and heterogeneous delays,” *International Journal of Automation and Computing*, vol. 8, no. 4, pp. 421–428, 2011.
- [17] T. H. Lee, J. H. Park, D. Ji, and H. Y. Jung, “Leader-following consensus problem of heterogeneous multi-agent systems with nonlinear dynamics using fuzzy disturbance observer,” *Complexity*, vol. 19, no. 4, pp. 20–31, 2014.
- [18] Z. Ding, “Consensus output regulation of a class of heterogeneous nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2648–2653, 2013.
- [19] Z. Qu, J. Chunyu, and J. Wang, “Nonlinear cooperative control for consensus of nonlinear and heterogeneous systems,” in *46th IEEE Conference on Decision and Control*, pp. 2301–2308, 2007.
- [20] B. Liu, X. Wang, H. Su, Y. Gao, and L. Wang, “Adaptive second-order consensus of multi-agent systems with heterogeneous nonlinear dynamics and time-varying delays,” *Neurocomputing*, vol. 118, pp. 289–300, 2013.
- [21] W. Sun, Y. Chen, and C. Li, “Multi-group consensus of heterogeneous fractional-order nonlinear agents via pinning control,” in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 377–383, American Society of Mechanical Engineers, 2011.

- [22] E. Peymani, H. F. Grip, A. Saberi, X. Wang, and T. I. Fossen, “H-inf almost output synchronization for heterogeneous networks of introspective agents under external disturbances,” *Automatica*, vol. 50, no. 4, pp. 1026–1036, 2014.
- [23] F. Sun and W. Zhu, “Finite-time consensus for heterogeneous multi-agent systems with mixed-order agents,” *International Journal of Systems Science*, no. ahead-of-print, pp. 1–10, 2013.
- [24] X. Wang, A. Saberi, and T. Yang, “Synchronization in heterogeneous networks of discrete-time introspective right-invertible agents,” *International Journal of Robust and Nonlinear Control*, 2013.
- [25] Y.-P. Tian and Y. Zhang, “High-order consensus of heterogeneous multi-agent systems with unknown communication delays,” *Automatica*, vol. 48, no. 6, pp. 1205–1212, 2012.
- [26] G. E. Dullerud and R. D’Andrea, “Distributed control of heterogeneous systems,” *IEEE Transactions on Automatic Control*, vol. 49, no. 12, pp. 2113–2128, 2004.
- [27] M. Saska, V. Vonásek, T. Krajník, and L. Preucil, “Coordination and navigation of heterogeneous uavs-ugvs teams localized by a hawk-eye approach,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2166–2171, 2012.
- [28] N. Chopra and M. W. Spong, “Output synchronization of nonlinear systems with relative degree one,” in *Recent advances in learning and control*, pp. 51–64, Springer, 2008.
- [29] Z. Qu, J. Chunyu, and J. Wang, “Nonlinear cooperative control for consensus of nonlinear and heterogeneous systems,” in *46th IEEE Conference on Decision and Control*, pp. 2301–2308, 2007.
- [30] P. Wieland, R. Sepulchre, and F. Allgöwer, “An internal model principle is necessary and sufficient for linear output synchronization,” *Automatica*, vol. 47, no. 5, pp. 1068–1074,

2011.

- [31] Y. Zheng, Y. Zhu, and L. Wang, “Consensus of heterogeneous multi-agent systems,” *IET Control Theory & Applications*, vol. 5, no. 16, pp. 1881–1888, 2011.
- [32] H. Kim, H. Shim, and J. H. Seo, “Output consensus of heterogeneous uncertain linear multi-agent systems,” *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 200–206, 2011.
- [33] X.-Y. Lu, H.-S. Tan, S. Shladover, and J. K. Hedrick, “Nonlinear longitudinal controller implementation and comparison for automated cars,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 2, pp. 161–167, 2001.
- [34] Z. Tang, H. Lei, D. Liu, and H. CUI, “Dynamic surface control of nonlinear autopilot for high maneuver missiles,” *Systems Engineering and Electronics*, vol. 30, no. 8, pp. 1523–1525, 2008.
- [35] S. J. Yoo, J. B. Park, and Y. H. Choi, “Adaptive dynamic surface control of flexible-joint robots using self-recurrent wavelet neural networks,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 6, pp. 1342–1355, 2006.
- [36] A. P. Maulana, H. Ohmori, and A. Sano, “Friction compensation strategy via smooth adaptive dynamic surface control,” in *Proceedings of the IEEE International Conference on Control Applications*, vol. 2, pp. 1090–1095, 1999.
- [37] Y. Wang, Q. Wu, and Y. Wang, “Distributed cooperative control for multiple quadrotor systems via dynamic surface control,” *Nonlinear Dynamics*, vol. 75, no. 3, pp. 513–527, 2014.
- [38] J. L. Mathieu and J. K. Hedrick, “Robust multivariable dynamic surface control for position tracking of a bicycle,” in *American Control Conference (ACC)*, pp. 1159–1165,

IEEE, 2010.

- [39] R. M. Gomes, J. B. Sousa, and F. L. Pereira, “Integrated maneuver and control design for roV operations,” in *OCEANS Proceedings*, vol. 2, pp. 703–710, IEEE, 2003.
- [40] A. Girard and J. K. Hedrick, “Dynamic positioning of ships using nonlinear dynamic surface control,” in *Proceedings of the Fifth IFAC Symposium on Nonlinear Control Systems*, pp. 1134–1140, 2001.
- [41] Y. Yang, J. Du, G. Li, W. Li, and C. Guo, “Dynamic surface control for nonlinear dynamic positioning system of ship,” in *Mechanical Engineering and Technology*, pp. 237–244, Springer, 2012.
- [42] B. Ren, H. Pei, Z. Sun, S. S. Ge, and T. H. Lee, “Decentralized cooperative control for swarm agents with high-order dynamics,” in *IEEE International Conference on Automation and Logistics, ICAL’09*, pp. 90–95, 2009.
- [43] M. Chen and B. Jiang, “Adaptive control and constrained control allocation for overactuated ocean surface vessels,” *International Journal of Systems Science*, vol. 44, no. 12, pp. 2295–2309, 2013.
- [44] K. Amezcua S, L. Yan, and W. A. Butt, “Adaptive dynamic surface control for a class of mimo nonlinear systems with actuator failures,” *International Journal of Systems Science*, vol. 44, no. 3, pp. 479–492, 2013.
- [45] X. Zhang and Y. Lin, “Adaptive control for a class of nonlinear time-delay systems preceded by unknown hysteresis,” *International Journal of Systems Science*, vol. 44, no. 8, pp. 1468–1482, 2013.
- [46] S. J. Yoo, J. B. Park, and Y. H. Choi, “Adaptive neural dynamic surface control of nonlinear time-delay systems with model uncertainties,” in *American Control Conference*,

- pp. 6–pp, IEEE, 2006.
- [47] W. A. Butt, L. Yan, and K. Amezcua S, “Adaptive integral dynamic surface control of a hypersonic flight vehicle,” *International Journal of Systems Science*, no. ahead-of-print, pp. 1–12, 2013.
- [48] B. Miao, T. Li, W. Luo, and X. Gao, “Nn based adaptive dynamic surface control for fully actuated auv,” in *Advances in Neural Networks–ISNN*, pp. 79–87, Springer, 2013.
- [49] A. R. Girard and J. K. Hedrick, “Formation control of multiple vehicles using dynamic surface control and hybrid systems,” *International Journal of Control*, vol. 76, no. 9-10, pp. 913–923, 2003.
- [50] Z. Peng, D. Wang, Z. Chen, X. Hu, and W. Lan, “Adaptive dynamic surface control for formations of autonomous surface vehicles with uncertain dynamics,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 513–520, 2013.
- [51] D. Wang, Z. Peng, G. Sun, and H. Wang, “Adaptive dynamic surface control for coordinated target tracking of autonomous surface vehicles using neural networks,” in *IEEE 31st Chinese Control Conference (CCC)*, pp. 2871–2876, 2012.
- [52] H. Wang, D. Wang, and Z. Peng, “Adaptive dynamic surface control for cooperative path following of marine surface vehicles with input saturation,” *Nonlinear Dynamics*, vol. 77, no. 1-2, pp. 107–117, 2014.
- [53] S. P. Berge, K. Ohtsu, and T. I. Fossen, “Nonlinear control of ships minimizing the position tracking errors,” *Modeling Identification and Control*, vol. 20, pp. 177–187, 1999.
- [54] J.-M. Godhavn, “Nonlinear tracking of underactuated surface vessels,” in *Proceedings of the 35th IEEE Conference on Decision and Control*, vol. 1, pp. 975–980, 1996.
- [55] G. J. Toussaint, T. Basar, and F. Bullo, “Tracking for nonlinear underactuated surface

- vessels with generalized forces,” in *Proceedings of the IEEE International Conference on Control Applications*, pp. 355–360, 2000.
- [56] H. Sira-Ramirez, “On the control of the underactuated ship: A trajectory planning approach,” in *Proceedings of the 38th IEEE Conference on Decision and Control*, vol. 3, pp. 2192–2197, 1999.
- [57] K. Y. Pettersen and H. Nijmeijer, “Global practical stabilization and tracking for an underactuated ship—a combined averaging and backstepping approach,” *Modeling Identification and Control*, vol. 20, no. 4, pp. 189–200, 1999.
- [58] K. Y. Pettersen and H. Nijmeijer, “Underactuated ship tracking control: theory and experiments,” *International Journal of Control*, vol. 74, no. 14, pp. 1435–1446, 2001.
- [59] E. Lefeber, K. Y. Pettersen, and H. Nijmeijer, “Tracking control of an underactuated ship,” *IEEE transactions on control systems technology*, vol. 11, no. 1, pp. 52–61, 2003.
- [60] Z.-P. Jiang, “Global tracking control of underactuated ships by Lyapunov’s direct method,” *Automatica*, vol. 38, no. 2, pp. 301–309, 2002.
- [61] T.-C. Lee and Z.-P. Jiang, “New cascade approach for global κ -exponential tracking of underactuated ships,” *IEEE Transactions on Automatic Control*, vol. 49, no. 12, pp. 2297–2303, 2004.
- [62] K. D. Do, Z.-P. Jiang, and J. Pan, “Underactuated ship global tracking under relaxed conditions,” *IEEE Transactions on Automatic Control*, vol. 47, no. 9, pp. 1529–1536, 2002.
- [63] K. Do, Z.-P. Jiang, and J. Pan, “Universal controllers for stabilization and tracking of underactuated ships,” *Systems & Control Letters*, vol. 47, no. 4, pp. 299–317, 2002.
- [64] A. Behal, D. M. Dawson, W. E. Dixon, and Y. Fang, “Tracking and regulation control

- of an underactuated surface vessel with nonintegrable dynamics,” *IEEE Transactions on Automatic Control*, vol. 47, no. 3, pp. 495–500, 2002.
- [65] A. Behal, D. Dawson, B. Xian, and P. Setlur, “Adaptive tracking control of underactuated surface vessels,” in *Proceedings of the IEEE International Conference on Control Applications, (CCA’01)*, pp. 645–650, 2001.
- [66] K. Do, Z.-P. Jiang, and J. Pan, “Robust adaptive path following of underactuated ships,” *Automatica*, vol. 40, no. 6, pp. 929–944, 2004.
- [67] G. Indiveri, M. Aicardi, and G. Casalino, “Robust global stabilization of an underactuated marine vehicle on a linear course by smooth time-invariant feedback,” in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 3, pp. 2156–2161, 2000.
- [68] K. D. Do, J. Pan, and Z.-P. Jiang, “Robust adaptive control of underactuated ships on a linear course with comfort,” *Ocean Engineering*, vol. 30, no. 17, pp. 2201–2225, 2003.
- [69] R. Zhang, Y. Chen, Z. Sun, F. Sun, and H. Xu, “Path control of a surface ship in restricted waters using sliding mode,” *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 722–732, 2000.
- [70] K. Y. Pettersen and E. Lefeber, “Way-point tracking control of ships,” in *IEEE Conference on Decision and Control*, vol. 1, pp. 940–945, 2001.
- [71] P. Encarnacao, A. Pascoal, and M. Arcak, “Path following for autonomous marine craft,” in *5th IFAC Conference on Maneuvering and Control of Marine Craft*, pp. 117–22, 2000.
- [72] P. Encarnacao and A. Pascoal, “3d path following for autonomous underwater vehicle,” in *Proc. 39th IEEE Conference on Decision and Control*, Citeseer, 2000.
- [73] L. Lapierre and D. Soetanto, “Nonlinear path-following control of an auv,” *Ocean Engineering*, vol. 34, no. 11, pp. 1734–1744, 2007.

- [74] D. Soetanto, L. Lapierre, and A. Pascoal, “Adaptive, non-singular path-following control of dynamic wheeled robots,” in *Proceedings. 42nd IEEE Conference on Decision and Control*, vol. 2, pp. 1765–1770, 2003.
- [75] L. Lapierre and B. Jouvencel, “Robust nonlinear path-following control of an auv,” *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 89–102, 2008.
- [76] A. P. Aguiar and J. P. Hespanha, “Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty,” *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1362–1379, 2007.
- [77] K. Do and J. Pan, “Global waypoint tracking control of underactuated ships under relaxed assumptions,” in *Proceedings. 42nd IEEE Conference on Decision and Control*, vol. 2, pp. 1244–1249, 2003.
- [78] K. Do and J. Pan, “Global tracking control of underactuated ships with nonzero off-diagonal terms in their system matrices,” *Automatica*, vol. 41, no. 1, pp. 87–95, 2005.
- [79] T. I. Fossen, M. Breivik, and R. Skjetne, “Line-of-sight path following of underactuated marine craft,” *Proceedings of the 6th IFAC MCMC, Girona, Spain*, pp. 244–249, 2003.
- [80] D. Edwards, T. Bean, D. Odell, and M. Anderson, *A leader-follower algorithm for multiple AUV formations*. IEEE, 2004.
- [81] Y. Wang, W. Yan, and W. Yan, “A leader-follower formation control strategy for auvs based on line-of-sight guidance,” in *ICMA International Conference on Mechatronics and Automation*, pp. 4863–4867, IEEE, 2009.
- [82] J. Yuan, “A feedback linearization based leader-follower optimal formation control for autonomous underwater vehicles,” *Advances in Computer Science and its Applications*, vol. 1, no. 1, pp. 45–48, 2012.

- [83] H. Yang and F. Zhang, “Geometric formation control for autonomous underwater vehicles,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4288–4293, 2010.
- [84] E. Yang, D. Gu, and H. Hu, “Improving the formation-keeping performance of multiple autonomous underwater robotic vehicles,” in *IEEE International Conference on Mechatronics and Automation*, vol. 4, pp. 1890–1895, 2005.
- [85] A. Martins, J. Almeida, and E. Silva, “Coordinated maneuver for gradient search using multiple auvs,” in *IEEE Proceedings on OCEANS*, vol. 1, pp. 347–352, 2003.
- [86] W. H. Huang, “Optimal line-sweep-based decompositions for coverage algorithms,” in *Proceedings on IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 27–32, 2001.
- [87] E. Yang and D. Gu, “Nonlinear formation-keeping and mooring control of multiple autonomous underwater vehicles,” *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 2, pp. 164–178, 2007.
- [88] R. Cui, S. S. Ge, B. Voon Ee How, and Y. S. Choo, “Leader-follower formation control of underactuated auvs with leader position measurement,” in *IEEE International Conference on Robotics and Automation, ICRA’09.*, pp. 979–984, 2009.
- [89] S. Khoo, L. Xie, Z. Man, and S. Zhao, “Observer-based robust finite-time cooperative consensus control for multi-agent networks,” in *4th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1883–1888, 2009.
- [90] S. Khoo, L. Xie, and Z. Man, “Robust finite-time consensus tracking algorithm for multi-robot systems,” *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 219–228, 2009.

- [91] W. Yan, R. Cui, and D. Xu, "Formation control of underactuated autonomous underwater vehicles in horizontal plane," in *IEEE International Conference on Automation and Logistics (ICAL)*, pp. 822–827, 2008.
- [92] D. Zhao, T. Zou, S. Li, and Q. Zhu, "Adaptive backstepping sliding mode control for leader–follower multi-agent systems," *IET control theory & applications*, vol. 6, no. 8, pp. 1109–1117, 2012.
- [93] Q. Jia and G. Li, "Formation control and obstacle avoidance algorithm of multiple autonomous underwater vehicles (auvs) based on potential function and behavior rules," in *IEEE International Conference on Automation and Logistics*, pp. 569–573, 2007.
- [94] S. Emrani, A. Dirafzoon, H. Talebi, S. Y. Nikravesh, and M. Menhaj, "An adaptive leader-follower formation controller for multiple auvs in spatial motions," in *IECON 36th Annual Conference on IEEE Industrial Electronics Society*, pp. 59–64, 2010.
- [95] S. Emrani, A. Dirafzoon, and H. A. Talebi, "Adaptive distributed formation control of multiple autonomous underwater vehicles," in *IEEE International Conference on Control Applications (CCA)*, pp. 693–698, 2011.
- [96] S. Soyly, B. J. Buckham, and R. P. Podhorodeski, "A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation," *Ocean Engineering*, vol. 35, no. 16, pp. 1647–1659, 2008.
- [97] M. L. Corradini and G. Orlando, "A robust observer-based fault tolerant control scheme for underwater vehicles," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 3, pp. 034504:1–11, 2014.
- [98] S. Longhi, A. Monteriu, and M. Vaccarini, "Cooperative control of underwater glider fleets by fault tolerant decentralized mpc," in *Proceedings of the 17th IFAC World Congress*,

Coex, South Korea, 2008.

- [99] Q. Liu and D. Zhu, “Fault-tolerant control of unmanned underwater vehicles with continuous faults: Simulations and experiments,” *International Journal of Advanced Robotic Systems*, vol. 6, no. 4, 2009.
- [100] Q. Liu, D. Zhu, and S. X. Yang, “Unmanned underwater vehicles fault identification and fault-tolerant control method based on fca-cmac neural networks, applied on an actuated vehicle,” *Journal of Intelligent & Robotic Systems*, vol. 66, no. 4, pp. 463–475, 2012.
- [101] D. Zhu, Q. Liu, and Y. Yang, “An active fault-tolerant control method of unmanned underwater vehicles with continuous and uncertain faults,” *International Journal of Advanced Robotic Systems*, vol. 5, no. 4, 2008.
- [102] X.-q. Cheng, J.-y. Qu, Z.-p. Yan, and X.-q. Bian, “H-inf robust fault-tolerant controller design for an autonomous underwater vehicle’s navigation control system,” *Journal of Marine science and Application*, vol. 9, no. 1, pp. 87–92, 2010.
- [103] J.-K. Choi and H. Kondo, “On fault-tolerant control of a hovering auv with four horizontal and two vertical thrusters,” in *OCEANS IEEE-Sydney*, pp. 1–6, 2010.
- [104] M. L. Corradini, A. Monteriu, and G. Orlando, “An actuator failure tolerant control scheme for an underwater remotely operated vehicle,” *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 1036–1046, 2011.
- [105] K. Yang, J. Yuh, and S. K. Choi, “Fault-tolerant system design of an autonomous underwater vehicle odin: an experimental study,” *International Journal of Systems Science*, vol. 30, no. 9, pp. 1011–1019, 1999.
- [106] E. Omerdic and G. Roberts, “Thruster fault diagnosis and accommodation for open-frame underwater vehicles,” *Control Engineering Practice*, vol. 12, no. 12, pp. 1575–1598, 2004.

- [107] M. Benosman and K.-Y. Lum, "Passive actuators' fault-tolerant control for affine nonlinear systems," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 1, pp. 152–163, 2010.
- [108] W. Ren and Y. Cao, *Distributed coordination of multi-agent networks: emergent problems, models, and issues*. Springer, 2010.
- [109] H. K. Khalil and J. Grizzle, *Nonlinear systems*, vol. 3. Prentice hall Upper Saddle River, 2002.
- [110] T. I. Fossen, *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*. Marine Cybernetics, 2002.
- [111] T. I. Fossen, *Guidance and control of ocean vehicles*, vol. 199. Wiley New York, 1994.
- [112] R. Viviani, *Underwater robotic vehicles for infrastructure inspections and environmental monitoring*. PhD thesis, Pisa University, 2008.
- [113] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425–464, 2001.
- [114] M. Qian and P. Kachroo, "Modeling and control of electromagnetic brakes for enhanced braking capabilities for automated highway systems," in *IEEE Conference on Intelligent Transportation System, ITSC'97.*, pp. 391–396, 1997.
- [115] B. Song and J. K. Hedrick, *Dynamic surface control of uncertain nonlinear systems: an LMI approach*. Springer, 2011.
- [116] L. Yi and L. Liu, "Adaptive dynamic surface control for a class of time-varying uncertain nonlinear systems and application," in *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, pp. 2523–2526, IEEE, 2010.

- [117] P. Balaji and D. Srinivasan, “An introduction to multi-agent systems,” in *Innovations in Multi-Agent Systems and Applications-1*, pp. 1–27, Springer, 2010.
- [118] W. Ren and R. Beard, *Distributed consensus in multi-vehicle cooperative control: theory and applications*. Springer, 2007.
- [119] D. Zhu and B. Sun, “Information fusion fault diagnosis method for unmanned underwater vehicle thrusters,” *IET Electrical Systems in Transportation*, vol. 3, no. 4, pp. 102–111, 2013.
- [120] E. Sobhani-Tehrani and K. Khorasani, *Fault diagnosis of nonlinear systems using a hybrid approach*, vol. 383. Springer, 2009.
- [121] M. Blanke and J. Schröder, *Diagnosis and fault-tolerant control*, vol. 2. Springer, 2006.
- [122] R. Ferrari, T. Parisini, and M. M. Polycarpou, “Distributed fault diagnosis with overlapping decompositions: An adaptive approximation approach,” *IEEE Transactions on automation Control*, vol. 54, no. 4, pp. 794–799, 2009.
- [123] Y. Cao, W. Ren, and Z. Meng, “Decentralized finite-time sliding mode estimators and their applications in decentralized finite-time formation tracking,” *Systems & Control Letters*, vol. 59, no. 9, pp. 522–529, 2010.
- [124] S. A. Fjerdigen, E. Kyrkjebø, and A. A. Transeth, “Auv pipeline following using reinforcement learning,” in *41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK)*, pp. 1–8, VDE, 2010.
- [125] H. Yang and F. Zhang, “Geometric formation control for autonomous underwater vehicles,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4288–4293, 2010.