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 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. 257

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. 257

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 (bottom), 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, 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. 166

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. 167

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. 169

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. 170

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. 171

Figure 4.22 Schematic of a 2 DOF robotic manipulator in the 2D space (Spong et al., 2006) 172

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. 175

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. 176

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 cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm. 197

Figure 4.41 Adapted values for unknown linear terms at a team of cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm. 198

Figure 4.42 Adapted values for unknown nonlinear terms at a team of cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm. 198

Figure 4.43 Controller gains for each of the cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm. 199

Figure 4.44 Control signals at each of the cooperative autonomous quadrotors, utilizing the CAMFC-1 algorithm. 199

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. 204

Figure 4.46 The values for first states of all agents using the DCC-1 algorithm (top) and the CAMFC-2 algorithm (bottom). 204

Figure 4.47 The values for control signals at all agents using DCC-1 algorithm (top) and the CAMFC-2 algorithm (bottom). 205

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. 205

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. 206

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. 206

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. 207

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. 207

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. 212

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 over-shoot in the response, but faster rising-time. 212

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. 213

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. 213

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. 214

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. 214

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. 215

Figure 4.60 The 3D position of quadrotors in the network using the CAMFC-2 algorithm. 219

Figure 4.61 The X-Y-Z positions of quadrotors in the network using the CAMFC-2 algorithm. 219

Figure 4.62 The consensus errors for translational motion of quadrotors in the network using the CAMFC-2 algorithm. 220

Figure 4.63 The tracking errors for rotational motion of quadrotors in the network using the AMFC algorithm. 220

Figure 4.64 The control signals for speed of electric motors at all quadrotors in the network using CAMFC-2 algorithm. 221

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 and LC algorithms, implemented on a network of non-cooperatively controlled mobile agents (Agent-3 and Agent-4) 236

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) 237

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) 237

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) 238

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) 238

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. 240

Figure 4.84 Localization consensus errors for ACL algorithm used in a team of cooperatively controlled quadrotors. 241

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. 241

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. 242

Figure 4.87 Estimated values for absolute position of the quadrotors in the network using ACL algorithm. 242

Figure 4.88 The schematic of HIL test setup for real-time evaluation of the AMFC algorithm on AMRs 245

Figure 4.89 The x-y position and z-axis rotation of the autonomous WMR in the HIL test. 250

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. 251

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. 251

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. 252

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. 252

Figure 4.94 The autonomous quadrotor position in 3D space and its Euler angles, in the HIL test for evaluating the proposed AMFC algorithm. 253

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 0.6 254

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. 254

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. 255

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. 255

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 ith 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
$u^\star$	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 ith 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

$\mathscr{A}$ adjacency matrix for the communication graph of a multi-agent system

$\mathscr{B}$ pinning gain matrix for the communication graph of a multi-agent system

$\mathscr{B}_b$ beacon pinning gain matrix in the communication graph of a multi-agent system

$\mathscr{D}$ in-degree matrix for the communication graph of a multi-agent system

$\mathscr{L}$ Laplacian matrix for the communication graph of a multi-agent system

$\mathscr{H}$ joint matrix representing the Laplacian and the leader pinning gains for the communication graph of a multi-agent system

$\mathscr{H}_b$ 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 ith row and the jth column of the adjacency matrix in a multi-agent system

β_i the ith diagonal element in the pinning gain matrix corresponding to the agent i in a multi-agent system

β_i^b the ith 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 bersepadu yang terdiri daripada kawalan bebas-model dan algoritma penyetempatan model dibentangkan untuk menangani masalah pengesanan dalam sistem dinamik bukan lelurus yang sepenuhnya tidak diketahui, masalah pengesanan-pembentukan dalam sistem dinamik bukan lelurus 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 *observer*s), 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.

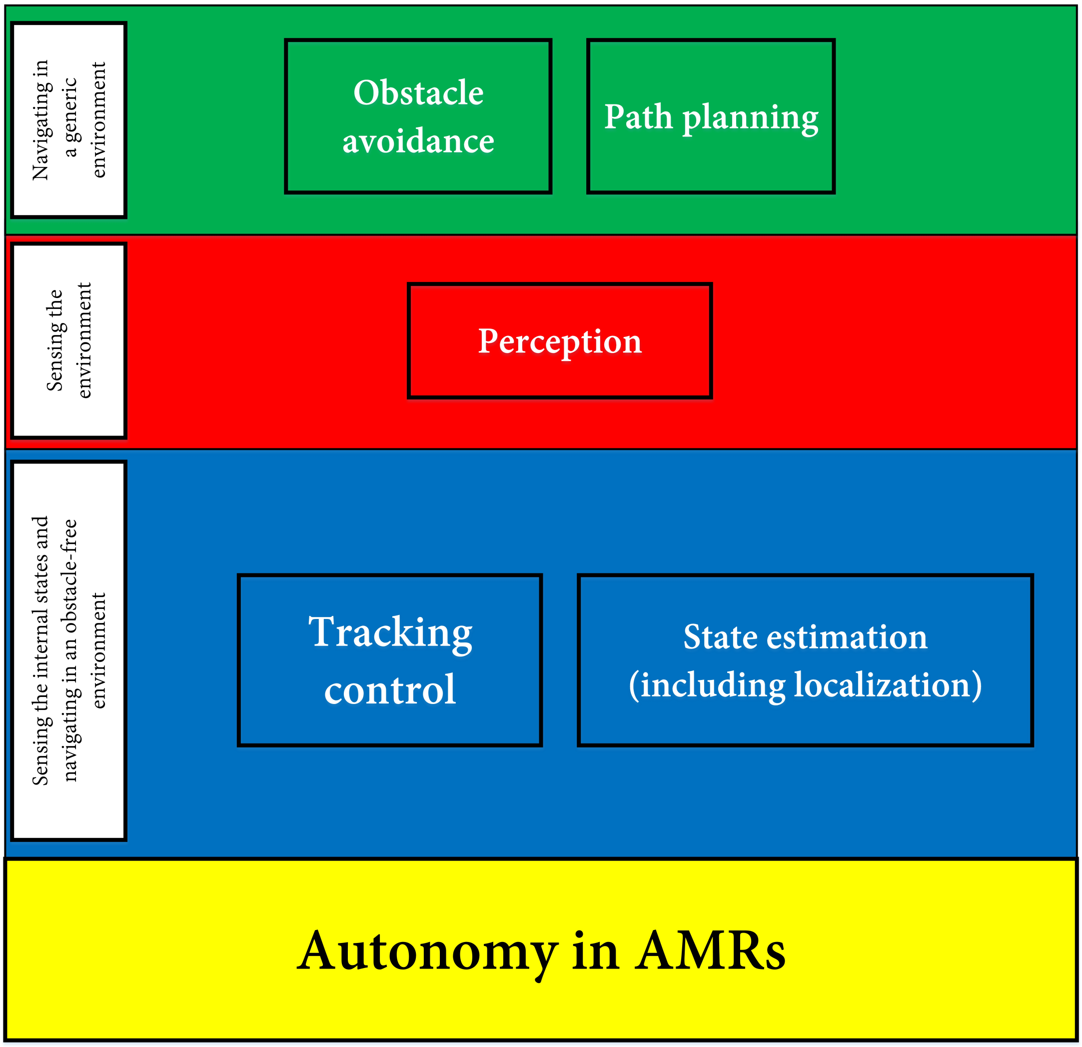

Figure 1.1: The structure of all problems that should be tackled to provide autonomy in AMRs

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)

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

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 & Söffker, 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

$$y = F + \alpha u \ , \tag{2.2}$$

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

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

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

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

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

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

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

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

where $r(.) \in \mathbb{R}$ and $J(.) \in \mathbb{R}$ 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

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

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),

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

and policy improvement

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

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 ith 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).

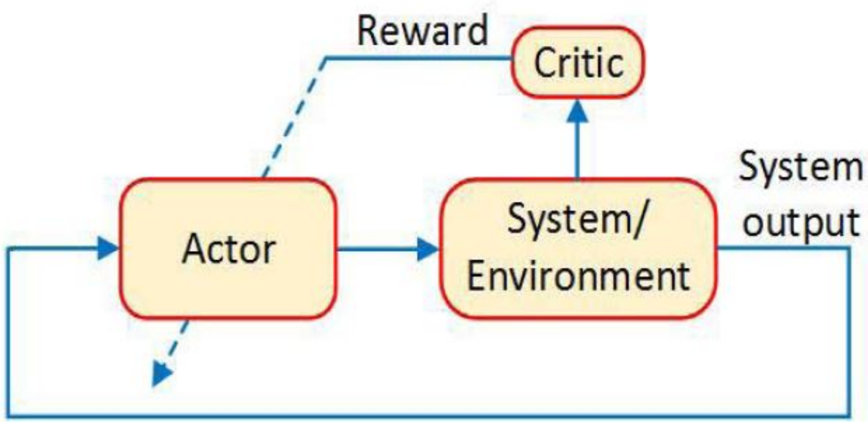

Figure 2.1: The structure of policy iteration method (Kiumarsi et al., 2018)