

**ADAPTIVE MODEL-FREE CONTROL AND
LOCALIZATION FOR SINGLE-AGENT AND
MULTI-AGENT NONLINEAR DYNAMIC
SYSTEMS**

ALI SAFAEI

UNIVERSITI SAINS MALAYSIA

2019

**ADAPTIVE MODEL-FREE CONTROL AND LOCALIZATION FOR
SINGLE-AGENT AND MULTI-AGENT NONLINEAR DYNAMIC SYSTEMS**

by

ALI SAFAEI

**Thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy**

June 2019

ACKNOWLEDGMENTS

This PhD thesis is fully supported by a TWAS-USM Postgraduate Fellowship awarded from The World Academy of Sciences. I want to thank my wife, Mahsa, for her patience and being along with me at USM. I must be grateful for the supports and comments received from my supervisor, Dr. Muhammad Nasiruddin bin Mahyuddin. Moreover, I should thank all the anonymous reviewers from the international journals who reviewed my work and provided constructive comments. Last but not least, I want to thank my parents, Parviz and Mitra, to their deep and unconditional supports during my whole life. In addition, a thanks should be given to my parents-in-law, Aziz and Farzaneh to their kind supports.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	ii
TABLE OF CONTENTS	iii
LIST OF TABLES	viii
LIST OF FIGURES.....	xi
LIST OF ABBREVIATIONS	xx
LIST OF SYMBOLS	xxii
ABSTRAK	xxx
ABSTRACT.....	xxxii
CHAPTER 1 – INTRODUCTION.....	1
1.1 Background and motivation	1
1.2 Problem statement	6
1.2.1 Tracking control problem in single-agent dynamic systems	6
1.2.2 Formation-tracking control problem in multi-agent dynamic systems	8
1.2.3 Localization problem in multi-agent dynamic systems	9
1.3 Research objectives.....	10
1.4 Research scopes	11
1.5 Thesis outline	14
CHAPTER 2 – LITERATURE REVIEW	16
2.1 Introduction	16
2.2 Model-free control for single-agent nonlinear dynamic systems.....	17

2.2.1	Model-free control for single-agent systems based on ultra-local model	19
2.2.2	Reinforcement learning as a model-free control algorithm	21
2.3	Cooperative model-free control for multi-agent systems	27
2.3.1	Using artificial neural network as online model-based estimator to generate cooperative control algorithms for multi-agent systems	28
2.3.2	Reinforcement learning for cooperative control algorithms	33
2.3.3	Fuzzy inference system for online estimation in cooperative control algorithms	35
2.3.4	Cooperative control algorithms without model-based estimation tools for multi-agent dynamic systems.....	36
2.4	Cooperative localization for real-time positioning.....	40
2.4.1	Localization methods with GPS	41
2.4.2	Localization methods without GPS	42
2.4.3	Characteristics of cooperative localization.....	43
2.4.4	Cooperative localization algorithms.....	47
2.5	Research gaps.....	53
2.5.1	Gap in the concept of Model-free control.....	53
2.5.2	Gap in the concept of cooperative model-free control	56
2.5.3	Gap in the concept of cooperative localization	57
2.6	Summary	58
CHAPTER 3 – DESIGN AND METHODOLOGY		61
3.1	Introduction	61
3.2	AMFC for a single SISO nonlinear dynamic system	66
3.2.1	Problem definition	66
3.2.2	Stability analysis	68

3.3	AMFC for a single MIMO nonlinear dynamic system	73
3.3.1	Problem definition	74
3.3.2	Stability analysis	79
3.3.3	Optimality analysis.....	86
3.3.4	AMFC algorithm.....	89
3.4	Cooperative AMFC for formation-tracking problem within a multi-agent MIMO dynamic system without inter-agent relative measurements	91
3.4.1	Problem definition	91
3.4.2	Design procedure.....	94
3.4.3	CAMFC-1 algorithm.....	102
3.5	Cooperative AMFC for formation-tracking problem within a multi-agent MIMO dynamic system using inter-agent relative measurements	104
3.5.1	Problem definition	105
3.5.2	Stability analysis	108
3.5.3	Optimality analysis.....	124
3.5.4	CAMFC-2 algorithm.....	127
3.6	ACL algorithm for real-time localization within a team of mobile agents using only one beacon agent	130
3.6.1	Design procedure for the adaptive relative position estimation algorithm	130
3.6.2	Design procedure for the cooperative observer in ACL algorithm .	137
3.6.3	ACL algorithm	141
3.7	Summary	143
	CHAPTER 4 – RESULTS AND DISCUSSION	144
4.1	Introduction	144

4.2	Results for the AMFC algorithm in a single SISO nonlinear dynamic systems	145
4.2.1	Comparison study for an unstable nonlinear system	145
4.2.2	Comparison study for a delay system	152
4.2.3	Application of AMFC algorithm on a Duffing-Holmes chaotic system	157
4.3	Results for the AMFC algorithm in a single MIMO nonlinear dynamic systems	162
4.3.1	Comparison study for an affine nonlinear system	162
4.3.2	Comparison study for a non-affine nonlinear system.....	168
4.3.3	Application of AMFC algorithm to a robotic manipulator.....	172
4.3.4	Application of AMFC algorithm to a wheeled mobile robot	178
4.3.5	Application of AMFC algorithm to a quadrotor	186
4.4	Results for the CAMFC-1 algorithm in a multi-agent nonlinear dynamic system	193
4.5	Results for the CAMFC-2 algorithm in a multi-agent nonlinear dynamic system	200
4.5.1	Comparison study in a consensus problem	201
4.5.2	Comparison study in a formation-tracking problem.....	208
4.5.3	Application of CAMFC-2 algorithm in a team of quadrotors.....	215
4.6	Results for ACL algorithm in a network of mobile agents	226
4.6.1	Comparison study for relative position estimation	226
4.6.2	Comparison study for cooperative localization in a network	232
4.6.3	Application of ACL and CAMFC-2 algorithms in a team of quadrotors	239
4.7	Hardware-in-the-loop test for implementation of the AMFC algorithm in AMRs	243
4.7.1	HIL test setup	243

4.7.2	State observer for AMRs	245
4.7.3	HIL test results	248
4.7.4	Analysis on amplitudes of the external disturbance and the measurement noise	256
4.8	Summary	258
CHAPTER 5 – CONCLUSION AND FUTURE WORKS		260
5.1	Conclusion	260
5.2	Future works	269
REFERENCES		270
APPENDICES		
Appendix A – Affine vs non-affine dynamic systems		
Appendix B – Homogeneous vs heterogeneous multi-agent dynamic systems		
Appendix C – Lipschitz continuous function		
Appendix D – Persistently excitation condition		
Appendix E – Lyapunov stability theorem		
Appendix F – LaSalle-Yoshizawa theorem		
Appendix G – Kronecker product		
Appendix H – M-matrix		
LIST OF PUBLICATIONS		

LIST OF TABLES

	Page	
Table 2.1	Comparison for the characteristics of the major MFC algorithms in the literature for single-agent unknown nonlinear dynamic systems	55
Table 2.2	Comparison for the characteristics of the major cooperative..... MFC algorithms in the literature for multi-agent unknown nonlinear dynamic systems	57
Table 2.3	Comparison for the characteristics of the major cooperative localization algorithms in the literature for a team of mobile agents	58
Table 3.1	AMFC algorithm	90
Table 3.2	CAMFC-1 algorithm	103
Table 3.3	CAMFC-2 algorithm	128
Table 3.4	CAMFC-2 algorithm (continued)	129
Table 3.5	ACL algorithm	142
Table 4.1	Properties of the MFC controllers implemented on an unstable nonlinear system	148
Table 4.2	Properties of the MFC controllers applied to a delay system	157
Table 4.3	Tuning parameters of the AMFC algorithm implemented on a Duffing-Holmes chaotic dynamic system.	158
Table 4.4	Properties of the MFC controllers applied to an affine nonlinear system.	164
Table 4.5	Properties of the MFC controllers applied to a non-affine nonlinear system	169
Table 4.6	Tuning parameters of the AMFC algorithm applied to the robotic manipulator.	175
Table 4.7	Tuning parameters of the AMFC used for the autonomous WMR	182

Table 4.8	Tuning parameters of the AMFC used for the autonomous quadrotor	189
Table 4.9	Tuning parameters of the CAMFC-1 algorithm used at each autonomous quadrotor in a network of four agents.	200
Table 4.10	Tuning parameters for the DCC-1 and CAMFC-2 algorithms implemented at agent i in Section 4.5.1	203
Table 4.11	Tuning parameters for the DCC-2 and CAMFC-2 algorithms implemented at agent i in Section 4.5.2	211
Table 4.12	The tuning parameters for the AMFC and the CAMFC-2 algorithms used for rotational and translational motions, respectively at each quadrotor in the network, as used in Section 4.5.3. ($X_3 = 1e - 6 \times [1; 1; 1]$)	218
Table 4.13	The values of tuning parameters for the controller (as in (4.41)) and the relative position estimating algorithms (as in (4.43) and (3.221))	230
Table 4.14	Comparing the performance of three relative position estimating algorithms with a stationary beacon agent ($p_d^b = [0; 0]$). C_1 and C_2 are defined in (4.44a) and (4.44b).	231
Table 4.15	Comparing the performance of three relative position estimating algorithms with a moving beacon agent ($p_d^b = [2; -3]$). C_1 and C_2 are defined in (4.44a) and (4.44b).	231
Table 4.16	Comparing the performance of the ACL algorithm and the LC algorithm, implemented on a network of non-cooperatively controlled mobile agents	235
Table 4.17	The tuning parameters for the AMFC algorithm at the beacon quadrotor and the rotational motion of follower quadrotors, the CAMFC-2 at translational motion of follower quadrotors and the ACL algorithm in the network. ($X_3 = 1e - 6 \times [1; 1; 1]$)	243
Table 4.18	Definition of the variables used in the Kalman-filter for linear motion of the AMRs (as in (4.49a) and in (4.49b)), to be incorporated in the HIL test	248
Table 4.19	Properties of the AMFC algorithm (as in Algorithm 1 in Table 3.1) with the Kalman-filter utilized in the two HIL tests ($X_8 = 10^{-6} \times \mathbf{1}_8$).	250

Table 4.20 Values of SSE for different amplitudes of the measurement 257
noise in the HIL test ($\alpha_w = 0.1$, $\alpha_q = 0.2$). The parameter
 β_0^n is declared in Section 4.7.3.

Table 4.21 Values of SSE for different amplitudes of the external 257
disturbance in the HIL test ($\beta_0^n = 0.5$). The parameters α_w
and α_q are declared in Section 4.7.1.

LIST OF FIGURES

	Page	
Figure 1.1	The structure of all problems that should be tackled to provide autonomy in AMRs	4
Figure 2.1	The structure of policy iteration method (Kiumarsi et al., 2018)	24
Figure 3.1	The diagram showing the design flow in Chapter 3. The design process is started with presenting AMFC for SISO and MIMO dynamic single-agent systems. Then, it is extended for multi-agent systems with and without interagent relative state measurements. Finally, the cooperative localization algorithm is presented.	65
Figure 3.2	The schematic for operation of the AMFC algorithm for..... a MIMO nonlinear dynamic system. Different blocks are dedicated for implementation of the adaptive laws for online estimation of the unknown linear and nonlinear terms, as well as the online solution of the DRE.	91
Figure 3.3	Schematic for operation of the CAMFC-1 algorithm for a..... multi-agent MIMO nonlinear dynamic system	104
Figure 3.4	Schematic for operation of the CAMFC-2 algorithm for a..... multi-agent MIMO nonlinear dynamic system	130
Figure 3.5	The schematic for adaptive cooperative localization (ACL) algorithm	141
Figure 4.1	Tracking performances of the AMFC, the SMC and iPI..... algorithms, implemented on an unstable nonlinear system; the whole (top) and in detail (bottom)	149
Figure 4.2	Tracking errors of the AMFC, SMC and iPI algorithms,..... implemented on an unstable nonlinear system.	150
Figure 4.3	The control signals of the AMFC, SMC and iPI algorithms, implemented on an unstable nonlinear system.	150
Figure 4.4	Estimated value for unknown nonlinear term, utilizing the AMFC algorithm on an unstable nonlinear system. Stable performance of the adaptive law is disclosed.	151

Figure 4.5	Estimated value for unknown linear term (top); and the main controller gain (bottom), utilizing the AMFC algorithm on an unstable nonlinear system. Stable performances for the adaptive law and the DRE are deduced.	151
Figure 4.6	The FFT for control inputs of the AMFC, SMC and iPI algorithms, implemented on an unstable nonlinear system.	152
Figure 4.7	Tracking performance of AMFC algorithm with the SMC and iPI algorithms in a delay system; the whole (top) and in detail (bottom)	154
Figure 4.8	Tracking errors of the AMFC, SMC and iPI algorithms, implemented on a delay system.	155
Figure 4.9	The control signals of the AMFC, SMC and iPI algorithms, implemented on a delay system.	155
Figure 4.10	Estimated values for unknown nonlinear term in a delay system, utilizing the AMFC algorithm. Stable performance of the adaptive law is disclosed.	156
Figure 4.11	Estimated value for unknown linear term (top); and the main controller gain (bottom), utilizing the AMFC algorithm on a delay system. Stable performances for the adaptive law and the DRE are deduced.	156
Figure 4.12	The FFT of control inputs for AMFC, SMC and iPI algorithms, implemented on a delay system.	157
Figure 4.13	Tracking performance of the Duffing-Holmes chaotic system, utilizing the AMFC algorithm. Bounded tracking error is observed.	159
Figure 4.14	Values for the control signal (top); and the controller gain, utilizing the AMFC algorithm on a Duffing-Holmes chaotic system. The values are all bounded.	160
Figure 4.15	Estimated values for unknown linear term (top); and unknown nonlinear term (bottom), utilizing the AMFC algorithm on a Duffing-Holmes chaotic system. The online estimated parameters are bounded.	161
Figure 4.16	Tracking performance of the AMFC and MFC-1 algorithms, implemented on an affine nonlinear system.	165

Figure 4.17	The control signals of the AMFC and MFC-1 algorithms,.....	166
	implemented on an affine nonlinear system. The control signals are bounded and the AMFC algorithm has faster reaction to the tracking error, leading to more control efforts at the initial times.	
Figure 4.18	Estimated values for the unknown linear and nonlinear terms in an affine nonlinear system, utilizing the AMFC algorithm.	167
	All the estimated values are bounded.	
Figure 4.19	Tracking performance of the AMFC and MFC-2 algorithms,.....	169
	implemented on a non-affine nonlinear system. Tracking errors are bounded and faster convergence is achieved with AMFC.	
Figure 4.20	The control signals of the AMFC and MFC-2 algorithms,	170
	implemented on a non-affine nonlinear system. The values are bounded and the AMFC algorithm has faster reaction to the tracking error.	
Figure 4.21	Estimated values for unknown linear and nonlinear terms in a..... non-affine nonlinear system, utilizing the AMFC algorithm.	171
	The estimated values are bounded. It can be seen that the estimated linear terms drive the updating process of the main controller gains.	
Figure 4.22	Schematic of a 2 DOF robotic manipulator in the 2D space	172
	(Spong et al., 2006)	
Figure 4.23	Values for the states (top) and the tracking error (bottom)..... for the robotic manipulator, utilizing the AMFC algorithm.	175
	The tracking error is bounded around origin with very small boundaries.	
Figure 4.24	Estimated values for unknown nonlinear (top) and unknown	176
	linear (bottom) terms for the robotic manipulator, utilizing the AMFC algorithm. The estimate values for unknown terms are bounded.	
Figure 4.25	Values for the control signals (top); and the Controller gains,..... i.e. P matrix (bottom) for the robotic manipulator, utilizing the AMFC algorithm. The control signal is bounded and values for the controller gains are updated based on the estimated values of unknown linear terms.	177
Figure 4.26	Schematic of a WMR	182

Figure 4.27	Tracking performance for the autonomous WMR by applying the AMFC algorithm. The tracking error is bounded around origin.	183
Figure 4.28	Control inputs (angular speeds of wheels) for the autonomous WMR and the values for P matrix, utilizing the AMFC algorithm. The control signals and the controller gains are bounded.	184
Figure 4.29	Estimated values for the unknown linear and nonlinear terms for the autonomous WMR, utilizing the AMFC algorithm. The estimated values for unknown terms are bounded.	185
Figure 4.30	Schematic of a quadrotor	189
Figure 4.31	Tracking performance for the autonomous quadrotor by applying the AMFC algorithm. Tracking errors are all bounded around the origin.	190
Figure 4.32	Control inputs (angular speeds of electric motors) for the autonomous quadrotor and the values for P matrix, utilizing the AMFC algorithm.	191
Figure 4.33	Estimated values for the unknown linear and nonlinear terms for the autonomous quadrotor, utilizing the AMFC algorithm.	192
Figure 4.34	Communication graph for a team of four autonomous quadrotors used for simulation study of CAMFC-1. Here, the 0-indexed node represents the virtual leader node. The communication links in the network are undirected.	194
Figure 4.35	Positions of four cooperative autonomous quadrotors in the 3D space, utilizing the CAMFC-1 algorithm.	195
Figure 4.36	Positions of four cooperative autonomous quadrotors in the 2D space, utilizing the CAMFC-1 algorithm.	195
Figure 4.37	Tracking errors for four cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.	196
Figure 4.38	Estimated values for reference trajectory at four cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.	196
Figure 4.39	Consensus errors for observing the reference trajectory at four cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.	197

Figure 4.40	Estimated values for the formation parameters at four.....	197
	cooperative autonomous quadrotors, utilizing the CAMFC-1	
	algorithm.	
Figure 4.41	Adapted values for unknown linear terms at a team of.....	198
	cooperative autonomous quadrotors, utilizing the CAMFC-1	
	algorithm.	
Figure 4.42	Adapted values for unknown nonlinear terms at a team of.....	198
	cooperative autonomous quadrotors, utilizing the CAMFC-1	
	algorithm.	
Figure 4.43	Controller gains for each of the cooperative autonomous	199
	quadrotors, utilizing the CAMFC-1 algorithm.	
Figure 4.44	Control signals at each of the cooperative autonomous	199
	quadrotors, utilizing the CAMFC-1 algorithm.	
Figure 4.45	The consensus errors for all agents using the DCC-1.....	204
	algorithm (top) and the CAMFC-2 algorithm (bottom) . The	
	convergence is provided in both cases.	
Figure 4.46	The values for first states of all agents using the DCC-1	204
	algorithm (top) and the CAMFC-2 algorithm (bottom).	
Figure 4.47	The values for control signals at all agents using DCC-1	205
	algorithm (top) and the CAMFC-2 algorithm (bottom).	
Figure 4.48	The estimated values for unknown nonlinear terms at all	205
	agents in the problem of Section 4.5.1, using the CAMFC-2	
	algorithm.	
Figure 4.49	The estimated values for unknown linear terms at all agents	206
	in the problem of Section 4.5.1, using the CAMFC-2	
	algorithm.	
Figure 4.50	The values for main controller gains at all agents in the.....	206
	problem of Section 4.5.1, using the CAMFC-2 algorithm.	
Figure 4.51	The estimated values for the leader state at all agents in the	207
	problem of Section 4.5.1, using the CAMFC-2 algorithm.	
Figure 4.52	The estimated values for the leader control input (which is.....	207
	the derivative of the leader state) at all agents in the problem	
	of Section 4.5.1, using the CAMFC-2 algorithm.	

- Figure 4.53 The consensus errors of the first and second states for all 212 agents using the DCC-2 algorithm (top) and the CAMFC-2 algorithm (bottom). The convergence is provided in both cases.
- Figure 4.54 The values of first and second states for all agents using 212 the DCC-2 algorithm (top) and the CAMFC-2 algorithm (bottom). The convergence is provided in both cases, while CAMFC-2 has more over-shoot in the response, but faster rising-time.
- Figure 4.55 The values of first and second control inputs for all 213 agents using the DCC-2 algorithm (top) and the CAMFC-2 algorithm (bottom). External disturbance has more effect on the DCC-2 algorithm and leads to severe fluctuations in the steady-state condition.
- Figure 4.56 The estimated values for unknown nonlinear terms at all 213 agents in the problem of Section 4.5.2, using the CAMFC-2 algorithm.
- Figure 4.57 The estimated values for unknown linear terms at all agents 214 in the problem of Section 4.5.2, using the CAMFC-2 algorithm.
- Figure 4.58 The values of main controller gains at all agents in the 214 problem of Section 4.5.2, using the CAMFC-2 algorithm.
- Figure 4.59 The estimated values of formation variables at Agent-4 in the 215 problem of Section 4.5.2, using the CAMFC-2 algorithm.
- Figure 4.60 The 3D position of quadrotors in the network using the 219 CAMFC-2 algorithm.
- Figure 4.61 The X-Y-Z positions of quadrotors in the network using the 219 CAMFC-2 algorithm.
- Figure 4.62 The consensus errors for translational motion of quadrotors 220 in the network using the CAMFC-2 algorithm.
- Figure 4.63 The tracking errors for rotational motion of quadrotors in the 220 network using the AMFC algorithm.
- Figure 4.64 The control signals for speed of electric motors at all 221 quadrotors in the network using CAMFC-2 algorithm.

Figure 4.65	The set-points for the Euler angles of quadrotors in the network using CAMFC-2 algorithm.	221
Figure 4.66	The estimated values of unknown nonlinear terms for translational motion of quadrotors in the network using CAMFC-2 algorithm.	222
Figure 4.67	The estimated values of unknown nonlinear terms for rotational motion of quadrotors in the network using AMFC algorithm.	222
Figure 4.68	The estimated values of unknown linear terms for translational motion of quadrotors in the network using CAMFC-2 algorithm.	223
Figure 4.69	The estimated values of unknown linear terms for rotational motion of quadrotors in the network using AMFC algorithm.	223
Figure 4.70	The values of main controller gains for translational motion of quadrotors in the network using CAMFC-2 algorithm.	224
Figure 4.71	The values of main controller gains for rotational motion of quadrotors in the network using AMFC algorithm.	224
Figure 4.72	The estimated values of formation variables for translational motion of the quadrotors in the network using CAMFC-2 algorithm.	225
Figure 4.73	The estimated values of the leader states using CAMFC-2 algorithm.	225
Figure 4.74	The estimated values of the leader control inputs using CAMFC-2 algorithm.	226
Figure 4.75	The absolute positioning estimation errors for all of the three presented relative position estimation algorithms.	231
Figure 4.76	The communication graph among the agents in the network for implementing the ACL algorithm. The network incorporates only one beacon agent.	233
Figure 4.77	Comparing the position estimation errors for the ACL and LC algorithms, implemented on a network of non-cooperatively controlled mobile agents (Agent-1 and Agent-2)	236

Figure 4.78	Comparing the position estimation errors for the ACL..... 236 and LC algorithms, implemented on a network of non-cooperatively controlled mobile agents (Agent-3 and Agent-4)
Figure 4.79	Values for the consensus error of the cooperative observer..... 237 in the ACL algorithm, implemented on a network of non-cooperatively controlled mobile agents (Agent-1 and Agent-2)
Figure 4.80	Values for the consensus error of the cooperative observer..... 237 in the ACL algorithm, implemented on a network of non-cooperatively controlled mobile agents (Agent-3 and Agent-4)
Figure 4.81	Comparing the estimated absolute positions with the ACL 238 and LC algorithms, in a network of non-cooperatively controlled mobile agents (Agent-1 and Agent-2)
Figure 4.82	Comparing the estimated absolute positions with the ACL 238 and LC algorithms, in a network of non-cooperatively controlled mobile agents (Agent-3 and Agent-4)
Figure 4.83	Position estimation errors for ACL algorithm used in a team..... 240 of cooperatively controlled quadrotors. Here, $\zeta = p - \hat{p}$. Refer to Section 3.6 and Section 4.6.1 for more information.
Figure 4.84	Localization consensus errors for ACL algorithm used in a 241 team of cooperatively controlled quadrotors.
Figure 4.85	Formation-tracking errors for a team of quadrotors..... 241 controlled with the CAMFC-2 algorithm using the relative position estimating algorithm for providing the relative position values.
Figure 4.86	Absolute position of the quadrotors and the leader quadrotor..... 242 in a network controlled with CAMFC-2 algorithm, using ACL algorithm for localization.
Figure 4.87	Estimated values for absolute position of the quadrotors in..... 242 the network using ACL algorithm.
Figure 4.88	The schematic of HIL test setup for real-time evaluation of 245 the AMFC algorithm on AMRs
Figure 4.89	The x-y position and z-axis rotation of the autonomous 250 WMR in the HIL test.

- Figure 4.90 The errors in linear and angular motions of the autonomous 251
WMR (top); and the control signals for the electric motors
of the WMR (bottom), in the HIL test for evaluating the
proposed AMFC algorithm.
- Figure 4.91 The values of P matrix for linear and angular motions of the 251
autonomous WMR (as updated by (3.49)), in the HIL test for
evaluating the proposed AMFC algorithm.
- Figure 4.92 The adapted values of unknown linear terms for linear and 252
angular motions of the autonomous WMR (as updated by
(3.50b)), in the HIL test for evaluating the proposed AMFC
algorithm.
- Figure 4.93 The adapted values of unknown nonlinear terms for the 252
linear and angular motions of the autonomous WMR (as
updated by (3.50a)), in the HIL test for evaluating the
proposed AMFC algorithm.
- Figure 4.94 The autonomous quadrotor position in 3D space and its Euler 253
angles, in the HIL test for evaluating the proposed AMFC
algorithm.
- Figure 4.95 The errors in position of the autonomous quadrotor (top);..... 254
and the control signals for the electric motors speeds
(bottom), in the HIL test for evaluating the proposed AMFC
algorithm. The values of errors are bounded with absolute
upper-bound of 0.6 .
- Figure 4.96 The values of P matrix for the autonomous quadrotor (as..... 254
updated by (3.49)), in the HIL test for evaluating the
proposed AMFC algorithm.
- Figure 4.97 The adapted values of unknown linear terms for the 255
autonomous quadrotor (as updated by (3.50b)), in the HIL
test for evaluating the proposed AMFC algorithm.
- Figure 4.98 The adapted values of unknown nonlinear terms for the 255
autonomous quadrotor (as updated by (3.50a)), in the HIL
test for evaluating the proposed AMFC algorithm.

LIST OF ABBREVIATIONS

AMR	Autonomous Mobile Robot
MFC	Model-Free Control
GPS	Geographical Positioning System
WMR	Wheeled Mobile Robot
UUB	Uniformly Ultimately Bounded
UWB	Ultra-Wide Band
IMU	Inertial Measurement Unit
SISO	Single-Input Single-Output
MIMO	Multi-Input Multi-Output
RL	Reinforcement Learning
LQR	Linear Quadratic Regulator
ANN	Artificial Neural Network
FIS	Fuzzy Inference System
PE	Persistent excitation
DRE	Differential Riccati Equation
LMI	Linear Matrix Inequality
HJB	Hamilton-Jacobi-Bellman
LS	Least Squares
EKF	Extended Kalman Filter

AMFC	Adaptive Model-Free Control
CAMFC	Cooperative Adaptive Model-Free Control
DCC	Distributed Cooperative Control
ACL	Adaptive Cooperative Localization
LC	Linear Convex
HIL	Hardware-In-the-Loop

LIST OF SYMBOLS

n_c	number of real control variables in a MIMO dynamic system
n_o	number of output variables in a MIMO dynamic system
n_s	number of states in a MIMO dynamic system
n_p	number of position states for a mobile agent
n	number of states in a MIMO dynamic system
N	number of agents in a multi-agent dynamic system
x_s	state of a SISO dynamic system
x	state vector for MIMO dynamic system
x^i	state vector for agent i in a multi-agent dynamic system
x^0	state vector of the leader agent in a multi-agent dynamic system
x_p^i	position states of the i th agent in a multi-agent dynamic system
x_p^b	position states of the beacon agent in a multi-agent dynamic system
x_c	vector of combined states of all agents in a multi-agent dynamic system
u_s	control input for a SISO dynamic system
u_0	real control vector for MIMO dynamic system
u^*	vector of auxiliary virtual control variables for a MIMO dynamic system
u	general vector of control variables for a MIMO dynamic system
u^0	vector of control variables at the leader agent in a multi-agent dynamic system
u^i	vector of general control variables at agent i in a multi-agent dynamic system

u_c	vector of combined control variables for all agents in a multi-agent dynamic system
b	scalar input gain in a SISO dynamic system
a	unknown system gain in a SISO dynamic system
C_0	output matrix for a MIMO dynamic system
B_0	input gain matrix for real control variables in a MIMO dynamic system
B	general input gain matrix for real and virtual control variables in a MIMO dynamic system
A	unknown system matrix for a MIMO dynamic system
A^i	unknown system matrix at agent i in a multi-agent dynamic system
A_c	matrix of combined unknown linear terms for all of the agents in a multi-agent dynamic system
f_s	unknown nonlinear function including a linear-in-state term for a SISO dynamic system
h	unknown nonlinear function in a SISO dynamic system
f_0	vector of unknown nonlinear functions including linear-in-state terms and real control variables for a MIMO dynamic system
f	vector of unknown nonlinear functions including real control variables for a MIMO dynamic system
g	general vector of unknown nonlinear functions for a MIMO dynamic system
g^i	vector of unknown nonlinear functions at agent i in a multi-agent dynamic system

g_c	vector of combined unknown nonlinear terms for all of the agents in a multi-agent dynamic system
y_s	output of a SISO dynamic system
y	output of a MIMO dynamic system
y_s^d	scalar desired output for a SISO dynamic system
y_0^d	vector of desired values for the states of a MIMO dynamic system
y^d	general vector of desired values for the states of a MIMO dynamic system including the virtual control variables
v	estimated value of the desired output for a dynamic system, as used in the sliding-mode observer
k_1	first scalar gain in the sliding-mode observer to estimate the desired output
k_2	second scalar gain in the sliding-mode observer to estimate the desired output
r	constant gain used in AMFC for SISO dynamic system
q	constant gain used in the online update of the controller gain for AMFC in SISO dynamic system
p	controller gain in AMFC for SISO dynamic system
R	positive definite gain matrix used in AMFC for MIMO dynamic system
Q	positive definite gain matrix used in the online update of the controller gains for AMFC in MIMO dynamic system
P	positive definite matrix for controller gains in AMFC for MIMO dynamic system

R^i	positive definite gain matrix used in CAMFC-2 algorithm at i th agent in a multi-agent dynamic system
Q^i	positive definite gain matrix used in the online update of the controller gains for CAMFC-2 algorithm at i th agent in a multi-agent dynamic system
P^i	positive definite matrix for controller gains at i th agent in a multi-agent dynamic system
\hat{a}	estimated unknown linear term in AMFC for SISO dynamic system
\hat{h}	estimated unknown nonlinear term in AMFC for SISO dynamic system
\hat{A}	estimated unknown linear terms in AMFC for MIMO dynamic system
\hat{g}	estimated unknown nonlinear terms in AMFC for MIMO dynamic system
\hat{A}^i	estimated values for unknown system matrix at agent i in a multi-agent dynamic system
\hat{g}^i	vector of estimated values for unknown nonlinear functions at agent i in a multi-agent dynamic system
\hat{x}_i^0	vector of estimated states of the leader agent at the i th agent in a multi-agent dynamic system
\hat{x}_p^i	estimated absolute position of agent i in a multi-agent dynamic system
${}^i\hat{\eta}^j$	vector of estimated values for the formation variables of agent j at the i th agent in a multi-agent dynamic system
\hat{q}^i	vector of estimated values for the leader control inputs at agent i in a multi-agent dynamic system
e_s	tracking error for a SISO dynamic system

ϑ	time integral of the tracking error for a SISO dynamic system
ς	joint cost function for tracking objective in a SISO dynamic system
e_0	tracking error considering only the states of a MIMO dynamic system
e	general tracking error considering the states and the virtual control variables for a MIMO dynamic system
e_p	distance estimation error for two mobile agents
e^i	consensus error at agent i for formation-tracking problem in a multi-agent dynamic system
e_c	vector of combined consensus errors for formation-tracking problem in a multi-agent dynamic system
ζ	time integral of the general tracking error for a MIMO dynamic system
ζ^i	time integral of the consensus error at agent i for the formation-tracking problem in a multi-agent dynamic system
ζ_c	vector of combined time integral values for the consensus errors at all of the agents in a multi-agent dynamic system
σ	joint tracking error for tracking objective of a MIMO dynamic system
\tilde{x}_i^0	consensus error for estimating the states of the leader agent at the i th agent in a multi-agent dynamic system
ε^i	consensus error for estimating the formation variables at agent i in a multi-agent dynamic system
τ^i	consensus error for estimating the leader control inputs at agent i in a multi-agent dynamic system

τ_p^i	consensus error for observing the absolute position of agent i in a multi-agent dynamic system
γ_1	adaptation gain for estimating unknown nonlinear term in AMFC for SISO dynamic system
γ_2	adaptation gain for estimating unknown linear term in AMFC for SISO dynamic system
ϖ_1	leakage gain for estimating unknown nonlinear term in AMFC for SISO dynamic system
ϖ_2	leakage gain for estimating unknown linear term in AMFC for SISO dynamic system
Γ_1	diagonal adaptation gain matrix for estimating unknown nonlinear terms in AMFC for MIMO dynamic system
Γ_2	diagonal adaptation gain matrix for estimating unknown linear terms in AMFC for MIMO dynamic system
ρ_1	leakage gain for estimating unknown nonlinear terms in AMFC for MIMO dynamic system
ρ_2	leakage gain for estimating unknown linear terms in AMFC for MIMO dynamic system
μ	scalar gain defining the learning rate in the cooperative observer for estimating the formation variables at each of the agent in a multi-agent dynamic system
λ	scalar gain defining the learning rate in the cooperative observer for estimating the leader states at each of the agent in a multi-agent dynamic system

λ_1	scalar gain defining the learning rate in the cooperative observer for estimating the leader control variables at each of the agent in a multi-agent system
λ_2	scalar gain defining the learning rate in the cooperative observer for estimating the absolute positions at each of the agent in a multi-agent system
κ	scalar gain for inclusion of the time integral of the consensus errors into the control signal in CAMFC-2 algorithm for a multi-agent system
α_p	scalar gain for tuning the adaptive relative position estimating algorithm
X_M	vector of maximum absolute values for the states dynamics at the leader agent in a multi-agent system
Υ^M	vector of maximum absolute values for the formation variables in a multi-agent system
M_p^b	vector of maximum absolute values for the velocity of beacon agent
$\hat{\Omega}^i$	estimated matrix for all of the formation variables at agent i in a multi-agent system
\mathcal{A}	adjacency matrix for the communication graph of a multi-agent system
\mathcal{B}	pinning gain matrix for the communication graph of a multi-agent system
\mathcal{B}_b	beacon pinning gain matrix in the communication graph of a multi-agent system
\mathcal{D}	in-degree matrix for the communication graph of a multi-agent system
\mathcal{L}	Laplacian matrix for the communication graph of a multi-agent system
\mathcal{H}	joint matrix representing the Laplacian and the leader pinning gains for the communication graph of a multi-agent system

\mathcal{H}_b	joint matrix representing the Laplacian and the beacon pinning gains for the communication graph of a multi-agent system
a_{ij}	element located at the i th row and the j th column of the adjacency matrix in a multi-agent system
β_i	the i th diagonal element in the pinning gain matrix corresponding to the agent i in a multi-agent system
β_i^b	the i th diagonal element in the beacon pinning gain matrix corresponding to the agent i in a multi-agent dynamic system
η^i	vector of time-varying formation parameters corresponding to the agent i in a multi-agent dynamic system
z^i	vector of changed states at agent i in a multi-agent dynamic system
z_c	vector of combined changed states for all of the agents in a multi-agent dynamic system
P_r	relative position between two mobile agents
V_r	relative velocity between two mobile agents
d_r	relative distance between two mobile agents
δ_{ib}	relative position between agent i and the beacon agent in a multi-agent dynamic system
δ_{ij}	relative position of agents i to the agent j in a multi-agent dynamic system
$\Delta_i^{\mathbb{P}_i}$	vector for position of agent i in a local coordinates frame

**KAWALAN BEBAS-MODEL SUAI DAN PENYETEMPATAN UNTUK AGEN
TUNGGAL DAN SISTEM DINAMIK TIDAK LURUS AGEN-PELBAGAI**

ABSTRAK

Dalam tesis ini, penyelesaian bersepada yang terdiri daripada kawalan bebas-model dan algoritma penyetempatan model dibentangkan untuk menangani masalah pengesanan dalam sistem dinamik bukan lurus yang sepenuhnya tidak diketahui, masalah pengesanan-pembentukan dalam sistem dinamik bukan lurus yang sepenuhnya tidak diketahui dan masalah penyetempatan kerjasama untuk pasukan ajen mudah alih. Algoritma kawalan bebas-model yang dirumuskan, tidak bergantung pada ciri penghampiran sejagat rangkaian saraf tiruan atau pengiraan berdasarkan regresi. Dengan penyesuaian dalam talian unsur-unsur dalam matriks sistem, persamaan Riccati perbezaan digunakan untuk mengemaskini keuntungan pengawal utama dalam talian. Berdasarkan hasil keputusan simulasi untuk sistem ejen tunggal, dipamerkan bahawa isyarat kawalan yang lancar dihasilkan dengan menggunakan pengawal bebas adaptif yang dicadangkan (memaparkan bilangan undang-undang adaptif yang lebih sedikit) berbanding pengawal PI pintar dan pengawal mod gelongsor. Nilai 49% yang lebih rendah daripada fungsi kos dicapai menggunakan pengawal yang dicadangkan terhadap pengawal dalam literasi yang menggunakan rangkaian saraf buatan. Algoritma kawalan bebas-model kerjasama yang dibentangkan untuk sistem berbilang-agen menggunakan kaedah yang teragih. Salah satu pengendali kerjasama yang dibentangkan bergantung pada pengukuran mutlak tempatannya, sementara pengawal kerjasama kedua memerlukan pengukuran nisbi antara agen dalam rangkaian. Berdasarkan keputusan simulasi pada sistem berbilang-agen, secara nisbahnya 5.5% dan 51.5% nilai-nilai yang lebih rendah untuk

fungsi kos dicapai untuk pengawal bebas model penyesuaian koperatif yang dicadangkan berbanding dua kaedah canggih yang lain dalam literasi. Selain itu, algoritma penganggar kedudukan nisbi penyesuaian dibangunkan untuk menganggarkan kedudukan nisbi di antara setiap pasangan agen mudah alih, tanpa mengukur sudut galas. Jarak nisbi dan halaju nisbi harus diukur antara agen mudah alih. Berdasarkan keputusan simulasi untuk anggaran kedudukan nisbi antara dua ejen mudah alih, ralat pengiraan secara nisbahnya 34% lebih rendah dicapai dalam senario kes terburuk, berbanding dua penganggar posisi nisbi yang lain. Algoritma pengiraan kedudukan nisbi dibangunkan dalam pemerhati kerjasama yang diagihkan untuk menghasilkan algoritma penyetempatan kerjasama penyesuaian untuk menentukan kedudukan nisbi dan mutlak setiap ejen mudah alih dalam rangkaian dengan hanya satu ejen matarah, dan mempunyai minimum kemungkinan bilangan perhubungan komunikasi di kalangan ejen. Ralat lebih daripada 93% penyetempatan diperoleh pada semua ejen dalam rangkaian menggunakan algoritma penyesuaian penyesuaian yang dicadangkan dengan kaedah penyetempatan cembung lelurus.

**ADAPTIVE MODEL-FREE CONTROL AND LOCALIZATION FOR
SINGLE-AGENT AND MULTI-AGENT NONLINEAR DYNAMIC SYSTEMS**

ABSTRACT

In this thesis, a unified solution comprising model-free control and localization algorithms is presented to address the tracking problem in single-agent completely unknown nonlinear dynamic systems, the formation-tracking problem in multi-agent completely unknown nonlinear dynamic system, and the cooperative localization problem for a team of mobile agents. The formulated model-free control algorithms, neither rely on the universal approximation characteristic of the artificial neural networks nor regressor-based approximation. By online adaptation of the elements in a system matrix, the differential Riccati equation is employed for online updating of the main controller gains. Based on the simulation results for single-agent systems, it is shown that smoother control signals are generated using the proposed adaptive model-free controller (featuring fewer number of adaptive laws) compared to an intelligent PI controller and a sliding-mode controller. A relatively 49% lower value of a cost function is achieved using the proposed controller against the controllers in the literature utilizing the artificial neural networks. The cooperative model-free control algorithms presented for multi-agent systems employ distributed methods. One of the presented cooperative controllers rely on its local absolute measurements, while the second cooperative controller needs inter-agent relative measurements in the network. Based on the simulation results on multi-agent systems, relatively 5.5% and 51.5% lower values for a cost function are achieved for the proposed cooperative adaptive model-free controller comparing with two state-of-the-art methods in the literature. Furthermore, an adaptive relative position estimating algorithm is

developed to estimate the relative position among each pair of mobile agents, without the requirement for bearing angle measurements. The relative distance and relative velocity should be measured between the mobile agents. Based on the simulation results for the relative position estimation between two mobile agents, relatively 34% lower estimation error is achieved in the worst case scenario, comparing to two other relative position estimators. The developed relative position estimation algorithm is incorporated within a distributed cooperative observer to generate an adaptive cooperative localization algorithm for determining the relative and absolute positions of each mobile agent in a network with only one beacon agent, and having the minimum possible number of communication links among the agents. Relatively more than 93% localization error is provided at all of the agents in the network utilizing the proposed adaptive localization algorithm with respect to a linear convex localization method. Throughout the thesis, the simulation results for application of the proposed control and localization algorithms on autonomous mobile robots with especial concern on quadrotors are presented.

CHAPTER 1

INTRODUCTION

1.1 Background and motivation

The dream of mechanical devices that can perform the works which a human is capable of, has much older history than a person can expect. One of the earliest samples of mechanical devices that would be described as a *robot* dated back to 400BC, when the Greek philosopher *Archytas* invented a steam-powered pigeon. In the year 1960, the first actual device called the Computerized Numerical Control machines, was invented to automate manufacturing tasks. After that, the *Unimate* as one of the first products in the category of *industrial robots*, was implemented on a General Motors plant in 1961 (Heintschel-von-Heinegg et al., 2018). The Robotic Industrial Association defined the term industrial robot as an automatically controlled, reprogrammable, multi purpose manipulator which is programmable in three or more axes for use in the industrial automation applications (Kumar et al., 2008). In the past decades, robot arms or *manipulators* delivered a high growth-rate industry. Normally, they are fixed to a specific location in the assembly line and perform repetitive tasks such as spot welding and painting. Despite all of their success and benefits, these industrial robots suffer from a fundamental disadvantage which is lack of *mobility* (Siegwart & Nourbakhsh, 2004).

Around the year 1990, a demand for robots which can handle the missions in hazardous environments as well as everyday routine tasks, causes further developments into field and service *mobile robots*. Furthermore, some manufacturers

provide *autonomous* functions such as keeping fixed distance and automatic parking to the mobile robots (Heintschel-von-Heinegg et al., 2018).

These days by superb enhancements in performance of microprocessors, sensor modules and the technology of battery as the main source of energy for mobile robots, the autonomous mobile robots (AMRs) are considered among the most essential tools in different parts of industry, from the manufacturing companies and agriculture to health care and the media (Siegwart & Nourbakhsh, 2004). Autonomous quadrotors, wheeled mobile robots and underwater autonomous vehicles are among the well-known AMRs. Several applications such as rescue and survival missions, underwater and space expeditions, aerial capturing and photography, intelligent agriculture management, transportation of the goods in the warehouses, carrying the medical assets to remote areas and also the air show entertainment, are good examples of the importance of AMRs in our today society (Siegwart & Nourbakhsh, 2004).

The emerging applications of AMRs rely on *autonomy*, as the main distinctive feature. The term autonomy is referring to *decisional autonomy*, meaning that an entity can decide what to do by itself (Heintschel-von-Heinegg et al., 2018). An autonomous mobile robot, for example, is able to accomplish the assigned task with the least possible human intervention/supervision. To provide AMRs with the proper level of autonomy, a range of problems from automatic control and state observations to localization, path planning and obstacle avoidance, needs to be resolved.

Navigation and path tracking is one of the main objectives for designing an AMR. At the lowest level in a conceptual paradigm presented in Fig. 1.1, AMRs require the

capability of automatic *control* (Kelly, 2013). In fact, the control algorithms ensure that the AMRs are able to reach its navigation or follow the path tracking objective. To achieve that, an AMR requires to have some methods for *state estimation* (i.e *observers*), in accomplishing the objective of locating the AMR, its corresponding speed and acceleration. The state estimation methods incorporate the robot internal data brought by some on-board *proprioceptive sensors*. Measuring the AMR position, velocity or acceleration are among the proprioceptive sensory data. *Localization* as the capability to locate the AMR in its local or global indoor/outdoor environment is categorized as a state estimation method. In general, the accurate position information of static and dynamic objects with reference to a fixed origin point has been an interesting ongoing discussion and academic debate (Mao & Fidan, 2009; Safavi & Khan, 2017). The localization problem can be addressed in the sea, air or on the ground and each of the environments has its own challenges and constraints to be tackled.

Besides the internal states, the AMRs need to be aware about their environment. This feature which is named as *perception* can be provided by the data gathered using the on-board *exteroceptive sensors*. Ranging from the objects in the environment as well as generating a map from the immediate surroundings are fallen in the category of exteroceptive observations (Kelly, 2013). Another aspect of autonomous mobility for AMRs is *path planning*. Path planning is a capability to predict the consequences of the possible alternative series of actions, so as to choose the most appropriate action at the current situation (Kelly, 2013). *Obstacle avoidance* is a feature that can be provided for an AMR by a suitable path planning generated based on awareness of the static and dynamic objects in the environment utilizing the exteroceptive sensory data.

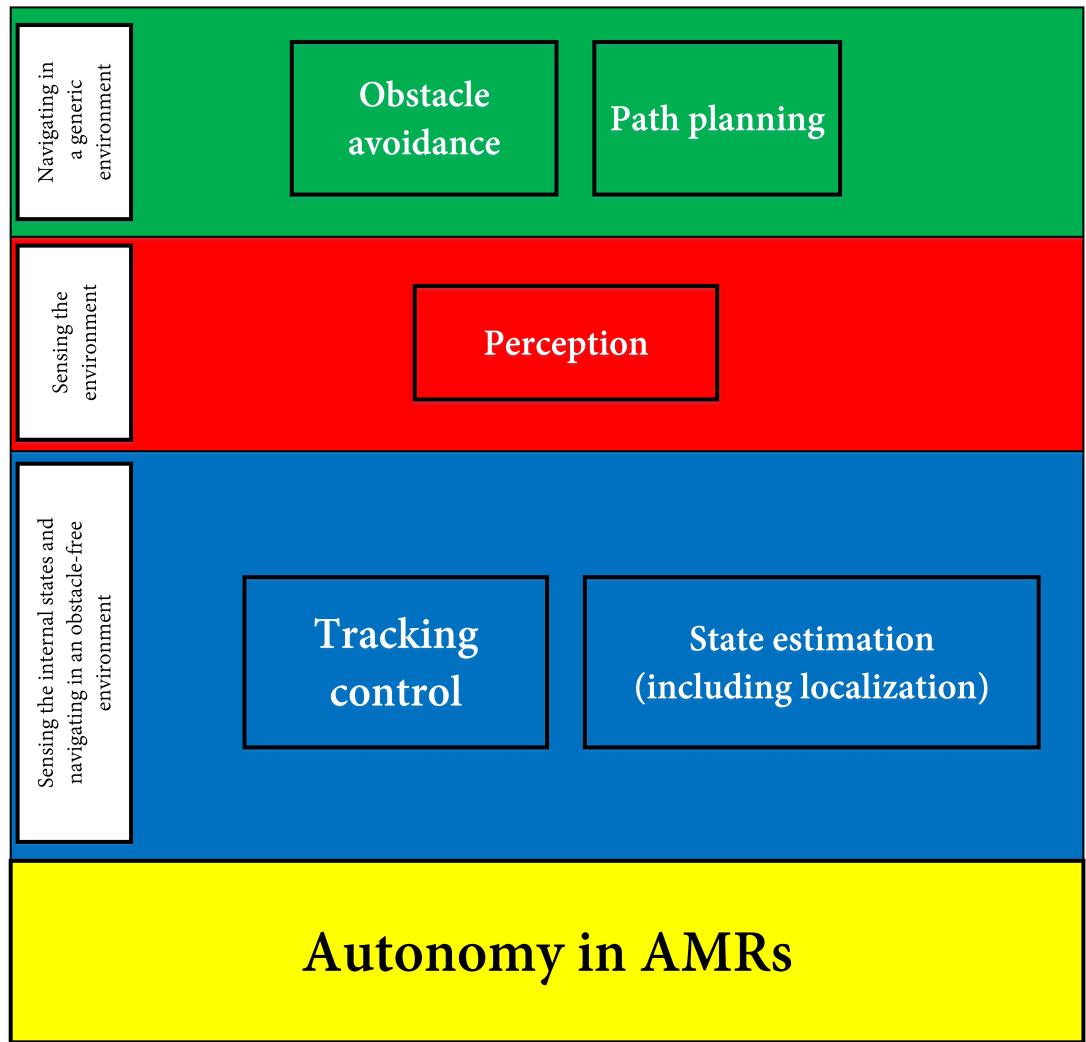


Figure 1.1: The structure of all problems that should be tackled to provide autonomy in AMRs

The above challenges need to be resolved for every individual AMRs. Although a single AMR is capable of completing diverse range of autonomous missions, there are some limitations that prevent the full potential usage of AMRs. The main constraint is the limited amount of energy provided by the batteries to the AMRs. This leads to the limited time of operation for a single AMR (Li & Duan, 2015). The battery technology is still emerging and several improvements are predicted in future (Rao & Shivakumar, 2018). In addition, since the remote wireless systems have a limited range of operation (Wanasinghe et al., 2015), a single AMR mobility is limited by the short distance from the central control station. This would limit the operational board of the AMRs.

Concerning by the above limitations, a *multi-agent system* constructed by several AMRs can be a reasonable short-term solution in order to utilize the AMRs toward their maximum potential extent (Li & Duan, 2015). A multi-agent system of AMRs can be considered as a network or a team of multiple (more than one) AMRs having a mutual objective and operating in a *cooperative* manner. The cooperative operation utilizes the inter-agent communication links within the network. Utilizing a team of AMRs instead of using an individual AMR, one can expect an increased number of operations during a fixed time window, as well as extended range of operation. Moreover, some specific missions can be performed only by using a team of AMRs. Carrying large cargos, satellite formation flying, and providing a night show to a large number of audiences with the purpose of entertainment are among these specific tasks that require a team of AMRs rather than only a single AMR (Li & Duan, 2015).

Considering a multi-agent system of AMRs, the capabilities listed in Fig. 1.1 should be brought to any individual robot included in the team. In this regard, the

concepts of *cooperative control*, *cooperative observer* and *cooperative localization* are proposed for a team of AMRs. It is shown that, it would be more beneficial to have the above problems be solved by designing some *decentralized* algorithms which are implemented locally at each agent, without any need for receiving/transmitting signals at every AMR in the team from/to a *centralized* control station (Li & Duan, 2015);(Lewis et al., 2014).

1.2 Problem statement

Among the features presented in Fig. 1.1, tracking control and localization problems are stated as the main problems to be investigated in this thesis. They are considered as the main subjects, since they are the core and basic problems for providing high level of autonomy in AMRs. Other features can be resolved by designing the appropriate algorithms providing the solutions to the tracking control and localization problems.

1.2.1 Tracking control problem in single-agent dynamic systems

Generally in a tracking control problem, the control signals are designed based on the dynamic system of the AMR. However, the exact dynamical system structure and its parameters are often unknown. The assumption derived thereafter may not be suitable all the time (Wang et al., 2011; Younes et al., 2016). For the nonlinear dynamic system of the AMRs, the values for mass, moment of inertia and even physical dimensions can change during the operation in different working conditions. These parameters can be considered as the unknown parameters in the internal dynamics of AMRs. Moreover, the unknown external disturbances such as a force

imposed by human or an external object and forces generated by wind and other severe environmental conditions, can change the structure of the nonlinear dynamic systems (Wang et al., 2011). In addition, if there are several classes for our AMRs (corresponding to the different sizes and the applications) and the existing tracking controller depends on the AMR internal dynamics and its related parameters, then one should design different control signals with different controller gains for each class.

In this regard, it would be great if one can design a controller that handles the tracking control problem by adapting itself in an online manner with the changes in internal dynamics structure of the nonlinear plant (including the unknown parameters) and the unknown external disturbances (Ioannou & Fidan, 2006). This is the place where the classic adaptive control algorithms, data-driven control algorithms and the recently-proposed model-free control (MFC) algorithms come into considerations.

The model-free algorithms are control methods in which the structure of dynamic system is supposed to be completely unknown (Hou & Jin, 2014). This is the major difference between the MFC with the classic adaptive control algorithms including the model-reference adaptive controllers and the adaptive pole-placement controllers presented by Ioannou and Fidan (2006). The later algorithms assume the structure of the dynamic system is completely known and only some unknown parameters need to be adapted online. Instead, the MFC algorithms consider a general structure for any unknown dynamic system (either linear or nonlinear) and use the measured input-output data from the system to estimate the unknown dynamic system in online manner and then generate the control policy for handling the tracking control problem. According to different methods used for estimating the unknown dynamics,

different MFC algorithms are proposed in the literature, which are discussed in Section 2.2. Most of these solutions use regressor-based estimators which incorporate artificial neural networks or fuzzy inference systems for parameter estimation.

1.2.2 Formation-tracking control problem in multi-agent dynamic systems

Recalling the benefits provided by a multi-agent system of AMRs over a single AMR as presented in Section 1.1, a great attention in the literature has been paid to the problems of controlling a multi-agent network of AMRs ranging from consensus to flocking movements, formation control and leader-following (Lewis et al., 2014; Li & Duan, 2015). The formation control problem is an interesting issue in diverse fields of technology including biology, automatic control and robotics, which requires each agent in the network to track a reference trajectory, while building a desired formation topology in cooperation with the other agents (Li & Duan, 2015).

Similar to the case of a single dynamic system, the issues of having unknown internal dynamics and unknown external disturbances exist for formation-tracking problem in a multi-agent dynamic system including a team of AMRs, as well. Further details can be found in Section 2.3. Hence, an extended synthesis for the MFC algorithms is vital for formation-tracking control problem in a team of AMRs. In this synthesis, the impact of inter-agent communications on the design of the cooperative control protocol should be taken into account. This impact can be further understood by the use of *graph theory* to represent the interactions among the agents (Jadbabaie et al., 2003).

In addition, the relative position information of each agent to some of the agents in

the network needs to be determined for computing the control signals in most of the cooperative MFC algorithms. This challenge may lead to the localization problem for a single dynamic system and also the cooperative localization problem in a multi-agent dynamic system.

1.2.3 Localization problem in multi-agent dynamic systems

As mentioned before in Section 1.1, estimating the position of stationary or mobile agents in a local or global frames is named as localization problem. One of the easy and cost-effective solution to the localization problem is to use Geographical Positioning System (GPS). It is shown that the position data provided by the commercial GPS modules in open sky conditions has the mean accuracy of 4.9 meters in radius (Diggelen & Enge, 2015). This level of accuracy can be acceptable in the localization task involving large dynamic systems like airplane, ship, car and landmark. For a small dynamic systems like AMRs, this amount of error adversely affects the localization and consequently the control tasks. In addition, GPS signals are not available inside buildings and also in jammed areas, due to non-line of sight condition (Safavi & Khan, 2017).

Hence, several solutions in the literature are provided to improve the accuracy of the positioning results in the localization problems. The solutions can be categorized as the methods based on GPS data and the methods which do not use the GPS data. Detailed list of solutions in this area is provided in Section 2.4. Among them, the cooperative localization algorithms are proposed to improve the positioning accuracy using the available information in a network of agents.

In a cooperative localization problem, the relative position (or absolute position in some solutions) of each agent in a multi-agent system is computed online in a two dimensional (i.e. 2D) or three dimensional (i.e. 3D) environments. Since the inter-agent communications are required for implementing the cooperative localization algorithms, these types of algorithms are interesting when we are dealing with a network of agents. Specifically, the cooperative localization algorithms are meaningful for a team of AMRs, where the information about the inter-agent relative distances and velocities in the network can be provided using the on-board sensors at each of the agents.

1.3 Research objectives

Recalling the problems stated in Section 1.2, the research objectives of this thesis are listed as follows;

- developing an adaptive MFC algorithm for tracking control problem in single-agent dynamic systems with completely unknown nonlinear dynamics, including single AMRs; a method is required to update the main controller gains in the adaptive MFC algorithm, utilizing the online estimated values for unknown dynamics, which in turn should be estimated online by regressor-free adaptive laws;
- developing decentralized cooperative adaptive MFC algorithms with and without accessing to inter-agent relative state measurements (or estimations), so as to achieve formation-tracking and consensus objectives in unknown multi-agent nonlinear dynamic systems, including a team of AMRs;

- developing an adaptive cooperative localization algorithm for real-time local and global positioning within a network of mobile agents with one beacon agent and minimum number of communication links among the agents, including a team of AMRs; the method should operate without requiring the relative bearing angles, utilizing only the measurements on relative distance and relative velocity vector.

All of the above objectives are requested for generic nonlinear dynamic systems with special applications to AMRs.

1.4 Research scopes

Based on the stated problems and the provided research objectives, the major scope of the current research is to use the adaptive methods to design the control and localization algorithms for unknown nonlinear dynamic systems including AMRs. Here, nonlinear dynamic systems are concerned, since almost all of the AMRs can be modeled in nonlinear dynamic systems, in general point of view. In addition, the proposed solutions are subject to this constraint that the external disturbances on the dynamic systems are bounded.

This thesis focuses on the solutions for continuous-time dynamic systems. Although discrete-time dynamic systems are beyond the scope of this thesis, proliferation of the presented solutions on a discrete-time setup can be made in the future work.

In this research, the adaptive methods utilize gradient descent update laws and provide a rate for online changing of some variables to construct online estimations for

the corresponding unknown terms. The input-output data of the dynamic systems are incorporated in the adaptive methods to form the adaptation process.

Moreover, *Lyapunov* and *LaSalle-Yoshizawa* stability theorems are used throughout the thesis in order to provide the proofs for stability and convergence of the algorithms. Incorporating the LaSalle-Yoshizawa theorem in the proofs, leads to uniformly ultimately bounded (UUB) convergence of the algorithms. UUB is relatively, a less conservative convergence property as compared to asymptotic, exponential and finite-time convergence property. However, the condition does not require to know about the system dynamics and disturbance other than its relative upper bounds. This characteristic offers convenience in particular when tracking performance is desired over the estimation performance. Moreover, hardware implementation of the algorithm with limited computational resources may benefit from this paradigm.

In addition, the adaptive methods are designed so as to include *leakage* and *signum* terms to confirm the robustness of the algorithms. Robustness is an interesting property that adaptive algorithms might have when dealing with the unknown terms.

For the solutions proposed to the cooperative network of dynamic agents (such as a team of AMRs), the concept of communication graph is adopted from the graph theory to incorporate the properties of the existing communications in the network into the design procedure. The focus of the current thesis is on the homogeneous networks with fixed communication graph, while the results can be extended with few modifications for the heterogenous multi-agent systems and the networks with

time-varying communication graph. This is due to the fact that the dynamics of all agents are assumed to be completely unknown throughout the thesis. Furthermore, all of the proposed algorithms in the thesis are accompanied with application results in AMRs.

Disclaimer. It should be noted that, in the current thesis, the simulation results (including the results from the hardware-in-the-loop test) of the proposed algorithms on a robotic manipulator, a wheeled mobile robot, a quadrotor and a network of four quadrotors are presented to show that the algorithms can be applied on the real platforms. Implementation of the proposed algorithm on a hardware-in-the-loop approach is not to validate, but rather to illustrate the efficacy of its practical feasibility.

1.5 Thesis outline

Recalling the problem statements and the research objectives of the thesis, an in-depth literature review is presented in Chapter 2. The literature review is presented in three different subsections corresponding to the three problems stated in Section 1.2. At the end of Chapter 2, research gaps are defined and the motivation for the designing of the algorithms are presented.

Chapter 3 of the thesis is dedicated to the design process and proofs for the algorithms. The chapter includes all the required definitions, propositions and assumptions for designing and presenting the algorithms. In this regard, a novel adaptive MFC algorithm is developed for the tracking control problem of a generic completely unknown continuous-time single-agent nonlinear dynamic system in Section 3.2 and Section 3.3, for single-input single-output and multi-input multi-output cases, respectively. Later in Section 3.4 and Section 3.5, the adaptive MFC algorithm has been extended for deriving the decentralized cooperative algorithms to solve the formation-tracking and consensus problems in multi-agent dynamic systems with unknown internal dynamics and unknown bounded external disturbances. The proposed adaptive cooperative algorithms are distinct from each other, based on the availability of the inter-agent relative state measurements in the network. At last in Section 3.6, an adaptive cooperative localization algorithm is developed for both relative and absolute positioning of the agents inside a network of mobile dynamic agents. Throughout Chapter 3, all the algorithms are provided with the adequate mathematical proofs.

In Chapter 4, several numerical simulation results, including the comparative

studies to the previously published solutions and also applications to real platforms, are provided for the developed algorithms. The results for the adaptive MFC in single-agent systems are presented in Section 4.2 and Section 4.3. Furthermore, Section 4.4 and Section 4.5 consist of the simulation results for cooperative adaptive MFC on multi-agent dynamic systems. The simulation results for the adaptive cooperative localization algorithms are presented in 4.6. In addition, the results for performing the hardware-in-the-loop test for application of the proposed adaptive MFC algorithm on a wheeled mobile robot and a quadrotor are presented in Section 4.8.

The thesis is concluded in Chapter 5 and some suggestions for the future investigations are made. The solutions presented in this thesis might be seen as an integrated package to provide basis of high level of autonomy for any types of AMRs with unknown internal dynamics, working under unknown external bounded disturbances.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, a review over the MFC algorithms investigated for single-agent and multi-agent dynamic systems is presented. Moreover, the proposed solutions in the literature to the localization problem are reviewed and a history of the cooperative localization algorithms is presented. Firstly, the MFC algorithms which are designed based on the generic ultra-local structure for the dynamics of unknown nonlinear systems are presented in Section 2.2. It is mentioned that the ultra-local model can be considered as the linearly-parameterized model for alternative representation of the nonlinear plants. This section also includes all of the modifications provided recently to the original MFC algorithm, including the fuzzy and sliding-mode extensions of the algorithm. Furthermore in that section, it is followed by introduction of the reinforcement learning algorithms as online optimal adaptive solutions for tracking control problem in unknown single-agent nonlinear dynamic systems. The use of reinforcement learning in context of MFC algorithms provides the optimality feature to the algorithm.

In Section 2.3, this is followed by a review on the history of the cooperative control algorithms on multi-agent nonlinear dynamic systems and the development of the distributed cooperative model-free control algorithms for multi-agent systems with unknown nonlinear dynamics is presented. The importance of the communication graph in the design process of the cooperative control algorithms is

declared and the recently-proposed solutions to address the unknown nonlinear terms in the agents dynamics are presented. Moreover, it is mentioned that the most of the state-of-the-art cooperative control algorithms in the literature rely on the artificial neural networks to provide the online estimations for unknown nonlinear terms or the control signals directly.

Later in Section 2.4, all of the available solutions for the localization problems in both outdoor and indoor environments are reviewed. It is shown that the cooperative localization algorithms are among the most emerging solutions. Different cooperative localization algorithms are presented and reviewed with more details.

Finally, according to the reviewed literature, the research gaps in the three stated problems of this thesis are provided in Section 2.5.

2.2 Model-free control for single-agent nonlinear dynamic systems

For the tracking control problem in a system with partially or completely unknown dynamics, one can design a model-based or model-free control algorithms. In a *model-based* control or estimation algorithm, the unknown dynamic model is represented in *linearly parameterized (LP)* format as follows (Na et al., 2015)

$$f_{lp}(x) = \phi_{lp}(x)\theta_{lp}, \quad (2.1)$$

where $f_{lp} \in \mathbb{R}^{N_e \times 1}$ is the unknown dynamic system which is going to be represented by LP, $\phi_{lp} \in \mathbb{R}^{N_e \times p}$ includes the known basis functions (or simply *regressors*) and $\theta_{lp} \in \mathbb{R}^{p \times 1}$ is the vector of unknown parameters needs to be estimated. Here, $x \in \mathbb{R}^{n \times 1}$ is the

vector of system's states, and n is the number of states, p is the number of unknown parameters and N_e is the number of dynamic equations in the system. According to the *persistently excitation (PE)* requirement (refer to *Appendix D* for more information), convergence of the adaptive laws in all model-based estimation algorithms would be achieved, if and only if the input signal is *sufficiently rich (SR)* (Ioannou & Fidan, 2006).

There are several investigations among the adaptive data-driven control algorithms in the literature, which are designed based on an LP model for the unknown dynamics and using the model-based estimation methods for online adaptation (Wang et al., 2018, 2017; Yu et al., 2016; Zhao et al., 2017). These work adopt the use of artificial neural network (ANN) to represent the nonlinear term, assumed to be LP. The weights to the corresponding basis functions in ANN are then estimated online. Although ANN approach does not require parametric information about a system model to be known, the adaptive laws are model-based estimation algorithms and the predefined regressor parameters need to be persistently excited. In addition, the parameter estimation error and its rate depend on the proper selection of the number of neurons (or neural nodes) used in the ANN; such that the error of parameter estimation converges to zero, if the numbers of neurons reach to infinity (Lewis et al., 2014).

Moreover, in the aforementioned algorithms, the main controller gains should be determined off-line; posing limitations to the development of achieving fully autonomous dynamic systems. Recently, model-free control approaches have become interesting methods in academic and industrial points of view for tracking problems in dynamic systems (Madadi & Söfftker, 2015).

2.2.1 Model-free control for single-agent systems based on ultra-local model

In 2013, Fliess proposed the MFC technique for single-input single-output (SISO) nonlinear dynamic systems for the first time (Fliess & Join, 2013). The model-free techniques proposed by Fliess, include the *intelligent P* (*iP*), the *intelligent PI* (*iPI*), the *intelligent PD* (*iPD*) and the *intelligent PID* (*iPID*) controllers. Presenting a general dynamic system in form of an *ultra-local* model as

$$y = F + \alpha u , \quad (2.2)$$

where $y \in \mathbb{R}$ is the system output, $u \in \mathbb{R}$ is the system controller and $F \in \mathbb{R}$ is the unknown nonlinear dynamics of the system; the iPID controller can be expressed as follows

$$u = -\frac{\hat{F} - \ddot{y}_d - K_p e - K_i \int e - K_d \dot{e}}{\alpha} , \quad (2.3a)$$

$$\hat{F} = \frac{1}{\tau_0} \int_{t-\tau_0}^t (\ddot{y}_d - \alpha u + K_p e + K_i \int e + K_d \dot{e}) d\tau . \quad (2.3b)$$

Here, \hat{F} is the estimated value for unknown nonlinear term, $\alpha > 0$ is the constant parameter as the gain for relation between the magnitude of y and u (assumed to be known), $\tau_0 > 0$ is a constant number of previous steps used in estimation of \hat{F} and $y_d \in \mathbb{R}$ is the desired trajectory for the system output, while $e = y_d - y$. In addition, K_p , K_i and K_d are the constant positive gains for the proportional, integral and derivative parts of the iPID controller, which should be determined manually. The controllers iP, iPI and iPD are defined similarly (Fliess & Join, 2013). Note that these controllers are considered as the model-free controllers, since the corresponding control signals are defined free from the unknown nonlinear dynamics of the system (i.e. F). Consequently, the model-free algorithm does not require nonlinear term to be

LP, since the estimation of F is achieved by merely the use of readily available information such as tracking error, past control input and \ddot{y}_d . In a generic point of view, the ultra-local model in (2.2), which is an *affine* dynamic model with regards to the control input variable (refer to *Appendix A* for more information), includes a lumped unknown nonlinear function and *a priori*-known constant input gain. Estimation of the unknown nonlinear term is performed by a simple algebraic equation utilizing the past input-output data, as in (2.3b).

In (Thabet et al., 2014), the ultra-local model in (2.2) is transformed to a linear time-invariant (LTI) state-space dynamic system, and then an adaptive observer is proposed for estimating the system states and the system's unknown nonlinear term (i.e. F). In that work, since the online estimation process is performed by a model-based estimation algorithm, there is a requirement of PE condition for the regressor parameters. In a similar approach, a method is presented by Carrill and Rotella (2015) for estimating the unknown nonlinear term and the unknown input gain utilizing a parametric model, where the PE condition is required for confirming the convergence.

Later, several applications of the MFC are provided in practical systems (Cao et al., 2016; Lafont et al., 2015; Younes et al., 2016; Zhou et al., 2016). The applications include the fault accommodation in a greenhouse, AC/DC converter for on-board battery charger, permanent magnet drive systems and autonomous quadrotors. In these applications, the MFC algorithm is modified accordingly so as to comply with the requirements and constraints of the corresponding dynamic system. Latest applications of the MFC algorithms include an acute inflammation process (Bara et

al., 2018) and a vapour-compression refrigeration process (Yu et al., 2018).

In 2016, Roman et al. compared the performance of the model-free controller with the virtual reference feedback tuning technique. Then, (Roman et al., 2017, 2018) presented a fuzzy version of the MFC algorithm with application to a twin-rotor set-up. The formulation of the fuzzy MFC algorithm based on an *iPD* is proposed as follows

$$u = -\frac{\hat{F} - \dot{y}_d - \phi_{fuzz}}{\alpha}, \quad (2.4a)$$

$$\hat{F} = \frac{1}{\tau_0} \int_{t-\tau_0}^t (\dot{y}_d - \alpha u) d\tau, \quad (2.4b)$$

where $\phi_{fuzz} \in \mathbb{R}$ is the fuzzy control signal generated based on the fuzzy membership functions defined for the tracking error e and its derivative signal. Note that in this algorithm, the parameters in the fuzzy membership functions need to be defined manually, where an off-line optimization process is used for this purpose (Roman et al., 2017). This approach was further extended by incorporating a sliding-mode MFC algorithm and its experimental validation (Percup et al., 2017). Similar to (2.4a), an additional term including the sliding-mode term is added in the sliding-mode MFC algorithm.

2.2.2 Reinforcement learning as a model-free control algorithm

Optimality is another feature which has been already brought to the MFC algorithms. In work by Roman et al. (2015), the MFC is formulated in a LTI system and an optimal MFC is proposed for a multi-input multi-output (MIMO) dynamic system. Utilizing the linear quadratic regulator (LQR) technique, an optimal term is included in the proposed solution by Roman et al. (2015). Since the optimal control

problem is solved off-line, the main controller gains need to be tuned by the control designer manually, before deploying the algorithm.

Incorporation of the optimal control theory into the area of MFC algorithms for dynamic systems, has led to proposing the *reinforcement learning (RL)* techniques for tracking control problem (Lewis et al., 2012; Song et al., 2017). Several RL algorithms have been used for solving the optimal tracking control problem in discrete and continuous-time systems with partially or completely unknown linear and nonlinear dynamics (Kiumarsi et al., 2018; Zhang et al., 2017; Zhu & Zhao, 2018). The optimal control policy for a linear continuous-time system can be designed by the solution of a *Hamilton-Jacobi-Bellman* (HJB) equation. Based on the HJB equation (Lewis et al., 2012), the optimal control signal u_{op} for a dynamic system should satisfy

$$0 = \min_{u=u_{op}} \{r(e, u) + \frac{d}{dt}J(e)\}, \quad (2.5)$$

where $r(\cdot) \in \mathbb{R}$ and $J(\cdot) \in \mathbb{R}$ are *value (cost-to-go)* and *utility* functions, respectively. It is shown that the value function for a linear dynamic system can be represented by a quadratic function of the system's states. This property leads to a straight-forward formulation of the optimal controller for linear systems, i.e. the LQR technique (Lewis et al., 2012). In contrary, since it is not possible to express the value function of a nonlinear dynamic system in form of a general quadratic function of the states, the solution of HJB equation in nonlinear systems is not as straight-forward as in the case of linear systems (Lewis & Vrabie, 2009). It is observed that the HJB equation for a nonlinear dynamic system is quadratic in gradient of the value function. In other words, the corresponding HJB equation is a nonlinear differential equation. The

iterative algorithms should be utilized for solving the HJB equation in nonlinear dynamic systems (Lewis & Vrabie, 2009). Suppose a generic nonlinear dynamic system with n states and m control inputs can be defined as

$$\dot{x} = f(x) + g(x)u , \quad (2.6)$$

where $x \in \mathbb{R}^{n \times 1}$ is the system's states, $u \in \mathbb{R}^{m \times 1}$ is the control inputs, $f(x) \in \mathbb{R}^{n \times 1}$ is a vector including the unknown nonlinear functions in the system dynamics and $g(x) \in \mathbb{R}^{n \times m}$ is the input matrix. Then, the *policy iteration* method for the nonlinear dynamic system defined in (2.6), is an iterative method utilized for solving the HJB equation. The method consists of two steps, as policy evaluation (Lewis & Vrabie, 2009; Vamvoudakis & Lewis, 2010),

$$0 = r(x, u_i) + (\nabla J_i(x))^T (f(x) + g(x)u_i(x)) ; \quad (2.7)$$

and policy improvement

$$u_{i+1}(x) = -\frac{1}{2}R^{-1}g^T(x)\nabla J_i(x) , \quad (2.8)$$

where $R \in \mathbb{R}^{m \times m}$ is a positive definite matrix and $(\nabla J_i(x)) \in \mathbb{R}^{n \times 1}$ is the gradient of the value function at the i th iteration. As can be seen in Fig. 2.1, the policy iteration method has an actor/critic RL structure, where the policy evaluation is handled by a critic agent and the policy improvement is performed by an actor agent. Usually, the actor and critic agents are generated by two separate ANNs or fuzzy inference systems (FISs). Note that the policy iteration algorithm in (2.7) and (2.8) are presented

for continuous-time systems. The similar approaches have been used for discrete-time systems and approximate dynamic programming and Q-learning algorithms are proposed (Lewis & Vrabie, 2009; Luo et al., 2016).

Besides the policy iteration algorithm, *value iteration* algorithms are also investigated as the second type of the RL solutions for optimal tracking control problem (Xiao et al., 2017; Zhang et al., 2017). But, most of the proposed RL solutions in the literature for optimal tracking control problem are categorized in policy iteration group (Kiumarsi et al., 2018). For value iteration algorithms, only the final converged optimal control law can be utilized to control the nonlinear dynamic system and all the controllers during the iteration procedure might be invalid. Therefore, the computational efficiency of the value iteration algorithm is low and requires infinite time to obtain the optimal control law. On the other hand, it is proven that the policy iteration algorithm converges in finite time and each of the iterative controllers achieved during the iteration process can stabilize the nonlinear dynamic system (Liu & Wei, 2014; Wei et al., 2016).

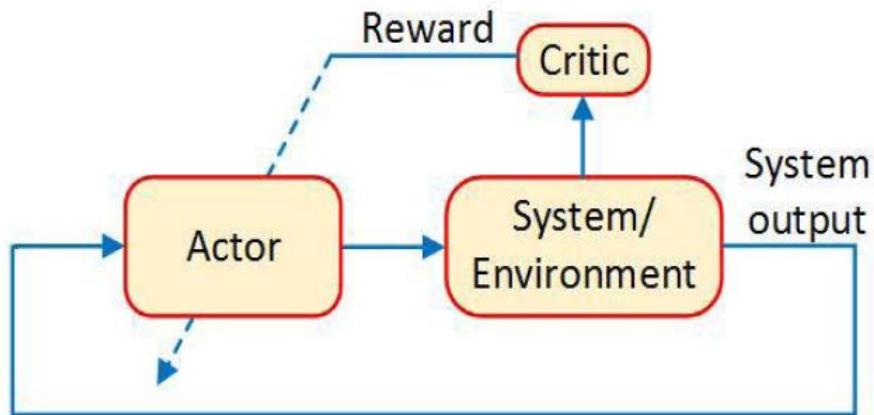


Figure 2.1: The structure of policy iteration method (Kiumarsi et al., 2018)

As it is observed in (2.7), the complete information of the nonlinear dynamic system (i.e. $f(\cdot)$ and $g(\cdot)$) is required in a normal policy iteration algorithm to generate the optimal controller. As mentioned before in Section 1, such requirement is hardly met for practical applications due to the unknown internal dynamics and external disturbances. Besides, the policy iteration algorithm is an off-line process. It is desirable to extend and apply it online in an adaptive control mechanism (Vamvoudakis et al., 2014), coined as *Integral Reinforcement Learning* (IRL) technique

$$J = \hat{W}_1^T \phi(x) \quad (2.9a)$$

$$u = -\frac{1}{2} T_0 R^{-1} g^T(x) \nabla \phi^T(x) \hat{W}_2, \quad (2.9b)$$

where $T_0 \in \mathbb{R}^+$ is the time interval of the past input-output data that is going to be incorporated in the design, $\phi \in \mathbb{R}^{N_n \times 1}$ is the regressor parameter which includes the activation functions for the N_n neurons in the ANNs, and $\hat{W}_1 \in \mathbb{R}^{N_n \times 1}$ and $\hat{W}_2 \in \mathbb{R}^{N_n \times 1}$ are the weights of the neurons in the critic and actor ANNs, respectively. The values for \hat{W}_1 and \hat{W}_2 are adapted online based on the input-output data in time interval $[t - T_0, t]$. The full definitions are available in (Vamvoudakis et al., 2014).

As it can be seen in (2.9b) by applying the IRL technique, one can avoid the dependency of the optimal control design procedure on the drift dynamics f , but the input gain matrix g is still necessary (Zhu et al., 2016). This issue should be addressed by proposing a model-free online adaptive control to solve the optimal tracking problem for nonlinear systems with completely unknown dynamics (Zhu et al., 2016). In this regard, an ANN as an adaptive identifier is suggested by Lv et al. (2016) for online estimation of the completely unknown dynamic systems (i.e. both

$g(x)$ and $f(x)$ in (2.6)). Then, it is merged with another ANN for estimating the value function, in order to produce the optimal adaptive control signals. On the other hand, the approach proposed by Zhu et al. (2016) is to estimate the whole components of control policy directly, by assuming that they can be linearly parameterized as follows

$$J = \hat{W}_c^T \phi_c(x) \quad (2.10a)$$

$$u = \hat{W}_a^T \phi_a(x), \quad (2.10b)$$

where $\hat{W}_c \in \mathbb{R}^{N_c \times 1}$ and $\hat{W}_a \in \mathbb{R}^{m \times N_a}$ are the weights for the critic and actor ANNs, and $\phi_c \in \mathbb{R}^{N_c \times 1}$ and $\phi_a \in \mathbb{R}^{N_a \times 1}$ are the corresponding regressor parameters. Here, N_c and N_a are the numbers of the neural nodes in critic and actor ANNs, respectively. While the values for weights are estimated online, the regressor parameters should be defined by the designer before running the algorithm. Similar control algorithm to (2.10b) is presented by Yang et al. (2017), where a data-based adaptive dynamic programming is proposed by using two separate ANNs for actor and critic units. In that work, the simulation results are provided for application of the proposed algorithm on a F-16 aircraft dynamics.

Recently, the optimal RL algorithms are also used in some practical applications. Luy (2017b) utilizes an adaptive dynamic programming algorithm for tracking control of wheeled mobile robot. In work by Zhang et al. (2018), ANNs with the sigmoid activation function are combined with the concept of the sliding-mode control and an optimal adaptive MFC solution is suggested for tracking problem in completely unknown nonlinear dynamic systems. In that work, the optimality analysis is provided using the HJB equation, as well.

As it is mentioned in Section 1.2, using a multi-agent system including multiple AMRs rather than using a single AMR is another solution to overcome the limited on-board energy resources on small-sized AMRs. In order to achieve tracking objective within a team of AMRs, design of cooperative control algorithms is indispensable.

2.3 Cooperative model-free control for multi-agent systems

An autonomous multi-agent dynamic system which can be cooperatively controlled is superior than single-agent counterpart. The direct robotic applications of such this system are carrying large cargos, time-efficient aerial photography, time-efficient spraying over the agricultural fields as well as recently presented light aerial shows for the sake of entertainment.

In this regard and in parallel with the investigations of the MFC algorithms for single-agent dynamic systems, there are lots of works performed to solve the cooperative consensus and formation-tracking problems in multi-agent dynamic systems constructed by the agents with unknown linear or nonlinear dynamics (Zhang et al., 2019). One of the initial research on cooperative control in multi-agent systems is the work done by Jadbabaie et al. on coordination of a group of autonomous mobile agents (Jadbabaie et al., 2002, 2003). Jadbabaie and his colleagues also investigated the flocking problem (Tanner et al., 2003a, 2003b). All of these works have been proposed for linear dynamic systems. Works done by Olfati-Saber (2006) on cooperative and flocking control of networks of single-integrator and double-integrator linear dynamic systems are also among of the pioneered work (Olfati-Saber, 2006; Olfati-Saber et al., 2007).

2.3.1 Using artificial neural network as online model-based estimator to generate cooperative control algorithms for multi-agent systems

Later, Lewis et al. have proposed a *distributed* adaptive control protocol in (Das & Lewis, 2010; Lewis et al., 2014) for *synchronization* (i.e. consensus) problem in a network of single-integrator and double-integrator unknown nonlinear dynamic systems. The protocol is a distributed or decentralized algorithm (and not a centralized one), since it defines a specific control law at each agent for achieving the consensus in the network. Suppose a network with N_c double-integrator agents with the following dynamics

$$\dot{x}_i^1 = x_i^2 \quad (2.11a)$$

$$\dot{x}_i^2 = f_i(x_i) + u_i + w_i, \quad (2.11b)$$

and a *leader* with the corresponding states of $[x_0^1; x_0^2]$. In (2.11a) and (2.11b), $f_i(\cdot) \in \mathbb{R}$ is the unknown nonlinear term and $w_i \in \mathbb{R}$ is the unknown bounded disturbance. Here, the dynamic system at agent i has one control input u_i and two system states as x_i^1 and x_i^2 . The agents in the network are communicating with each other according to a *communication graph*. The adjacency matrix element, a_{ij} is defined as a weight for the communication link between the agent i and the agent j ; then $d_i = \sum_{j=1}^{N_c} a_{ij}$. Moreover, β_i is defined as a binary value (i.e. 0 or 1), indicating the existence of a communication link between the leader and the i th agent.

A distributed control law for the dynamic system presented in (2.11a) and (2.11b)

at each agent is proposed as follows (Das & Lewis, 2010; Lewis et al., 2014)

$$u_i = cr_i + \frac{\lambda_c}{d_i + \beta_i} e_i^2 - \hat{W}_i^T \phi_i \quad (2.12a)$$

$$\dot{\hat{W}}_i = -F_c \phi_i r_i p_i (d_i + \beta_i) - k_c F_c \hat{W}_i, \quad (2.12b)$$

where

$$e_i^1 = \sum_{j=1}^N a_{ij} (x_j^1 - x_i^1) + \beta_i (x_0^1 - x_i^1) \quad (2.13a)$$

$$e_i^2 = \sum_{j=1}^N a_{ij} (x_j^2 - x_i^2) + \beta_i (x_0^2 - x_i^2) \quad (2.13b)$$

$$r_i = e_i^2 + \lambda_c e_i^1, \quad (2.13c)$$

and c , λ_c , k_c and F_c are constant positive scalars for tuning the control algorithm (which defined by several off-line try-and-error simulation tests), $p_i > 0$ is a constant scalar gain which is determined according to the communication graph properties, $\phi_i \in \mathbb{R}^{N_{nr} \times 1}$ is a vector including the regressor parameters at agent i and $\hat{W}_i \in \mathbb{R}^{N_{nr} \times 1}$ is the vector of weights used for estimating the unknown nonlinear dynamics at the i th agent. According to term $\hat{W}_i^T \phi_i$ in (2.12a), the unknown nonlinear dynamics of each agent is estimated online using an ANN with N_{nr} neurons. The objective is to reach consensus on the states of agents in the network, i.e.

$$\lim_{t \rightarrow \infty} r_i = 0 \quad for \quad i \in [1, N_c]. \quad (2.14)$$

The parameters e_i^1 and e_i^2 in (2.13a) and (2.13b) are named as the *consensus* errors in the literature. Thus, r_i can be considered as a sliding-mode consensus error. The control algorithm in (2.12a) has been extended for the multi-agent systems with higher-order dynamics at each agent by Zhang and Lewis (2012).

The above cooperative control algorithm that uses a nonlinear control protocol, inspired other researchers for investigating the novel distributed protocols for consensus and formation-tracking problems in multi-agent nonlinear systems with unknown dynamics. Work by Wang et al. (2015) reported the design procedure of a distributed state-output feedback cooperative control algorithm for a network of multiple uncertain agents, with directed communication graphs. Similarly, ANNs are utilized by Peng et al. (2015, 2013) to estimate the unknown nonlinear terms at each agent, whilst solving for the cooperative tracking problem.

There are other works which employed a direct adaptive approach using ANNs to approximate an appropriate control signal to deal with unknown dynamics of agents in a network (Meng et al., 2014); (Cui, Xu, et al., 2016). In the mentioned algorithms, the controller gains matrix is defined off-line by the solution of a continuous-time algebraic Riccati equation (CARE), locally at each agent in the network. Besides the unknown internal and external dynamics in the agents, the dead-zone dynamics for control input has been addressed by Shen and Shi (2016), utilizing the realm of ANNs.

Later, an asymptotical convergence in consensus error is proven by Meng et al. (2017). The algorithm includes ANNs for estimating the unknown internal nonlinear dynamics of the agents, as well as a robust term to cancel the effect of unknown bounded external disturbance. It is also extended to the formation-tracking problem in a network of uncertain nonlinear dynamic agents. In this regard, for a network of completely unknown nonlinear dynamic systems with the dynamics at agent i considered as follows

$$\dot{x}_i = f_i(x_i) + u_i + d_i , \quad (2.15)$$

where $x_i \in \mathbb{R}^n$ is the vector of system states, $u_i \in \mathbb{R}^n$ includes the control variables, $f_i \in \mathbb{R}^n$ is the unknown internal dynamics of the i th agent and $d_i \in \mathbb{R}^n$ is the unknown bounded external disturbance; the distributed control signal is defined as follows (Meng et al., 2017)

$$u_i = -k^c e_i - \hat{W}_i^T \phi(x_i) - \frac{\hat{B}_i^2}{\|e_i\| \hat{B}_i + e^{-t}} \quad (2.16a)$$

$$\dot{\hat{W}}_i = \Gamma^c \phi(x_i) e_i \quad (2.16b)$$

$$\dot{B}_i = \theta^c \|e_i\| . \quad (2.16c)$$

In this algorithm, the consensus error is defined as follows

$$e_i = \sum_{j=1}^{N_c} a_{ij}(x_i - x_j) , \quad (2.17)$$

and $\Gamma^c = (\Gamma^c)^T$ is a positive definite adaptation gain matrix, $\theta^c > 0$ is a positive learning rate and $k^c > 0$ is the scalar controller gain. As it is observed, a robust term is added in (2.16a) (the third term), with the corresponding adaptive law for its gain, i.e \hat{B}_i .

In both of the algorithms presented in (2.12a) to (2.12b) and (2.16a) to (2.16c), regressor parameters should be defined prior to the operation of the system. The corresponding PE condition that follows should be fulfilled as the associated regressors residing within ANNs model require the persistently exciting basis signal. Similarly, an ANN is used for online estimation of unknown nonlinear dynamics in (Wang, Wang, & Peng, 2017). An observer is also proposed to predict the unmeasured states of the nonlinear dynamic system, locally at each agent.

Work by Wang, Wen, and Huang (2017) introduced estimations for the nonlinear functions in the system dynamics using a set of ANNs by which a consensus tracking problem is solved for a network of nonlinear agents with globally uniformly bounded control signal at each of the agents. Moreover, a sliding-mode observer is used for parameter estimation. In that algorithm, it is considered that the information of the desired trajectory to be tracked, is not available at all agents in the network. Instead, the knowledge on the upper bound for the changing rates of the desired trajectory is used in order to track the trajectory. In the work by Wang (2017), the ANNs are used locally to reconstruct the uncertain dynamics of each agent as well as the unknown dynamics of the leader agent. The information of the leader states is assumed to be available to only a portion of the agents, while the leader dynamic system is completely unknown for all of the agents in the network. In that work, two distinct model-based adaptive laws, which are incorporating some regressor variables based on the consensus errors, have been proposed for online estimation of the unknown leader dynamics and unknown nonlinear agents dynamics.

Later, in the works proposed by Shi and Shen (2017) and Wen et al. (2017), the Linear Matrix Inequality (LMI) is employed to find the values of the controller gain matrix from the Schur complement in an off-line manner, while ANNs are used to estimate the unknown dynamics in the network of SISO and MIMO nonlinear dynamic agents. In the presented algorithm by Li et al. (2019), ANNs are used for the online approximation of the unknown nonlinearities at the agents and the consensus problem in a heterogeneous network of first and second order nonlinear dynamic systems is solved. The solution guarantees consensus error convergence to be UUB in a cooperative network setup.

While most of the proposed ANN-based cooperative control policies for completely unknown nonlinear dynamic systems provide UUB convergence of the consensus errors, the solution presented by Cui et al. (2018) provides an asymptotic convergence on the consensus errors for a high-order completely unknown multi-agent dynamic system. This special characteristic is delivered by adding an extra requirement on the boundedness of the second derivative of the unknown internal dynamics and external disturbance. A solution free of the back-stepping method is suggested by Meng et al. (2019) to address the consensus problem in a network of high-order unknown nonlinear dynamic systems. In that solution, two different sets of ANNs are used for estimating a portion of the system states as well as the unknown internal dynamics at each agent, based on the consensus error on the measured state (or the output) of the system.

2.3.2 Reinforcement learning for cooperative control algorithms

Besides the work reviewed in the previous section, techniques from reinforcement learning concept are used to solve the optimal tracking control problem in multi-agent dynamic systems. A distributed extension of the adaptive dynamic programming algorithm is utilized by Luy (2017a) to design a cooperative control policy in a network of partially unknown nonlinear dynamic systems. In (Cui et al., 2016), two separate ANNs with associated laws for online adaptation of the weights locally at each agent in network, are used for estimating the unknown nonlinear dynamics and the reinforcement control signal, respectively. In the work presented by Zuo et al. (2018), the off-policy RL algorithm is utilized for providing a solution for the consensus problem in a network of completely unknown linear dynamic systems.

The policy iteration algorithm incorporated with online estimations by ANNs is proposed by Zhang et al. (2018) to deliver an optimal solution for the consensus problem in a network of unknown dynamic systems. In the work by Zhang et al. (2018), the value function is approximated with a set of ANNs and the weights of neural nodes are updated online utilizing a least squares (LS) technique over the sampled input-output data gathered in a definite past time window of the network operation. After that, a normal policy iteration algorithm is used for updating the optimal control policy. The optimality analysis of that algorithm is presented using the *coupled HJB* equation (Zhang et al., 2018).

Recently, a cooperative MFC RL algorithm is proposed by Modares et al. (2018) for optimal synchronization problem in a heterogeneous multi-agent system with unknown nonlinear dynamics. In that work, a decentralized cooperative observer is designed for estimating the leader states at each agent in the network. Then, two ANNs are utilized for solving the HJB equation and consequently estimating the value function and the optimal controller at each agent. The regressor parameters used in the ANNs, include the past values of the local optimal controller, the past values of the local value function and the local consensus error for observing the leader states. The weights of ANNs at each agent are estimated online using the simple LS algorithm. Since no learning rate is considered in the LS estimation, its robustness cannot be guaranteed (Kofahl, 1988). Using recursive LS presented in (Nelles, 2001), can be a good solution to provide robustness for the proposed algorithm by Modares et al. (2018).

2.3.3 Fuzzy inference system for online estimation in cooperative control algorithms

There is a wide range of nonlinear basis functions that can be utilized for online estimation of the unknown nonlinear terms in a dynamic system, the corresponding optimal controller and the value function. While in most of the ANNs, sigmoid or hyperbolic tangent are used as the basis functions, the fuzzy membership functions are incorporated in a FIS for online nonlinear estimation. A solution utilizing a FIS for online local estimation of the unmodeled nonlinear dynamics, is investigated by Shen et al. (2016) for the synchronization problem in a network of nonlinear agents. Similarly, a FIS network is used by Wang et al. (2016) for online estimation of the unknown nonlinear terms and reconstructing the unmeasured states at each agent in a multi-agent system. In that work, a distributed adaptive fuzzy output feedback controller is designed to address the consensus problem.

In (Sakhre et al., 2017), a fuzzy competitive learning algorithm is used for online learning of ANNs, leading to design of an algorithm for online estimation of the nonlinearities in the dynamic systems within a network. In (Ren et al., 2017), a heterogenous network of uncertain nonlinear dynamic systems has been provided with a solution for the leader-following consensus problem, by incorporating the FIS for approximating the unknown dynamics. A solution using radial basis function (RBF)-ANN is proposed by Yang et al. (2017) for distributed formation control problem in a team of non-holonomic autonomous vehicles. Similarly, the RBF-ANN is used in (Chen et al., 2016) and (Wang et al., 2016).

Distributed fault-tolerant cooperative control within a team of nonlinear dynamic

agents with actuator failures is another application for the incorporation of the RBF-ANNs to approximate the unknown nonlinear terms (Yang & Yue, 2017). In the work by Cui et al. (2018), the fuzzy approximation and the prescribed performance techniques are combined to deliver a solution for consensus problem in a network of completely unknown dynamic systems. The Fourier series are another nonlinear basis functions which are considered for online estimation of the nonlinearities at the agents dynamics in a multi-agent system (Wang et al., 2016, 2017).

2.3.4 Cooperative control algorithms without model-based estimation tools for multi-agent dynamic systems

Similar to the case of single agent dynamic systems, the computational complexity increases with the number of nodes being employed in ANNs for the estimation purpose. This issue can have adverse effects on the inter-agent data communication which is a vital requirement for practical implementation of the cooperative estimation and control frameworks. To solve the issue, a solution is proposed by Cui, Zhuang, and Lu (2016) for the problem of distributed synchronization, using the dynamic-surface control (DSC) technique. Despite decreasing the complexity, the proposed algorithm relied on the use of ANNs.

The leader-follower consensus control problem is solved in (Ding & Li, 2016) for a network of nonlinear dynamic agents with unknown polynomial functions as the internal nonlinearities. In the design procedure of that work, an algebraic inequality is used to define a definite upper bound on the unknown polynomials, and a distributed control policy is suggested with no information needed about the unknown

nonlinearities. While the solution by Ding and Li (2016) does not use ANNs or FISs for online estimation of the nonlinear terms, it is designed only for especial case of polynomial functions as the unknown internal dynamics and it is not viable for generic unknown nonlinearities in the agents dynamic system.

Later, a decentralized cooperative protocol, in which no online estimation of the agent nonlinear dynamics is investigated is proposed by Bechlioulis and Rovithakis (2017) for the synchronization problem. Instead of the online estimation, an exponential dynamics is proposed for upper bound of the absolute consensus errors in the network. The characteristics of the proposed dynamic bound are suggested according to the desired performance of the system in transient and steady state conditions. Hence, neither the prior knowledge of the nonlinearities in the agents' dynamics nor the popular online estimation tools (i.e. ANN, FIS and etc.) are incorporated in the proposed solution. This provides less computation, which is suitable for practical implementations especially when onboard power supply is limited on each agent in the network (for example as in the case of small wheeled mobile robots (WMRs) and quadrotors). However, the main controller gains need to be determined off-line via expert knowledge.

The idea of incorporating the prescribed performance as dynamic bound for the consensus error is used by Hashim et al. (2017), as well. Adaptive projection algorithm is considered rather than the ANNs for online approximation of the unknown nonlinear terms in the agents' dynamics in the network. For practical implementation of the proposed adaptive projection algorithm, the upper bounds for the agents' states are required. In addition, the proposed adaptive law for online estimation is driven by a

transformed consensus error, which is a nonlinear function of the ratio of consensus error to the system's prescribed performance. An exponential function is employed as the required nonlinear function. As it can be seen, the use of nonlinear basis functions is still evident for online estimation of the nonlinearities. Moreover, it is supposed that there is *a priori* known system matrix in the dynamics of each agent in the network, which makes the solution to be applicable for partially-known dynamic systems (and not the completely-unknown ones).

In the work by Bu et al. (2018), a model-free iterative learning control algorithm is presented to solve the formation-tracking problem in a network of completely unknown non-affine nonlinear dynamic systems. In that algorithm, it is assumed that the whole unknown non-affine nonlinear dynamics at each agent is non-zero and can be modeled using the compact form dynamic linearizing approach. In addition, it is shown that the provided estimations by the pseudo-partial derivative technique for unknown parameters at each agent are bounded. As it can be observed, the PE condition is required in that algorithm, as the unknown parameters are assumed to be non-zero. Moreover, the formation-tracking problem is solved by assuming that all of the agents have access to the deviation or formation parameters desired at all of the other agents. This is not a reasonable assumption, especially when the desired formation parameters are time-varying. Same solution as in (Bu et al., 2018), is presented for the consensus problem in the designed algorithm by Bu et al. (2019).

Recently, a robust adaptive dynamic programming algorithm is presented in (Gao et al., 2019) for the cooperative output regulation problem in a network of unknown nonlinear dynamic systems. That algorithm incorporates the techniques from linear

optimal control theory and also an iterative solution for a CARE to design the distributed control signals. The algorithm is able to approximate the controller gains without relying on the systems dynamics. While there is not any ANN used for online parameter estimations, the controller gains are approximated at each step of the proposed algorithm in an online manner through two consecutive iterating loops. Thus, it requires huge computations at each operation step and at each of the agents in the network, which normally leads to inefficient energy consumption in the network. In addition, the input-output data gathered in a past time window is needed for running the mentioned iterating loops to approximate the controller gains.

The applications of novel cooperative control algorithms for formation-tracking problem in the team of AMRs with some uncertainties in internal and external dynamics are presented in (Abbasi et al., 2017; Ghommam & Saad, 2018; Khan et al., 2016; Lee, 2018; Lu et al., 2018; Mahmood & Kim, 2017; Mao et al., 2016; Zhao et al., 2017).

Besides the presence of unknown terms in the nonlinear dynamics at each agent in a multi-agent system, some system states might be unknown; or available without acceptable accuracy. In most of the control protocols, the accurate values for system states must be available online in order to achieve an appropriate tracking or formation-tracking performance. As mentioned before in Section 1.2, the information on the local or global position of the AMRs needs to be available so as to achieve a high level of autonomy. Hence, the localization problem should be solved for a single AMR or a multi-agent system including several AMRs. Several solutions to the problem of low positioning accuracy of the GPS data have been investigated in the literature to provide

more accurate and applicable solutions.

2.4 Cooperative localization for real-time positioning

In a decentralized cooperative control algorithm for a team of AMRs, the need for inter-agent relative position (not relative distance) information in 2D or 3D environments is vital, depending on different applications (refer to *Definition 2-1* for more details). Such requirement inherently exists in almost all of the distributed cooperative control protocols, as all of these algorithms need the relative measurements among the states of neighboring agents in the network to construct the values of consensus errors. In addition, the inter-agent relative position information can be used in conjunction with a set of absolute position data in a predefined local or global coordinates system, in order to improve the absolute localization of each member in the team (Wanasinghe et al., 2015).

Definition 2-1. The relative position between two agents in a 3D environment, is defined as

$$\vec{P}_{rel} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} - \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}. \quad (2.18)$$

Moreover, the relative distance between these two agents is defined as follows

$$d_{rel} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}. \quad (2.19)$$

Note that the above definitions for a 2D environment can be proposed in a similar way by only incorporating the x and y data.

Since the majority of the AMRs (especially the drones) are comparatively small moving agents, they have limited capability to carry heavy sensors such as Radar or Lidar (Kang et al., 2017). In addition, the energy consumption of the on-board power supply (i.e. batteries), which is an important factor for design and application of the AMRs, will be increased effectively by adding extra weights imposed by the Radar or Lidar modules. Therefore, other relative positioning solutions utilizing ZigBee, WiFi and UWB modules have been widely investigated, recently (Guo et al., 2017). In spite of these advancements, the localization problem is still an active and open issue, especially in obtaining the location of the AMRs accurately with minimal on-board resources (Kang et al., 2017). These solutions or methods can be categorized in two major categories presented in the following.

2.4.1 Localization methods with GPS

The first category is dedicated to methods which are intending to improve the accuracy of localization with GPS modules. These methods are

- *Method-1:* Improving the GPS data accuracy by receiving the correction signals from some nodes with exact known position. The Differential GPS (DGPS) and Real-Time Kinematics (RTK) methods are considered in this group. Although these methods can provide appropriate accuracy, the high expenses to support the infrastructures prevent the method to be a good solution in ad-hoc networks of AMRs with specific application and geographical area (Hedgecock, 2014);
- *Method-2:* Providing GPS modules with the actual information of orbiting satellites in the earth atmosphere. This technique, which is named as

Assisted-GPS (A-GPS), is now available in smart phones without essential improvement in the positioning accuracy (Zandbergen, 2009).

2.4.2 Localization methods without GPS

The second category consists the methods which do not incorporate the GPS modules on the agents in the localization process. These methods are

- *Method-3:* By generating a map of the environment that the mobile agent is moving around and then localizing the agent in that map according to the distances measured to some fixed landmarks in the map, an online solution can be provided for mobile-agent localization problem in unexplored environments.

These methods are identified as simultaneous localization and mapping (SLAM) algorithms. The main drawback of SLAM algorithms is the high computational complexity imposed by implementing the algorithms, which makes them less efficient specially in large network of multi-agent mobile systems. Moreover, due to the rich information contained in the generated map, there will be occurrence of data transmission lag issue between the neighboring agents (Safavi & Khan, 2017; Safavi et al., 2018).

- *Method-4:* By integrating the translational and angular velocities (i.e. *odometry data*) of a moving agent, the information about the position and orientation of that agent can be obtained with reference to the initial states (so-called *dead-reckoning*). Inertial sensors such as Inertial Measurement Units or IMU modules in short, are utilized popularly for measuring the kinematics variables of a mobile agent (Kok et al., 2017). However, errors in the

measurement will accumulate in numerical integrating procedure, leading to the estimation drift in a definite time (Kia et al., 2016; Kok et al., 2017; Russel et al., 2017; Tron et al., 2016). There are several filtering methods such as extended Kalman filter (EKF), which are proposed to deal with the error drift (Lewis et al., 2007; Wheeler et al., 2018). These filtering methods mostly depend on the dynamics of the mobile agent in order to have suitable performance. Moreover, they are applicable for systems with the Gaussian measurement noise. Therefore, complementary filters are proposed to provide drift-free estimations without requirements on the agent dynamic system (Madgwick et al., 2011; Mahony et al., 2008). Most of the complementary filters are designed for estimation of the angular position (i.e. Euler angles).

- *Method-5*: Defining the target location within a network using the received data from several fixed or mobile nodes with *a-priori* position data, which are named as *beacon* agents. Two well-known methods, i.e. WiFi positioning and Cellular positioning belong to this group (Zandbergen, 2009). Moreover, the localization protocols investigated in the literature of wireless sensor networks (WSNs) are considered in this category (Aspnes et al., 2006; Boukerche et al., 2007; Mao et al., 2007; Stanoev et al., 2016). According to (Bouzera et al., 2017; Zandbergen, 2009), the accuracy of these localization protocols is not better than the one for GPS modules.

2.4.3 Characteristics of cooperative localization

While continuous progress is evident in all of the above methods, cooperative localization algorithms have been receiving special attention, recently. This is due to

the fast-developed emerging technologies in connectivity among the agents and objects (i.e. Internet of Things (IOT)), as well as the trending in embedded control systems, aligned with the Industrial Revolution 4.0 (IR4.0).

The work by Kia et al. (2016) and Liu et al. (2017) reveals that using only the measured data on a single mobile agent cannot improve the accuracy of localization appropriately, especially when the agent is included in a multi-agent system. Incorporating data available in a network including several agents can be useful for improving the localization accuracy. In other words, if each agent in the network relies not only on its own motion measurements (e.g IMU measurements), but also on the sensory data measured about the neighboring agents in the network (e.g. UWB measurements), further enhancement in localization accuracy at each agent can be achieved (Liu et al., 2017). Thus, the concept of *cooperative localization* is investigated (Kia et al., 2016; Wanasinghe et al., 2015).

In addition, the cooperative localization solutions are attractive to be involved in different types of applications such as a team of AMRs. Cooperative localization incorporating multiple mobile agents would deliver some potential advantages over single-agent localization solutions (i.e. GPS modules, dead-reckoning, Kalman filter and complementary filter). These advantages are enhanced localization accuracy, extended coverage area, algorithm robustness and flexibility especially in case of the possible measurement failures due to harsh environmental conditions (Chai et al., 2016; Wanasinghe et al., 2015).

For a network of mobile agents, there are several localization solutions proposed

in the literature, which are reviewed with details by Chelouah et al. (2018). According to that survey, cooperative localization algorithms for the network of mobile agents are divided into five categories as grid-based algorithms, probability distribution solutions, timing-based localization, constrained localization algorithms and the solutions for underwater sensor networks (Chelouah et al., 2018). Among them, the algorithms based on the probability distribution are used in many localization cases. Particularly, the *Bay*'s rule is utilized to provide the likelihood distribution for agents position in the network (Safavi et al., 2018). These algorithms include EKF and particle filters at each agent in the network.

The EKF filters can be used in a network with known dynamic and measurement models for each agent in the network (Kia et al., 2016; Ouimet et al., 2018). By applying the EKF filters, likelihood distribution of the agents' positions have been approximated by a Gaussian estimation method. This leads to large localization error if the exact distribution for the agents position is not a Gaussian one (Safavi et al., 2018).

On the other hand, particle filters or sequential Monte Carlo methods can be implemented for non-Gaussian systems, leading to many applications in practical localization problems. The particle filters rely less on the agents dynamics; instead they are working on the probability distribution generated based on the previous location of the mobile agents and the inter-agent relative observations. Several Monte Carlo localization algorithms can be found in (Fox et al., 2000; Özkucur et al., 2009) and (Huang & Záruba, 2009).

Despite lots of the benefits provided by particle filters such as flexibility, simple implementation and no requirement for huge memory to store the past data; these solutions are very time consuming due to several samples needed to produce the likelihood distribution of the agents' position (Safavi et al., 2018). Novel sampling and clustering techniques are proposed by Prorok and Martinoli (2011) and Prorok et al. (2012) so as to reduce the set of the particles as well as the computational cost of the Monte Carlo localization solutions. However, the problem of cross-correlation between the local position estimations at each of the mobile agents is still existed in the above solutions, leading to major unknown errors in the localization task (Wanasinghe et al., 2015).

Besides the categories mentioned in above, all of the solutions proposed in the category of cooperative localization algorithms for a network of mobile agents, can be distinguished according to the following characteristics;

- ranged-based or range-free; regarding the availability of inter-agent measurements in the network (including distance and bearing (i.e. angle) data between the neighboring agents (Stanoev et al., 2016);
- anchor-based or anchor-free; regarding the availability of beacon agents in the network (Priyantha et al., 2003), where the anchor-free algorithms are used for relative localization only;
- static or mobile anchor; regarding the mobility of beacon agents in the network (Chenji & Stoleru, 2013; Halder & Ghosal, 2016);
- distributed or centralized; depending on whether or not a central processing unit

is used for implementing the algorithm and sending the corresponding localization results to the agents in the network (Lin et al., 2015; Zhao & Zelazo, 2016).

2.4.4 Cooperative localization algorithms

The early investigations on the cooperative localization can be credited to the work by Kurazume et al. (1994), where the *leap-frogging* motion pattern is suggested for a team of mobile agents. The main disadvantage of that algorithm is that, only one mobile agent or at-most a portion of them is allowed to navigate in the environment at each time step, leading to longer completion time of the navigation-localization mission (Wanasinghe et al., 2015). The leap-frogging technique is further improved by Rekleitis et al. (2003a, 2003b) and Trawny and Barfoot (2004). All of those solutions are in category of centralized cooperative localization algorithms.

Due to the high computational costs of the centralized algorithms, decentralized (or distributed) cooperative localization algorithms are developed so as to reduce the computational cost (Wanasinghe et al., 2015). One of the first distributed solution for the cooperative localization problem in a team of mobile robots is presented in (Roumeliotis & Bekey, 2000, 2002). The algorithm is further improved in (Nerurkar et al., 2009) and huge enhancement of the computational cost has been achieved over the previously developed centralized solutions. These algorithms are based on the Kalman-filter estimation which is extended for decentralized implementation within a team of mobile agents. In this sense, they depend on the dynamic system of each mobile agent in the network.

Oh and Ahn (2013, 2014) proposed the following cooperative position estimation law

$$\dot{\hat{p}}_i = u_i - k_0 \sum_{j=1}^N [(\hat{p}_i - \hat{p}_j) - (p_i - p_j)], \quad (2.20)$$

for a network of N single-integrator mobile agents with dynamic system of $\dot{p}_i = u_i$, where, $p_i \in \mathbb{R}$ is position of the i th agent and $\hat{p}_i \in \mathbb{R}$ is its corresponding estimation, $u_i \in \mathbb{R}$ is the control input at i th agent and $k_0 \in \mathbb{R}$ is a positive constant gain for tuning the estimation law. As it is observed, the position estimation law is coupled with the control signal. Moreover, it is claimed that the estimated values converge to the actual positions of the agents with only using one beacon agent, if the communication graph for the network is uniformly connected. Furthermore, the authors have developed a formation control law based on the estimated positions, which allows the agents to actively control their positions and provide a desired formation with other agents, cooperatively.

Kia et al. (2016) presented two distributed EKF algorithms for online position estimation in a team of WMRs. It is assumed that each agent has access to the position measurement corrupted by the bounded measurement white noise. Moreover, the internal dynamics of each WMR should be known completely to let the algorithm be implemented. In the same way, the authors in (Lu et al., 2016) proposed a cooperative localization protocol using EKF among a team of three WMRs. Similar solution is suggested by Ouimet et al. (2018). In the work presented by Morbidi and Mariottini (2013), a team of aerial vehicles is used to track a moving target agent. Each drone is equipped with a 3D range-finding sensor and a cooperative EKF algorithm to localize the moving target. In addition, the optimal path that drones may

track in a 3D space is defined so as to maximize the accuracy of the drones' position estimation and that of the mobile target. As all of these solutions contain cooperative EKF protocols, the localization algorithm is coupled with the dynamics of the agents as well as the control inputs. Hence, the performance of the controller affects the cooperative localization algorithm.

The solution presented by Tron et al. (2016), deals with the cooperative localization and the cooperative formation control problems by utilizing the vision-based measurements. In general, vision-based sensors provide projected measurements in a 2D plane, which would not contain distance information along the normal direction. Consequently, these measurements can provide only bearing information between two desired agents (Tron et al., 2016). Using additional depth sensor, the distance information can be provided into the vision-based measurements.

The localization algorithm designed by Wang et al. (2016), aims to locate three quadrotors in an outdoor environment using a WMR operating as the mobile beacon agent with known local position (no global position information is available). A motion capturing system with eight cameras is used for evaluating the performance of the proposed algorithm. In (Chai et al., 2016), a cooperative localization protocol is developed at each mobile agent within a network to estimate the relative coordinates of each agent with respect to a stationary landmark, which is acting as a beacon agent. In that work, on-board modules are used at each agent for measuring the absolute velocity. Moreover, distances to the nearby agents and the rates of change in these distances are computed using exteroceptive sensors.

Similarly, the probabilistic localization solutions presented by Wanasinghe et al. (2015) requires the availability of the exteroceptive sensors to directly measure the relative pose of the neighbors. Moreover, the range and bearing information of the neighboring agents are assumed to be measured locally at all of the agents in the network. Such limitation is regarded as the drawback for flying AMRs, especially when minimal total mass for each agent is desirable for the purpose of maximizing the robot operating time. Han et al. (2019) also reported that it is practically difficult to recover a consistent measurements for range distance and bearing angle between the AMRs.

In (Meyer et al., 2016), the distributed localization problem alongside with the mobile target tracking problem is solved for two distinct cooperatively-controlled and non-cooperatively-controlled multi-agent systems with nonlinear dynamics. The problem resolved by Russel et al. (2017) is how to localize a quadrotor which does not have GPS module, with the assistance of a nearby GPS-enabled quadrotor. The angle of arrival for the signals communicated between the two agents, should be measured as a requirement for implementing the proposed algorithm.

A solution for indoor localization using only one beacon agent is proposed by Grobwindhager et al. (2018). The solution which is named as *SALMA* (Single-Anchor Localization system using Multipath Assistance), exploits the multipath signals received from a tag agent at the beacon agent in an indoor environment. By computing the angle of departure for each signal, a maximum likelihood algorithm is used for determining the position of the tag agent in a 2D environment. According to the presented experimental results by Grobwindhager et al. (2018), decimeter-level

accuracy has been achieved for multiple fixed tag agents using only one beacon agent. The SALMA requires a pre-defined map of the environment representing all of the walls and existing landmarks. Moreover, an array of antennas is needed at the beacon agent for more accurate measurement of the angle of departure for all of the signals. In addition, since the algorithm works based on the existence of multipath signals between the tag and beacon agents, the SALMA cannot be used in outdoor environments without walls.

As mentioned before, although the angle of departure can be measured within a network of fixed agents easily, the synchronization of angle of arrival measurement and the inter-agent distance measurement for the moving agents such as drones is hardly achieved in practical applications (Han et al., 2019). To remedy this problem, an integrated solution for relative localization and leader-following formation control is presented by Han et al. (2019) for a team of drones, in which the direct relative position between the two neighboring agents is updated as follows

$$\dot{\hat{z}}_{ij} = v_{ij} + \alpha^c v_{ij} (2d_{ij}\dot{d}_{ij} - v_{ij}^* \hat{z}_{ij} - v_{ij} \hat{z}_{ij}^*) , \quad (2.21)$$

where $z_{ij} \in \mathbb{R}^{n_d \times 1}$ is the relative position (refer to *Definition 2-1*) between agents i and j , $v_{ij} \in \mathbb{R}^{n_d \times 1}$ is the relative velocity computed at agent i by measuring the global velocity at this agent and receiving the global velocity at agent j via the available communication graph, $d_{ij} \in \mathbb{R}^+$ and $\dot{d}_{ij} \in \mathbb{R}$ are the measured relative distance and its corresponding rate, respectively and $\hat{z}_{ij} \in \mathbb{R}^{n_d \times 1}$ is the estimated relative position information. Here, $n_d \in \{2, 3\}$, $\alpha^c > 0$ is a constant tuning scalar gain and the superscript $*$ stands for the conjugate of the corresponding vector.

Later in (Han et al., 2019), the estimation provided in (2.21) is augmented by an indirect relative localization incorporating the third agent in the loop. Finally, the estimated relative positions are incorporated into a cooperative formation control protocol. It is assumed that each quadrotor has on-board sensors to measure the local velocity (by integrating the IMU measurements) and the inter-agent relative distance (using UWB module). In that proposed integrated solution, the communication graph between the agents needs to have at least two paths from one agent in the network to reach another agent, which is restrictive and often hard to be practically achieved.

Recently, a distributed solution is proposed for cooperative localization within a network of mobile dynamic agents (for example AMRs), including at least one beacon agent and at least three listening agents (Safavi & Khan, 2017; Safavi et al., 2018). The main novelty of that solution is to solve the localization in a network of mobile agents with only one beacon agent, without coupling to the agents' dynamics or corresponding control signals. The solution uses the well-known triangulation method among at least three neighboring agents in the network. Hence, there should be three or more communication links connected to each agent. This imposes a restrictive requirement for implementing the algorithm in practical cases, where minimal possible communication links is preferable. The formulation of that solution is as follows

$$\hat{x}_{k+1}^i = \alpha_k \hat{x}_k^i + (1 - \alpha_k) \left[\sum_{j=1}^N a_k^{ij} \hat{x}_k^j \right] + v_{k+1}^i, \quad (2.22)$$

where $\hat{x}_k^i \in \mathbb{R}^{n \times 1}$ is the estimation for position of the i th mobile agent at the k th step in time domain, $0 < \alpha_k < 1$ is a design scalar parameter, $a_k^{ij} \in \mathbb{R}^+$ is the barycentric coordinates of the i th agent with respect to the neighboring agent j , $v_{k+1}^i \in \mathbb{R}^{n \times 1}$ is the

absolute speed measured at agent i . The value of α_k is one, when the least required numbers of neighboring agents (e.g. three or above) are not available in vicinity of the agent i at time step k . In other words, when the required condition to incorporate the triangulation method is not satisfied, the update on position estimation is derived only by the measured absolute velocity of the agent. Otherwise, the value of α_k is designed so as to have a trade-off between the estimation updates between the previous values and the current value computed by the triangulation (Safavi & Khan, 2017). As can be seen in (2.22), the estimation algorithm is decoupled from the agents' dynamics; only the first-order kinematics for mobile agents are used. Moreover, the proposed algorithm is applicable for discrete-time dynamic systems.

2.5 Research gaps

Here, the research gaps are presented corresponding to the main problems defined and reviewed in above.

2.5.1 Gap in the concept of Model-free control

According to the presented literature in Section 2.2, there are two school of thoughts in the concept of MFC. The first branch which is pioneered by (Fliess & Join, 2013) and then followed by (Hou & Jin, 2014) and (Roman et al., 2017), used simple time-integration of the fixed time window on past input-output data of the system for online parameter estimation. This requires ample amount of memory to store data, especially in complex systems with highly nonlinear dynamics. There is not any ANN or FIS used for online estimation of the unknown terms. On the other hand, in all of these solutions, no methods are proposed for online updating of the

controller gains and several try-and-errors are needed in an off-line manner to tune the controller gains. Moreover, these solutions are not the optimal ones. The characteristics of the major works in this branch are reviewed in the first three rows of Table 2.1.

Note that the work by (Hou & Jin, 2014) is superior compared to the solution presented by (Fliess & Join, 2013), in the sense that it can be applied on completely unknown nonlinear dynamic systems. But, the algorithm by (Fliess & Join, 2013) is more interested in the literature due to its simplicity, leading to several practical applications of it in the literature. Anyway, lack of the optimality and having a method for online updating of the main controller gains are their main disadvantages of the both solutions, which are addressed in the current thesis.

In the second branch, which is started by Vamvoudakis et al. (2014) and then followed by Zhu et al. (2016) and then Zhang et al. (2018), the concept of optimal control theory is brought into the MFC algorithms using the RL methods. In these solutions, the ANNs are utilized to estimate the value function and the control policy, performing the policy iteration algorithm in an online manner. Also, in some of the works in this branch, the ANNs are utilized for online estimation of the unknown nonlinear terms. This requires the control designer to choose the regressor variables as the activation functions for online estimating of the weights in ANNs, just before the algorithm operation.

In addition, one may increase the number of neurons in ANNs in order to achieve more accurate estimation of either the optimal controller or the unknown system

dynamics, which in turn leads to a solution with more computation complexity (Cui, Zhuang, & Lu, 2016). This is an important issue, especially for the small-sized AMRs where the on-board source of energy is mostly scarce and limited. The requirement for exhaustive estimation algorithms to reconstruct the unknown nonlinear terms would increase the control efforts and then would cast extra energy consumption in actuators. Hence, more energy-efficient control and estimation algorithms are vital concerning the practical aspects of AMRs. The characteristics of the major solutions in the second branch are also depicted in the last three rows of Table 2.1.

As can be seen, there is a research gap for bridging between the above two schools of thought. An MFC algorithm is demanded which would be optimal (or at least includes some optimal terms), and on the other hands shouldn't rely on ANNs or FISs. Moreover, having some controller gains be updated online is preferable, as it is not proposed in the literature previously.

Table 2.1: Comparison for the characteristics of the major MFC algorithms in the literature for single-agent unknown nonlinear dynamic systems

Reference to the algorithm	Utilizing integration over past data	Utilizing ANN/FIS	Req. * of PE condition	Optimality	Robustness	Complete unknown dynamics	Online tuning of the main gains
(Fliess & Join, 2013)	✓	✗	✗	✗	✓	✗	✗
(Hou & Jin, 2014)	✓	✗	✗	✗	✓	✓	✗
(Roman et al., 2017)	✓	✓	✗	✗	✓	✗	✗
(Vamvoudakis et al., 2014)	✓	✓	✓	✓	✓	✗	✗
(Zhu et al., 2016) and (Yang et al., 2017)	✓	✓	✓	✓	✓	✓	✗
(Zhang et al., 2018)	✓	✓	✓	✓	✓	✓	✗

* Req. : Requirement

2.5.2 Gap in the concept of cooperative model-free control

According to the literature presented in Section 2.3 and also by continuing from the discussion in Subsection 2.5.1, the research gap in single-agent systems for proposing a MFC algorithm which would be optimal and will not rely on ANNs, can be similarly defined for multi-agent systems in order to achieve consensus and formation-tracking objectives. In this regard, the characteristic for major solutions of cooperative MFC algorithms are reviewed in Table 2.2. Most of these algorithms are utilizing the ANNs or FISs at each agent in a network to deal with the unknown nonlinearities in the dynamic systems, and consequently define decentralized MFC algorithms. Just recently, the prescribed performance method is used by (Hashim et al., 2017) (as in the 7th row of Table 2.2) and the requirement of the ANNs is revoked. That solution is not optimal and there is not any method proposed for online updating of the controller gains.

The work by (Bu et al., 2018) also tried to remove the dependencies of cooperative MFC algorithms to the ANNs, using the iterative learning method. As can be seen in the last row of Table 2.2, that solution is not optimal, the PE condition needs to be satisfied and the controller gains cannot be updated online. Hence, there is a research gap to develop a cooperative MFC algorithm for dealing with the completely unknown nonlinear dynamics in multi-agent systems without relying on ANNs, and then generating the decentralized control signal which include optimal terms. The online updating of the controller gains is another task that can be considered as a contribution.

Table 2.2: Comparison for the characteristics of the major cooperative MFC algorithms in the literature for multi-agent unknown nonlinear dynamic systems

Reference to the algorithm	Utilizing integration over past data	Utilizing ANN/FIS	Utilizing prescribed perf. *	Req. * of PE condition	Optimality	Robustness	Complete unknown dynamics	Online tuning of the main gains
(Lewis et al., 2014)	✗	✓	✗	✓	✓	✓	✓	✗
(Meng et al., 2017)	✗	✓	✗	✓	✗	✓	✓	✗
(Ren et al., 2017)	✗	✓	✗	✓	✗	✓	✓	✗
(Wang, 2017)	✗	✓	✗	✓	✗	✗	✓	✗
(Modares et al., 2018)	✓	✓	✗	✓	✓	✗	✓	✓
(Sakhre et al., 2017)	✗	✓	✗	✓	✗	✓	✓	✓
(Hashim et al., 2017)	✗	✗	✓	✗	✗	✓	✗	✗
(Bu et al., 2018)	✓	✗	✗	✓	✗	✓	✓	✗

* Req.: Requirement; Perf.: Performance

2.5.3 Gap in the concept of cooperative localization

According to the literature reviewed in Section 2.4 and Table 2.3, there is a research gap for developing the cooperative localization algorithms which don't require the relative bearing angle measurements, and just rely on the range (i.e. distance) and relative velocity vector measurements. By looking at the 5th and 6th rows of Table 2.3, it is observed that there is not any such solution in the literature, which can be applied to a network with only one beacon agent and only one mobile agent. This is the research gap in the concept of cooperative localization which would be addressed in this thesis.

Table 2.3: Comparison for the characteristics of the major cooperative localization algorithms in the literature for a team of mobile agents

Reference to the algorithm	Absolute localization	Range meas.	Bearing meas. *	Relative velocity meas. *	Availability for one beacon agent	Availability for one mobile agent	Dependence on the controller
(Oh & Ahn, 2013)	✓	✓	✓	✗	✓	✓	✓
(Kia et al., 2016) and (Ouimet et al., 2018)	✓	✓	✓	✗	✓	✓	✓
(Wanasinghe et al., 2015)	✓	✓	✓	✗	✓	✓	✗
(Grobwindhager et al., 2018)	✓	✓	✓	✗	✓	✓	✗
(Han et al., 2019)	✗	✓	✗	✓	✗	✗	✗
(Safavi & Khan, 2017) and (Safavi et al., 2018)	✓	✓	✗	✓	✓	✗	✗

* meas. : measurement

2.6 Summary

This chapter provides an in-depth literature review on the MFC for single-agent and multi-agent dynamic systems. Subsequently, a review on the cooperative localization algorithms is presented. According to the presented literature review, the research gaps are revealed and presented in Section 2.5. Recalling the reviewed literature, it has been motivated that the tracking control of a nonlinear system can be accomplished with a model-free control paradigm without primarily depending on the full-knowledge of the system dynamics. In the previous works, the nonlinear dynamics is assumed as a lumped parameter vector which can be estimated online to be used in the control loop. The estimation does not require either full or partial information about the system nonlinearities. As it has been previously articulated, the use of ANNs (in place of the model-based approach) to estimate the nonlinearities in the system dynamics is also hindered by its design compromise, i.e. estimation accuracy at the expense of complexity, thereby increasing the computational effort.

In this regard, here in this thesis, the development of adaptive MFC (AMFC)

algorithm is investigated to solve tracking control problem involving continuous-time nonlinear dynamic systems with completely unknown nonlinearities working under unknown bounded disturbances. The proposed AMFC features online adaptive estimations which are driven by the tracking error (not the estimation error). The following features for the proposed AMFC algorithm should be emphasized;

- the adaptive laws are novel, since they are regressor-free estimators, which means that there is not any regressor parameters and hence the requirement of PE condition is eliminated;
- there is not any preliminary process for choosing the regressors or basis functions, just before the implementation of the AMFC protocols;
- the updated values of unknown linear terms are incorporated for online determination of the main controller gains using the differential Riccati equation (DRE), which provides the minimum efforts for off-line tuning of the controller parameters;
- the UUB convergence is confirmed for the estimation errors as well as the tracking error.

Later in this thesis, according to the second research gap presented in Section 2.5, the presented AMFC protocol for a MIMO single-agent system has been extended to a multi-agent system containing completely unknown continuous-time nonlinear dynamic agents. This development consists of a cooperative AMFC algorithm which doesn't require the inter-agent relative state measurements among the neighboring agents in the network. As it is mentioned in Section 2.3, the problem of measuring or

estimating the relative states among the agents in a network is an open problem and lots of investigations are ongoing. Moreover, a cooperative AMFC algorithm is presented which needs the inter-agent relative state measurements. Both algorithms are decentralized cooperative controllers that include optimal terms and the main controller gains are updated online. In addition, the adaptive laws are regressor-free estimators. These cooperative AMFC algorithms can be used for achieving consensus and formation-tracking objectives in completely unknown nonlinear multi-agent dynamic systems.

Finally and recalling the third research gap presented in Section 2.5, a cooperative localization algorithm is developed in the current thesis, so as to provide the estimated position vector of dynamic agents in a network with only one beacon agent. In the proposed adaptive cooperative localization algorithm, the dynamics of agents are assumed to be unknown and minimum number of communication links among the agents are required. The relative bearing measurements among the neighboring agents are not necessary and only the measurement of relative distance and relative velocity vector need to be available. Moreover, the position of agents are estimated in local and global frames, both. Combining all the proposed algorithms throughout this thesis, a design package would be available for control and localization of completely unknown multi-agent dynamic systems (including a team of AMRs).

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Introduction

This chapter begins with the design of an AMFC for a nonlinear system (single agent) with both SISO (in Section 3.2) and MIMO (in Section 3.3) unknown dynamics. In AMFC algorithm, the unknown nonlinearities are estimated by two separate regressor-free adaptive laws. According to the LaSalle-Yoshizawa theorem (see *Appendix F* for more details), the estimation of the nonlinearities in the system dynamic can be guaranteed bounded by proving that the estimation error convergence is UUB. Stability analyses are provided for the proposed AMFC protocol in SISO systems (in *Theorem 3-1*) and in MIMO systems (in *Theorem 3-2*). Furthermore by the virtue of an HJB equation, it is shown that the proposed AMFC algorithm is innovated to allow a compromise between achieving an excellent tracking error and reducing the minimal effort.

Then, the developed AMFC algorithm is extended to achieve consensus and formation-tracking objectives in multi-agent systems. In this way, two decentralized cooperative approaches are investigated.

- First in Section 3.4, a formation-tracking problem is solved for a network of completely unknown dynamic agents without the need for inter-agent relative position information. In that approach (named as CAMFC-1), it is assumed that the time-varying reference trajectory and the time-varying desired formation

topology are not available at all agents. Information on the virtual leader is only accessible to a subset of agents (pinned agents). Thus, to recover that information, a cooperative observer is designed which utilizes the communication graph properties. The stability analysis for the cooperative observer is provided in *Theorem 3-4*, based on Lyapunov stability theorem. By the help of a robustification term in the cooperative observer, the mentioned parameters are estimated in a finite time.

Utilizing the observed values for the desired trajectory and the formation topology parameters, the formation-tracking problem is transformed to N tracking problems corresponding to N agents in the network. In other words, the desired path at each agent is determined by utilizing the observed reference trajectory and the formation topology parameters. Then, the previously proposed AMFC is used at each agent in the network to follow the desired path. The proposed cooperative formation-tracking controller based on the novel AMFC algorithm is a distributed solution which works even under the condition of *simply-connected* communication graph.

Most of the work on formation-tracking problem involved fixed or time-varying reference trajectory but with a fixed desired formation topology. The work being presented in this thesis expounds the novelty in cooperative solution whereby both reference trajectory and the desired formation topology are time-varying. In the first cooperative control algorithm, the absolute states information (such as the position and velocity for an AMR) are however needed by each agent to accomplish the demand objective. But, eventually in the second approach, the absolute information is exempted and be replaced with inter-agent relative

position information.

- In the second cooperative control approach presented in Section 3.5, the consensus problem is solved for a network of completely unknown MIMO dynamic agents using the distributed model-free paradigm. The main contribution of this approach (named as CAMFC-2) is proposing a cooperative AMFC for consensus problem of unknown nonlinear systems without the use of any ANNs or FIS. Moreover, the inter-agent relative position information is utilized in this approach.

In this second scheme of cooperative control approach, the model-free adaptive laws for online parameter estimation at each agent incorporate communication graph information. Similar to the case of AMFC in a single dynamic system, a set of DREs are proposed for updating the main controller gains at the agents in the network. The stability analysis is presented in *Theorem 3-5* with LaSalle-Yoshizawa stability theorem and an UUB convergence is achieved. In addition, a cooperative optimality analysis based on the solution of a HJB equation is presented to show the inclusion of an optimal term in the proposed cooperative AMFC protocol. The HJB equation contains the information of the network's communication graph. Moreover, an observer similar to the cooperative observer designed in the first cooperative control approach is proposed for all the follower agents, which are not pinned to the virtual leader. The observer estimates the leader's states and control inputs, locally at each agent.

The main difference between the above cooperative control approaches is that the first one needs an accurate absolute position information, while the second approach

needs only the inter-agent relative position data. It is to note that inter-agent relative position information is practically preferable for a cooperative control system as compared to absolute state information. As reviewed in Chapter 2, to-date, cooperative localisation accomplished so far is with at least 3 mobile agents (one of them is the beacon agent). In Section 3.6, a novel cooperative localisation algorithm is presented for the first time exploiting the minimum number of communication links among the agents. This makes the feature of achieving comprehensive level of autonomy for a group of AMRs complete.

In the presented solution, an adaptive estimation algorithm is proposed for estimating the relative position of two moving agents using the measurements on relative velocity and the inter-agent relative distance. Then, a cooperative observer (similar to the two previous cooperative observers) is designed for estimating the absolute positions of the agents in the network using the estimated relative positions between the neighboring agents and the absolute position of only beacon agent in the network. The solution is applicable to a network of mobile agents with any number of agents. The Lyapunov stability theorem is utilized for convergence analysis for different parts of the ACL algorithm as in *Theorem 3-7* and *Lemma 3-8*. Each of the proposed algorithms throughout this chapter is accompanied by a schematic and a table, generating more appropriate presentation. The design flow imposed throughout this chapter is presented in Fig. 3.1

Design flow

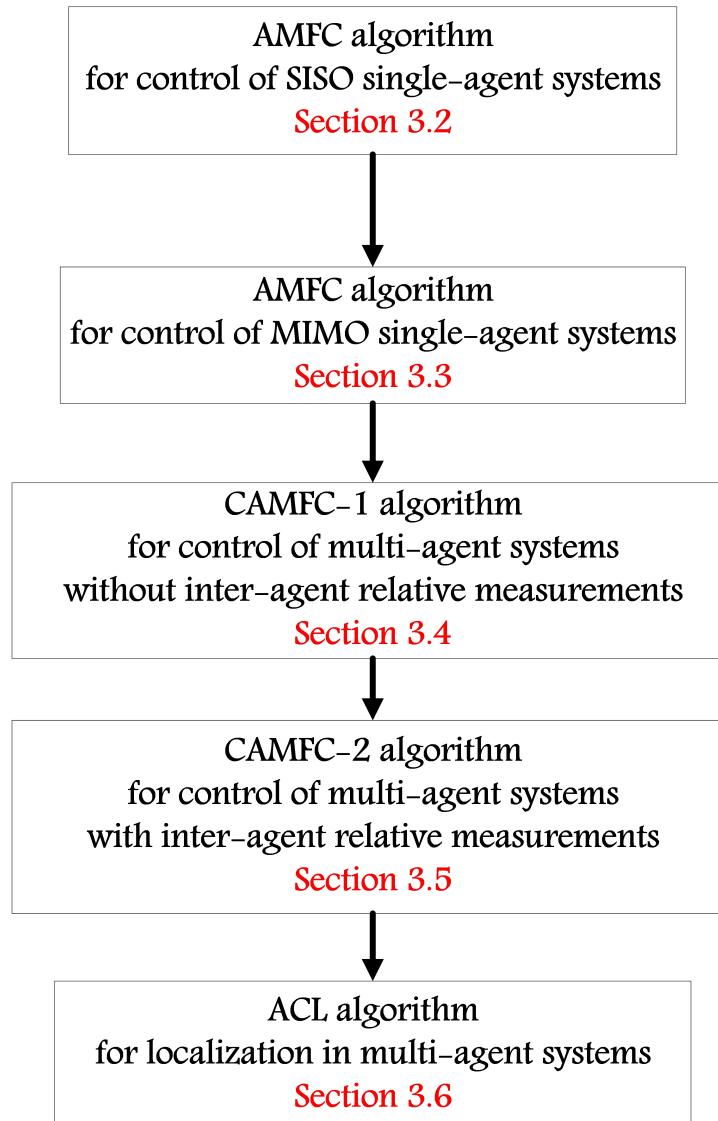


Figure 3.1: The diagram showing the design flow in Chapter 3. The design process is started with presenting AMFC for SISO and MIMO dynamic single-agent systems. Then, it is extended for multi-agent systems with and without interagent relative state measurements. Finally, the cooperative localization algorithm is presented.

3.2 AMFC for a single SISO nonlinear dynamic system

This section includes the design procedure of the AMFC algorithm for a SISO continuous-time nonlinear dynamic system.

3.2.1 Problem definition

Here, the nonlinear dynamics of a generic unknown SISO system as well as the corresponding tracking control problem are defined.

Definition 3-1. The unknown nonlinear dynamics of a SISO nonlinear system can be represented using the following ultra-local model (Fliess & Join, 2013)

$$\dot{x}_s = f_s(x_s) + bu_s , \quad (3.1a)$$

$$y_s = x_s , \quad (3.1b)$$

where $x_s \in \mathbb{R}$ is the system state, $f_s : \mathbb{R} \rightarrow \mathbb{R}$ is the unknown Lipschitz continuous nonlinear function (refer to *Appendix C*) depending only on x_s (f_s also can include bounded disturbances), parameter $b \in \mathbb{R}$ is a non-zero non-physical priori-known constant input gain which is chosen by the practitioner such that bu_s and \dot{x}_s are of the same order (Fliess & Join, 2013) (the value of b can be considered as 1, without any lack of generality), $u_s \in \mathbb{R}$ is the control input and $y_s \in \mathbb{R}$ is the system output. Without loss of generality, one can express f_s as follows

$$f_s = ax_s + h , \quad (3.2)$$

where $a = a(t) \in \mathbb{R}$ is an unknown time-varying state gain which is bounded and

Lipschitz; and $h = h(x) \in \mathbb{R}$ is another unknown Lipschitz continuous nonlinear function. According to (3.2), unknown dynamics f_s is composed of linear-in-states portion, i.e. ax_s , and nonlinear term h . Hence, the system proposed in (3.1a) can be alternatively presented as follows

$$\dot{x}_s = ax_s + bu_s + h , \quad (3.3a)$$

$$y_s = x_s . \quad (3.3b)$$

Definition 3-2. Considering a time-varying reference signal y_s^d , the tracking error is defined

$$e_s = y_s^d - y_s = y_s^d - x_s . \quad (3.4)$$

The control tracking objective is defined as converging e_s to zero, when time goes to infinity. Moreover, a joint cost function is defined as follows

$$\varsigma = e_s + \vartheta , \quad (3.5)$$

where $\vartheta = \int e_s dt$. By accompanying the tracking error with its time integral, the steady-state error can be eliminated (Tan et al., 2000).

Lemma 3-1 (Atassi & Khalil, 1999). Based on the *Separation Principle*, the combination of a stable controller and a stable observer leads to a stable dynamic system. For further readings, please refer to (Atassi & Khalil, 1999).

Lemma 3-2 (Levant, 2003). Recalling the sliding-mode differentiator, the

derivative of a reference signal y_s^d can be estimated as

$$\dot{y}_s^d = v_1 , \quad (3.6)$$

where

$$\dot{v} = v_1 , \quad (3.7a)$$

$$v_1 = -k_1 |v - y_s^d|^{1/2} sgn(v - y_s^d) + v_2 , \quad (3.7b)$$

$$\dot{v}_2 = -k_2 sgn(v - y_s^d) , \quad (3.7c)$$

with $k_1 > 0$ and $k_2 > 0$ are two constant parameters for tuning the sliding-mode differentiator and $sgn(\cdot)$ is the signum function. The differentiator presented in (3.6) and (3.7b), is going to be used in the design procedure of AMFC, wherever \dot{y}_s^d is required. For further readings, refer to (Levant, 2003).

3.2.2 Stability analysis

Theorem 3-1. For the SISO dynamic system presented by (3.3a), if one can construct the control input $u_s = u_s^1 + u_s^2$ as follows

$$u_s^1 = \frac{1}{2} r b p \zeta , \quad (3.8a)$$

$$u_s^2 = \frac{1}{b} (\dot{y}_s^d - \hat{a}(x_s - \zeta) - \hat{h} - \vartheta + (1 + \frac{2q}{p})\zeta) - \frac{3}{4} r b p \zeta , \quad (3.8b)$$

where $r > 0$, $q > 0$ are two constant values and $p > 0$ is updated online using the following scalar DRE

$$\dot{p} = 2\hat{a}p - rb^2 p^2 + 2q , \quad p(0) > 0 , \quad (3.9)$$

with the following adaptive laws

$$\dot{\hat{h}} = -\gamma_1 p \varsigma - \varpi_1 \gamma_1 \hat{h}, \quad (3.10a)$$

$$\dot{\hat{a}} = -\gamma_2 p \varsigma (x_s - \varsigma) - \varpi_2 \gamma_2 \hat{a}, \quad (3.10b)$$

where γ_1 , γ_2 , ϖ_1 and ϖ_2 are constant positive gains; then the tracking objective presented in *Definition 3-2* will be achieved.

Proof. By considering the estimation errors $\tilde{h} = h - \hat{h}$ and $\tilde{a} = a - \hat{a}$, define the following Lyapunov function

$$V_1 = \frac{1}{2} p \varsigma^2 + \frac{1}{2\gamma_1} \tilde{h}^2 + \frac{1}{2\gamma_2} \tilde{a}^2. \quad (3.11)$$

The time derivative of V_1 is

$$\dot{V}_1 = p \varsigma \dot{\varsigma} + \frac{1}{2} \varsigma^2 \dot{p} + \frac{1}{\gamma_1} \dot{\tilde{h}} \tilde{h} + \frac{1}{\gamma_2} \dot{\tilde{a}} \tilde{a}. \quad (3.12)$$

Then, we have

$$\dot{V}_1 = p \varsigma (\dot{y}_s^d - ax_s - bu_s - h + e_s) + \frac{1}{2} \varsigma^2 \dot{p} + \frac{1}{\gamma_1} \dot{\tilde{h}} \tilde{h} + \frac{1}{\gamma_2} \dot{\tilde{a}} \tilde{a}. \quad (3.13)$$

By adding and subtracting $pa\varsigma^2$ to the right-hand side of (3.13), it will be written as

$$\dot{V}_1 = p \varsigma (\dot{y}_s^d - a(x_s - \varsigma) - bu_s - h + e_s - a\varsigma) + \frac{1}{2} \varsigma^2 \dot{p} + \frac{1}{\gamma_1} \dot{\tilde{h}} \tilde{h} + \frac{1}{\gamma_2} \dot{\tilde{a}} \tilde{a}. \quad (3.14)$$

Then, also by adding and subtracting $p\varsigma(\hat{h} + \hat{a}(x_s - \varsigma))$ to the right-hand side of (3.14),

we have

$$\begin{aligned}\dot{V}_1 = p\zeta(\ddot{y}_s^d - \hat{a}(x_s - \zeta) - bu_s - \hat{h} + e_s - a\zeta) + \frac{1}{2}\zeta^2\dot{p} + (\frac{1}{\gamma_1}\dot{\tilde{h}}\tilde{h} - p\zeta\tilde{h}) + \\ (\frac{1}{\gamma_2}\dot{\tilde{a}}\tilde{a} - p\zeta(x_s - \zeta)\tilde{a}).\end{aligned}\quad (3.15)$$

Hence,

$$\begin{aligned}\dot{V} = p\zeta(y_s^d - \hat{a}(x_s - \zeta) - bu_s - \hat{h} + e_s - a\zeta) + \frac{1}{2}\zeta^2\dot{p} + (\frac{1}{\gamma_1}\dot{h}\tilde{h} - \frac{1}{\gamma_1}\dot{\tilde{h}}\tilde{h} - p\zeta\tilde{h}) + \\ (\frac{1}{\gamma_2}\dot{a}\tilde{a} - \frac{1}{\gamma_2}\dot{\tilde{a}}\tilde{a} - p\zeta(x_s - \zeta)\tilde{a}).\end{aligned}\quad (3.16)$$

Referring to *Definition 3-1*, $a = a(t)$ and $h = h(x(t))$ are time-varying but bounded, i.e.

$\dot{h} \neq 0$ and $\dot{a} \neq 0$. By adding and subtracting

$$s_1 = \frac{1}{4\varpi_1}(\frac{1}{\gamma_1}\dot{h} + \varpi_1 h)^2 + \varpi_1 \tilde{h}^2 + \varpi_1 \tilde{h}\hat{h}\quad (3.17)$$

and

$$s_2 = \frac{1}{4\varpi_2}(\frac{1}{\gamma_2}\dot{a} + \varpi_2 a)^2 + \varpi_2 \tilde{a}^2 + \varpi_2 \tilde{a}\hat{a}\quad (3.18)$$

and also the term $p\zeta\vartheta$, one leads to

$$\begin{aligned}
\dot{V}_1 = & p\zeta(\dot{y}_s^d - \hat{a}(x_s - \zeta) - bu_s - \hat{h} - \vartheta + (1-a)\zeta) + \frac{1}{2}\zeta^2\dot{p} - \\
& ((\boldsymbol{\varpi}_1\hat{h} + \frac{1}{\gamma_1}\dot{\hat{h}} + p\zeta)\tilde{h}) - (\frac{1}{4\boldsymbol{\varpi}_1}(\frac{1}{\gamma_1}\dot{h} + \boldsymbol{\varpi}_1 h)^2 + (\sqrt{\boldsymbol{\varpi}_1}\tilde{h})^2 - \\
& 2(\sqrt{\boldsymbol{\varpi}_1}\tilde{h})(\frac{1}{2\sqrt{\boldsymbol{\varpi}_1}})(\rho_1\tilde{h} + \rho_1\hat{h} + \frac{1}{\gamma_1}\dot{h})) + \frac{1}{4\boldsymbol{\varpi}_1}(\frac{1}{\gamma_1}\dot{h} + \boldsymbol{\varpi}_1 h)^2 - \\
& ((\boldsymbol{\varpi}_2\hat{a} + \frac{1}{\gamma_2}\dot{\hat{a}} + p\zeta(x_s - \zeta))\tilde{a}) - (\frac{1}{4\boldsymbol{\varpi}_2}(\frac{1}{\gamma_2}\dot{a} + \boldsymbol{\varpi}_2 a)^2 + (\sqrt{\boldsymbol{\varpi}_2}\tilde{a})^2 - \\
& 2(\sqrt{\boldsymbol{\varpi}_2}\tilde{a})(\frac{1}{2\sqrt{\boldsymbol{\varpi}_2}})(\boldsymbol{\varpi}_2\tilde{a} + \boldsymbol{\varpi}_2\hat{a} + \frac{1}{\gamma_2}\dot{a})) + \frac{1}{4\boldsymbol{\varpi}_2}(\frac{1}{\gamma_2}\dot{a} + \boldsymbol{\varpi}_2 a)^2. \quad (3.19)
\end{aligned}$$

Using the adaptive laws proposed in (3.10a) and (3.10b), the third and sixth terms in (3.19) are zero. Hence, one can reach to

$$\begin{aligned}
\dot{V}_1 = & p\zeta(\dot{y}_s^d - \hat{a}(x_s - \zeta) - bu_s - \hat{h} - \vartheta + (1-a)\zeta) + \frac{1}{2}\zeta^2\dot{p} + \\
& \frac{1}{4\boldsymbol{\varpi}_1}(\frac{1}{\gamma_1}\dot{h} + \boldsymbol{\varpi}_1 h)^2 - ((\sqrt{\boldsymbol{\varpi}_1}\tilde{h}) - \frac{1}{2\sqrt{\boldsymbol{\varpi}_1}}(\frac{1}{\gamma_1}\dot{h} + \boldsymbol{\varpi}_1 h))^2 + \\
& \frac{1}{4\boldsymbol{\varpi}_2}(\frac{1}{\gamma_2}\dot{a} + \boldsymbol{\varpi}_2 a)^2 - ((\sqrt{\boldsymbol{\varpi}_2}\tilde{a}) - \frac{1}{2\sqrt{\boldsymbol{\varpi}_2}}(\frac{1}{\gamma_2}\dot{a} + \boldsymbol{\varpi}_2 a))^2. \quad (3.20)
\end{aligned}$$

Since f_s is bounded and Lipschitz, h and a are two bounded Lipschitz nonlinear variables (refer to *Definition 3-1*). Consequently, we have $|h| \leq M_h$, $|\dot{h}| \leq M_{\dot{h}}$, $|a| \leq M_a$, $|\dot{a}| \leq M_{\dot{a}}$, where M_h , $M_{\dot{h}}$, M_a , $M_{\dot{a}}$ are four positive constants. Thus, the following inequality is achieved

$$\dot{V}_1 \leq p\zeta(\dot{y}_s^d - \hat{a}(x_s - \zeta) - bu_s - \hat{h} - \vartheta + (1-a)\zeta) + \frac{1}{2}\zeta^2\dot{p} - H_1 + \delta, \quad (3.21)$$

where

$$H_1 = ((\sqrt{\varpi_1} \tilde{h}) - \frac{1}{2\sqrt{\varpi_1}} (\frac{1}{\gamma_1} \dot{h} + \varpi_1 h))^2 + ((\sqrt{\varpi_2} \tilde{a}) - \frac{1}{2\sqrt{\varpi_2}} (\frac{1}{\gamma_2} \dot{a} + \varpi_2 a))^2 \quad (3.22)$$

and

$$\delta = \frac{1}{4\varpi_1} (\frac{1}{\gamma_1} M_{\dot{h}} + \varpi_1 M_h)^2 + \frac{1}{4\varpi_2} (\frac{1}{\gamma_2} M_{\dot{a}} + \varpi_2 M_a)^2. \quad (3.23)$$

Note that δ is a positive constant. Furthermore, by replacing $u_s = u_s^1 + u_s^2$ as proposed in (3.8a) and (3.8b), one can have

$$\dot{V}_1 \leq p\zeta((-\frac{2q}{p}\zeta) + (-a\zeta) + (-\frac{1}{2}rb^2p\zeta) + (+\frac{3}{4}rb^2p\zeta)) + \frac{1}{2}\zeta^2\dot{p} - H_1 + \delta. \quad (3.24)$$

The equation in (3.24) can be written as

$$\dot{V}_1 \leq \zeta^2(-2q - ap + \frac{1}{4}rb^2p^2 + \frac{1}{2}\dot{p}) - H_1 + \delta. \quad (3.25)$$

Then, by utilizing

$$\dot{p} = 2ap - rb^2p^2 + 2q, \quad (3.26)$$

we reach to

$$\dot{V}_1 \leq -(q + \frac{1}{4}rb^2p^2)\zeta^2 - H_1 + \delta. \quad (3.27)$$

Here, u_s^1 can be considered as the *tracking part* of the control signal, while u_s^2 is the *compensating part*. Finally, by defining $H_2 = \{(q + \frac{1}{4}rb^2p^2)\zeta^2 + H_1\} > 0$, one leads to

$$\dot{V}_1 \leq -H_2 + \delta. \quad (3.28)$$

Referring to the LaSalle-Yoshizawa theorem (refer to *Appendix F*), V_1 is UUB. Since

V_1 includes the tracking error and the estimation errors, one can deduce that ζ , \tilde{h} and \tilde{a} converge to a small radially bounded space around origin. Moreover, since \tilde{a} is converging to zero, we can use \hat{a} instead of a in (3.26) by recalling *Lemma 3-1*. Then, one would reach to (3.9). This completes the proof. ■

Remark 3-1. The parameter δ defined in (3.23), can be expanded as

$$\begin{aligned} \delta = & \left(\left(\frac{1}{4\varpi_1\gamma_1^2} M_h^2 \right) + \left(\frac{1}{4} \varpi_1 M_h^2 \right) + \left(\frac{1}{2\gamma_1} M_h M_a \right) \right) + \\ & \left(\left(\frac{1}{4\varpi_2\gamma_2^2} M_a^2 \right) + \left(\frac{1}{4} \varpi_2 M_a^2 \right) + \left(\frac{1}{2\gamma_2} M_a M_h \right) \right). \quad (3.29) \end{aligned}$$

By choosing γ_1 and γ_2 to be large enough and also defining ϖ_1 and ϖ_2 adequately small, the value of δ will decrease and can reach to zero. Therefore, the convergence of ζ , \tilde{a} and \tilde{h} can be satisfied faster.

Remark 3-2. The value of p as the main gain for proposed controller and adaptive laws in *Theorem 3-1*, is updated online utilizing (3.9). Referring to this property, there is not too much effort to tune other controller gains (i.e. r , q) off-line. Moreover, since the value of p is computed using the DRE in (3.26), it is confirmed that $p > 0$ which is a requirement for the stability analysis proposed in *Theorem 3-1*. For further discussions, refer to (Deshpande, 2011; Lewis et al., 2007).

3.3 AMFC for a single MIMO nonlinear dynamic system

This section includes the design procedure of the AMFC algorithm for a MIMO continuous-time nonlinear dynamic system. The stability and optimality analyses are presented.

3.3.1 Problem definition

Here, the nonlinear dynamics of a generic unknown MIMO system, the required auxiliary control variables and the tracking control problem are defined.

Definition 3-3. Consider a general unknown MIMO nonlinear dynamic system as

$$\dot{x} = f_0(x, u_0) , \quad (3.30a)$$

$$y = C_0 x , \quad (3.30b)$$

where $x \in \mathbb{R}^{n_s \times 1}$ is the vector of system states, $u_0 \in \mathbb{R}^{n_c \times 1}$ is the vector of control inputs, $f_0(\cdot) \in \mathbb{R}^{n_s \times 1}$ (here and afterwards, the notation “(.)” is used to denote a function of several variable for brevity) is the vector of unknown bounded nonlinear Lipschitz continuous functions depending on both x and u_0 , $C_0 \in \mathbb{R}^{n_o \times n_s}$ is the output matrix and $y \in \mathbb{R}^{n_o \times 1}$ is the vector of system outputs. The system has n_s states, n_c control inputs and n_o outputs. Assuming that all the states are measurable and the nonlinear system is *output controllable*, one can have a state-feedback controller $u_0 = u_0(x)$ to make the system stable (Hou & Jin, 2014). In this regard, one can represent (3.30a) as follows

$$\dot{x} = f_0(x, u_0(x)) - B_0 u_0(x) + B_0 u_0(x) , \quad (3.31)$$

where $B_0 = [b_{0ij}] \in \mathbb{R}^{n_s \times n_c}$ is an input gain matrix defined as

$$b_{0ij} = \begin{cases} 0 & , \text{ if } \dot{x}_i \text{ does not depend on } u_{0j} \\ 1 & , \text{ if } \dot{x}_i \text{ depends on } u_{0j} . \end{cases} \quad (3.32)$$

The only information required about the dynamic system is the relation between the states and the control inputs, qualitatively and not quantitatively. In other words, the control designer should specify whether each system states depends on each of the control inputs or not; and consequently, the matrix B_0 can be constructed by this information. The function f_0 can be partitioned into two components; a linear-in-states term and nonlinear function term, i.e.

$$f_0(x, u_0(x)) = Ax + f(x, u_0(x)) . \quad (3.33)$$

Here, $A = A^T \in \mathbb{R}^{n_s \times n_s}$ is a diagonal matrix including unknown time-varying gains on its main diameter. Here, superscript “ T ” is to denote the matrix transpose. In this sense, one leads to

$$\dot{x} = Ax + f(x, u_0(x)) - B_0u_0(x) + B_0u_0(x) . \quad (3.34)$$

Finally, by considering

$$g(x) = f(x, u_0(x)) - B_0u_0(x) , \quad (3.35)$$

the unknown nonlinear system in (3.30a) can be presented as

$$\dot{x} = Ax + B_0u_0 + g , \quad (3.36)$$

where $g = g(x) \in \mathbb{R}^{n_s \times 1}$ is a vector of unknown bounded Lipschitz continuous nonlinear functions. It should be noted that the off-diagonal elements of A are considered to be zero (this consideration has been made before in (Horowitz & Tomizuka, 1986)). In other words, the coupling terms which relate the dynamics of

each state to another states are assumed to be included in the lumped nonlinear functions in g . Utilizing the structure defined in (3.36), a technique can be proposed for online determination of the controller gains.

Remark 3-3. A system is said to be *output controllable* at a given input-output data point at current time, if the output of the system can be driven to a feasible specified setting point within finite time by a sequence of control inputs (Hou & Jin, 2014).

Definition 3-4. For any time-varying reference signal $y_0^d \in \mathbb{R}^{n_o \times 1}$, the tracking error is defined as

$$e_0 = y_0^d - y . \quad (3.37)$$

The value of tracking error can be determined according to the reference trajectory and the measured system outputs (which are available by a set of on-board sensors). The tracking objective will be achieved when e_0 converges to zero.

Definition 3-5. Here, a technique is defined for making the matrix B_0 to be a full-rank square matrix. In this sense, a vector of virtual control parameters $u^* \in \mathbb{R}^{(n_s - n_o) \times 1}$ is defined to be merged with the real control variables as follows

$$u = [u^* \ u_0]^T . \quad (3.38)$$

In (3.38), $u \in \mathbb{R}^{n_s \times 1}$ is a new vector of control inputs including the virtual controllers and the main control input parameters. It should be noted that for under-actuated dynamic systems (such as quadrotors) where $n_o > n_c$, the number of control inputs can be increased virtually in an initial step in order to make all the system states be

controllable, either directly or indirectly. Desired values for the internal states which are not considered as the system outputs, are set to be the desired values for the corresponding virtual controllers. Then, one can define

$$y^d = [y_0^d \ u^\star]^T \quad (3.39)$$

and consequently

$$e = y^d - x . \quad (3.40)$$

Note that here all of the system states in x are considered in computing the tracking error. This is against the partial inclusion of the system states for defining e_0 as in (3.37). Moreover, a joint cost-function is defined as

$$\sigma = e + \zeta , \quad (3.41)$$

where $\zeta = \int e dt \in \mathbb{R}^{n_s \times 1}$. By accompanying the tracking error with its time integral, both transient and steady-state portions of the tracking error are considered in the design procedure for AMFC algorithm.

Definition 3-6. According to *Definition 3-5*, the dynamics proposed in (3.36) can be presented as

$$\dot{x} = Ax + Bu + g , \quad (3.42a)$$

$$y = x , \quad (3.42b)$$

where $B = [b_{ij}] \in \mathbb{R}^{n_s \times n_s}$ is a gain matrix defined as

$$b_{ij} = \begin{cases} 0 & , \text{ if } \dot{x}_i \text{ does not depend on } u_j \\ 1 & , \text{ if } \dot{x}_i \text{ depends on } u_j . \end{cases} \quad (3.43)$$

It can be seen that the matrix B is full-rank. Later in the simulation results, it is shown that $B = I_n$ is used in most of the practical cases, to denote the system full controllability.

Definition 3-7. A vectorizing function $\mathcal{V}(.)$ is defined for generating a vector $l \in \mathbb{R}^{n_0 \times 1}$ (for any $n_0 \in \mathbb{R}^+$) constructed by diagonal elements of a matrix $M \in \mathbb{R}^{n_0 \times n_0}$ as follows (Zhu et al., 2016)

$$l = \mathcal{V}(M) = v_M , \quad (3.44)$$

where $l[i] = M[i, i]$ for $i = \{1, 2, \dots, n_0\}$.

Definition 3-8. A function $\mathcal{M}(.)$ is defined for generating a diagonal matrix $M \in \mathbb{R}^{n_0 \times n_0}$ with all off-diagonal elements be equal to zero. The diagonal elements of M are constructed by the elements of a vector $l \in \mathbb{R}^{n_0 \times 1}$ as follows

$$M = \mathcal{M}(l) = \mathcal{M}_l , \quad (3.45)$$

where $M[i, i] = l[i]$ for $i = \{1, 2, \dots, n_0\}$.

Lemma 3-3 (Levant, 2003). Similar to *Lemma 3-1*, by utilizing a sliding-mode differentiator for a MIMO dynamic system, one can compute the derivative of reference vector signal as

$$\dot{y}^d = v , \quad (3.46)$$

where for $y^d = [y_1^d, y_2^d, \dots, y_n^d]^T$, we have ($i = \{1, 2, \dots, n\}$)

$$\dot{v}^i = v_1^i, \quad (3.47a)$$

$$v_1^i = -k_1 |v^i - y_i^d|^{\frac{1}{2}} \operatorname{sgn}(v^i - y_i^d) + v_2^i, \quad (3.47b)$$

$$\dot{v}_2^i = -k_2 \operatorname{sgn}(v^i - y_i^d), \quad (3.47c)$$

and $v = [v^1, v^2, \dots, v^n]^T$, $v_1 = [v_1^1, v_1^2, \dots, v_1^n]^T$ and $v_2 = [v_2^1, v_2^2, \dots, v_2^n]^T$ are three vectors in $\mathbb{R}^{n \times 1}$. Here, $n = n_s$ is the total number of system states. In the following analysis, the differentiator presented in (3.47b) has been used for computing \dot{y}^d for MIMO dynamic systems, wherever required.

3.3.2 Stability analysis

Theorem 3-2. For the dynamic system proposed in (3.42a), if one can define the controller $u = u_1 + u_2$ as

$$u_1 = \frac{1}{2} RB^T P \sigma, \quad (3.48a)$$

$$u_2 = B^{-1} [\dot{y}^d - \hat{A}x - \hat{g} - \zeta + (I_n + 2P^{-1}Q + \hat{A})\sigma] - \frac{3}{4} RB^T P \sigma, \quad (3.48b)$$

where I_n is the identity matrix with dimensions of $n = n_s$, $R \in \mathbb{R}^{n \times n}$ and $Q \in \mathbb{R}^{n \times n}$ are constant positive definite matrices and $P = P^T \in \mathbb{R}^{n \times n}$ ($P = P(t)$, but here and afterwards the notation " (t) " is omitted just for sake of simplicity) is a positive definite matrix defined using

$$\dot{P} = \hat{A}^T P + P \hat{A} - P B R B^T P + 2Q, \quad (3.49)$$

with incorporating the following adaptive laws

$$\dot{\hat{g}} = -\Gamma_1 P \sigma - \rho_1 \Gamma_1 \hat{g}, \quad (3.50a)$$

$$\dot{v}_{\hat{A}} = -\Gamma_2 P \mathcal{M}_\sigma(x - \sigma) - \rho_2 \Gamma_2 v_{\hat{A}}, \quad (3.50b)$$

where Γ_1 and Γ_2 are two positive definite diagonal matrices in $\mathbb{R}^{n \times n}$ including the adaptive gains, ρ_1 and ρ_2 are two positive scalar leakage gains and $v_{\hat{A}} = \mathcal{V}(\hat{A})$, $v_{\hat{A}} = \mathcal{V}(\hat{A})$ and $\mathcal{M}_\sigma = \mathcal{M}(\sigma)$ (according to *Definition 3-7* and *Definition 3-8*); then the tracking objective in *Definition 3-5* will be achieved.

Proof. Consider the estimation errors $\tilde{g} = g - \hat{g}$ and $\tilde{A} = A - \hat{A}$. Note that since the values of g and A are unknown, the values of \tilde{g} and \tilde{A} are also totally unknown and cannot be estimated or measured. Define the following Lyapunov function (Khalil, 1996)

$$V_3 = \frac{1}{2} \sigma^T P \sigma + \frac{1}{2} \tilde{g}^T \Gamma_1^{-1} \tilde{g} + \frac{1}{2} v_{\tilde{A}}^T \Gamma_2^{-1} v_{\tilde{A}}, \quad (3.51)$$

where $v_{\tilde{A}} = \mathcal{V}(\tilde{A})$. Continuing by the time derivative of V_3 , the following is achieved

$$\dot{V}_3 = \sigma^T P \dot{\sigma} + \frac{1}{2} \sigma^T \dot{P} \sigma + \tilde{g}^T \Gamma_1^{-1} \dot{\tilde{g}} + v_{\tilde{A}}^T \Gamma_2^{-1} \dot{v}_{\tilde{A}}. \quad (3.52)$$

Then by considering (3.42a), one may lead to

$$\dot{V}_3 = \sigma^T P [\ddot{y}^d - Ax - Bu - g + e] + \frac{1}{2} \sigma^T \dot{P} \sigma + \tilde{g}^T \Gamma_1^{-1} \dot{\tilde{g}} + v_{\tilde{A}}^T \Gamma_2^{-1} \dot{v}_{\tilde{A}}. \quad (3.53)$$

By adding and subtracting $\sigma^T PA\sigma$, one reaches to

$$\dot{V}_3 = \sigma^T P[\ddot{y}^d - A(x - \sigma) - Bu - g + e - A\sigma] + \frac{1}{2}\sigma^T \dot{P}\sigma + \tilde{g}^T \Gamma_1^{-1} \tilde{g} + v_{\tilde{A}}^T \Gamma_2^{-1} \dot{v}_{\tilde{A}}. \quad (3.54)$$

Subsequent adding and subtracting $\sigma^T P(\hat{A}(x - \sigma) + \hat{g})$ in (3.54) leads to,

$$\begin{aligned} \dot{V}_3 = & \sigma^T P[\ddot{y}^d - \hat{A}(x - \sigma) - Bu - \hat{g} + e - A\sigma] + \frac{1}{2}\sigma^T \dot{P}\sigma + [\tilde{g}^T \Gamma_1^{-1} \tilde{g} - \sigma^T P\tilde{g}] + \\ & [v_{\tilde{A}}^T \Gamma_2^{-1} \dot{v}_{\tilde{A}} - \sigma^T P\tilde{A}(x - \sigma)]. \end{aligned} \quad (3.55)$$

Moreover, by considering $\dot{\tilde{g}} = \dot{g} - \dot{\hat{g}}$ and $\dot{v}_{\tilde{A}} = \dot{v}_A - \dot{v}_{\hat{A}}$, one can have

$$\begin{aligned} \dot{V}_3 = & \sigma^T P[\ddot{y}^d - \hat{A}(x - \sigma) - Bu - \hat{g} + e - A\sigma] + \frac{1}{2}\sigma^T \dot{P}\sigma + \\ & [\tilde{g}^T \Gamma_1^{-1} \dot{g} - \tilde{g}^T \Gamma_1^{-1} \dot{\hat{g}} - \sigma^T P\tilde{g}] + [v_{\tilde{A}}^T \Gamma_2^{-1} \dot{v}_A - v_{\tilde{A}}^T \Gamma_2^{-1} \dot{v}_{\hat{A}} - \sigma^T P\tilde{A}(x - \sigma)], \end{aligned} \quad (3.56)$$

where $\dot{v}_A = \mathcal{V}(\dot{A})$ and $\dot{v}_{\hat{A}} = \mathcal{V}(\dot{\hat{A}})$. It can be shown easily that

$$\sigma^T P\tilde{A}(x - \sigma) = v_{\tilde{A}}^T P\mathcal{M}_\sigma(x - \sigma). \quad (3.57)$$

Recalling *Definition 3-3*, we have $\dot{g} \neq 0$ and $\dot{v}_A \neq 0$. Then, by adding and subtracting

$$S_2 = \frac{1}{4\rho_1} [\Gamma_1^{-1} \dot{g} + \rho_1 g]^T [\Gamma_1^{-1} \dot{g} + \rho_1 g] + \rho_1 \tilde{g}^T \tilde{g} + \rho_1 \tilde{g}^T \hat{g} \quad (3.58)$$

and also

$$S_3 = \frac{1}{4\rho_2} [\Gamma_2^{-1} \dot{v}_A + \rho_2 v_A]^T [\Gamma_2^{-1} \dot{v}_A + \rho_2 v_A] + \rho_2 v_{\tilde{A}}^T v_{\tilde{A}} + \rho_2 v_{\tilde{A}}^T v_{\hat{A}}, \quad (3.59)$$

in right-hand side of (3.56), the following is achieved

$$\begin{aligned}\dot{V}_3 = & \boldsymbol{\sigma}^T P [\dot{y}^d - \hat{A}(x - \boldsymbol{\sigma}) - Bu - \hat{g} + e - A\boldsymbol{\sigma}] + \frac{1}{2} \boldsymbol{\sigma}^T \dot{P} \boldsymbol{\sigma} - \\ & \tilde{g}^T [\rho_1 \hat{g} + \Gamma_1^{-1} \dot{\hat{g}} + P\boldsymbol{\sigma}] + S_4 - v_{\tilde{A}}^T [\rho_2 v_{\hat{A}} + \Gamma_2^{-1} \dot{v}_{\hat{A}} + P\mathcal{M}_{\boldsymbol{\sigma}}(x - \boldsymbol{\sigma})] + S_5 , \quad (3.60)\end{aligned}$$

where

$$\begin{aligned}S_4 = & \frac{1}{4\rho_1} [\Gamma_1^{-1} \dot{g} + \rho_1 g]^T [\Gamma_1^{-1} \dot{g} + \rho_1 g] - \left\{ \frac{1}{4\rho_1} [\Gamma_1^{-1} \dot{g} + \rho_1 g]^T [\Gamma_1^{-1} \dot{g} + \rho_1 g] + \right. \\ & \left. \rho_1 \tilde{g}^T \tilde{g} - 2(\sqrt{\rho_1} \tilde{g}^T) \frac{1}{(2\sqrt{\rho_1})} [\rho_1 \tilde{g} + \rho_1 \hat{g} + \Gamma_1^{-1} \dot{g}] \right\} \quad (3.61)\end{aligned}$$

and

$$\begin{aligned}S_5 = & \frac{1}{4\rho_2} [\Gamma_2^{-1} \dot{v}_A + \rho_2 v_A]^T [\Gamma_2^{-1} \dot{v}_A + \rho_2 v_A] - \left\{ \frac{1}{4\rho_2} [\Gamma_2^{-1} \dot{v}_A + \rho_2 v_A]^T [\Gamma_2^{-1} \dot{v}_A + \rho_2 v_A] + \right. \\ & \left. \rho_2 v_{\tilde{A}}^T v_{\tilde{A}} - 2(\sqrt{\rho_2} v_{\tilde{A}}^T) \frac{1}{(2\sqrt{\rho_2})} [\rho_2 v_{\tilde{A}} + \rho_2 v_{\hat{A}} + \Gamma_2^{-1} \dot{v}_A] \right\} . \quad (3.62)\end{aligned}$$

Utilizing the adaptive laws proposed in (3.50a) and (3.50b), the terms

$$\tilde{g}^T [\rho_1 \hat{g} + \Gamma_1^{-1} \dot{\hat{g}} + P\boldsymbol{\sigma}] \quad (3.63)$$

and

$$v_{\tilde{A}}^T [\rho_2 v_{\hat{A}} + \Gamma_2^{-1} \dot{v}_{\hat{A}} + P\mathcal{M}_{\boldsymbol{\sigma}}(x - \boldsymbol{\sigma})] \quad (3.64)$$

in (3.60) will be zero. In other hand, since g and A are bounded Lipschitz functions (according to *Definition 3-3*), one can have $|g| \leq L_g$, $|\dot{g}| \leq L_{\dot{g}}$, $|v_A| \leq L_A$ and $|v_{\dot{A}}| \leq L_{\dot{A}}$ where L_g , $L_{\dot{g}}$, L_A and $L_{\dot{A}}$ are four constant positive vectors in $\mathbb{R}^{n \times 1}$ (note that, $|\cdot|$

is a symbol to present a vector consisting the absolute values of the elements in the corresponding vector). Consequently, the following inequalities are achieved

$$S_4 \leq -H_4 + \delta_1 , \quad (3.65a)$$

$$S_5 \leq -H_5 + \delta_2 , \quad (3.65b)$$

where

$$H_4 = \left(\frac{1}{2\sqrt{\rho_1}} [\Gamma_1^{-1} \dot{g} + \rho_1 g] - \sqrt{\rho_1} \tilde{g} \right)^T \left(\frac{1}{2\sqrt{\rho_1}} [\Gamma_1^{-1} \dot{g} + \rho_1 g] - \sqrt{\rho_1} \tilde{g} \right) , \quad (3.66a)$$

$$H_5 = \left(\frac{1}{2\sqrt{\rho_2}} [\Gamma_2^{-1} \dot{v}_A + \rho_2 v_A] - \sqrt{\rho_2} v_{\tilde{A}} \right)^T \left(\frac{1}{2\sqrt{\rho_2}} [\Gamma_2^{-1} \dot{v}_A + \rho_2 v_A] - \sqrt{\rho_2} v_{\tilde{A}} \right) , \quad (3.66b)$$

and

$$\delta_1 = \frac{1}{4\rho_1} [\Gamma_1^{-1} L_{\dot{g}} + \rho_1 L_g]^T [\Gamma_1^{-1} L_{\dot{g}} + \rho_1 L_g] , \quad (3.67a)$$

$$\delta_2 = \frac{1}{4\rho_2} [\Gamma_2^{-1} L_{\dot{A}} + \rho_2 L_A]^T [\Gamma_2^{-1} L_{\dot{A}} + \rho_2 L_A] . \quad (3.67b)$$

By incorporating (3.65a) and (3.65b) in (3.60), one leads to

$$\dot{V}_3 \leq \boldsymbol{\sigma}^T P [\dot{y}^d - \hat{A}(x - \boldsymbol{\sigma}) - Bu - \hat{g} + e - A\boldsymbol{\sigma}] + \frac{1}{2} \boldsymbol{\sigma}^T \dot{P} \boldsymbol{\sigma} - H_3 + \delta_3 , \quad (3.68)$$

where $H_3 = H_4 + H_5$ and $\delta_3 = \delta_1 + \delta_2$. It should be noted that δ_3 is a constant positive scalar. Now, by replacing $u = u_1 + u_2$ from (3.48a) and (3.48b), we have

$$\dot{V}_3 \leq -\boldsymbol{\sigma}^T [2Q + PA + \frac{1}{2} PBRB^T P - \frac{3}{4} PBRB^T P - \frac{1}{2} \dot{P}] \boldsymbol{\sigma} - H_3 + \delta_3 . \quad (3.69)$$

Since

$$\boldsymbol{\sigma}_2^T PA\boldsymbol{\sigma} = \frac{1}{2} \boldsymbol{\sigma}^T PA\boldsymbol{\sigma} + \frac{1}{2} \boldsymbol{\sigma}^T A^T P \boldsymbol{\sigma} , \quad (3.70)$$

the equation in (3.69) can be written as follows

$$\dot{V}_3 \leq -\boldsymbol{\sigma}^T [2Q + \frac{1}{2}PA + \frac{1}{2}A^TP - \frac{1}{4}PBRB^TP - \frac{1}{2}\dot{P}]\boldsymbol{\sigma} - H_3 + \delta_3 . \quad (3.71)$$

Then, by considering P to be derived by the following DRE

$$\dot{P} = A^TP + PA - PBRB^TP + 2Q , \quad (3.72)$$

one would have

$$\dot{V}_3 \leq -H_6 + \delta_3 , \quad (3.73)$$

where

$$H_6 = H_3 + \boldsymbol{\sigma}^T [Q + \frac{1}{4}PBRB^TP]\boldsymbol{\sigma} . \quad (3.74)$$

Recalling the LaSalle-Yoshizawa theorem, V_3 is UUB. Therefore, $\boldsymbol{\sigma}$, \tilde{g} and $v_{\tilde{A}}$ converge to a small radially bounded space around the origin. Here, the proposed adaptive laws in (3.50a) and (3.50b) for estimating the unknown terms \hat{g} and \hat{A} , are gradient-based algorithms (Ioannou & Fidan, 2006). Moreover by referring to *Lemma 3-1*, one can use \hat{A} instead of A in (3.72). Thus, the equation in (3.49) is achieved. This completes the proof.

■

Remark 3-4. The parameters δ_1 and δ_2 in (3.67a) and (3.67b), can be expanded as

$$\delta_1 = (\frac{1}{4\rho_1}L_g^T\Gamma_1^{-2}L_g) + (\frac{1}{4}\rho_1 L_g^T L_g) + (\frac{1}{2}L_g^T\Gamma_1^{-1}L_g) \quad (3.75)$$

and

$$\delta_2 = \left(\frac{1}{4\rho_2} L_{\dot{A}}^T \Gamma_2^{-2} L_{\dot{A}} \right) + \left(\frac{1}{4} \rho_2 L_A^T L_A \right) + \left(\frac{1}{2} L_{\dot{A}}^T \Gamma_2^{-1} L_A \right). \quad (3.76)$$

Note that Γ_1 and Γ_2 are diagonal matrices in $\mathbb{R}^{n \times n}$ including $n - n_c$ adaptive gains on the main diameter, corresponding to the virtual controllers (refer to *Definition 3-5*) and n_c adaptive gains corresponding to the real control inputs. By choosing the diagonal elements in Γ_1 and Γ_2 large enough (smaller values should be set for the elements corresponding to the virtual control inputs) and also some small values for ρ_1 and ρ_2 , the values for δ_1 and δ_2 and consequently δ_3 will be small, appropriately. This leads to better convergence performance of the adaptive laws in (3.50a) and (3.50b) and consequently better tracking performance for the proposed AMFC algorithm for MIMO dynamic systems.

Remark 3-5. Reminding that the off-diagonal elements of P are zero, the diagonal values in P as the main gains for proposed AMFC in *Theorem 3-2*, are updated online utilizing (3.49). Hence, there is minimal effort for off-line tuning of any other constants in the AMFC controller for MIMO dynamic systems.

Remark 3-6. Based on *Lemma 3-1* and recalling that the controlled system in (3.42a) and the adaptive laws in (3.50a) and (3.50b) are stable according to *Theorem 3-2*, one can present the dynamic system in (3.42a) as follows

$$\dot{x} = \hat{A}x + Bu + \hat{g}. \quad (3.77)$$

Remark 3-7. The number of adaptive laws suggested by (3.50a) and (3.50b),

depends only on the number of system states and is equal to $2n$, where n is the total number of states in the system.

3.3.3 Optimality analysis

In this section, it is shown that the proposed AMFC algorithm in the preceding section includes an optimal term. This would make the comparative analysis between the AMFC algorithm and the novel RL solutions investigated in the literature reasonable.

Definition 3-9. For the dynamic system proposed in (3.42a) with the tracking objective presented in *Definition 3-5*, a *cost-to-go* function is defined as

$$J_2 = \int_t^\infty L_2(\sigma, u) d\tau \quad (3.78)$$

where

$$L_2(\sigma, u) = \sigma^T Q \sigma + u_1^T R^{-1} u_1 \quad (3.79)$$

is the *utility* function which is the cost for performance of the system at each time step.

Lemma 3-4. The cost function J_2 for time t to infinity can be presented as

$$J_2 = \frac{1}{2} \sigma^T P \sigma \quad (3.80)$$

which is an energy function depends only on the cost at current time step t .

Proof. Consider

$$S_6 = \int_t^\infty \frac{d}{dt} \left[\frac{1}{2} \boldsymbol{\sigma}^T P \boldsymbol{\sigma} \right] d\tau . \quad (3.81)$$

One can have

$$S_6 = V_4(\infty) - V_4(t) \quad (3.82)$$

where

$$V_4(t) = \frac{1}{2} \boldsymbol{\sigma}^T P \boldsymbol{\sigma} . \quad (3.83)$$

Recalling *Theorem 3-2* and complying with *Remark 3-4*, we know that V_4 reaches to zero when time goes to infinity, i.e. $V_4(\infty) = 0$. Hence,

$$S_6 = -V_4(t) . \quad (3.84)$$

Then, by adding S_6 to (3.78) and subtracting $-V_4(t)$ from this equation, one leads to

$$J_2 = V_4(t) + \int_t^\infty \left[\frac{d}{dt} \left(\frac{1}{2} \boldsymbol{\sigma}^T P \boldsymbol{\sigma} \right) + \boldsymbol{\sigma}^T Q \boldsymbol{\sigma} + u_1^T R^{-1} u_1 \right] d\tau \quad (3.85)$$

and then

$$J_2 = V_4(t) + \int_t^\infty \left[\boldsymbol{\sigma}^T P (\dot{x}_d - \hat{A}x - Bu - \hat{g} + e) + \boldsymbol{\sigma}^T Q \boldsymbol{\sigma} + u_1^T R^{-1} u_1 \right] d\tau . \quad (3.86)$$

Here, the equation in (3.77) has been used as the representation of the system dynamics. By replacing u from (3.48a) and (3.48b) in (3.86), one leads to

$$\begin{aligned} J_2 = V_4(t) + \int_t^\infty & \left\{ \boldsymbol{\sigma}^T \left[-P\hat{A} - 2Q - \frac{1}{2} PBRB^T P + \frac{3}{4} PBRB^T P + \frac{1}{2} \dot{P} + Q \right] \boldsymbol{\sigma} \right. \\ & \left. + \frac{1}{4} \boldsymbol{\sigma}^T PBRB^T P \boldsymbol{\sigma} \right\} d\tau . \end{aligned} \quad (3.87)$$

Moreover, by replacing \dot{P} from (3.49) and also recalling (3.70), the term inside the integral in (3.87) is zero. Then, one can reach to (3.80). This completes the proof. ■

Proposition 3-1. Based on the HJB equation (Lewis et al., 2012), the optimal control for a MIMO dynamic system proposed in (3.42a), should satisfy

$$0 = \min_{u=u_{op}} \{L_2(\sigma, u) + \frac{d}{dt}J_2(\sigma)\}, \quad (3.88)$$

where L_2 and J_2 are defined in (3.79) and (3.80).

Theorem 3-3. The designed control input u in (3.48a) and (3.48b) includes a part that satisfies the optimality condition presented in *Proposition 3-1*.

Proof. Consider

$$W_2 = L_2(\sigma, u) + \frac{d}{dt}J_2(\sigma). \quad (3.89)$$

By replacing L_2 and J_2 form (3.79) and (3.80) and recalling *Remark 3-6*, one reaches to

$$W_2 = \sigma^T Q \sigma + u_1^T R^{-1} u_1 + \sigma^T P [\dot{x}_d - \hat{A}x - Bu - \hat{g} + e] + \frac{1}{2} \sigma^T \dot{P} \sigma. \quad (3.90)$$

Then, by replacing u from (3.48a) and (3.48b) and some algebraic simplification, it leads to

$$W_2 = \sigma^T [Q + \frac{1}{4} P B R B^T P - 2Q - P \hat{A} - \frac{1}{2} P B R B^T P + \frac{3}{4} P B R B^T P + \frac{1}{2} \dot{P}] \sigma, \quad (3.91)$$

where it is zero by utilizing \dot{P} as proposed in (3.49). Moreover, by computing the gradient with respect to u_1 for both sides of (3.90), the following is achieved

$$\nabla_{u_1} W_2 = \frac{1}{2} u_1^T R^{-1} - \sigma^T P B . \quad (3.92)$$

This is zero by incorporating u_1 defined in (3.48a). Hence, u_1 makes W_2 to be minimum and the optimality condition in (3.88) is achieved. This completes the proof. ■

3.3.4 AMFC algorithm

Recalling *Theorem 3-1* and based on the presented *Theorem 3-2*, the AMFC algorithm for a MIMO nonlinear dynamic system is presented in Table 3.1. The algorithm requires an initializing procedure for some constant variables. Subsequently, the algorithm operates in the main control loop to track the time-varying reference trajectory. The desired reference trajectory y^d and the measured values of the system states x are the inputs variables for each step in the main loop and the output is the control signal u . In Table 3.1, $\underline{\mathbf{0}}_n \in \mathbb{R}^{n \times n}$ is a matrix with zero for all elements, $\mathbf{0}_n \in \mathbb{R}^{n \times 1}$ is a vector constituted by zero elements, $d_t > 0$ is a small scalar value as the sampling time of the controller and $t_f > 0$ is the final time for operation. Here, the forward Euler method is used for doing the integration in discrete time space. In addition, the values of adaptive and leakage gains are tuned according to the hints presented in *Remark 3-4*. The other controller parameters are set trivially as proposed in *Algorithm 1*. A schematic of the proposed AMFC algorithm is presented in Fig. 3.2.

Table 3.1: AMFC algorithm

Algorithm 1: AMFC
Initialization:
$k_1 = 1$, $k_2 = 1$
$B = I_n$, $R = I_n$, $Q = 0.1 \times I_n$
$P(0) = I_n$, $\hat{A}(0) = \underline{\mathbf{0}}_n$, $\hat{g}(0) = \mathbf{0}_n$, $\zeta(0) = \mathbf{0}_n$
$\{\Gamma_1, \Gamma_2, \rho_1, \rho_2\}$ are set according to <i>Remark 3-4</i> .
Note-1: $v_{\hat{A}} = \mathcal{V}(\hat{A})$ and $\mathcal{M}_{\sigma} = \mathcal{M}(\sigma)$ (according to <i>Definition 3-7</i> and <i>Definition 3-8</i>) .
Note-2: t_f is the simulation/implementation duration time, and dt is the sampling time .
Main Loop:
for $\{t = d_t : d_t : t_f\}$ do
Inputs: $\{x(t), y^d(t)\}$
1: $e(t) = y^d(t) - x(t)$
2: $\zeta(t) = \zeta(t - d_t) + \{e(t) \times d_t\}$
3: $\sigma(t) = e(t) + \zeta(t)$
4: $\dot{y}_d(t) = w(t)$;(based on <i>Lemma 3-3</i>)
5: $P(t) = P(t - d_t) + \{[\hat{A}^T(t - d_t) P(t - d_t) + P(t - d_t) \hat{A}(t - d_t) - P(t - d_t) B R B^T P(t - d_t) + 2Q] \times d_t\}$
6: $\hat{g}(t) = \hat{g}(t - d_t) + \{[-\Gamma_1 P(t) \sigma(t) - \rho_1 \Gamma_1 \hat{g}(t - d_t)] \times d_t\}$
7: $v_{\hat{A}}(t) = v_{\hat{A}}(t - d_t) + \{[-\Gamma_2 P(t) \mathcal{M}_{\sigma}(t) [x(t) - \sigma(t)] - \rho_2 \Gamma_2 v_{\hat{A}}(t - d_t)] \times d_t\}$
8: $\hat{A}(t) = diag(v_{\hat{A}(t)})$
9: $u_1(t) = \frac{1}{2} R B^T P(t) \sigma(t)$
10: $u_2(t) = B^{-1} [\dot{y}^d(t) - \hat{A}(t)x(t) - \hat{g}(t) - \zeta(t) + (I_n + 2P^{-1}(t)Q + \hat{A}(t))\sigma(t)] - \frac{3}{4} R B^T P(t) \sigma(t)$
11: $u(t) = u_1(t) + u_2(t)$
Outputs: $u(t)$
end for

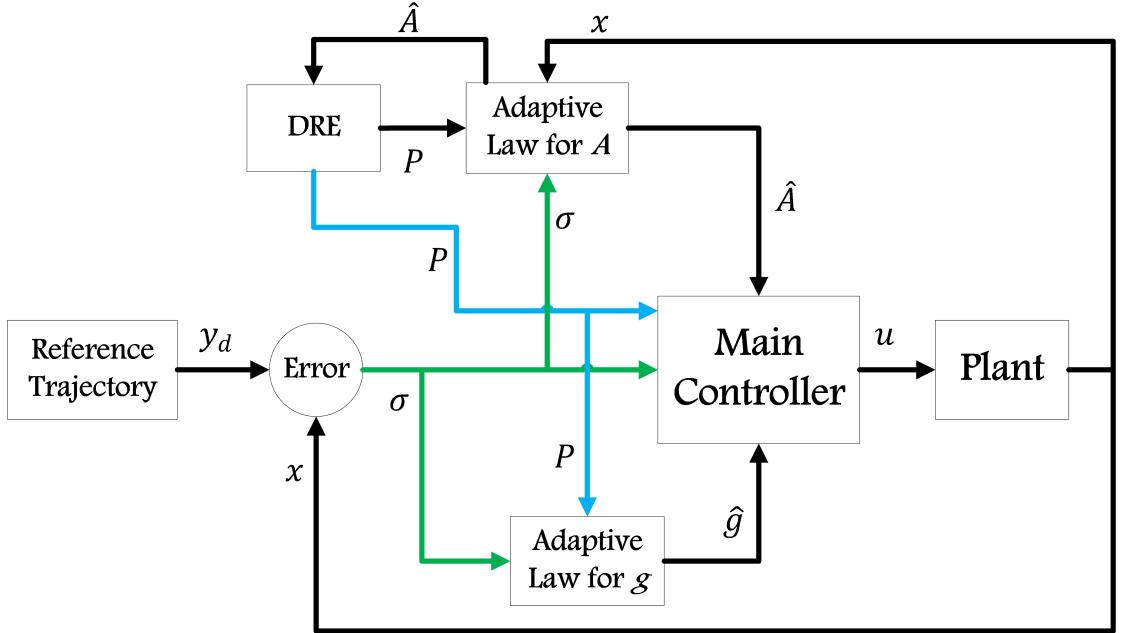


Figure 3.2: The schematic for operation of the AMFC algorithm for a MIMO nonlinear dynamic system. Different blocks are dedicated for implementation of the adaptive laws for online estimation of the unknown linear and nonlinear terms, as well as the online solution of the DRE.

3.4 Cooperative AMFC for formation-tracking problem within a multi-agent MIMO dynamic system without inter-agent relative measurements

In this section, the design procedure of a cooperative adaptive model-free control (CAMFC-1) policy is presented for a multi-agent dynamic system without inter-agent relative measurements in the network.

3.4.1 Problem definition

First in this subsection, the communication graph representing the inter-agent communication in the networked MIMO dynamic systems is defined by virtue of graph theory. This is the essential ingredient to be considered in the cooperative control design and synthesis, in particular, the Lyapunov analysis. Next, each agent MIMO dynamic system including the unknown nonlinearities is described.

Extrapolating the inherent nonlinearities across the network, the network-wise dynamic of the whole MIMO systems is defined with the mathematical aid from the communication graph. Lastly, the formation tracking problem for a network of MIMO dynamic systems is presented.

Definition 3-10. Consider a network consisting N homogeneous nonlinear dynamic agents. Assume $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$ be a *communication graph* with the set of N nodes $\mathcal{V} = (v_1, v_2, \dots, v_N)$, a set of edges $\mathcal{E} = [\varepsilon_{ij}] \in \mathbb{R}^{N \times N}$ and associated adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$. An edge ε_{ij} in \mathcal{G} is a link between a pair of nodes (v_i, v_j) , representing the flow of information from v_j to v_i . The ε_{ij} is in existence if and only if $a_{ij} > 0$. The graph is supposed to be undirected, i.e. the ε_{ij} and ε_{ji} in \mathcal{G} are the same. The v_i and v_j are considered as neighbors, if and only if $\varepsilon_{ij} \in \mathcal{E}$. The communication graph is connected, meaning that there is an undirected path between each pair of agents in the network. The *in-degree matrix* is defined as $\mathcal{D} = \text{diag}(d_1^{in}, d_2^{in}, \dots, d_N^{in}) \in \mathbb{R}^{N \times N}$, where each d_i is the input degree to each node, i.e. $d_i^{in} = \sum_{j=1}^N a_{ij}$. Hence, we can define the *Laplacian matrix* $\mathcal{L} = [L_{ij}] \in \mathbb{R}^{N \times N}$ as below (Lewis et al., 2014; Li & Duan, 2015)

$$\mathcal{L} = \mathcal{D} - \mathcal{A}. \quad (3.93)$$

Furthermore, by considering a *virtual leader* with access to the desired trajectory to be tracked by the whole network and also the formation parameters, one can define the *leader pinning gain matrix* as follows

$$\mathcal{B} = \text{diag}(\beta_1, \beta_2, \dots, \beta_N) \in \mathbb{R}^{N \times N}, \quad (3.94)$$

in which $\beta_i \in \{0, 1\}$ indicates the existence of a communication link between the virtual leader and the i th agent in the network (Lewis et al., 2014; Li & Duan, 2015).

Then, $\mathcal{H} \in \mathbb{R}^{N \times N}$ is defined as

$$\mathcal{H} = \mathcal{L} + \mathcal{B}. \quad (3.95)$$

Assumption 3-1. There is at least one communication connection between one of the agents in the network and the virtual leader. In other words, at least one of the diagonal elements in \mathcal{B} is non-zero.

Assumption 3-2. Here, it is assumed that the relative measurements between the states of neighboring agents are not available locally at each agent. Instead, each agent can measure its own states using the available on-board sensors.

Definition 3-11. Assume that each dynamic agent in the network has n states. Then, $\eta^i \in \mathbb{R}^{n \times 1}$ is defined as the time-varying formation parameter corresponding to agent i (0 is for the virtual leader). The values for elements of η^i are determined regarding the desired formation (i.e. separation) among the agents in the network in a mutual reference frame. These values are set to zero, when the *consensus* among all the agents states is the main objective in the cooperative control problem.

Definition 3-12. Utilizing *Definition 3-6*, the nonlinear dynamic system for the i th

agent in the network is assumed as

$$\dot{x}^i = A^i x^i + B u^i + g^i , \quad (3.96a)$$

$$y^i = x^i . \quad (3.96b)$$

As declared in *Definition 3-6*, A^i and g^i are the unknown linear and nonlinear terms at the agent i , while B is defined according to the existing relation between each system state and each of the control inputs.

Definition 3-13. The objective of the formation-tracking problem is to track a reference trajectory which is available at the virtual leader as its states (i.e x^0), while providing the demand formation topology as defined in *Definition 3-11* among the agents in the network. According to *Assumption 3-2*, this objective is going to be achieved in this section by assuming that the inter-agent relative measurements for the states of neighboring agents are not available.

3.4.2 Design procedure

In order to satisfy the formation-tracking objective presented in *Definition 3-13*, the desired trajectory and the demand formation topology should be reconstructed at each of the agents in the network. This is due to the fact that this data is only available at the virtual leader which is connected to a portion of the agents (not to all of the agents). Hence, the need for a cooperative observer arises to provide each agent in the network with the mentioned data in a distributed manner. Here, first a cooperative observer algorithm is proposed for estimating the reference trajectory, which is the state of virtual leader x^0 . This is later extended to estimate the values for formation

variables at agent i (i.e. η^i) in the network. The combination of these two distributed cooperative observers with the AMFC algorithm constructs the basis of CAMFC-1 algorithm.

Proposition 3-2. The estimation error for observing x_0 at agent i can be represented as follows

$$\tilde{x}_i^0 = \sum_{j=1}^N a_{ij}(\hat{x}_i^0 - \hat{x}_j^0) + b_i(\hat{x}_i^0 - x^0), \quad (3.97)$$

where $\hat{x}_i^0 \in \mathbb{R}^{n \times 1}$ includes the estimated values of x^0 at i th agent. The objective of reaching consensus on x_0 among the agents in the network is presented by

$$\lim_{t \rightarrow \infty} \tilde{\bar{x}}^0 = \mathbf{0}, \quad (3.98)$$

where $\mathbf{0}$ is a vector in $\mathbb{R}^{Nn \times 1}$ with zero elements and $\tilde{\bar{x}}^0 = [\tilde{x}_1^0; \tilde{x}_2^0; \dots; \tilde{x}_N^0] \in \mathbb{R}^{Nn \times 1}$ is defined as follows

$$\tilde{\bar{x}}^0 = (\mathcal{H} \otimes I_n) \bar{x}^0 - (\mathcal{B} \otimes x^0) \mathbf{1}. \quad (3.99)$$

Here, $\bar{x}^0 = [\hat{x}_1^0; \hat{x}_2^0; \dots; \hat{x}_N^0] \in \mathbb{R}^{Nn \times 1}$ and $\mathbf{1}$ is a vector in $\mathbb{R}^{N \times 1}$ with all elements equal to one.

Theorem 3-4. If one uses the following equation as the rate for observing x^0 at agent i

$$\dot{\hat{x}}_i^0 = -\lambda \hat{x}_i^0 - [\mathcal{M}(sgn\{\sum_{j=1}^N \mathcal{H}(i, j)\hat{x}_j^0\}) X_M], \quad (3.100)$$

where $\lambda > 0$ is a scalar gain, $\mathcal{M}(sgn\{\Sigma_{j=1}^N \mathcal{H}(i, j)\hat{x}_j^0\}) \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose elements on the main diameter are the sign of elements in $\Sigma_{j=1}^N \mathcal{H}(i, j)\hat{x}_j^0 \in \mathbb{R}^{n \times 1}$ (referring to *Definition 3-8*) and $X_M \in \mathbb{R}^{n \times 1}$ includes the maximum absolute

values for the elements of \dot{x}^0 ; then the consensus objective proposed in *Proposition 3-2* will be achieved.

Proof. Considering the following Lyapunov function

$$V_5 = \frac{1}{2}(\tilde{\bar{x}}^0)^T \tilde{\bar{x}}^0 , \quad (3.101)$$

and obtaining its corresponding derivatives,

$$\dot{V}_5 = (\tilde{\bar{x}}^0)^T [(\mathcal{H} \otimes I_n) \dot{\tilde{\bar{x}}}^0 - (\mathcal{B} \otimes \dot{x}^0) \mathbf{1}] . \quad (3.102)$$

Here, \otimes is the notation for *Kronecker product* (refer to *Appendix G*). Since the summation of all elements in each row of the Laplacian matrix is zero (Lewis et al., 2014; Li & Duan, 2015), one can say that

$$(\mathcal{L} \otimes \dot{x}^0) \mathbf{1} = \mathbf{0} . \quad (3.103)$$

Then by recalling (3.95), the equation in (3.102) can be represented as

$$\dot{V}_5 = (\tilde{\bar{x}}^0)^T (\mathcal{H} \otimes I_n) \dot{\tilde{\bar{x}}}^0 - (\tilde{\bar{x}}^0)^T (\mathcal{H} \otimes \dot{x}^0) \mathbf{1} . \quad (3.104)$$

Considering $\dot{\tilde{\bar{x}}}^0 = -\lambda \tilde{\bar{x}}^0 + \hat{s}$ where \hat{s} is defined as

$$\hat{s} = -\mathcal{M}(sgn\{(\tilde{\bar{x}}^0)^T (\mathcal{H} \otimes I_n)\})(I_N \otimes X_M) \mathbf{1} , \quad (3.105)$$

it leads to

$$\dot{V}_5 = -\lambda \tilde{x}_0^T (\mathcal{H} \otimes I_n) \tilde{x}^0 + (\tilde{x}^0)^T (\mathcal{H} \otimes I_n) \hat{s} - (\tilde{x}^0)^T (\mathcal{H} \otimes \dot{x}^0) \mathbf{1}. \quad (3.106)$$

Recalling *Definition 3-10* and *Assumption 3-1*, $(\mathcal{H} \otimes I_n)$ is a symmetric matrix with positive diagonal and non-positive off-diagonal elements. This means that, $(\mathcal{H} \otimes I_n)$ has positive determinant and positive eigenvalues. Hence, it is a non-singular M-matrix (Lewis et al., 2014; Li & Duan, 2015) (refer to *Appendix H* for more information). As a result, one can say that $(\mathcal{H} \otimes I_n) > 0$. Besides, let $\dot{V}_5 = \dot{V}_5^1 + \dot{V}_5^2$, where

$$\dot{V}_5^1 = -\lambda (\tilde{x}^0)^T (\mathcal{H} \otimes I_n) \tilde{x}^0 < 0 \quad (3.107)$$

and

$$\dot{V}_5^2 = (\tilde{x}^0)^T (\mathcal{H} \otimes I_n) \hat{s} - (\tilde{x}^0)^T (\mathcal{H} \otimes \dot{x}^0) \mathbf{1}. \quad (3.108)$$

To achieve $\dot{V}_5 < 0$, it should be shown that

$$\dot{V}_5^2 = (\tilde{x}^0)^T (\mathcal{H} \otimes I_n) \hat{s} - (\tilde{x}^0)^T (\mathcal{H} \otimes \dot{x}^0) \mathbf{1} \leq 0. \quad (3.109)$$

Recalling the mixed-product property of Kronecker product (refer to *Appendix G*), one would have

$$(\mathcal{H} \otimes \dot{x}^0) = (\mathcal{H} \otimes I_n)(I_N \otimes \dot{x}^0). \quad (3.110)$$

Hence, (3.109) can be written as follows

$$\dot{V}_5^2 = (\tilde{x}^0)^T (\mathcal{H} \otimes I_n) \hat{s} - (\tilde{x}^0)^T (\mathcal{H} \otimes I_n)(I_N \otimes \dot{x}^0) \mathbf{1}. \quad (3.111)$$

Then, it reaches

$$\dot{V}_5^2 \leq (\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n) \hat{s} + ABS((\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n)) (I_N \otimes X_M) \mathbf{1}, \quad (3.112)$$

where for $v \in \mathbb{R}^{1 \times Nn}$, one can define

$$ABS(v) = [|v(1)|, |v(2)|, \dots, |v(Nn)|], \quad (3.113)$$

where $ABS(\cdot)$ is a function to compute the absolute values for elements of a vector.

Now for satisfying (3.109), one should only show that

$$(\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n) \hat{s} + ABS((\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n)) (I_N \otimes X_M) \mathbf{1} = 0. \quad (3.114)$$

By replacing \hat{s} from (3.105), the first term on the left-hand side of (3.114) will be transformed as follows

$$(\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n) \hat{s} = -(\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n) \mathcal{M}(sgn\{(\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n)\}) (I_N \otimes X_M) \mathbf{1}, \quad (3.115)$$

where by recalling the definition in (3.113), we have

$$(\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n) \mathcal{M}(sgn\{(\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n)\}) = ABS((\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n)). \quad (3.116)$$

Thus, by replacing (3.115) and (3.116) in (3.114), the equality requirement in (3.114) is satisfied and then the rates for observed parameters are suggested as follows

$$\dot{\hat{x}}^0 = -\lambda \tilde{\vec{x}}^0 - \mathcal{M}(sgn\{(\tilde{\vec{x}}^0)^T (\mathcal{H} \otimes I_n)\}) (I_N \otimes X_M) \mathbf{1}. \quad (3.117)$$

By using $\dot{\tilde{x}}^0$ from (3.117), one would have $\dot{V}_5 < 0$, which in turn shows that the consensus error on observation of x_0 (i.e. \tilde{x}^0) is asymptotically stable and converges to zero, recalling the Lyapunov stability theorem (refer to *Appendix E*). Hence, the *Proposition 3-2* is achieved. Moreover, the observer for agent i can be represented as proposed in (3.100). This completes the proof.

■

Remark 3-8. According to (3.100), only the values on the i th row of \mathcal{H} are required to compute the rates for the cooperative observer parameters at agent i . These values are available locally at agent i , based on the existing communication links between the agent and its neighboring ones. In other words, only the information from the neighboring agents are required at the i th agent (no need for global information of the whole network).

Remark 3-9. The values for X_M can be determined according to the actuators specifications used in the agents. For example, if x^0 is the reference positions of a mobile robot, then X_M is the maximum absolute values for the mobile robot speed, which can be defined according to the actuators specifications and some data from previous experiments.

Remark 3-10. The value of λ should be large enough to increase the convergence rate of the proposed distributed observer in (3.100).

Lemma 3-5. Referring to *Definition 3-11*, let us define

$$\Omega = \begin{bmatrix} (\eta^0)^T \\ (\eta^1)^T \\ \dots \\ (\eta^N)^T \end{bmatrix} \in \mathbb{R}^{(N+1) \times n}. \quad (3.118)$$

Since the formation variables η^i 's are only available at the virtual leader and there is at-least one communication link between one of the agents and the virtual leader (refer to *Assumption 3-1*), one can propose an observer similar to (3.100) as follows

$$\dot{\hat{\Omega}}^i = -\mu \varepsilon^i - [(\operatorname{sgn}\{\sum_{j=1}^N \mathcal{H}(i, j)\varepsilon^j\}) \mathcal{M}(\Upsilon^M)], \quad (3.119)$$

to estimate and construct the formation variables Ω at agent i . In (3.118), $\mu > 0$ is a constant scalar, $\hat{\Omega}^i \in \mathbb{R}^{(N+1) \times n}$ includes the estimated formation variables at the i th agent, $|\dot{\eta}^0| \leq \Upsilon^M \in \mathbb{R}^{n \times 1}$ and

$$\varepsilon^i = \sum_{j=1}^N a_{ij}(\hat{\Omega}^i - \hat{\Omega}^j) + b_i(\hat{\Omega}^i - \Omega). \quad (3.120)$$

The proof procedure for this lemma is similar to the procedure presented in the proof of *Theorem 3-4*.

Proposition 3-3. Utilizing *Theorem 3-4* and *Lemma 3-5*, the reference trajectory and the desired formation topology among the agents are constructed synchronously at all agents in the network. In this regard, one can represent the formation-tracking problem in form of a simple tracking problem to be solved locally at each agent. In other words, by recalling *Lemma 3-1*, the desired path to be followed by the i th agent

in the network is defined as follows

$$y_d^i = \hat{x}_i^0 + \hat{\Omega}^i(1,:)^T + \hat{\Omega}^i(i,:)^T , \quad (3.121)$$

where \hat{x}_i^0 and $\hat{\Omega}^i$ are determined using the cooperative observers defined in (3.100) and (3.118). Furthermore, the local tracking error at agent i is defined as

$$e^i = y_d^i - y^i . \quad (3.122)$$

Also, the time-integral of this error is considered to eliminate the steady-state error and thus the following joint error variable is proposed

$$\sigma^i = e^i + \zeta^i , \quad (3.123)$$

where $\zeta^i = \int e^i dt$. The tracking objective at agent i is achieved when σ^i reaches to zero as time goes to infinity. As a secondary result of achieving the above tracking objective, the formation topology among the agents is satisfied according to (3.121). In other words, the formation topology in the network will be satisfied if there is a local controller at agent i that can track y_d^i proposed in (3.121).

Lemma 3-6. Recalling *Theorem 3-2* and *Assumption 3-2*, for the dynamic system proposed in (3.96a), if one uses the AMFC policy at agent i and defines $u^i = u_1^i + u_2^i$ as

$$u_1^i = \frac{1}{2}RB^TP^i\sigma^i , \quad (3.124a)$$

$$u_2^i = B^{-1}[y_d^i - \hat{A}^i x^i - \hat{g}^i - \zeta^i + (I_n + 2(P^i)^{-1}Q + \hat{A}^i)\sigma^i] - \frac{3}{4}RB^TP^i\sigma^i , \quad (3.124b)$$

where I_n , R and Q are defined same as in *Theorem 3-2*; and $P^i = (P^i)^T > 0 \in \mathbb{R}^{n \times n}$ is suggested by

$$\dot{P}^i = (\hat{A}^i)^T P^i + P^i \hat{A}^i - P^i B R B^T P^i + 2Q, \quad (3.125)$$

with incorporating the following adaptive laws

$$\dot{g}^i = -\Gamma_1 P^i \sigma^i - \rho_1 \Gamma_1 \dot{g}^i, \quad (3.126a)$$

$$v_{\dot{A}^i} = -\Gamma_2 P^i \mathcal{M}_{\sigma^i} (x^i - \sigma^i) - \rho_2 \Gamma_2 v_{\dot{A}^i}, \quad (3.126b)$$

where Γ_1 , Γ_2 , ρ_1 and ρ_2 are defined same as in *Theorem 3-2*; then the formation-tracking objective requested in *Proposition 3-3* will be achieved.

3.4.3 CAMFC-1 algorithm

The proposed cooperative AMFC algorithm in this section, which is named as the CAMFC-1 algorithm, is presented in Table 3.2 and Fig. 3.3. As mentioned before, the inter-agent relative state measurements are supposed to be unavailable in CAMFC-1 algorithm. But, the values for \hat{x}_i^0 , $\hat{\Omega}^i$, \tilde{x}_i^0 and ε^i should be transmitted from agent i to its neighboring agents via the existing communication links.

Table 3.2: CAMFC-1 algorithm

Algorithm 2: CAMFC-1 (at agent i in the network)
Initialization:

$\lambda = \mu$ are set at a large value , $X_M = \Upsilon_M$ are set at a small value

$k_1 = 1$, $k_2 = 1$, $B = I_n$, $R = I_n$, $Q = 0.1 \times I_n$

$P^i(0) = I_n$, $\hat{A}^i(0) = \underline{\mathbf{0}}_n$, $\hat{g}^i(0) = \mathbf{0}_n$, $\zeta^i(0) = \mathbf{0}_n$

$\{\Gamma_1, \Gamma_2, \rho_1, \rho_2\}$ are set according to *Remark 3-4* .

Note-1: t_f is the simulation/implementation duration time, and dt is the sampling time .

Main Loop:

for $\{t = d_t : d_t : t_f\}$ **do**

Inputs: $\{x^i(t), \hat{x}_i^0(t - d_t), \hat{\Omega}^i(t - d_t),$

$\hat{x}_j^0(t - d_t), \hat{\Omega}^j(t - d_t), \hat{x}_j^0(t - d_t), \varepsilon^j(t - d_t) \quad (j \neq i \in [1, N])\}$

1: $\tilde{x}_i^0(t) = \beta_i \times (\hat{x}_i^0(t - d_t) - x^0(t))$

2: $\varepsilon^i(t) = \beta_i \times (\hat{\Omega}^i(t - d_t) - \Omega(t))$

for $\{j = 1 : N\}$ **do**

 3-1: $\tilde{x}_i^0(t) = \tilde{x}_i^0(t) + a(i, j) \times (\hat{x}_i^0(t - d_t) - \hat{x}_j^0(t - d_t))$

 3-2: $\varepsilon^i(t) = \varepsilon^i(t) + a(i, j) \times (\hat{\Omega}^i(t - d_t) - \hat{\Omega}^j(t - d_t))$

end for

4: $\dot{\tilde{x}}_i^0(t) = -\lambda \tilde{x}_i^0(t)$

5: $\dot{\hat{\Omega}}^i(t) = -\mu \varepsilon^i(t)$

for $\{j = 1 : N\}$ **do**

 6-1: $\dot{\tilde{x}}_i^0(t) = \dot{\tilde{x}}_i^0(t) - \mathcal{M}(sgn\{\mathcal{H}(i, j)\tilde{x}_0^j(t - d_t)\}) \times X_M$

 6-2: $\dot{\hat{\Omega}}^i(t) = \dot{\hat{\Omega}}^i(t) - sgn\{\mathcal{H}(i, j)\varepsilon^j(t - d_t)\} \times \mathcal{M}(\Upsilon_M)$

end for

7: $\dot{\tilde{x}}_i^0(t) = \dot{\tilde{x}}_i^0(t - d_t) + \dot{\tilde{x}}_i^0(t) \times d_t$

8: $\dot{\hat{\Omega}}^i(t) = \dot{\hat{\Omega}}^i(t - d_t) + \dot{\hat{\Omega}}^i(t) \times d_t$

9: $y_d^i(t) = \hat{x}_i^0(t) + \hat{\Omega}^i(1, :)^T(t) + \hat{\Omega}^i(i, :)^T(t)$

10: $e^i(t) = y_d^i(t) - x^i(t)$

11: $\zeta^i(t) = \zeta^i(t - d_t) + \{e^i(t) \times d_t\}$

12: $\sigma^i(t) = e^i(t) + \zeta^i(t)$

13: $\dot{y}_d^i(t) = w^i(t) \quad (\text{based on Lemma 3-3})$

14: $P^i(t) = P^i(t - d_t) + \{[(\hat{A}^i)^T(t - d_t) P^i(t - d_t) + P^i(t - d_t) \hat{A}^i(t - d_t) - P^i(t - d_t) BRB^T P^i(t - d_t) + 2Q] \times d_t\}$

15: $\hat{g}^i(t) = \hat{g}^i(t - d_t) + \{[-\Gamma_1 P^i(t) \sigma^i(t) - \rho_1 \Gamma_1 \hat{g}^i(t - d_t)] \times d_t\}$

16: $v_{\hat{A}^i}(t) = v_{\hat{A}^i}(t - d_t) + \{[-\Gamma_2 P^i(t) \mathcal{M}_{\sigma^i}(t) [x^i(t) - \sigma^i(t)] - \rho_2 \Gamma_2 v_{\hat{A}^i}(t - d_t)] \times d_t\}$

17: $u_1^i(t) = \frac{1}{2} RB^T P^i(t) \sigma^i(t)$

18: $u_2^i(t) = B^{-1} [\dot{y}_d^i(t) - \hat{A}^i(t)x^i(t) - \hat{g}^i(t) - \zeta^i(t) + (I_n + 2(P^i)^{-1}(t)Q + \hat{A}^i(t))\sigma^i(t)] - \frac{3}{4} RB^T P^i(t) \sigma^i(t)$

19: $u^i(t) = u_1^i(t) + u_2^i(t)$

Outputs: $u^i(t), \hat{x}_i^0(t), \hat{\Omega}^i(t)$

end for

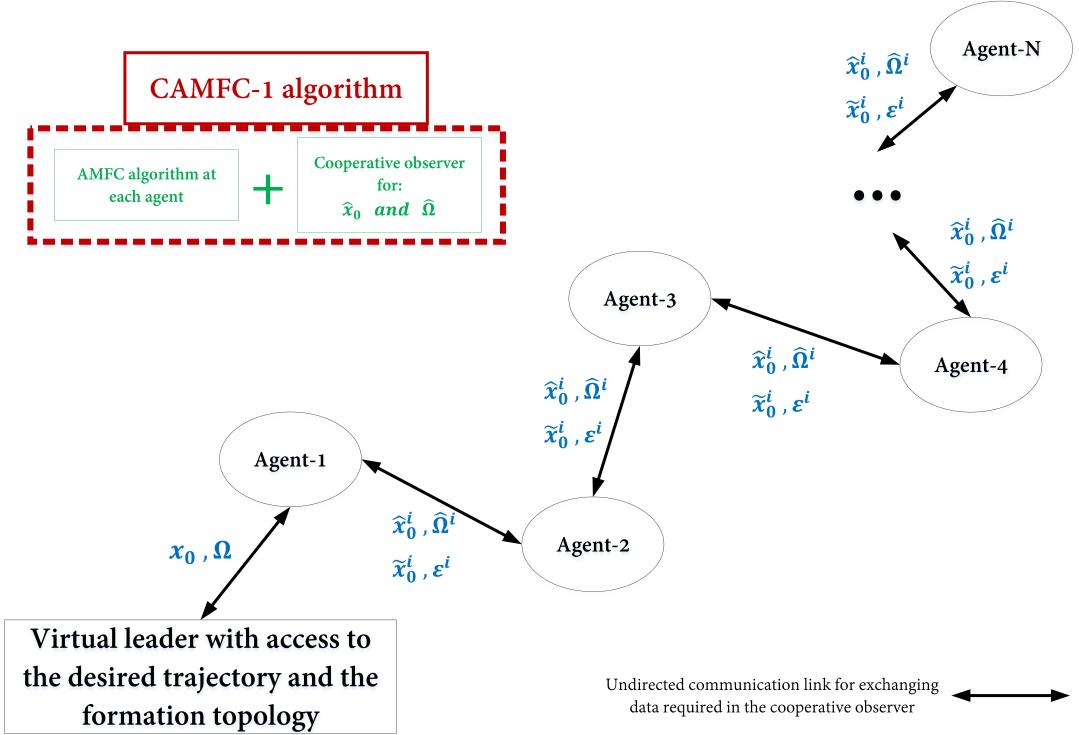


Figure 3.3: Schematic for operation of the CAMFC-1 algorithm for a multi-agent MIMO nonlinear dynamic system

3.5 Cooperative AMFC for formation-tracking problem within a multi-agent MIMO dynamic system using inter-agent relative measurements

In this section, a cooperative adaptive model-free control (CAMFC-2) algorithm is presented for a multi-agent system, in which absolute information about the systems' states is not available. Instead, the inter-agent relative state measurement, assumed available and can be used to carry out the formation-tracking problem using the formulated CAMFC-2. In this regard, the major difference between the CAMFC-1 algorithm proposed in Section 3.4 and the investigated CAMFC-2 solution in this section is the availability of the relative state measurements among the neighboring agents in the network.

3.5.1 Problem definition

Here, the formation-tracking control problem is defined for a multi-agent dynamic system accessing the inter-agent relative measurements of the states among the agents. In addition, the corresponding consensus errors are presented.

Definition 3-14. For a multi-agent system with the communication graph suggested in *Definition 3-10* and *Assumption 3-1*, and the agents dynamics as defined in *Definition 3-12*, one can define the dynamics of the whole network as follows

$$\dot{x}_c = A_c x_c + (I_N \otimes B) u_c + g_c , \quad (3.127)$$

where $A_c = \text{diag}(A^1, \dots, A^N) \in \mathbb{R}^{Nn \times Nn}$, $x_c = [x^1, \dots, x^N]^T \in \mathbb{R}^{Nn \times 1}$, $u_c = [u^1, \dots, u^N]^T \in \mathbb{R}^{Nn \times 1}$, $g_c = [g^1, \dots, g^N]^T \in \mathbb{R}^{Nn \times 1}$ and $I_N \in \mathbb{R}^{N \times N}$ is an identity matrix. Note that u^i includes the real control inputs and the virtual control inputs at agent i , as declared in *Definition 3-5*.

Definition 3-15. Dynamics of the virtual leader for the network defined in *Definition 3-14*, is assumed as follows

$$\dot{x}^0 = u^0 , \quad (3.128)$$

where, $x^0 \in \mathbb{R}^{n \times 1}$ is the states vector for the leader and $u^0 \in \mathbb{R}^{n \times 1}$ is the vector of control inputs for the leader. The values for elements of u^0 are defined as time-derivative of the reference trajectory (i.e. x^0) in the state space of agents, which should be followed by the entire network.

Assumption 3-3. It is assumed that the inter-agent relative measurements for the states of the neighboring agents are available in the network defined in *Definition 3-14* and *Definition 3-15*. These measurements can be provided using the on-board sensors and especial observer algorithms.

Proposition 3-4. Considering the formation-tracking objective within the network defined in *Definition 3-14* and *Definition 3-15*, having access to the desired formation topology parameters is an important issue. Since the values for formation parameters are not available at all agents in the network (by recalling *Assumption 3-1*), here the cooperative observer suggested in *Lemma 3-5* are utilized for estimation of these parameters locally at each of the agents. In this regard, ${}^i\hat{\eta}^j$ as the estimated formation parameters of the j th agent ($j = 0$ is for the formation parameters of the virtual leader) at the i th agent, is proposed as follows

$${}^i\dot{\hat{\eta}}^j = -\mu {}^i\tilde{\eta}^j - [\mathcal{M}(sgn\{\sum_{k=1}^N (\mathcal{H}(i,k) {}^k\tilde{\eta}^j)\}) \Upsilon^M], \quad (3.129)$$

where $|{}^i\dot{\hat{\eta}}^j| = |{}^0\dot{\eta}^j| \leq \Upsilon^M \in \mathbb{R}^{n \times 1}$ and

$${}^i\tilde{\eta}^j = \sum_{k=1}^N a_{ik}({}^i\hat{\eta}^j - {}^k\hat{\eta}^j) + \beta_i({}^i\hat{\eta}^j - \eta^j). \quad (3.130)$$

Here, $\eta^j \in \mathbb{R}^{n \times 1}$ includes the formation parameters of the j th agent available at the virtual leader.

Definition 3-16. For the network defined in *Definition 3-14* and *Definition 3-15* and by considering *Assumption 3-3* and *Proposition 3-4*, a consensus error is defined

for the formation-tracking problem as follows (Lewis et al., 2014)

$$e^i = \sum_{i=1}^N a_{ij} [({}^i\hat{\eta}^i - {}^i\hat{\eta}^j) - (x^i - x^j)] + \beta_i [({}^i\hat{\eta}^i - {}^i\hat{\eta}^0) - (x^i - x^0)] , \quad (3.131)$$

where a_{ij} and β_i are suggested in *Definition 3-10* and ${}^i\hat{\eta}^j$ s are defined in *Proposition 3-4*. This error can be rearranged as

$$e^i = \sum_{i=1}^N a_{ij} [({}^i\hat{\eta}^i - x^i) - ({}^i\hat{\eta}^j - x^j)] + \beta_i [({}^i\hat{\eta}^i - x^i) - ({}^i\hat{\eta}^0 - x^0)] . \quad (3.132)$$

Let $z^i = {}^i\hat{\eta}^i - x^i$, $z^j = {}^i\hat{\eta}^j - x^j$ and $z^0 = {}^i\hat{\eta}^0 - x^0$. Then, it leads to

$$e^i = \sum_{i=1}^N a_{ij} (z^i - z^j) + \beta_i (z^i - z^0) . \quad (3.133)$$

One can express the consensus errors of all agents in the network in the following lumped format

$$e_c = (\mathcal{H} \otimes I_n) z_c - (\mathcal{B} \otimes z^0) \mathbf{1} , \quad (3.134)$$

where $e_c = [e^1, e^2, \dots, e^N]^T$, $z_c = [z^1, z^2, \dots, z^N]^T = \eta - x_c$, ${}^i\hat{\eta} = [{}^i\hat{\eta}^1, {}^i\hat{\eta}^2, \dots, {}^i\hat{\eta}^N] \in \mathbb{R}^{Nn \times 1}$ and $I_n \in \mathbb{R}^{n \times n}$ is an identity matrix. Similarly, lumped dynamics of the multi-agent system by considering the formation parameters can be presented as

$$\dot{z}_c = {}^i\dot{\hat{\eta}} - A_c x_c - (I_N \otimes B) u_c - g_c , \quad (3.135)$$

where ${}^i\dot{\hat{\eta}} \in \mathbb{R}^{Nn \times 1}$ is the rate of changes in the estimated values at agent i for the desired formation topology among the agents in the network. For the leader, we have

$$\dot{z}^0 = \dot{\eta}^0 - u^0 . \quad (3.136)$$

Proposition 3-5. If the consensus errors for all agents converge to zero, i.e.

$$\lim_{t \rightarrow \infty} e_c = \mathbf{0}, \quad (3.137)$$

where $\mathbf{0} = [0, 0, \dots, 0]^T \in \mathbb{R}^{Nn \times 1}$, then the objective of formation-tracking problem would be achieved. Furthermore, one can define $\zeta_c = [\zeta^1, \zeta^2, \dots, \zeta^N]^T$ where $\zeta^i = \int e^i dt$ for $i = [1, N]$.

3.5.2 Stability analysis

Theorem 3-5. For the network defined in *Definition 3-14*, *Definition 3-15* and *Definition 3-16*, if one defines the control inputs for agent i as follows

$$u^i = \frac{1}{\mathcal{H}(i, i)} B^{-1} \left\{ \frac{1}{2} R^i P^i e^i - \left[\sum_{j=1, j \neq i}^N (\mathcal{H}(i, j) B u^j) \right] - \hat{A}^i \left[\left(\sum_{j=1}^N \mathcal{H}(i, j) (2x^j - {}^i \hat{\eta}^j) \right) + \beta_i z^0 - e^i \right] - \left[\sum_{j=1}^N \mathcal{H}(i, j) \hat{g}^j \right] + \left[\sum_{j=1}^N \mathcal{H}(i, j) {}^i \dot{\eta}^j \right] - \beta_i (\dot{\eta}^0 - u^0) + \kappa \zeta^i + (2Q^i - \frac{3}{4} R^i P^i) e^i \right\}, \quad (3.138)$$

where κ is a positive scalar, $R^i \in \mathbb{R}^{n \times n}$ is a positive definite matrix and $P^i \in \mathbb{R}^{n \times n}$ is defined using

$$\dot{P}^i = +2(\hat{A}^i)^T P^i + 2P^i \hat{A}^i - P^i R^i P^i + 2Q^i, \quad (3.139)$$

where $Q^i \in \mathbb{R}^{n \times n}$ is another positive definite matrix; and also uses the following adaptive laws for estimation of unknown linear and nonlinear terms at agent i ,

$$\dot{\hat{g}}^i = -\Gamma_1 \sum_{j=1}^N [\mathcal{H}(j,i)P^j e^j] - \rho_1 \Gamma_1 \hat{g}^i \quad (3.140a)$$

$$v_{\hat{A}^i} = -\Gamma_2 \mathcal{M} \left(\left\{ \sum_{j=1}^N [\mathcal{H}(i,j)(2x^j - {}^i\hat{\eta}^j)] + \beta_i z^0 - e^i \right\} P^i e^i - \rho_2 \Gamma_2 v_{\hat{A}^i} \right), \quad (3.140b)$$

where (similar to *Theorem 3-2*) $\Gamma_1 \in \mathbb{R}^{n \times n}$ and $\Gamma_2 \in \mathbb{R}^{n \times n}$ are two positive definite diagonal matrices acting as the adaptation gains, and ρ_1 and ρ_2 are positive scalars as the leakage rates; then the formation-tracking objective in *Proposition 3-5* would be achieved.

Proof. Let us define

$$\tilde{A}_c = A_c - \hat{A}_c, \quad (3.141)$$

where $\hat{A}_c = \text{diag}(\hat{A}^1, \hat{A}^2, \dots, \hat{A}^N)$. Note that, the diagonal matrix \hat{A}^i is the estimated values of A^i . Then according to *Definition 3-7*, one can have

$$v_{\hat{A}_c} = v_{A_c} - v_{\hat{A}_c}, \quad (3.142)$$

where only the diagonal elements are listed to form a vector equation. Besides, the estimation errors for nonlinear terms at all agents are defined as follows

$$\tilde{g}_c = g_c - \hat{g}_c. \quad (3.143)$$

Considering the estimation errors as well as the formation-tracking error, one can

define the following Lyapunov function

$$V_6 = \frac{1}{2} e_c^T P_c e_c + \frac{1}{2} \kappa \zeta_c^T P_c \zeta_c + \frac{1}{2} \tilde{g}_c^T \Gamma_1^{-1} \tilde{g}_c + \frac{1}{2} v_{\tilde{A}_c}^T \Gamma_2^{-1} v_{\tilde{A}_c}, \quad (3.144)$$

where, $P_c = \text{diag}(P^1, P^2, \dots, P^N)$. Continuing with the derivative of V_6 , it reaches to

$$\dot{V}_6 = e_c^T P_c [(\mathcal{H} \otimes I_n) \dot{z}_c - (\mathcal{B} \otimes \dot{z}^0) \mathbf{1}] + \frac{1}{2} e_c^T \dot{P}_c e_c + \kappa e_c^T P_c \zeta_c + \tilde{g}_c^T \Gamma_1^{-1} \dot{\tilde{g}}_c + v_{\tilde{A}_c}^T \Gamma_2^{-1} v_{\tilde{A}_c}. \quad (3.145)$$

By replacing \dot{z}_c from (3.135) and \dot{z}^0 from (3.136), it leads to

$$\begin{aligned} \dot{V}_6 = & e_c^T P_c \{ (\mathcal{H} \otimes I_n) \dot{\eta} - (\mathcal{H} \otimes I_n) A_c x_c - (\mathcal{H} \otimes I_n) (I_N \otimes B) u_c - (\mathcal{H} \otimes I_n) g_c - \\ & (\mathcal{B} \otimes [\dot{\eta}^0 - u^0]) \mathbf{1} \} + \frac{1}{2} e_c^T \dot{P}_c e_c + \kappa e_c^T P_c \zeta_c + \tilde{g}_c^T \Gamma_1^{-1} \dot{\tilde{g}}_c + v_{\tilde{A}_c}^T \Gamma_2^{-1} v_{\tilde{A}_c}. \end{aligned} \quad (3.146)$$

Moreover, by adding and subtracting

$$c_1 = e_c^T P_c A_c [(\mathcal{H} \otimes I_n) (\eta + 2x_c) + (\mathcal{B} \otimes z^0) \mathbf{1} + e_c] \quad (3.147)$$

from (3.146) and recalling the mixed-product property of the Kronecker product (refer to *Appendix G*), one can have

$$\begin{aligned} \dot{V}_6 = & e_c^T P_c \{ A_c [(\mathcal{H} \otimes I_n) (-\eta + x_c) + (\mathcal{B} \otimes z^0) \mathbf{1}] - A_c e_c - A_c [(\mathcal{H} \otimes I_n) (2x_c - \eta) + \\ & (\mathcal{B} \otimes z^0) \mathbf{1} - e_c] - (\mathcal{H} \otimes B) u_c - (\mathcal{H} \otimes I_n) g_c + (\mathcal{H} \otimes I_n) \dot{\eta} - (\mathcal{B} \otimes [\dot{\eta}^0 - u^0]) \mathbf{1} \} + \\ & \frac{1}{2} e_c^T \dot{P}_c e_c + \kappa e_c^T P_c \zeta_c + \tilde{g}_c^T \Gamma_1^{-1} \dot{\tilde{g}}_c + v_{\tilde{A}_c}^T \Gamma_2^{-1} v_{\tilde{A}_c}. \end{aligned} \quad (3.148)$$

Recalling (3.134), it is followed by

$$\begin{aligned}\dot{V}_6 = & e_c^T P_c \{-2A_c e_c - A_c [(\mathcal{H} \otimes I_n)(2x_c - \eta) + (\mathcal{B} \otimes z^0) \mathbf{1} - e_c] - (\mathcal{H} \otimes B) u_c - \\ & (\mathcal{H} \otimes I_n) g_c + (\mathcal{H} \otimes I_n) \dot{\eta} - (\mathcal{B} \otimes [\dot{\eta}^0 - u^0]) \mathbf{1}\} + \\ & \frac{1}{2} e_c^T \dot{P}_c e_c + \kappa e_c^T P_c \zeta_c + \tilde{g}_c^T \Gamma_1^{-1} \dot{\tilde{g}}_c + v_{\tilde{A}_c}^T \Gamma_2^{-1} v_{\tilde{A}_c}.\end{aligned}\quad (3.149)$$

Then, by adding and subtracting

$$c_2 = e_c^T P_c \{\hat{A}_c [(\mathcal{H} \otimes I_n)(2x_c - \eta) + (\mathcal{B} \otimes z^0) \mathbf{1} - e_c] + (\mathcal{H} \otimes I_n) \hat{g}_c\} \quad (3.150)$$

from (3.149), we have

$$\begin{aligned}\dot{V}_6 = & e_c^T P_c \{-2A_c e_c - \hat{A}_c [(\mathcal{H} \otimes I_n)(2x_c - \eta) + (\mathcal{B} \otimes z^0) \mathbf{1} - e_c] - (\mathcal{H} \otimes B) u_c - \\ & (\mathcal{H} \otimes I_n) \hat{g}_c + (\mathcal{H} \otimes I_n) \dot{\eta} - (\mathcal{B} \otimes [\dot{\eta}^0 - u^0]) \mathbf{1} + \kappa \zeta_c\} + \frac{1}{2} e_c^T \dot{P}_c e_c + c_3 + c_4,\end{aligned}\quad (3.151)$$

where

$$c_3 = \tilde{g}_c^T \Gamma_1^{-1} \dot{\tilde{g}}_c - e_c^T P_c (\mathcal{H} \otimes I_n) \tilde{g}_c \quad (3.152)$$

and

$$c_4 = v_{\tilde{A}_c}^T \Gamma_2^{-1} v_{\tilde{A}_c} - e_c^T P_c \tilde{A}_c [(\mathcal{H} \otimes I_n)(2x_c - \eta) + (\mathcal{B} \otimes z^0) \mathbf{1} - e_c]. \quad (3.153)$$

If it is assumed that

$$c_5 = (\mathcal{H} \otimes I_n)(2x_c - \eta) + (\mathcal{B} \otimes z^0) \mathbf{1} - e_c, \quad (3.154)$$

then it can be shown easily that

$$\tilde{A}_c[(\mathcal{H} \otimes I_n)(2x_c - \eta) + (\mathcal{B} \otimes z^0)\mathbf{1} - e_c] = \mathcal{M}_{c_5}v_{\tilde{A}_c}, \quad (3.155)$$

where \mathcal{M}_{c_5} is defined recalling *Definition 3-8*. Thus, it leads to

$$c_4 = v_{\tilde{A}_c}^T \Gamma_2^{-1} v_{\tilde{A}_c} - e_c^T P_c \mathcal{M}_{c_5} v_{\tilde{A}_c}. \quad (3.156)$$

At this point, c_3 is going to be represented in an appropriate form to derive the adaptive law for online estimation of unknown g . By recalling (3.143) and also adding and subtracting

$$c_6 = \frac{1}{4\rho_1} [\Gamma_1^{-1} \dot{g}_c + \rho_1 g_c]^T [\Gamma_1^{-1} \dot{g}_c + \rho_1 g_c] + \rho_1 \tilde{g}_c^T \tilde{g}_c + \rho_1 \tilde{g}_c^T \hat{g}_c \quad (3.157)$$

from c_3 , it is continued by

$$c_3 = \tilde{g}_c^T [-\Gamma_1^{-1} \dot{\hat{g}}_c - \rho_1 \hat{g}_c - (\mathcal{H}^T \otimes I_n) P_c e_c] - c_7 + c_8, \quad (3.158)$$

where

$$\begin{aligned} c_7 &= \frac{1}{4\rho_1} [\Gamma_1^{-1} \dot{g}_c + \rho_1 g_c]^T [\Gamma_1^{-1} \dot{g}_c + \rho_1 g_c] + \rho_1 \tilde{g}_c^T \tilde{g}_c - 2(\sqrt{\rho_1} \tilde{g}_c^T) \frac{1}{2\sqrt{\rho_1}} [\rho_1 \tilde{g}_c + \rho_1 \hat{g}_c + \Gamma_1^{-1} \dot{g}_c] \\ &= \left\{ \frac{1}{2\sqrt{\rho_1}} [\Gamma_1^{-1} \dot{g}_c + \rho_1 g_c] - \sqrt{\rho_1} \tilde{g}_c \right\}^T \left\{ \frac{1}{2\sqrt{\rho_1}} [\Gamma_1^{-1} \dot{g}_c + \rho_1 g_c] - \sqrt{\rho_1} \tilde{g}_c \right\} > 0 \end{aligned} \quad (3.159)$$

and

$$c_8 = \frac{1}{4\rho_1} [\Gamma_1^{-1} \dot{g}_c + \rho_1 g_c]^T [\Gamma_1^{-1} \dot{g}_c + \rho_1 g_c] > 0. \quad (3.160)$$

Furthermore, by defining the following adaptive law

$$\hat{g}_c = -(\mathcal{H}^T \otimes I_n) \Gamma_1 P_c e_c - \rho_1 \Gamma_1 \hat{g}_c , \quad (3.161)$$

one can have

$$c_3 = -c_7 + c_8 . \quad (3.162)$$

Moreover, since g_c is a vector of unknown bounded continuous Lipschitz functions (referring to *Definition 3-3* and *Definition 3-14*), one can define $|g_c| \leq L_{g_c}$ and $|\dot{g}_c| \leq L_{\dot{g}_c}$. Hence,

$$c_3 \leq -c_7 + M_8 , \quad (3.163)$$

where

$$M_8 = \frac{1}{4\rho_1} [\Gamma_1^{-1} L_{\dot{g}_c} + \rho_1 L_{g_c}]^T [\Gamma_1^{-1} L_{\dot{g}_c} + \rho_1 L_{g_c}] . \quad (3.164)$$

Similar procedure is provided for deriving the adaptive law for estimating $v_{\hat{A}_c}$.

Specifically, by adding and subtracting

$$c_9 = \frac{1}{4\rho_2} [\Gamma_2^{-1} v_{\hat{A}_c} + \rho_2 v_{A_c}]^T [\Gamma_2^{-1} v_{\hat{A}_c} + \rho_2 v_{A_c}] + \rho_2 v_{\hat{A}_c}^T v_{\tilde{A}_c} + \rho_2 v_{\hat{A}_c}^T v_{\hat{A}_c} \quad (3.165)$$

from (3.153), one can reach

$$c_4 = v_{\hat{A}_c}^T [-\Gamma_2^{-1} v_{\hat{A}_c} - \rho_2 v_{\hat{A}_c} - \mathcal{M}_{c_5} P_c e_c] - c_{10} + c_{11} . \quad (3.166)$$

Here, the following definitions have been used

$$\begin{aligned}
c_{10} &= \frac{1}{4\rho_2} [\Gamma_2^{-1} v_{\dot{A}_c} + \rho_2 v_{A_c}]^T [\Gamma_2^{-1} v_{\dot{A}_c} + \rho_2 v_{A_c}] + \rho_2 v_{\dot{A}_c}^T v_{\dot{A}_c} - 2(\sqrt{\rho_2} v_{\tilde{A}_c}^T) \frac{1}{2\sqrt{\rho_2}} [\rho_2 v_{\dot{A}_c} + \\
&\quad \rho_2 v_{\tilde{A}_c} + \Gamma_2^{-1} v_{\dot{A}_c}] \\
&= \left\{ \frac{1}{2\sqrt{\rho_2}} [\Gamma_2^{-1} v_{\dot{A}_c} + \rho_2 v_{A_c}] - \sqrt{\rho_2} v_{\tilde{A}_c} \right\}^T \left\{ \frac{1}{2\sqrt{\rho_2}} [\Gamma_2^{-1} v_{\dot{A}_c} + \rho_2 v_{A_c}] - \sqrt{\rho_2} v_{\tilde{A}_c} \right\} > 0
\end{aligned} \tag{3.167}$$

and

$$c_{11} = \frac{1}{4\rho_2} [\Gamma_2^{-1} v_{\dot{A}_c} + \rho_2 v_{A_c}]^T [\Gamma_2^{-1} v_{\dot{A}_c} + \rho_2 v_{A_c}] > 0. \tag{3.168}$$

In addition, by suggesting the following adaptive law

$$v_{\dot{A}_c} = -\mathcal{M}_{c_5} \Gamma_2 P_c e_c - \rho_2 \Gamma_2 v_{\dot{A}_c}, \tag{3.169}$$

it leads to

$$c_4 = -c_{10} + c_{11}. \tag{3.170}$$

Similarly, by defining $|A_c| \leq L_{A_c} \in \mathbb{R}^{Nn \times Nn}$ and $|\dot{A}_c| \leq L_{\dot{A}_c} \in \mathbb{R}^{Nn \times Nn}$ (referring to *Definition 3-3* and *Definition 3-14*), one can have

$$c_4 \leq -c_{10} + M_{11} \tag{3.171}$$

where

$$M_{11} = \frac{1}{4\rho_2} [\Gamma_2^{-1} v_{L_{\dot{A}_c}} + \rho_2 v_{L_{A_c}}]^T [\Gamma_2^{-1} v_{L_{\dot{A}_c}} + \rho_2 v_{L_{A_c}}] > 0. \tag{3.172}$$

Furthermore, by replacing (3.163) and (3.171) in (3.151), the following inequality is

achieved

$$\begin{aligned} \dot{V}_6 \leq & e_c^T P_c \{ -2A_c e_c - \hat{A}_c [(\mathcal{H} \otimes I_n)(2x_c - \eta) + (\mathcal{B} \otimes z^0) \mathbf{1} - e_c] - (\mathcal{H} \otimes B) u_c - \\ & (\mathcal{H} \otimes I_n) \hat{g}_c + (\mathcal{H} \otimes I_n) \dot{\eta} - (\mathcal{B} \otimes [\dot{\eta}^0 - u^0]) \mathbf{1} + \kappa \zeta_c \} + \frac{1}{2} e_c^T \dot{P}_c e_c - c_0 + M_0 , \end{aligned} \quad (3.173)$$

where

$$c_0 = c_7 + c_{10} , \quad (3.174a)$$

$$M_0 = \min(M_8, M_{11}) . \quad (3.174b)$$

Thus, by setting

$$\begin{aligned} (\mathcal{H} \otimes B) u_c = & +\frac{1}{2} R P_c e_c - \hat{A}_c [(\mathcal{H} \otimes I_n)(2x_c - \eta) + (\mathcal{B} \otimes z^0) \mathbf{1} - e_c] - (\mathcal{H} \otimes I_n) \hat{g}_c + \\ & (\mathcal{H} \otimes I_n) \dot{\eta} - (\mathcal{B} \otimes [\dot{\eta}^0 - u^0]) \mathbf{1} + \kappa \zeta_c + [2Q_c - \frac{3}{4} R_c P_c] e_c , \end{aligned} \quad (3.175)$$

where $R_c = \text{diag}(R^1, R^2, \dots, R^N) = R^T > 0 \in \mathbb{R}^{Nn \times Nn}$ and $Q_c = \text{diag}(Q^1, Q^2, \dots, Q^N) = Q^T > 0 \in \mathbb{R}^{Nn \times Nn}$, and also by recalling that A_c is a diagonal matrix and consequently $A_c^T = A_c$; one can reach

$$\dot{V}_6 \leq -e_c^T [A_c^T P_c + P_c A_c + \frac{1}{2} P_c R_c P_c + 2Q_c - \frac{3}{4} P_c R_c P_c - \frac{1}{2} \dot{P}_c] e_c - c_0 + M_0 . \quad (3.176)$$

Moreover, by utilizing the following DRE

$$\dot{P}_c = +2A_c^T P_c + 2P_c A_c - P_c R_c P_c + 2Q_c , \quad (3.177)$$

we would have

$$\dot{V}_6 \leq -[e_c^T (\frac{1}{4} P_c R_c P_c + Q_c) e_c + c_0] + M_0 . \quad (3.178)$$

Thus, according to LaSalle-Yoshizawa theorem, V_6 is UUB. Therefore, e_c , \tilde{g}_c and $v_{\tilde{A}_c}$ converges to small bounded sets around zero, according to the definition of V_6 proposed in (3.144). Moreover by recalling *Lemma 3-1*, one can replace A_c in (3.177) with \hat{A}_c and consequently it leads to

$$\dot{P}_c = +2\hat{A}_c^T P_c + 2P_c \hat{A}_c - P_c R_c P_c + 2Q_c . \quad (3.179)$$

The only thing remained for finalizing the proof is expression of the control policies and also the adaptive laws locally for each agent in the network. The i th row (corresponding to the i th agent) in (3.171) can be presented as follows

$$\begin{aligned} \sum_{j=1}^N (\mathcal{H}(i, j) B u^j) &= \frac{1}{2} R^i P^i e^i - \hat{A}^i \left[\left(\sum_{j=1}^N \mathcal{H}(i, j) (2x^j - \eta^j) \right) + \beta_i z^0 - e^i \right] - \\ &\quad \left[\sum_{j=1}^N \mathcal{H}(i, j) \hat{g}^j \right] + \left[\sum_{j=1}^N \mathcal{H}(i, j) \dot{\eta}^j \right] - \beta_i (\dot{\eta}^0 - u^0) + \kappa \zeta^i + [2Q^i - \frac{3}{4} R^i P^i] e^i . \end{aligned} \quad (3.180)$$

By rearranging (3.180) and also replacing η^j by the corresponding estimated values at agent i (i.e. ${}^i\eta^j$ per *Proposition 3-4*), one can have

$$\begin{aligned} \mathcal{H}(i, i) B u^i &= - \left[\sum_{j=1, j \neq i}^N (\mathcal{H}(i, j) B u^j) \right] + \frac{1}{2} R^i P^i e^i - \hat{A}^i \left[\left(\sum_{j=1}^N \mathcal{H}(i, j) (2x^j - {}^i\eta^j) \right) + \right. \\ &\quad \left. \beta_i z^0 - e^i \right] - \left[\sum_{j=1}^N \mathcal{H}(i, j) \hat{g}^j \right] + \left[\sum_{j=1}^N \mathcal{H}(i, j) {}^i\dot{\eta}^j \right] - \beta_i (\dot{\eta}^0 - u^0) + \\ &\quad \kappa \zeta^i + [2Q^i - \frac{3}{4} R^i P^i] e^i . \end{aligned} \quad (3.181)$$

Finally, by reminding that B is a square full-rank matrix (according to *Definition 3-6*) and $\mathcal{H}(i, i) \neq 0$ (referring to *Definition 3-10*), it reaches (3.138). In addition, by considering (3.161), (3.169) and (3.154) and also *Proposition 3-4*, one can express the

adaptive laws for agent i as presented in (3.140a) and (3.140b). Besides, the controller gain matrix at agent i is updated utilizing the equation proposed in (3.139), according to (3.179). Then, the proof is completed.

■

Remark 3-11. According to (3.164) and (3.172), the parameters M_8 and M_{11} can be represented as follows

$$M_8 = \frac{1}{4\rho_1} L_{\dot{g}_c}^T \Gamma_1^{-2} L_{\dot{g}_c} + \frac{1}{2} L_{\dot{g}_c}^T \Gamma_1^{-1} L_{g_c} + \frac{1}{4} \rho_1 L_{g_c}^T L_{g_c} \quad (3.182)$$

and

$$M_{11} = \frac{1}{4\rho_2} L_{\dot{A}_c}^T \Gamma_2^{-2} L_{\dot{A}_c} + \frac{1}{2} L_{\dot{A}_c}^T \Gamma_2^{-1} L_{A_c} + \frac{1}{4} \rho_2 L_{A_c}^T L_{A_c}. \quad (3.183)$$

As can be seen, by choosing some relatively large values for Γ_1 and Γ_2 and also some relatively small values for ρ_1 and ρ_2 , the values of M_8 and M_{11} and consequently M_0 (according to (3.174b)) converge to zero and more appropriate performance of the adaptive laws and consequently more accurate formation-tracking performance in the network would be achieved.

Remark 3-12. Referring to *Theorem 3-5*, the estimation errors \tilde{g}_c and $v_{\tilde{A}_c}$ converge to small bounded sets around zero and the radius of these sets can be adjusted by M_0 , which in turn converges to zero by choosing some especial values for Γ_1 , Γ_2 , ρ_1 and ρ_2 (as declared in *Remark 3-11*). As a result and also by recalling the *separation principle* presented in *Lemma 3-1*, one can use \hat{g}^i instead of g^i and \hat{A}^i instead of A in (3.127) and (3.96a), in order to represent the nonlinear dynamics of agent i . Moreover, the estimated values for η^i can be incorporated by recalling *Proposition 3-4*. Hence,

we lead to

$$\dot{z}^i = {}^i\dot{\hat{\eta}}^i - \hat{A}^i x^i - B u^i - \hat{g}^i, \quad (3.184)$$

which is followed by a formulation for the dynamics of whole network as

$$\dot{z}_c = {}^i\dot{\hat{\eta}} - \hat{A}_c x_c - (I_N \otimes B) u_c - \hat{g}_c. \quad (3.185)$$

Note that the estimated ${}^i\dot{\hat{\eta}}$ at all agents are equal, regarding the cooperative observer suggested in *Proposition 3-4*.

Remark 3-13. Looking at (3.138), the controller inputs at neighboring agents u^j are required in order to compute the control inputs at agent i (i.e. u^i). But, the controller inputs at the neighboring agents are also being computed at the same time and that computation also needs u^j for its own neighboring agents. As can be seen there is a loop between the controller inputs of the all agents in the network. To solve this issue, the following lemma is proposed.

Lemma 3-7. For a multi-agent dynamic system consisting of N agents defined in *Definition 3-14* and *Definition 3-15*, if the formation-tracking objective in *Proposition 3-5* is satisfied by utilizing *Theorem 3-5*, then one would have the following approximation for expressing the relation between the control inputs at agent i and the control inputs of the virtual leader

$$u^i \simeq B^{-1}[u^0 - \hat{A}^i x^i - \hat{g}^i + {}^i\dot{\hat{\eta}}^i - {}^i\dot{\hat{\eta}}^0]. \quad (3.186)$$

Proof. By incorporating *Proposition 3-4* in (3.136), it leads to

$$\dot{z}^0 = {}^i\hat{\eta}^0, \quad (3.187)$$

where ${}^i\hat{\eta}^0 \in \mathbb{R}^{n \times 1}$ is the estimated formation parameters of the virtual leader at agent i . By subtracting both sides of (3.187) from (3.184), one can have

$$\dot{z}^i - \dot{z}^0 = {}^i\hat{\eta}^i - {}^i\hat{\eta}^0 - \hat{A}^i x^i - B u^i + u^0 - \hat{g}^i. \quad (3.188)$$

Besides, by reaching consensus in formation-tracking problem (i.e. (3.129)) using the distributed control protocol suggested in *Theorem 3-5* (i.e. (3.138)), one can state that

$$\lim_{t \rightarrow \infty} (z^i - z^0) = 0. \quad (3.189)$$

Using the time derivation of both sides of (3.189), it reaches

$$\lim_{t \rightarrow \infty} (\dot{z}^i - \dot{z}^0) = 0. \quad (3.190)$$

Moreover, by replacing (3.188) in (3.190), one leads to

$$\lim_{t \rightarrow \infty} ({}^i\hat{\eta}^i - {}^i\hat{\eta}^0 - \hat{A}^i x^i - B u^i + u^0 - \hat{g}^i) = 0. \quad (3.191)$$

Recalling *Theorem 3-5*, one can use the following approximation

$${}^i\hat{\eta}^i - {}^i\hat{\eta}^0 - \hat{A}^i x^i - B u^i + u^0 - \hat{g}^i \simeq 0. \quad (3.192)$$

Finally, the approximated values for controller inputs of agent i can be expressed as suggested in (3.186). Then, the proof is completed.

■

Proposition 3-6. Since the control inputs of the virtual leader u^0 are not available at all of the agents in the network (by recalling *Assumption 3-1*), a cooperative observer similar to the one designed in *Theorem 3-4* can be used for reconstructing u^0 at agent i , as follows

$$\dot{\hat{q}}^i = -\lambda_1 \tau^i - [\mathcal{M}(\operatorname{sgn}\{\sum_{j=1}^N (\mathcal{H}(i, j) \tau^j)\}) U^M] \quad (3.193)$$

where $\hat{q}^i \in \mathbb{R}^{n \times 1}$ includes the estimated values for the controller inputs at the i th agent, $|u^0| \leq U^M \in \mathbb{R}^{n \times 1}$, $\lambda_1 \in \mathbb{R}^+$, and

$$\tau^i = \sum_{j=1}^N a_{ij}(\hat{q}^i - \hat{q}^j) + \beta_i(\hat{q}^i - u^0) . \quad (3.194)$$

Proposition 3-7. According to *Lemma 3-7* and *Proposition 3-6*, the control inputs for the neighboring agent j to agent i are computed as follows

$$\hat{u}^j = B^{-1}[\hat{q}^j - \hat{A}^j x^j - \hat{g}^j + {}^j \dot{\hat{\eta}}^j - {}^j \dot{\hat{\eta}}^0] . \quad (3.195)$$

Thus, the cooperative control protocol in (3.138) is modified as follows

$$\begin{aligned}
u^i \simeq & \frac{1}{\mathcal{H}(i,i)} B^{-1} \left\{ \frac{1}{2} R^i P^i e^i - \left[\sum_{j=1, j \neq i}^N (\mathcal{H}(i,j) (\hat{q}^j - \hat{A}^j x^j - \hat{g}^j + {}^j \dot{\hat{\eta}}^j - {}^j \dot{\hat{\eta}}^0)) \right] - \right. \\
& \hat{A}^i \left[\left(\sum_{j=1}^N \mathcal{H}(i,j) (2x^j - {}^i \hat{\eta}^j) \right) + \beta_i z^0 - e^i \right] - \left[\sum_{j=1}^N \mathcal{H}(i,j) \hat{g}^j \right] + \left[\sum_{j=1}^N \mathcal{H}(i,j) {}^i \dot{\hat{\eta}}^j \right] - \\
& \left. \beta_i (\dot{\hat{\eta}}^0 - u^0) + \kappa \zeta^i + (2Q^i - \frac{3}{4} R^i P^i) e^i \right\}. \quad (3.196)
\end{aligned}$$

Furthermore, consensus can be achieved over the estimated values of formation variables (i.e. ${}^i \hat{\eta}^j$) and the estimated values of the leader control inputs (i.e. \hat{q}^i) at all agents in the network, according to *Proposition 3-4* and *Proposition 3-6*. Hence for any $k \in \{0, 1, 2, \dots, N\}$ and any $i, j \in \{1, 2, \dots, N\}$, one can have

$${}^j \hat{\eta}^k \simeq {}^i \hat{\eta}^k, \quad (3.197a)$$

$$\hat{q}^j \simeq \hat{q}^i, \quad (3.197b)$$

in finite time. In this regard and by canceling the similar variables in (3.196), one would lead to

$$\begin{aligned}
u^i \simeq & \frac{1}{\mathcal{H}(i,i)} B^{-1} \left\{ \frac{1}{2} R^i P^i e^i - \hat{g}^i + {}^i \dot{\hat{\eta}}^i - \left[\sum_{j=1, j \neq i}^N \mathcal{H}(i,j) \right] (\hat{q}^i - {}^i \dot{\hat{\eta}}^0) - \right. \\
& \left[\sum_{j=1, j \neq i}^N (\mathcal{H}(i,j) \hat{A}^j (x^j - {}^i \hat{\eta}^j)) \right] - \hat{A}^i \left[(\mathcal{H}(i,i) (2x^i - {}^i \hat{\eta}^i)) + \beta_i z^0 - e^i \right] - \\
& \left. \beta_i (\dot{\hat{\eta}}^0 - u^0) + \kappa \zeta^i + (2Q^i - \frac{3}{4} R^i P^i) e^i \right\}. \quad (3.198)
\end{aligned}$$

In addition, according to *Theorem 3-5*, the formation-tracking consensus error at the i th agent (i.e. e^i) defined in (3.133) has been reached to zero. Consequently, one can have $z^j \simeq z^0$ for all $j \in \{1, 2, \dots, N\}$. Hence by regarding the definition of z^j , we lead

to

$$({}^i\hat{\eta}^j - x^j) \simeq ({}^i\hat{\eta}^0 - x^0). \quad (3.199)$$

By replacing (3.199) in (3.198) for all of the agents, one would have

$$\begin{aligned} u^i \simeq & \frac{1}{\mathcal{H}(i,i)} B^{-1} \left\{ \frac{1}{2} R^i P^i e^i - \hat{g}^i + {}^i\dot{\hat{\eta}}^i - \left[\sum_{j=1 \neq i}^N \mathcal{H}(i,j) \right] (\hat{q}^i - {}^i\hat{\eta}^0) - \right. \\ & \left[\sum_{j=1 \neq i}^N (\mathcal{H}(i,j) \hat{A}^j (x^0 - {}^i\hat{\eta}^0)) \right] - \hat{A}^i [(\mathcal{H}(i,i) (2x^0 - 2{}^i\hat{\eta}^0 + {}^i\hat{\eta}^i)) + \beta_i z^0 - e^i] - \\ & \left. \beta_i (\dot{\eta}^0 - u^0) + \kappa \zeta^i + (2Q^i - \frac{3}{4} R^i P^i) e^i \right\}, \quad (3.200) \end{aligned}$$

where the value for x^0 can be estimated by utilizing the cooperative observer suggested in *Theorem 3-4*, as only a portion of the agents are connected to the leader and the exact value of x^0 is not available at all agents in the network. Moreover, as the networks in this thesis are assumed to be homogeneous (refer to *Definition 3-10*), the estimated values for the unknown linear terms at all of the agents (i.e. \hat{A}^i) would converge to similar values (according to the design procedure presented in the proof of *Theorem 3-5*). In this regard, one can replace \hat{A}^j in (3.200) with \hat{A}^i , and achieve the following expression for the distributed control policy

$$\begin{aligned} u^i \simeq & \frac{1}{\mathcal{H}(i,i)} B^{-1} \left\{ \frac{1}{2} R^i P^i e^i - \hat{g}^i + {}^i\dot{\hat{\eta}}^i - \left[\sum_{j=1 \neq i}^N \mathcal{H}(i,j) \right] [\hat{q}^i - {}^i\hat{\eta}^0 - (\hat{A}^i (\hat{x}_i^0 - {}^i\hat{\eta}^0))] - \right. \\ & \left. \hat{A}^i [(\mathcal{H}(i,i) (2\hat{x}_i^0 - 2{}^i\hat{\eta}^0 + {}^i\hat{\eta}^i)) + \beta_i z^0 - e^i] - \beta_i (\dot{\eta}^0 - u^0) + \kappa \zeta^i + (2Q^i - \frac{3}{4} R^i P^i) e^i \right\}. \quad (3.201) \end{aligned}$$

This is an appropriate expression for the control policy in CAMFC-2 algorithm, as it does not rely on any absolute measurement of the local states at each agent in the network. Instead, the algorithm needs inter-agent relative measurements in order to

construct the values of the formation-tracking consensus errors (i.e. e^i) locally at each agent (refer to the definition in (3.131)). These relative measurements should be provided at agent i for each of the neighboring agents j , which has the communication link to the agent i . The values for P^i are updated online utilizing (3.139) at each agent, without any need for data from other agents in the network. Besides, the values for ${}^i\dot{\eta}^i$, ${}^i\dot{\eta}^0$, \hat{q}^i and \hat{x}_i^0 are all estimated using the distributed cooperative observers with the corresponding data transfer, locally at agent i in the network.

Proposition 3-8. According to the distributed adaptive law presented in (3.140a), the values for \hat{g}^i can be estimated locally at agent i , while the values for P^j and \hat{g}^j are being transferred from the neighboring agents j . Also, the value of consensus error e^i is constructed by having the inter-agent relative measurements with the neighboring agents. No measurements for local states are required in this adaptive law. On the other hand, the measurements of local states should be available for operation of the distributed adaptive law for estimation of \hat{A}^i in (3.140b). In order to eliminate this requirement, one can use the approximation in (3.199) and consequently lead to

$$v_{\hat{A}^i} \simeq -\Gamma_2 \mathcal{M} \left(\left\{ \sum_{j=1}^N [\mathcal{H}(i, j)(2\hat{x}_i^0 - 2{}^i\dot{\eta}^0 + {}^i\dot{\eta}^j)] + \beta_i z^0 - e^i \right\} \right) P^i e^i - \rho_2 \Gamma_2 v_{\hat{A}^i}. \quad (3.202)$$

Here, the cooperative observer suggested in *Theorem 3-4* are applied to provide the estimated values of \hat{x}_i^0 at the agents without direct connection to the leader. Utilizing the above adaptive law, the values of \hat{A}^i can be updated locally at agent i without any measurements of the local states. Instead, the values for \hat{x}_i^0 , ${}^i\dot{\eta}^0$ and ${}^i\dot{\eta}^j$ should be provided using the proposed cooperative observers in *Theorem 3-4* and *Proposition 3-4*. Corresponding data must be transferred from the neighboring agents. Again, here

the inter-agent relative measurements are required to construct the consensus error e^i .

3.5.3 Optimality analysis

Definition 3-17. For the network defined in *Definition 3-14* and *Definition 3-15*, one can define the following cost-to-go function for measuring the performance of a designed set of distributed control inputs u_c regarding the formation-tracking objective presented in *Proposition 3-5*,

$$J_3(e_c, u_c) = \int_t^\infty L_3(e_c(\tau_t), u_c(\tau_t)) d\tau_t , \quad (3.203)$$

where $J_3(\cdot)$ is a scalar cost defined as follows

$$J_3 = \frac{1}{2} e_c^T P_c e_c + \frac{1}{2} \kappa \zeta_c^T P_c e_c , \quad (3.204)$$

and $L_3(\cdot)$ is the scalar value for systems utility suggested as

$$L_3 = e_c^T Q_c e_c + (u_c^1)^T (\mathcal{H}^T \otimes B^T) R_c^{-1} (\mathcal{H} \otimes B) u_c^1 , \quad (3.205)$$

where

$$(\mathcal{H} \otimes B) u_c^1 = +\frac{1}{2} R_c P_c e_c . \quad (3.206)$$

Proposition 3-9. Consider the cost function proposed in (3.203) for the network defined in *Definition 3-14*. The objective is to define the optimal time sequence of control inputs $u_c^{op} = \{u_c^{op}(t), \dots, u_c^{op}(\infty)\}$ for $u_c^{op}(t) = [u_{op}^1(t), u_{op}^2(t), \dots, u_{op}^N(t)]^T$ in

order to minimize J_3 , i.e.

$$\min_{u_c[t,\infty)} J_3(e_c(t), u_c(t)) = \min_{u_c[t,\infty)} \int_t^\infty L_3(e_c(\tau_t), u_c(\tau_t)) d\tau_t . \quad (3.207)$$

The solution for this problem is proposed as the solution of an HJB equation as follows

$$0 = \min_{u_c=u_c^{op}(t)} \left\{ L_3(e_c(t), u_c(t)) + \frac{dJ_3(e_c(t), u_c(t))}{dt} \right\} . \quad (3.208)$$

Theorem 3-6. For the formation-tracking problem defined in *Proposition 3-5*, the distributed cooperative controller proposed in (3.196) includes an optimal term.

Proof. Let us define the following *Hamiltonian*

$$\begin{aligned} H_7 = L_3 + \frac{dJ_3}{dt} &= e_c^T P_c \dot{e}_c + \frac{1}{2} e_c^T \dot{P}_c e_c + \kappa e_c^T P_c \zeta_c + e_c^T Q_c e_c + \\ &\quad (u_c^1)^T (\mathcal{H}^T \otimes B^T) R_c^{-1} (\mathcal{H} \otimes B) u_c^1 . \end{aligned} \quad (3.209)$$

By substituting \dot{e}_c from (3.134), one can have

$$\begin{aligned} H_7 = e_c^T P_c [(\mathcal{H} \otimes I_n) \dot{\zeta}_c - (\mathcal{B} \otimes \dot{z}^0) \mathbf{1}] + \frac{1}{2} e_c^T \dot{P}_c e_c + \kappa e_c^T P_c \zeta_c + e_c^T Q_c e_c + \\ (u_c^1)^T (\mathcal{H}^T \otimes B^T) R_c^{-1} (\mathcal{H} \otimes B) u_c^1 . \end{aligned} \quad (3.210)$$

Utilizing (3.185), it leads to

$$H_7 = e_c^T P_c [(\mathcal{H} \otimes I_n) \dot{\hat{\eta}} - (\mathcal{H} \otimes I_n) \hat{A}_c x_c - (\mathcal{H} \otimes I_n) (I_N \otimes B) u_c - (\mathcal{H} \otimes I_n) \hat{g}_c - (\mathcal{B} \otimes \dot{z}^0) \mathbf{1}] + \frac{1}{2} e_c^T \dot{P}_c e_c + \kappa e_c^T P_c \zeta_c + e_c^T Q_c e_c + (u_c^1)^T (\mathcal{H}^T \otimes B^T) R_c^{-1} (\mathcal{H} \otimes B) u_c^1. \quad (3.211)$$

Moreover, by recalling the mixed-product property for Kronecker product and replacing \dot{z}^0 from (3.136), one can have

$$H_7 = e_c^T P_c [(\mathcal{H} \otimes I_n) \dot{\hat{\eta}} - (\mathcal{H} \otimes I_n) \hat{A}_c x_c - (\mathcal{H} \otimes B) u_c - (\mathcal{H} \otimes I_n) \hat{g}_c - (\mathcal{B} \otimes [\dot{\eta}^0 - u^0]) \mathbf{1}] + \frac{1}{2} e_c^T \dot{P}_c e_c + \kappa e_c^T P_c \zeta_c + e_c^T Q_c e_c + (u_c^1)^T (\mathcal{H}^T \otimes B^T) R_c^{-1} (\mathcal{H} \otimes B) u_c^1. \quad (3.212)$$

By recalling (3.206) and assuming

$$(\mathcal{H} \otimes B) u_c^2 = -\hat{A}_c [(\mathcal{H} \otimes I_n) (2x_c - \hat{\eta}) + (\mathcal{B} \otimes z^0) \mathbf{1} - e_c] - (\mathcal{H} \otimes I_n) \hat{g}_c + (\mathcal{H} \otimes I_n) \dot{\hat{\eta}}_c - (\mathcal{B} \otimes [\dot{\eta}^0 - u^0]) \mathbf{1} + \kappa \zeta_c + [2Q_c - \frac{3}{4} R_c P_c] e_c, \quad (3.213)$$

one can replace $u_c = u_c^1 + u_c^2$ in (3.212) and have

$$H_7 = e_c^T P_c \{ \hat{A}_c [(\mathcal{H} \otimes I_n) (-\eta + x_c) + (\mathcal{B} \otimes z^0) \mathbf{1}] - \hat{A}_c e_c - \frac{1}{2} R_c P_c e_c - 2Q_c e_c + \frac{3}{4} R_c P_c e_c \} + \frac{1}{2} e_c^T \dot{P}_c e_c + e_c^T Q_c e_c + \frac{1}{4} e_c^T P_c R_c P_c e_c. \quad (3.214)$$

Furthermore by recalling $\hat{A}_c^T = \hat{A}_c$, it leads to

$$H_7 = e_c^T [-\hat{A}_c^T P_c - P_c \hat{A}_c + \frac{1}{2} P_c R_c P_c - Q_c + \frac{1}{2} \dot{P}_c] e_c, \quad (3.215)$$

which is equal to zero by utilizing (3.179). Besides, let's differentiate H_7 in (3.212) with respect to u_c^1 ,

$$[\frac{\partial H_7}{\partial u_c^1}]^T = -e_c^T P_c (\mathcal{H} \otimes B) + 2(u_c^1)^T (\mathcal{H}^T \otimes B^T) R_c^{-1} (\mathcal{H} \otimes B) . \quad (3.216)$$

This equation is equal to zero by substituting u_c^1 from (3.206). It means that a part of u_c is designed in such a way that makes the first derivative of H_7 to be zero. Consequently, the optimality condition as defined in *Proposition 3-9* is satisfied and the proof is completed. ■

3.5.4 CAMFC-2 algorithm

The proposed cooperative AMFC algorithm in this section, which is a solution to consensus and formation-tracking problems in completely unknown multi-agent nonlinear dynamic systems utilizing the inter-agent relative measurements among the neighboring agents in the network, is presented in Table 3.3 and Fig. 3.4. This decentralized algorithm is named as CAMFC-2. Here, the values for several variables as shown in Fig. 3.4 should be transmitted from agent i to its neighboring agents via the existing communication links.

Table 3.3: CAMFC-2 algorithm

Algorithm 3: CAMFC-2 (at agent i in the network)

Initialization:

$\lambda = \lambda_1 = \mu$ are set at a large value , $X_M = \Upsilon^M = U^M$ are set at a small value

$k_1 = k_2 = 1$, $\kappa = 10$, $B = I_n$, $R^i = I_n$, $Q^i = 10 \times I_n$

$P^i(0) = I_n$, $\hat{A}^i(0) = \mathbf{0}_n$, $\hat{g}^i(0) = \mathbf{0}_n$, $\zeta^i(0) = \mathbf{0}_n$

$\{\Gamma_1, \Gamma_2, \rho_1, \rho_2\}$ are set according to *Remark 3-4* .

Note-1: t_f is the simulation/implementation duration time, and dt is the sampling time .

Main Loop:

for $\{t = d_t : d_t : t_f\}$ **do**

Inputs: $\{x^0(t), u^0(t), \eta^0(t), (x^i(t) - x^j(t)), \hat{x}_i^0(t - d_t), \hat{q}^i(t - d_t), \hat{\Omega}^i(t - d_t), e^j(t - d_t), P^j(t - d_t), \hat{x}_j^0(t - d_t), \hat{q}^j(t - d_t), \hat{\Omega}^j(t - d_t), \hat{u}^j(t - d_t), \hat{g}^j(t - d_t), (j \neq i \in [1, N])\}$

for $\{j = 1 : N\}$ **do**

1: ${}^i\tilde{\eta}^j(t) = \beta_i [{}^i\hat{\eta}^j(t - d_t) - \eta^j]$

for $\{k = 1 : N\}$ **do**

2: ${}^i\tilde{\eta}^j(t) = {}^i\tilde{\eta}^j(t) + a(i, k) [{}^i\hat{\eta}^j(t - d_t) - {}^k\hat{\eta}^j(t - d_t)]$

end for

3: ${}^i\dot{\tilde{\eta}}^j(t) = -\mu {}^i\tilde{\eta}^j(t)$

for $\{k = 1 : N\}$ **do**

4: ${}^i\dot{\tilde{\eta}}^j(t) = {}^i\dot{\tilde{\eta}}^j(t) - \mathcal{M}(sgn\{\mathcal{H}(i, k)^k \tilde{\eta}^j(t - d_t)\}) \times \Upsilon^M$

end for

5: ${}^i\hat{\eta}^j(t) = {}^i\tilde{\eta}^j(t - d_t) + \{{}^i\dot{\tilde{\eta}}^j(t) \times d_t\}$

end for

6: $\tilde{x}_i^0(t) = \beta_i [\hat{x}_i^0(t - d_t) - x^0(t)]$

7: $\tau^i(t) = \beta_i [\hat{q}^i(t - d_t) - u_0(t)]$

8: $e^i(t) = \beta_i [({}^i\hat{\eta}^i(t) - {}^i\hat{\eta}^0(t)) - (x^i(t) - x^0(t))]$

9: $\dot{\hat{g}}_1^i(t) = 0$, $S_H = 0$

10: $z^0(t) = \eta^0 - x^0$

(the algorithm is continued in Table 3.4)

Table 3.4: CAMFC-2 algorithm (continued)

Algorithm 3: CAMFC-2 (at agent i in the network)

(the algorithm in Table 3.3 is continued here)

```

for  $\{j = 1 : N\}$  do
11-1:  $\hat{x}_i^0(t) = \tilde{x}_i^0(t) + a(i, j) [\hat{x}_i^0(t - d_t) - \hat{x}_j^0(t - d_t)]$ 
11-2:  $\tau^i(t) = \tau^i(t) + a(i, j) [\hat{q}^i(t - d_t) - \hat{q}^j(t - d_t)]$ 
11-3:  $e^i(t) = e^i(t) + a(i, j) [{}^i\hat{\eta}^i(t) - {}^i\hat{\eta}^j(t) - (x^i(t) - x^j(t))]$ 
11-4:  $\dot{g}_1^i(t) = \dot{g}_1^i(t) + \{\mathcal{H}(i, j)P^j(t - d_t)e^j(t - d_t)\}$ 
      if  $\{i \neq j\}$ 
11-5:  $S_H = S_H + \mathcal{H}(i, j)$ 
      end if
end for
12:  $\dot{\hat{x}}_i^0(t) = -\lambda \hat{x}_i^0(t)$ 
13:  $\dot{\hat{q}}^i(t) = -\lambda_1 \tau^i(t)$ 
14:  $v_{\hat{A}^i}^1(t) = \beta_i z^0(t) - e^i(t)$ 
      for  $\{j = 1 : N\}$  do
15-1:  $\hat{x}_i^0(t) = \hat{x}_i^0(t) - \{\mathcal{M}(sgn\{\mathcal{H}(i, j)\tilde{x}_0^j(t - d_t)\}) \times X_M\}$ 
15-2:  $\dot{\hat{q}}^i(t) = \dot{\hat{q}}^i(t) - \{\mathcal{M}(sgn\{\mathcal{H}(i, j)\tau^j(t - d_t)\}) \times U^M\}$ 
15-3:  $v_{\hat{A}^i}^1(t) = v_{\hat{A}^i}^1(t) + \{\mathcal{H}(i, j)(2\hat{x}_i^0 - 2^i\hat{\eta}^0 + {}^i\hat{\eta}^j)\}$ 
      end for
16:  $\hat{x}_i^0(t) = \hat{x}_i^0(t - d_t) + \{\hat{x}_i^0(t) \times d_t\}$ 
17:  $\hat{q}^i(t) = \hat{q}^i(t - d_t) + \{\hat{q}^i(t) \times d_t\}$ 
18:  $\zeta^i(t) = \zeta^i(t - d_t) + \{e^i(t) \times d_t\}$ 
19:  $P^i(t) = P^i(t - d_t) + \{[2(\hat{A}^i)^T(t - d_t) P^i(t - d_t) + 2P^i(t - d_t) \hat{A}^i(t - d_t) - P^i(t - d_t) BR^i B^T P^i(t - d_t) + 2Q^i] \times d_t\}$ 
20:  $\hat{g}^i(t) = \hat{g}^i(t - d_t) + \{[-\Gamma_1 \dot{\hat{g}}_1^i(t) - \rho_1 \Gamma_1 g^i(t - d_t)] \times d_t\}$ 
21:  $v_{\hat{A}^i}^1(t) = v_{\hat{A}^i}^1(t - d_t) + \{[-\Gamma_2 \mathcal{M}(v_{\hat{A}^i}^1(t)) P^i(t) e^i(t) - \rho_2 \Gamma_2 v_{\hat{A}^i}^1(t - d_t)] \times d_t\}$ 
22:  $u^i(t) = \frac{1}{\mathcal{H}(i, i)} B^{-1} \left\{ \frac{1}{2} R^i P^i(t) e^i(t) - S_H [\hat{q}^i(t) - {}^i\dot{\hat{\eta}}^0(t) - \hat{A}^i(t)(\hat{x}_i^0(t) - {}^i\hat{\eta}^0(t))] - \right.$ 
       $\left. \hat{A}^i(t)[\mathcal{H}(i, i)(2\hat{x}_i^0(t) - 2^i\hat{\eta}^0(t) + {}^i\hat{\eta}^i(t)) + \beta_i z^0(t) - e^i(t)] - \right.$ 
       $\left. \hat{g}^i(t) + {}^i\dot{\hat{\eta}}^i(t) - \beta_i [\hat{\eta}^0(t) - u^0(t)] + \kappa \zeta^i(t) + [2Q^i - \frac{3}{4} R^i P^i(t)] e^i(t) \right\}$ 
Outputs:  $u^i(t), \hat{x}_i^0(t), \hat{q}^i(t), e^i(t), P^i(t), \hat{g}^i(t), \hat{\Omega}^i(t)$ 
end for

```

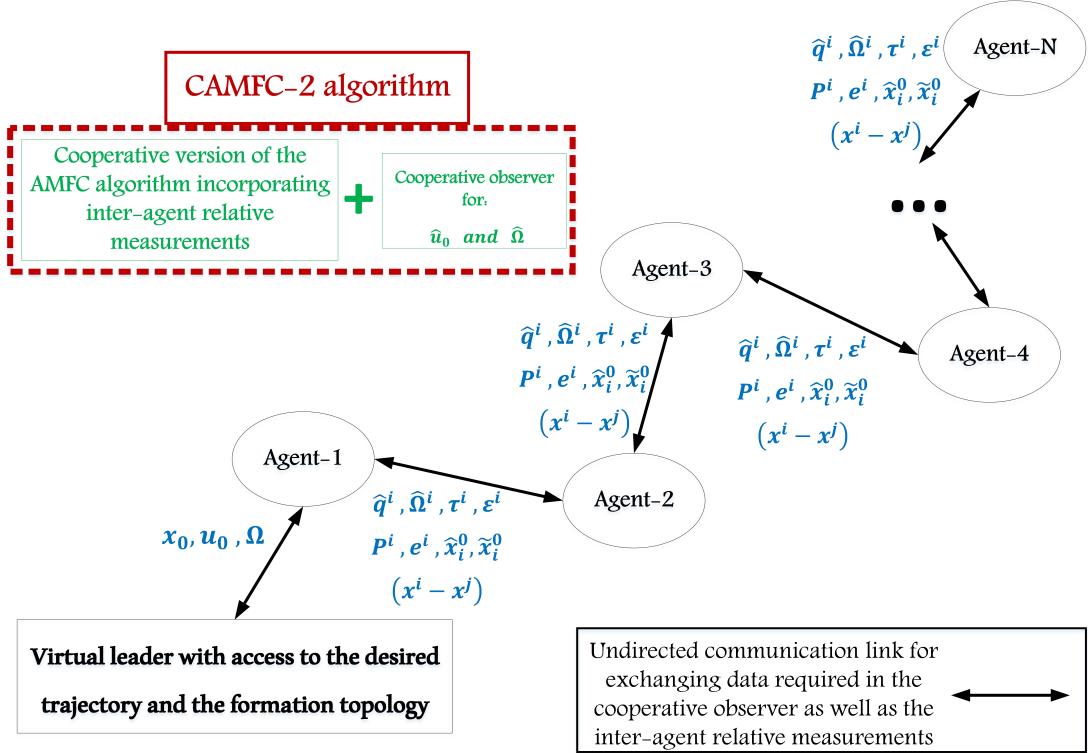


Figure 3.4: Schematic for operation of the CAMFC-2 algorithm for a multi-agent MIMO nonlinear dynamic system

3.6 ACL algorithm for real-time localization within a team of mobile agents using only one beacon agent

In this section, an adaptive cooperative localization (ACL) algorithm is presented.

The ACL algorithm incorporates an adaptive relative position estimating algorithm into a cooperative observer, so as to estimate the absolute position of all agents in the network using one beacon agent.

3.6.1 Design procedure for the adaptive relative position estimation algorithm

Definition 3-18. Consider two moving agents with unknown dynamics establishing a communication link between each other, through which the relative distance and the relative velocity of one to another can be measured. In this regard, $d_r \in \mathbb{R}^+$ is defined

as the relative distance and $V_r \in \mathbb{R}^{n_p \times 1}$ as the relative velocity. The kinematics of relative motion between the two moving agents can be presented as follows

$$\dot{P}_r = V_r , \quad (3.217)$$

where $P_r \in \mathbb{R}^{n_p \times 1}$ is the unknown unmeasurable relative position between the agents.

In this regard, one can observe

$$d_r^2 = P_r^T P_r . \quad (3.218)$$

Here, $n_p > 0$ is the number of position dimensions in the environment where the agents are located. Obviously, we have $n_p \in \{2, 3\}$.

Proposition 3-10. The objective of the adaptive relative position estimating algorithm is to estimate the relative position $\hat{P}_r \in \mathbb{R}^{n_p \times 1}$ between the two agents defined in *Definition 3-18*, such that the relative position estimation error as

$$\epsilon = P_r - \hat{P}_r \quad (3.219)$$

and the corresponding distance estimation error defined by

$$e_p = d_r^2 - \hat{P}_r^T \hat{P}_r , \quad (3.220)$$

converge to small sets around zero as $t \rightarrow \infty$, where $\hat{P}_r \in \mathbb{R}^{n_p \times 1}$ is the estimated relative position between the agents.

Assumption 3-4. It is assumed that the agents defined in *Definition 3-18* move in a way that their relative position and velocity are always non-zero and bounded,

i.e. $V_m^2 \leq V_r^T V_r \leq V_M^2$ and $P_m^2 \leq P_r^T P_r \leq P_M^2$ for $t \in [0, \infty)$ and $V_M, V_m, P_M, P_m \in \mathbb{R}^+$.

Consequently, one can have $D_m \leq d_r \leq D_M$ and $D_{dm} \leq \dot{d}_r \leq D_{dM}$, where $D_M, D_m, D_{dm}, D_{dM} \in \mathbb{R}^+$.

Assumption 3-5. The orientation of the local frames at the two agents defined in *Definition 3-18* are consistent (Han et al., 2019). This can be achieved by having access to the orientation of the earth magnetic field and also by measuring the Euler angles of the local frames using especial IMU modules (Madgwick et al., 2011).

Assumption 3-6. It is assumed that the absolute velocity of each agent in *Definition 3-18* can be computed using a Kalman-filter on the acceleration measurement provided by the on-board IMU modules (Kia et al., 2016). Other techniques can also be used for measuring the absolute velocity vector of AMRs (Boiko & Chehadeh, 2018; Boutayeb et al., 2008; Leishman et al., 2014; Mahony et al., 2012; Rafaralahy et al., 2012; Santosuoso et al., 2011).

Assumption 3-7. The initial relative position (i.e $P_r(0)$ at $t = 0$) between the agents defined in *Definition 3-18* is available while they are stationary. According to (Han et al., 2019), this can be achieved during an *initialization process* for the two static agents. The algorithm proposed in (Diao et al., 2014) can be considered as a solution for the initialization process.

Theorem 3-7. Providing *Assumption 3-4* to *Assumption 3-7*, if one can estimate the relative position between the two agents defined in *Definition 3-18*, by

$$\dot{\hat{P}}_r = [1 + \alpha_p \operatorname{sgn}(e_p h_p)] V_r , \quad (3.221)$$

where

$$h_p = V_r^T \hat{P}_r , \quad (3.222)$$

$sgn(.)$ is the signum function and $\alpha_p \in (0, 1]$; then the objectives presented in *Proposition 3-10* will be achieved.

Proof. Let's define

$$V_8 = \frac{1}{4} e_p^2 . \quad (3.223)$$

By taking the time-derivative, it leads to

$$\dot{V}_8 = e_p (d_r \dot{d}_r - \dot{\hat{P}}_r^T \hat{P}_r) . \quad (3.224)$$

Besides, using the time-derivative of (3.220), one can reach to

$$d_r \dot{d}_r = V_r^T P_r . \quad (3.225)$$

By replacing (3.225) and (3.221) in (3.224), it would be followed by

$$\dot{V}_8 = e_p [V_r^T P_r - V_r^T \hat{P}_r - \alpha_p sgn(e_p h_p) V_r^T \hat{P}_r] . \quad (3.226)$$

Then, by using (3.222) and a little rearrangement, one can have

$$\dot{V}_8 = e \delta_p - \alpha |e_p| |h_p| , \quad (3.227)$$

where $\delta_p = V_r^T P_r - V_r^T \hat{P}_r$ and $|.|$ is the symbol for representing the absolute values.

Recalling *Assumption 3-7* and by considering (3.219) and (3.221), one can have (for

$t > 0$)

$$P_r = P_r(0) + V_r t \quad (3.228)$$

and

$$\hat{P}_r = P_r(0) + [1 + \alpha_p \operatorname{sgn}(e_p h_p)] V_r t . \quad (3.229)$$

Then, by utilizing (3.228) and (3.229), δ_p can be written as

$$\delta_p = V_r^T P_r(0) + V_r^T V_r t - V_r^T P_r(0) - [1 + \alpha \operatorname{sgn}(e_p h_p)] V_r^T V_r t , \quad (3.230)$$

and consequently, one reaches to

$$\delta_p = -\alpha_p \operatorname{sgn}(e_p h_p) V_r^T V_r t . \quad (3.231)$$

Hence, the first term in the right-hand side of equation in (3.227) is written as follows

$$e_p \delta_p = -\alpha_p \operatorname{sgn}(h_p) |e_p| V_r^T V_r t . \quad (3.232)$$

In this equation, we have

$$h_p = V_r^T P_r(0) + [1 + \alpha_p \operatorname{sgn}(e_p h_p)] V_r^T V_r t . \quad (3.233)$$

It can be easily shown that the second term in h_p is positive as long as $0 < \alpha_p \leq 1$.

Thus, by assuming $P(0) = 0$ without any loss of generality, one can say that $h_p > 0$ and consequently $\operatorname{sgn}(h_p) > 0$. By applying this in (3.232) and rephrasing (3.227), one can reach to

$$\dot{V}_8 = -\alpha_p \operatorname{sgn}(h_p) |e_p| V_r^T V_r t - \alpha_p |e_p| |h_p| \leq 0 . \quad (3.234)$$

Then, since $V_8 > 0$ and $\dot{V}_8 \leq 0$, the value of e_p converges to zero asymptotically, based on Lyapunov stability theorem. Since d_r is bounded based on *Assumption 3-4*, and by considering the definition of e_p in (3.220), one can have

$$P_{hm}^2 \leq \hat{P}_r^T \hat{P}_r \leq P_{hM}^2 , \quad (3.235)$$

where $P_{hm}^2 = D_m^2$, $P_{hM}^2 = D_M^2$ and $P_{hm}, P_{hM} \in \mathbb{R}^+$. This shows that \hat{P}_r is bounded. Consequently $\varepsilon = P_r - \hat{P}_r$ is bounded, since P_r is assumed to be bounded per *Assumption 3-4*. Thus for $E_M \in \mathbb{R}^+$, one can define

$$\varepsilon^T \varepsilon \leq E_M . \quad (3.236)$$

In addition, by utilizing (3.228) and (3.229), the value of ε can be represented as follows

$$\begin{aligned} \varepsilon &= P_r(0) + V_r t - P_r(0) - V_r t - \alpha_p \operatorname{sgn}(e_p h_p) V_r t \\ &= -\alpha_p \operatorname{sgn}(e_p h_p) V_r t . \end{aligned} \quad (3.237)$$

Besides, one can have

$$\dot{\varepsilon} = V_r - [1 + \alpha_p \operatorname{sgn}(e_p h_p)] V_r = -\alpha_p \operatorname{sgn}(e_p h_p) V_r = \frac{1}{t} \varepsilon . \quad (3.238)$$

As the second part of the proof, the following Lyapunov function is defined

$$V_9 = \frac{1}{2} \varepsilon^T \varepsilon + V_8 . \quad (3.239)$$

Then, one would have

$$\dot{V}_9 = \dot{\varepsilon}^T \varepsilon + \dot{V}_8 \quad (3.240)$$

By replacing (3.238), it leads to

$$\dot{V}_9 = \frac{1}{t} \varepsilon^T \varepsilon + \dot{V}_8 . \quad (3.241)$$

After that, by recalling (3.234) and (3.236), one reaches to

$$\dot{V}_9 \leq -H_8 + H_9 , \quad (3.242)$$

where

$$H_8 = \alpha_p \operatorname{sgn}(h_p) |e_p| V_r^T V_r t + \alpha_p |e_p| |h_p| \geq 0 \quad (3.243)$$

and

$$H_9 = E_M . \quad (3.244)$$

Based on LaSalle-Yoshizawa theorem, since H_9 is a positive constant value and it is shown previously that e_p is converging to zero asymptotically, one can say that ε is UUB and converges to a bounded set around the origin. Then, the objective presented in *Proposition 3-10* is satisfied and the proof is completed.

■

Remark 3-14. The adaptive law proposed in (3.221) can be used for estimating the relative position between each pair of moving agents that have a joint communication link. An intuitive rival for this algorithm is $\dot{P}_r = V_r$ (Safavi & Khan, 2017). In addition, a novel estimator is presented in (Han et al., 2019) to have more suitable estimation of

the relative positioning among the moving agents. A comparative study is presented in Section 4.6.1 to show that the proposed algorithm in (3.221) outperforms the other two methods.

3.6.2 Design procedure for the cooperative observer in ACL algorithm

Definition 3-19. Consider a network of N mobile agents with the communication graph as defined in *Proposition 3-2*. By assuming a *beacon* agent with the known absolute position, a *beacon pinning gain matrix* is defined as follows

$$\mathcal{B}_b = \text{diag}(\beta_1^b, \beta_2^b, \dots, \beta_N^b) \in \mathbb{R}^{N \times N}, \quad (3.245)$$

in which $\beta_i^b \in \{0, 1\}$ indicates the existence of a communication link between the beacon agent and the i th agent in the network. Furthermore, one can define

$$\mathcal{H}_b = \mathcal{A} + \mathcal{B}_b. \quad (3.246)$$

Here, it is assumed that there is at least one communication connection between one of the agents and the beacon. In other words, at least one of the diagonal elements in \mathcal{B}_b is non-zero.

Proposition 3-11. For the connected agent i to the beacon agent ($i \in [1, N]$), position of the agent $x_p^i \in \mathbb{R}^{n_p \times 1}$ (for $n_p \in \{2, 3\}$) can be presented as

$$x_p^i = x_p^b + \delta_{ib}, \quad (3.247)$$

where $x_p^b \in \mathbb{R}^{n_p \times 1}$ is the position vector for the beacon agent and $\delta_{ib} \in \mathbb{R}^{n_p \times 1}$ is the

measured *relative position* between the connected agent and the beacon. It is assumed that δ_{ib} is available locally at agent i connected to the beacon.

Proposition 3-12. For the unconnected agent i to the beacon ($i \in [1, N]$), the position x_p^i is represented as follows

$$x_p^i = x_p^j + \delta_{ij}, \quad (3.248)$$

where $\delta_{ij} \in \mathbb{R}^{n_p \times 1}$ is the measured relative position between the agent i and its neighboring agent $j \in [1, N]$. It is assumed that the values of δ_{ij} are available at agent i , if there is a communication link between the agents i and j (i.e. $a_{ij} \neq 0$).

Proposition 3-13. Suppose that all of the agents in the network are following the conditions presented in *Assumption 3-4* and *Assumption 3-5*, then, one can use \hat{P}_r estimated by (3.221) in *Theorem 3-7*, to compute the values of δ_{ij} and δ_{ib} as the relative positions (not relative distances) between the neighboring agents, required in (3.247) and (3.248).

Proposition 3-14. The relative difference between two vectors is identical in all of the *orthogonal* local or global coordinates frames (Kane et al., 1993). Hence, the relative difference between the agents position in a local frame \mathbb{P}_i at agent i can be presented as follows

$$\delta_{ij} = [\delta_{ij}]_{\mathbb{P}_i} = [\Delta_i^{\mathbb{P}_i} - \Delta_j^{\mathbb{P}_i}]_{\mathbb{P}_i}, \quad (3.249)$$

where $\Delta_i^{\mathbb{P}_i} \in \mathbb{R}^{n_p \times 1}$ for $i \in [1, N]$ is the vector for position of agent i in the local frame \mathbb{P}_i . In addition, $[.]_{\mathbb{P}_i}$ is a symbol for representing the relative variables in the local frame

\mathbb{P}_i . Since the local frame \mathbb{P}_i is fixed to agent i , the time-derivative of $\Delta_i^{\mathbb{P}_i}$ in \mathbb{P}_i is zero, i.e.

$$[\dot{\Delta}_i^{\mathbb{P}_i}]_{\mathbb{P}_i} = 0. \quad (3.250)$$

Similarly, one can define

$$\delta_{ib} = [\delta_{ib}]_{\mathbb{P}_i} = [\Delta_i^{\mathbb{P}_i} - \Delta_b^{\mathbb{P}_b}]_{\mathbb{P}_i}, \quad (3.251)$$

where $\Delta_b^{\mathbb{P}_b} \in \mathbb{R}^{n_p \times 1}$ is the position vector of the beacon agent at the local frame \mathbb{P}_b fixed to it.

Definition 3-20. According to *Proposition 3-11* and *Proposition 3-12*, a consensus error can be defined for observing the absolute position of agent i as follows

$$\tau_p^i = \sum_{j=1}^N a_{ij} [\hat{x}_p^i - (\hat{x}_p^j + \delta_{ij})] + \beta_i^b [\hat{x}_p^i - (x_p^b + \delta_{ib})]. \quad (3.252)$$

By recalling (3.249) and defining

$$\hat{s}_p^i = \hat{x}_p^i - \Delta_i^{\mathbb{P}_i}, \quad (3.253a)$$

$$\hat{s}_p^j = \hat{x}_p^j - \Delta_j^{\mathbb{P}_j}; \quad (3.253b)$$

the observer consensus error can be represented in a lumped format as follows

$$\tau_p = (\mathcal{H} \otimes I_{n_p}) \hat{s}_p - (\mathcal{B}_b \otimes s_p^b) \mathbf{1}, \quad (3.254)$$

where $\hat{s}_p = [\hat{s}_p^1; \hat{s}_p^2; \dots; \hat{s}_p^N]$, $s_p^b = x_p^b - \Delta_b^{\mathbb{P}_b}$ and $\mathbf{1} \in \mathbb{R}^{Nn_p \times 1}$ is a vector including one for all the elements.

Proposition 3-15. If the consensus error defined in (3.254) converges to zero, one can say that the distributed observation objective is achieved and \hat{x}_p^i reaches to x_p^i for all $i \in [1, N]$.

Lemma 3-8. If one uses the following equation as the rate for observing the position of agent i ,

$$\dot{x}_p^i = -\lambda_2 \tau_p^i - [\mathcal{M}(\operatorname{sgn}\{\sum_{j=1}^N (\mathcal{H}_b(i, j) \tau_p^j)\} M_p^b)] \quad (3.255)$$

where $\lambda_2 > 0$ is a scalar gain, $\mathcal{M}(\operatorname{sgn}\{\sum_{j=1}^N (\mathcal{H}_b(i, j) \tau_p^j)\}) \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose elements on the main diameter are the sign of elements in $\sum_{j=1}^N (\mathcal{H}_b(i, j) \tau_p^j) \in \mathbb{R}^{n_p \times 1}$ (refer to *Definition 3-8*) and $M_p^b \in \mathbb{R}^{n_p \times 1}$ includes the maximum absolute values for the elements of s_p^b ; then the *Proposition 3-15* would be achieved.

Proof. The proof is same as the procedure provided as the proof for *Theorem 3-4*, with a slightly different. Following the same procedure in the proof of *Theorem 3-4*, the cooperative observer algorithm would be produced for \dot{s}_p^i , i.e.

$$\dot{s}_p^i = -\lambda_2 \tau_p^i - [\mathcal{M}(\operatorname{sgn}\{\sum_{j=1}^N (\mathcal{H}_b(i, j) \tau_p^j)\} M_p^b)]. \quad (3.256)$$

This is due to the fact that the consensus error in (3.254) includes \dot{s}_p^i and not the observed parameter itself (i.e. the position \hat{x}_p^i). But the consensus error included in *Theorem 3-4*, contains the observed parameter. Having this in mind and according to (3.256), the cooperative observer in (3.255) is achieved by referring to (3.250), (3.253a) and (3.253b); knowing that the change in rate of $\Delta_i^{\mathbb{P}_i}$ is zero in the local coordinates frame \mathbb{P}_i . ■

Remark 3-15. Here, M_p^b is the maximum absolute value for speed of the mobile agent, which can be defined according to the actuators specifications and some data from previous experiments.

Remark 3-16. As it is suggested in *Remark 3-10*, the value of scalar gain λ_2 should be large enough in order to reach fast finite-time convergence of the distributed estimation algorithm.

3.6.3 ACL algorithm

The ACL algorithm is presented in Table 3.5 and Fig. 3.5. The algorithm combines the adaptive relative position estimation algorithm and a cooperative observer to provide the relative and absolute positions of the mobile agents in a network with only one beacon agent.

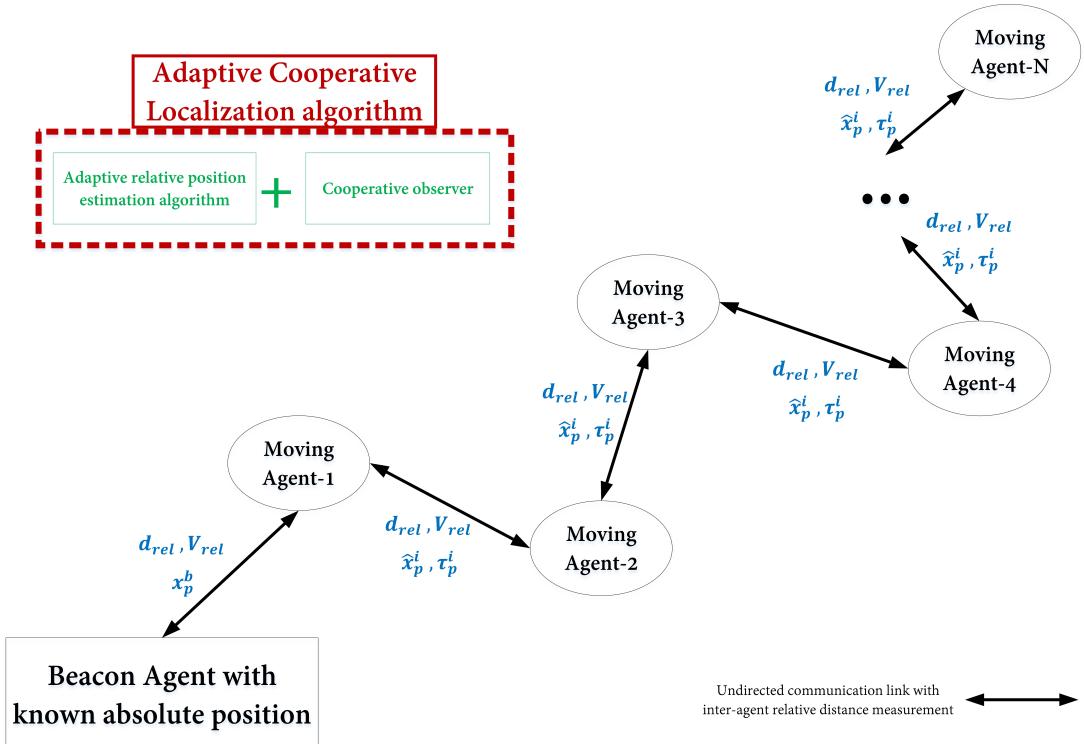


Figure 3.5: The schematic for adaptive cooperative localization (ACL) algorithm

Table 3.5: ACL algorithm

Algorithm 4: ACL (at agent i in the network)
Initialization:
$\alpha_p = (0, 1]$, $\lambda_2 = 100$, $M_p^b = 10 \times \mathbf{1}_n$
Note-1: t_f is the simulation/implementation duration time, and dt is the sampling time .
Main Loop:
for $\{t = d_t : d_t : t_f\}$ do Inputs: $\{x_p^b, V_r^{ij}(t), d_r^{ij}(t), V_r^{ib}(t), d_r^{ib}(t), \hat{x}_p^j(t - d_t), \tau_p^j(t - d_t), a(i, j), \beta_i, \mathcal{H}_b(i, j) \ (j \neq i \in [1, N])\}$ for $\{j = 1 : N\}$ do if $\{a(i, j) == 1\}$ do 1-1: $e_p^{ij}(t) = (d_r^{ij}(t))^2 - (\hat{P}_r^{ij}(t - d_t))^T \hat{P}_r^{ij}(t - d_t)$ 1-2: $h_p^{ij}(t) = (V_r^{ij}(t))^T \hat{P}_r^{ij}(t - d_t)$ 1-3: $\dot{\hat{P}}_r^{ij}(t) = [1 + \alpha_p \operatorname{sgn}(e_p^{ij}(t)h_p^{ij}(t))]V_r^{ij}(t)$ 1-4: $\delta_{ij}(t) = \delta_{ij}(t - d_t) + \{\dot{\hat{P}}_r^{ij}(t) \times d_t\}$ end if end for if $\{\beta_i^b == 1\}$ do 2-1: $e_p^{ib}(t) = (d_r^{ib}(t))^2 - (\hat{P}_r^{ib}(t - d_t))^T \hat{P}_r^{ib}(t - d_t)$ 2-2: $h_p^{ib}(t) = (V_r^{ib}(t))^T \hat{P}_r^{ib}(t - d_t)$ 2-3: $\dot{\hat{P}}_r^{ib}(t) = [1 + \alpha_p \operatorname{sgn}(e_p^{ib}(t)h_p^{ib}(t))]V_r^{ib}(t)$ 2-4: $\delta_{ib}(t) = \delta_{ib}(t - d_t) + \{\dot{\hat{P}}_r^{ib}(t) \times d_t\}$ end if 3: $\tau_p^i(t) = \beta_i^b [\hat{x}_p^i(t - d_t) - (x_p^b(t) + \delta_{ib}(t))]$ for $\{j = 1 : N\}$ do 4: $\tau_p^i(t) = \tau_p^i(t) + a(i, j) [\hat{x}_p^i(t - d_t) - (\hat{x}_p^j(t - d_t) + \delta_{ij}(t))]$ end for 5: $\dot{\hat{x}}_p^i(t) = -\lambda_2 \tau_p^i(t)$ for $\{j = 1 : N\}$ do 6: $\dot{\hat{x}}_p^i(t) = \dot{\hat{x}}_p^i(t) - \mathcal{M}(\operatorname{sgn}\{\mathcal{H}_b(i, j)\tau_p^j(t - d_t)\}) \times M_p^b$ end for 7: $\hat{x}_p^i(t) = \hat{x}_p^i(t - d_t) + \{\dot{\hat{x}}_p^i(t) \times d_t\}$ Outputs: $\hat{x}_p^i(t), \delta_{ij}(t), \tau_p^i(t)$ end for

3.7 Summary

In this chapter, detailed design procedures of the AMFC algorithm for SISO and MIMO dynamic systems, two cooperative AMFC protocols for consensus and formation control problems within a multi-agent system of MIMO dynamic systems and the ACL algorithm for cooperative localization in a team of mobile agents including one beacon agent are all presented with detailed proofs. The design procedures include the stability analyses based on the Lyapunov and LaSalle-Yoshizawa stability theorems. It is shown that the tracking error as well as the parameter estimation errors converge to a small bounded set around the origin, following the UUB property. All of the provided cooperative observers are designed in a way to force the corresponding estimation consensus errors to zero in finite-time. In addition, it is shown that the consensus errors for formation-tracking problem in the cooperative AMFC algorithms are all UUB. Similarly, the UUB convergence is provided for the proposed relative position estimation algorithm. In this chapter, each of the solutions is accompanied by a schematic diagram and a detailed step-by-step algorithm to provide better representation. Here, all the single-agent and multi-agent systems are supposed to have completely unknown dynamics and working under unknown bounded external disturbances.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the simulation results are presented to assess the performance of the algorithms proposed in this thesis. In Section 4.2, two comparison case studies are investigated to show that the AMFC for SISO single-agent nonlinear dynamic systems outperforms the well-known iPI and sliding-mode controllers in terms of the tracking objective as well as the smoothness of the produced control signals. Moreover, the algorithm is exemplified in a chaotic nonlinear dynamic system. After that in Section 4.3, the AMFC algorithm for MIMO nonlinear dynamic systems is compared with two state-of-the-arts reinforcement learning algorithms. It is shown that the AMFC algorithm requires fewer number of adaptive laws to achieve the tracking objective. In addition, the proposed algorithm outperforms the established algorithm in terms of tracking performance with minimal control effort. It is then followed by the application of the AMFC algorithm in an automated robotic manipulator, an autonomous wheeled mobile robot and an autonomous quadrotor.

Furthermore in Section 4.4, the CAMFC-1 algorithm is applied on a team of four autonomous quadrotors for formation-tracking problem in a 3D environment. In Section 4.5, a comparative analysis is presented to evaluate the performance of the CAMFC-2 algorithm. Three comparative case studies have been presented to showcase the application of CAMFC-2. The first case study evaluate the proposed CAMFC-2 on a consensus problem. This is followed by an application on a formation

control problem involving a group of agents with nonlinear dynamics. The section ends with an application of the proposed CAMFC-2 on a team of autonomous quadrotors.

Then in section 4.6, the ACL algorithm is compared in two different case studies with the state-of-the-art cooperative localization solutions in the literature. In addition, the combination of ACL and CAMFC-2 algorithms is applied in a team of autonomous quadrotors to evaluate the performance of these algorithms for achieving the localization and formation-tracking objectives.

The last phase of study is a hardware-in-the-loop test to validate the AMFC algorithm one step before the real implementation of this algorithm. In the provided test, the AMFC algorithm is embedded on a development board, while the dynamics of a wheeled mobile robot and an autonomous quadrotor are emulated on the second development board.

4.2 Results for the AMFC algorithm in a single SISO nonlinear dynamic systems

In this section, the control signals for the AMFC algorithm in all of the simulation case studies are defined according to (3.8a) to (3.10b). The corresponding tuning parameters for the AMFC algorithm are presented for each case study wherever required.

4.2.1 Comparison study for an unstable nonlinear system

This is the case-1 of the comparative study performed for AMFC algorithm in SISO systems. Here, an unstable SISO nonlinear system is considered for evaluating

the performance of the proposed AMFC algorithm. Dynamic system for the unstable plant including one control input and one system output is considered as follows (Fliess & Join, 2013)

$$\begin{aligned}\dot{x}_s &= x_s + u_s^3 \\ y_s &= x_s .\end{aligned}\tag{4.1}$$

Here, the performance of AMFC is compared with a well-known sliding-mode controller (SMC) defined as (Levant, 2003)

$$\begin{aligned}u_s &= -\frac{\dot{y}_s^d - e_s + \rho_s \text{sat}(\varsigma)}{b} \\ \text{sat}(\varsigma) &= \frac{\varsigma}{|\varsigma| + \varepsilon_{slid}}\end{aligned}\tag{4.2}$$

with $\rho_s \in \mathbb{R}^+$; and an MFC policy (i.e. iPI) proposed in (Fliess & Join, 2013) as

$$\begin{aligned}u_s &= -\frac{\hat{f}_s - \dot{y}_s^d - k_p e_s - k_i \vartheta}{b} \\ \hat{f}_s &= \frac{1}{l_0} \int_{t-l_0}^t [\dot{y}_s^d - bu_s + k_p e_s + k_i \vartheta] d\tau_t .\end{aligned}\tag{4.3}$$

Here, the parameters y_s^d , ς , ϑ and e_s are defined in Section 3.2, $\varepsilon_{slid} \in \mathbb{R}^+$ is very small value preventing chattering in SMC, and k_p , k_i and l_0 are positive values for the tuning parameters in the iPI control policy. Besides, the following cost function

$$\mathcal{C}_s = \int_0^{t_f} [e_s^2 + u_s^2] d\tau_t ,\tag{4.4}$$

where t_f is the terminal time for the simulation study, is computed to measure the controllers performance. The cost function in (4.4) considers the tracking performance with the expense of control effort. Hence the controller with less

tracking error but too much control effort is also penalized. Tuning parameters for the controllers are determined such that one can have nearly equal values of \mathcal{C}_s among them, since the controllers are strong enough to have the same tracking performance (with minor differences). In this way, one can compare the controllers in terms of the *fluctuations* in the control signals. The properties of the three mentioned controllers for simulation of the dynamic system in (4.1) are presented in Table 4.1. Here, $r = q = 1$. For AMFC algorithm, the adaptive gains (i.e. γ_1 and γ_2) are tuned to be large enough, while the leakage gains (i.e. ϖ_1 and ϖ_2) are chosen small enough, in order to have fast convergence in the online estimations, as suggested in *Remark 3-1*. As mentioned before, the values of \mathcal{C}_s corresponding to each controller are kept almost the same by choosing the appropriate tuning parameters in SMC and iPI methods.

The simulation results for comparison of the controllers' performances are depicted in Fig. 4.1 to Fig. 4.5. As can be seen in Fig. 4.1, the tracking performance of AMFC is superior to those of iPI and SMC. Moreover as in Fig. 4.4 and Fig. 4.5, the convergence to finite values is observed for unknown linear and nonlinear parameters and consequently the controller gain in the AMFC algorithm. Note that there is not online estimation of neither unknown terms nor the controller gain in the SMC algorithm. Also, the linear term and the controller gain are not updated online in the iPI algorithm. Hence, there is not any graph for these two algorithms in Fig. 4.4 and Fig. 4.5. In the mentioned two figures, the graphs for AMFC algorithm are depicted only to show that the update rules (i.e. the two adaptive laws for unknown terms and the DRE for the controller gain) work well and provide stable online estimations.

Relatively smoother control signal for AMFC algorithm is disclosed in Fig. 4.2. The control signals produced by SMC and iPI exhibit more fluctuations and aggressive perturbations. On contrary, the AMFC control signal suggests a *smoother* control effort, as depicted in Fig. 4.6 where the fast Fourier transform (FFT) of the control input signals produced by SMC, iPI and AMFC are compared. The control signals generated by SMC and iPI have several dominant frequencies less than 150 Hz as presented in Table 4.1. The control signal produced by AMFC does not have any dominant frequencies. The dominant frequencies in a control signal can excite the natural and structural frequencies of any dynamical system and consequently can lead to severe problems in practice. Hence, smoother control signal is preferred (Rad & Tsang, 2000).

Table 4.1: Properties of the MFC controllers implemented on an unstable nonlinear system

Parameter	SMC (as in (4.2))	iPI (as in (4.3))	proposed AMFC (as in (3.10a) and (3.10b))
Tuning parameters	$\rho_s = 10$ $\varepsilon_{slid} = 0.1$	$k_p = 100$ $k_i = 100$ $l_0 = 10$	$\gamma_1 = 1e5$ $\gamma_2 = 1e3$ $\varpi_1 = 0.01$ $\varpi_2 = 0.001$
Value of \mathcal{C}_s	23.26	23.26	23.54
Dominant freq. (Hz)	40, 89 118, 160	23, 63	-

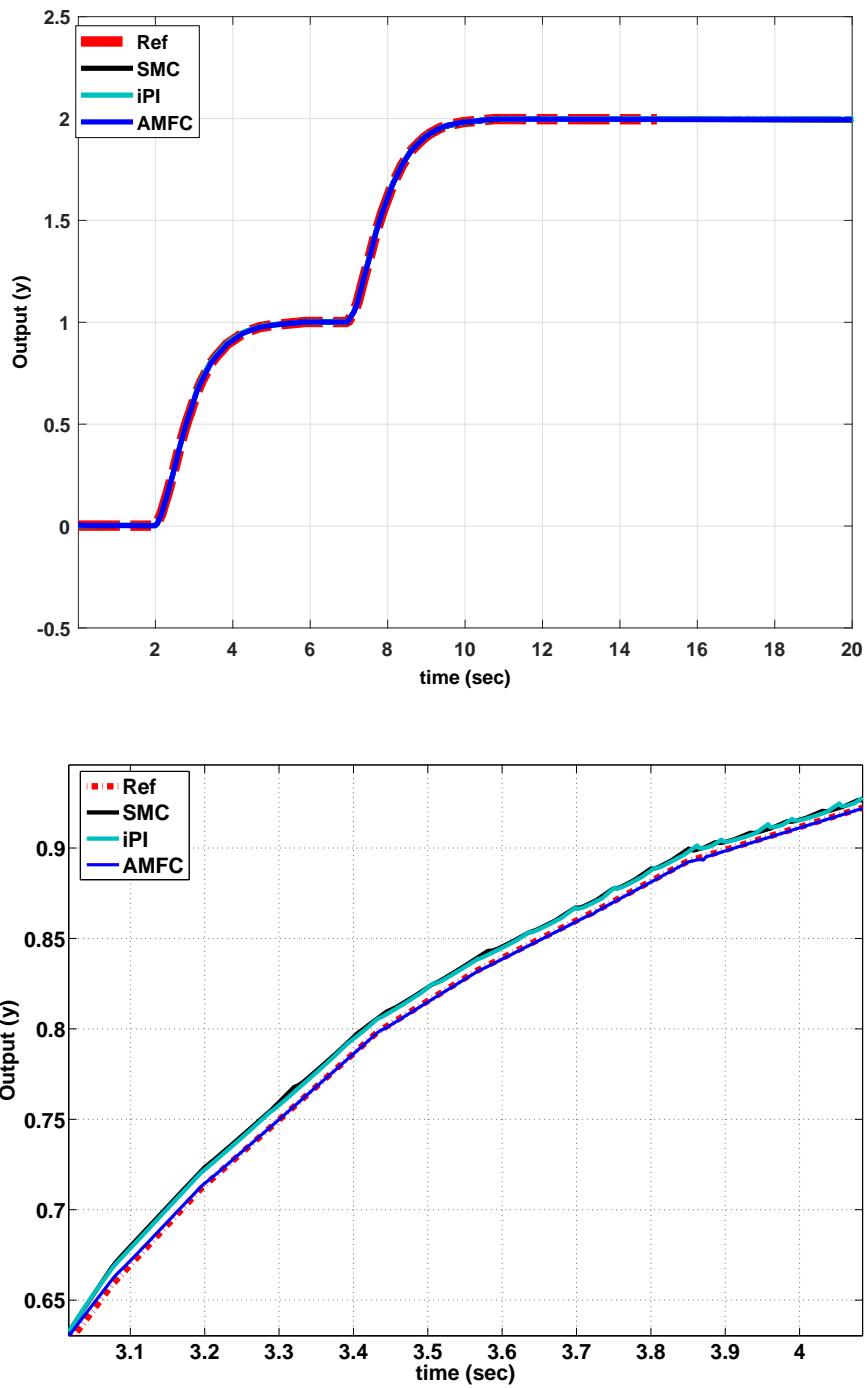


Figure 4.1: Tracking performances of the AMFC, the SMC and iPI algorithms, implemented on an unstable nonlinear system; the whole (top) and in detail (bottom)

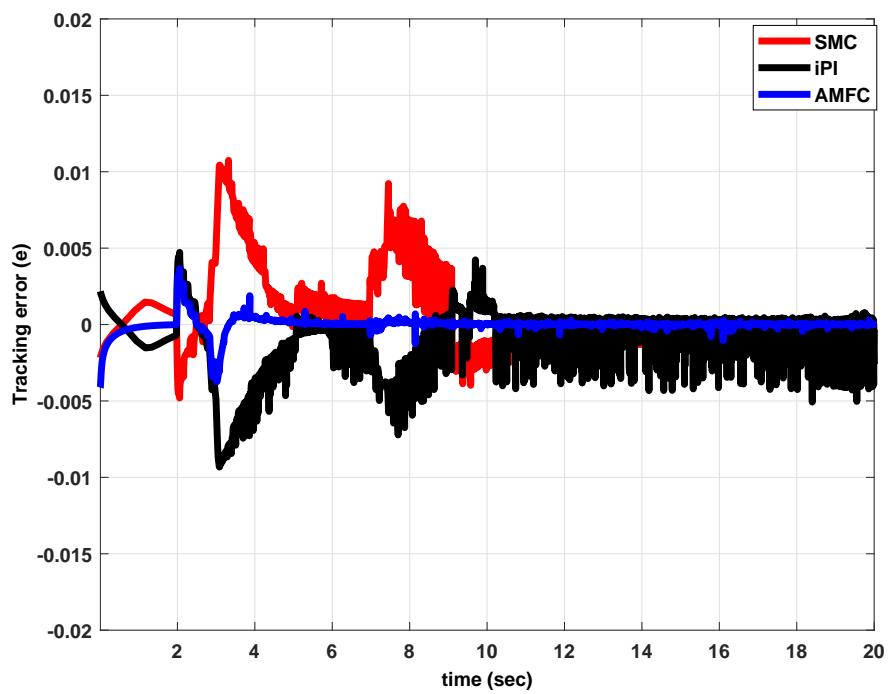


Figure 4.2: Tracking errors of the AMFC, SMC and iPI algorithms, implemented on an unstable nonlinear system.

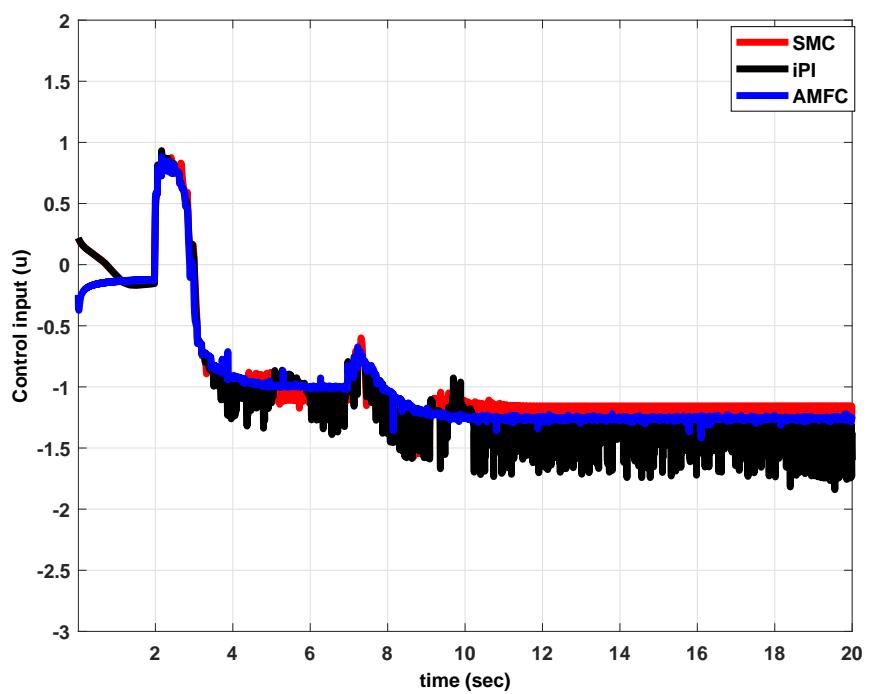


Figure 4.3: The control signals of the AMFC, SMC and iPI algorithms, implemented on an unstable nonlinear system.

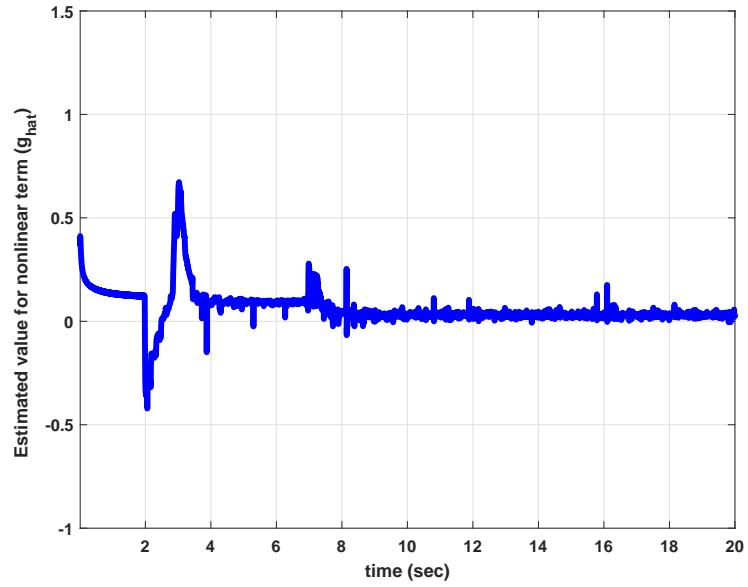


Figure 4.4: Estimated value for unknown nonlinear term, utilizing the AMFC algorithm on an unstable nonlinear system. Stable performance of the adaptive law is disclosed.

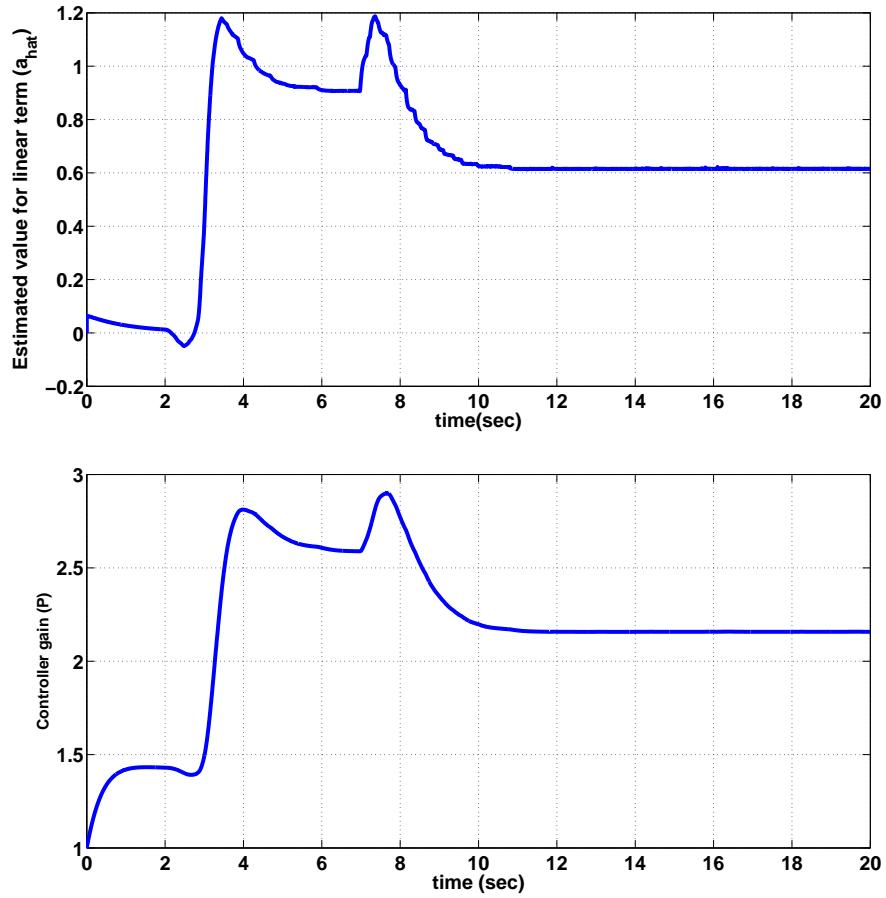


Figure 4.5: Estimated value for unknown linear term (top); and the main controller gain (bottom), utilizing the AMFC algorithm on an unstable nonlinear system. Stable performances for the adaptive law and the DRE are deduced.

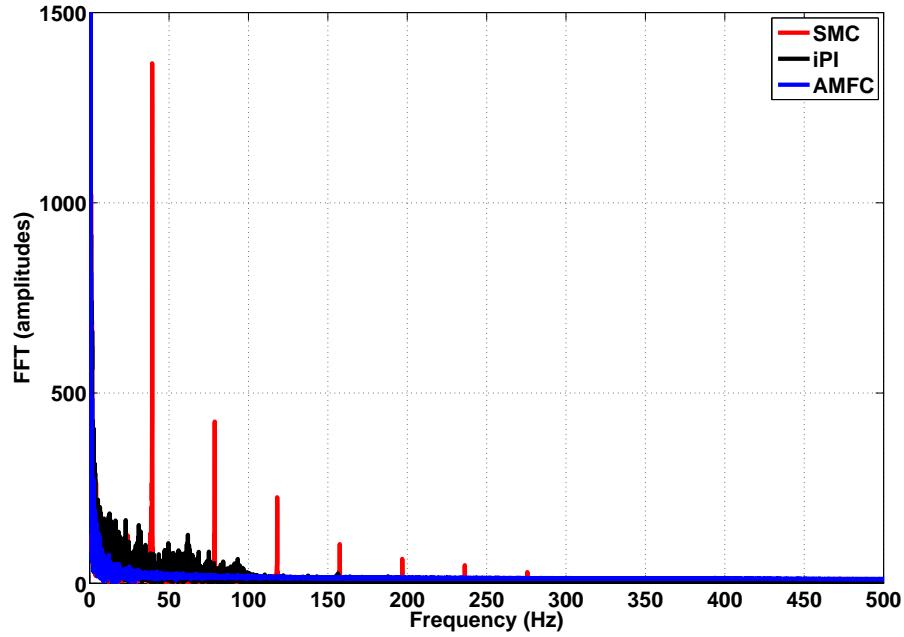


Figure 4.6: The FFT for control inputs of the AMFC, SMC and iPI algorithms, implemented on an unstable nonlinear system.

4.2.2 Comparison study for a delay system

This is the case-2 of the comparative study performed for AMFC algorithm in SISO systems. After observing the performance of the AMFC in a SISO unstable nonlinear system, a system with time delay dynamics as follows (Fliess & Join, 2013)

$$\dot{y}_s(t) = y_s(t) + 5y_s(t - \tau_d) + u_s(t) \quad (4.5)$$

where τ_d is a time-varying delay function as

$$\tau_d(t) = \tau_d(t - T_0) + 10T_0 sgn(N_d(t)) , \quad \tau_d(0) = 2.5s , \quad (4.6)$$

is considered for the second comparative study. Here, T_0 is a positive constant value and $N_d(t)$ is a zero-mean Gaussian distribution with standard deviation equal to 1. Assuming the SMC and iPI controllers as defined in (4.3) and (4.4), a comparative

analysis is provided for the AMFC algorithm, upon the tracking performances in the delay dynamic system. The tuning parameters and the properties of the controllers for simulation of the dynamic system in (4.5) are presented in Table 4.2.

The AMFC is tuned same as in 4.2.1, without any change in the values of tuning parameters. But the tuning parameters for SMC and iPI are adjusted so as to have equal values of \mathcal{C}_s among all the three control policies. The simulation results for this comparison study are presented in Fig. 4.7 to Fig. 4.9, showing that performance of the proposed AMFC algorithm supersedes the rest. In addition, the achieved asymptotic convergence for \hat{h} , \hat{a} and consequently p in AMFC algorithm are depicted in Fig. 4.10 and Fig. 4.11. As it is observed, the estimated values for unknown terms as well as the value for main controller gain are bounded and converge to some finite-values; i.e. the estimations have reached to stable solutions. Online adaptation of p , reduces the efforts needed for off-line tuning of the other tuning parameters in AMFC, while the tuning parameters for SMC and iPI control policies should be adjusted again to provide appropriate tracking performance. In this case, there is a dominant frequency about 28 Hz in the control signals of SMC and iPI, while the AMFC control signal is smoother without any dominant frequency (Fig. 4.12).

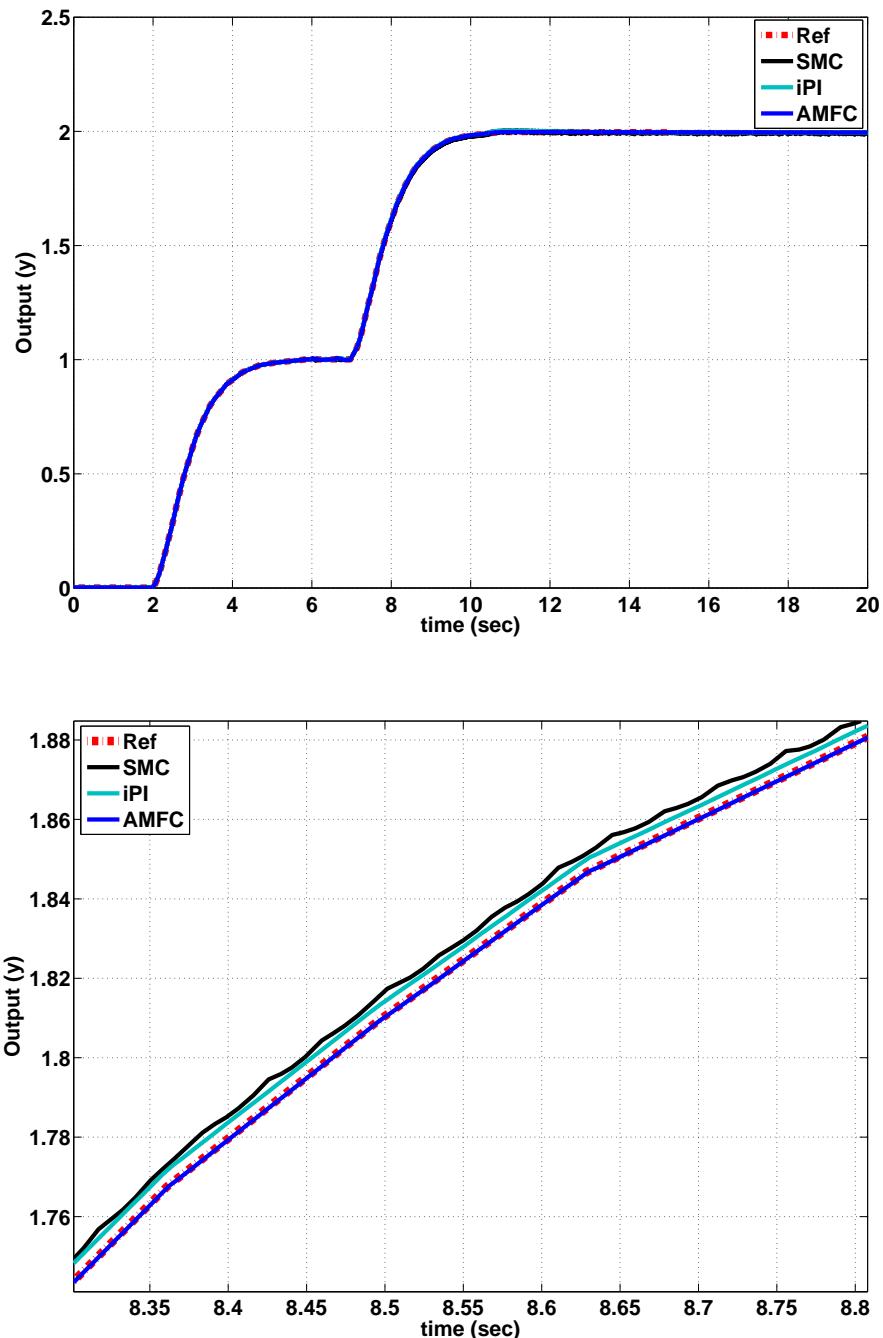


Figure 4.7: Tracking performance of AMFC algorithm with the SMC and iPI algorithms in a delay system; the whole (top) and in detail (bottom)

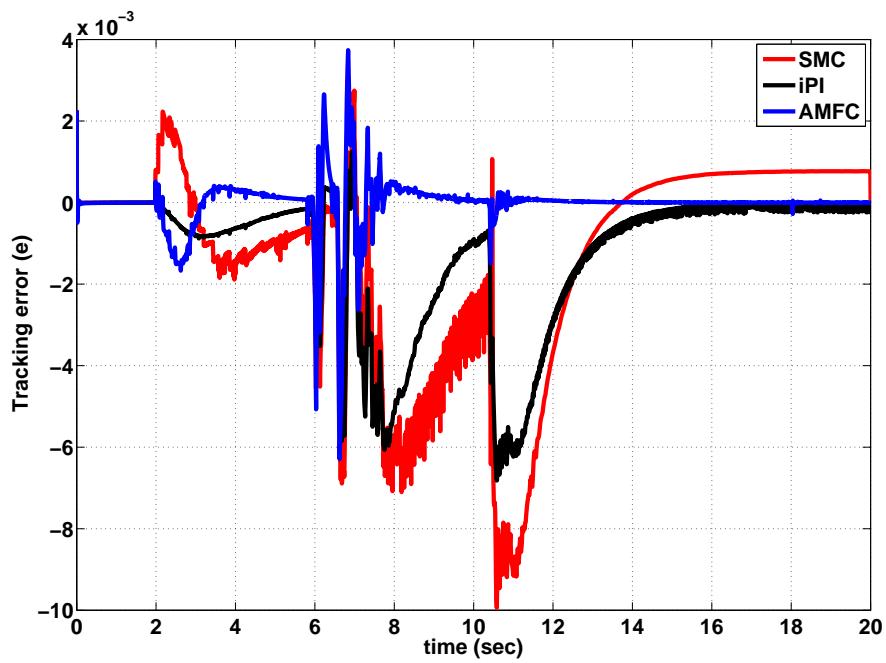


Figure 4.8: Tracking errors of the AMFC, SMC and iPI algorithms, implemented on a delay system.

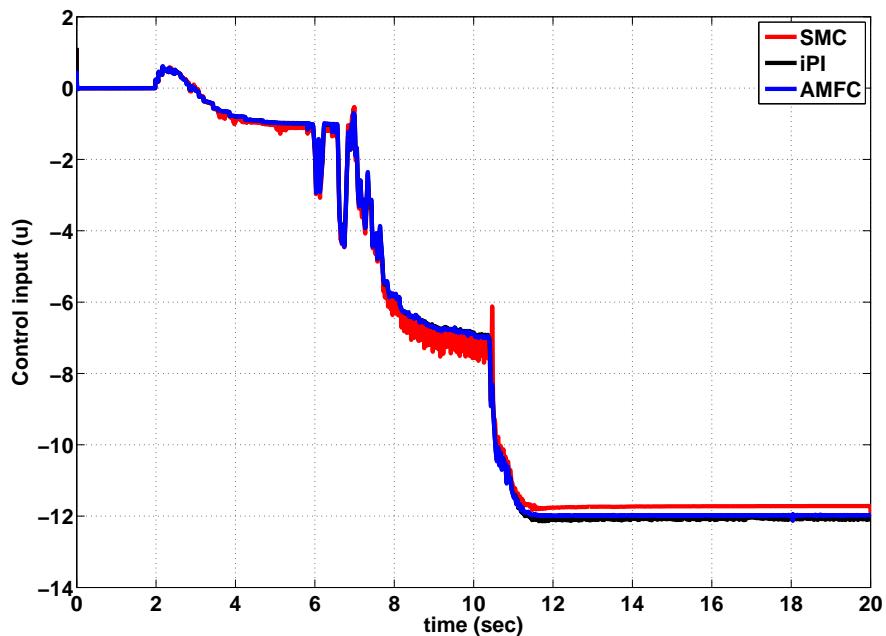


Figure 4.9: The control signals of the AMFC, SMC and iPI algorithms, implemented on a delay system.

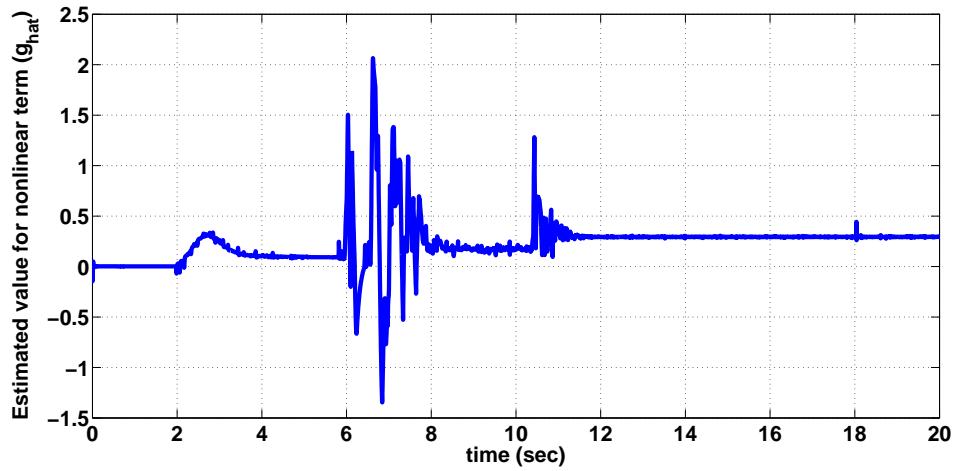


Figure 4.10: Estimated values for unknown nonlinear term in a delay system, utilizing the AMFC algorithm. Stable performance of the adaptive law is disclosed.

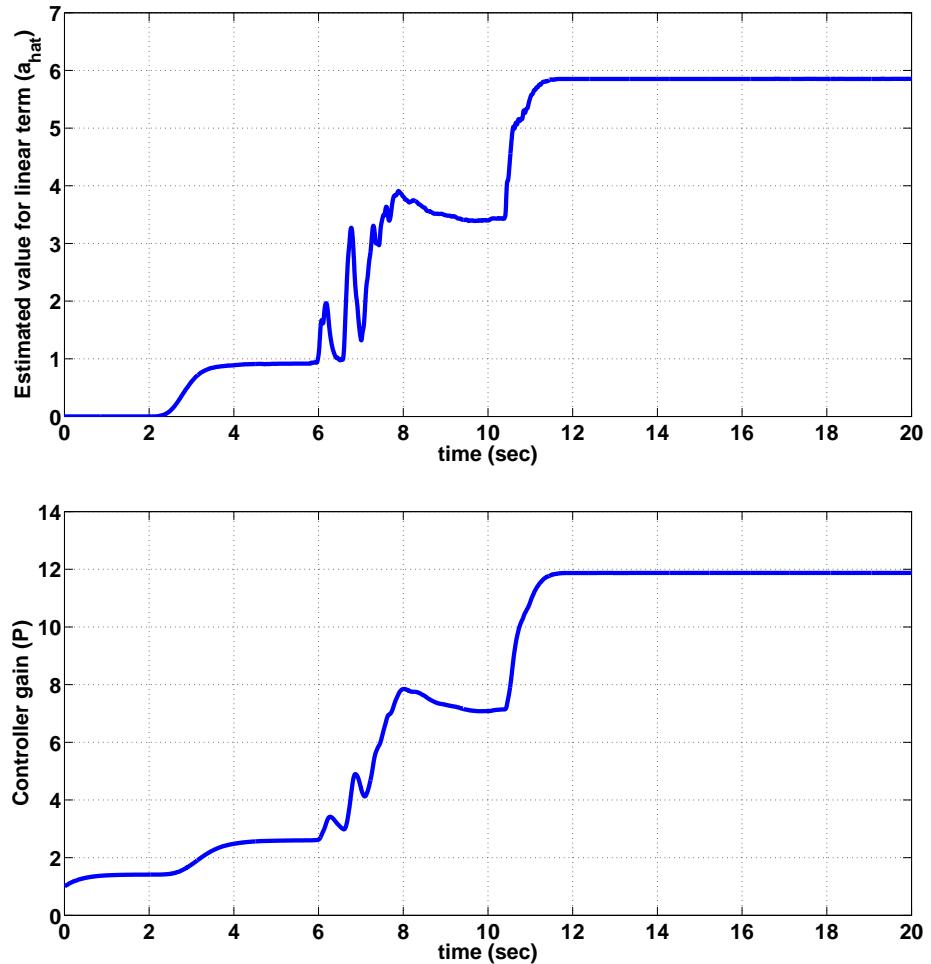


Figure 4.11: Estimated value for unknown linear term (top); and the main controller gain (bottom), utilizing the AMFC algorithm on a delay system. Stable performances for the adaptive law and the DRE are deduced.

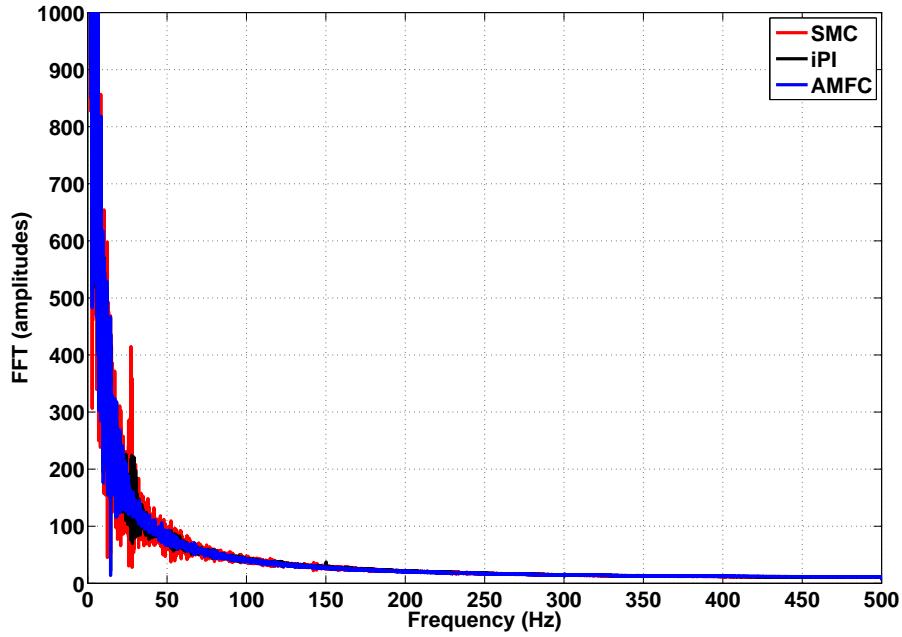


Figure 4.12: The FFT of control inputs for AMFC, SMC and iPI algorithms, implemented on a delay system.

Table 4.2: Properties of the MFC controllers applied to a delay system

Parameter	SMC (as in (4.2))	iPI (as in (4.3))	proposed AMFC (as in (3.10a) and (3.10b))
Tuning parameters	$\rho_s = 50$ $\varepsilon_{slid} = 0.1$	$k_p = 500$ $k_i = 500$ $l_0 = 10$	$\gamma_1 = 1e5$ $\gamma_2 = 1e3$ $\varpi_1 = 0.01$ $\varpi_2 = 0.001$
Value of \mathcal{C}_s	1475	1486	1484
Dominant freq. (Hz)	28	28	-

4.2.3 Application of AMFC algorithm on a Duffing-Holmes chaotic system

In this part for evaluating the AMFC in SISO nonlinear dynamic systems, the proposed AMFC is applied to a Duffing-Holmes chaotic system which is a well-known dynamic oscillator. This system which has a double-integrator dynamics,

is modeled as follows (Chen & Ge, 2013)

$$\dot{x}_s^1 = x_s^2$$

$$\dot{x}_s^2 = -p_1 x_s^1 - p_2 x_s^2 - (x_s^1)^3 + q_D \cos \omega_D t + h_D(x_s, u_s) + d_D(t) \quad (4.7)$$

$$y_s = x_s^1,$$

where $p_1 = 0.3 + 0.2 \sin 10t$, $q_D = 5 + 0.1 \cos t$, $p_2 = 0.2 + 0.2 \cos 5t$, $\omega_D = 0.5 + 0.1 \sin t$, $h_D(x, u) = u_s + 0.5 \cos u_s$ and the external disturbance is $d_D = 0.4 \sin 0.2\pi t + 0.3 \sin x_s^1 x_s^2$. The desired trajectory is considered as $y_s^d = \sin t + \cos 0.5t$. The tuning parameters for AMFC algorithm in this case is presented in Table 4.3. These parameters are tuned off-line utilizing the procedure suggested in *Remark 3-1* (i.e. the adaptive gains should be large enough and the leakage gains should be small enough). According to the simulation results depicted in Fig. 4.13 to Fig. 4.15, the tracking objective is satisfied and the tracking error is bounded around origin. Furthermore, by observing Fig. 4.15, since the dynamics in (4.7) is a double-integrator SISO system, there are two unknown linear and nonlinear parts corresponding to the two system states (i.e. x_s^1 and x_s^2). The values for the first parts of \hat{h} and \hat{a} are relatively too small, as the main nonlinear dynamics of the system is incorporated in the dynamics of x_s^2 (i.e. in the second equation in (4.7)).

Table 4.3: Tuning parameters of the AMFC algorithm implemented on a Duffing-Holmes chaotic dynamic system.

Parameters and the corresponding values (as in (3.10a) and (3.10b))
$\gamma_1 = 1e3$
$\gamma_2 = 1$
$\varpi_1 = 0.1$
$\varpi_2 = 0.01$

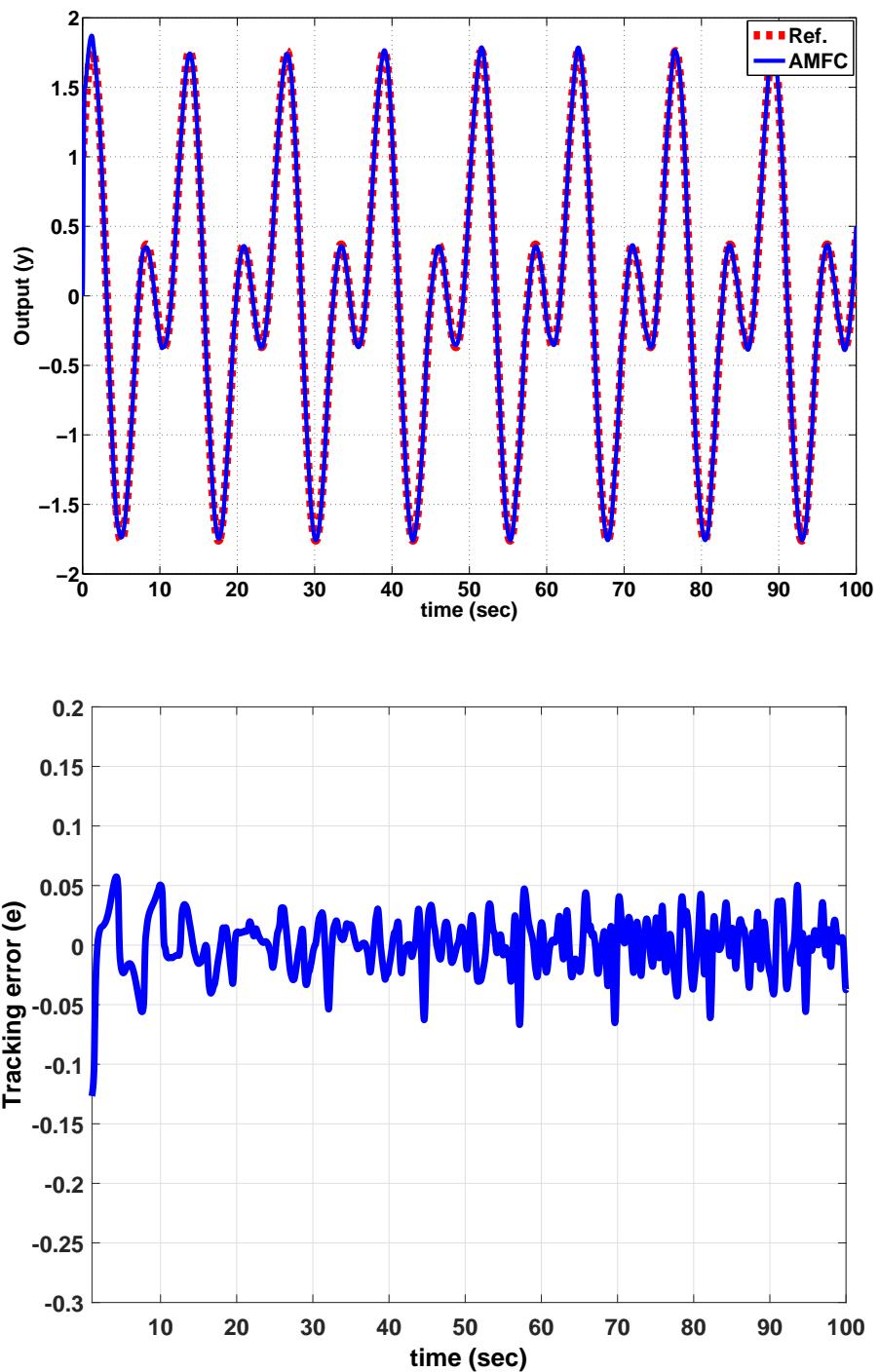


Figure 4.13: Tracking performance of the Duffing-Holmes chaotic system, utilizing the AMFC algorithm. Bounded tracking error is observed.

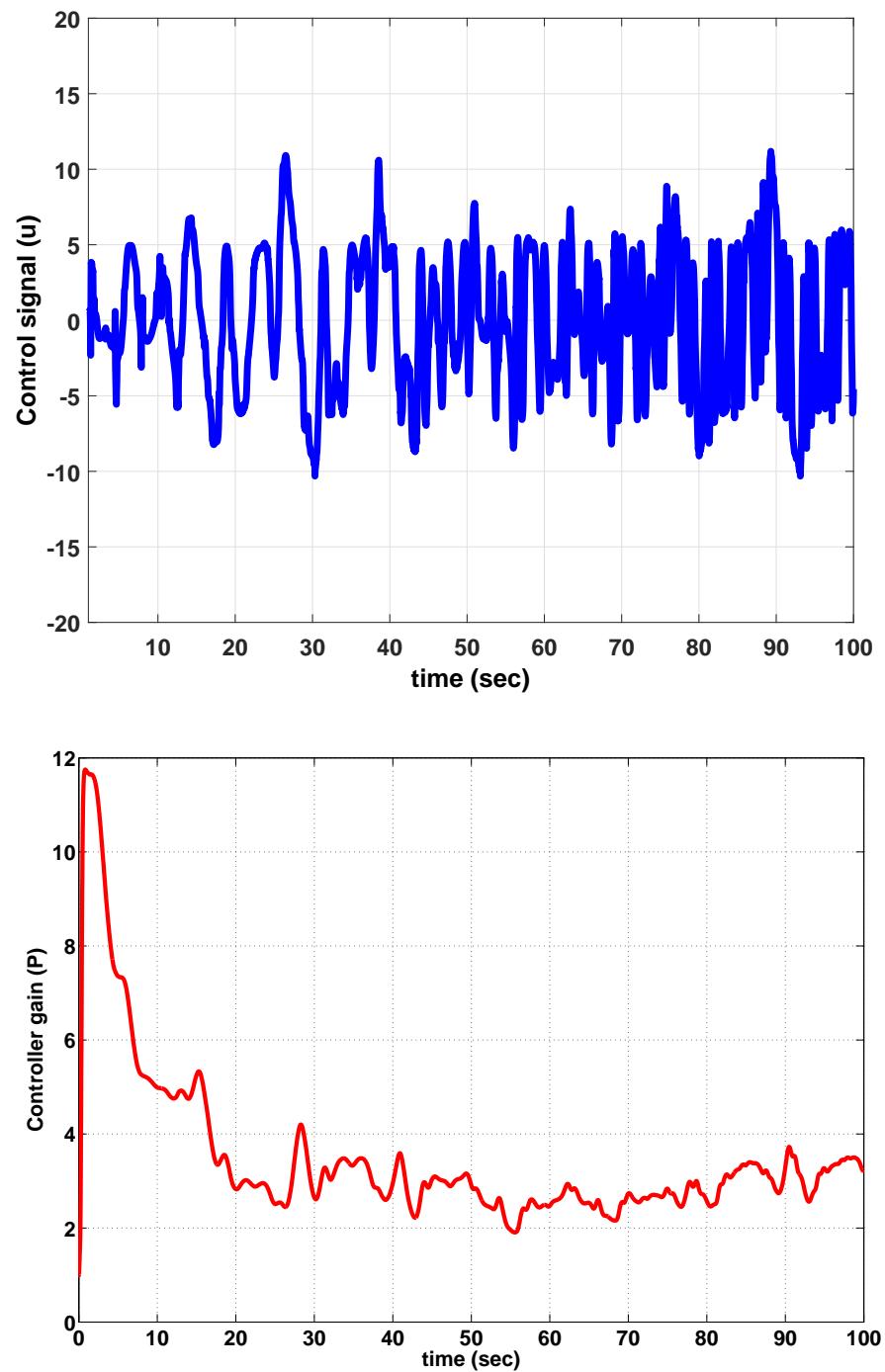


Figure 4.14: Values for the control signal (top); and the controller gain (bottom), utilizing the AMFC algorithm on a Duffing-Holmes chaotic system. The values are all bounded.

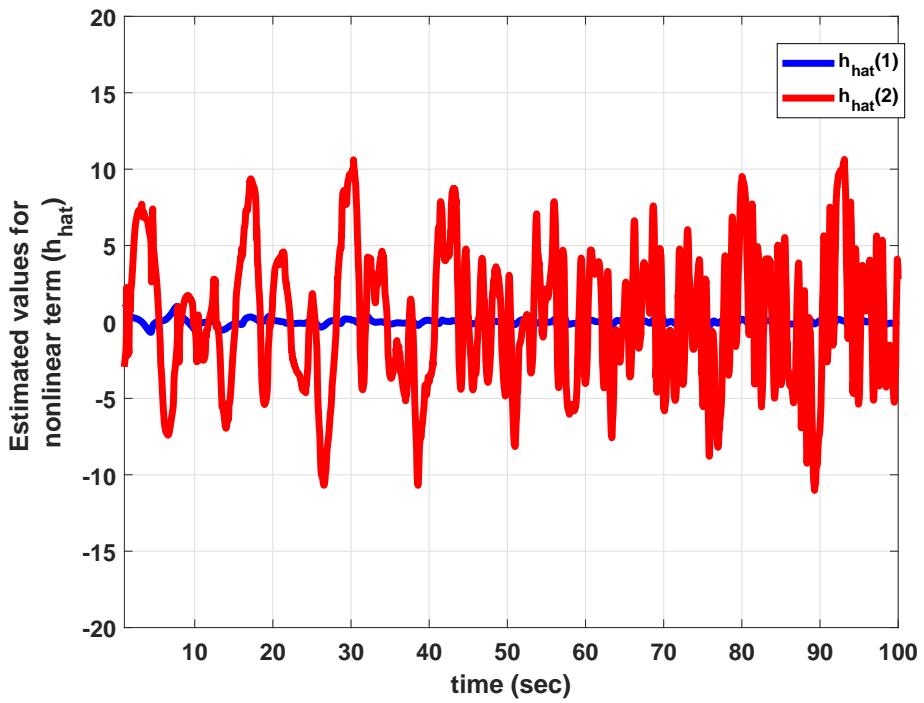
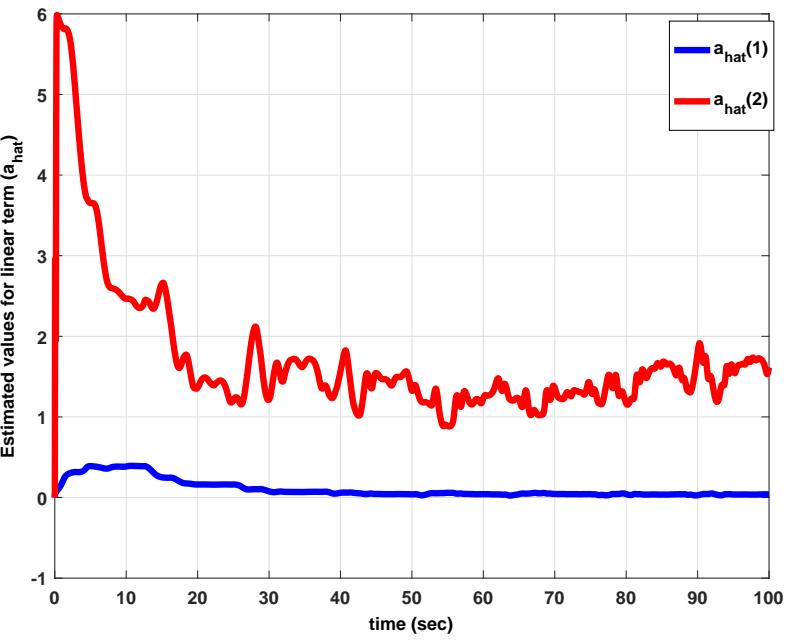


Figure 4.15: Estimated values for unknown linear term (top); and unknown nonlinear term (bottom) , utilizing the AMFC algorithm on a Duffing-Holmes chaotic system. The online estimated parameters are bounded.

4.3 Results for the AMFC algorithm in a single MIMO nonlinear dynamic systems

In this section, first two different nonlinear dynamic systems are used to compare the performance of the proposed AMFC algorithm with the performance of the two state-of-the-art data-driven control algorithms, which are based on RL methods. Here, two cost functions are used to quantify the comparative results. Similar to the previous section, the first cost function is a joint-cost constructed by the tracking error and the control effort, as follows

$$\mathcal{C} = \int_0^{t_f} (e^T e + u^T u) dt , \quad (4.8)$$

where, t_f is final time in the simulation, $e \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$. The second cost function is the integral absolute error $\mathcal{I}_{\mathcal{E}}$, which is defined as

$$\mathcal{I}_{\mathcal{E}} = \int_0^{t_f} E dt , \quad (4.9)$$

where $E = \sum_{i=1}^n |e(i)|$.

4.3.1 Comparison study for an affine nonlinear system

This is the case-1 of the comparative study performed for AMFC algorithm in MIMO systems. For this case, the dynamic system is (Zhu et al., 2016)

$$\begin{aligned} \dot{x}_1 &= -\sin(x_1) + x_2 , \\ \dot{x}_2 &= -x_1^3 + u , \\ y &= x_1 . \end{aligned} \quad (4.10)$$

Here, simulation results from the proposed AMFC are compared with the results from the MFC algorithm proposed by Zhu et al. (2016) (this algorithm is named as MFC-1 in the followings). The case is a tracking problem with the reference signal of $\dot{y}^d = 0.03 \cos 0.3t$. The tuning parameters and the values for the two cost functions are presented in Table 4.4. The values for Q , R , k_1 and k_2 are set trivially, according to *Algorithm-1* in Table 3.1. Based on the guidelines in *Remark 3-4*, some relatively large values are chosen for Γ_1 and Γ_2 , while ρ_1 and ρ_2 are set to be small values. According to Table 4.4, the value of \mathcal{C} is almost the same for AMFC and MFC-1 policies, while the value of $\mathcal{J}_{\mathcal{E}}$ for AMFC is less (about 27 %) than the one for MFC-1. In addition, the number of adaptive laws incorporated in the AMFC is very few in comparison with the number of adaptive laws used in MFC-1. This is an indication for lower computation cost of AMFC algorithm.

Furthermore, the simulation results for this case study are depicted in Fig. 4.16 to Fig. 4.18. As can be seen in Fig. 4.16, the tracking performance of AMFC is more appropriate than the one for the MFC-1 algorithm. In fact, we can say that the settling time in AMFC is lower, which is more preferable. Regarding Fig. 4.17, the AMFC algorithm reacts to the tracking error in a faster way; hence it has a pick in the generated control signal at the beginning of the simulation. Furthermore, as presented in Fig. 4.18, the estimated linear terms by AMFC algorithm are very small, meaning that the estimation and control can be performed just by the utilization of the nonlinear terms, alone. As can be seen in that figure, the estimated values for unknown nonlinear terms are all bounded around origin.

A major disadvantage of MFC-1 algorithm is that it requires huge off-line efforts

to adjust the tuning parameters to achieve acceptable tracking performance. This issue is eliminated by using the AMFC algorithm, where the guidelines for setting the tuning parameters are evidently provided (as in *Remark 3-4*) and the main control gains are updated online (by the equation in (3.49)). Moreover, in the MFC-1 algorithm, one needs to define some regressor parameters, which adds some extra efforts on the implementation process for practical systems. This issue is completely revoked in the proposed AMFC algorithm, as the adaptive laws are regressor-free estimators.

Table 4.4: Properties of the MFC controllers applied to an affine nonlinear system.

Parameter	MFC-1(Zhu et al., 2016) (as in (2.10a) and (2.10b))	proposed AMFC (as in Algorithm 1 in Table 3.1)
Tuning parameters	$\alpha_z = 4e3$ $\gamma_z = 0.01$ $Q_z = 2.3 \times I_4$ $R_z = 1$ $T_0 = 20$	$\Gamma_1 = 1e3 \times \text{diag}(0.01, 1)$ $\Gamma_2 = 1e1 \times \text{diag}(0.01, 1)$ $\rho_1 = \rho_2 = 0.1$ $Q = 0.1 \times I_2$ $R = I_2$ $k_1 = k_2 = 1$
Value of \mathcal{C}	0.05645	0.05717
Value of $\mathcal{J}_{\mathcal{E}}$	0.5468	0.3965
Number of adaptive laws	14	4

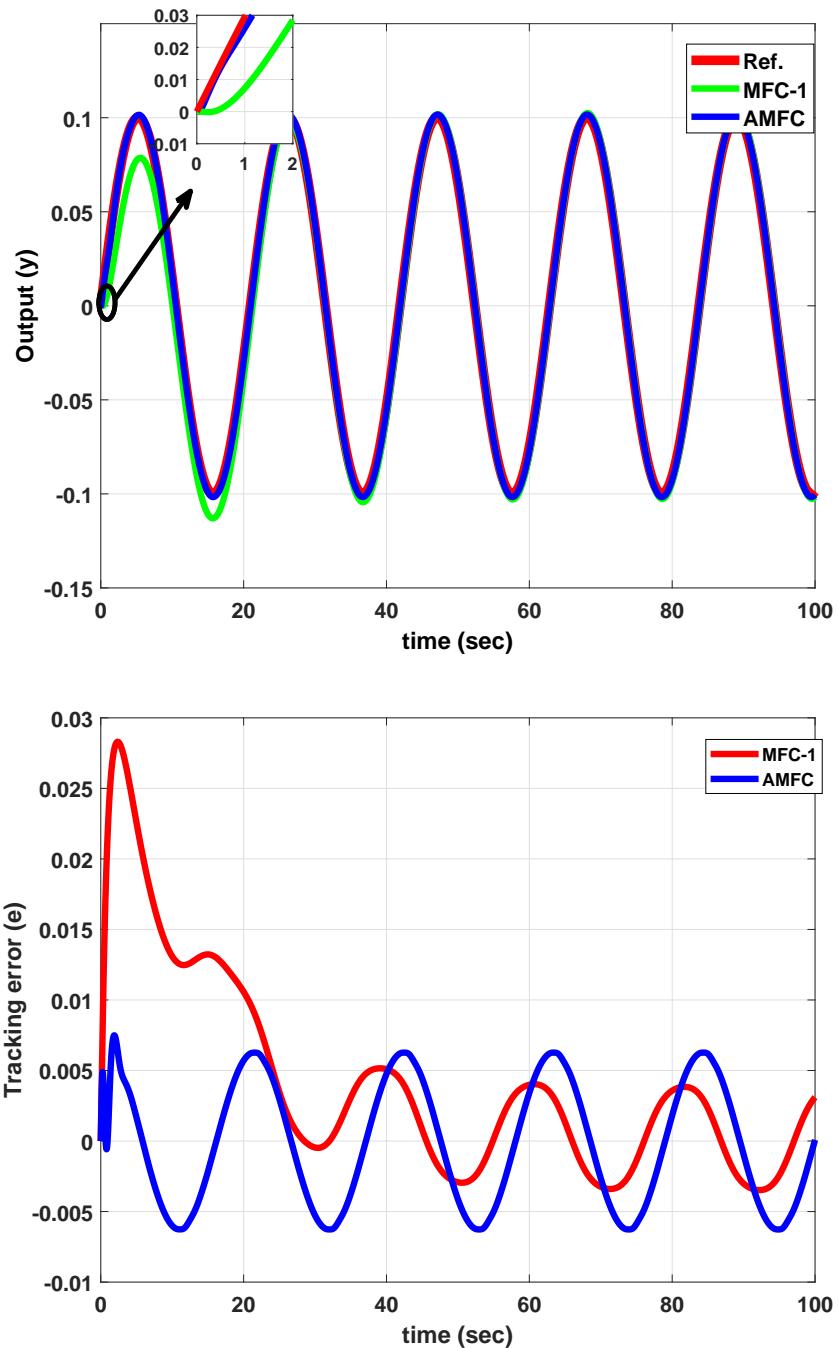


Figure 4.16: Tracking performance of the AMFC and MFC-1 algorithms, implemented on an affine nonlinear system.

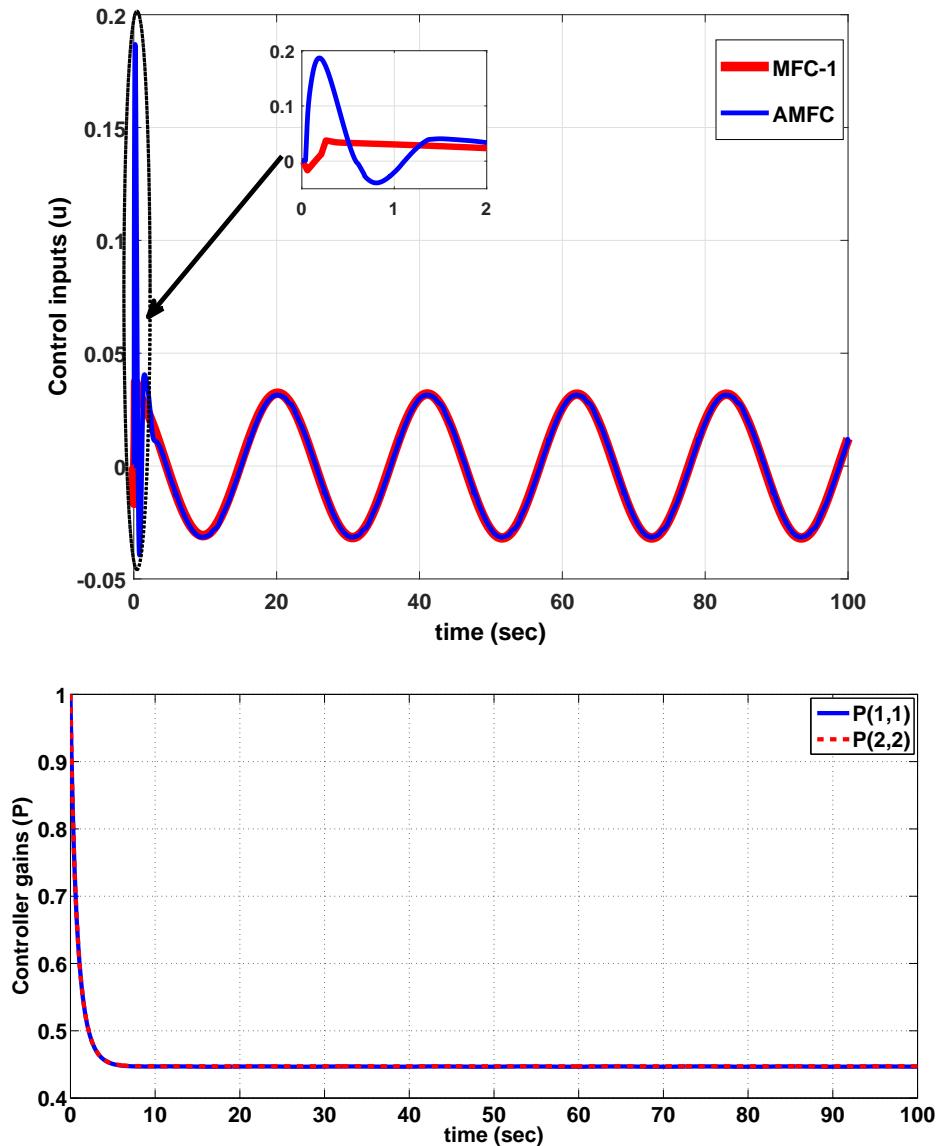


Figure 4.17: The control signals of the AMFC and MFC-1 algorithms, implemented on an affine nonlinear system. The control signals are bounded and the AMFC algorithm has faster reaction to the tracking error, leading to more control efforts at the initial times.

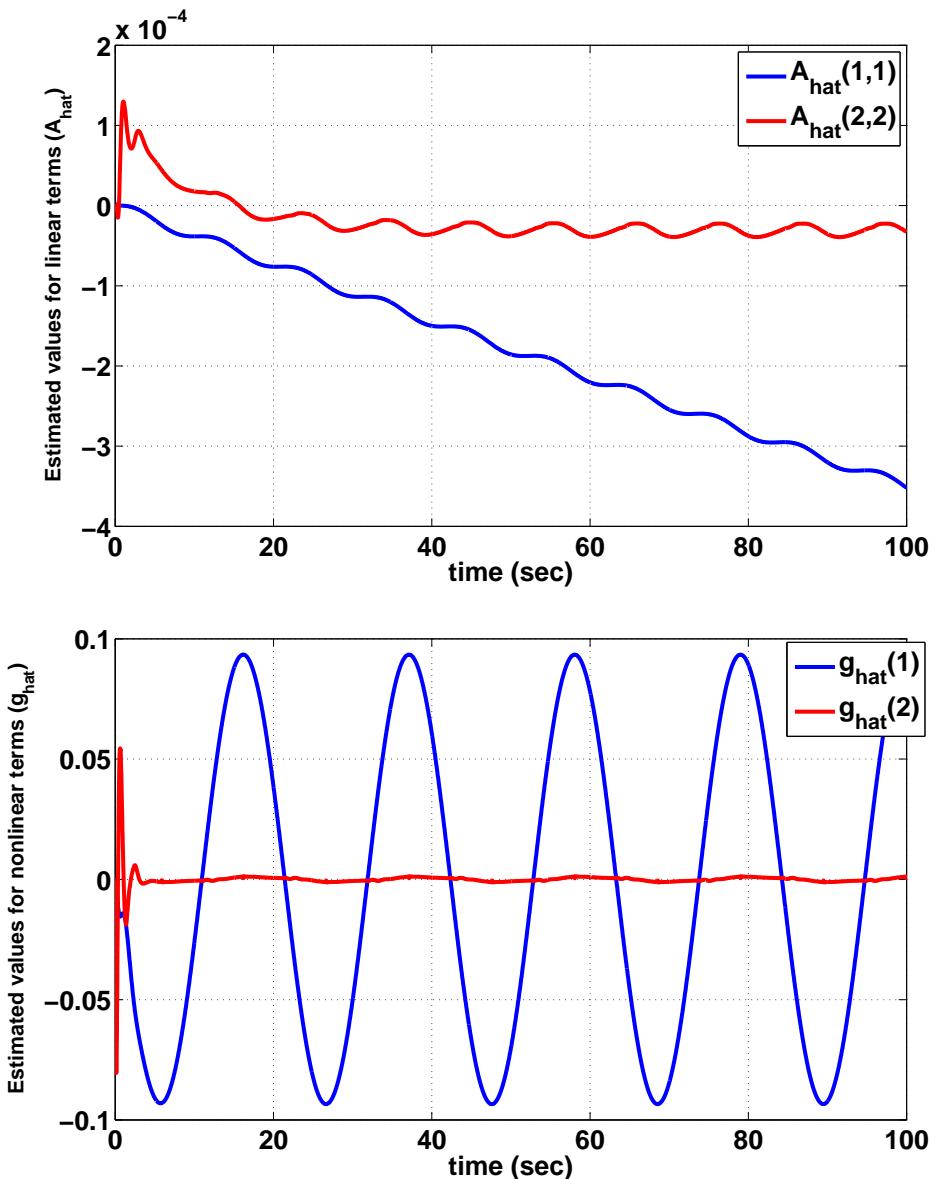


Figure 4.18: Estimated values for the unknown linear and nonlinear terms in an affine nonlinear system, utilizing the AMFC algorithm. All the estimated values are bounded.

4.3.2 Comparison study for a non-affine nonlinear system

This is the case-2 of the comparative studies performed for AMFC algorithm in MIMO systems. Here, the following dynamic system is considered (Vamvoudakis et al., 2014)

$$\begin{aligned}\dot{x}_1 &= -x_1 + x_2 , \\ \dot{x}_2 &= -x_1^3 - x_2 - \frac{x_1^2}{x_2} + 0.25x_2[\cos(2x_1 + x_1^3) + 2]^2u , \\ y &= x_1 .\end{aligned}\tag{4.11}$$

The system is non-affine, since it has a multiplying term of states and the control input. The problem is a regulating problem with the initial value of $x_1 = 1$. Here, the simulation results from AMFC are compared with the ones from a MFC designed by Vamvoudakis et al. (2014), which is named as MFC-2 in the followings. The properties of this comparison are presented in Table 4.5. Similar to the case study in 4.3.1, most of the controller constants for AMFC are tuned trivially according to Table 3.1, except the values for adaptation and leakage gains which are tuned utilizing the guidelines in *Remark 3-4*. Here, the value of \mathcal{C} corresponding to the AMFC algorithm is lower (about 49 %) than the value for MFC-2 policy, while the value of $\mathcal{J}_{\mathcal{E}}$ is substantially lower (about 93 %) for the AMFC, which shows the good tracking performance of the AMFC algorithm.

Moreover, the requirement for the smaller number of adaptive laws in AMFC algorithm indicates the lower computation complexity in this algorithm. The simulation results for this case are depicted in Fig. 4.19 to Fig. 4.21. As it is observed in Fig. 4.19, the tracking performance of AMFC is much better than the one for

MFC-2 algorithm. The error for AMFC case has fewer amount of fluctuations, as well. Regarding Fig. 4.20, the control signal for AMFC algorithm has larger picks at the start of the simulation and converges faster than the control signal for MFC-2 algorithm. Also, the updated controller gains in AMFC are all bounded. Observing Fig. 4.21, one can see that the estimated values for unknown terms are bounded. The estimated nonlinear terms converge to zero very fast, while after that the estimated linear terms drive the controller gain and the system, both.

Table 4.5: Properties of the MFC controllers applied to a non-affine nonlinear system

Parameter	MFC-2(Vamvoudakis & Lewis, 2010) (as in (2.9a) and (2.9b))	proposed AMFC (as in Algorithm 1 in Table 3.1)
Tuning parameters	$a_1 = 0.1$ $a_2 = 0.1$ $F_1 = 1e3 \times [1; 1e2; 1; 1]$ $F_2 = 1e-5$	$\Gamma_1 = 10 \times \text{diag}(0.001, 1)$ $\Gamma_2 = 1 \times \text{diag}(0.001, 1)$ $\rho_1 = 0.1, \rho_2 = 0.01$ $Q = 0.1 \times I_2$ $R = I_2$ $k_1 = k_2 = 1$
Value of \mathcal{C}	22.23	11.4
Value of $\mathcal{I}_{\mathcal{E}}$	13.17	0.9695
Number of adaptive laws	8	4

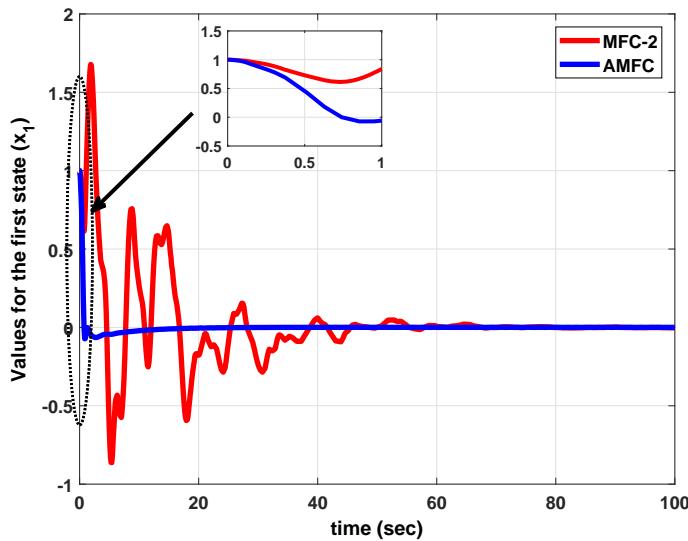


Figure 4.19: Tracking performance of the AMFC and MFC-2 algorithms, implemented on a non-affine nonlinear system. Tracking errors are bounded and faster convergence is achieved with AMFC.

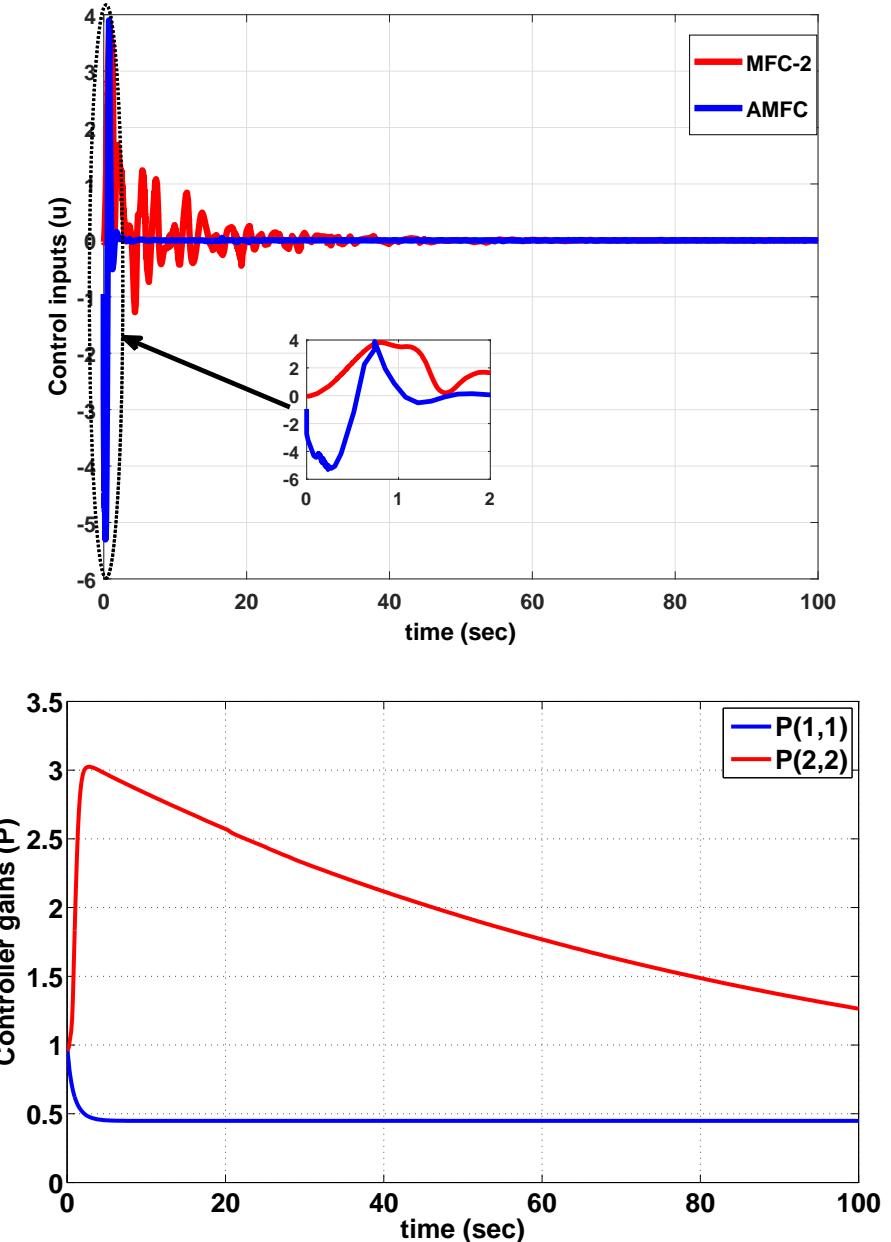


Figure 4.20: The control signals of the AMFC and MFC-2 algorithms, implemented on a non-affine nonlinear system. The values are bounded and the AMFC algorithm has faster reaction to the tracking error.

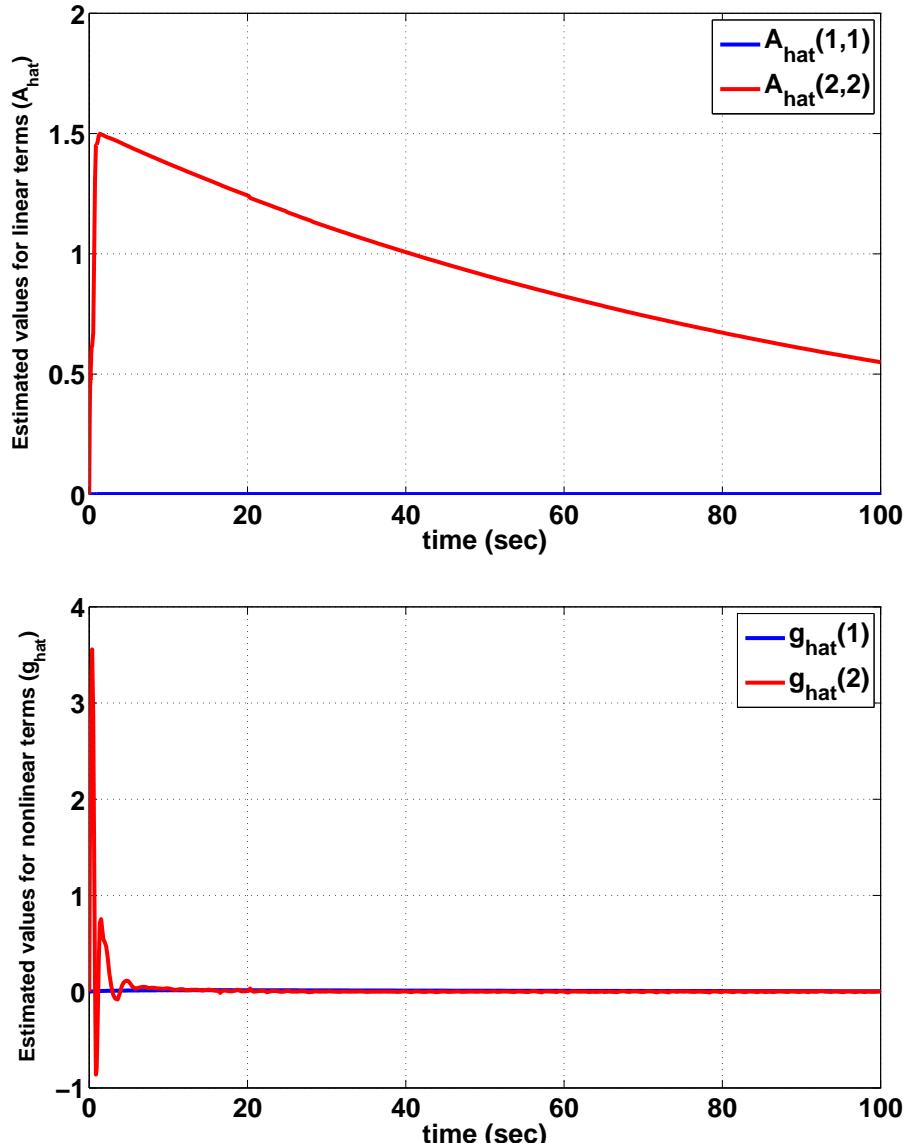


Figure 4.21: Estimated values for unknown linear and nonlinear terms in a non-affine nonlinear system, utilizing the AMFC algorithm. The estimated values are bounded. It can be seen that the estimated linear terms drive the updating process of the main controller gains.

4.3.3 Application of AMFC algorithm to a robotic manipulator

In this section, the AMFC algorithm is implemented on a two degree-of-freedom (DOF) planar robotic manipulator. The equations for motion of a 2-DOF robotic manipulator in a 2D space is considered as following (Spong et al., 2006)

$$\begin{aligned} & \left[\frac{1}{4}m_1l_1^2 + j_1 + m_2(l_1^2 + \frac{1}{4}l_2^2 + l_1l_2 \cos \theta_2) + j_2 \right] \ddot{\theta}_1 + \left[m_2(\frac{1}{4}l_2^2 + \frac{1}{2}l_1l_2 \cos \theta_2) + j_2 \right] \ddot{\theta}_2 - \\ & \frac{1}{2}m_2l_1l_2 \sin \theta_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) + \frac{1}{2}m_1g_e l_1 \cos \theta_1 + m_2g_e(l_1 \cos \theta_1 + \frac{1}{2}\cos(\theta_1 + \theta_2)) = \tau_1 \\ & \left[m_2(\frac{1}{4}l_2^2 + \frac{1}{2}l_1l_2 \cos \theta_2) + j_2 \right] \ddot{\theta}_1 + \left(\frac{1}{4}m_2l_2^2 + j_2 \right) \ddot{\theta}_2 + \frac{1}{2}m_2l_1l_2 \sin \theta_2 \dot{\theta}_1^2 + \\ & \frac{1}{2}m_2g_e l_2 \cos(\theta_1 + \theta_2) = \tau_2 \quad (4.12) \end{aligned}$$

where, $m_i = 1$ Kg, $l_i = 1$ m, $j_i = 0.1$ Kg.m² and τ_i for $i = \{1, 2\}$ are the mass, length, moment of inertia and external driving torque associated to each arm and g_e is the gravity acceleration. Moreover, θ_1 and θ_2 are the two degrees of freedom for the system as shown in Fig. 4.22.

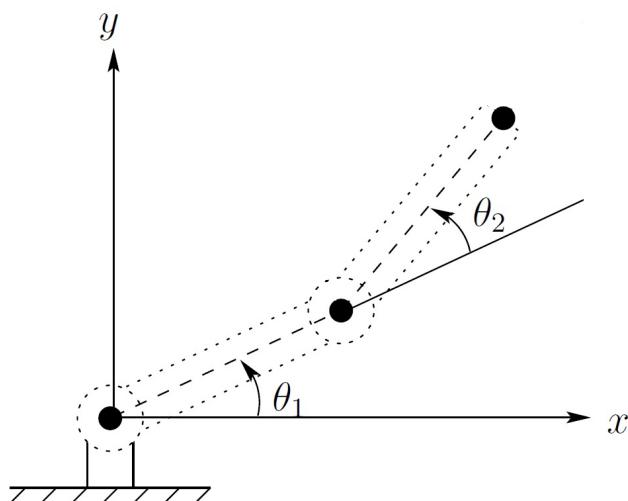


Figure 4.22: Schematic of a 2 DOF robotic manipulator in the 2D space (Spong et al., 2006)

In order to implement the AMFC algorithm for the tracking problem on the system presented in (4.12), the coupling terms are incorporated into the unknown nonlinear terms, which are going to be estimated online. Actually by considering *Definition 3-3*, the dynamic system in (4.12) can be represented as follows

$$\begin{aligned}\tau_1 &= b_1^r \ddot{\theta}_1 + f_1 \\ \tau_2 &= b_2^r \ddot{\theta}_2 + f_2,\end{aligned}\tag{4.13}$$

where $b_1^r = b_2^r = 1 \text{ Kg.m}^2$ and

$$\begin{aligned}f_1 &= [m_2(\frac{1}{4}l_2^2 + \frac{1}{2}l_1l_2 \cos \theta_2) + j_2]\ddot{\theta}_2 - \frac{1}{2}m_2l_1l_2 \sin \theta_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) + q_1^r(\theta_1, \theta_2) \\ f_2 &= [m_2(\frac{1}{4}l_2^2 + \frac{1}{2}l_1l_2 \cos \theta_2) + j_2]\ddot{\theta}_1 + \frac{1}{2}m_2l_1l_2 \sin \theta_2 \dot{\theta}_1^2 + \\ &\quad \frac{1}{2}m_2g_e l_2 \cos(\theta_1 + \theta_2) + q_2^r(\theta_1, \theta_2)\end{aligned}\tag{4.14}$$

and q_1^r and q_2^r are two unknown nonlinear functions depending only on θ_1 and θ_2 . Here, it is assumed that the terms f_1 and f_2 are completely unknown, i.e. they have unknown structure and unknown parameters. According to *Definition 3-5*, two virtual controllers (i.e. u_1^* and u_2^*) are defined in order to have the input gain matrix B in a full-rank form. While these virtual controllers define the desired values to be tracked by the 3rd and 4th system states, the 1st and 2nd system states should track the desired trajectory defined as the inputs (i.e. θ_1^d and θ_2^d). In this regard, the dynamic system of a 2 DOF manipulator presented in (4.12) can be presented in form of (3.42a) as follows

$$\dot{x}_r = A_r x_r + B_r u_r + g_r,\tag{4.15}$$

where $x_r = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^T$ is the system states, $A_r \in \mathbb{R}^{4 \times 4}$ is the unknown diagonal system matrix, $B_r = I_4 \in \mathbb{R}^{4 \times 4}$ is the inputs gain matrix, $u_r = [u_1^*, u_2^*, \tau_1, \tau_2]^T$ is the vector of control inputs, $g_r = [g_1, g_2, g_3, g_4]^T$ is the vector of unknown nonlinearities, all in the dynamic system of a 2DOF robotic manipulator. Note that the unknown dynamics f_1 and f_2 are included in $g_3, A(3,3)$ and $g_4, A(4,4)$, respectively. Here, the vector of desired trajectory including the virtual control inputs as the set-points of the angular rates is defined as $y^d = [\theta_1^d, \theta_2^d, u_1^*, u_2^*]^T$; while the actual demand trajectory to be tracked by the angular positions is presented by $y_0^d = [\theta_1^d; \theta_2^d]$. Furthermore, the actual control inputs sent to the 2 DOF planar manipulator are generated as $u_0 = [\tau_1; \tau_2]$. Here, $m_1 = 2 \text{ Kg}$, $m_2 = 1 \text{ Kg}$, $l_1 = l_2 = 1 \text{ m}$, $j_1 = j_2 = 0.1 \text{ Kg.m}^2$ and $g_e = 9.81 \text{ m/s}^2$ are used.

The simulation results for this case study are reported in Fig. 4.23 to Fig. 4.25. Regarding Fig. 4.23, tracking objective is satisfied obviously for the reference trajectory as $\theta_1^d = (\pi/2)$ and $\theta_2^d = (\pi/4) \times \sin t$. The tracking error is bounded around origin with small boundaries. According to Fig. 4.24, the estimated values for unknown terms are bounded and it can be seen that they are driving the control signals in AMFC algorithm. Looking at Fig. 4.25, one can see that the control signal is bounded as well and the main controller gains are updating online, following the online estimations on the unknown linear terms. The tuning parameters for this simulation are presented in Table 4.6. In this table, I_4 means an identity matrix in $\mathbb{R}^{4 \times 4}$. Similar to the previous cases, the adaptive and leakage gains are chosen according to *Remark 3-4*. In total, eight adaptive laws (including 4 adaptation on unknown linear terms and 4 adaptation on unknown nonlinear terms) are used in the AMFC algorithm implemented on a 2 DOF robotic manipulator.

Table 4.6: Tuning parameters of the AMFC algorithm applied to the robotic manipulator.

Tuning parameters and the corresponding values (as in Algorithm 1 in Table 3.1)
$\Gamma_1 = 100 \times \text{diag}(0.001, 0.001, 1, 1]$
$\Gamma_2 = 10 \times \text{diag}(0.001, 0.001, 1, 1)$
$\rho_1 = 0.001$
$\rho_2 = 0.001$
$Q = 0.1 \times I_4$
$R = I_4$
$k_1 = k_2 = 1$

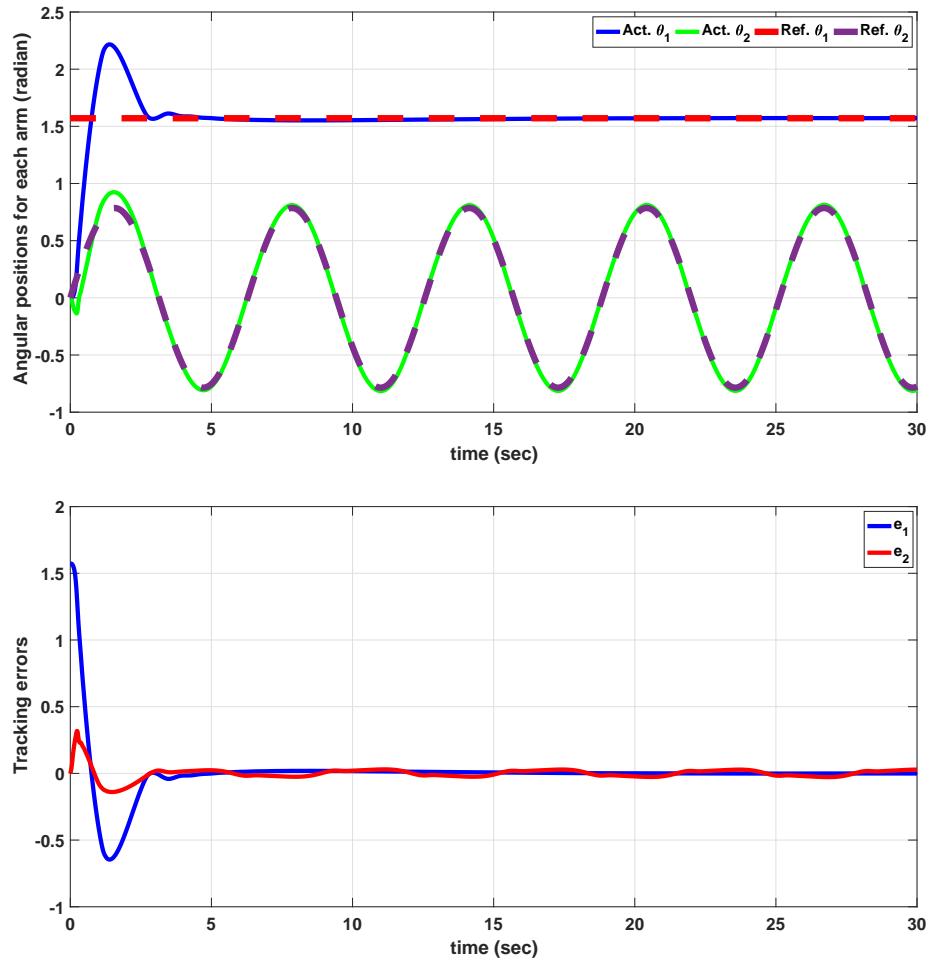


Figure 4.23: Values for the states (top) and the tracking error (bottom) for the robotic manipulator, utilizing the AMFC algorithm. The tracking error is bounded around origin with very small boundaries.

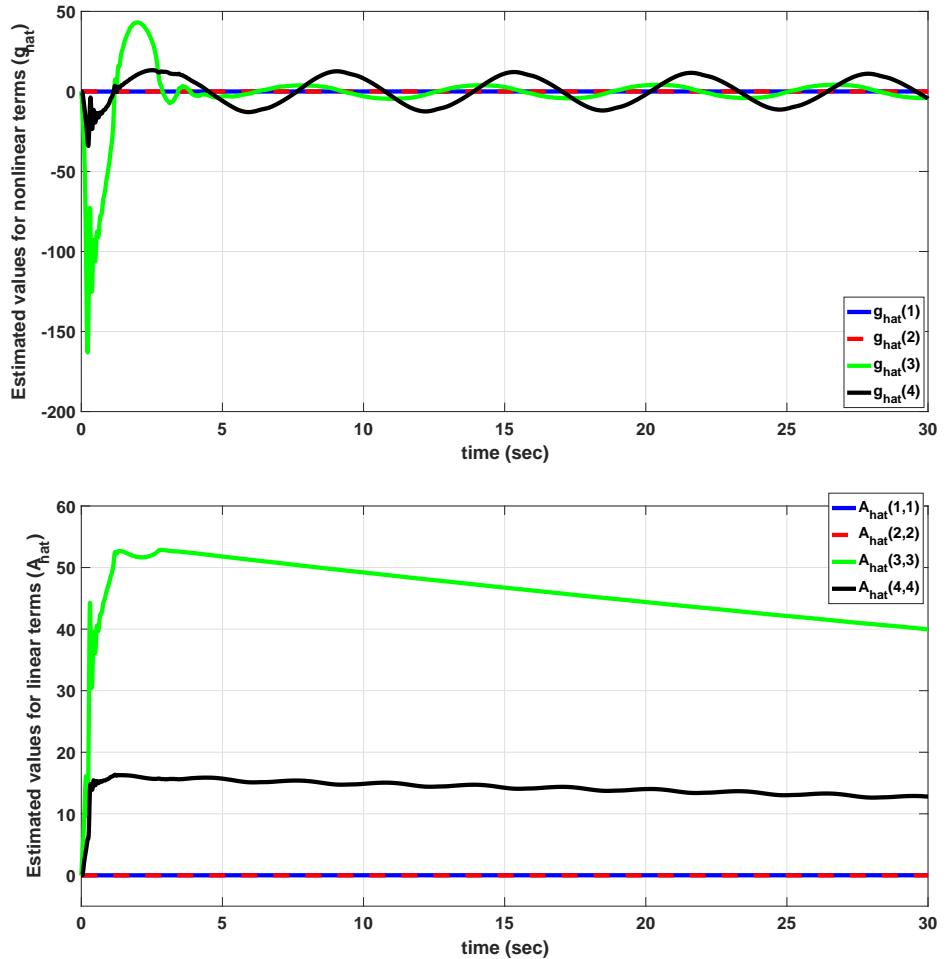


Figure 4.24: Estimated values for unknown nonlinear (top) and unknown linear (bottom) terms for the robotic manipulator, utilizing the AMFC algorithm. The estimate values for unknown terms are bounded.

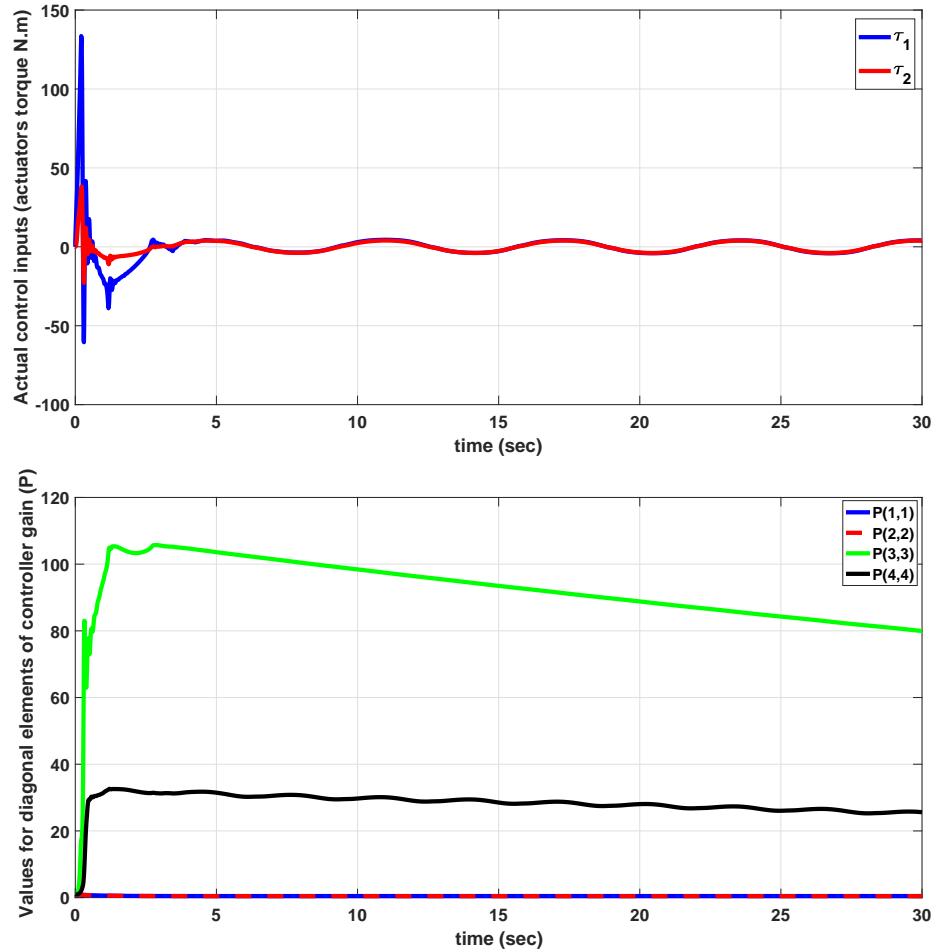


Figure 4.25: Values for the control signals (top); and the Controller gains, i.e. P matrix (bottom) for the robotic manipulator, utilizing the AMFC algorithm. The control signal is bounded and values for the controller gains are updated based on the estimated values of unknown linear terms.

4.3.4 Application of AMFC algorithm to a wheeled mobile robot

A WMR is a ground mobile robot which uses the differential drive for moving in a 2D environment. The mathematical model for kinematics and dynamics of a WMR (Fig. 4.26) can be considered as follows (Arab & Fateh, 2015; Asif et al., 2016)

$$\dot{x}_w = v \cos(\theta_w), \quad (4.16a)$$

$$\dot{y}_w = v \sin(\theta_w), \quad (4.16b)$$

$$\dot{\theta} = w, \quad (4.16c)$$

$$\dot{v} = \frac{1}{m_w} [F_w - k_F v + f_w], \quad (4.16d)$$

$$\dot{w} = \frac{1}{j_w} [\tau_w - k_\tau w + t_w]. \quad (4.16e)$$

Here, the position of WMR in X-Y plane and its pose direction are defined by $[x_w, y_w, \theta_w]^T$. The linear and angular velocities of the WMR are presented by v and w , respectively. m_w and j_w are mass and inertia of the WMR. The friction force and torque are modeled using k_F and k_τ coefficients and the unknown disturbance force and torque on the WMR are presented by f_w and t_w , respectively. In addition, force and torque generated by the set of electric motors located on the WMR are defined as

$$F_w = c_f (\bar{f}_1 + \bar{f}_2), \quad (4.17a)$$

$$\tau_w = c_\tau (\bar{f}_1 - \bar{f}_2), \quad (4.17b)$$

where \bar{f}_1 and \bar{f}_2 are two transformed control inputs to be defined later in the following, and c_f and c_τ are two mapping constants between the angular speed of wheels and the generated force and torque on the WMR rigid-body diagram. Further discussions on

the modeling process of electric motors can be found in (Aung, 2007; Nouri, 2005).

Here, in order to perform the simulation, the values for parameters in the WMR model are set as $m_w = 1 \text{ Kg}$, $j_w = 0.1 \text{ Kg.m}^2$, $k_F = 0.1 \text{ Kg/s}$, $k_\tau = 0.1 \text{ Kg.m}^2/\text{s}$, $k_e = 0.1 \text{ Kg.m}^2/(\text{rad.s})$, $r_w = l_w = 0.1 \text{ m}$, $f_w = \sin t \text{ N}$ and $t_w = \sin t \text{ N.m}$. In addition, the desired trajectory is a square in the 2D space with the length of 10m for each edge.

In order to implement the AMFC algorithm on the WMR model in (4.16a) to (4.16e), a technique is adopted from the human attitude to reach the destination point from the current pose of the WMR. When a human wants to move from point $P_w^1 = [x_w^1; y_w^1]$ to point $P_w^2 = [x_w^2; y_w^2]$, first he turns toward the right direction and then moves in that direction in order to reach the destination (Nguyen & Le, 2016). By adopting this idea, one can split the tracking problem of a WMR into a steering problem and a driving velocity control problem (Koh & Cho, 1999). In this regard, the desired direction for WMR is defined based on its desired destination (i.e. $[x_w^d; y_w^d]$) and its current position (i.e. $[x_w; y_w]$), as

$$\theta_w^d = \tan^{-1}\left(\frac{y_w^d - y_w}{x_w^d - x_w}\right) , \quad -\frac{\pi}{2} < \theta_w^d \leq \frac{\pi}{2} . \quad (4.18)$$

When the correct direction is targeted, the WMR should move in that direction to reach the destination. Hence, a new position variable in the current direction of the robot is defined as

$$z_w = x_w \cos \theta_w + y_w \sin \theta_w , \quad (4.19)$$

where the desired value for this variable is defined as follows

$$z_w^d = x_w^d \cos \theta_w + y_w^d \sin \theta_w . \quad (4.20)$$

Then, the new formulation of WMR tracking problem can be defined as

$$\dot{z}_w = v + w(-x_w \sin \theta_w + y_w \cos \theta_w) , \quad (4.21a)$$

$$\dot{\theta}_w = w , \quad (4.21b)$$

$$\dot{v} = \frac{1}{m_w} [F_w - k_F v + f_w] , \quad (4.21c)$$

$$\dot{w} = \frac{1}{j_w} [\tau_w - k_\tau w + t_w] . \quad (4.21d)$$

Since this equation is in form of

$$\dot{x}_w = A_w x_w + B_w u_w + g_w , \quad (4.22)$$

the AMFC can be implemented directly on the problem, without any need to know the values of parameters in the dynamic model. In (4.22), $x_w = [z_w, \theta_w, v, w]^T$ is the system states, $A_w \in \mathbb{R}^{4 \times 4}$ is the unknown diagonal system matrix, $B_w = I_4 \in \mathbb{R}^{4 \times 4}$ is the inputs gain matrix, $u_w = [u_1^*, u_2^*, u_1, u_2]^T$ is the vector of control inputs including the virtual control variables and $g_w = [g_1, g_2, g_3, g_4]^T$ is the vector of unknown nonlinear terms, all in the dynamic system of a WMR. Similar to the case in previous section, the vector of desired trajectory is defined as $y^d = [z_w^d, \theta_w^d, u_1^*, u_2^*]^T$, which includes the virtual control variables as the set-points for the translational and angular speeds. The unknown dynamics in (4.21a) to (4.21d) are included in g_w and A_w . Note that the values of u_1 and u_2 are not exactly equal to the values of F_w and τ_w , instead they are indirectly equal utilizing the transformed control inputs. By using the AMFC policy, the values of u_1 and u_2 are determined. Then, the transformed control inputs are generated as

follows

$$\bar{f}_1 = \frac{1}{2}(u_1 + u_2) , \quad (4.23a)$$

$$\bar{f}_2 = \frac{1}{2}(u_1 - u_2) . \quad (4.23b)$$

By replacing \bar{f}_1 and \bar{f}_2 from (4.23a) and (4.23b) in (4.17a) and (4.17b), the dynamics of WMR can be controlled. The values of \bar{f}_1 and \bar{f}_2 are used to define the desired set-points for rotational speed of the WMR electric motors (ω_1 and ω_2), which in turn are controlled by dc motor speed controllers at a lower level loop.

The values for tuning parameters to implement AMFC on the WMR model are presented in Table 4.7. Here, $\mathbf{1}_2$ is a vector in $\mathbb{R}^{2 \times 1}$ with all elements are equal to one. The adaptive and leakages are tuned recalling *Remark 3-4*, while other parameters are set trivially. The simulation results for this case is depicted in Fig. 4.27 to Fig. 4.29. Observing Fig. 4.27, appropriate tracking performance is achieved for a square path in the 2D space. The tracking errors are bounded. Although there are relatively large overshoots at the corners of the square path, these overshoots are damped very fast. In addition, by more accurate tunings of the adaptive and leakage gains, the overshoot values would be decreased. In Fig. 4.28, it is shown that the control signals are bounded around origin, while the main controller gain is updated and reached to a finite stable value at each corner of the desired square path. The boundedness and convergence of the estimated values for unknown linear and nonlinear terms can be seen in Fig. 4.29. It is deduced that the controller gain is updated online based on the estimated value of the unknown linear term. Since the WMR has 4 internal states, there are 8 adaptive laws used in the implemented AMFC, including four adaptive laws for \hat{A} and the other four for adaptation of \hat{g} .

Table 4.7: Tuning parameters of the AMFC used for the autonomous WMR

Tuning parameters and the corresponding values (as in Algorithm 1 in Table 3.1)
$\Gamma_1 = 100 \times \text{diag}(0.001 \times \mathbf{1}_2, \mathbf{1}_2)$
$\Gamma_2 = 0.1 \times \text{diag}(0.001 \times \mathbf{1}_2, \mathbf{1}_2)$
$\rho_1 = 0.001$
$\rho_2 = 0.0001$
$Q = 0.1 \times I_4$
$R = I_4$
$k_1 = k_2 = 1$

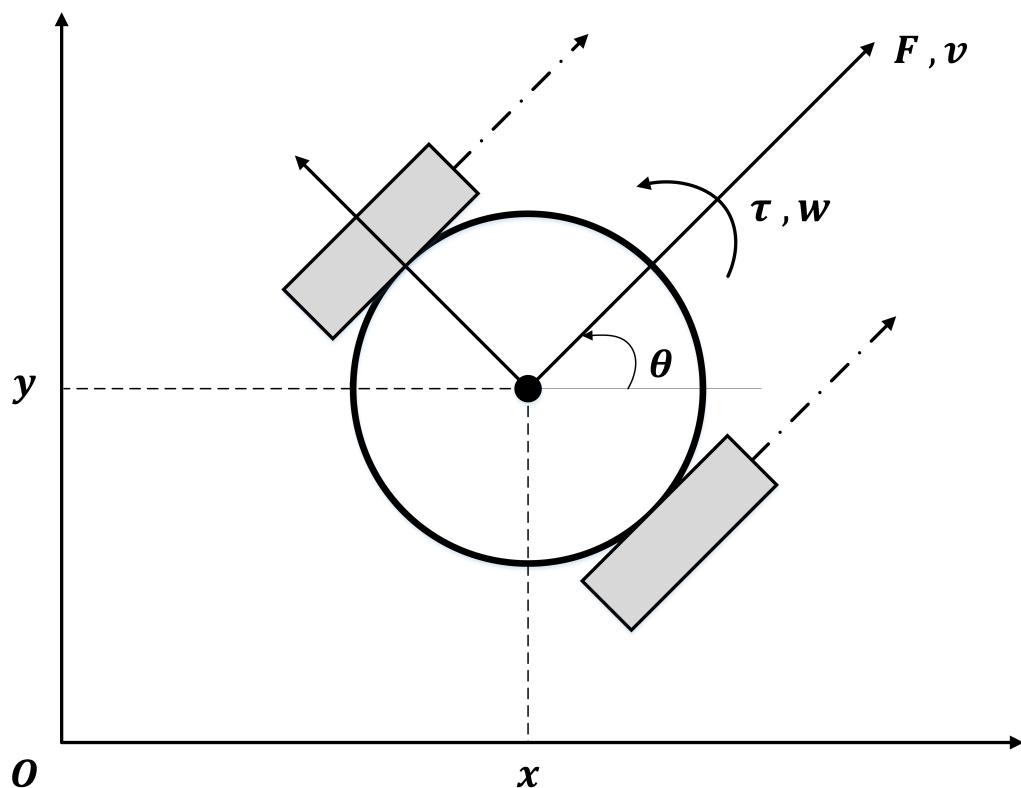


Figure 4.26: Schematic of a WMR

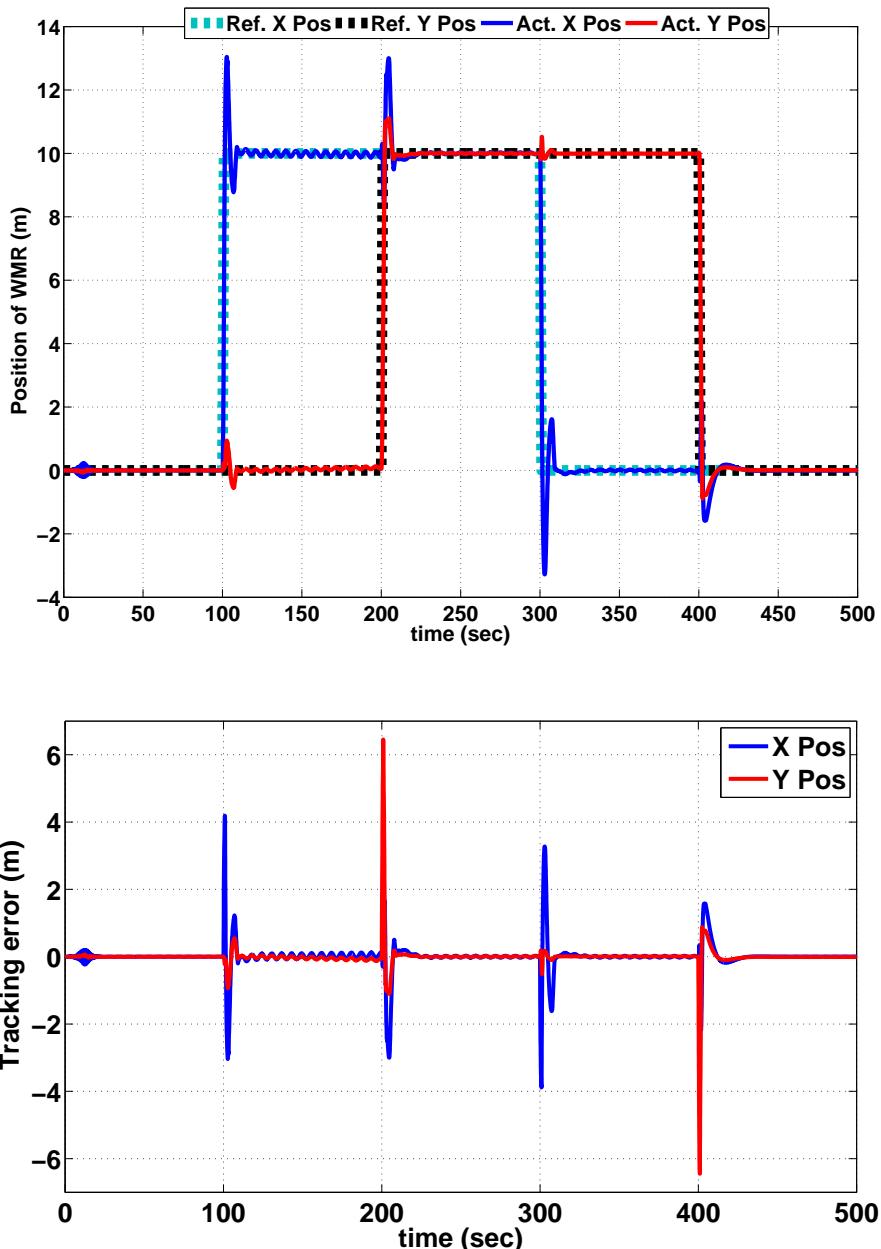


Figure 4.27: Tracking performance for the autonomous WMR by applying the AMFC algorithm. The tracking error is bounded around origin.

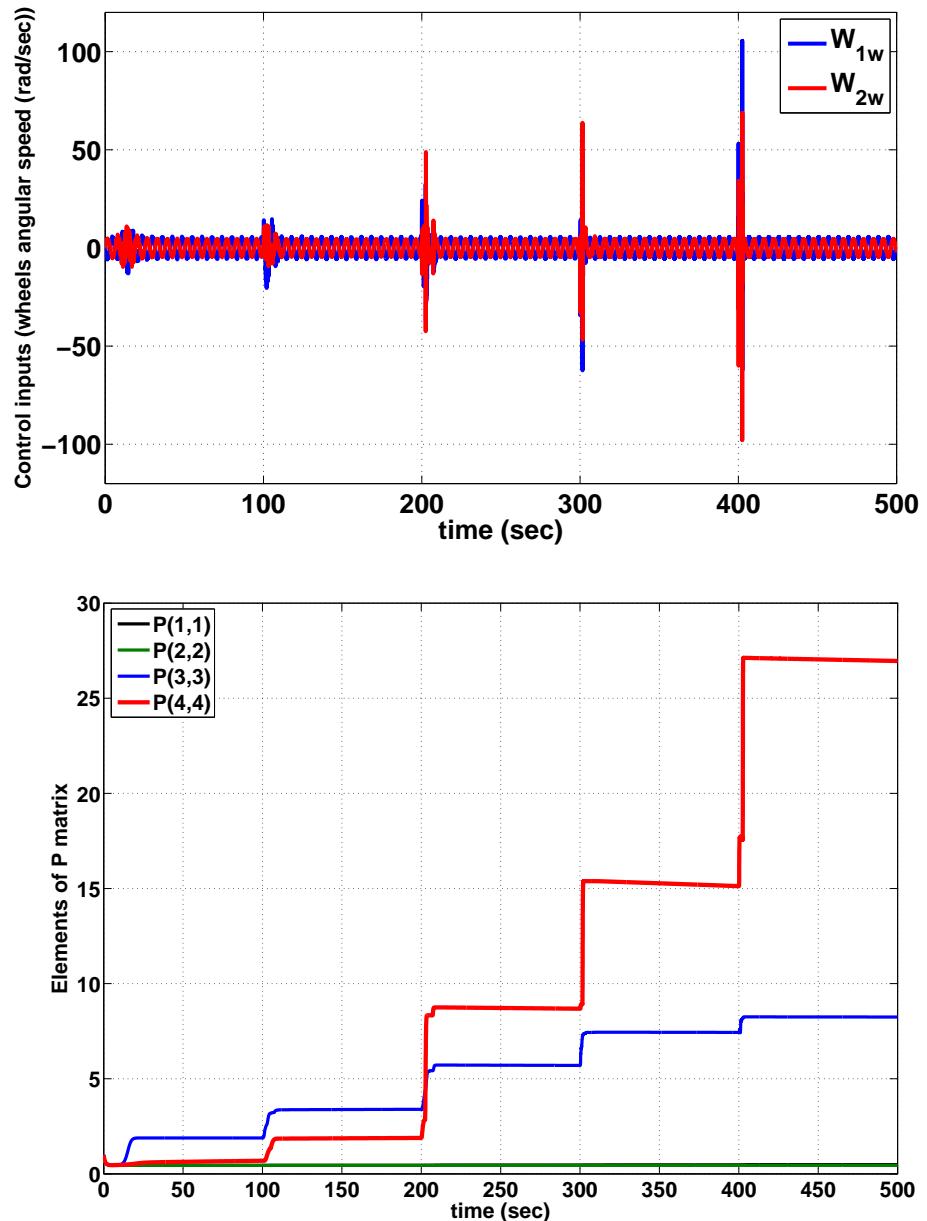


Figure 4.28: Control inputs (angular speeds of wheels) for the autonomous WMR and the values for P matrix, utilizing the AMFC algorithm. The control signals and the controller gains are bounded.

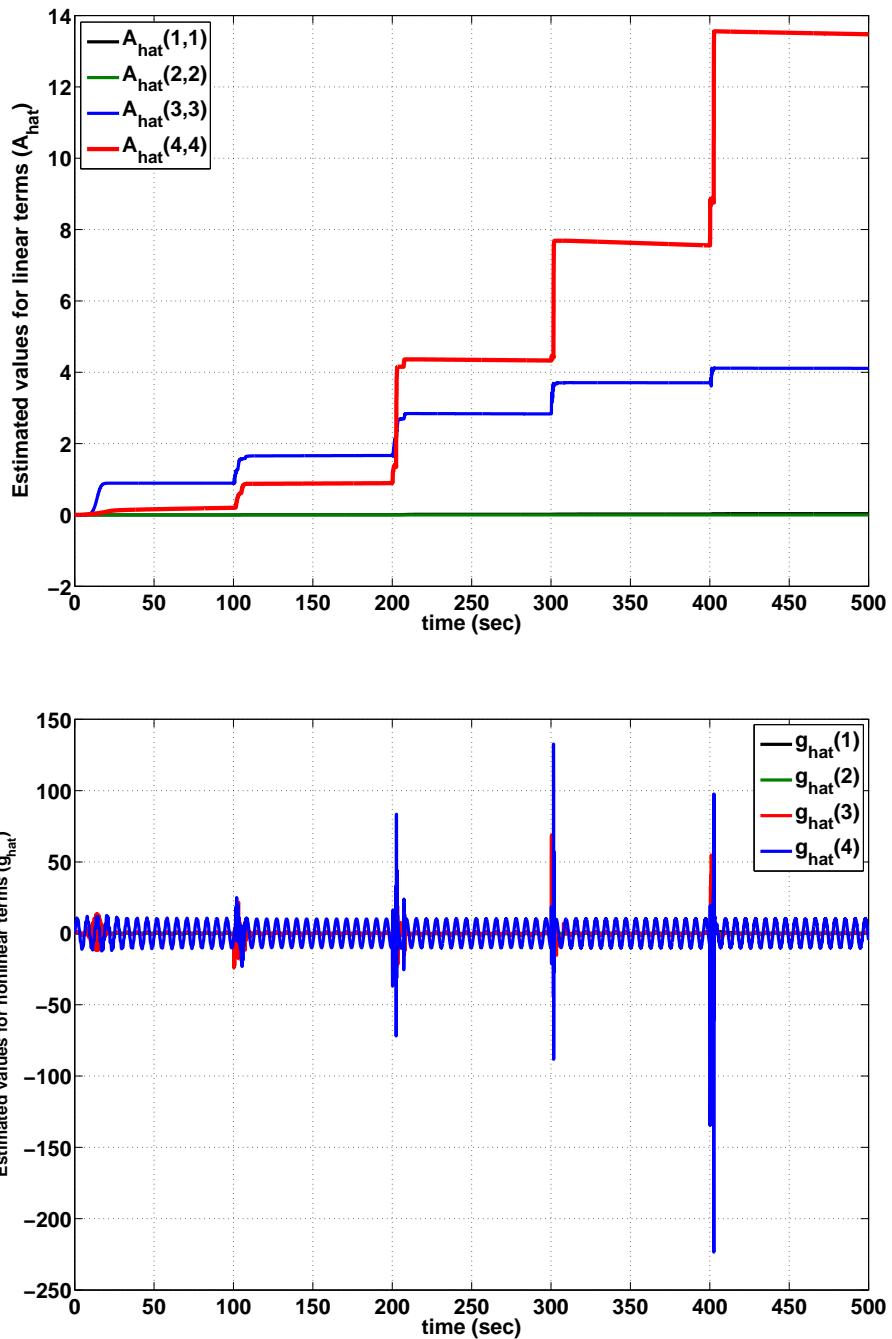


Figure 4.29: Estimated values for the unknown linear and nonlinear terms for the autonomous WMR, utilizing the AMFC algorithm. The estimated values for unknown terms are bounded.

4.3.5 Application of AMFC algorithm to a quadrotor

A quadrotor is a flying robot with four vertically located propellers and an under-actuated dynamic system, meaning that the number of control inputs (i.e. the rotational speeds of the four electric motors) are lower than the number of system outputs (which are six variables including three position variables and three angles). The model considered for simulation of quadrotor is as follows (Boudjedir et al., 2014)

$$\dot{\vec{p}}_q = \vec{v}_q , \quad (4.24a)$$

$$\dot{\vec{\Phi}}_q = R_{qt}^{-1} \vec{w}_q , \quad (4.24b)$$

$$\dot{\vec{v}}_q = \frac{1}{m_q} [R_q \vec{F}_q - K_d \vec{v}_q - m_q \vec{F}_g + \vec{f}_q] , \quad (4.24c)$$

$$\dot{\vec{w}}_q = j_q^{-1} [\vec{\tau}_q - K_a \vec{w}_q - \vec{w}_q \times j_q \vec{w}_q + \vec{t}_q] , \quad (4.24d)$$

where $\vec{p}_q = [x_q; y_q; z_q]$ and $\vec{\Phi}_q = [\phi_q; \theta_q; \psi_q]$ are the absolute positions and Euler angles (roll, pitch and yaw) of the quadrotor, respectively. Also, \vec{v}_q and \vec{w}_q are the vectors of linear and angular velocities. m_q is the quadrotor mass and $j_q \in \mathbb{R}^{3 \times 3}$ is its inertia matrix. k_d and k_a are two constants coefficients for the friction forces and torques. $\vec{F}_g = [0; 0; g_e]$ is the vector of gravity force and \vec{f}_q and \vec{t}_q are the vectors of unknown disturbances. The generated force and torques by the electric motors are represented by $\vec{F}_q = [0; 0; F_T]$ and $\vec{\tau}_q = [\tau_x; \tau_y; \tau_z]$, where (Boudjedir et al., 2014)

$$\begin{bmatrix} F_T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} k_l & k_l & k_l & k_l \\ -k_l l_q & 0 & k_l l_q & 0 \\ 0 & -k_l l_q & 0 & k_l l_q \\ k_t & -k_t & k_t & -k_t \end{bmatrix} \begin{bmatrix} w_{1q}^2 \\ w_{2q}^2 \\ w_{3q}^2 \\ w_{4q}^2 \end{bmatrix} . \quad (4.25)$$

Here, k_l and k_t are the constants of each electric motor for generating lift force and torque, respectively. l_q is the length for each arm of the quadrotor and w_{iq} for $i = \{1, 2, 3, 4\}$ are the angular speeds for the i th electric motor. In the quadrotor model, there are two matrices to transform the coordinates as follows (refer to Fig. 4.30)

$$R_q = \begin{bmatrix} C\psi_q.C\theta_q & C\psi_q.S\theta_q.S\phi_q - S\psi_q.C\phi_q & C\psi_q.S\theta_q.C\phi_q + S\psi_q.S\phi_q \\ S\psi_q.C\theta_q & S\psi_q.S\theta_q.S\phi_q + C\psi_q.C\phi_q & S\psi_q.S\theta_q.C\phi_q - C\psi_q.S\phi_q \\ -S\theta_q & C\theta_q.S\phi_q & C\theta_q.C\phi_q \end{bmatrix} \quad (4.26)$$

and

$$R_{qt} = \begin{bmatrix} 1 & 0 & -S\theta_q \\ 0 & C\phi_q & C\theta_q.S\phi_q \\ 0 & -S\phi_q & C\theta_q.C\phi_q \end{bmatrix}, \quad (4.27)$$

where S and C are for sine and cosine functions, respectively. The values of parameters for this simulation study are $l_q = 0.1$ m, $m_q = 2$ Kg, $j_q = (1.24e - 3) \times \text{diag}(1, 1, 2)$ Kg.m², $k_d = 0.01$ Kg/s, $k_a = 0.01$ Kg.m²/s, $k_l = (1e - 5)$ Kg.m, $k_t = (1e - 7)$ Kg.m², $g_e = 9.81$ m/s², $\vec{f}_q = \sin t \vec{v}_1$ N and $\vec{t}_q = \sin t \vec{v}_3$ N.m, where $\vec{v}_1 = [0; 0; 1]$ and $\vec{v}_3 = [1; 1; 1]$. Also, here the desired trajectory is a helix with radius of 10m which goes up in the Z direction by the rate of 0.1 m/s.

The dynamic system in (4.24a) to (4.24d), has 12 system states as $[\vec{p}_q; \vec{\Phi}_q; \vec{v}_q; \vec{w}_q]$ and 4 actual control inputs as $[u_3; u_4; u_5; u_6]$ corresponding to $[F_T; \tau_q]$. Hence, another six virtual control inputs as $[u_i^*]$ for $i = 1 : 6$ and two semi-virtual control inputs as $[u_1; u_2]$ should be considered for implementation of the AMFC algorithm to the quadrotor model. The semi-virtual control inputs are going to be used to define the desired values of the euler angles, as it is declared later in this section. When the values for control inputs are determined using the AMFC policy, the values for angular speeds of electric motors are defined using (4.25), where (Boudjedir et al.,

2014)

$$F_T = \sqrt{u_1^2 + u_2^2 + u_3^2}, \quad (4.28a)$$

$$\tau_x = u_4, \quad (4.28b)$$

$$\tau_y = u_5, \quad (4.28c)$$

$$\tau_z = u_6. \quad (4.28d)$$

One issue regarding the AMFC implementation on quadrotor is about the desired values of Euler angles. While the desired values for absolute positions of quadrotor are available from the reference trajectory to be tracked, the desired values of Euler angles should be determined internally by the control algorithm. In this exemplified system, yaw angle $\psi_q^d = 0$ is assumed. The desired values for roll and pitch angles are also defined as (Boudjedir et al., 2014)

$$\phi_{dq} = \sin^{-1}\left(\frac{\sin \psi_{dq} u_1 - \cos \psi_{dq} u_2}{F_T}\right), \quad (4.29a)$$

$$\theta_{dq} = \tan^{-1}\left(\frac{\cos \psi_{dq} u_1 + \sin \psi_{dq} u_2}{u_3}\right). \quad (4.29b)$$

The values of constant parameters for AMFC implemented on the quadrotor are provided in Table 4.8. Here, $\mathbf{1}_6$ is a vector in $\mathbb{R}^{6 \times 1}$ with all elements are equal to one. Similar to the case of the robotic manipulator and the WMR, the values of Q , R , k_1 and k_2 are chosen trivially, while the parameters Γ_1 , Γ_2 , ρ_1 and ρ_2 are tuned based on *Remark 3-4*. According to *Remark 3-7*, the number of adaptive laws in the AMFC algorithm in this case is 24, corresponding to the 12 system states.

The simulation results for evaluating the performance of AMFC on the quadrotor model are presented in Fig. 4.31 to Fig. 4.33. The trajectory tracking objective in the

3D space is satisfied appropriately, as shown in Fig. 4.31. As it is confirmed in this figure, the tracking errors are all bounded around zero. Regarding Fig. 4.32, the control signals are bounded around a non-zero value, which is the required control effort for the quadrotor hovering in the air. Furthermore, the controller gains are updated online and are confirmed to be bounded. Note that, the element in the P matrix which is associated with the earth gravity has a relatively larger value. The online adapted values for unknown linear and nonlinear terms are presented in Fig. 4.33. It is confirmed that all of the estimations are bounded and reach some stable points. Note that the values for parameters $\hat{A}(9, 9)$, $\hat{g}(9)$, $\hat{P}(9, 9)$ are associated with the gravity acceleration g_e and hence their stable points are biased from the origin.

Table 4.8: Tuning parameters of the AMFC used for the autonomous quadrotor

Tuning parameters and the corresponding values (as in Algorithm 1 in Table 3.1)
$\Gamma_1 = 1000 \times \text{diag}(0.001 \times \mathbf{1}_6, \mathbf{1}_6)$
$\Gamma_2 = 1 \times \text{diag}(0.001 \times \mathbf{1}_6, \mathbf{1}_6)$
$\rho_1 = 0.1$
$\rho_2 = 0.1$
$Q = 0.01 \times I_{12}$
$R = I_{12}$
$k_1 = k_2 = 1$

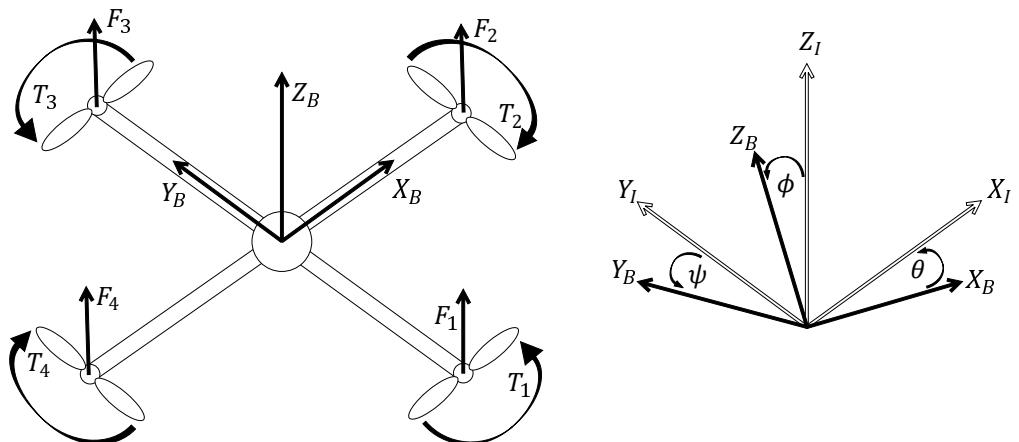


Figure 4.30: Schematic of a quadrotor

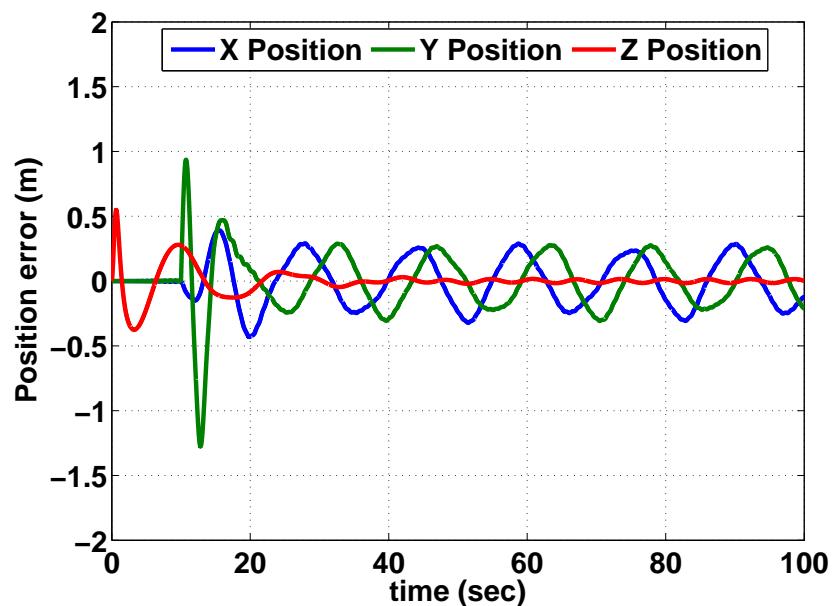
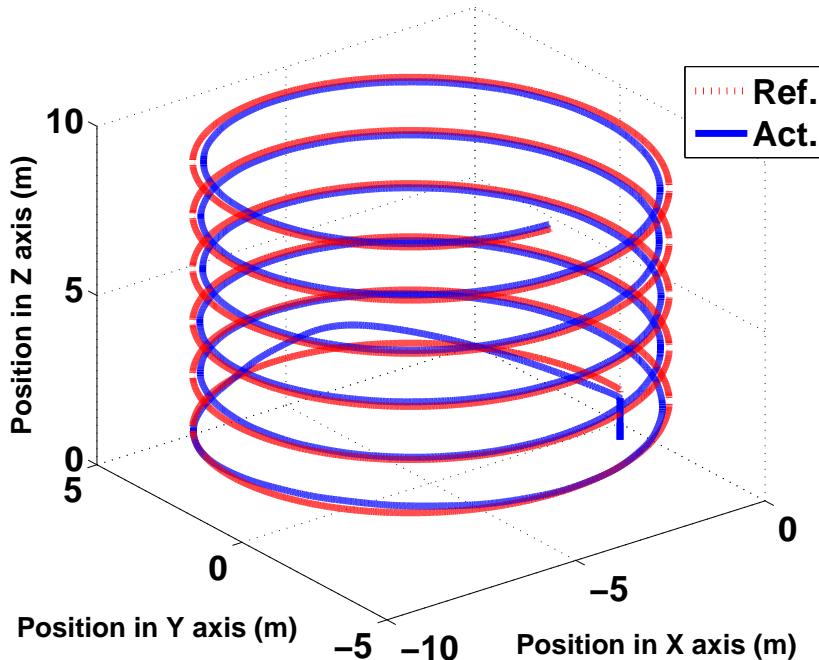


Figure 4.31: Tracking performance for the autonomous quadrotor by applying the AMFC algorithm. Tracking errors are all bounded around the origin.

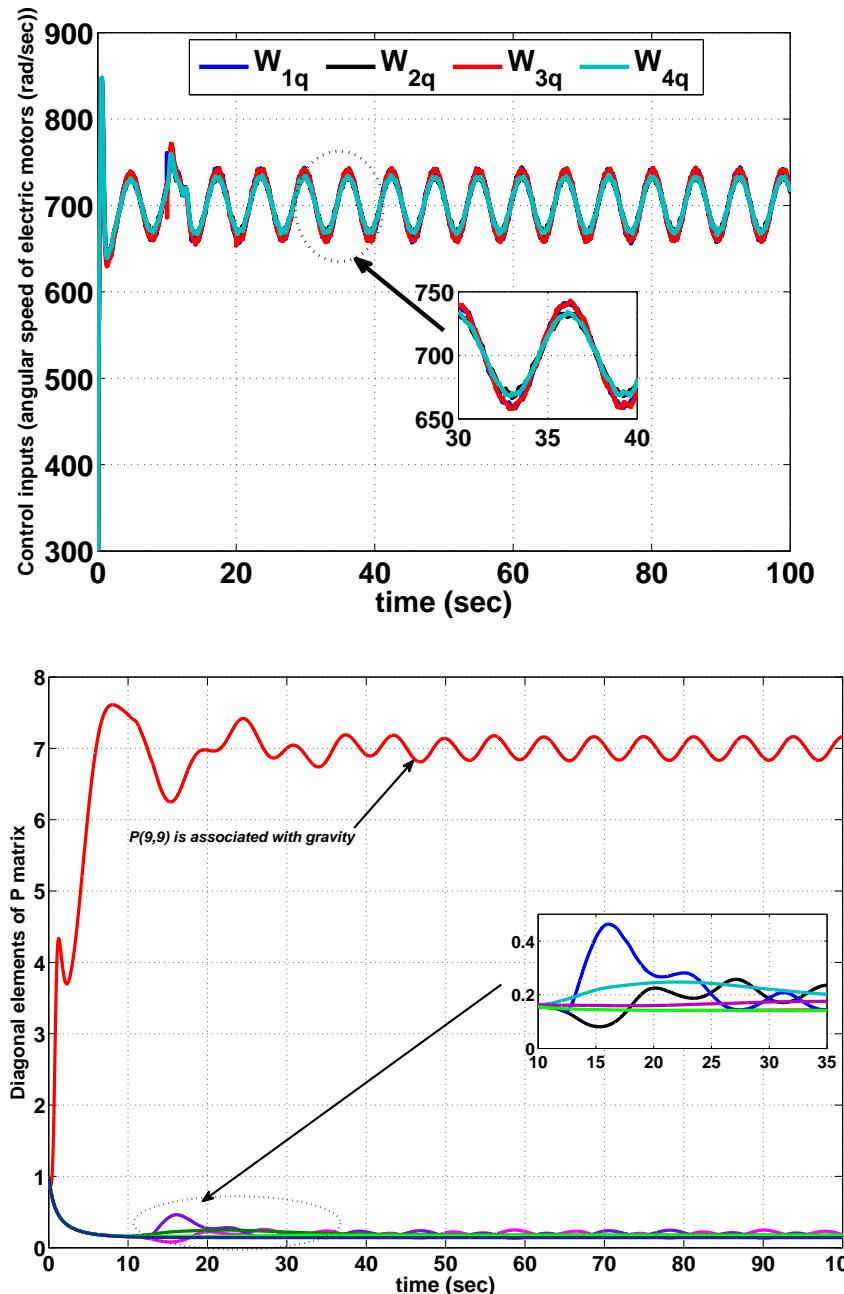


Figure 4.32: Control inputs (angular speeds of electric motors) for the autonomous quadrotor and the values for P matrix, utilizing the AMFC algorithm.

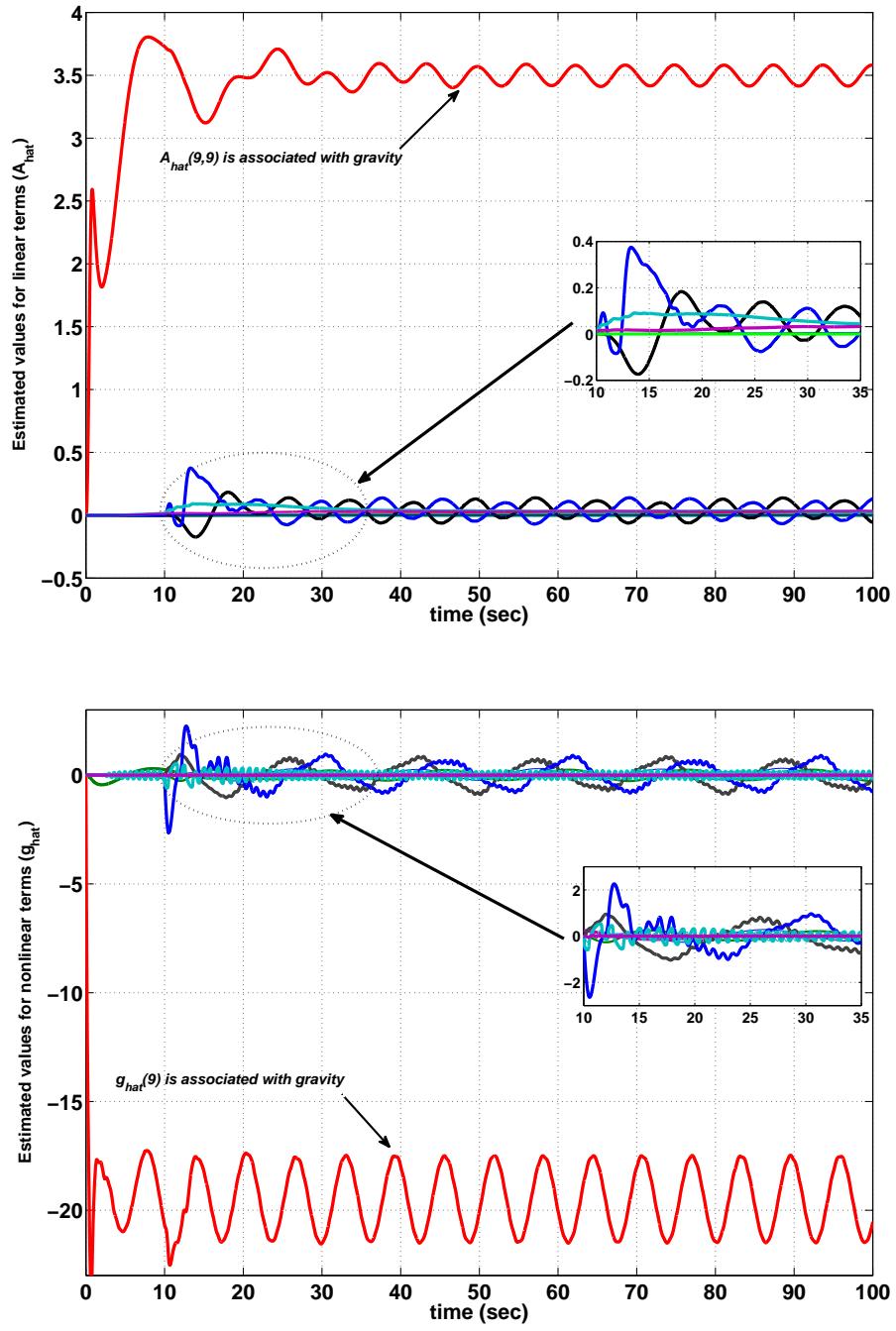


Figure 4.33: Estimated values for the unknown linear and nonlinear terms for the autonomous quadrotor, utilizing the AMFC algorithm.

4.4 Results for the CAMFC-1 algorithm in a multi-agent nonlinear dynamic system

In this section, a team of four quadrotors is considered for evaluating the performance of the CAMFC-1 algorithm which has been presented in Section 3.4. The dynamic model used for each quadrotor is same as the one presented in Section 4.3.5. Note that the inter-agent relative measurements among the agents are not available in the network of this case study. Moreover, it is assumed that only one of the quadrotors is connected to a virtual leader and has access to the desired trajectory to be tracked and the required formation topology among the four quadrotors in the network. Hence, the adjacency matrix, the pinning gain matrix and the in-degree matrix for the communication graph among the quadrotor are defined as follows,

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4.30)$$

The corresponding communication graph to the network with the specifications defined in (4.30) is presented in Fig. 4.34. In addition, the reference trajectory is a helix in the 3D space. Also, the parameter Ω for formation topology is considered as follows

$$\Omega = \begin{bmatrix} 0 & 0 & 0 \\ +r_x & +r_y & +r_z \\ -r_x & +r_y & -r_z \\ -r_x & -r_y & +2 \times r_z \\ +r_x & -r_y & -2 \times r_z \end{bmatrix}. \quad (4.31)$$

The values of formation parameters are $r_x = -0.5$, $r_y = -1$ and $r_z = 2$ for simulation time under 50 seconds. These values are changed to $r_x = 0.5$, $r_y = 1$ and $r_z = 3$ for the rest of simulation time till 120 seconds.

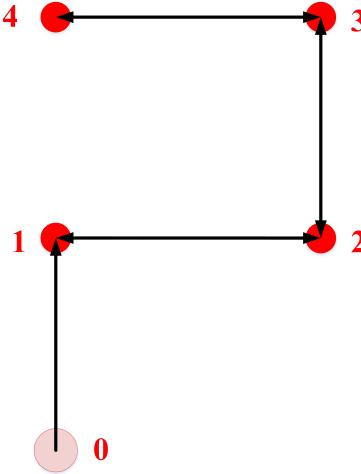


Figure 4.34: Communication graph for a team of four autonomous quadrotors used for simulation study of CAMFC-1. Here, the 0-indexed node represents the virtual leader node. The communication links in the network are undirected.

Table 4.9 shows the tuning parameters of the CAMFC-1 algorithm, the cooperative observer and the AMFC policy. The simulation results are also depicted in Fig. 4.34 to Fig. 4.43. As can be seen, the finite-time convergence is provided for estimating the reference trajectory and the formation topology parameters at all of the quadrotors in the network, by the use of the designed cooperative observer. Moreover, the tracking error for all the agents are bounded in a small region around zero. Referring to Fig. (4.35), Fig. (4.36) and Fig. (4.40), the locations of agents 1 and 3 and similarly the locations of agents 2 and 4 are exchanged after 50 seconds of simulation time. The desired formation is satisfied before and after this change. Note that the values for parameters $\hat{A}(9,9)$, $\hat{g}(9)$, $\hat{P}(9,9)$ and the other corresponding parameters in the figures are associated with the gravity acceleration g_e .

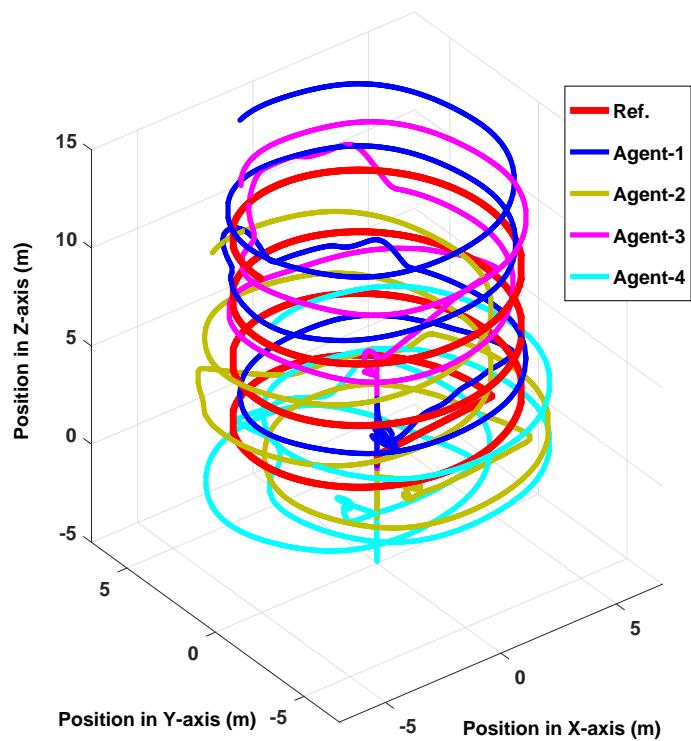


Figure 4.35: Positions of four cooperative autonomous quadrotors in the 3D space, utilizing the CAMFC-1 algorithm.

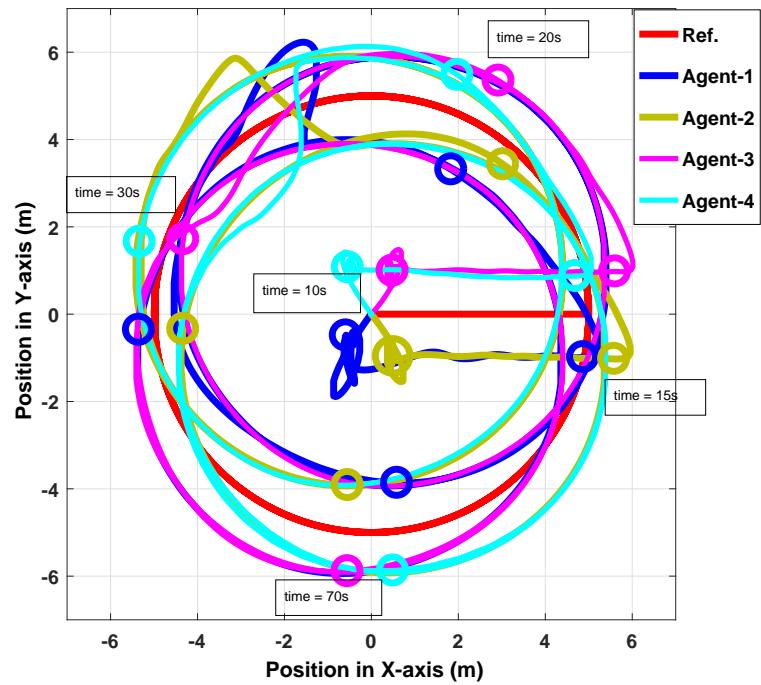


Figure 4.36: Positions of four cooperative autonomous quadrotors in the 2D space, utilizing the CAMFC-1 algorithm.

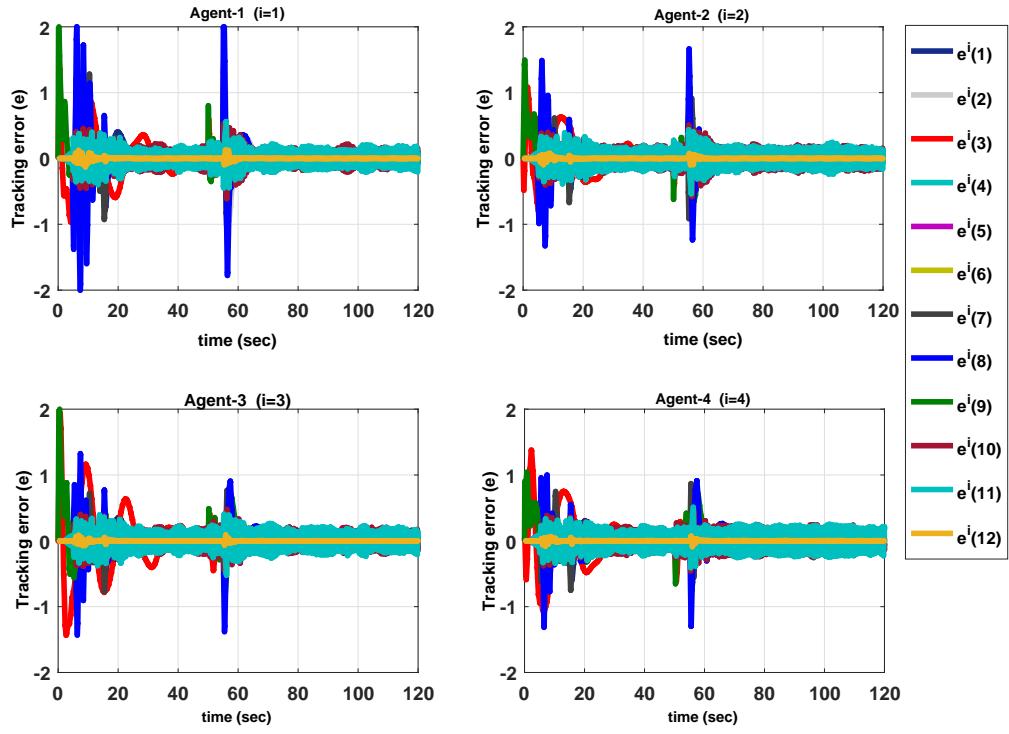


Figure 4.37: Tracking errors for four cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.

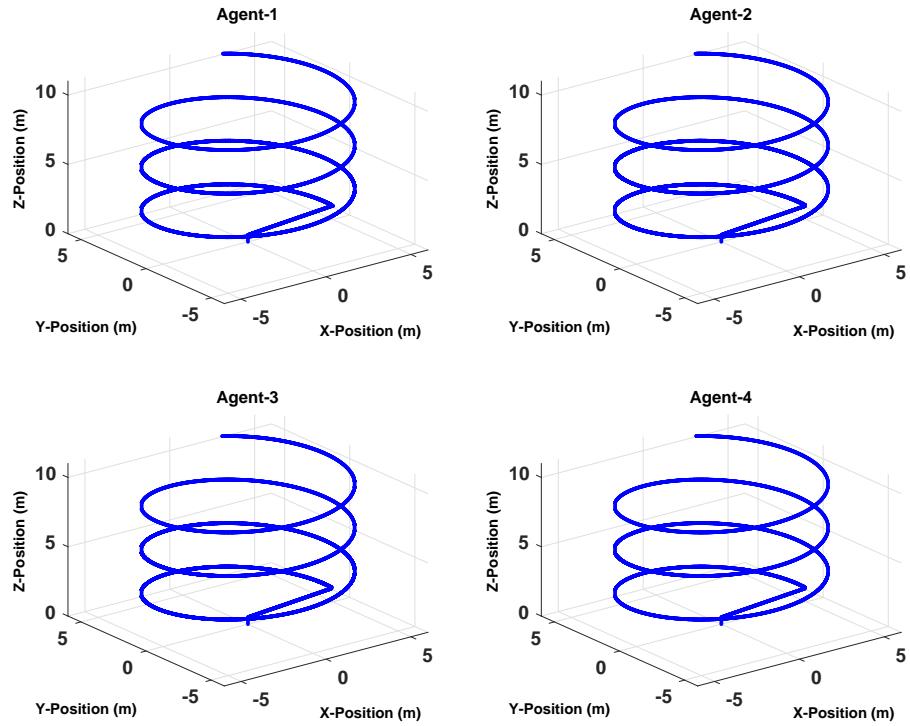


Figure 4.38: Estimated values for reference trajectory at four cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.

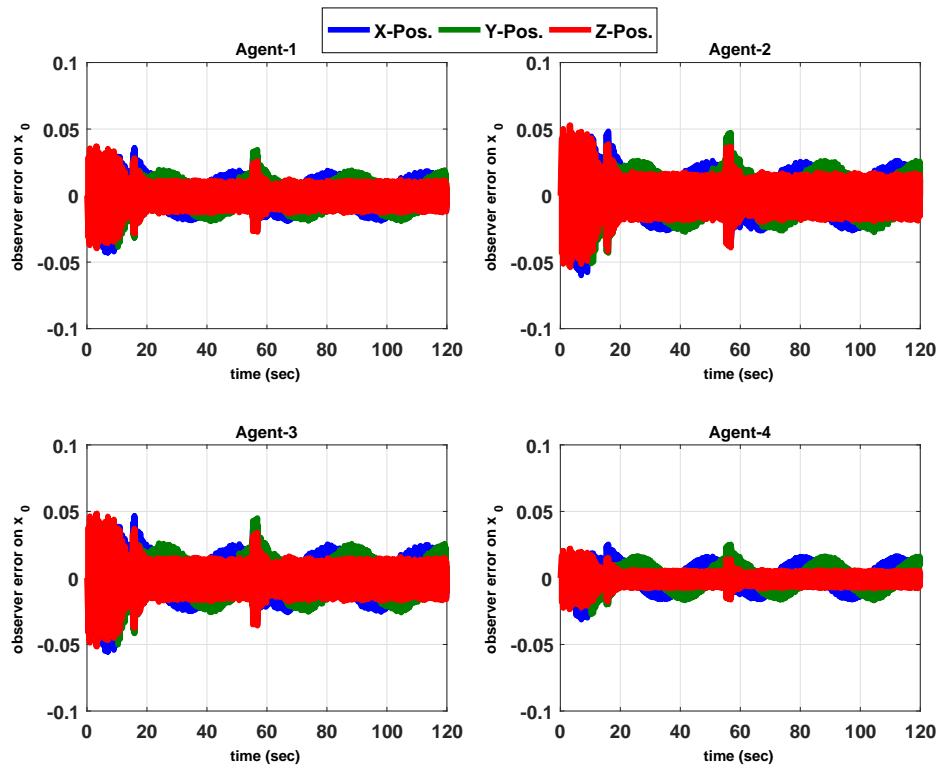


Figure 4.39: Consensus errors for observing the reference trajectory at four cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.

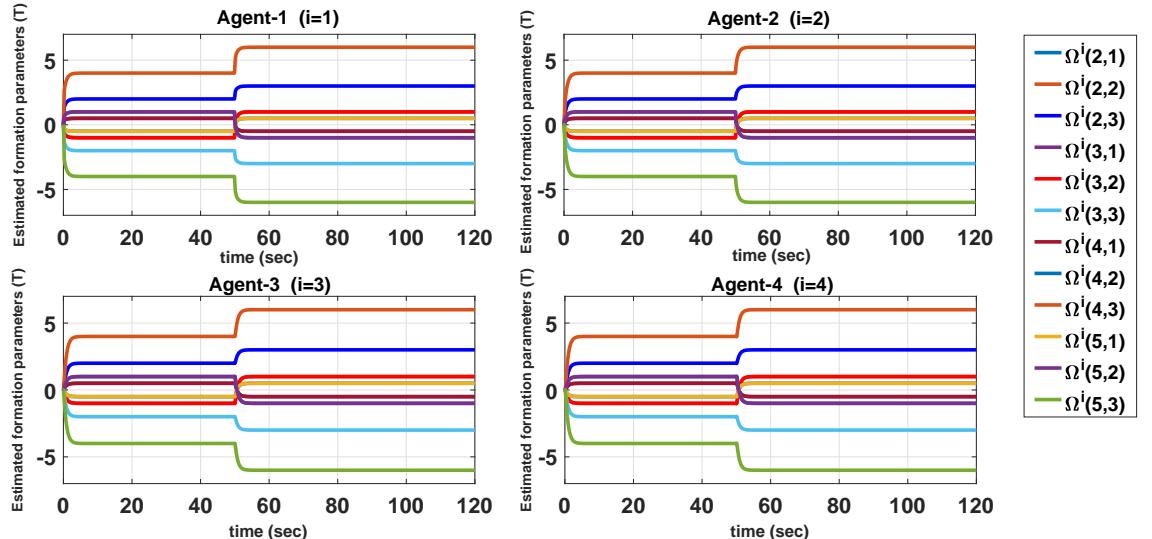


Figure 4.40: Estimated values for the formation parameters at four cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.

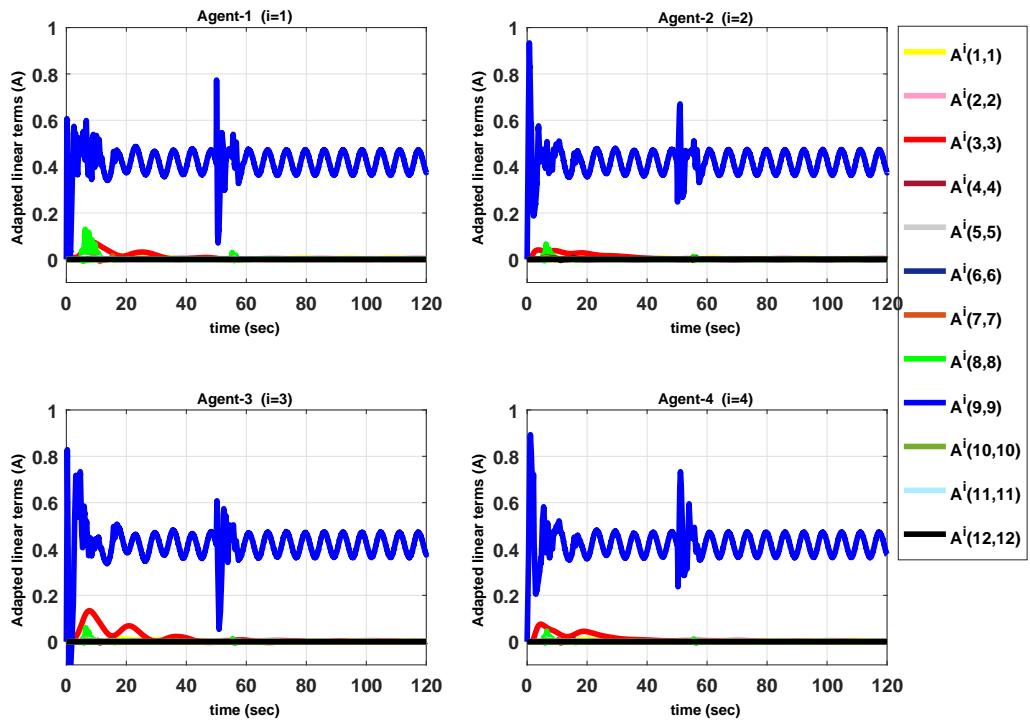


Figure 4.41: Adapted values for unknown linear terms at a team of cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.

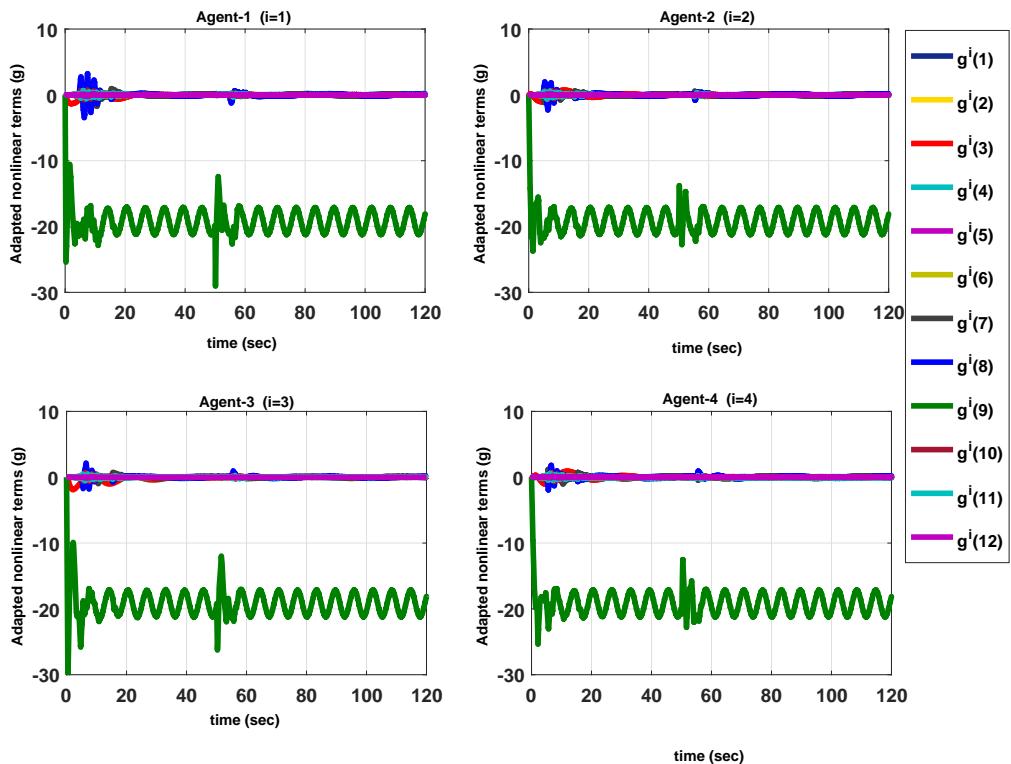


Figure 4.42: Adapted values for unknown nonlinear terms at a team of cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.

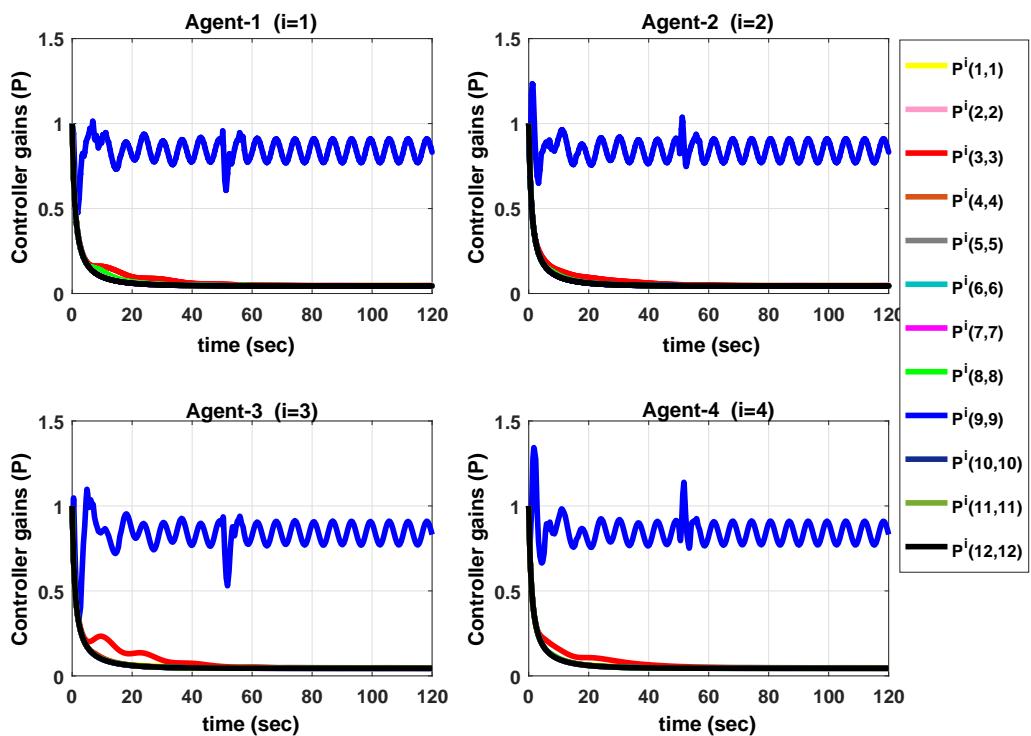


Figure 4.43: Controller gains for each of the cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.

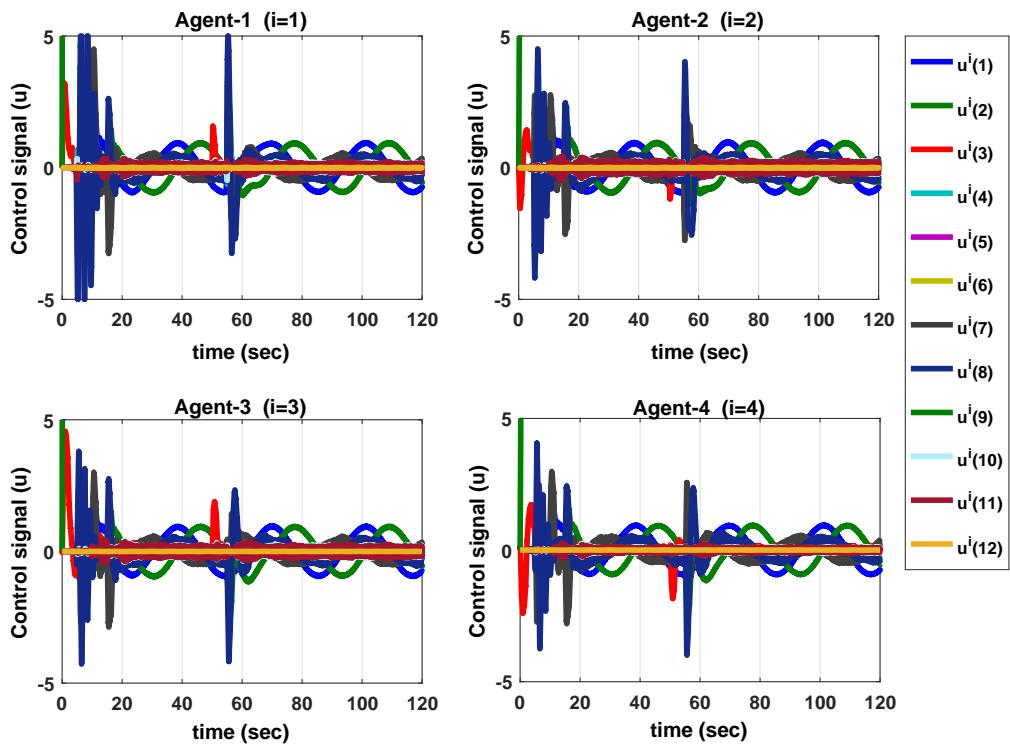


Figure 4.44: Control signals at each of the cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm.

Table 4.9: Tuning parameters of the CAMFC-1 algorithm used at each autonomous quadrotor in a network of four agents.

Parameter (as in Algorithm 2 in Table 3.2)	Value
$R = B$	I_{12}
Q	$0.001 \times I_{12}$
ρ_1	0.1
ρ_2	10
Γ_1	$[I_6; 1000 \times I_6]$
Γ_2	$[0.01 \times I_6; 10 \times I_6]$
λ	100
μ	10
X^M	$10 \times [1; 1; 1]$
Υ	$1 \times [1; 1; 1]$

4.5 Results for the CAMFC-2 algorithm in a multi-agent nonlinear dynamic system

In this section, two comparative studies are presented to evaluate the CAMFC-2 algorithm against the two state-of-the-art distributed cooperative control policies proposed using the realm of ANNs. The first comparative case is dedicated to observe the performance of the CAMFC-2 algorithm in a consensus problem, while a formation control problem is investigated in the second comparative study. In these two comparative studies, the CAMFC-2 algorithm is implemented to the same problems solved by the previous distributed cooperative solutions in the literature. At the end of this section, the CAMFC-2 algorithm is simulated in a team of autonomous quadrotors for achieving a desired formation-tracking objective in the 3D environment. Throughout this section, the comparison has been presented utilizing two following cost functions

$$\mathcal{C}^c = \int_0^{t_f} \left[\sum_{i=1}^N ((e^i)^T e^i + (u^i)^T u^i) \right] dt , \quad (4.32)$$

and

$$\mathcal{J}_{\mathcal{E}}^c = \int_0^{t_f} \left[\sum_{i=1}^N ((e^i)^T e^i) \right] dt . \quad (4.33)$$

4.5.1 Comparison study in a consensus problem

This is the case-1 of the comparative studies provided for evaluating the performance of CAMFC-2 algorithm. In this section the performance of CAMFC-2 algorithm is compared with a well-known model-free distributed cooperative control (DCC-1) policy designed and presented in (Lewis et al., 2014), to achieve a consensus over the first states of all agents in the network. The control signal at agent i in the aforementioned algorithm is defined as presented in (2.12a) and (2.12b). In that solution, the values for the main controller gains (i.e. p_i) are defined based on the solution of a Lyapunov equation (Lewis et al., 2014). Moreover, the dynamic system for the sake of current comparison study is a reverse pendulum presented with the following model (Lewis et al., 2014)

$$\dot{x}_1 = x_2 , \quad (4.34a)$$

$$\dot{x}_2 = \frac{1}{j_p} (u - b_p x_2 - m_p l_p g_e \sin x_1 + D_p(t)) , \quad (4.34b)$$

$$y = x_1 , \quad (4.34c)$$

where $j_p = 1$, $m_p = 0.1$ and $l_p = 0.1$ are the moment of inertia, mass and the length of the pendulum, $b_p = 0.01$ is a constant for the damping force and $D_p(t) = \sin(2t)$ is the external disturbance. Here, we have a network of five agents with the dynamic system as presented in (4.34a) to (4.34c) at each agent. The communication graph in this network is defined as follows (Lewis et al., 2014)

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4.35)$$

Only the third agent is pinned to the leader (Lewis et al., 2014). The desired consensus value for x_1 at all agents in the network is $x_1^d = 3 \sin(0.5t)$. Initial values for states of agents are $x^1(0) = [0; 0]$, $x^2(0) = [-0.5; 0]$, $x^3(0) = [+0.5; 0]$, $x^4(0) = [-1; 0]$, $x^5(0) = [+1; 0]$. The values for tuning parameters of the two cooperative control policies are presented in Table 4.10. The tuning parameters for both policies are chosen so as to have same values of $\mathcal{I}_{\mathcal{E}}^c$. In this sense, the required control efforts in the network to reach at the same level of consensus error, can be compared for the cooperative policies DCC-1 and CAMFC-2. Values for \mathcal{C}^c and $\mathcal{I}_{\mathcal{E}}^c$ for the two algorithms are presented in Table 4.10, as well. As it can be seen, by utilizing the CAMFC-2 algorithm, one can achieve a similar level of consensus errors with implying less control efforts in the network. The value of \mathcal{C}^c for CAMFC-2 algorithm is about 5.5 % lower than the value of this parameter for DCC-1 algorithm. Additionally, it should be noted that there is no need for defining the activation functions or regressor variables in CAMFC-2 algorithm, while log-sigmoid activation function as $\frac{1}{1+e^{-kn_t}}$ is used at each of the four neural nodes in the hidden layer of ANN incorporated in the DCC-1 policy.

The simulation results for this comparison study are presented in Fig. 4.45 to Fig.

4.47. As it is observed in Fig. 4.46, the fewer control effort is required by the CAMFC-2 algorithm, during the transient portion of the response. Furthermore, the estimated values for linear and nonlinear terms, as well as the main controller gains and also the estimated values for the leader state and control input are depicted in Fig. 4.48 to Fig. 4.52. All of these values are bounded and the required convergence has been achieved for each of them. Specifically, the estimated values for the leader state and control input at the agents are converged in finite-time, very fast (as observed in Fig. (4.51) and Fig. (4.52)). Based on Fig. (4.48), the unknown nonlinear terms at all the agents are estimated in the transient phase of the simulation, and then reach and remain at the stable points. On the other hand, as deduced from Fig. (4.49) and Fig. (4.50), the estimated values for unknown linear terms are updated throughout the simulation time and the main controller gains are tuned online at the agents, correspondingly.

Table 4.10: Tuning parameters for the DCC-1 and CAMFC-2 algorithms implemented at agent i in Section 4.5.1

Parameter	DCC-1(Lewis et al., 2014) (as in (2.12a) and (2.12b))	CAMFC-2 (as in Algorithm 3 in Table 3.3)
Tuning parameters	$F_c = 1$ $c = 10$ $k_c = 1 \times I_4$ $\lambda_c = 12.9$ $k_n = 1$	$\Gamma_1 = 1e3 \times diag(1, 1)$ $\Gamma_2 = 0.01 \times diag(0.001, 1)$ $\rho_1 = 1$ $\rho_2 = 0.01$ $Q^i = 15 \times I_2$ $R^i = I_2$ $\kappa = 10$ $\lambda = \lambda_1 = \mu = 1000$ $X_M = \Upsilon^M = U^M = 10 \times \mathbf{1}_n$
Value of $\mathcal{J}_{\mathcal{E}}^c$ for the network	2.286	2.281
Value of \mathcal{C}^c for the network	5561	5255
Number of adaptive laws at each agent	4	4

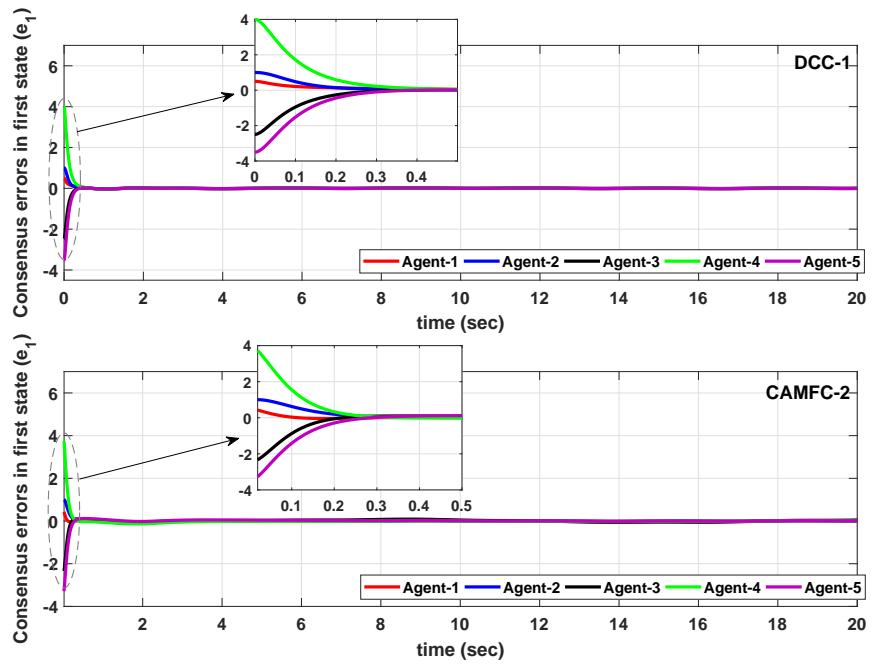


Figure 4.45: The consensus errors for all agents using the DCC-1 algorithm (top) and the CAMFC-2 algorithm (bottom) . The convergence is provided in both cases.

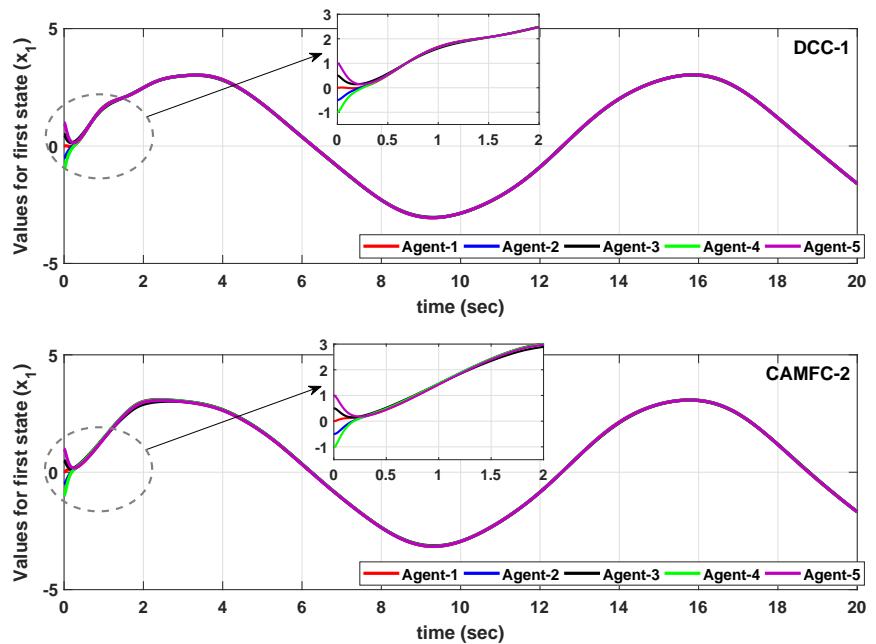


Figure 4.46: The values for first states of all agents using the DCC-1 algorithm (top) and the CAMFC-2 algorithm (bottom).

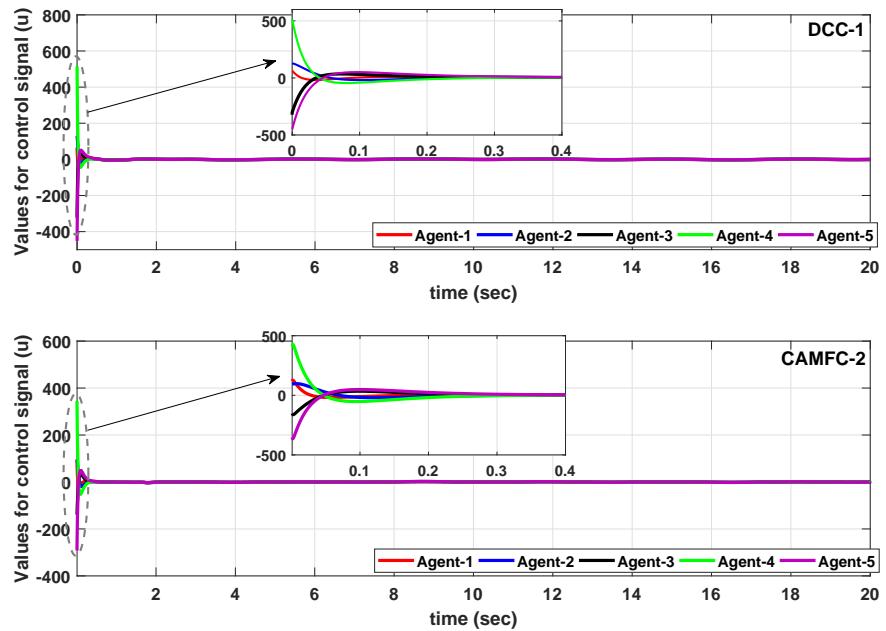


Figure 4.47: The values for control signals at all agents using DCC-1 algorithm (top) and the CAMFC-2 algorithm (bottom).

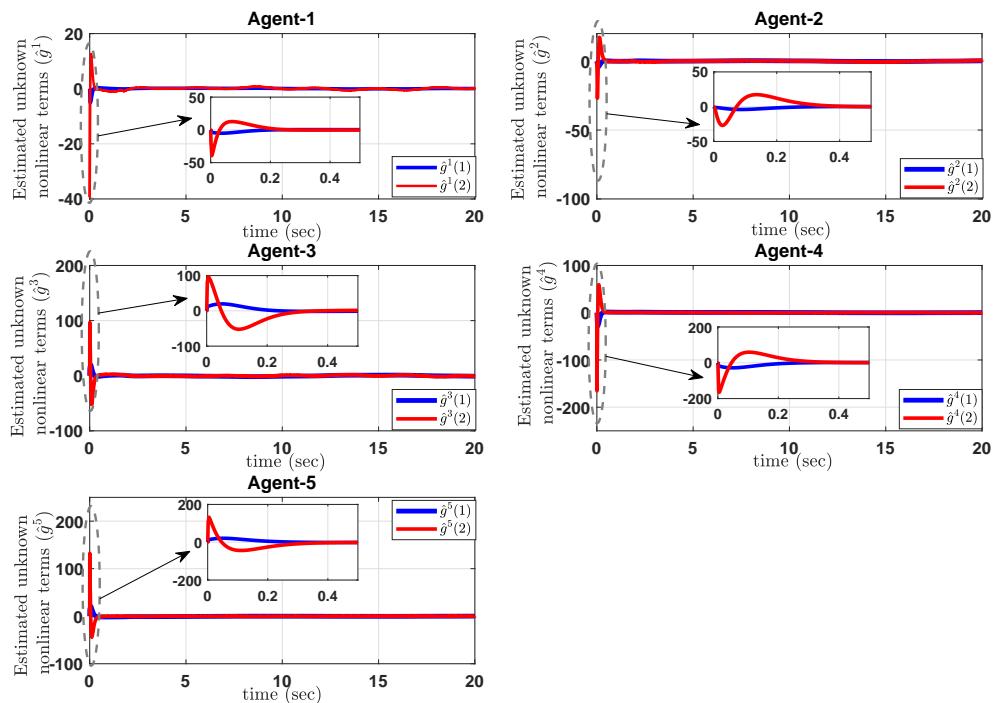


Figure 4.48: The estimated values for unknown nonlinear terms at all agents in the problem of Section 4.5.1, using the CAMFC-2 algorithm.

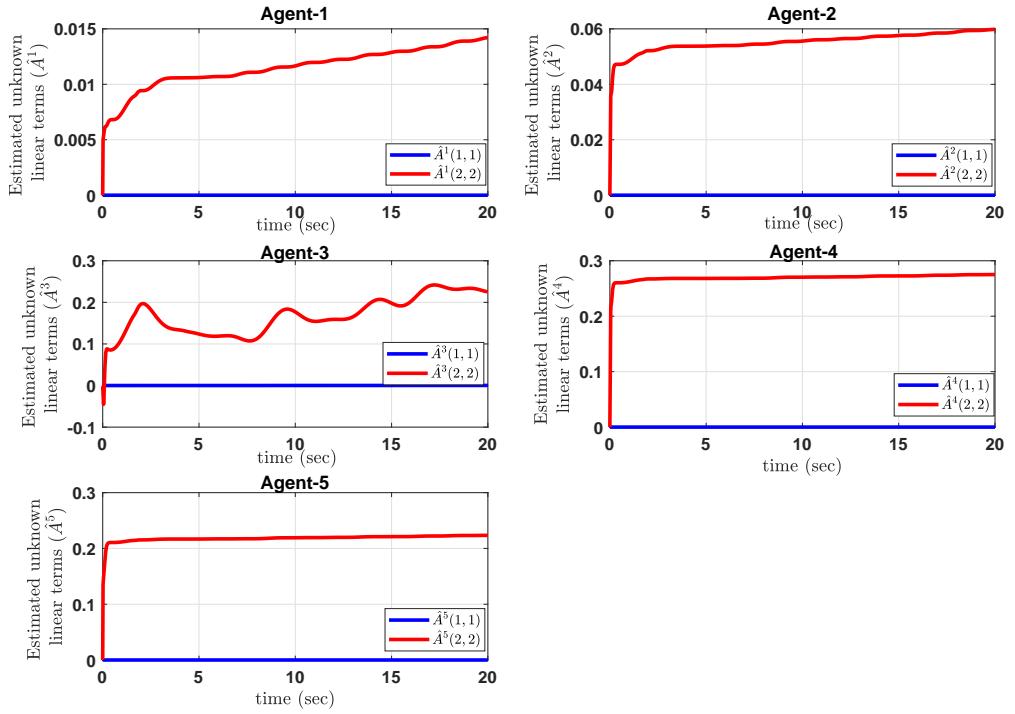


Figure 4.49: The estimated values for unknown linear terms at all agents in the problem of Section 4.5.1, using the CAMFC-2 algorithm.

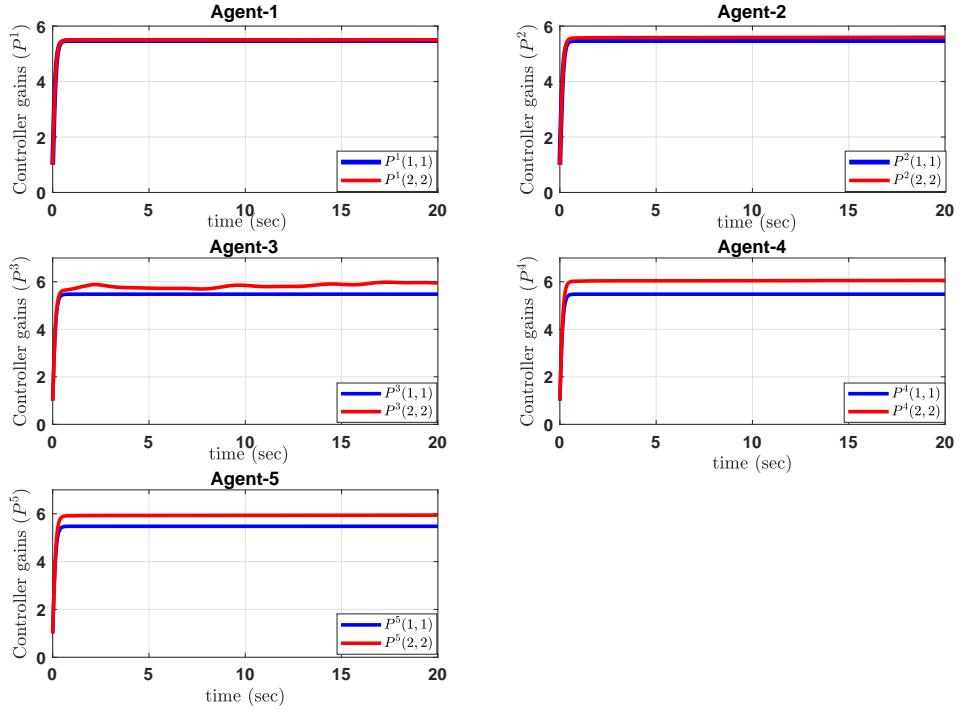


Figure 4.50: The values for main controller gains at all agents in the problem of Section 4.5.1, using the CAMFC-2 algorithm.

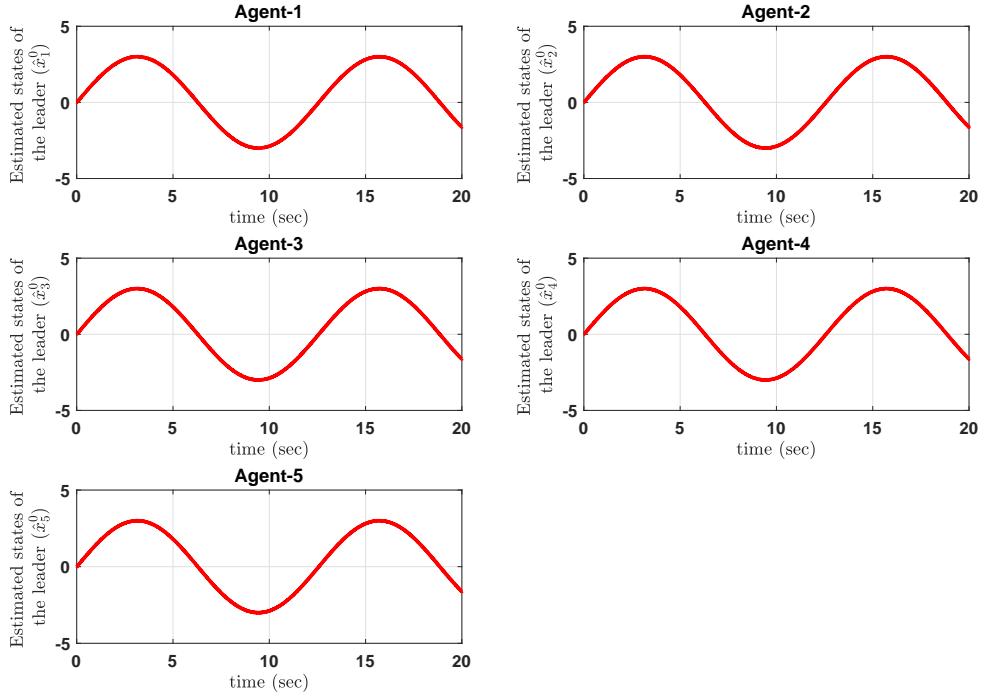


Figure 4.51: The estimated values for the leader state at all agents in the problem of Section 4.5.1, using the CAMFC-2 algorithm.

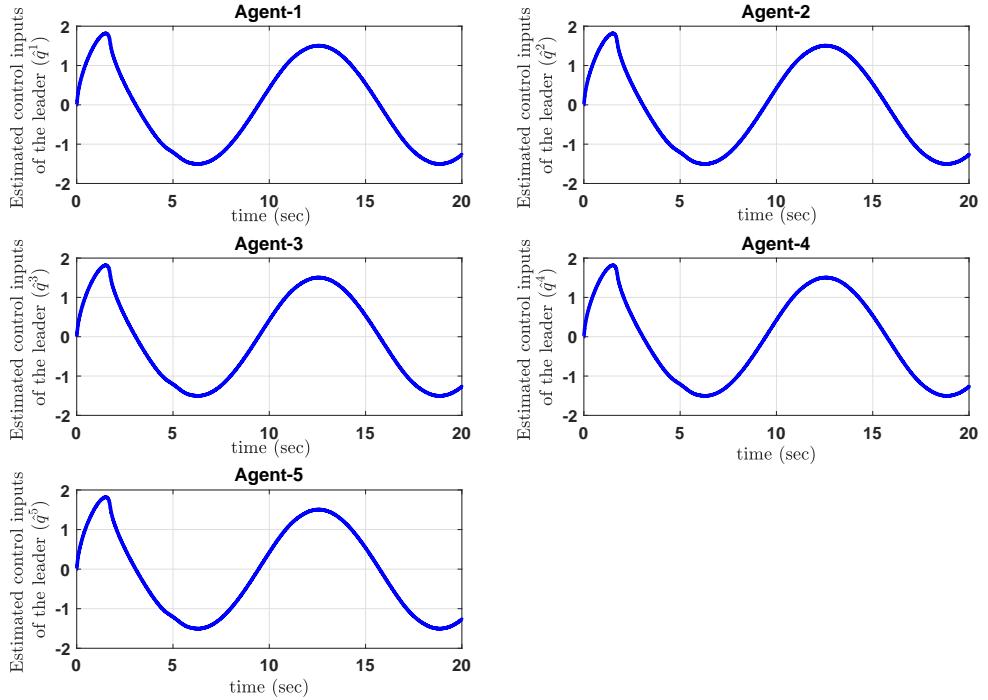


Figure 4.52: The estimated values for the leader control input (which is the derivative of the leader state) at all agents in the problem of Section 4.5.1, using the CAMFC-2 algorithm.

4.5.2 Comparison study in a formation-tracking problem

This is the case-2 of the comparative studies provided for evaluating the performance of CAMFC-2 algorithm. Here, a network consisting six nonlinear dynamic systems with internal and external uncertainties is used to evaluate the performance of the proposed CAMFC-2 algorithm in formation control problem. The objective is to provide a predefined constant formation among the agents, which are initially located at different positions far from each other. The benchmark solution, which the performance of CAMFC-2 is compared with is the distributed cooperative control (DCC-2) algorithm proposed in (Meng et al., 2017). The DCC-2 algorithm added a robust term into the realm of ANNs and provided a solution for consensus and formation control problems in a network of completely unknown nonlinear dynamic systems. The DCC-2 policy is presented in (2.16a). In the current problem, the agents dynamics is given by (Meng et al., 2017)

$$\frac{d}{dt} \begin{bmatrix} x_1^i \\ x_2^i \end{bmatrix} = \begin{bmatrix} x_2^i \sin(c_1^i x_1^i) \\ x_1^i \cos(c_2^i (x_2^i)^2) \end{bmatrix} + \begin{bmatrix} u_1^i \\ u_2^i \end{bmatrix} + D^i(t), \quad (4.36)$$

where the constant variables c_1^i and c_2^i for all of the agents in the network are set as $[c_1^1; c_2^1] = [0.5; 0.4]$, $[c_1^2; c_2^2] = [-0.5; 0.4]$, $[c_1^3; c_2^3] = [6; -5]$, $[c_1^4; c_2^4] = [-10; 12]$, $[c_1^5; c_2^5] = [10; 12]$, $[c_1^6; c_2^6] = [0.01; 10]$. In addition, the external disturbances at all of the agents are defined as $D^1(t) = [\sin(-t)\exp(-2t); \cos(t^2)]$, $D^2(t) = [\exp(-3t); \sin(t^2)\cos(t)]$, $D^3(t) = [\cos(t^2); \exp(-t)]$, $D^4(t) = [-\sin(t^2); \cos(t)\exp(-3t)]$, $D^5(t) = [\sin(t^2); \cos(t)\exp(-5t)]$, $D^6(t) = [\exp(-4t); \sin(t)\cos(t^2)]$. Here, the communication graph in the network is defined by (Meng et al., 2017)

$$\mathcal{A} = \begin{bmatrix} 0 & 0.2 & 0 & 0 & 0 & 0.4 \\ 0.2 & 0 & 0.3 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0.3 & 0 \\ 0 & 0 & 0 & 0.3 & 0 & 0.5 \\ 0.4 & 0 & 0 & 0 & 0.5 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 0.6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9 \end{bmatrix},$$

$$\mathcal{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$
(4.37)

As it is obvious, only the agent 1 is connected to the leader. The demand formation variables for the agents are $\eta^1 = [\frac{\sqrt{3}}{2}; \frac{1}{2}]$, $\eta^2 = [0; 1]$, $\eta^3 = [\frac{-\sqrt{3}}{2}; \frac{1}{2}]$, $\eta^4 = [\frac{-\sqrt{3}}{2}; -\frac{1}{2}]$, $\eta^5 = [0; -1]$ and $\eta^6 = [\frac{\sqrt{3}}{2}; -\frac{1}{2}]$. Moreover, the initial values for states of agents are $x^1(0) = [6; 2]$, $x^2(0) = [3; 3\sqrt{3}]$, $x^3(0) = [-3; 3\sqrt{3}]$, $x^4(0) = [-6; -2]$, $x^5(0) = [-3; -3\sqrt{3}]$ and $x^6(0) = [3; -3\sqrt{3}]$.

The simulation results for this case are presented in Fig. 4.53 to Fig. 4.59. The values for tuning parameters of the DCC-2 and CAMFC-2 algorithms implemented in the current problem are presented in Table 4.11. Similar to the previous comparative study in Section 4.5.2, here the tuning parameters for both policies are chosen in order to reach at the same values of \mathcal{C}^c . In this regard, the control efforts of the two cooperative control algorithms can be compared. As it can be observed, while the value for $\mathcal{I}_{\mathcal{E}}^c$ for the DCC-2 and CAMFC-2 algorithms are almost similar, the

required control efforts which is revealed by the value of \mathcal{C}^c is fewer for the CAMFC-2 algorithm. Furthermore, while tangent-hyperbolic activation function as $\tanh(k_n e^i)$ is used at each of the ten neural nodes in the hidden layer of ANN embedded in the DCC-2 algorithm, there is no need to define the regressor parameters for implementation of the CAMFC-2 algorithm.

As it can be observed from the simulation results, the consensus errors for all of the agents using both cooperative control policies have been converged to zero. In addition, the desired formation variables are achieved by all of the agents. While CAMFC-2 has more over-shoot in the response in comparison with the DCC-2, the faster rising time is provided by the CAMFC-2 algorithm. Specifically, the rising time for CAMFC-2 is 0.1 and 0.2 seconds for the first and second states, respectively. These values are 0.2 and 0.4 seconds for the DCC-2 algorithm. But, less over-shoot is achieved with the DCC-2 algorithm. Having all of these issues into account, the values for \mathcal{J}_e^c for the two algorithms are almost the same. According to Fig. 4.55, the control inputs of the DCC-2 algorithm have been affected more by the external disturbances. This leads to more fluctuations in the control signal of the DCC-2 algorithm and consequently larger value of \mathcal{C}^c . The value of \mathcal{C}^c for CAMFC-2 is 51.5% lower than the value of this parameter for DCC-2 algorithm.

The estimated values for unknown terms are presented in Fig. 4.56 and Fig. 4.57, and the main controller gains are depicted in Fig. 4.58. These values are updated during the transient phase of the response and reached to some finite values at the steady-state condition. Finally, the estimated values for the formation variables at the 4th agent are presented in Fig. 4.59. The convergence of these variables is provided in

finite-time, due to the use of signum function in the corresponding distributed observer.

The estimation of the formation variables at the other agents in the network, follows similar characteristic.

Table 4.11: Tuning parameters for the DCC-2 and CAMFC-2 algorithms implemented at agent i in Section 4.5.2

Parameter	DCC-2(Meng et al., 2017) (as in (2.16a) to (2.16c))	CAMFC-2 (as in Algorithm 3 in Table 3.3)
Tuning parameters	$\Gamma^c = 0.1 \times I_{10}$ $\theta^c = 10$ $k^c = 28$ $\hat{B}(0) = 5$ $k_n = 1$	$\Gamma_1 = 1e3 \times diag(1, 1)$ $\Gamma_2 = 1e2 \times diag(1, 1)$ $\rho_1 = \rho_2 = 1$ $Q^i = 20 \times I_2$ $R^i = I_2$ $\kappa = 10$ $\lambda = \lambda_1 = \mu = 1000$ $X_M = Y^M = U^M = 10 \times \mathbf{1}_n$
Value of $\mathcal{J}_{\mathcal{E}}^c$ for the network	1.321	1.314
Value of \mathcal{C}^c for the network	4419	2144
Number of adaptive laws at each agent	10	4

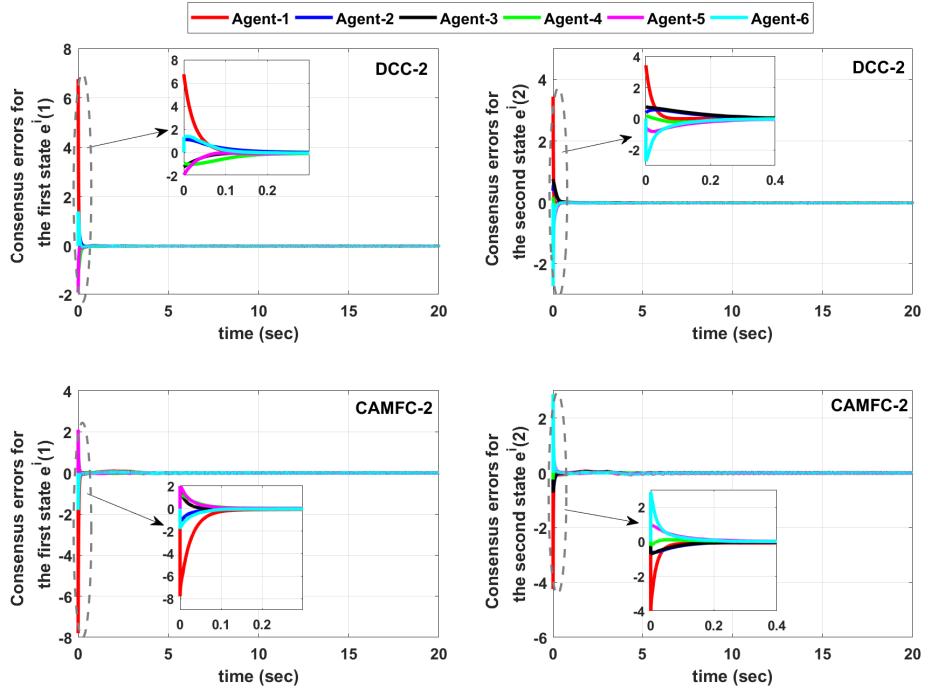


Figure 4.53: The consensus errors of the first and second states for all agents using the DCC-2 algorithm (top) and the CAMFC-2 algorithm (bottom). The convergence is provided in both cases.

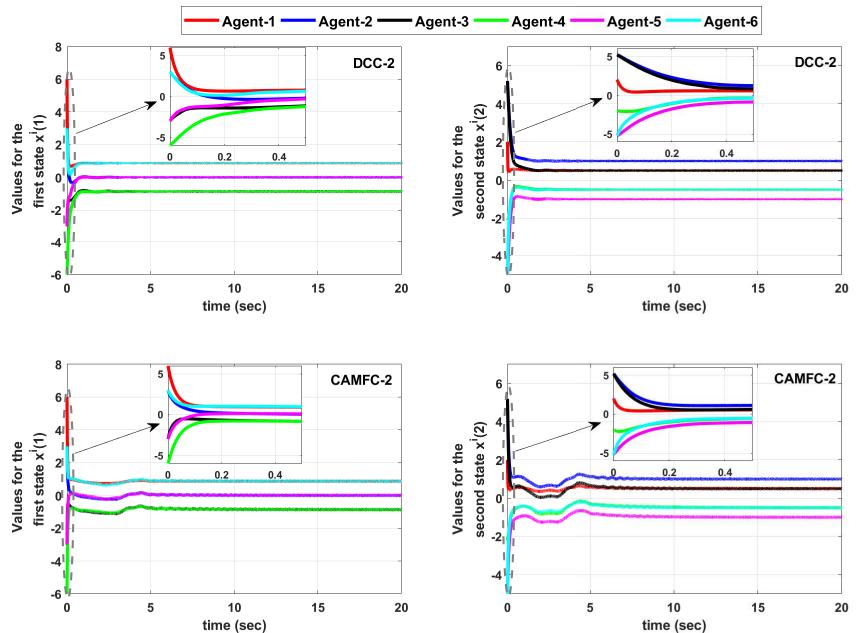


Figure 4.54: The values of first and second states for all agents using the DCC-2 algorithm (top) and the CAMFC-2 algorithm (bottom). The convergence is provided in both cases, while CAMFC-2 has more overshoot in the response, but faster rising-time.

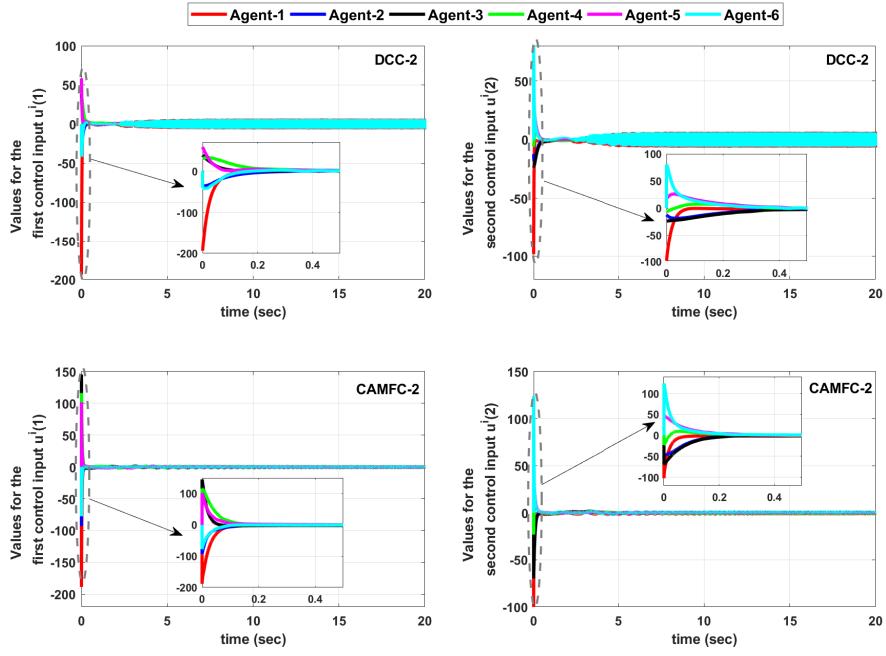


Figure 4.55: The values of first and second control inputs for all agents using the DCC-2 algorithm (top) and the CAMFC-2 algorithm (bottom). External disturbance has more effect on the DCC-2 algorithm and leads to severe fluctuations in the steady-state condition.

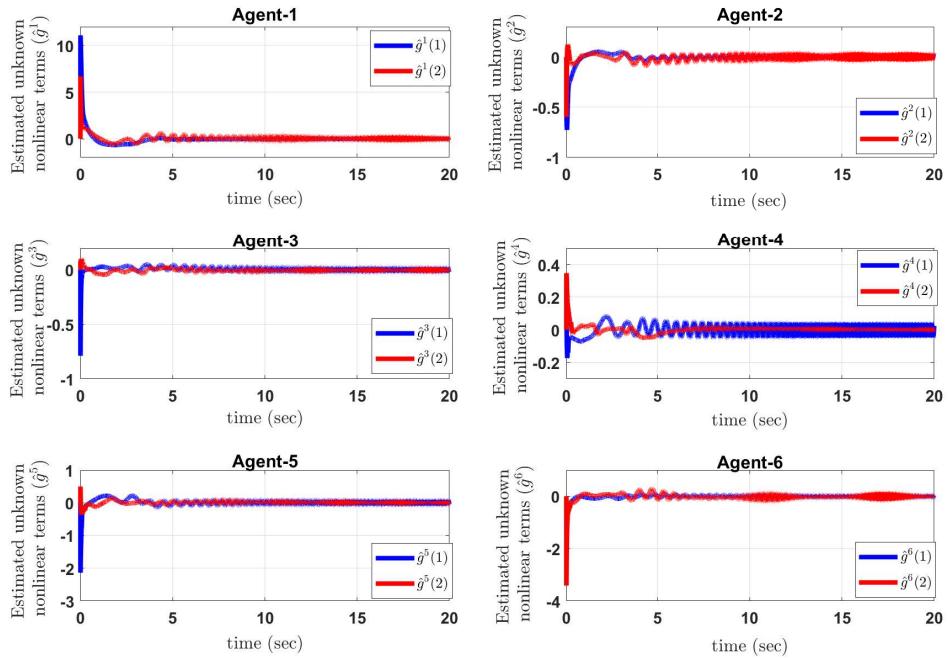


Figure 4.56: The estimated values for unknown nonlinear terms at all agents in the problem of Section 4.5.2, using the CAMFC-2 algorithm.

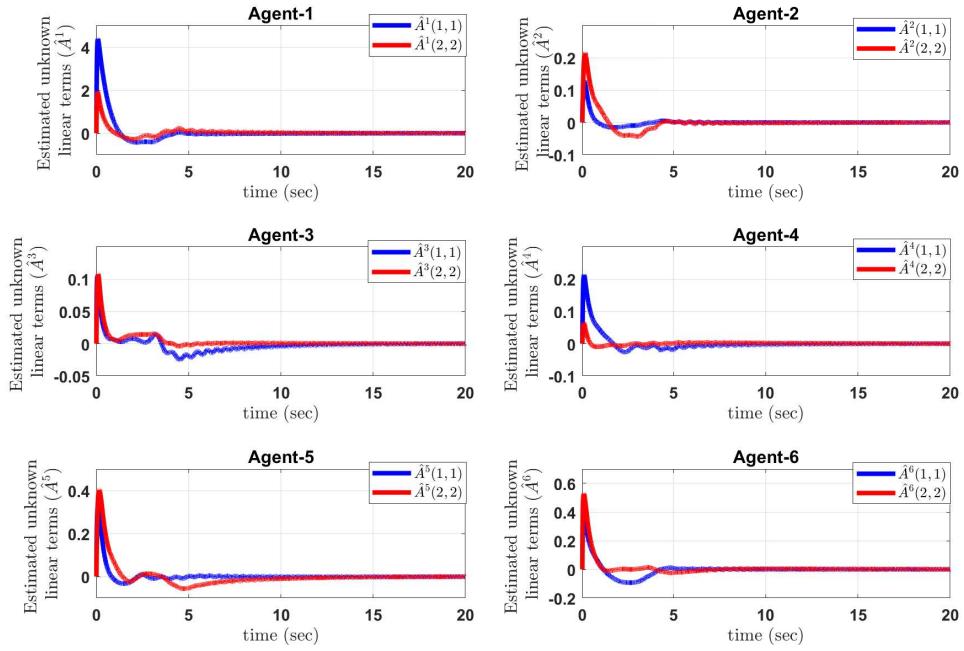


Figure 4.57: The estimated values for unknown linear terms at all agents in the problem of Section 4.5.2, using the CAMFC-2 algorithm.

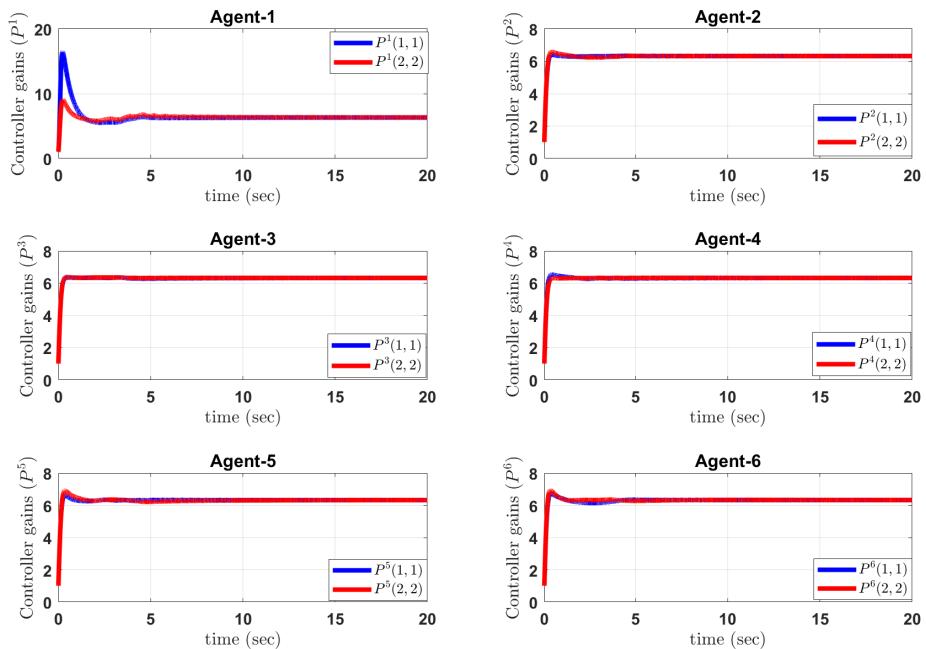


Figure 4.58: The values of main controller gains at all agents in the problem of Section 4.5.2, using the CAMFC-2 algorithm.

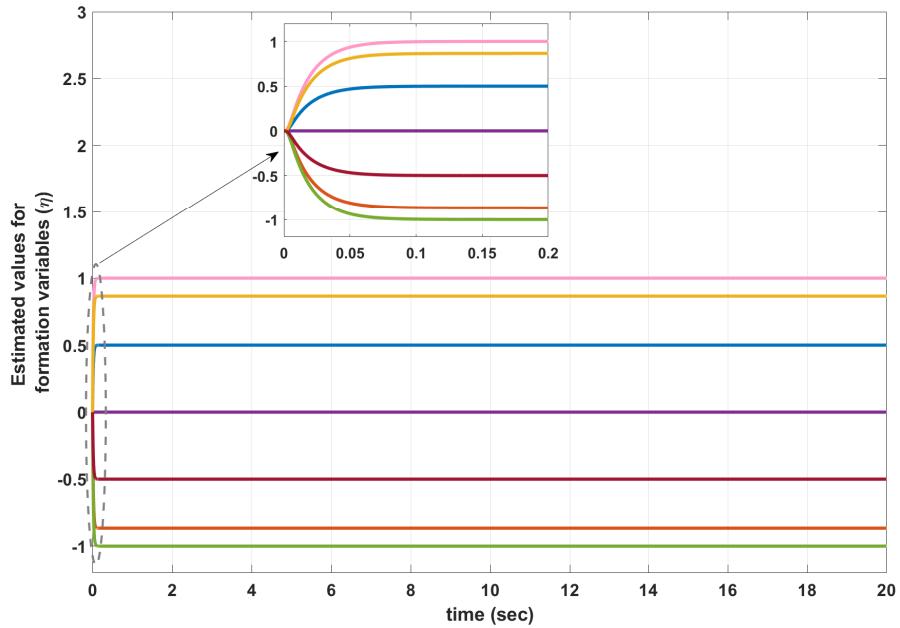


Figure 4.59: The estimated values of formation variables at Agent-4 in the problem of Section 4.5.2, using the CAMFC-2 algorithm.

4.5.3 Application of CAMFC-2 algorithm in a team of quadrotors

Here, the application of the CAMFC-2 algorithm is presented in a team of four autonomous quadrotors. As it is shown in Section 4.3.5, the dynamic system of a quadrotor is under-actuated and the desired Euler angles of the quadrotor is determined according to the required thrust forces that bring the quadrotor to the desired coordinates in the 3D space (i.e. the set-points for X-Y-Z positions). This relation is presented in (4.29a) and (4.29b). In this regard, a cascade control scheme can be employed at each agent in the network (Lee et al., 2011), in which the translational motion of the quadrotor is controlled with the CAMFC-2 algorithm using only the inter-agent relative measurements of positions and linear velocities, while the rotational motion of the quadrotor is controlled with the AMFC algorithm using locally-measured Euler angles and angular velocities. As mentioned before, the

set-points of the Euler angles are defined according to the outputs of CAMFC-2 algorithm at each quadrotor in the network. Here, it is assumed that the relative position and relative velocity for each pair of neighboring quadrotors are measured accurately with some on-board sensors. Later in Section 4.6.3, the relative position estimating and the ACL algorithms are used for estimating the relative positions in the network.

In the current simulation, the network is same as the one depicted in Fig. 4.34, with the corresponding adjacency, in-degree and pinning gain matrices as presented in (4.30). The desired trajectory in 3D space which is followed by the virtual leader, includes some step functions as follows

$$x_d = \begin{cases} 0 & , \quad t < 100s \\ 5 & , \quad 100 \leq t < 300s \\ 0 & , \quad t \geq 300s \end{cases}, \quad (4.38a)$$

$$y_d = \begin{cases} 0 & , \quad t < 200s \\ 5 & , \quad 200 \leq t < 400s \\ 0 & , \quad t \geq 400s \end{cases}, \quad (4.38b)$$

$$z_d = \begin{cases} 0 & , \quad t < 10s \\ 5 & , \quad 10 \leq t < 500s \\ 0 & , \quad t \geq 500s \end{cases}. \quad (4.38c)$$

Note that a smoothing transfer function $T_F = \frac{1}{\tau_s s + 1}$ with time constant of $\tau_s = 10$ is used to make the above step functions reachable for the quadrotor dynamics. The dynamics at each quadrotor in the network is similar to the model in (4.24a) to (4.24d).

The values for constant variables are same as the ones used in Section 4.3.5 with the disturbances as $\vec{f}_q = 0.1 \sin([0; 0; 1])$ and $\vec{\tau}_q = 1 \sin([1; 1; 1])$. In order to implement the controllers and determine the set-points for the speed of electric motors at each

quadrotor, only the values of $k_l = 1e-5$ needs to be defined exactly with no knowledge required on other variables, including k_t , m_q and l_q . Moreover, the parameter Ω for formation topology is considered as follows

$$\Omega = \begin{bmatrix} 0 & 0 & 0 \\ +r_x & +r_y & +r_z \\ -r_x & +r_y & +2 \times r_z \\ -r_x & -r_y & +3 \times r_z \\ +r_x & -r_y & +4 \times r_z \end{bmatrix}, \quad (4.39)$$

where $r_x = r_y = r_z = 1m$. Note that, the first row in (4.39) shows the position of the virtual leader in 3D space in a local coordinate system fixed to it, while each of the other rows is the position of the corresponding agent in that local coordinate system. In addition, the desired difference between the linear velocities of the all quadrotors in the network is zero.

The values for tuning parameters of the CAMFC-2 and the AMFC algorithms are presented in Table 4.12. Note that here a smaller value is used for k_2 in order to avoid high frequency signals in the sliding differentiators used in the AMFC and CAMFC-2 algorithms. The simulation results for this study are depicted in Fig. 4.60 to Fig. 4.74. As it is shown in Fig. 4.60 and Fig. 4.61, desired formation-tracking objective defined by (4.38a) to (4.38b) and (4.39), has been achieved for a network of four quadrotors in a 3D space. Also, the consensus errors for translational motions and the tracking errors for rotational motions are all bounded in small sets around the origin, as observed in Fig. 4.62 and Fig. 4.63. From Fig. 4.64, the set-points for the speed of electric motors as the control signals are all bounded and small differences in these speeds provide the movements in X or Y directions. This comes from changes in the desired Euler

angles at each quadrotor (Fig. 4.65), corresponding to the desired movement in X or Y directions.

According to Fig. 4.66 to Fig. 4.69, all the estimated linear and nonlinear terms for translational and rotational motions are bounded within small sets around zero. Larger values are computed for the terms corresponding to the translational motion in Z direction, which is associated with the earth gravity. Moreover, the changes in the estimated values of linear and nonlinear terms are observed at the points where there are changes in the desired position in 3D space, as presented in (4.38a) to (4.38b) and (4.39). Corresponding to the changes in the estimated linear terms, the values of main controller gains are also updated during the simulation (Fig. 4.70 and Fig. 4.71). According to Fig. 4.72 to Fig. 4.74, the estimated values for the formation variables, the states and control inputs of the leader are provided at all quadrotors in the network in finite time, due to the existence of the signum term in the proposed cooperative observer in Section 3.4.

Table 4.12: The tuning parameters for the AMFC and the CAMFC-2 algorithms used for rotational and translational motions, respectively at each quadrotor in the network, as used in Section 4.5.3. ($X_3 = 1e - 6 \times [1; 1; 1]$)

Parameter	Value	Parameter	Value
AMFC (as in Algorithm 1 in Table 3.1)			
$B = R$	I_6	Q	I_6
k_1	1	k_2	0.01
Γ_1	$10 \times \text{diag}([X_3; 1; 1; 1])$	ρ_1	0.01
Γ_2	$0.01 \times \text{diag}([X_3; 1; 1; 1])$	ρ_2	0.01
CAMFC-2 (as in Algorithm 3 in Table 3.3)			
$B = R^i$	I_6	$Q^i = \kappa$	$10 \times I_6$
k_1	1	k_2	0.01
Γ_1	$1000 \times \text{diag}([X_3; 1; 1; 1])$	ρ_1	1
Γ_2	$0.1 \times \text{diag}([X_3; 1; 1; 1])$	ρ_2	0.1
$\mu = \lambda = \lambda_1$	10	$X_M = \Upsilon^M = U^M$	[1; 1; 1]

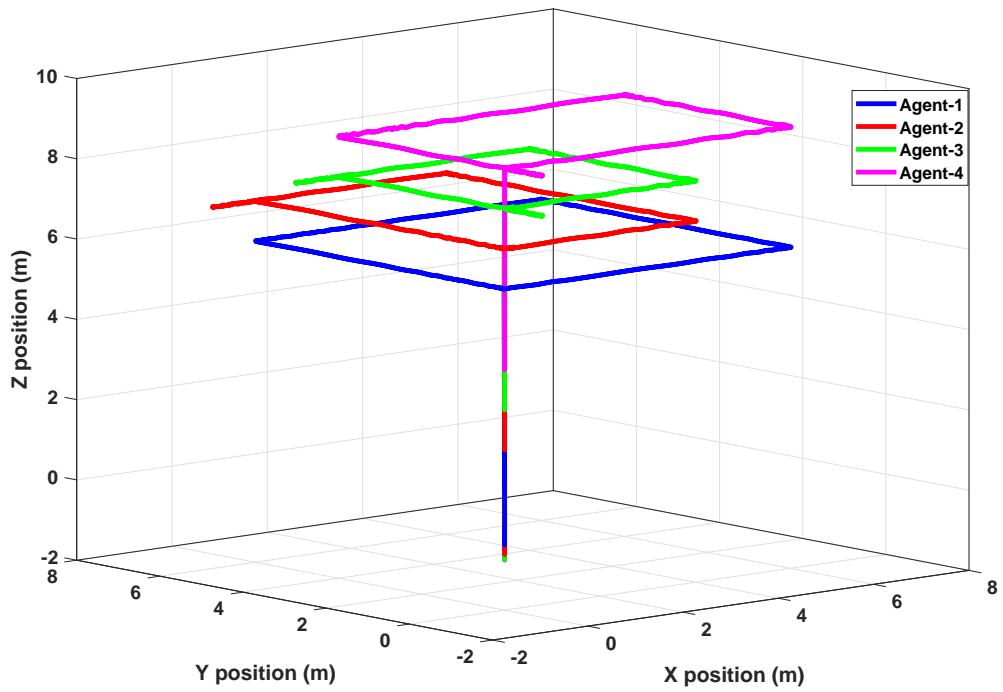


Figure 4.60: The 3D position of quadrotors in the network using the CAMFC-2 algorithm.

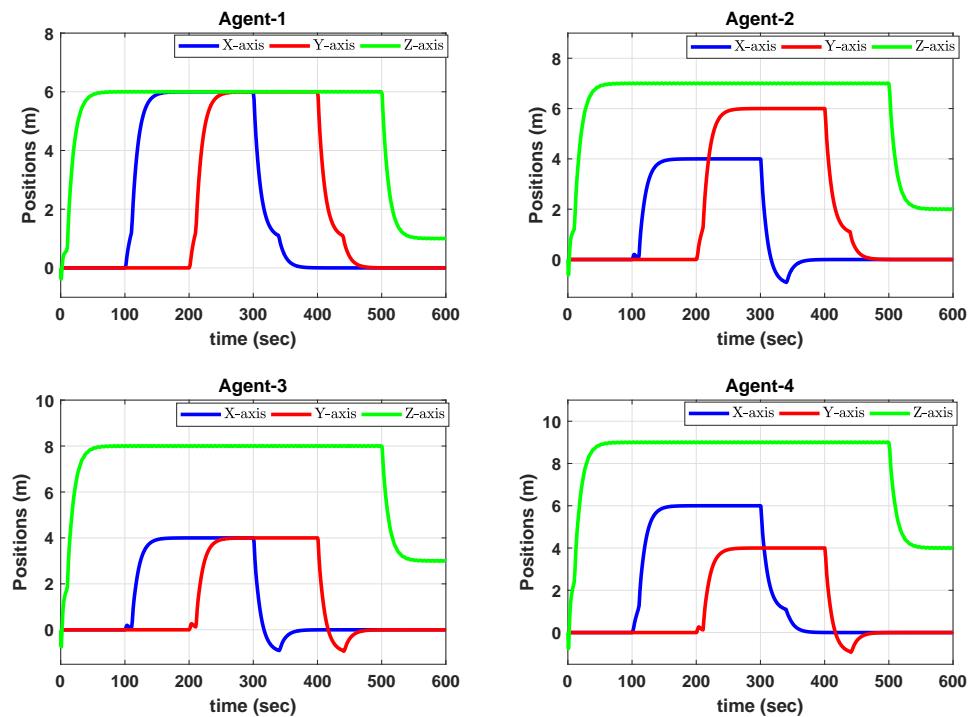


Figure 4.61: The X-Y-Z positions of quadrotors in the network using the CAMFC-2 algorithm.

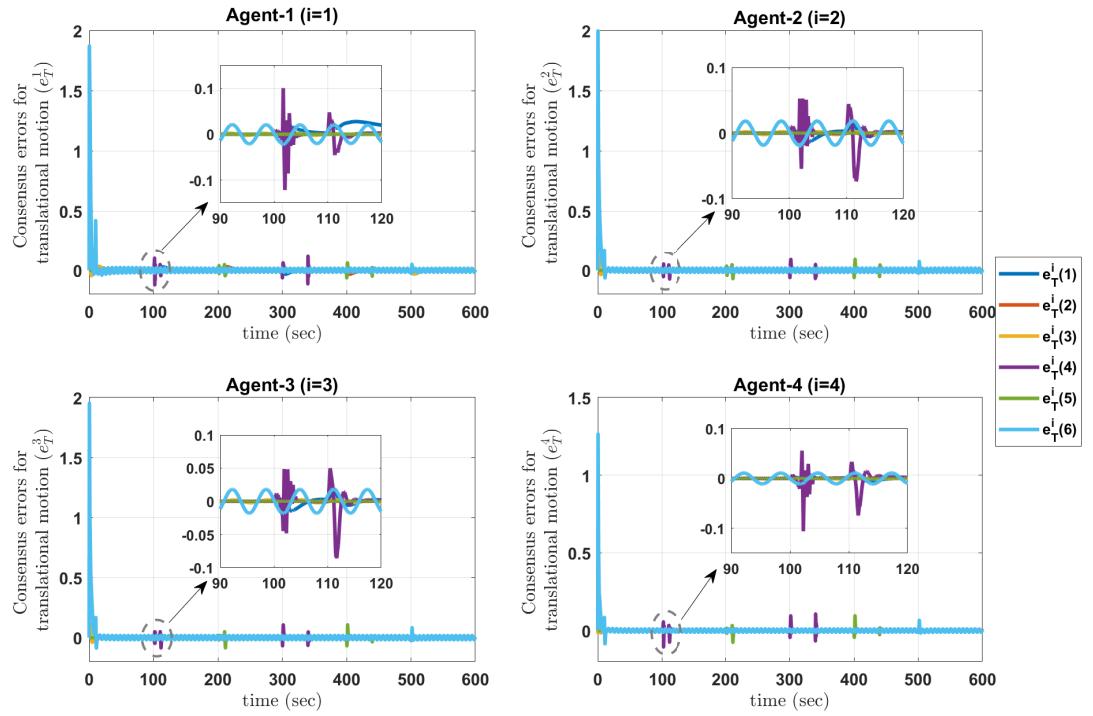


Figure 4.62: The consensus errors for translational motion of quadrotors in the network using the CAMFC-2 algorithm.

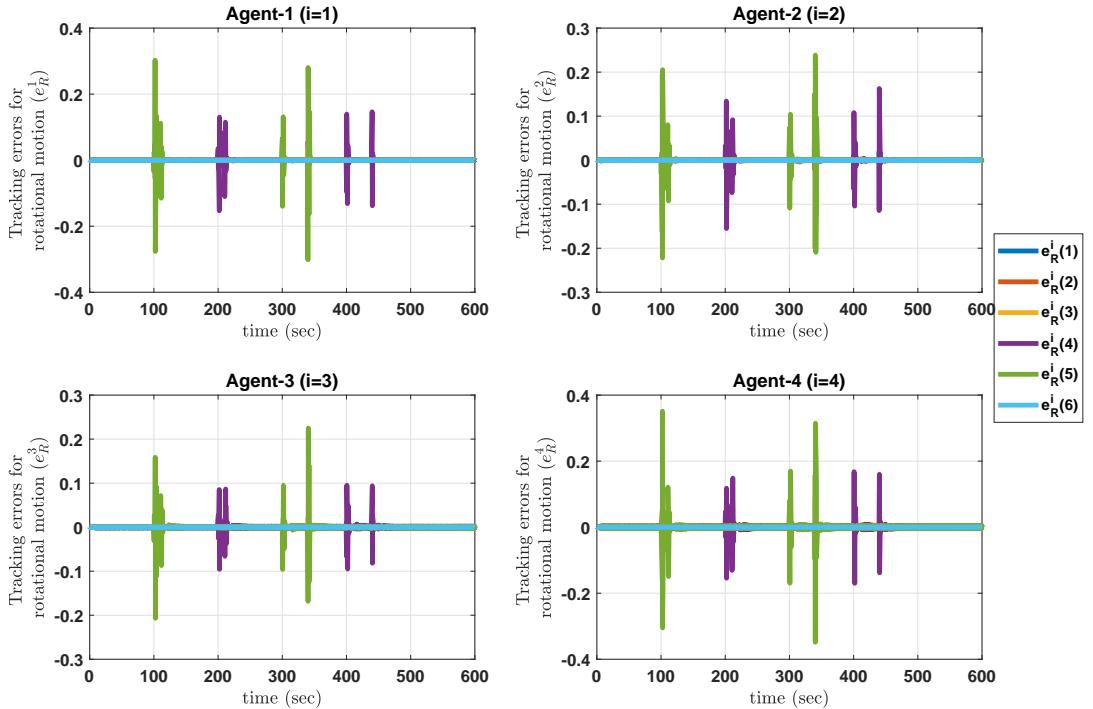


Figure 4.63: The tracking errors for rotational motion of quadrotors in the network using the AMFC algorithm.

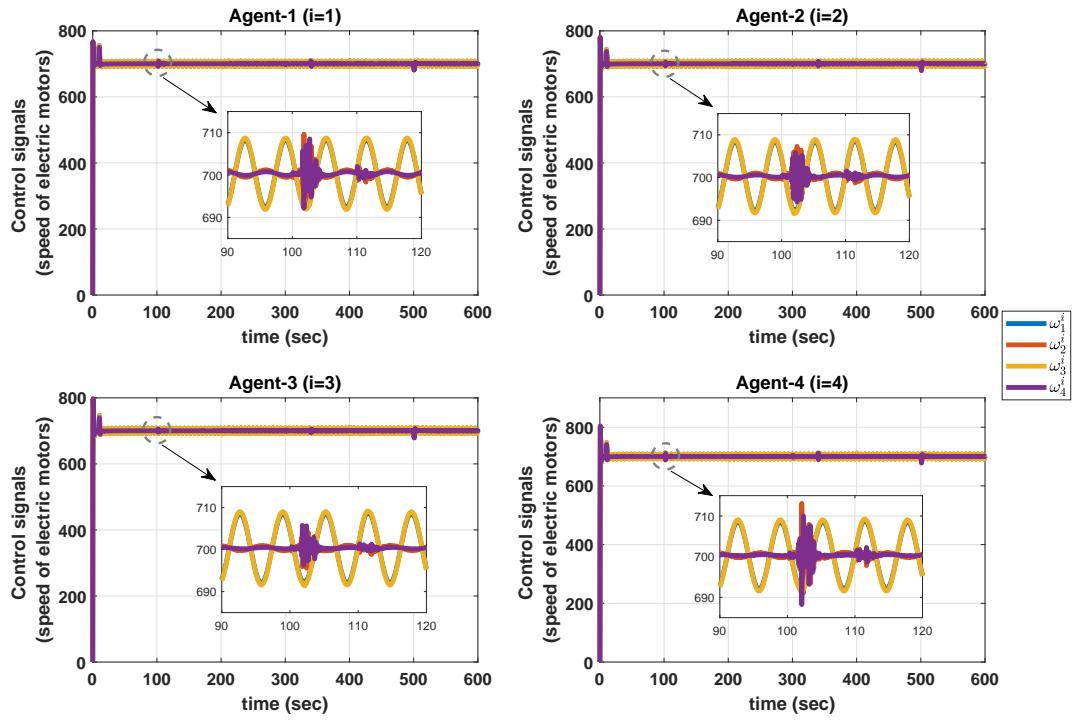


Figure 4.64: The control signals for speed of electric motors at all quadrotors in the network using CAMFC-2 algorithm.

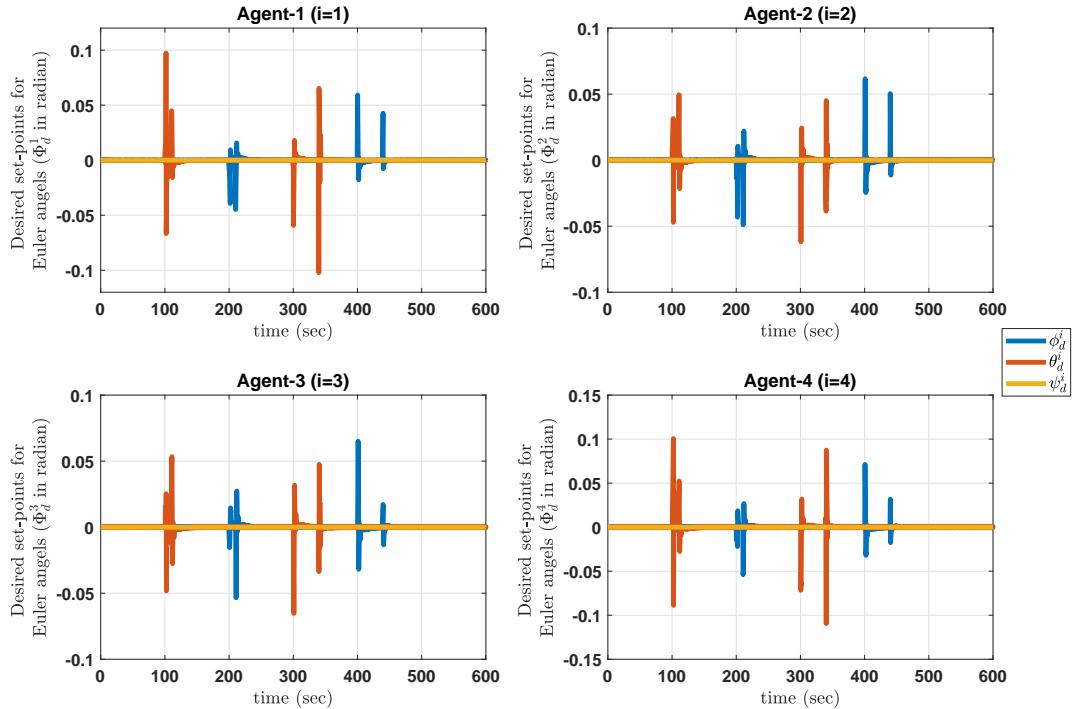


Figure 4.65: The set-points for the Euler angles of quadrotors in the network using CAMFC-2 algorithm.

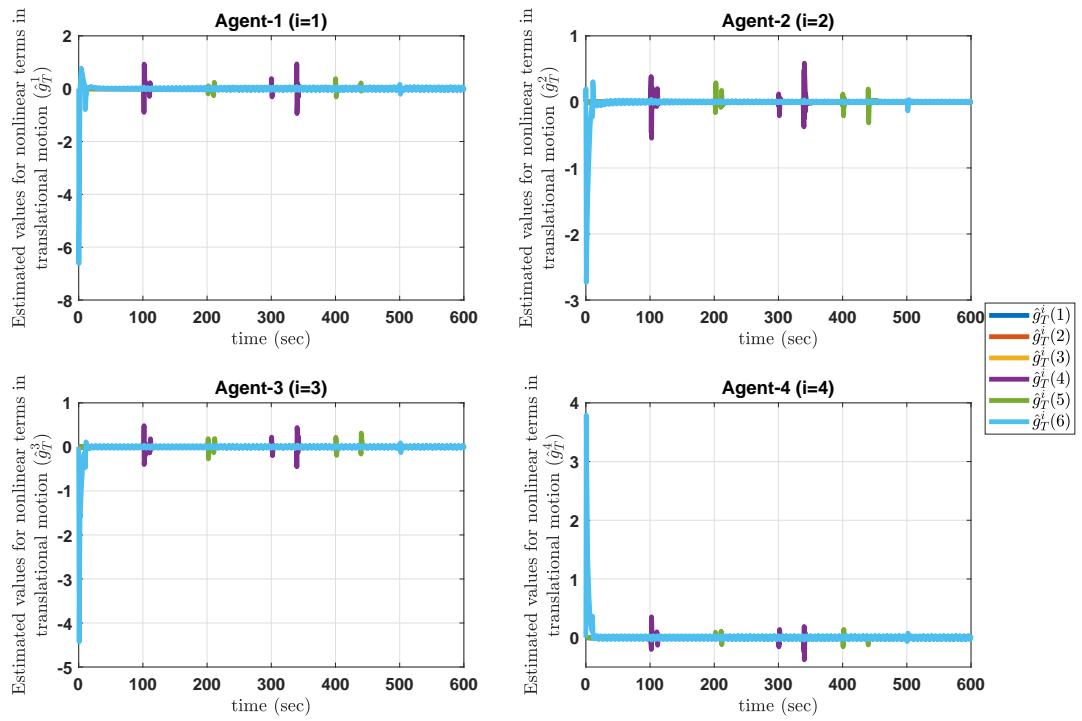


Figure 4.66: The estimated values of unknown nonlinear terms for translational motion of quadrotors in the network using CAMFC-2 algorithm.

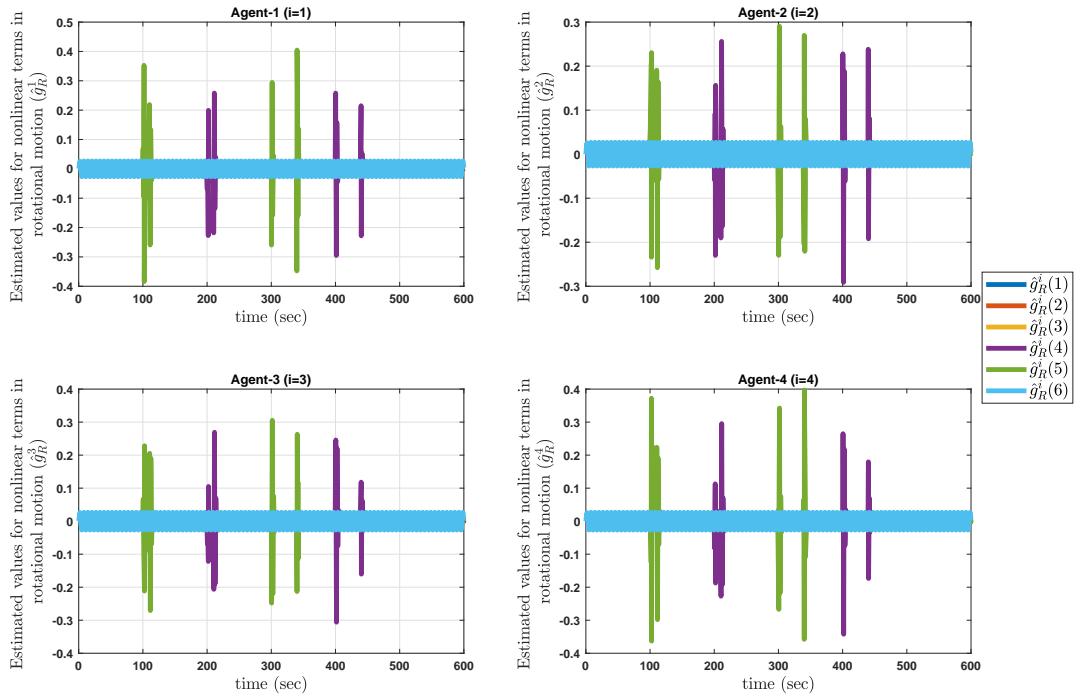


Figure 4.67: The estimated values of unknown nonlinear terms for rotational motion of quadrotors in the network using AMFC algorithm.

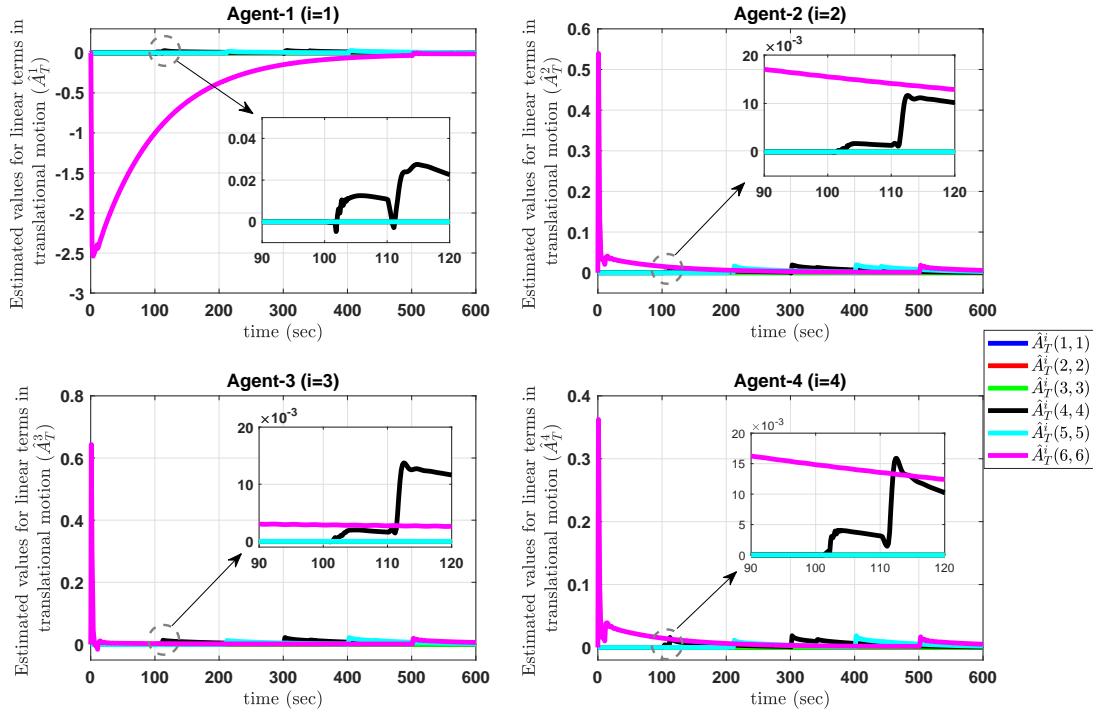


Figure 4.68: The estimated values of unknown linear terms for translational motion of quadrotors in the network using CAMFC-2 algorithm.

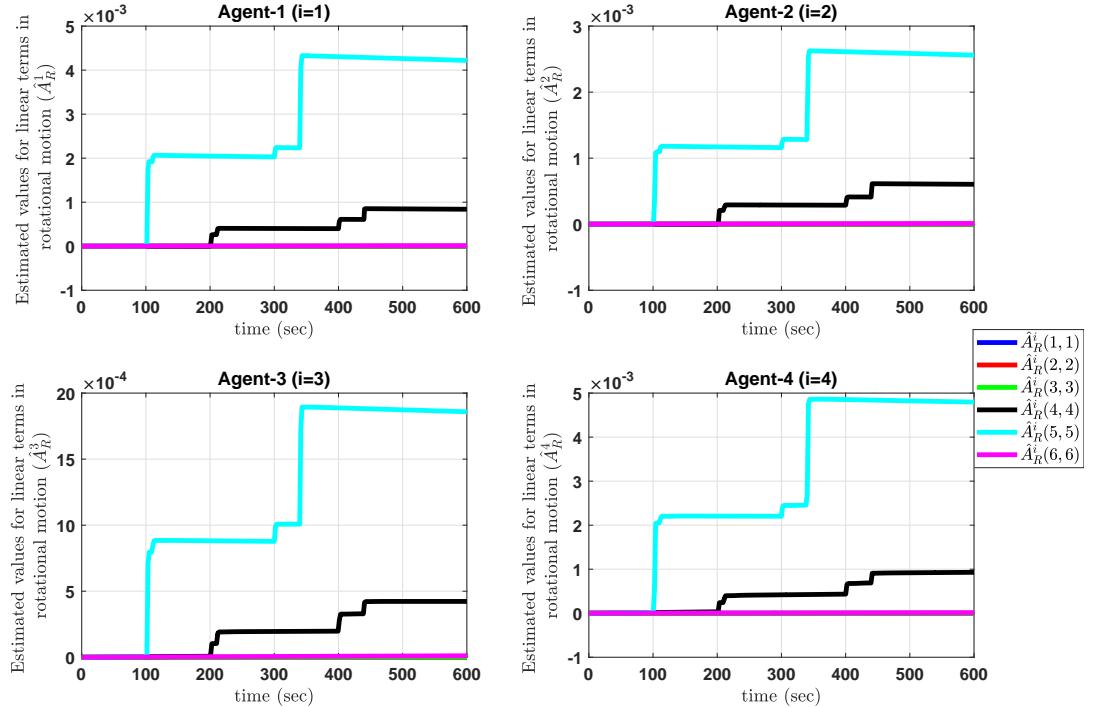


Figure 4.69: The estimated values of unknown linear terms for rotational motion of quadrotors in the network using AMFC algorithm.

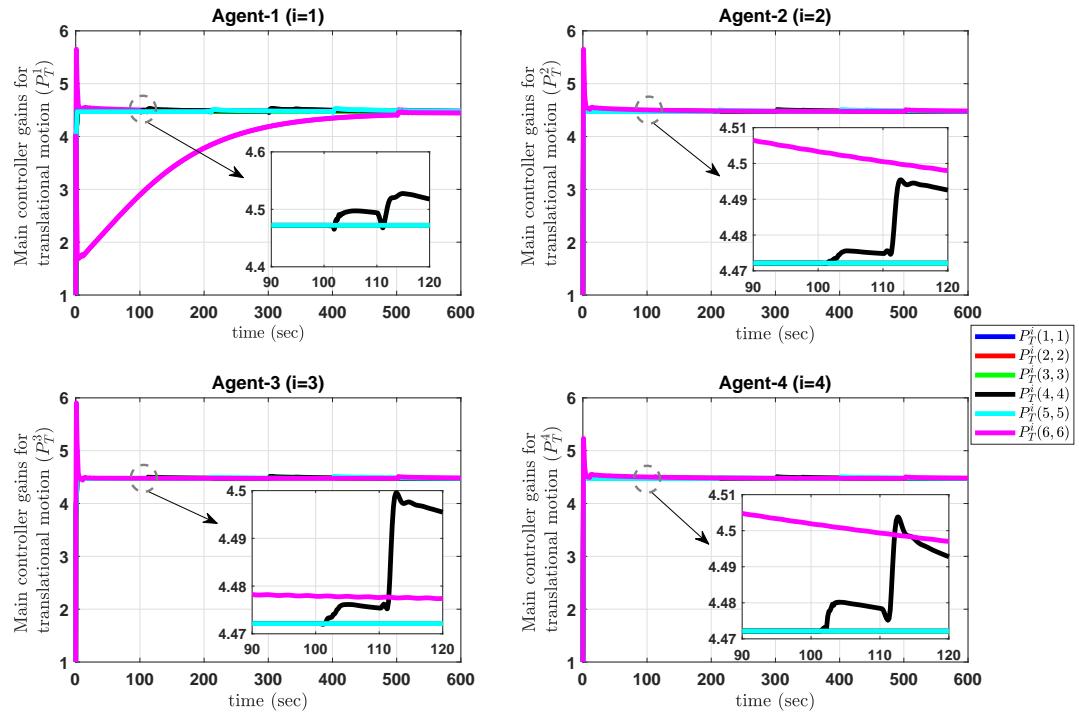


Figure 4.70: The values of main controller gains for translational motion of quadrotors in the network using CAMFC-2 algorithm.

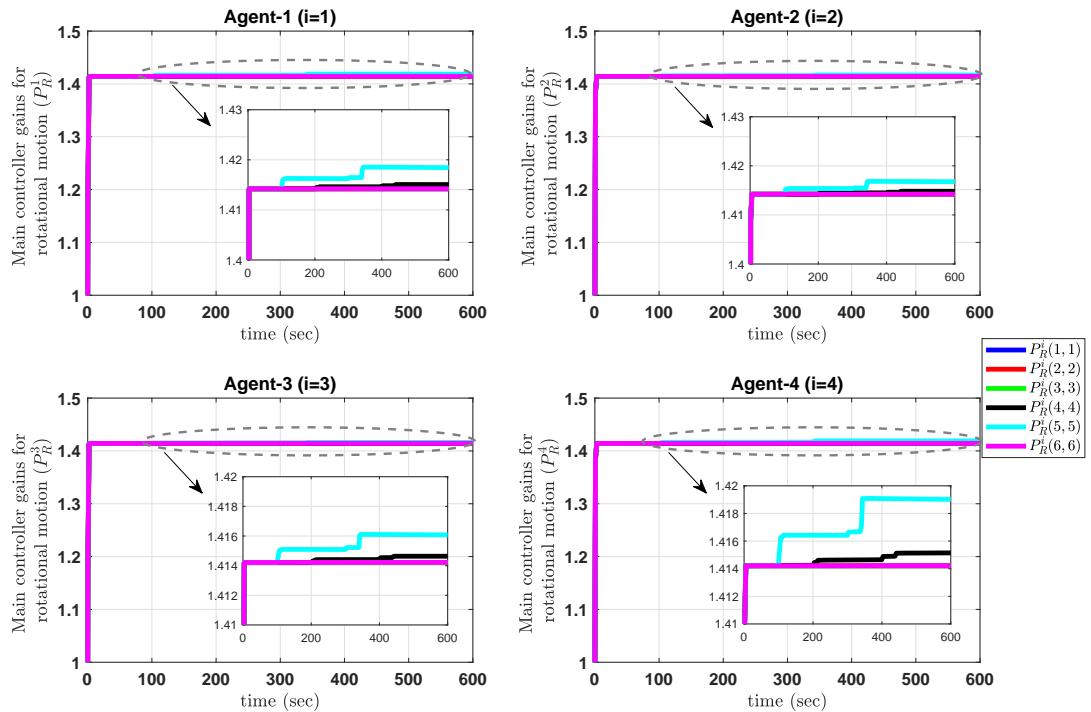


Figure 4.71: The values of main controller gains for rotational motion of quadrotors in the network using AMFC algorithm.

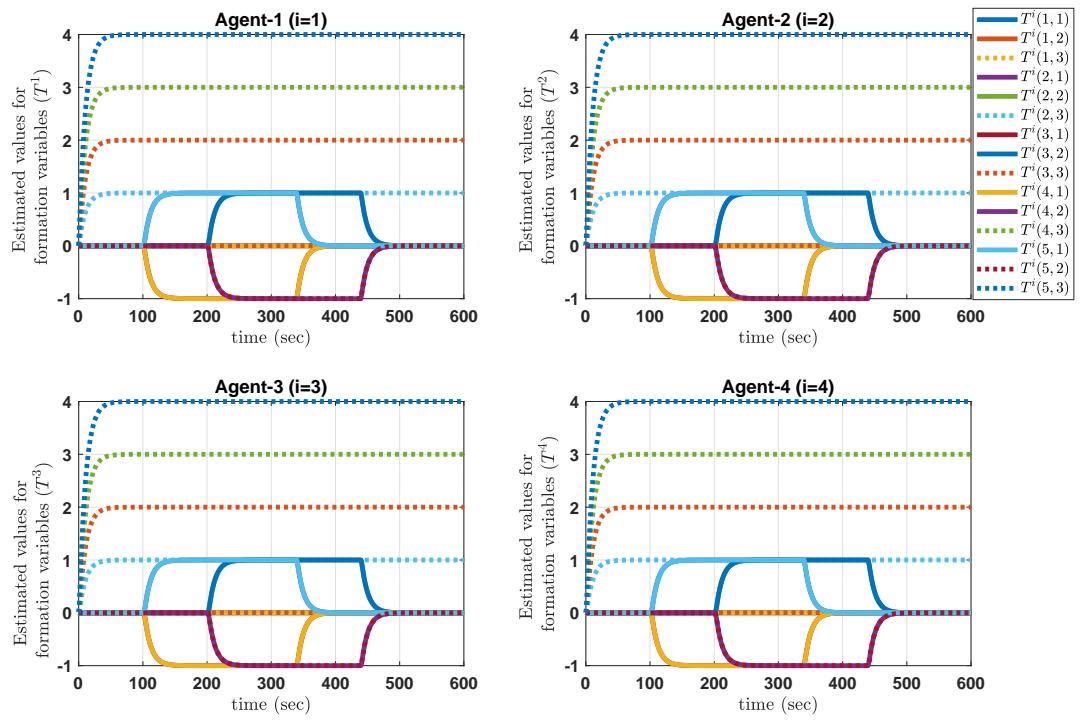


Figure 4.72: The estimated values of formation variables for translational motion of the quadrotors in the network using CAMFC-2 algorithm.

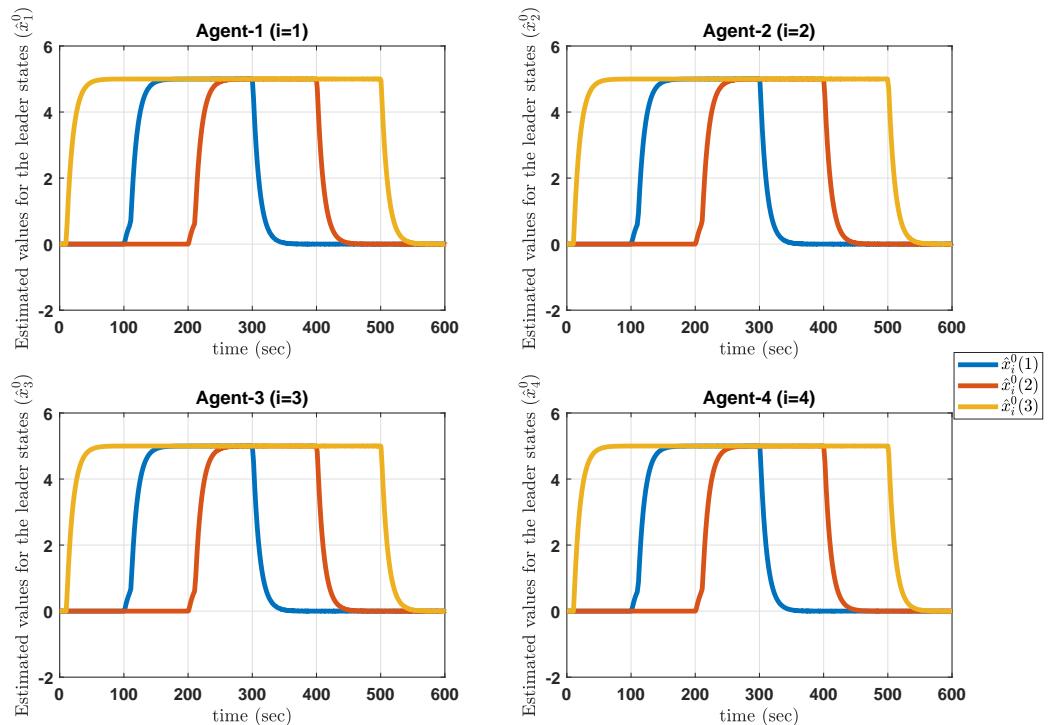


Figure 4.73: The estimated values of the leader states using CAMFC-2 algorithm.

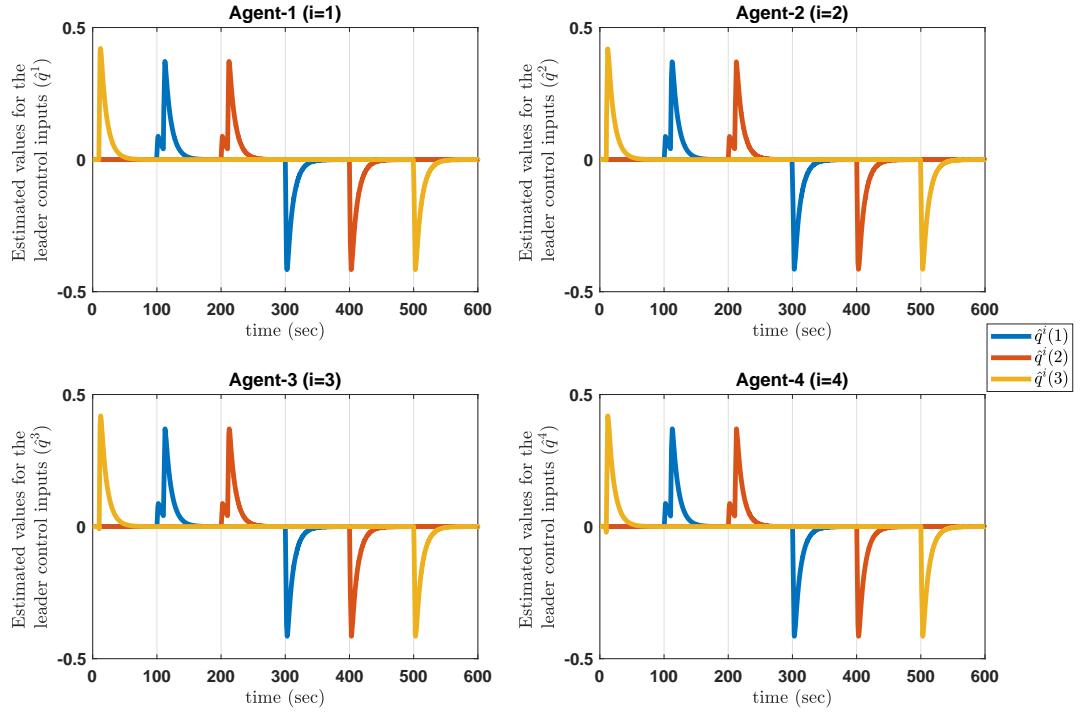


Figure 4.74: The estimated values of the leader control inputs using CAMFC-2 algorithm.

4.6 Results for ACL algorithm in a network of mobile agents

In this section, three different case studies are presented to evaluate the ACL algorithm. In the first two cases, the performance of the ACL algorithm is also compared with two recently presented localization solutions in the literature.

4.6.1 Comparison study for relative position estimation

This is the case-1 of the comparative studies provided for evaluating the performance of ACL algorithm. Suppose that we have a mobile agent in 2D environment with double-integrator dynamics as follows

$$\dot{p}^m = v^m , \quad (4.40a)$$

$$\dot{v}^m = u^m - v^m \sin(0.2t) + 0.5p^m \cos(0.1t) , \quad (4.40b)$$

where $p^m = [p_x^m; p_y^m] \in \mathbb{R}^2$, $v^m = [v_x^m; v_y^m] \in \mathbb{R}^2$ and $u^m = [u_x^m; u_y^m] \in \mathbb{R}^2$ are the position, velocity and the control input of the mobile agent, respectively. No on-board sensor to measure the absolute position on the mobile agent is assumed. Instead, the *relative distance* to the beacon agent can be measured using the UWB antennas located on both agents. Moreover, the velocity of the mobile agent can be computed locally according to *Assumption 3-6* (in Section 3-6). In addition, the velocity and absolute position of the beacon agent are transmitted to the mobile agent using a wireless communication link. The beacon agent has an on-board sensor to measure the absolute position with an acceptable level of accuracy. In this sense, the relative velocity and the relative distance between the mobile agent and the beacon are available at the mobile agent. Thus, the estimation algorithm proposed in (3.221) can be used to estimate the relative position of the mobile agent to the beacon. After that, the estimated absolute position of the mobile agent \hat{p}^m in a 2D environment can be computed online according to (3.247).

In this study, the mobile agent is using a simple back-stepping controller (Khalil, 1996) to satisfy the tracking objective for the dynamics model presented in (4.40a) and (4.40b), as follows

$$u^m = k_1^c e_c - k_2^c v^m , \quad (4.41)$$

where k_1^c and k_2^c are two positive constant scalars and $e_c = p^{des} - \hat{p}^m$ is the position tracking error (p^{des} is the desired position). The beacon agent also uses the same controller as presented in (4.41), but its corresponding position tracking error is computed using the measured absolute values, i.e. p_b . Note that the use of the controller in (4.41) can satisfy the conditions on the boundedness of the relative velocity and the relative distance of the two agents required in *Assumption 3-4* (in

Section 3-6). Here, the initial value for estimated position of the mobile agent is set at $[0; 0]$.

In order to evaluate the performance of the proposed adaptive relative position estimator in (3.221), the algorithm is compared with two other relative position estimation solutions. The first solution has an intuitive formula as (Safavi & Khan, 2017)

$$\dot{\hat{P}}_r = V_r . \quad (4.42)$$

Here, \hat{P}_r is the estimated vector of relative position and V_r is the measured vector of relative velocity. The second solution is presented as follows (Han et al., 2019)

$$\dot{\hat{P}}_r = [1 + \alpha_L(2d_r \dot{d}_r - 2V_r^T \hat{P}_r)]V_r , \quad (4.43)$$

where $\alpha_L \in \mathbb{R}^+$. The comparative study among the mentioned three algorithms is presented in Table 4.14 and Table 4.15 for three different desired trajectories of the moving agent. These trajectories are comprised of step inputs as $p_d^{(1)} = [5; -6]$, a square wave signal as $p_d^{(2)} = [Sq(5, 1000, 50); Sq(10, 1500, 50)]$; and also a sine wave as $p_d^{(3)} = [5 \sin(0.1t); 3 \sin(0.2t)]$. Here, $Sq(A_s, T_s, W_s)$ is a square wave signal with "A_s" amplitude, "T_s" duration and "W_s" pulse width percentage. Moreover, the study has been done for two cases with stationary and moving beacon agent. The desired position of the beacon agent is $p_d^b = [0; 0]$ for the first case with stationary beacon and $p_d^b = [2; -3]$ for the second simulation with moving beacon.

Furthermore, the initial position of moving agent is set at $p(0) = \hat{p}(0) = [0; 0]$ for all of the simulations and the parameters of the controller and the relative position

estimation algorithms are tuned according to Table 4.13. According to this table, the duration of simulation is different for different desired input signal, in order to have adequate time to gather significant data for analysing and comparing the results, for each simulation. Moreover, the tuning parameters for controller and the localization algorithms are adjusted so as to achieve the best acceptable results in the simulation results. As can be seen, the estimation algorithm in (4.42) does not have any tuning knob (obviously this is an advantage), while the other two estimators used in these simulation cases have one tuning knob, each; i.e. α_L is the tuning parameter for the relative position estimator in (4.43) and α_P is the tuning parameter for the relative position estimation algorithm proposed in this thesis (as in (3.221)).

Here, the cumulative values for positioning and tracking errors are considered for the comparison, as follows

$$C_1 = \int \xi^T \xi dt , \quad (4.44a)$$

$$C_2 = \int e_r^T e_r dt , \quad (4.44b)$$

where $\xi = p - \hat{p}$ is the absolute positioning error and $e_r = p_d - p$ is the tracking error based on the real position of the mobile agent. As can be observed in Table 4.14 and Table 4.15, while a similar tracking controller is used for all of the algorithms, the proposed adaptive relative position estimation algorithm in (3.221) has lower values for C_1 and C_2 parameters and consequently outperforms the two other algorithms. In particular for the $p_d^{(2)}$ input (which is the worst case among the others) in the simulation results with the stationary beacon agent (according to Table 4.14), the value of relative estimation error (i.e. C_1) is 57.9% lower than the value of this parameter for the algorithm in (4.42) and 52.2% lower the value corresponding to the

algorithm in (4.43). For the case of the moving beacon agent, these values would be 34.4% and 34.1%, respectively (according to Table 4.15). Achieving better relative position estimation performance with the estimator in (3.221) is led to having better tracking performance with exactly same controller, as it can be deduced by comparing the values for C_2 parameter in the provided tables.

One thing should be declared about the small values of C_1 associated with the estimator in (3.221), in Table 4.14 and Table 4.15, relative to the other two relative position estimators. As can be seen in (4.42) and (4.43), the parameter e_p (defined in (3.220)) doesn't included in the estimation. Hence, the sensitivity of these methods to the distance estimation error is low. But on the other hand, the parameter e_p is included in the proposed relative position estimation algorithm in the current thesis, leading to have more sensitivity about the distance estimation error. In this regard, not being aware about the distance estimation error provides small bias in the absolute positioning error (i.e. ζ) for the estimators in (4.42) and (4.43). This small bias is accumulated through the simulation time and generates large values for the parameter C_1 , as it is a time integration of the squared absolute positioning error (as defined in 4.44a). The values of absolute positioning errors for the three different relative position estimation algorithms with the moving beacon agent, are compared in Fig. 4.75 to confirm this statement.

Table 4.13: The values of tuning parameters for the controller (as in (4.41)) and the relative position estimating algorithms (as in (4.43) and (3.221))

Desired Trajectory	k_1^c	k_2^c	α_L	α_p	Simulation time (sec)
$p_d^{(1)}$	10	10	0.001	1	200
$p_d^{(2)}$	10	100	0.001	0.01	3000
$p_d^{(3)}$	10	10	0.01	0.01	2000

Table 4.14: Comparing the performance of three relative position estimating algorithms with a stationary beacon agent ($p_d^b = [0; 0]$). C_1 and C_2 are defined in (4.44a) and (4.44b).

Desired Trajectory	Eq. (4.42)		Eq. (4.43)		Eq. (3.221)	
	C_1	C_2	C_1	C_2	C_1	C_2
$p_d^{(1)}$	73.38	115.7	42.5	88.42	0.1565	49.3
$p_d^{(2)}$	12.66	2894	11.14	2891	5.327	2883
$p_d^{(3)}$	2182	2584	86.22	476.2	28.05	462.1

Table 4.15: Comparing the performance of three relative position estimating algorithms with a moving beacon agent ($p_d^b = [2; -3]$). C_1 and C_2 are defined in (4.44a) and (4.44b).

Desired Trajectory	Eq. (4.42)		Eq. (4.43)		Eq. (3.221)	
	C_1	C_2	C_1	C_2	C_1	C_2
$p_d^{(1)}$	20.23	66.9	18.69	65.49	0.1796	50.66
$p_d^{(2)}$	6.908	2887	6.872	2885	4.531	2887
$p_d^{(3)}$	1257	1647	463.2	831.2	56.62	455.6

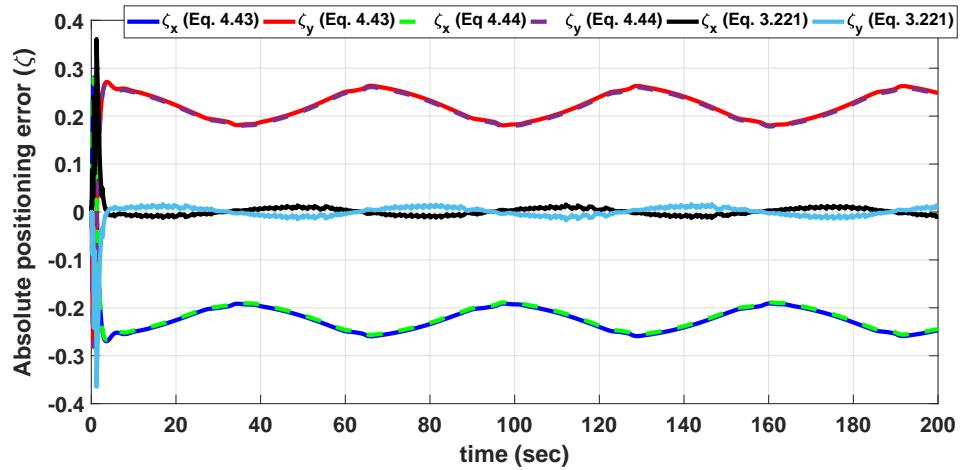


Figure 4.75: The absolute positioning estimation errors for all of the three presented relative position estimation algorithms.

4.6.2 Comparison study for cooperative localization in a network

This is the case-2 of the comparative studies provided for evaluating the performance of ACL algorithm. In this case study, there is a network of four mobile agents with the dynamic system in (4.40a) and (4.40b); and one stationary beacon agent in a 2D environment. Here, each of the agents has a local controller as presented in (4.41) and the control task at the agents is performed locally without any cooperative protocol among them. These mobile agents move according to their individual desired trajectory oblivious of their neighboring agents. Similar to the first case study in Section 4.6.1, here the mobile agents do not have any on-board sensor for measuring the absolute position. Instead, they communicate with the neighboring agents in the network so as to estimate their absolute position by using the ACL algorithm with the estimation laws presented in (3.221) and (3.255). The relative distance and the relative velocity to the neighboring agents can be determined using the data provided via the communication graph. The communication graph of the network (Fig. 4.76) satisfies the conditions requested in *Definition 3-19*. In this regard, there is a beacon agent which has a communication link with only one of the mobile agents in the network. The absolute position of the beacon agent is communicated to that mobile agent via the provided communication link. The corresponding adjacency and pinning gain matrix of the communication graph of the network are as follows

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathcal{B}_b = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4.45)$$

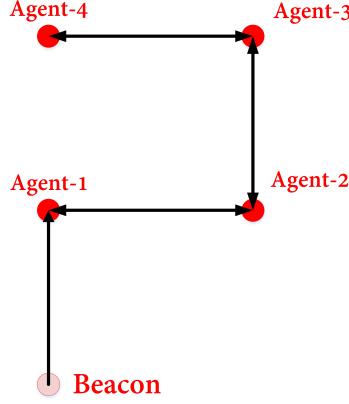


Figure 4.76: The communication graph among the agents in the network for implementing the ACL algorithm. The network incorporates only one beacon agent.

Here, the performance of the ACL algorithm, is compared with a recently investigated linear-convex (LC) algorithm for localization in a network of mobile robots which is presented in (Safavi & Khan, 2017). The algorithm in question, incorporates the triangulation method among the neighboring agents in the network and provides a trade-off between the triangulation and the intuitive relative position estimation proposed in (4.42). Based on (Safavi & Khan, 2017), the position of the i th mobile agent can be estimated by the following adaptive law (updating at k step)

$$\hat{p}_{k+1}^i = \alpha_S \hat{p}_k^i + (1 - \alpha_S) \left[\sum_{j=1}^3 a_k^{ij} \hat{p}_k^j \right] + v^i, \quad (4.46)$$

where $\alpha_S > 0 \in \mathbb{R}^+$ is a design parameter, if there is a triangulation set available around agent i . In addition, the values for $0 \leq a_k^{ij} \leq 1$ are the *barycentric* coordinates of agent i with respect to the neighboring agent j at step k and is computed as follows (Safavi & Khan, 2017)

$$a_k^{ij} = \frac{A_{\Theta_i}^j(k)}{A_{\Theta_i}(k)}, \quad (4.47)$$

in which Θ_i is the triangulation set around agent i including its three neighboring

agents. $A_{\Theta_i}(k)$ is value for area of the triangle produced by the three neighboring agents at the k th step, while $A_{\Theta_i}^j(k)$ is area of that triangle by removing the j th neighboring agent and replacing the agent i . The values for areas (or volumes in a 3D environment) are computed by exploiting the *Cayley-Menger* determinant (Safavi & Khan, 2017).

The LC algorithm needs to have communication links to three neighboring agents in order to utilize the triangulation technique, otherwise the estimation would be driven with only the velocity measurements, i.e v^l . In this sense, for a network of four mobile agents and one stationary beacon agent, there will be a requirement of seven undirected communication links among the agents in the network, including only one communication link between the beacon and agent-1. By contrast, the proposed ACL algorithm in this thesis requires four undirected inter-agent communication links (referring to Fig. 4.76). Lesser number of communication links requirement contribute significantly in general in context of resource consumption. Lower energy consumption and fewer number of hardware communication pose practical application advantage in context of multi-agent application.

Besides, as it is declared in Table 4.16, the tracking and positioning errors are much lower for the ACL algorithm in comparison with the LC algorithm suggested in (Safavi & Khan, 2017). In particular, the values of relative position estimation error (i.e. C_1) associated with the ACL algorithm are 98% at Agent-1, 97% at Agent-2, 96% at Agent-3 and 93% at Agent-4 lower than the values of this error with the use of LC algorithm. The lower relative position error reduces the tracking error at agents, having same controller for both localization algorithms. In fact, the value of tracking errors

(i.e. C_2) are 8% at Agent-1, 7% at Agent-2, 11.3% at Agent-3 and 8% at Agent-4 using the ACL algorithm.

Here, the simulations are performed for tracking the square wave desired trajectory at each agent and the values for tuning parameters are set same as in the second row of Table 4.13. Also, $\alpha_S = 0.25$ is used as suggested in (Safavi & Khan, 2017). The simulation results for this comparative study are presented in Fig. 4.77 to Fig. 4.82. It is evident that, by combining the ACL algorithm and the stable controller in (4.41), absolute positions of the mobile agents converge to the desired values, asymptotically (Fig. 4.79 and Fig. 4.80). Moreover, the absolute positioning errors are much smaller using ACL algorithm in comparison with the ones utilizing the LC algorithm (Fig. 4.77 and Fig. 4.78).

Table 4.16: Comparing the performance of the ACL algorithm and the LC algorithm, implemented on a network of non-cooperatively controlled mobile agents

Agent	LC (as in (2.22))		ACL (as in Algorithm 4 in Table 3.5)	
	C_1	C_2	C_1	C_2
Agent-1	4.447	156.1	0.0897	143.5
Agent-2	8.852	322.5	0.2683	299.4
Agent-3	10.7	241.4	0.4341	214.0
Agent-4	7.94	225.7	0.5311	207.7

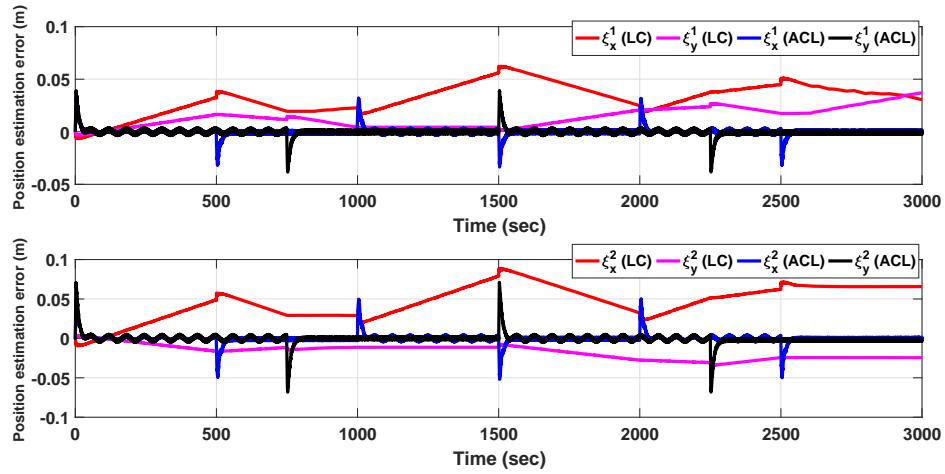


Figure 4.77: Comparing the position estimation errors for the ACL and LC algorithms, implemented on a network of non-cooperatively controlled mobile agents (Agent-1 and Agent-2)

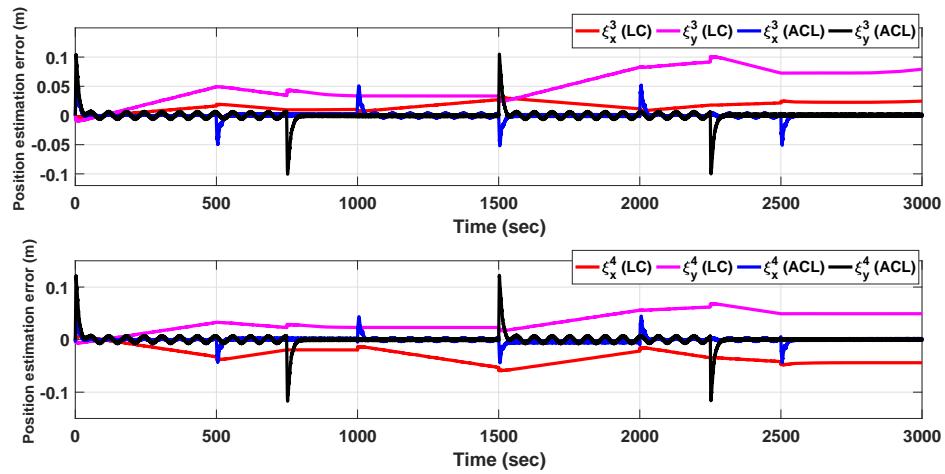


Figure 4.78: Comparing the position estimation errors for the ACL and LC algorithms, implemented on a network of non-cooperatively controlled mobile agents (Agent-3 and Agent-4)

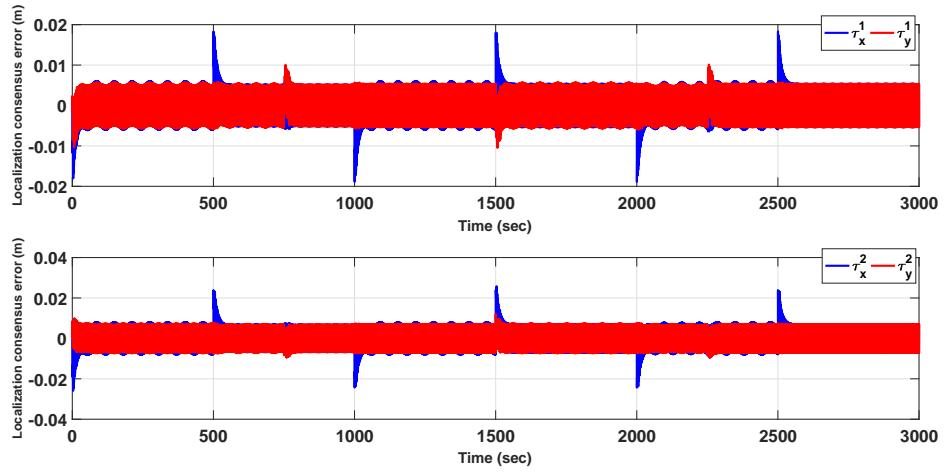


Figure 4.79: Values for the consensus error of the cooperative observer in the ACL algorithm, implemented on a network of non-cooperatively controlled mobile agents (Agent-1 and Agent-2)

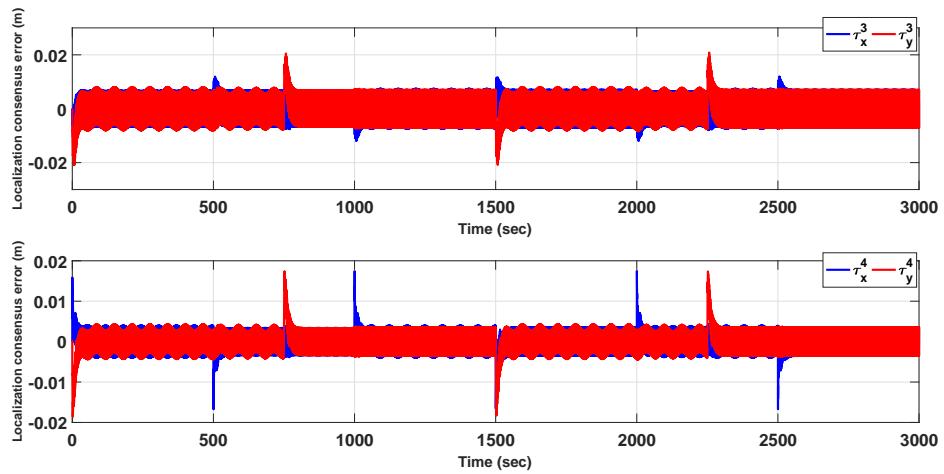


Figure 4.80: Values for the consensus error of the cooperative observer in the ACL algorithm, implemented on a network of non-cooperatively controlled mobile agents (Agent-3 and Agent-4)

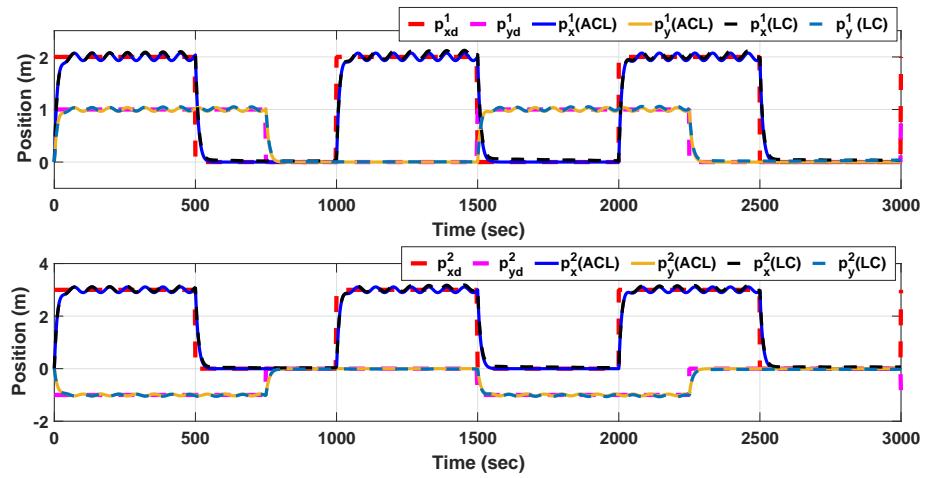


Figure 4.81: Comparing the estimated absolute positions with the ACL and LC algorithms, in a network of non-cooperatively controlled mobile agents (Agent-1 and Agent-2)

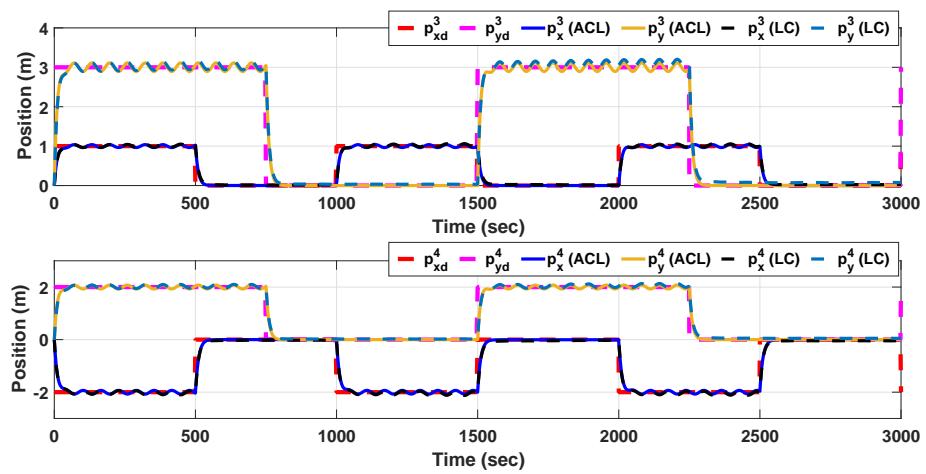


Figure 4.82: Comparing the estimated absolute positions with the ACL and LC algorithms, in a network of non-cooperatively controlled mobile agents (Agent-3 and Agent-4)

4.6.3 Application of ACL and CAMFC-2 algorithms in a team of quadrotors

In this section, performance of the ACL algorithm for achieving the localization objective is evaluated in a network of cooperatively controlled autonomous quadrotors in a 3D environment. In this sense, the four quadrotors which are simulated in Section 4.5.3 and the desired formation-tracking objective has been achieved for them using the decentralized cooperative controllers defined by CAMFC-2. The relative position of the neighboring agents is computed using the estimation algorithm suggested in (3.221). In addition, their absolute positions in 3D space are estimated using the ACL algorithm.

The simulation study carried out in this section incorporates the same setting as in Section 4.5.3, in terms of network properties, desired trajectory and the desired formation topology. But in the current simulation, a quadrotor beacon also acting as the leader is added in the network. The beacon quadrotor and all of the other four quadrotors in the network are supposed to have the dynamics as presented in (4.24a) to (4.24d). Moreover, the beacon quadrotor is controlled with the AMFC policies implemented on both rotational and translational motions, as suggested in (3.48a) and (3.48b) to achieve its own tracking objective. The other four quadrotors use the cascade control scheme in which the translational motion is controlled with CAMFC-2 algorithm, while the AMFC algorithm is used for rotational motion of each quadrotor. This is the same scheme incorporated in Section 4.5.3. The simulation results for this study are depicted in Fig. 4.83 to Fig. 4.87.

The values for tuning parameters in the ACL algorithm at each of the autonomous quadrotors, the AMFC algorithm used at the beacon quadrotor and the rotational

motion of other quadrotors, and the CAMFC-2 algorithm employed for the translational motion at the quadrotors are presented in Table 4.17.

By incorporating the CAMFC-2 and ACL algorithms in a network of quadrotors, the position estimation errors (as in Fig. 4.83), the localization consensus errors (as in Fig. 4.84), and the consensus errors for formation-tracking objective (as in Fig. 4.85) at all agents in the network, are converged and bounded in small sets around the origin. By employing the relative position estimating formula presented in (3.221) for determining the relative position of neighboring quadrotors in the network, the desired formation-tracking objective is satisfied, as observed in Fig. 4.86. Note that only relative measurements of translational velocities are required among the neighboring quadrotors in the network, in order to provide the estimated values of the absolute positions for all quadrotors in the network in a 3D space (Fig. 4.87).

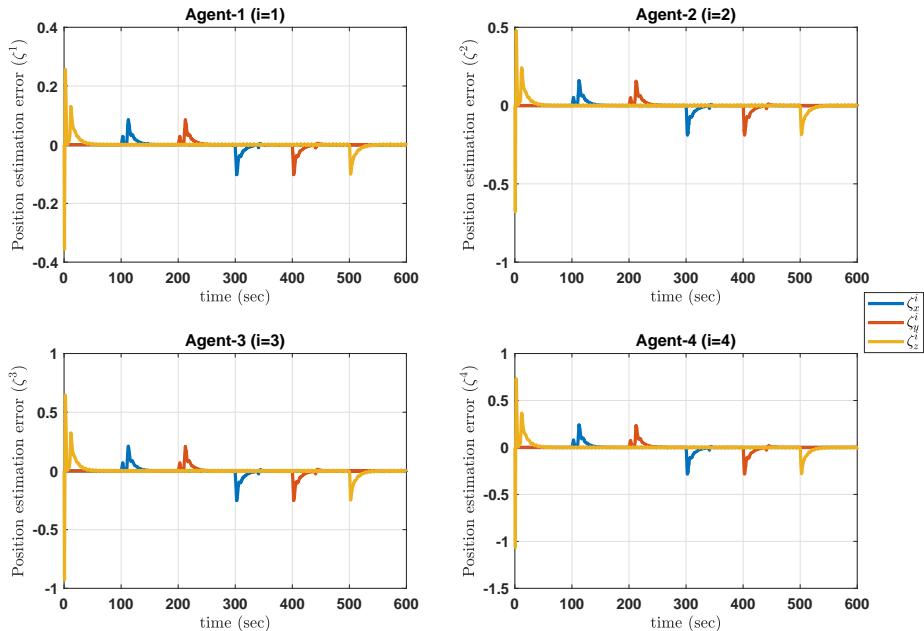


Figure 4.83: Position estimation errors for ACL algorithm used in a team of cooperatively controlled quadrotors. Here, $\zeta = p - \hat{p}$. Refer to Section 3.6 and Section 4.6.1 for more information.

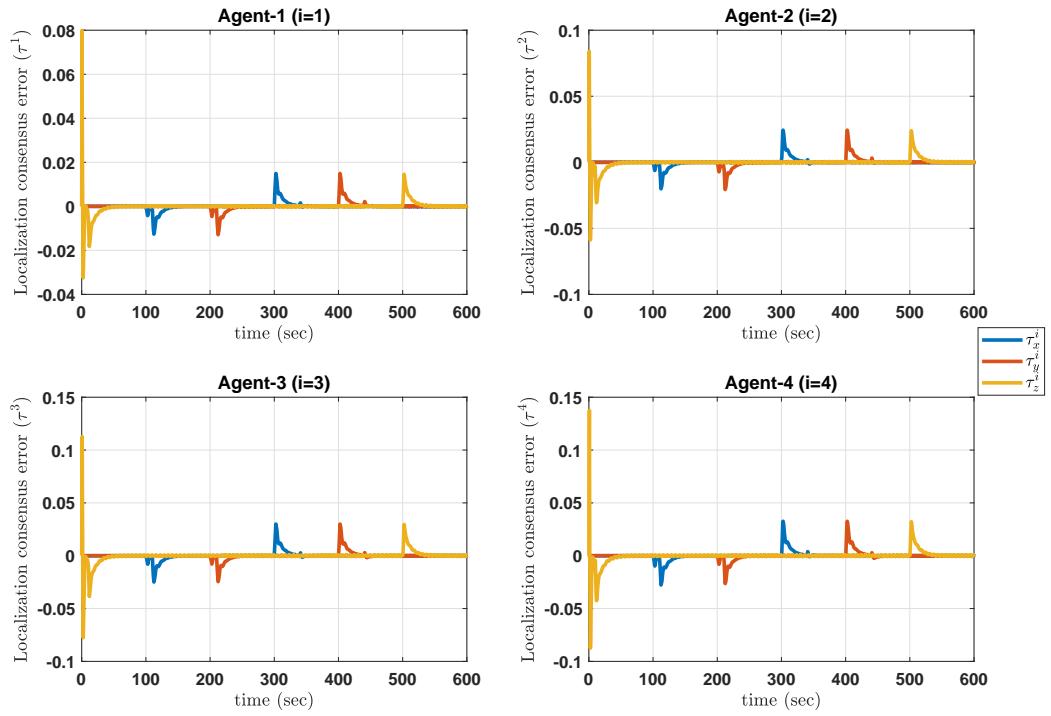


Figure 4.84: Localization consensus errors for ACL algorithm used in a team of cooperatively controlled quadrotors.

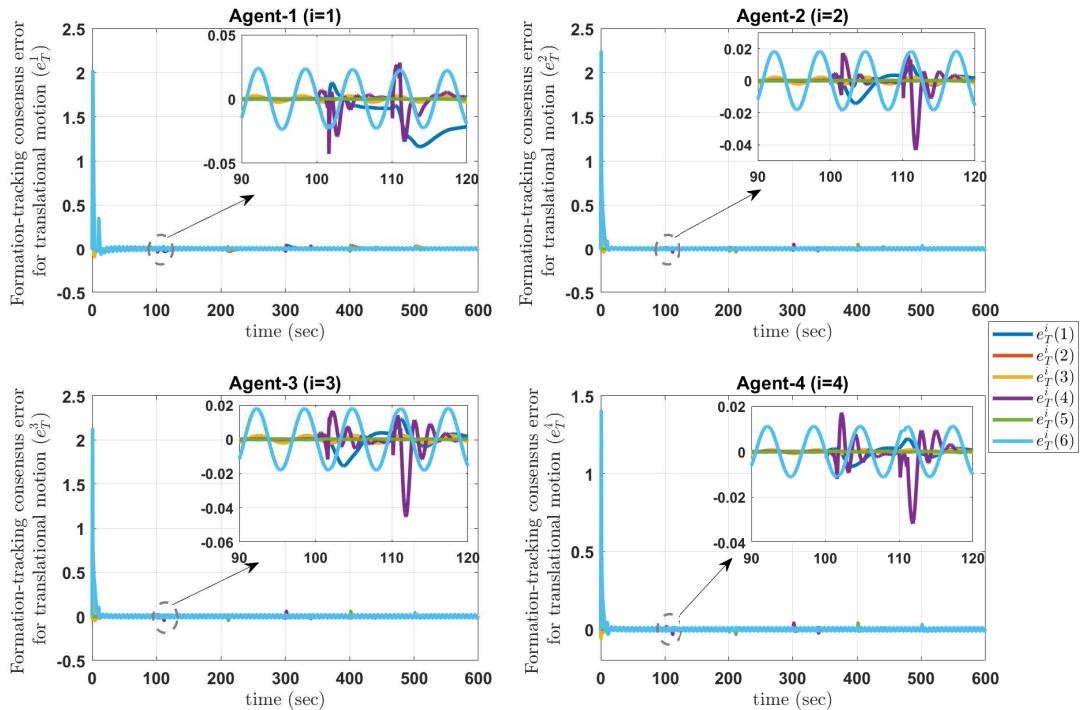


Figure 4.85: Formation-tracking errors for a team of quadrotors controlled with the CAMFC-2 algorithm using the relative position estimating algorithm for providing the relative position values.

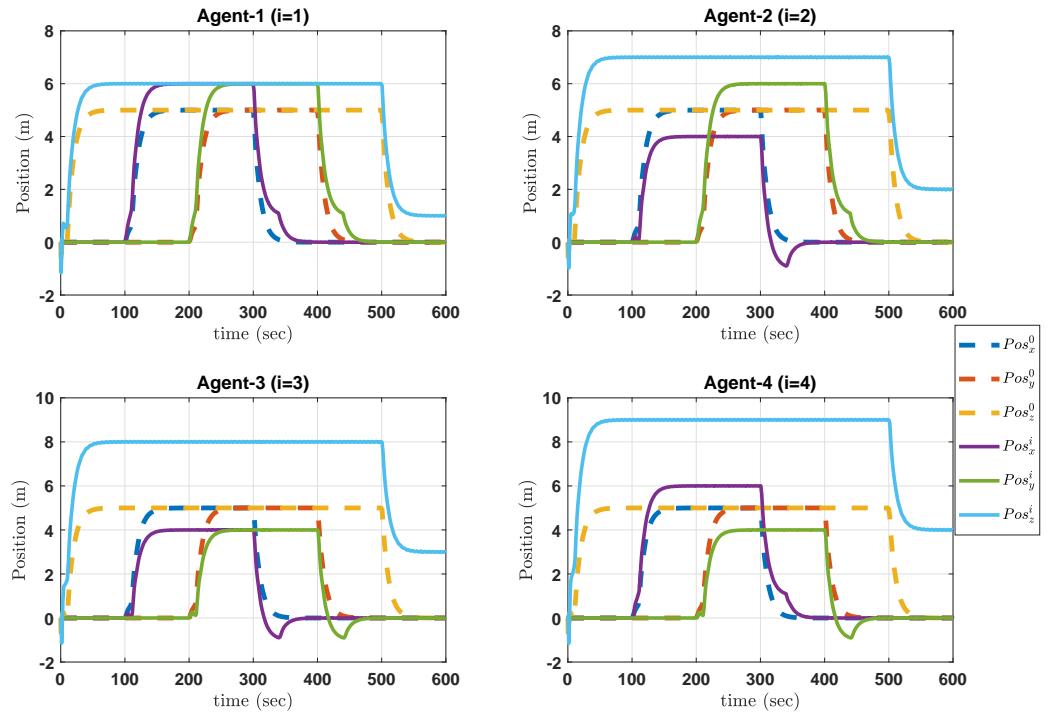


Figure 4.86: Absolute position of the quadrotors and the leader quadrotor in a network controlled with CAMFC-2 algorithm, using ACL algorithm for localization.

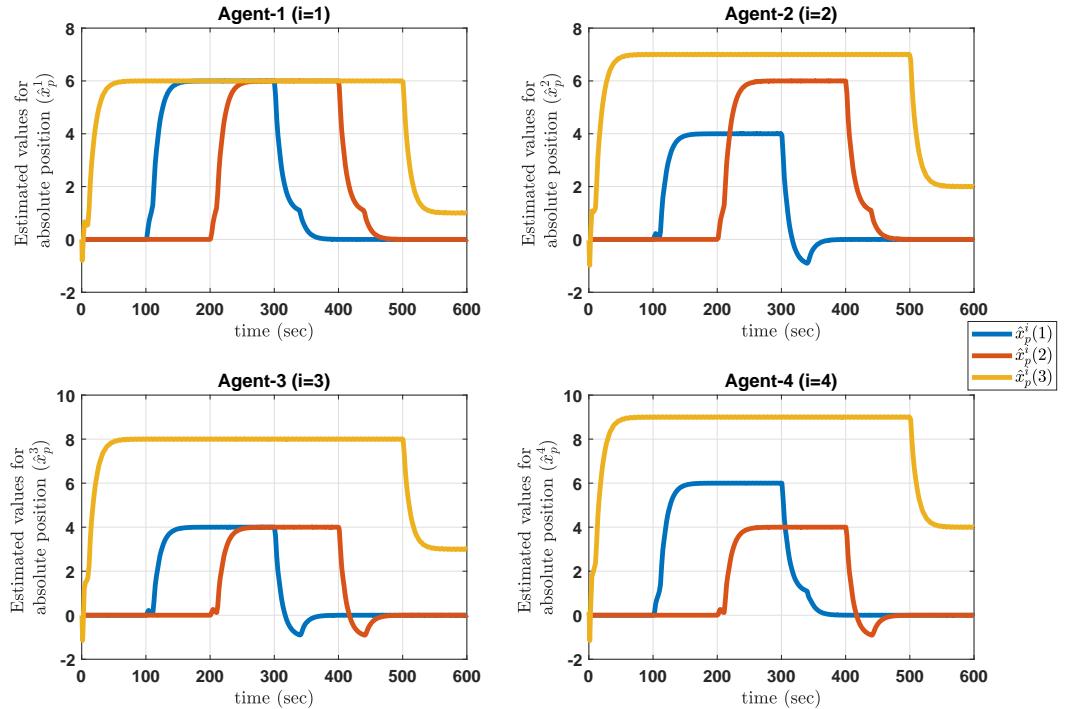


Figure 4.87: Estimated values for absolute position of the quadrotors in the network using ACL algorithm.

Table 4.17: The tuning parameters for the AMFC algorithm at the beacon quadrotor and the rotational motion of follower quadrotors, the CAMFC-2 at translational motion of follower quadrotors and the ACL algorithm in the network. ($X_3 = 1e-6 \times [1; 1; 1]$)

Parameter	Value	Parameter	Value
AMFC at leader			
(as in Algorithm 1 in Table 3.1)			
$B = R$	I_6	Q	$0.1 \times I_6$
k_1	10	k_2	0.01
Γ_1	$100 \times \text{diag}([X_3; 1; 1; 1])$	ρ_1	1
Γ_2	$1 \times \text{diag}([X_3; 1; 1; 1])$	ρ_2	1
AMFC at follower			
(as in Algorithm 1 in Table 3.1)			
$B = R$	I_6	Q	I_6
k_1	10	k_2	0.01
Γ_1	$10 \times \text{diag}([X_3; 1; 1; 1])$	ρ_1	0.01
Γ_2	$0.01 \times \text{diag}([X_3; 1; 1; 1])$	ρ_2	0.01
CAMFC-2			
(as in Algorithm 3 in Table 3.3)			
$B = R^i$	I_6	$Q^i = \kappa$	$10 \times I_6$
k_1	10	k_2	0.01
Γ_1	$1000 \times \text{diag}([X_3; 1; 1; 1])$	ρ_1	1
Γ_2	$0.1 \times \text{diag}([X_3; 1; 1; 1])$	ρ_2	0.1
$\mu = \lambda = \lambda_1$	10	$X_M = \Upsilon^M = U^M$	[1;1;1]
ACL			
(as in Algorithm 4 in Table 3.5)			
α_p	0.01		
λ_2	10	M_p^b	[1;1;1]

4.7 Hardware-in-the-loop test for implementation of the AMFC algorithm in AMRs

In this section, performance of the AMFC algorithm as the core solution of the current thesis for tracking problem in AMRs is evaluated using a hardware-in-the-loop (HIL) test platform.

4.7.1 HIL test setup

The HIL test setup constitutes two development boards. The AMFC algorithm is implemented on one of the boards (board-A), while the other board (board-B) is

emulating the dynamics of a real mobile robot (either WMR or quadrotor) based on a proven dynamic model. The dynamic models for the WMR and the quadrotor are same as the ones presented in Section 4.3.4 and Section 4.3.5, respectively. But, since the purpose of HIL test is not the validation over the simulation results, different desired trajectories with respect to the ones used in Section 4.3.4 and Section 4.3.5 are utilized here for HIL test. The scheme of the HIL test setup is shown in Fig. 4.88. Here, the code implementing the AMFC algorithm in board-A is optimized in the way to have the least possible computation cost for the algorithm. The forward Euler integration is employed for numerical integration. The communication between board-A and board-B is provided using the I²C protocol. Here, the speed of I²C communication is set to 400 *KB/s* and the maximum buffer size for the communicated message in each single loop of the algorithm is 64 bytes. In addition, the data is logged during the HIL test in MATLAB using serial communication between the development boards and a laptop. The baud rate of the serial communication is set to 115 200 *KB/s*, in order to ensure fast data logging process and thereby the sampling time for the algorithm and the emulated dynamic model can be conveniently lowered. The achieved sampling time for the HIL test is about 5 millisecond.

It should be declared that all of the parameters in the dynamic model of a WMR are considered to be unknown for implementation of the AMFC algorithm (including k_e , l_w and r_w in (4.17a) and (4.17b)). In the case of quadrotor, the values for all of the parameters are unknown (including k_t , m_q , j_q and l_q in (4.25)) except the value of k_l which is the constant for computing the lift force of each propeller based on its rotational speed. This value can be determined in an initial test (measuring the angular speed of electric motors, while the quadrotor is hovering), just before the real

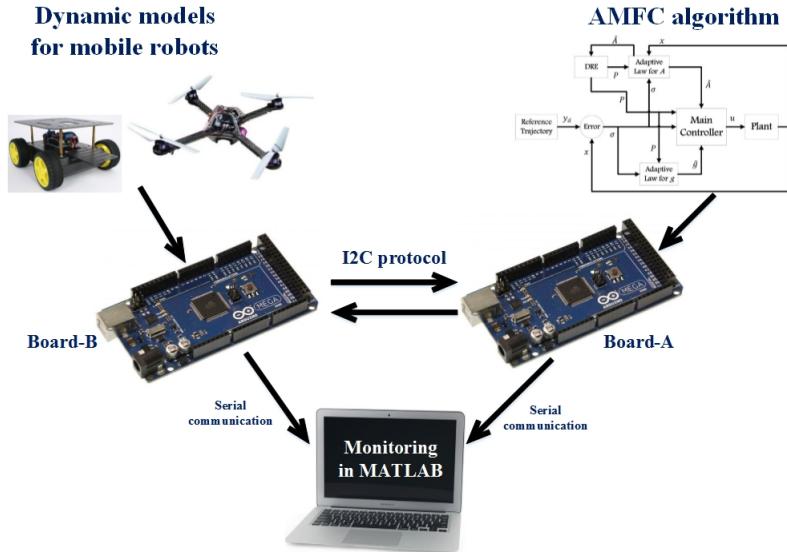


Figure 4.88: The schematic of HIL test setup for real-time evaluation of the AMFC algorithm on AMRs

implementation of the AMFC on a quadrotor. Moreover, the external disturbances on the WMR model (as in (4.21c) and in (4.21d)), are considered in the form of $f_w = \alpha_w \sin(0.1t)$ and $t_w = 0.1 \times \alpha_w \sin(0.1t)$, for $\alpha_w > 0$ as a parameter to study different amplitudes of the disturbances on the WMR model. The external disturbance on the quadrotor model (as in (4.24c) and in (4.24d)) are defined by $\vec{f}_q = \sin(0.3t)\vec{v}_1$ and $\vec{t}_q = \sin(0.3t)\vec{v}_3$, where $\vec{v}_1 = [0; 0; \alpha_q]$ and $\vec{v}_3 = \alpha_q \times [1; 1; 1]$. Similarly, $\alpha_q > 0$ is the parameter for analyzing different amplitudes of the disturbances on the quadrotor model.

4.7.2 State observer for AMRs

The AMFC algorithm can be used only for handling the unknown disturbances and the unknown internal dynamics of an AMR. In order to eliminate the random measurement noise of the measured data received from the on-board sensors, one should use state observers or filters. Here, the states of an AMR is divided into two parts as the first part includes the linear displacement and the linear velocity, while the

second one constitutes the angular displacement and the angular velocity. It should be noted that the presented observers in this section are not among the problem statements as well as the contributions of the current thesis and they are just adopted from previously published works. The corresponding references are cited wherever it was required.

For part one, a simple Kalman-filter based on the linear kinematics of the AMRs is used for observing the states (Cui et al., 2016). It is assumed that the absolute or local linear displacement of an AMR can be measured using GPS or UWB sensors and there is not any on-board sensor for measuring the local or absolute linear velocity of the mobile robot. On the other hand, there is an IMU available for measuring the linear acceleration. In this regard, the general kinematics of the linear motion for a mobile robot can be considered as follows (Cui et al., 2016)

$$\dot{\hat{x}} = A_o \hat{x} + B_o u_o + G_o w_o , \quad (4.48a)$$

$$z_o = H_o \hat{x} + v_o , \quad (4.48b)$$

where $\hat{x}_o \in \mathbb{R}^{n_l \times 1}$ is the observed system states of linear motion (including the linear displacement and the linear velocity), $u_o \in \mathbb{R}^{r_l \times 1}$ is the measured dynamics of system states (the linear acceleration) and $z_o \in \mathbb{R}^{p_l \times 1}$ is the measured states of the system. Here, $n_l \in \mathbb{R}^+$ is the number of system states in linear motion of the AMR (i.e. position and velocity), $r_l \in \mathbb{R}^+$ is the number of measured system dynamics (i.e. the acceleration) and $p_l \in \mathbb{R}^+$ is the number of available measurements among the total system states in the linear motion (i.e. the position). In (4.48a) and (4.48b), $A_o \in \mathbb{R}^{n_l \times n_l}$ is the known state matrix for the system kinematics, $B_o \in \mathbb{R}^{n_l \times r_l}$ is the

dynamics measurement matrix, $G_o \in \mathbb{R}^{n_l \times n_l}$ is the input matrix for the noise on dynamic measurements and $H_o \in \mathbb{R}^{p_l \times n_l}$ is the measurement matrix. These matrices are known and are all depending on the linear kinematics of the AMR. The specific values of the variables in (4.48a) and (4.48b) for the case the WMR and quadrotor are presented in Table 4.18. In this table, $\mathbf{0}_3$ is a matrix in $\mathbb{R}^{3 \times 3}$ with all elements are equal to zero, while $\mathbf{I}_3 \in \mathbb{R}^{3 \times 3}$ and $\mathbf{I}_6 \in \mathbb{R}^{6 \times 6}$ are identity matrices. In addition, the variables with superscript m are the measured variables using the on-board sensors. Note that w_o and v_o are vectors of random noise, exist in the state and output equations, respectively.

The equation in (4.48a) can be considered as the process model for implementing the standard Kalman-filter, while the equation in (4.48b) is the measurement model. Taking this into account, a continuous Kalman-filter is suggested as follows (Lewis et al., 2007)

$$\dot{P}_o = A_o P_o + P_o A_o^T + G_o Q_o Q_o^T - P_o H_o^T R_o^{-1} H_o P_o , \quad (4.49a)$$

$$\dot{\hat{x}} = A_o \hat{x} + B_o u_o + P_o H_o^T R_o^{-1} (z_o - H_o \hat{x}) , \quad (4.49b)$$

where $P_o \in \mathbb{R}^{n_l \times n_l}$ is the gain matrix for the proposed Kalman-filter. Moreover, $Q_o \in \mathbb{R}^{n_l \times n_l}$ and $R_o \in \mathbb{R}^{p_l \times p_l}$ are the positive definite matrices for tuning the performance of the proposed Kalman-filter, based on the covariance of measurement noises.

For the second part of the system's states which includes the angular displacement and the angular velocity of the AMR, a complementary filter proposed in (Madgwick et al., 2011) can be used for eliminating the noise from the gyro measurement and also for computing the 3D angular displacements represented by Euler angles. The noise-free data for angular motion is available by utilizing the commercial NGIMU

Table 4.18: Definition of the variables used in the Kalman-filter for linear motion of the AMRs (as in (4.49a) and in (4.49b)), to be incorporated in the HIL test

Variable	WMR	Quadrrotor
\hat{x}	$[\hat{x}_w \quad \hat{y}_w \quad \hat{v}_w]^T$	$[\vec{\hat{p}}_q \quad \vec{\hat{v}}_q]^T$
u_o	a_w^m	\vec{d}_q^m
z_o	$[x_w^m \quad y_w^m]^T$	\vec{p}_q^m
A_o	$\begin{bmatrix} 0 & 0 & \cos(\theta_w) \\ 0 & 0 & \sin(\theta_w) \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$
B_o	$[0 \quad 0 \quad 1]^T$	$\begin{bmatrix} \mathbf{0}_3 \\ R_q \end{bmatrix}^T$
G_o	\mathbf{I}_3	\mathbf{I}_6
H_o	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	$[\mathbf{I}_3 \quad \mathbf{0}_3]$

sensor, which has the proposed complementary filter embedded in a compact module alongside with the low-cost IMU. The derivation and implementation of the filter is presented in detail in (Madgwick et al., 2011). Hence, here it is assumed that the measured data for angular displacements and the angular velocities are noise-free for the WMR and the quadrotor.

4.7.3 HIL test results

Here, the data logged using MATLAB during the HIL tests are presented for both cases including the WMR and the quadrotor. In both of the tests, the measurement noise on the measured states (i.e. z_o in Table 4.18) is a uniformly distributed noise with minimum value of $-\beta_0^n$ and the maximum value of β_0^n , where $\beta_0^n \in \mathbb{R}^+$ is a constant value for the noise bounds. On the other hand, the type of measurement noise on the measured dynamics of AMRs (i.e. u_o in Table 4.18) is same as the previous case, but with the lower band of $-0.1 \times \beta_0^n$ and the maximum band of $0.1 \times \beta_0^n$. In fact, it is assumed that the noise on the acceleration measurement is 10 times lower than the noise on the position measurements. This assumption is not far from reality, according

to the available data sheets of the corresponding commercial sensors. In addition, the constant parameters in the AMFC algorithm are tuned with least possible efforts as shown in Table 4.19. In this table, $\mathbf{1}_z$ is a vector in $\mathbb{R}^{z \times 1}$ with all elements equal to one, where $z > 0$ is a constant scalar number. Moreover, \mathbf{I}_z is an identity matrix in $\mathbb{R}^{z \times 1}$.

The simulation results for the WMR are presented in Fig. (4.89) to Fig. (4.93). Here, the value of α_w is equal to 0.1. The desired trajectory in x-y plane is a rectangle with length of 5m and width of 3m, as follows

$$x_d^w = \begin{cases} 0 & , t \leq 10 \\ 5 & , 10 < t \leq 70 \\ 0 & , 70 < t \end{cases}, \quad (4.50a)$$

$$y_d^w = \begin{cases} 0 & , t \leq 40 \\ 3 & , 40 < t \leq 100 \\ 0 & , 100 < t \end{cases}. \quad (4.50b)$$

As it can be seen in the results, the errors for linear and rotational motions are bounded while converging to zero. Moreover, the adapted \hat{A} and \hat{g} and consequently the control signals are all bounded. In Fig. 4.89 (top plot), a delay is observed in the WMR movement, so as to track the desired path in Y direction. This is due to the fact that the WMR has the initial pose of zero with regards to the horizon; and when it is commanded to move in the Y direction only, first it tries to rotate 90 degree and looks toward the right direction and then start moving. This can be confirmed by looking at the rotation amount of the WMR around the Z direction in the bottom plot in Fig. 4.89. In addition, in Fig. 4.90 (bottom plot), some fluctuations are observed in the control signals (which are the set-points of the WMR electric motors). This fluctuations are

bounded and as can be observed are damped through time. Hence, there are not much concerns about that.

Table 4.19: Properties of the AMFC algorithm (as in Algorithm 1 in Table 3.1) with the Kalman-filter utilized in the two HIL tests ($X_8 = 10^{-6} \times \mathbf{1}_8$).

HIL test-1: autonomous WMR (each of linear and angular motions)	HIL test-2: autonomous quadrotor
$\Gamma_1 = \text{diag}([0.001; 10])$	$\Gamma_1 = \text{diag}([X_8; 0.1; (10 \times \mathbf{1}_3)])$
$\Gamma_2 = \text{diag}([0.001; 0.1])$	$\Gamma_2 = \text{diag}([X_8; (0.01 \times \mathbf{1}_4)])$
$\rho_1 = 0.001$	$\rho_1 = 0.001$
$\rho_2 = 0.001$	$\rho_2 = 0.001$
$Q = 0.01 \times \mathbf{I}_2$	$Q = 0.01 \times \mathbf{I}_{12}$
$R = \mathbf{I}_2$	$R = \mathbf{I}_{12}$
$k_1 = 1, k_2 = 1$	$k_1 = 1, k_2 = 1$
$Q_o = \mathbf{I}_2$	$Q_o = 10 \times \mathbf{I}_2$
$R_o = \mathbf{I}_2$	$R_o = 0.1 \times \mathbf{I}_2$

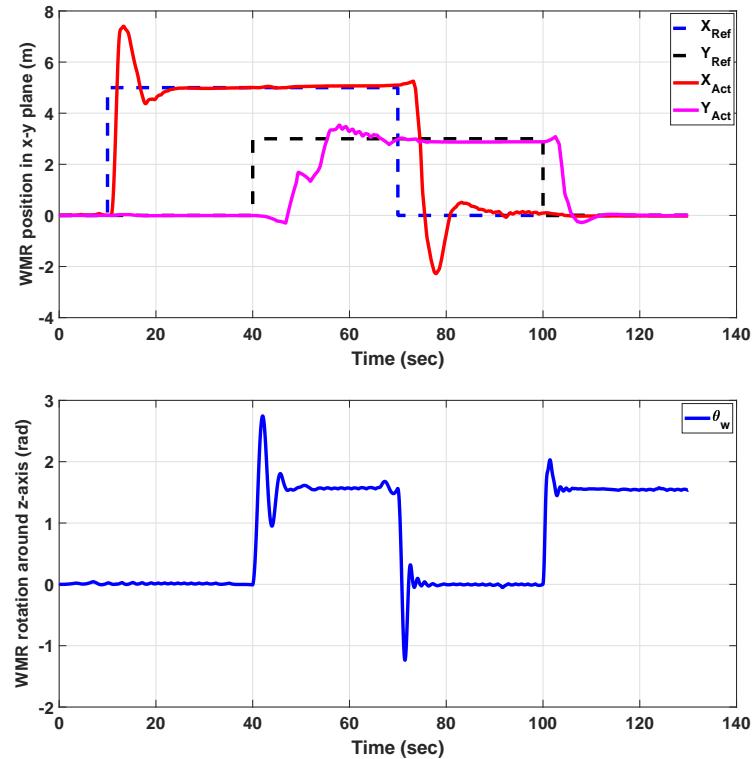


Figure 4.89: The x-y position and z-axis rotation of the autonomous WMR in the HIL test.

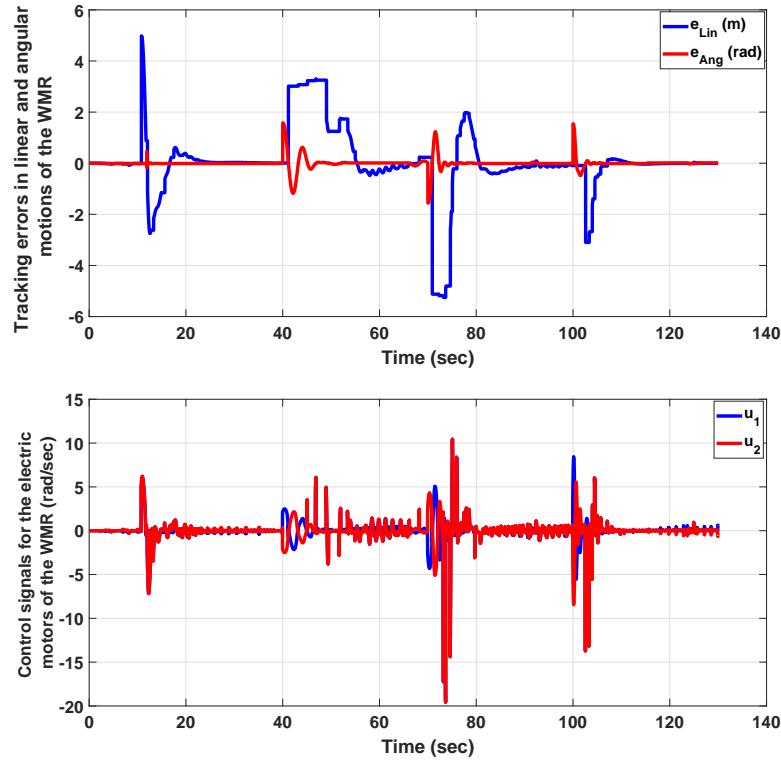


Figure 4.90: The errors in linear and angular motions of the autonomous WMR (top); and the control signals for the electric motors of the WMR (bottom), in the HIL test for evaluating the proposed AMFC algorithm.

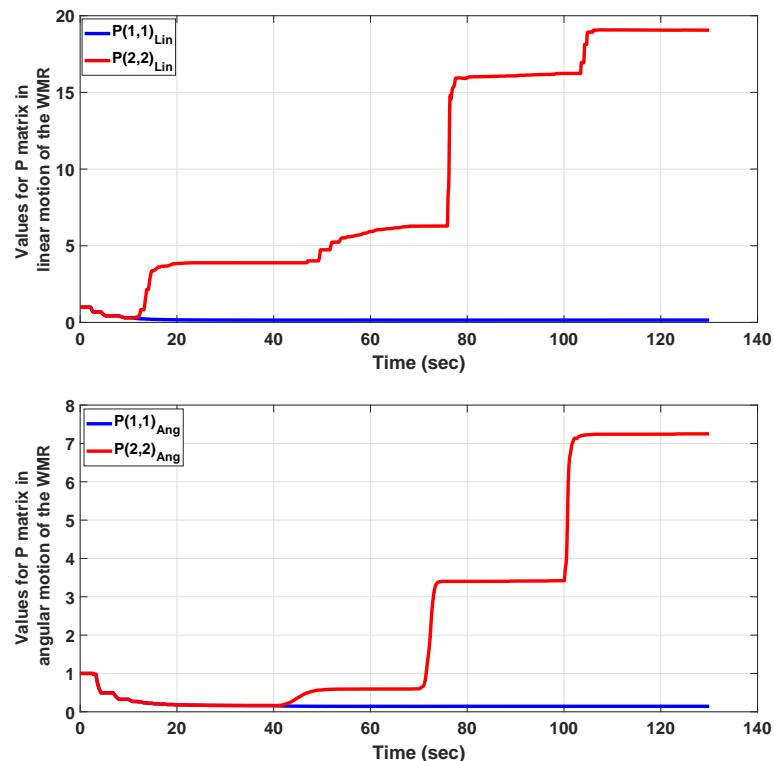


Figure 4.91: The values of P matrix for linear and angular motions of the autonomous WMR (as updated by (3.49)), in the HIL test for evaluating the proposed AMFC algorithm.

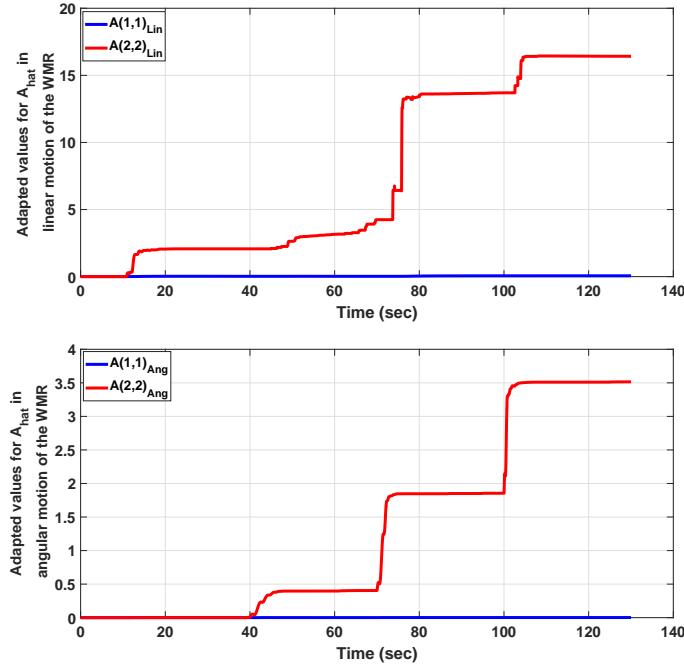


Figure 4.92: The adapted values of unknown linear terms for linear and angular motions of the autonomous WMR (as updated by (3.50b)), in the HIL test for evaluating the proposed AMFC algorithm.

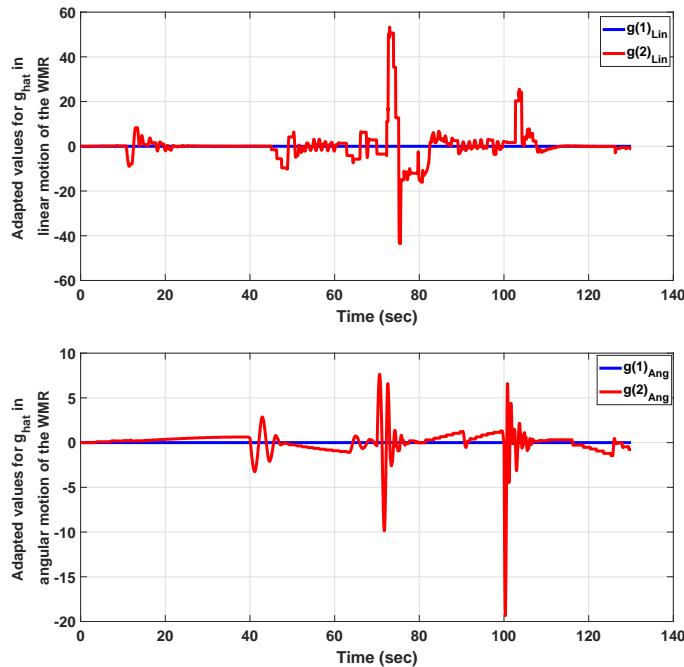


Figure 4.93: The adapted values of unknown nonlinear terms for the linear and angular motions of the autonomous WMR (as updated by (3.50a)), in the HIL test for evaluating the proposed AMFC algorithm.

The simulation results for the quadrotor are depicted in Fig. (4.94) to Fig. (4.98).

Here, the value of α_q is equal to 0.2. The desired trajectory is a helix along z-axis as follows (all of the values are in m)

$$x_d^Q = \begin{cases} 0 & , t < 20 \\ 2\cos(0.2[t - 20]) - 2 & , t \geq 20 \end{cases}, \quad (4.51a)$$

$$y_d^Q = \begin{cases} 0 & , t < 20 \\ 2\sin(0.2[t - 20]) & , t \geq 20 \end{cases}, \quad (4.51b)$$

$$z_d^Q = 0.2t. \quad (4.51c)$$

Considering the unknown disturbances and the suggested measurement noise, the position errors are converged to zero while the control signals and the estimated values are all bounded.

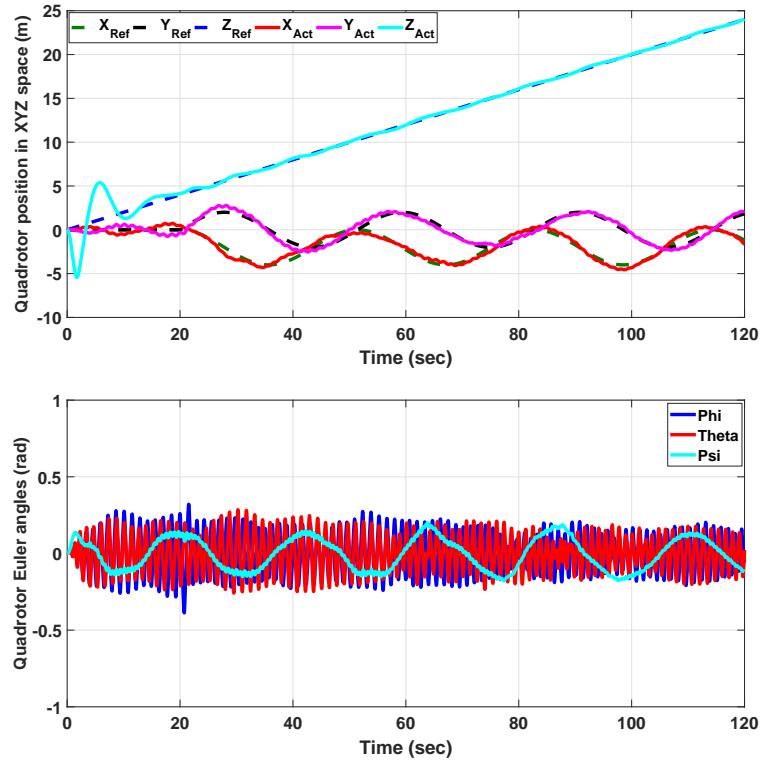


Figure 4.94: The autonomous quadrotor position in 3D space and its Euler angles, in the HIL test for evaluating the proposed AMFC algorithm.

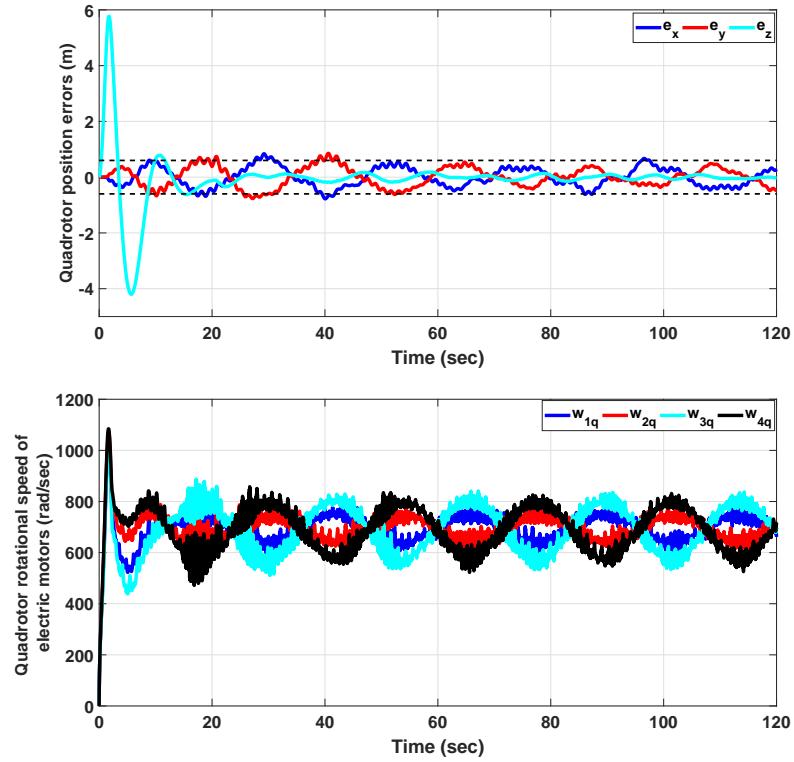


Figure 4.95: The errors in position of the autonomous quadrotor (top); and the control signals for the electric motors speeds (bottom), in the HIL test for evaluating the proposed AMFC algorithm. The values of errors are bounded with absolute upper-bound of 0.6 .

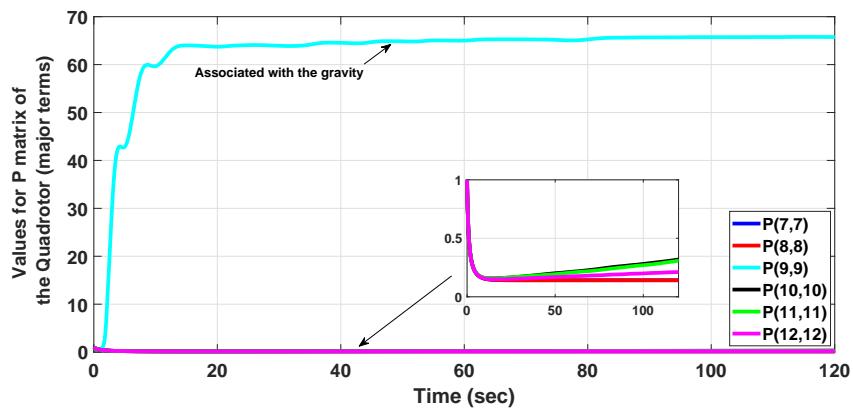


Figure 4.96: The values of P matrix for the autonomous quadrotor (as updated by (3.49)), in the HIL test for evaluating the proposed AMFC algorithm.

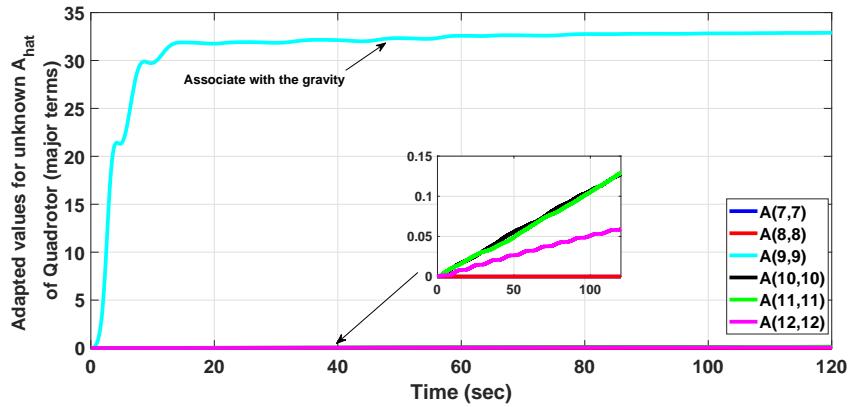


Figure 4.97: The adapted values of unknown linear terms for the autonomous quadrotor (as updated by (3.50b)), in the HIL test for evaluating the proposed AMFC algorithm.

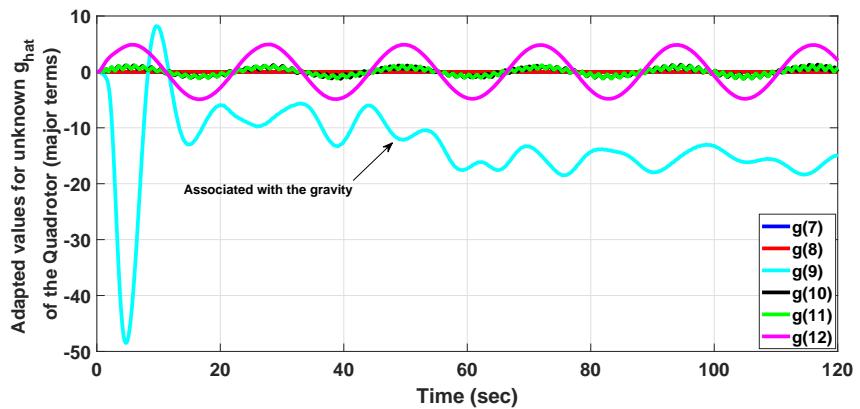


Figure 4.98: The adapted values of unknown nonlinear terms for the autonomous quadrotor (as updated by (3.50a)), in the HIL test for evaluating the proposed AMFC algorithm.

4.7.4 Analysis on amplitudes of the external disturbance and the measurement noise

As it is shown in the test results (specifically Fig. 4.89 and Fig. 4.94), the AMFC algorithm incorporated with the Kalman-filter (adopted from (Cui et al., 2016; Lewis et al., 2007)) can provide an appropriate performance for the dynamic systems operating under the unknown external disturbances as well as bounded measurement noise. In this section, the effects of different values for amplitudes of the disturbances and the measurement noise are evaluated. Hence, the values for tuning parameters of the controller and the observer are fixed at the values presented in Table 4.19. The analysis includes two parts. First, the amplitudes of the external disturbances are supposed to be fixed and the effects of change in the amplitudes of the measurement noise are studied. At the second step, different values for amplitudes of the disturbances are considered, while the maximum values for random measurement noise are fixed. For the purpose of evaluation, the sum of squared errors (SSE) for position of the WMR and the quadrotor is considered as follows

$$S_e^w = \sum_{i=1}^{N_w} (\{x_w - x_d^w\}_i^2 + \{y_w - y_d^w\}_i^2), \quad (4.52a)$$

$$S_e^Q = \sum_{i=1}^{N_q} (\{x_q - x_d^Q\}_i^2 + \{y_q - y_d^Q\}_i^2 + \{z_q - z_d^Q\}_i^2), \quad (4.52b)$$

where N_w and N_q are the total number of time steps in HIL tests for WMR and quadrotor, respectively. The analysis on amplitude of the measurement noise is proposed in Table 4.20 for both HIL tests, according to the values of SSE. These values are derived for the cases where the amplitudes of the disturbance are fixed at $\alpha_w = 0.1$ and $\alpha_q = 0.2$. In addition, the values of SSE for the case where the value of

$\beta_0^n = 0.5$ is constant, is presented in Table 4.21. These values can be used for evaluating effects of the disturbance amplitude on the performance of the AMFC algorithm. Since the test results are affected by the random nature of measurement noise, the presented data in Table 4.20 and Table 4.21 are the average values for 10 sets of consecutive simulation results. According to the data provided in these tables, one can say that the performance of AMFC algorithm degrades with increasing the amplitude of the disturbance. Obviously, the performance would be improved by tuning the values in Γ_1 and Γ_2 matrices. On the other hand, the effect of increasing the amplitude of the measurement noise is minor on the performance of the AMFC algorithm. It is also an indication for the usefulness of the Kalman-filter.

Table 4.20: Values of SSE for different amplitudes of the measurement noise in the HIL test ($\alpha_w = 0.1$, $\alpha_q = 0.2$). The parameter β_0^n is declared in Section 4.7.3.

Value of SSE	$\beta_0^n = 0.1$	$\beta_0^n = 0.5$	$\beta_0^n = 0.9$
S_e^w	2.59e4	2.81e4	2.93e4
S_e^Q	4.66e3	4.68e3	4.70e3

Table 4.21: Values of SSE for different amplitudes of the external disturbance in the HIL test ($\beta_0^n = 0.5$). The parameters α_w and α_q are declared in Section 4.7.1.

Value of SSE	$\alpha_w = 0.05$	$\alpha_w = 0.1$	$\alpha_w = 0.2$
S_e^w	2.48e4	2.81e4	6.05e4
	$\alpha_q = 0.1$	$\alpha_q = 0.2$	$\alpha_q = 0.4$
S_e^Q	3.96e3	4.66e3	7.32e3

4.8 Summary

In this chapter, the simulations results including the comparative studies as well as the applications on the models of practical dynamic systems are provided for each of the algorithms proposed in Chapter 3. Section 4.2 includes two comparative studies for evaluating the AMFC algorithm on single-agent SISO systems, against an intelligent PI controller as well as a sliding-mode controller. The study is provided by comparing the dominant exciting frequencies in the control signals of the aforementioned controllers. Later in that section, the application of AMFC algorithm on a chaotic dynamic system is presented.

In Section 4.3, the AMFC algorithm for single-agent MIMO dynamic systems is studied. First, the algorithm is compared with two state-of-the-art optimal model-free control algorithms, which are using ANNs for generating the control signal. It is then followed by applications of the AMFC algorithm on a robotic manipulator, an autonomous WMR and an autonomous quadrotor. In all of these applications, there are uncertain internal dynamics as well as unknown bounded external disturbances in the model of the simulated dynamic systems. The provided results include the tracking performance, estimated values for unknown linear and nonlinear terms and also the adapted values for the main controller gains (diagonal elements P matrices).

Sections 4.4 and 4.5 are dedicated to evaluation of the cooperative AMFC algorithms in multi-agent dynamic systems. First in Section 4.4, the CAMFC-1 algorithm is applied on a network of four autonomous quadrotors, which has a spanning-tree communication graph. In that application, there is not any need for inter-agent relative measurements among the quadrotors in the network. Each

quadrotor uses local on-board sensors to measure its states. But, the quadrotors are in cooperation, in this sense that the desired time-varying trajectory and the demand time-varying formation topology are estimated using two cooperative observers. Later in Section 4.5, the CAMFC-2 algorithm is compared against two distributed cooperative control algorithms from the literature, on different multi-agent dynamic systems. While, the first comparative study has an objective to reach consensus among the states of agents in the network, the objective of the second comparison is to solve a formation-tracking problem in a network. Finally, the CAMFC-2 algorithm is implemented on a network of four autonomous quadrotors, assuming the availability of relative measurements over the translational position and velocity vectors among the neighboring agents.

Then, the localization performance of ACL algorithm is evaluated in Section 4.6. The evaluation is provided against the recently published solutions via different case studies. Later in that Section, the application of the CAMFC-2 and ACL algorithm on a network of autonomous quadrotors is presented.

Finally, the AMFC algorithm as the core solution in the current thesis is further tested for autonomous WMR and quadrotor in a hardware-in-the-loop test platform. In the test platform, the AMFC algorithm is implemented on a developing board (which can be the one to be used in real practice application) and the dynamics of AMRs are emulated on another developing board. The two developing boards are connected via an I²C serial communication in order to transmit the control commands as well as the emulated states.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

This chapter includes the conclusion of the current thesis as well as proposing possible related works for future investigations.

5.1 Conclusion

By recalling the research objectives defined in Section 1.3, and according to the algorithms developed in Chapter 3 and the numerical results presented in Chapter 4, the followings have been successfully accomplished in this thesis;

- the AMFC algorithm is developed for tracking control problem in single-agent dynamic systems with completely unknown nonlinear dynamics;
- an online version of a DRE is utilized in AMFC algorithm, in order to update the main controller gains in the, incorporating the online estimated values for unknown linear terms in the system dynamics;
- two regressor-free adaptive laws are developed for online estimation of the unknown linear and nonlinear terms in the dynamic systems, so as to generate the control signals in AMFC algorithm without incorporating the ANNs or FISs; in this regard, the requirement of PE condition is revoked;
- the decentralized CAMFC-1 algorithm is developed to address the consensus or formation-tracking objectives in multi-agent systems with completely unknown

nonlinear dynamics, while the inter-agent relative state measurements are no more required among the neighboring agents;

- the decentralized CAMFC-2 algorithm is developed to tackle the consensus or formation-tracking objectives in multi-agent systems with completely unknown nonlinear dynamics, while the inter-agent relative state measurements among the neighboring agents are utilized for generating the control signals;
- the ACL algorithm is developed to estimate the relative and absolute position of each agent within a network of mobile agents with only one beacon agent (agent accessing the global absolute position data), while the minimum number of communication links among the agents is needed;
- the measurements over relative bearing angles among neighboring agents in the network are not required in the proposed ACL algorithm, while only the measurements on relative distance and relative velocity vector between the neighboring agents should be available.

The brief presentations of the developed algorithms and their specifications are provided as follow:

- **AMFC algorithm;** The design procedure for an adaptive model-free control policy based on the ultra-local model is presented in Section 3.2 for tracking control problem in a generic SISO dynamic system. It is then extended in Section 3.3 for tracking control of MIMO systems, by incorporating a set of auxiliary virtual control variables into the generally-defined structure of unknown nonlinear dynamic systems. The structure represents the unknown

nonlinearities of a generic dynamic system as a combination of an unknown linear-in-states part and one unknown nonlinear function. The proposed AMFC has two regressor-free adaptive laws for estimating unknown linear and nonlinear terms without any requirement for PE condition, due to exemption of requiring any form of regressors. By utilizing the estimated linear terms in a differential Riccati equation, the main controller gains can be adapted online. Such efficacy brings convenience in terms of controller tuning, as there is minimal adjustment needed on the available tuning knobs.

Since the ANNs or FISs are not used for estimation of unknown terms in the proposed AMFC algorithm, the demand accuracy in tracking control problem can be achieved without increasing the number of unknown variables. Consequently, the number of required adaptive laws in the controller is fixed regardless of the level of nonlinearity and parametric model complexity in the system. The appropriate accuracy is achieved only by adjusting the adaptive and leakage rates, based on the guidelines provided in the design procedure. It means that computational complexity of the proposed AMFC algorithm is preserved, without depending on the nature state of the dynamical system complexity. On contrary with the case of ANN utilization, one needs to increase the number of nodes and consequently the number of adaptive laws to reach desired accuracy in the corresponding MFC algorithms. As a result, the computational complexity of those algorithms increases in proportion with the level of nonlinearity in the dynamic system. Stability and optimality analyses are provided in Section 3.2 and 3.3 for the proposed AMFC algorithm.

According to the simulation results provided in Section 4.2, the tracking

performance of the systems using AMFC algorithm is almost similar to those of SMC and iPI controllers. However, the control policy generated by AMFC is smoother than the ones generated by SMC and iPI. The control signals of the SMC and iPI algorithms have dominant frequencies that can lead to resonances in the dynamic systems, while the AMFC control signal does not have any dominant frequency (referring to Table 4.1 and Table 4.2). In this sense, the proposed AMFC policy has a smooth control effort that makes the algorithm applicable directly on dynamic systems, as it is shown in application for a chaotic oscillator plant.

Moreover, the performance of the proposed AMFC algorithm is compared in Section 4.3 with two state-of-the-art RL-inspired MFC algorithms (MFC-1 and MFC-2 algorithms). Referring to those simulation studies, most of the controller parameters residing in the novel AMFC can be conveniently set with minimal tuning knobs, since the main controller gains are updated online. The designer only needs to change the adaptation and the associated leakage rates to vary the convergence of the algorithm. In addition, it is shown that only 4 adaptive laws are required in the AMFC algorithm, while the MFC-1 algorithm needs 14 adaptive laws (as in Table 4.4) and the MFC-2 algorithm has 8 adaptive laws (as in Table 4.5). In addition, the values of a cost function combining the tracking performance and the amount of control efforts, are compared and it is observed that the AMFC algorithm has equivalently same cost compared to the MFC-1 algorithm; and 49% lower cost compared to the MFC-2 algorithm.

In addition, the ease of convenience in controller parameters tuning without requiring regressor parameters is an attractive feature exhibited by the AMFC

algorithm, especially when implementing on practical complex dynamic systems such as autonomous robotic systems. The applications of AMFC algorithm on an automated 2DOF robotic manipulator, an autonomous WMR and an autonomous quadrotor are presented in Section 4.3. Furthermore in Section 4.8, to validate the performance of the proposed AMFC algorithm previously simulated, the algorithm is implemented on the emulated AMRs via the HIL concept.

- **CAMFC-1 algorithm;** Section 3.4 contains the design procedure of a decentralized cooperative observer to estimate a time-varying reference trajectory and also a set of time-varying formation parameters in a network of nonlinear dynamic agents. The proposed cooperative observer includes a term for providing the finite-time convergence of the observer error. This is an essential property to be featured, especially when the reference trajectory and the desired formation parameters are time-varying. Moreover, the cooperative observer is completely decoupled from the agents dynamics in the network, which leads to more stable performance of the observer algorithm, regardless of the controller performance.

Based on the estimated parameters by the cooperative observer, the desired path to be followed by each dynamic agent in the network is defined locally at the agent, so as to satisfy the objective of formation-tracking problem. Thereby, the formation-tracking problem in a multi-agent system is converted to a tracking problem locally at each agent in the network. Then, the previously proposed AMFC policy is applied at each agent for satisfying the local tracking objective.

By combination of the proposed cooperative observer and the AMFC

algorithm, the CAMFC-1 algorithm is suggested for formation-tracking problem, without the need for information on the inter-agent relative positions among the agents in the network. This essentially brings a significant cost saving in terms of hardware development point of view as number of sensors can be reduced. Moreover, since the relative localization problem remains an active problem, the solution can be considered for implementations in real applications, with fewer technological and practical concerns.

Furthermore, to have CAMFC-1 algorithm be implemented in a multi-agent system, only an undirected spanning-tree communication graph is required, which is a minimal requirement. The exemplified simulation in Section 4.4 yields a promising results, showing the preferable formation-tracking performance in a network of four autonomous quadrotors. In this simulation study, the dynamics of quadrotors are considered to be completely unknown while unknown disturbances are included, as well.

- **CAMFC-2 algorithm;** a model-free distributed control algorithm for consensus and formation-tracking problems in a network of nonlinear agents with completely unknown dynamics and external disturbances is presented in Section 3.5. The main purpose is to achieve tracking objective for the whole network while all agents are synchronized with a virtual leader in the network. The algorithm includes two distributed adaptive laws for estimating both unknown linear and nonlinear terms in the agents dynamic systems. In addition, a cooperative observer is designed based on a consensus-type error for estimating the leader states and its control inputs, locally at each agent. Since there are partial information links between the leader and the agents, the system

states and control inputs of the leader are required to be estimated at each agent in the distributed control protocol. While the stability of entire design is analyzed by using Lyapunov stability theorem, an optimality analysis is presented to show that the proposed distributed controller has an optimal term.

The major difference between CAMFC-1 and CAMFC-2 algorithms is that the inter-agent relative measurements over the states of agents in the network are required in CAMFC-2. While this is an extra hardware and software requirement for practical implementations, in some applications it may provide better consensus performance. The presented simulation results in Section 4.5 for three different cases indicate the appropriate performance of the proposed CAMFC-2 algorithm.

According to the provided comparative studies, the value of the cost function integrating both tracking error and the control effort at all agents in the network, is 5.5% lower for the CAMFC-2 algorithm when it is compared to the DCC-1 algorithm (as in Table 4.10). Also, the value of this cost for CAMFC-2 algorithm is 51.5% lower than that of the DCC-2 algorithm (as in Table 4.11). Note that in these comparisons, the CAMFC-2 algorithm has 4 adaptive laws at each agent in the network, while the DCC-1 algorithm requires 4 adaptive law, and the DCC-2 algorithm needs 10 adaptive laws at each agent. In addition, less control effort is required in the CAMFC-2 algorithm and there is not too much effort for off-line adjusting of the tuning parameters.

Moreover, since the adaptive laws are regressor-free, there is no requirement to define the regressor (activation) functions for implementation of the distributed controller. These two properties provide more convenient implementation of the

proposed CAMFC-2 algorithm in practical cases over the ANN-based distributed cooperative MFC algorithms, which use ANNs for generating the control law in unknown multi-agent dynamic systems. Finally, an application of the CAMFC-2 algorithm on a team of four autonomous quadrotors is presented in Section 4.5.

- **ACL algorithm;** In Section 3.6, an adaptive relative position estimator is incorporated in a cooperative observer to provide a solution for the cooperative localization problem in a network including one or more mobile agents. The solution which is named as the ACL algorithm, does not depend on the dynamics of the agents and the local control signals. The ACL algorithm needs each pair of the mobile agents to have non-zero relative velocities and non-zero relative distances. There is no need for measurement of the angle of arrival for the communicated signals. In addition, the absolute positions of the mobile agents can be estimated by accessing to only one beacon agent in the network. There is only a requirement for an undirected communication path between each two agents in the network to confirm the stability and convergence of the solution. This is the least possible requirement for a spanning-tree communication graph, which leads to more convenience in practical implementations.

According to the provided simulation studies in Section 4.6, the adaptive relative position estimator works well for each pair of the mobile agents and it outperforms two other state-of-the-art relative position estimation algorithms. In particular, the relative position estimation error using the proposed adaptive relative position estimator (as in (3.221)) is 57.9% lower than the relative position estimator in (4.42) and 52.2% lower than the relative position estimator

in (4.43), in the worst case scenario with stationary beacon agent (referring to Table 4.14). These values for the simulation results for the moving beacon agent, are 34.4% and 34.1%, respectively (as in Table 4.15).

Furthermore for a network of four dynamic agents and one beacon agent, it is shown that the proposed ACL algorithm has 93.3% lower localization error than the LC algorithm, at the fourth agent in the network (as in Table 4.16). The LC algorithm exploits the triangulation technique among the three neighboring agents in a network. Moreover, the ACL algorithm requires 4 communication links among the agents in the network, while the LC algorithm needs 7 links in the network (this is the maximum number of links for a network of 4 agents and one beacon agent).

In addition, the appropriate localization performance for the ACL algorithm has been observed in a team of autonomous quadrotors, which are cooperatively controlled using the CAMFC-2 algorithm. The proven results promise a profound advantage especially in the context of a network of AMRs with application in a remote place such as search and rescue mission, when GPS signal is degraded and often obscured.

5.2 Future works

The proposed package in the current thesis, can be considered as a solution for basic problems in real-life applications of the AMRs, including the automatic tracking control and localization problems. Future works might include

- the implementations of the proposed algorithms on real platforms, including homogeneous and heterogeneous agents as well as the fixed and time-varying communication graphs;
- the inclusion of a detailed model for energy consumption on AMRs and consequently to have the design process more energy-oriented;
- the extension of the algorithms so as to have as few tuning knobs as possible;
- the design of an algorithm for estimating the relative velocity vector between two mobile agents.

REFERENCES

- Abbasi, Y., Moosavian, S. A. A., & Novinzadeh, A. B. (2017). Formation control of aerial robots using virtual structure and new fuzzy-based self-tuning synchronization. *Transactions of the Institute of Measurement and Control*, 39(12), 1906-1919.
- Arab, A., & Fateh, M. (2015). An uncertainty compensator for robust control of wheeled mobile robots. *Advanced Robotics*, 29(20), 1303-1313.
- Asif, M., Memon, A. Y., & Khan, M. J. (2016). Output feedback control for trajectory tracking of wheeled mobile robot. *Intelligent Automation and Soft Computing*, 22(1), 75-87.
- Aspnes, J., Eren, T., Goldenberg, D. K., Morse, A. S., Whiteley, W., Yang, Y. R., ... Belhumeur, P. N. (2006). A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12), 1663–1678.
- Atassi, A. N., & Khalil, H. K. (1999). A separation principle for the stabilization of a class of nonlinear systems. *IEEE Transactions on Automatic Control*, 44(9), 1672-1687.
- Aung, W. (2007). Analysis on modeling and simulink of dc motor and its driving system used for wheeled mobile robot. *International Scholarly and Scientific Research and Innovation*, 1(8), 1149-1156.
- Bara, O., Fliess, M., Join, C., Day, J., & Djouadi, S. M. (2018). Toward a model-free feedback control synthesis for treating acute inflammation. *Journal of Theoretical Biology*, 448, 26 - 37.
- Bechlioulis, C. P., & Rovithakis, G. A. (2017). Decentralized robust synchronization of unknown high order nonlinear multi-agent systems with prescribed transient and steady state performance. *IEEE Transactions on Automatic Control*, 62(1), 123-134.
- Boiko, I., & Chehadeh, M. (2018). Sliding mode differentiator/observer for quadcopter velocity estimation through sensor fusion. *International Journal of Control*, 91(9), 2113-2120.
- Boudjedir, H., Bouhali, O., & Rizoug, N. (2014). Adaptive neural network control based on neural observer for quadrotor unmanned aerial vehicle. *Advanced Robotics*, 28(17), 1151-1164.

- Boukerche, A., Oliveira, H. A., Nakamura, E. F., & Loureiro, A. A. (2007). Localization systems for wireless sensor networks. *IEEE Wireless Communication*, 14(6), 6-12.
- Boutayeb, M., Richard, E., Rafaralahy, H., Ali, H., & Zaloylo, G. (2008). A simple time-varying observer for speed estimation of uav. In *Proceedings of the 17th world congress, the international federation of automatic control* (p. 1760-1765). Seoul, Korea.
- Bouzera, N., Oussalah, M., Mezhout, N., & Khaireddine, A. (2017). Fuzzy extended kalman filter for dynamic mobile localization in urban area using wireless network. *Applied Soft Computing Journal*, 57, 452-467.
- Bu, X., Cui, L., Hou, Z., & Qian, W. (2018). Formation control for a class of nonlinear multiagent systems using model-free adaptive iterative learning. *International Journal of Robust and Nonlinear Control*, 28(4), 1402-1412.
- Bu, X., Yu, Q., Hou, Z., & Qian, W. (2019). Model-free adaptive iterative learning consensus tracking control for a class of nonlinear multiagent systems. *IEEE Transactions on Systems, Man and Cybernetics: Systems*. doi: 10.1109/TSMC.2017.2734799
- Cao, L., Li, H., & Zhang, H. (2016). Model-free power control of front-end pfc ac/dc converter for on-board charger. In *IEEE 8th international power electronics and motion control conference (IPEMC2016-ECCE Asia)*. Hefei, China.
- Carrill, F. J., & Rotella, F. (2015). Some contributions to estimation for model-free control. *IFAC-PapersOnline*, 48(28), 150-155.
- Chai, G., Lin, C., Lin, Z., & Fu, M. (2016). Single landmark based collaborative multi-agent localization with time-varying range measurements and information sharing. *Systems and Control Letters*, 87, 56-63.
- Chelouah, L., Semchedine, F., & Bouallouche-Medjkoune, L. (2018). Localization protocols for mobile wireless sensor networks: A survey. *Computers and Electrical Engineering*, 71, 733-751.
- Chen, C. L. P., Wen, G., Liu, Y., & Liu, Z. (2016). Observer-based adaptive backstepping consensus tracking control for high-order nonlinear semi-strict-feedback multiagent systems. *IEEE Transactions on Cybernetics*, 46(7), 1591-1601.
- Chen, M., & Ge, S. S. (2013). Direct adaptive neural control for a class of uncertain nonaffine nonlinear systems based on disturbance observer. *IEEE Transactions on Cybernetics*, 43(4), 1213-1225.
- Chenji, H., & Stoleru, R. (2013). Toward accurate mobile sensor network localization in noisy environment. *IEEE Transactions on Mobile Computing*, 12(6), 1094-1106.

- Cui, G., Xu, S., Lewis, F. L., Zhang, B., & Ma, Q. (2016). Distributed consensus tracking for non-linear multi-agent systems with input saturation: a command filtered backstepping approach. *IET Control Theory and Applications*, 10(5), 509-516.
- Cui, G., Xu, S., Ma, Q., Li, Y., & Zhang, Z. (2018). Prescribed performance distributed consensus control for nonlinear multi-agent systems with unknown dead-zone input. *International Journal of Control*, 91(5), 1053-1065.
- Cui, G., Zhuang, G., & Lu, J. (2016). Neural-network-based distributed adaptive synchronization for nonlinear multi-agent systems in pure-feedback form. *Neurocomputing*, 218, 234-241.
- Cui, J. Q., Lai, S., Dong, X., & Chen, B. M. (2016). Autonomous navigation of uav in foliage environment. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 84(1-4), 259-276.
- Cui, L., Li, Y., Yang, R., & Zhang, X. (2018). Asymptotic cooperative tracking control for unknown high-order multi-agent systems via distributed adaptive critic design. *IEEE Access*, 6, 24650-24659.
- Cui, L., Wang, X., & Zhang, Y. (2016). Reinforcement learning-based asymptotic cooperative tracking of a class multi-agent dynamic systems using neural networks. *Neurocomputing*, 171(1), 220-229.
- Das, A., & Lewis, F. L. (2010). Distributed adaptive control for synchronization of unknown nonlinear networked systems. *Automatica*, 46(12), 2014-2021.
- Deshpande, A. S. (2011). Max-plus representation for the fundamental solution of the time-varying differential riccati equation. *Automatica*, 47(8), 1667-1676.
- Diao, Y., Lin, Z., & Fu, M. (2014). A barycentric coordinate based distributed localization algorithm for sensor networks. *IEEE Transactions on Signal Processing*, 62(18), 4760–4771.
- Diggelen, F. V., & Enge, P. (2015). The world's first gps mooc and worldwide laboratory using smartphones. In *Proceedings of the 28th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2015)* (p. 361-369). Tampa, Florida.
- Ding, Z., & Li, Z. (2016). Distributed adaptive consensus control of nonlinear output-feedback systems on directed graphs. *Automatica*, 72, 46-52.
- Eriksson, K., Estep, D., & Johnson, C. (2004). *Applied mathematics: Body and soul; volume 1: Derivatives and geometry in ir3*. Berlin, Heidelberg, Germany: Springer-Verlag.
- Fischer, N., Kamalapurkar, R., & Dixon, W. E. (2013). Lasalle-yoshizawa corollaries for nonsmooth systems. *IEEE Transactions on Automatic Control*, 58(9), 2333-2338.

- Fliess, M., & Join, C. (2013). Model-free control. *International Journal of Control*, 86(12), 2228-2252.
- Fox, D., Burgard, W., Kruppa, H., & Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3), 325-344.
- Gao, W., Jiang, Y., & Davari, M. (2019). Data-driven cooperative output regulation of multi-agent systems via robust adaptive dynamic programming. *IEEE Transactions on Circuits and Systems II: Express Briefs*. doi: 10.1109/TCSII.2018.2849639
- Ghommam, J., & Saad, M. (2018). Adaptive leader-follower formation control of underactuated surface vessels under asymmetric range and bearing constraints. *IEEE Transactions on Vehicular Technology*, 67(2), 852-865.
- Grobwindhager, B., Rath, M., Kulmer, J., Bakr, M. S., Boano, C. A., Witrisal, K., & Romer, K. (2018). Salma: Uwb-based single-anchor localization system using multipath assistance. In *Proceedings of the 16th ACM conference on embedded networked sensor systems (SenSys '18)* (p. 132-144). Shenzhen, China.
- Guo, K., Qiu, Z., Meng, W., Xie, L., & Teo, R. (2017). Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments. *International Journal of Micro Air Vehicles*, 9(3), 169-186.
- Halder, S., & Ghosal, A. (2016). A survey o mobility-assisted localization techniques in wireless sensor networks. *Journal of Network and Computer Applications*, 60, 82-94.
- Han, Z., Guo, K., Xie, L., & Lin, Z. (2019). Integrated relative localization and leader-follower formation control. *IEEE Transactions on Automatic Control*, 64(1), 17-31.
- Hashim, H. A., El-Ferik, S., & Lewis, F. L. (2017). Adaptive synchronization of unknown nonlinear networked systems with prescribed performance. *International Journal of Systems Science*, 48(4), 885-898.
- Hedgecock, R. W. (2014). *Precise real-time localization using single-frequency gps* (Unpublished doctoral dissertation). a dissertation submitted to the faculty of the graduate school of Vanderbilt University in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering, Tennessee, USA.
- Heintschel-von-Heinegg, W., Frau, R., & Singer, T. (2018). *Dehumanization of warfare: Legal implications of new weapon technologies*. Gewerbestrasse 11, 6330 Cham, Switzerland: Springer.
- Horowitz, R., & Tomizuka, M. (1986). An adaptive control scheme for mechanical manipulators - compensation of nonlinearity and decoupling control. *ASME Journal of Dynamic Systems, Measurements and Control*, 108(2), 127-135.

- Hou, Z., & Jin, S. (2014). *Model-free adaptive control; theory and applications*. Boca Raton, Florida, USA: CRC Press, Taylor and Francis Group.
- Huang, R., & Záruba, G. V. (2009). Monte carlo localization of wireless sensor networks with a singel mobile beacon. *Wireless Networks*, 15, 978-990.
- Ioannou, P., & Fidan, B. (2006). *Adaptive control tutorial*. Philadelphia, USA: SIAM.
- Jadbabaie, A., Lin, J., & Morse, A. S. (2002). Coordination of groups of mobile autonomous agents using nearest neighbor rules. In *Proceddings of the 41st IEEE conference on decision and control (CDC '02)* (p. 2953-2958). Las Vegas, Nevada, USA.
- Jadbabaie, A., Lin, J., & Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6), 988-1001.
- Kane, T. R., Likins, P. W., & Levinson, D. A. (1993). *Spacecraft dynamics*. New York, USA: McGraw-Hill Book Company.
- Kang, S., Park, M., & Ahn, H. (2017). Distance-based cycle-free persistent formation: global convergence and experimental test with a group of quadcopters. *IEEE Transactions on Industrial Electronics*, 64(1), 380-389.
- Kelly, A. (2013). *Mobile robots: mathematics, models and methods*. 32 Avenue of the Americas, New York, USA: Cambridge University Press.
- Khalil, H. K. (1996). *Nonlinear systems*. Upper Saddle River, New Jersey, USA: Prentice-Hall, Inc.
- Khan, M. U., Li, S., Wang, Q., & Shao, Z. (2016). Formation control and tracking for co-operative robots with non-holonomic constraints. *Journal of Intelligent and Robotic Systems*, 82(1), 163-174.
- Kia, S. S., Rounds, S., & Martinez, S. (2016). Cooperative localization for mobile agents: A recursive decentralized algorithm based on kalman-filter decoupling. *IEEE Control Systems Magazine*, 36(2), 86-101.
- Kiumarsi, B., Vamvoudakis, K. G., Modares, H., & Lewis, F. L. (2018). Optimal and autonomous control using reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6), 2042-2062.
- Kofahl, R. J. (1988). Robustness and eigenvalue analysis of least-squares estimators for parameter adaptive control. *IFAC Proceedings Volumes*, 21(10), 33-38.
- Koh, K. C., & Cho, H. S. (1999). A smooth path tracking algorithm for wheeled mobile robots with dynamics constraints. *Journal of Intelligent and Robotic Systems*, 24(4), 367-385.
- Kok, M., Hol, J. D., & Schon, T. B. (2017). Using inertial sensors for position and orientation estimations. *Foundations and Trends in Signal Processing*, 11(1-2),

1-153.

- Kumar, V., Bekey, G. A., Ambrose, R., Lavery, D., Sanderson, D., Yuh, J., & Wilcox, B. (2008). *Robotics: state of the art and future challenges*. London, UK: Imperial College Press.
- Kurazume, R., Nagata, S., & Hirose, S. (1994). Cooperative positioning with multiple robots. In *Proceedings of the 1994 IEEE international conference on robotics and automation (ICRA '94)* (p. 1250-1257). San Diego, CA, USA.
- Lafont, F., Balmat, J.-F., Pessel, N., & Fliess, M. (2015). A model-free control strategy for an experimental greenhouse with an application to fault accommodation. *Journal of Computers and Electronics in Agriculture*, 110, 139-149.
- Lee, S.-H., Kang, S. H., & Kim, Y. (2011). Trajectory tracking control of quadrotor uav. In *Ieee 11th international conference on control, automation and systems* (p. 281-285). Gyeonggi-do, South Korea.
- Lee, T. (2018). Geometric control of quadrotor uavs transporting a cable-suspended rigid body. *IEEE Transactions on Control Systems Technology*, 26(1), 255-264.
- Leishman, R. C., Jr., J. C. M., Beard, R. W., & McLain, T. W. (2014). Quadrotors and accelerometers: state estimation with an improved dynamic model. *IEEE Control System Magazine*, 34(1), 28-41.
- Levant, A. (2003). High-order sliding modes, differentiation and output-feedback control. *International Journal of Control*, 76(9-10), 924-941.
- Lewis, F. L., & Vrabie, D. (2009). Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits and Systems Magazine*, 9(3), 32-50.
- Lewis, F. L., Vrabie, D., & Syrmos, V. L. (2012). *Optimal control*. USA: John Wiley and Sons Inc.
- Lewis, F. L., Xie, L., & Popa, D. (2007). *Optimal and robust estimation; with an introduction to stochastic control theory*. Boca Raton, Florida, USA: CRC Press, Taylor and Francis Group.
- Lewis, F. L., Zhang, H., Hengster-Movric, K., & Das, A. (2014). *Cooperative control of multi-agent systems*. London, UK: Springer-Verlog.
- Li, X., Shi, P., & Wang, Y. (2019). Distributed cooperative adaptive tracking control for heterogeneous systems with hybrid nonlinear dynamics. *Nonlinear Dynamics*. doi: 10.1007/s11071-018-4681-4
- Li, Z., & Duan, Z. (2015). *Cooperative control of multi-agent systems*. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group.
- Lin, Z., Fu, M., & Diao, Y. (2015). Distributed self localization for relative position sensing networks in 2d space. *IEEE Transactions on Signal Processing*, 63(14).

- Liu, D., & Wei, Q. (2014). Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, 25(3), 621-634.
- Liu, R., Yuen, C., Do, T.-N., Jiao, D., Liu, X., & Tan, U.-X. (2017). Cooperative relative positioning of mobile users by fusing imu inertial and uwb ranging information. In *IEEE international conference on robotics and automation (ICRA '17)* (p. 5623-5629). Singapore.
- Lu, J., Li, C., & Su, W. (2016). Extended kalman filter based localization for a mobile robot team. In *Proceedings of chinese control and decision conference (CCDC '16)* (p. 939-944). Yinchuan, China.
- Lu, Y., Zhang, G., Sun, Z., & Zhang, W. (2018). Adaptive cooperative formation control of autonomous surface vessels with uncertain dynamics and external disturbances. *Ocean Engineering*, 167, 36-44.
- Luo, B., Liu, D., Huang, T., & Wang, D. (2016). Model-free optimal tracking control via critic-only q-learning. *IEEE Transactions on Neural Networks and Learning Systems*, 27(10), 2134–2144.
- Luy, N. T. (2017a). Adaptive dynamic programming-based design of integrated neural network structure for cooperative control of multiple mimo nonlinear systems. *Neurocomputing*, 237, 12-24.
- Luy, N. T. (2017b). Robust adaptive dynamic programming based online tracking control algorithm for real wheeled mobile robot with omni-directional vision system. *Transactions of the Institute of Measurement and Control*, 39(6), 832-847.
- Lv, Y., Na, J., Yang, Q., Wu, X., & Guo, Y. (2016). Online adaptive optimal control for continuous-time nonlinear systems with completely unknown dynamics. *International Journal of Control*, 89(1), 99-112.
- Madadi, E., & Söfker, D. (2015). Model-free approaches applied to the control of nonlinear systems: A brief survey with special attention to intelligent pid iterative learning control. In *Proceedings of asme dynamic systems and control conference (dscc2015)*. Columbus, Ohaio, USA.
- Madgwick, S. O. H., Harrison, A. J. L., & Vaidyanathan, R. (2011). Estimation of imu and marg orientation using a gradient descent algorithm. In *Procedings of 2011 IEEE international conference on rehabilitation robotics (ICRA '11)* (p. 1-7). Zurich, Switzerland.
- Mahmood, A., & Kim, Y. (2017). Decentralized formation flight control of quadcopters using robust feedback linearization. *Journal of the Franklin Institute*, 354, 852-871.
- Mahony, R., Hamel, T., & Pflimlin, J. (2008). Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5),

1203-1218.

- Mahony, R., Kumar, V., & Corke, P. (2012). Multirotor aerial vehicles: modeling, estimation and control of quadrotors. *IEEE Robotics and Automation Magazine*, 19(3), 20-32.
- Mao, G., & Fidan, B. (2009). *Localization algorithms and strategies for wireless sensor networks*. 701 E. Chocolate Avenue, Hershey, PA, USA: Information Science Reference (an imprint of IGI Global).
- Mao, G., Fidan, B., & Anderson, B. D. O. (2007). Wireless sensor network localization techniques. *Computer Networks*, 51(10), 2529-2553.
- Mao, S., Tan, W. K., & Low, K. H. (2016). Autonomous formation flight of indoor uavs based on model predictive control. *AIAA Infotech @ Aerospace, AIAA SciTech Forum (AIAA 2016-0515)*, 1-9. doi: 10.2514/6.2016-0515
- Meng, W., Yang, Q., Jagannathan, S., & Sun, Y. (2019). Distributed control of high-order nonlinear input constrained multiagent systems using a backstepping-free method. *IEEE Transactions on Cybernetics*. doi: 10.1109/TCYB.2018.2853623
- Meng, W., Yang, Q., Sarangapani, J., & Sun, Y. (2017). Distributed control of nonlinear multiagent systems with asymptotic consensus. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 47(5), 749-757.
- Meng, W. C., Yang, Q. M., Jagannathan, S., & Sun, Y. X. (2014). Adaptive neural control of high-order uncertain non-affine systems: A transformation to affine systems approach. *Automatica*, 50(5), 1473-1480.
- Meyer, F., Hlinka, O., Wymeersch, H., Riegler, E., & Hlawatsch, F. (2016). Distributed localization and tracking of mobile networks including noncooperative objects. *IEEE Transactions on Signal and Information Processing over Networks*, 2(1), 57-71.
- Mhatre, V., & Rosenborg, C. (2004). Homogenous vs heterogenous clustered sensor networks: A comparative study. In *2004 IEEE international conference on communications* (p. 3646-3652). Paris, France.
- Modares, H., Lewis, F. L., Kang, W., & Davoudi, A. (2018). Optimal synchronization of heterogeneous nonlinear systems with unknown dynamics. *IEEE Transactions on Automatic Control*, 63(1), 117-131.
- Morbidi, F., & Mariottini, G. L. (2013). Active target tracking and cooperative localization for teams of aerial vehicles. *IEEE Transactions on Control Systems Technology*, 21(5), 1694-1707.
- Na, J., Mahyuddin, M. N., Herrmann, G., Ren, X., & Barber, P. (2015). Robust adaptive finite time parameter estimation and control for robotic systems. *International Journal of Robust and Nonlinear Control*, 25(16), 3045-3071.
- Nelles, O. (2001). *Nonlinear system identification: from classical approaches to*

neural networks and fuzzy models. Berlin, Heidelberg, Germany: Springer-Verlag.

- Nerurkar, E. D., Roumeliotis, S., & Martinelli, A. (2009). Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *Proceedings of 2009 IEEE international conference on robotics and automation (ICRA '09)* (p. 1402-1409). Kobe, Japan.
- Nguyen, H. T., & Le, H. X. (2016). Path planning and obstacle avoidance approaches for mobile robots. *International Journal of Computer Science Issues*, 13(4), 1-10.
- Nouri, B. (2005). Modeling and control of mobile robots. In *Proceedings of the first international conference on modeling, simulation and applied optimization* (p. 1-6). Sharjah, UAE.
- Oh, K. K., & Ahn, H. S. (2013). Formation control of mobile agents based on distributed position estimation. *IEEE Transactions on Automatic Control*, 58(3), 737-742.
- Oh, K. K., & Ahn, H. S. (2014). Formation control and network localization via orientation alignment. *IEEE Transactions on Automatic Control*, 59(2), 540-545.
- Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3), 401-420.
- Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215-233.
- Ouimet, M., Iglesias, D., Ahmed, N., & Martinez, S. (2018). Cooperative robot localization using event-triggered estimation. *Journal of Aerospace Information Systems*, 15(7), 427-449.
- Özkucur, N. E., Kurt, B., & Akin, H. L. (2009). A collaborative multi-robot localization method without robot identification. In *Robocup 2008: Robot soccer world cup XII. RoboCup 2008. lecture notes in computer science* (Vol. 5399, p. 189-199). Springer, Berlin, Heidelberg.
- Peng, Z., Wang, D., Zhang, H., & Lin, Y. (2015). Cooperative output feedback adaptive control of uncertain nonlinear multi-agent systems with a dynamic leader. *Neurocomputing*, 149, 132-141.
- Peng, Z., Zhang, D. W. H., Sun, G., & Wang, H. (2013). Distributed model reference adaptive control for cooperative tracking of uncertain dynamical multi-agent systems. *IET Control Theory and Applications*, 7(8), 1079-1087.
- Percup, R., Radac, M., Roman, R., & Petriu, E. M. (2017). Model-free sliding mode control of nonlinear systems: algorithms and experiments. *Information Sciences*, 381, 176-192.
- Priyantha, N. B., Balakrishnan, H., Demaine, E., & Teller, S. (2003). Anchor-free distributed localization in sensor networks. In *Proceeding of 1st international*

conference on embedded networked sensor systems (p. 340-341). LA, California, USA.

- Prorok, A., Bahr, A., & Martinoli, A. (2012). Low-cost collaborative localization for large-scale multi-robot systems. In *Proceedings of 2012 IEEE international conference on robotics and automation (ICRA '12)* (p. 4236-4241). Saint Paul, MN, USA.
- Prorok, A., & Martinoli, A. (2011). A reciprocal sampling algorithms for lightweight distributed multi-robot localization. In *Proceedings of 2011 IEEE/RSJ international conference on intelligent robots and systems* (p. 3241-3247). San Fransisco, CA, USA.
- Rad, A. B., & Tsang, K. M. (2000). Adaptive time-delay controller. *IEEE Transactions on Industrial Electronics*, 47(6), 1350-1353.
- Rafaralahy, H., Richard, E., Boutayeb, M., & Zasadzinski, M. (2012). Sensor diagnosis and state estimation for a class of skew symmetric time-varying systems. *Automatica*, 48, 2284-2289.
- Rao, M. V. S., & Shivakumar, M. (2018). Overview of battery monitoring and recharging of autonomous mobile robot. *International journal on recent and innovation trends in computing and communication*, 6(5), 174-179.
- Rekleitis, I., Dudek, G., & Milios, E. (2003a). Experiments in free-space triangulation using cooperative localization. In *Proceedings of 2003 IEEE/RSJ international conference on intelligent robotics and systems (IROS2003)* (p. 1777-1782). Las Vegas, NV, USA.
- Rekleitis, I., Dudek, G., & Milios, E. (2003b). Probabilistic cooperative localization and mapping in practice. In *2003 IEEE international conference on robotics and automation (ICRA '03)* (p. 1907-1912). Taipei, Taiwan.
- Ren, C.-E., Chen, L., & Chen, C. L. P. (2017). Adaptive fuzzy leader-following consensus control for stochastic multiagent systems with heterogeneous nonlinear dynamics. *IEEE Transactions on Fuzzy Systems*, 25(1), 181-190.
- Roman, R., Percup, R., & Radac, M. (2017). Model-free fuzzy control of twin rotor aerodynamic systems. In *Proceedings of 25th mediteranian conference on control and automation (MED)* (p. 559-564). Valletta, Malta.
- Roman, R., Precup, R., & David, R. (2018). Second order intelligent proportional-integral fuzzy control of twin rotor aerodynamic systems. *Procedia Computer Science*, 6th International Conference on Information Technology and Quantitative Management, 139, 372 - 380.
- Roman, R.-C., Radac, M.-B., Percup, R.-E., & Petriu, E. (2015, March). Data-driven optimal model-free control of twin rotor aerodynamic system. In *Proceedings of 2015 IEEE international conference on industrial technology (ICIT)* (p. 161-166). Sevil, Spain.

- Roumeliotis, S., & Bekey, G. A. (2000). Collective localization: a distributed kalman filter approach to localization of groups of mobile robots. In *Proceedings of 2000 IEEE international conference on robotics and automation (ICRA '00)* (p. 2958-2965). San Francisco, CA, USA.
- Roumeliotis, S., & Bekey, G. A. (2002). Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5), 781-795.
- Russel, J., Ye, M., Anderson, B. D. O., Hmam, H., & Sarunic, P. (2017). Cooperative localization of a gps-denied uav in 3-dimensional space using direction of arrival measurements. *IFAC-PapersOnline*, 50(1), 8019-8024.
- Safavi, S., & Khan, U. A. (2017). An opportunistic linear-convex algorithm for localization in mobile robot networks. *IEEE Transactions on Robotics*, 33(4), 875-888.
- Safavi, S., Khan, U. A., Kar, S., & Moura, J. M. F. (2018). Distributed localization: A linear theory. *Proceedings of the IEEE*, 106(7), 1204-1223.
- Sakhre, V., Singh, P., & Jain, S. (2017). Fcnp approach for uncertain nonlinear dynamical system with unknown disturbance. *International Journal of Fuzzy Systems*, 19(2), 452-469.
- Santosuoso, G. L., Benzemrane, K., & Damm, G. (2011). Nonlinear speed estimation of a gps-free uav. *International Journal of Control*, 84(11), 1873-1885.
- Shen, Q., & Shi, P. (2016). Output consensus control of multiagent systems with unknown nonlinear dead zone. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 46(10), 1329-1337.
- Shen, Q., Shi, P., & Shi, Y. (2016). Distributed adaptive fuzzy control for nonlinear multiagent systems via sliding mode observers. *IEEE Transactions on Cybernetics*, 46(12), 3086-3097.
- Shi, P., & Shen, Q. K. (2017). Observer-based leader-following consensus of uncertain nonlinear multi-agent systems. *International Journal of Robust and Nonlinear Control*, 27(17), 3794-3811.
- Siegwart, R., & Nourbakhsh, I. R. (2004). *Introduction to autonomous mobile robots*. Cambridge, Massachusetts, USA: The MIT Press.
- Slotine, J.-J. E., & Li, W. (1991). *Applied nonlinear control*. Englewood Cliffs, New Jersey, USA: Prentice Hall.
- Song, R., Lewis, F. L., & Wei, Q. (2017). Off-policy integral reinforcement learning method to solve nonlinear continuous-time multiplayer nonzero-sum games. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3), 704-713.
- Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot modeling and control*. New York, US: John Wiley and Sons.

- Stanoev, A., Filiposka, S., In, V., & Kocarev, L. (2016). Cooperative method for wireless sensor network localization. *Ad Hoc Networks*, 40, 61-72.
- Tan, Y., Chang, J., Tan, H., & Hu, J. (2000). Integral backstepping control and experimental implementation for motion system. In *Proceedings of 2000 IEEE international conference on control applications* (p. 367-372). Alaska, USA.
- Tanner, H. G., Jadbabaie, A., & Pappas, G. J. (2003a). Stable flocking of mobile agents, part i: fixed topology. In *42nd IEEE international conference on decision and control (CDC '03)* (p. 2010-2015). Maui, HI, USA.
- Tanner, H. G., Jadbabaie, A., & Pappas, G. J. (2003b). Stable flocking of mobile agents, part ii: dynamic topology. In *42nd IEEE international conference on decision and control (CDC '03)* (p. 2016-2021). Maui, HI, USA.
- Thabet, H., Ayadi, M., & Rotella, F. (2014). Ultra local model control based on an adaptive observer. In *IEEE conference on control applications (CCA)* (p. 122-127). Juan Les Antibes, France.
- Trawny, N., & Barfoot, T. D. (2004). Optimized motion strategies for cooperative localization of mobile robots. In *Proceedings of IEEE international conference on robotics and automation (ICRA '04)* (p. 1027-1032). New Orleans, LA, USA.
- Tron, R., Thomas, J., Loianno, G., Daniilidis, K., & Kumar, V. (2016). A distributed optimization framework for localization and formation control. *IEEE Control Systems Magazine*, 36(4), 22-44.
- Vamvoudakis, K. G., & Lewis, F. L. (2010). Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 46(5), 878-888.
- Vamvoudakis, K. G., Vrabie, D., & Lewis, F. L. (2014). Online adaptive algorithm for optimal control with integral reinforcement. *International Journal of Robust and Nonlinear Control*, 24(17), 2686-2710.
- Wanasinghe, T. R., Mann, G. K. I., & Gosine, R. G. (2015). Distributed leader-assistive localization method for a heterogeneous multirobotic system. *IEEE Transactions on Automation Science and Engineering*, 12(3), 795-809.
- Wang, G., Wang, C., Du, Q., & Cai, X. (2016). Distributed adaptive output consensus control of second-order systems containing unknown non-linear control gains. *International Journal of Systems Science*, 47(14), 3350-3363.
- Wang, G., Wang, C., Yan, Y., Li, L., & Cai, X. (2017). Distributed adaptive output feedback tracking control for a class of uncertain nonlinear multi-agent systems. *International Journal of Systems Science*, 48(3), 587-603.
- Wang, H., Liu, P. X., Li, S., & Wang, D. (2018). Adaptive neural output-feedback control for a class of nonlower triangular nonlinear systems with unmodeled dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8),

3658-3668.

- Wang, H., Liu, P. X., & Shi, P. (2017). Observer-based fuzzy adaptive output-feedback control of stochastic nonlinear multiple time-delay systems. *IEEE Transactions on Cybernetics*, 47(9), 2568-2578.
- Wang, H., Shi, P., Li, H., & Zhou, Q. (2016). Adaptive neural tracking control for a class of nonlinear systems with dynamic uncertainties. *IEEE Transactions on Cybernetics*, 47(10), 3075-3087.
- Wang, J. (2017). Distributed coordinated tracking control for a class of uncertain multiagent systems. *IEEE Transactions on Automatic Control*, 62(7), 3423-3429.
- Wang, J., Mounier, H., Cela, A., & Niculescu, S.-I. (2011). Event driven intelligent pid controllers with applications to motion control. In *Proceedings of the 18th world congress the international federation of automatic control (IFAC)* (p. 10080-10085). Milano, Italy.
- Wang, S., Gu, D., Chen, L., & Hu, H. (2016). Single beacon-based localization with constraints and unknown initial poses. *IEEE Transactions on industrial electronics*, 63(4), 2229-2241.
- Wang, W., Wang, D., & Peng, Z. (2017). Predictor-based adaptive dynamic surface control for consensus of uncertain nonlinear systems in strict-feedback form. *International Journal of Adaptive Control and Signal Processing*, 31(1), 68-82.
- Wang, W., Wang, D., Peng, Z., & Wang, H. (2016). Cooperative adaptive fuzzy output feedback control for synchronization of nonlinear multi-agent systems in the presence of input saturation. *Asian Journal of Control*, 18(2), 619-630.
- Wang, W., Wang, D., & Peng, Z. H. (2015). Cooperative fuzzy adaptive output feedback control for synchronisation of nonlinear multi-agent systems under directed graphs. *International Journal of Systems Science*, 46(16), 2982-2995.
- Wang, W., Wen, C., & Huang, J. (2017). Distributed adaptive asymptotically consensus tracking control of nonlinear multi-agent systems with unknown parameters and uncertain disturbances. *Automatica*, 77, 133-142.
- Wei, Q., Liu, D., & Lin, H. (2016). Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems. *IEEE Transactions on Cybernetics*, 46(3), 840-853.
- Wen, G., Yu, W., Li, Z., Yu, X., & Cao, J. (2017). Neuro-adaptive consensus tracking of multiagent systems with a high-dimensional leader. *IEEE Transactions on Cybernetics*, 47(7), 1730-1742.
- Wheeler, D. O., Koch, D. P., Jackson, J. S., McLain, T. W., & Beard, R. W. (2018). Relative navigation: a keyframe-based approach for observable gps-degraded navigation. *IEEE Control Systems Magazine*, 38(4), 30-48.
- Wu, L.-B., He, X.-Q., & Zhang, D.-Q. (2017). Cooperative adaptive fuzzy control for

- a class of uncertain nonlinear multi-agent systems with time delays. *Journal of Control and Decision*, 4(3), 131-152.
- Xiao, G., Zhang, H., Luo, Y., & Qu, Q. (2017). General value iteration based reinforcement learning for solving optimal tracking control problem of continuous-time affine nonlinear systems. *Neurocomputing*, 245, 114-123.
- Yang, S., Cao, Y., Peng, Z., Wen, G., & Guo, K. (2017). Distributed formation control of nonholonomic autonomous vehicle via rbf neural network. *Mechanical Systems and Signal Processing*, 87, 81-95.
- Yang, X., He, H., Liu, D., & Zhu, Y. (2017). Adaptive dynamic programming for robust neural control of unknown continuous-time non-linear systems. *IET Control Theory and Applications*, 11(14), 2307-2316.
- Yang, Y., & Yue, D. (2017). Distributed adaptive fault-tolerant control of pure-feedback nonlinear multi-agent systems with actuator failures. *Neurocomputing*, 221, 72-84.
- Younes, Y. Y. A., Drak, A., Noura, H., Rabhi, A., & Hajjaji, A. E. (2016). Robust model-free control applied to a quadrotor uav. *Journal of Intelligent Robotic Systems*, 84, 37-52.
- Yu, M., Huang, D., & He, W. (2016). Robust adaptive iterative control for discrete-time nonlinear systems with both parametric and nonparametric uncertainties. *International Journal of Adaptive Control and Signal Processing*, 30(7), 972-985.
- Yu, X., Hou, Z., & Zhang, X. (2018). Model-free adaptive control for a vapour-compression refrigeration benchmark process. *IFAC-PapersOnLine, 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID 2018*, 51(4), 527 - 532.
- Zandbergen, P. A. (2009). Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning. *Transactions in GIS*, 13(s1), 5-26.
- Zhang, H., Jiang, H., Luo, Y., & Xiao, G. (2017). Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method. *IEEE Transactions on Industrial Electronics*, 64(5), 4091-4100.
- Zhang, H., & Lewis, F. L. (2012). Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics. *Automatica*, 48(7), 1432-1439.
- Zhang, J., Wang, Z., & Zhang, H. (2019). Data-based optimal control of multi-agent systems: A reinforcement learning design approach. *IEEE Transactions on Cybernetics*, PP, 1-9. doi: 10.1109/TCYB.2018.2868715
- Zhang, J., Zhang, H., & Feng, T. (2018). Distributed optimal consensus control for nonlinear multiagent system with unknown dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 3339-3348.

- Zhang, K., Zhang, H., Xiao, G., & Su, H. (2017). Tracking control optimization scheme of continuous-time nonlinear system via online single network adaptive critic design method. *Neurocomputing*, 251, 127-135.
- Zhang, Y., Li, S., & Liu, X. (2018). Neural network-based model-free adaptive near-optimal tracking control for a class of nonlinear systems. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12), 6227-6241.
- Zhao, S., & Zelazo, D. (2016). Localizability and distributed protocols for bearing-based network localization in arbitrary dimensions. *Automatica*, 69, 334-341.
- Zhao, X., Yang, H., Xia, W., & Wang, X. (2017). Adaptive fuzzy hierarchical sliding-mode control for a class of mimo nonlinear time-delay systems with input saturation. *IEEE Transactions on Fuzzy Systems*, 25(5), 1062-1077.
- Zhao, X.-W., Guan, Z.-H., Li, J., Zhang, X.-H., & Chen, C.-Y. (2017). Flocking of multi-agent nonholonomic systems with unknown leader dynamics and relative measurements. *International Journal of Robuts and Nonlinear Control*, 27(17), 3685-3702.
- Zhou, Y., Li, H., & Yao, H. (2016). Model-free control of surface mounted pmsm drive system. In *Proceedings of 2016 IEEE international conference of industrial technologies (ICIT)* (p. 175-180). Taipei, Taiwan.
- Zhu, Y., & Zhao, D. (2018). Comprehensive comparison of online adp algorithms for continuous-time optimal control. *Artificial Intelligence Review*, 49(4), 531-547.
- Zhu, Y., Zhao, D., & Li, X. (2016). Using reinforcement learning techniques to solve continuous-time non-linear optimal tracking problem without system dynamics. *IET Control Theory and applications*, 10(12), 1339-1347.
- Zuo, S., Song, Y., Lewis, F. L., & Davoudi, A. (2018). Optimal robust output containment of unknown heterogeneous multiagent system using off-policy reinforcement learning. *IEEE Transactions on Cybernetics*, 48(11), 3197-3207.

APPENDICES

APPENDIX A

AFFINE VS NON-AFFINE DYNAMIC SYSTEMS

If the dynamics of a nonlinear system can be presented by

$$\dot{x} = f(x) + g(x)u , \quad (\text{A.1})$$

where u is the input or control variable, then the dynamic system is said to be *linear in control* or *affine* (Slotine & Li, 1991). Otherwise, if the relation between the system dynamics and the control variable is in a general nonlinear form, i.e.

$$\dot{x} = f(x) + h(x, u) ; \quad (\text{A.2})$$

then the system is non-affine. There are some techniques available in the literature that can transform a non-affine dynamic system to be represented in an affine format (Wu et al., 2017).

APPENDIX B

HOMOGENEOUS VS HETEROGENEOUS MULTI-AGENT DYNAMIC SYSTEMS

In homogeneous network or multi-agent dynamic system, all the nodes or agents are identical in terms of the dynamic system. Hence, a similar dynamic system can be considered for representing all of the dynamic agents in the network. On the other hand, two or more different types of dynamic systems are used for representing a heterogeneous network (Mhatre & Rosenborg, 2004). In this regard, the difference in dynamic systems of agents should be taken into account, while dealing with the cooperative control or localization problems in a heterogenous multi-agent dynamic system.

APPENDIX C

LIPSCHITZ CONTINUOUS FUNCTION

We say that the function $f = f(x)$ is *Lipschitz continuous* with Lipschitz constant $\beta \in \mathbb{R}^+$, if the following condition is satisfied (Eriksson et al., 2004)

$$|f(x_1) - f(x_2)| \leq \beta |x_1 - x_2| . \quad (\text{C.1})$$

For instance, a continuous function with bounded first derivative must be Lipschitz continuous.

APPENDIX D

PERSISTENTLY EXCITATION CONDITION

The vector $\phi \in \mathbb{R}^n$ is persistently excited if it satisfies the following inequality

$$\int_t^{t+T_0} \phi(\tau_t) \phi^T(\tau_t) d\tau_t \geq \alpha_0 T_0 I_n , \quad (\text{D.1})$$

for some constant positive values for α_0 and T_0 (Ioannou & Fidan, 2006). Since $\phi \phi^T$ is always positive semi-definite, the PE condition requires that the integral of $\phi \phi^T$ over any time window with length of T_0 be a positive definite matrix (Ioannou & Fidan, 2006).

APPENDIX E

LYAPUNOV STABILITY THEOREM

Suppose that a system is described by $\dot{x} = f(x, t)$, where $f(0, t) = 0$ for all $t \geq 0$.

If there exists a scalar function $V(x, t)$ having continuous, first partial derivatives and satisfying the conditions

- $V(x, t)$ is positive definite (i.e. $V(x, t) > 0$);
- $\dot{V}(x, t)$ is negative definite (i.e. $\dot{V}(x, t) < 0$);

then the equilibrium state at the origin is uniformly asymptotically stable (Khalil, 1996). Note that if $\dot{V}(x, t)$ is negative semi-definite (i.e. $\dot{V}(x, t) \leq 0$), then the equilibrium state at the origin is only stable and no asymptotically convergence is guaranteed.

APPENDIX F

LASALLE-YOSHIZAWA THEOREM

For a system $\dot{x} = f(x, t)$, where $f(x, t)$ is locally Lipschitz in x and piecewise continuous in t , assume that there exists a continuously differentiable function $V(x, t)$ such that along any trajectory of the system, one can have

$$\begin{aligned} \alpha_1 ||x|| &\leq V(x, t) \leq \alpha_2 ||x|| \\ \dot{V}(x, t) &\leq -\alpha_3 ||x|| + \varepsilon, \end{aligned} \tag{F.1}$$

where $\varepsilon > 0 \in \mathbb{R}$, α_1 and α_2 are class \mathcal{K}_∞ functions, and α_3 is a class \mathcal{K} function; then, the solution $x(t)$ of $\dot{x} = f(x, t)$ is *uniformly ultimately bounded* (Fischer et al., 2013; Li & Duan, 2015). Note that a continuous function $\alpha : [0, a) \rightarrow [0, \infty)$ is said to belong to class \mathcal{K} if it is strictly increasing and $\alpha(0) = 0$. It is said to belong to class \mathcal{K}_∞ if $a = \infty$ and $\alpha(r) \rightarrow \infty$ as $r \rightarrow \infty$.

APPENDIX G

KRONECKER PRODUCT

The Kronecker product of matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ is defined as follows

(Li & Duan, 2015)

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}. \quad (\text{G.1})$$

The Kronecker product has the following properties

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$$

$$(A \otimes B)^T = A^T \otimes B^T \quad (\text{G.2})$$

$$A \otimes (B + C) = (A \otimes B) + (A \otimes C).$$

Among the above properties, the first one is named as mixed-product property of the Kronecker product.

APPENDIX H

M-MATRIX

A square matrix $n \times n$ is a singular M-matrix, if all its off-diagonal elements are non-positive and all its principal eigenvalues are non-negative (Lewis et al., 2014). It is a nonsingular M-matrix, if all its principal eigenvalues are positive. In other words, we have

$$M = \begin{bmatrix} + & \leq 0 \\ \leq 0 & + \end{bmatrix}. \quad (\text{H.1})$$

Any M-matrix can be written as $M = sI - A$ for some $s > 0$ and $A \geq 0$ (Lewis et al., 2014).

LIST OF PUBLICATIONS

- Safaei, A. and Mahyuddin, M. N. (2018). Adaptive model-free control based on an ultra-local model with model-free parameter estimations for a generic SISO system. *IEEE Access*, 6, 4266-4275.
- Safaei, A. and Mahyuddin, M. N. (2018). Optimal model-free control for a generic MIMO nonlinear system with application to autonomous mobile robots. *International Journal of Adaptive Control and Signal Processing*, 32(6), 792-815.
- Safaei, A. and Mahyuddin, M. N. (2018). Distributed adaptive model-free cooperative control for a network of generic unknown nonlinear systems. *International Journal of Advanced Robotic Systems*, 15(5), 1-23.
- Safaei, A. and Mahyuddin, M. N. (2018). A Solution for the cooperative formation-tracking problem in a network of completely unknown nonlinear dynamic systems without relative position information. *International Journal of Systems Science*, 49(16), 3459-3475.
- Safaei, A. and Mahyuddin, M. N. (2019). Adaptive cooperative localization using one beacon agent within a network including any number of mobile agents with minimum communication links. *IEEE Access*, 7, 32368-32382.
- Safaei, A., Koo, Y. C. and Mahyuddin, M. N. (2017). Adaptive model-free control for robotic manipulators. In *Proceedings of 2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS2017)* (p. 7-12). Ottawa, Canada.

- Safaei, A. and Mahyuddin, M. N. (2017). An optimal adaptive model-free control with a Kalman-filter-based observer for a generic nonlinear MIMO system. In *Proceedings of IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS2017)* (p. 56-61). Kota Kinabalu, Malaysia.
- Safaei, A. and Mahyuddin, M. N. (2017). Adaptive model-free consensus control for a network of nonlinear agents under the presence of measurement noise. In *2017 11th Asian Control Conference (ASCC2017)* (p. 1701-1706). Gold Coast, Australia.
- Safaei, A. and Mahyuddin, M. N. (2017). Distributed observer for a team of autonomous underwater vehicles utilizing a beacon unit on the surface. In *Proceedings of IEEE 7th Underwater System Technology: Theory and Applications (USYS2017)* (p. 1-5). Kuala Lumpur, Malaysia.
- Safaei, A. and Mahyuddin, M. N. (2018). Application of the optimal adaptive model-free control algorithm on an autonomous underwater vehicle. In *2018 IEEE International Conference on Advanced Robotics and Mechatronics (ICARM2018)* (p. 346-350). Singapore City, Singapore.