

**ROBUST BEHAVIORAL-CONTROL OF MULTI-
AGENT SYSTEMS**

BY

YAZAN MOHAMMAD AL-RAWASHDEH

A Dissertation Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

SYSTEMS AND CONTROL ENGINEERING


MARCH 2018

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN- 31261, SAUDI ARABIA
DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Yazan Mohammad Al-Rawashdeh** under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN SYSTEMS & CONTROL ENGINEERING.**



Dr. Hesham K. Al-Fares
Department Chairman

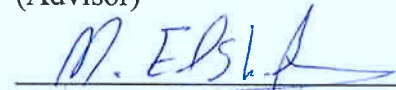


Dr. Salam A. Zummo
Dean of Graduate Studies

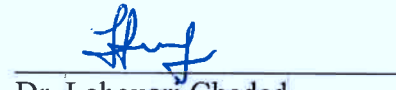
11/4/14
Date




Dr. Sami Elferik
(Advisor)




Dr. Moustafa Elshafei
(Member)



Dr. Lahouari Cheded
(Member)



Dr. Mohammed A. Abido
(Member)



Dr. Muhammad A. Hawwa
(Member)

© Yazan Mohammad Al-Rawashdeh
2018

|

To My Beloved Mother

To the Memory of My Beloved Father

To My Dear Brothers & Sisters

To the Memory of Sheikh Mohammad Al-Luhiadan |

ACKNOWLEDGMENTS

I would like to thank the Ministry of Higher Education in the Kingdom of Saudi Arabia for supporting me during my Ph.D. study. Many thanks go to King Fahd University of Petroleum and Minerals, KFUPM Dhahran-Saudi Arabia, for its endless support. Never forget my thesis advisor, Dr. Sami El-Ferik, for his kindness, encouragement and guidance at all times. Living in such a wonderful academic environment is an indescribable joy for which I am grateful to all people in our department for taking care of every aspect affecting our academic prosperity with great devotion. Great thanks go as well to my thesis committee members Dr. Elshafei, Dr. Cheded, Dr. Abido and Dr. Hawwa for their valuable comments and suggestions which helped in forming my thesis in its current form. From the bottom of my heart, I am indebted to my family for expressing patience, for their prayers and wishes, and for sharing the journey with me despite the distances separating us. Their positive influence cannot be expressed by words. At last, I would like to ask Almighty ALLAH to have mercy on the soul of the businessman Sheikh Mohammad Al-Luhiadan for supporting me during my Ph.D. study through the University Endowment department.

Yazan M. Al-Rawashdeh |

TABLE OF CONTENTS

ACKNOWLEDGMENTS	V
TABLE OF CONTENTS	VI
LIST OF TABLES	IX
LIST OF FIGURES	X
LIST OF ABBREVIATIONS	XXV
ABSTRACT	XXVII
ملخص الرسالة	XXIX
CHAPTER 1: INTRODUCTION	1
1.1 Algebraic Graph Theory	2
1.2 Non-negative Matrices	5
1.3 Thesis Motivation	8
1.4 Context-aware Intelligent Controllers	11
1.5 Thesis Layout	18
CHAPTER 2: LITERATURE REVIEW	21
CHAPTER 3: THE FUNDAMENTALS OF THE PROPOSED FRAMEWORK	40
3.1 Consensus and Clustering of N-identical Single-Integrator MAS over State-dependent Network Protocols	40
3.1.1 Arithmetic Mean Consensus	45
3.1.2 Geometric Mean Consensus	48
3.1.3 Harmonic Mean Consensus	49
3.1.4 Mean-of-order- p Consensus	50
3.2 The proposed framework	52
3.2.1: Nonlinear Eigenvalue-based Design Approach	52

3.2.2: First Integral Design Approach	66
3.2.3: Relation between first integral and nonlinear eigenvalue approaches.....	69
3.2.4: Nonlinear Distributed Consensus Protocols over undirected graphs using the proposed framework	70
3.2.5: Importance of the nonlinear right eigenvector	85
3.2.6: The concept of prescribed connectivity	88
3.2.7: Stability and Convergence Analysis	93
3.2.8: Performance Analysis	98
3.2.9: Analyzing consensus protocols using vector calculus.....	106
3.2.10: Constructing consensus protocols using the proposed framework:	110
3.2.11: Analyzing the C-S model using the proposed framework	113
3.2.12: Different Types of <i>g_{ij}</i> Functions.....	116
3.3 Simulation Results	118
3.3.1: Behaviors using semi-linear network protocols	118
3.3.2: Behaviors using nonlinear network protocols.....	123
3.4 Conclusion	128
 CHAPTER 4: STATE/DISTANCE-DEPENDENT BEHAVIORS	 129
4.1 The framework Utilization in state-dependent and motion-related behaviors realization	129
4.1.1: Consensus using the common neighborhood approach	133
4.1.2: Formation in obstacle-free working environment	143
4.1.3: Deployment in obstacle-free 3-D space	176
4.1.4: Connectivity-preserving protocols	200
4.1.5: Shape consensus and clustering behaviors	224
4.1.6: Containment and Escorting in obstacle-free 2-D plane.....	236
4.1.7: Switching behaviors.....	239
4.1.8: Agents with scalar and nonlinear affine-in-control-input	243
4.2 Conclusion	247
 CHAPTER 5: ADVANCED BEHAVIORS	 249
5.1 MAS utilizing trajectory-generators	251
5.1.1: Stability of augmented systems.....	258
5.1.2: Effect of agents dynamics on the MAS connectivity	262
5.2 Kino-dynamics of Trajectory-generator systems	274
5.2.1: Introducing acceleration signals into the trajectory-generator kinematics	274
5.2.2: Designing harmonic potential fields using the proposed framework	281
5.2.3: Modeling the environment into the Kino-dynamics of trajectory-generator systems.....	287
5.2.4: Sensing the environment using on-board sensors	290
5.2.5: The Complete picture of trajectory-generator systems	292

5.3 Multi-agent Systems of N-Identical double-integrator systems.....	297
5.3.1: Similar communication graphs	299
5.3.2: Distinct communication graphs.....	300
5.3.3: Studying the invariant quantity of MAS comprising N -identical double-integrator systems.....	301
5.4 Multi-agent Systems of N-Identical coupled-integrators systems	302
5.5 Multi-agent Systems of N-Identical general LTI systems	308
5.6 Multi-agent Systems of N-Identical nonlinearly-coupled higher-order integrator systems.....	320
5.6.1: Dynamic containment-escorting behaviors.....	324
5.7 Multi-agent Systems of N-Identical affine-in-input nonlinear systems	331
5.7.1: Dynamic containment-escorting-orbiting behaviors.....	332
5.7.2: Non-holonomic wheeled robots.....	338
5.7.3: Under-actuated Flying robots.....	341
5.8 Conclusion	358
 CHAPTER 6: THE POSSIBLE EXTENSIONS	 360
 REFERENCES.....	 364
 VITAE	 384

LIST OF TABLES

Table 5.1	: Consensus Tracking using (5.23.3) when $k_2=20$	270
Table 5.2	: Consensus Tracking using (5.23.4) when $k_2=20$	271
Table 5.3	: Consensus Tracking using (5.33).	273

LIST OF FIGURES

Figure 1.1	: A mixed multigraph consists of three connected agents	4
Figure 1.2	: Search and Rescue Team in an unknown devastating environment	9
Figure 1.3	: First hierarchical structure of agents according to the proposed SRM perspective	9
Figure 1.4	: Second hierarchical structure of agents according to the proposed SRM perspective	10
Figure 1.5	: Third hierarchical structure of agents according to the proposed SRM perspective	11
Figure 1.6	: General structure of a Reactive Controller	12
Figure 1.7	: Integration between the robot local and the reactive controllers	13
Figure 1.8	: Graphical representation of the reactive force used in the reactive controller	13
Figure 1.9	: Structure of Model-based Motor Schema intelligent controller	14
Figure 1.10	: Some of motion-related behaviors under MMS intelligent controller	14
Figure 1.11	: Desired force fields along the x and y axes	15
Figure 1.12	: Resulting reference signal generated by the motion planner under MMS intelligent controller	16
Figure 1.13	: Hybrid schema integrating both the RC and the MMS intelligent controllers	16
Figure 1.14	: The implementation of HMR controller using one preferred configuration	17
Figure 1.15	: Multi-agent system in obstacle-free context	17

Figure 1.16	: Multi-agent system in obstacle context	17
Figure 1.17	: Relation between distributed control and communication protocols	18
Figure 3.1	: A regular undirected link	54
Figure 3.2	: The effect of having $\mathcal{L}(\vec{x})$ linearly dependent on \mathcal{L}_f	54
Figure 3.3	: MAS with multiple neighborhoods	61
Figure 3.4	: First Integral subset formed by the initial conditions of three dynamical agents in 3D space	70
Figure 3.5	: two single-integrator agents interacting over an undirected link	77
Figure 3.6	: Effect of using protocols (3.3) and (3.72) on the domain of interest	84
Figure 3.7	: three single-integrator agents interacting over an undirected graph under (3.54).	87
Figure 3.8	: A MAS with permanent and intermittent link connecting agents within multiple neighborhoods	91
Figure 3.9	: A MAS of two agents connected by two links (edges).	108
Figure 3.10	: The control signal of (3.108) when g_{12} being the C-S model with different values of β	114
Figure 3.11	: The control signal of (3.108) when g_{12} being the C-S model with different values of H, γ and δ are used with $\beta = 0.5$ in all cases. The values are given for the upper part of the graph from left to right.	115
Figure 3.12	: Visualization of g_{ij} being as an activation function used in NN	117
Figure 3.13	: A general undirected graph comprising six agents	118
Figure 3.14	: Arithmetic mean consensus among agents using protocol (3.5) utilizing (3.124). The first row from left	119

to right: states responses, control signal and eigenvalues. The second row is their corresponding distance-dependent quantities, where eigenvalue interlacing is discussed in Remark 3.16.

Figure 3.15	: Figure 3.15. Different types of mean consensus among agents using protocol (3.39) utilizing (3.124). (a): states responses. (b): Algebraic connectivity $ \lambda_2 $, in red, with upper bound, in black, evaluated using (3.101) at each time instance for different types of means. The eigenvalue evolution is discussed in Remark 3.16. 120
Figure 3.16	: Figure 3.16. States responses while achieving consensus to different types of mean using protocol (3.39) utilizing (3.124) and the graph shown in Figure 3.13. 121
Figure 3.17	: States responses while achieving different types of mean consensus among agents using protocol (3.39) utilizing (3.121). 122
Figure 3.18	: Clustering example available in [1] solved here using protocol (3.6) utilizing (3.88) with $\beta = 1$ 123
Figure 3.19	: MAS response with or without prescribed connectivity levels over a complete graph and under (a): relatively small initial conditions and (b): large initial conditions cause the expected time for consensus to occur as $t \rightarrow \infty$.	123- 124
Figure 3.20	: States responses while achieving different types of mean consensus among agents using nonlinear protocol (3.54) over a complete graph 124
Figure 3.21	: States responses while achieving different types of mean consensus among agents using nonlinear protocol (3.54) over the graph shown in Figure 3.13 125
Figure 3.22	: States responses while achieving different types of mean consensus among agents using nonlinear protocol (3.67.15) over a complete graph. To achieve 125

the harmonic mean, the convergence rate was 30 times larger than what was used with the other two types

Figure 3.23	: States responses while achieving the average consensus among agents using the graph shown in Figure 3.13 utilizing a semi-linear protocol with C-S model and a prescribed connectivity of 0.006 with $\beta = H = \gamma = \delta = 1$. (b): Point-wise calculation of $\lambda_2(\mathcal{L}(\vec{x}))$ to have an insight about the propagation of the algebraic connectivity of the MAS as will be justified in Part II. See Remark 3.16 126
Figure 3.24	: States responses while achieving consensus among agents using the graph shown in Figure 3.13 utilizing (3.68) with sign function (top) and tanh function (bottom), with $\varpi = 0.3$ and $\alpha = 0.1$, where the two black dashed lines represent the constraints 127
Figure 4.1	: An agent behavior-bank consists of k -motion-related behaviors 133
Figure 4.2	: MAS with multiple neighborhoods created by the operator 134
Figure 4.3	: Response with prescribed connectivity 0.006 using the concept of multi-neighborhood. Compare it with Figure 3.23.a 134
Figure 4.4	: A MAS with permanent and intermittent link connecting agents within multiple neighborhoods and no prescribed connectivity. (a): illustration. (b): response with intermittent links enabled with threshold equals 0.1 and (c): disabled	135- 136
Figure 4.5	: A MAS response while achieving the average consensus under the global neighborhood 137
Figure 4.6	: A MAS response under the global neighborhood while achieving the average consensus on 2-D working space 137

Figure 4.7	: A universal neighborhood comprising multiple complete and partial neighborhoods among which permanent and persistent links may exist 143
Figure 4.8	: An illustration of the essence of formation protocol (4.22) showing two agents and the relative distances among them using a moving reference frame 150
Figure 4.9	: Formation control of three single-integrator agents interacting over an undirected graph under: (a) protocol (4.16). (b) protocol (4.22). 153
Figure 4.10	: Communication signal power in the working space 160
Figure 4.11	: A hybrid automaton used in autonomous decision making 160
Figure 4.12	: Evolution of SS_T over time using the finite state-machine shown in Figure 4.11 160
Figure 4.13	: Control signal of the overall MAS during the whole switching 161
Figure 4.14	: Final behavior of the MAS is consensus; since the average value of agents initial condition was outside the desired region of $SS_i(\vec{x}, \vec{y})$ 161
Figure 4.15	: Final behavior of the MAS is formation; since the average value of agents initial condition was inside the desired region of $SS_i(\vec{x}, \vec{y})$ 161
Figure 4.16	: Finite state-machine used to drive agents towards desired region of $SS_i(\vec{x}, \vec{y})$ where formation is required 162
Figure 4.17	: Evolution of SS_T over time using the finite state-machine shown in Figure 4.16 162
Figure 4.18	: Example 4.1: (a) the 2-D working space and the leader path along with the available signal strengths contours with zero level at -30 dBm. (b): 3D view of the working space overlapped with the received signal power including path loss, shadowing and multipath fading	168- 169

	effects. (c): communication network topology connecting agents through the router. (d): the global interaction neighborhood, i.e., shape, graph used	
Figure 4.19	: Example 4.1: (a) the value of $\overline{SS_T}$ obtained for different NSBS with/out incorporating the multipath fading effect as a noise source. (b) the corresponding formation radii	170- 171
Figure 4.20	: Example 4.2: the 2-D working space and the leader path along with the available signal strengths contours with zero level at -30 dBm 173
Figure 4.21	: Example 4.2: (a) the total received signal power and the resulting formation radius. (b)-(e) agents positions and corresponding control signals without and with incorporating the multipath fading effects as noise. (f) a close view on (e) at 346 seconds which looks like the control signal in (d).	174- 175
Figure 4.22	: The equilibria set (the lamina in red) introduced by (4.34). 178
Figure 4.23	: The value of $\lambda_2^{\{1,2\}}(\vec{x})$ of edge $\{1,2\} \in \mathcal{E}$ using C-S model 183
Figure 4.24	: The phase portrait of (4.36.3) with C-S model used in g_{ij} functions 184
Figure 4.25	: The value of $\lambda_2^{\{1,2\}}(\vec{x})$ of edge $\{1,2\} \in \mathcal{E}$ using C-S models, (solid): (4.39), (dashed): (4.39.2) with $\beta = 0.5$, (dotted): (4.39.2) with $\beta = 0.5$ and neglecting the higher-order-terms (H.O.T), all with $\gamma = H = \delta = \alpha = 1$ 184
Figure 4.26	: Mechanical analogy of edge $\{i,j\}$ under deployment protocol with the reference exactly at the average value of the two agents 190
Figure 4.27	: Phase portrait of: (a) the first agent. (b) the second agent both under deployment protocol (4.34) with $TH = 0.1, H = \delta = \gamma = 1$ and $k = 10$ 190

Figure 4.28	: Phase portrait of: (a) the first agent. (b) the second agent both under deployment protocol (4.35) with $g(x_i) = x_i$, i.e., geometric mean deployment. (c) the first agent. (d) the second agent both under deployment protocol (4.35) with $g(x_i) = x_i^2$, i.e., harmonic mean deployment all with $TH = 0.1$, $k = 10$ and $H = \delta = \gamma = 1$.	191- 192
Figure 4.29	: The basis function of $\lambda_2^{\{i,j\}}(\vec{x})$ of edge $\{i,j\} \in \mathcal{E}$ using (4.49) with $TH = 0.1$, $H = \delta = k = \beta = 1$ and $\gamma = 10$ 193
Figure 4.30	: Deployment of six single-integrator agents in obstacle-free 3-D space. (a): Case 1 without disturbance. (b)-(c): Disturbance acting only on agent 1 at 100 second for 15 second duration under Case 1. (d): Case 2 deployment response when agents are initially close to each other. (e): the corresponding overall connectivity of Case 1. (f): edges contributions in the overall connectivity of Case 1. (g): the corresponding overall connectivity of Case 2. (h): edges contributions in the overall connectivity of Case 2. (i) connectivity of the two cases. (j) control signals in the two cases	196- 198
Figure 4.31	: MAS response when agents have different TH values in (4.34) along the x-axis. The $eig(\mathcal{L}(\vec{x}))$ MATLAB function was used	199- 200
Figure 4.32	: Single-edge response due to a multistep input under (4.52.2). 203
Figure 4.33	: Simulation results of agents' deployment in 3-D using the universal protocol (4.47.8). (a) agents' positions under average deployment. (b) average deployment using (4.34) with disturbance enables- only agents positions along the x-axis are shown. (c) another experiment of average deployment using (4.47.8) with disturbance enables- only agents positions along the x-axis are shown. (d) the three types of mean deployment shown along with the needed control signals with 20	206- 207

times increase in the convergence rate for the last two types. (e) average deployment of the same agents using (4.47.8) over a complete graph with identical TH_{ij} values

Figure 4.34	: Simulation results of agents' formation in 3-D using the universal protocol (4.47.8). (a) trajectories. (b) 3-D minimum distance among agents during the whole course of formation. (c) agents' positions in 3-D and the needed control signals	209- 210
Figure 4.35	: Simulation results of agents' formation in 2-D using the universal protocol (4.48) over a general network while changing the undirected weights, i.e., unbalanced, or making an edge directed. (a)-(e) agents' positions. (f) minimum distance to collide	210- 211
Figure 4.36	: A communication graph showing the links and edges connectivity gradients from the perspective of agent 2 215
Figure 4.37	: The control signals under multi connectivity-preserving protocols running in parallel with another consensus protocol realized using the C-S model with $\beta = 2$ with a convergence rate $\alpha = 100$ and the contribution of eigenvector elements was ignored 217
Figure 4.38	: The simulation results of the MAS shown in Figure 4.36 with and without the connectivity-preserving protocols while achieving the average consensus. The maximum separation was between agents 1 and 3 along the x-axis and was equal to 955 length units 218
Figure 4.39	: Simulation results of agents' formation in 3-D using the universal protocol (4.54) over a general network while changing the undirected weights, i.e., unbalanced, or making an edge directed. (a) -(c) agents' positions. (d) agents' positions versus time and the needed control signal for (c).	223- 224
Figure 4.40	: Two clusters C_1 and C_2 with agent 1 pinned to the virtual leader / target point (triangle) selected by an operator 226

Figure 4.41	: The simulation results of the example shown in Figure 4.40 using the 3-D version of the C-S model with $\gamma = 1$, $H = 10$, $\delta = 0$ and $\beta = 0.5$. (a) agents' positions along the x-axis. (b) agents' positions in 3-D versus time. (c) agents' positions in the working space. (d) min-max distance among neighboring agents on the graph shown in Figure 4.36. (e) the equilibria of (4.55.4) where all equilibria can be easily written in terms of x_1 starting with x_2 and x_3	227- 228
Figure 4.42	: The simulation results of the example shown in Figure 4.40 using the 3-D version of the C-S model with $\gamma = 1$, $H = 10$, $\delta = 0$ and $\beta = 0.5$ and the modified clustering weights given by (4.55.5). (a) agents' positions along the x-axis. (b) agents' positions in 3-D versus time. (c) agents' positions in the working space. (d) min-max distance among neighboring agents on the graph shown in Figure 4.40. (e) the equilibria of edge $\{1,2\} \in \mathcal{E}$ written in terms of x_1 and x_2 when (4.55.5) is used	228- 230
Figure 4.43	: The simulation results of the example shown in Figure 4.40 using the 3-D version of the C-S model with $\gamma = 1$, $H = 10$, $\delta = 0$ and $\beta = 0.5$ and the modified clustering weights given by (4.55.5). (a)-(b) and (c)-(d) agents positions along the x-axis and the minimum and maximum distance among neighboring agents on the graph shown in Figure 4.40 without and with the clustering size control with $TH = 81$ respectively, and $p_i = 1$	230- 231
Figure 4.44	: Effect of cluster size control approach on clustering behavior 233
Figure 4.45	: Shape control results under C-S distance-based protocol where agents $\{6\}$, $\{2\}$, $\{4\}$, $\{1,3\}$ and $\{5,7\}$ form five clusters in the x-direction while agents $\{3,5\}$, $\{2,4,6\}$ and $\{1,7\}$ form three clusters in the y-direction. Agent 4 is the center of this shape 234

Figure 4.46	: Network topology used in 2-D shape control shown in Figure 4.38 235
Figure 4.47	: Network topology used in containment and escorting example 236
Figure 4.48	: Desired containment region, its approximate side view and the available forces among agents 237
Figure 4.49	: Containment and escorting results. (a): containment phase. (b): close view. (c): Agent 2 is initially close to agent 1. (d): escorting phase where all agents are connected to the virtual leader	238- 239
Figure 4.50	: The switching behaviors results. (a): in 2-D. (b): Agents y-positions 243
Figure 5.1	: The tracking error of the i^{th} agent 253
Figure 5.2	: The local and tracking controllers of the 1^{st} agent 254
Figure 5.3	: Three identical agents with scalar dynamics building a MAS over an undirected graph without tracking controller. Left: Network topology. Middle: Signal flow diagram. Right: Effect of local controller. 254
Figure 5.4	: The effect of using different couplings in g_{ij} functions. (a) $g_{ij}(\Gamma_i, \Gamma_j)$. (b) $g_{ij}(x_i, \Gamma_j)$. (c) $g_{ij}(\Gamma_i, x_j)$. (d) $g_{ij}(x_i, x_j)$. Examples are shown specifically for agent 1 255
Figure 5.5	: Visualization of the effect of using different couplings in g_{ij} functions where stars denote the actual systems states and the squares denote the states of their trajectory-generator systems 256
Figure 5.6	: The containment behavior of six nonidentical agents with scalar dynamics when $g_{ij}(\Gamma_i, \Gamma_j)$ under a disturbance acting on agent 3 between 5-20 seconds. (a) reference trajectory. (b) actual system response 260

Figure 5.7	: The containment behavior of six nonidentical agents with scalar dynamics under a disturbance acting on agent 3 between 5-20 seconds: ideal trajectories denoted by solid lines and actual trajectories denoted by dashed lines, when (5.3) uses: (a)-(b) reference trajectory and actual system response when $g_{ij}(x, x_j)$. (c) actual systems position along the x-axis and the needed control signals when $g_{ij}(x_i, x_j)$. (d) control signal of the trajectory-generator along the x-direction when $g_{ij}(x_i, x_j)$.	261- 262
Figure 5.8	: The tracking error of all agents during the containment behavior using (5.9). 262
Figure 5.9	: A visualization of the effect of adding a connectivity-preserving controls to (5.6) in (red). Stars denote the actual positions of agents and actual directions of motion (blue), while squares denote the ideal trajectory to follow (black). The dashed lines are the resultants directions to follow 264
Figure 5.10	: A realization of the concept of Node-to-Node behavior in view of the current investigation. SA denotes the scout agents. CA denotes the communication agents. AA denotes the active agents 273
Figure 5.11	: A visualization of a MAS system navigating a working space with obstacles approximated as point obstacles whose field strength is shown in red. Each agent is building a source potential field to avoid colliding with its neighbors on the graph 2888
Figure 5.12	: Rolling around the obstacles boundaries in favor to increase connectivity with other graph neighboring agents 289
Figure 5.13	: A situation where the connectivity-preserving gradient and the two tangents are perpendicular 290
Figure 5.14	: The soft-force as a function of an agent position with respect to an obstacle surface. The non-localized form 291

in red and the localized form in blue where K is a real constant

Figure 5.15	: A wheeled-robot having four symmetrically distributed identical proximity sensors onboard. The global inertial coordinate frame is IXY and the local coordinate frame is cxy . The orientation of the i^{th} agent is given with respect to IXY 292
Figure 5.16	: The overall structure of the proposed trajectory-generator system 293
Figure 5.17	: Simulation results of the 2-D containment behavior realized by six heterogeneous linear and scalar systems where stationary obstacles exist in the working space. The red squares represent the initial positions (a) an overview of the complete behavior. (b) a close view of the containment behavior. (c) the needed control signals in the x-direction. (d) the tracking error during the behavior	293- 294
Figure 5.18	: Simulation results of the 2-D consensus behavior realized by six heterogeneous linear and scalar systems where stationary obstacles exist in the working space. The red squares represent the initial positions (a) a 3-D overview of the complete behavior. (b) a close 2-D view of the working space and the trajectory of agents during behavior conduction. (c) the tracking errors. (d) the control signals in the x-direction. (e) the same behavior when obstacle avoidance is inactive. (f) the needed control signals in the x-direction when working in an obstacle-free working space	295- 296
Figure 5.19	: The orbiting behavior. (a): 3-D view of trajectories of all agents. (b): synchronization of trajectories 308
Figure 5.20	: Three 2 nd -order LTI agents with a single leader where each follower state is connected via a different network topology. The same can be said about the pinning gains denoted by the letter h 313

Figure 5.21	: Three 2 nd -order LTI agents with a single leader where each followers state is connected via a different network topology where the self-loops added are evident 316
Figure 5.22	: The resulting dynamics of the first 2nd-order LTI agent under (5.55). 318
Figure 5.23	: The 3-D containment behavior with $g_{ij}^{x,y,z}(\Gamma_i, \Gamma_j)$. (a) ideal trajectories of all agents. (b) ideal trajectory of agent 3. (c) algebraic connectivity of all networks using the MATLAB <i>eig</i> (\cdot) function	322- 323
Figure 5.24	: The 3-D containment behavior with $g_{ij}^{x,y,z}(x_i, x_j)$ in all networks. (a) ideal trajectories of all agents. (b) algebraic connectivity of all networks using the MATLAB <i>eig</i> (\cdot) function 323
Figure 5.25	: The 3-D dynamic containment behavior with $g_{ij}(\Gamma_i^x, x_j)$ for all networks. (a): ideal trajectories of all agents. (b): ideal trajectory of agent 3. 324
Figure 5.26	: Difference between parallel and sequential behaviors 325
Figure 5.27	: The 3-D dynamic containment-escorting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ for all networks. (a): ideal trajectories of all agents. (b): ideal trajectories of agent 3 while escorting agent 1 combining both containment and escorting behaviors	326- 327
Figure 5.28	: The 3-D dynamic containment-escorting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ for all networks $\{a_{xx}, a_{yy}, a_{zz}, a_z\} = 0$ and $\{a_x, a_y\} = 1$. (a) ideal trajectories of all agents. (b) ideal trajectories of agent 3. 327
Figure 5.29	: Alternating ideal trajectories of agents 4 and 5 during the 3-D dynamic containment-escorting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ for all networks $\{a_{xx}, a_{yy}, a_{zz}, a_x\} = 0$ and $\{a_y, a_z\} = 1$. (a) ideal trajectory signals. (b) actual agents' trajectories. 328

Figure 5.30	: Resulting trajectories during containment behavior. (a) tracking error of all agents under (5.58.6). (b) actual trajectories of agent 3 under (5.58.6). (c) tracking errors of all agents under (5.58.4). 330
Figure 5.31	: The 3-D dynamic containment-escorting-orbiting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ when A in (5.59.1) is Hurwitz 332
Figure 5.32	: The 3-D dynamic containment-escorting-orbiting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ and additional damping given by (5.59.1). (a) containment-orbiting trajectory-generator signals. (b) actual trajectory of all agents during the complete behavior when agent 1 is initially at the origin. (c) actual trajectory of all agents during the complete behavior when agent 1 is initially away from the origin	332- 333
Figure 5.33	: The 3-D dynamic containment-orbiting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ and additional damping given by (5.59.1) when agents are spaced on a hemisphere. (a)-(b) not equally or ellipsoid like. (c)-(d) equally or circle like	334- 335
Figure 5.34	: Results of dynamic containment-escorting-orbiting during dynamical single-point-obstacle avoidance. (a) ideal trajectories of all agents due to containment-orbiting trajectory-generator system. (b) actual trajectory of all agents during the complete behavior 336
Figure 5.35	: Results of unplanned attack of four agents targeting an asset protected by dynamic containment-escorting-orbiting utilizing five agents. (a): ideal trajectories in 2-D. (b): actual trajectories in 2-D. (c): actual trajectories of 3-D followed by 2-D dynamic containment-escorting-orbiting behaviors based on the time elapsed	337- 338
Figure 5.36	: Simulation results of guiding a group of six FSR in a working space that contains stationary obstacles. (a) trajectory-generator, nonlinear transformation and FSR integration. (b) top-view of working space and FSR actual trajectories. (c) FSR positions versus time. (d)	340- 341

tracking error of order 10^{-8} . (e) control signal along the x-axis

Figure 5.37	: The communication and interaction graph used in quadrotors' simulation example	346
Figure 5.38	: The Simulink model used in simulating the dynamic containment behavior under (5.67).	346-347
Figure 5.39	: The simulation results for case 1 without disturbances acting on quadrotor 2. (a) Actual 3-D trajectories for all agents. (b) Actual 3-D trajectories for the second quadrotor. (c) its attitude. (d) the tracking errors	348-349
Figure 5.40	: The simulation results for case 1 with disturbances acting on quadrotor 2. (a) Actual 3-D trajectories for all agents. (b) Actual 3-D trajectories for the second quadrotor. (c) its attitude. (d) the tracking errors	350-351
Figure 5.41	: The simulation results for case 2 with disturbances acting on quadrotor 2. (a) Actual 3-D trajectories for all agents. (b) Actual 3-D trajectories for the second quadrotor. (c) its attitude. (d) the tracking errors.	352-353
Figure 5.42	: The simulation results for case 3 with disturbances acting on quadrotor 2. (a) Actual 3-D trajectories for all agents. (b) Actual 3-D trajectories for the second quadrotor. (c) its attitude. (d) the tracking errors	353-355
Figure 5.43	: The simulation results for case 4 with disturbances acting on quadrotor 2. (a) Actual 3-D trajectories for all agents. (b) Actual 3-D trajectories for the second quadrotor. (c) its attitude. (d) the tracking errors	355-357
Figure 5.44	: A rescue-team working in a devastated environment	358

LIST OF ABBREVIATIONS

AA	: Active Agents
ARE	: Algebraic Riccati Equation
BER	: End-To-End Bit Error Rate
CA	: Communication Agent
C-S	: Cucker-Smale Flocking Model
DSC	: Dynamic Surface Control
FSR	: Front-Wheel Steered Robots
H.O.T	: Higher-Order-Terms
HMR	: Hybrid Model-Based Reactive
HPF	: Harmonic Potential Fields
ISS	: Input-to-State Stable
LQR	: Linear Quadratic Regulator
MANET	: Mobile Ad Hoc Networks
MAS	: Multi-Agent Systems
MMS	: Model-Based Motor Schema
NSBS	: Number- Of-Samples-Before-Scaling-Or-Switching
OPR	: Operating Regions
PDEs	: Partial Differential Equations
PLGs	: Programmable Logic Gates
RBF	: Radial Basis Functions
RC	: Reactive Controller
SA	: Scout Agent

SDRE	:	State-Dependent Riccati Equation
SMC	:	Sliding Mode Controller
SRM	:	Search and Rescue Missions
UAVs	:	Un-Manned Aerial Vehicles
UGVs	:	Un-Manned Ground Vehicles
VSS	:	Variable Structure System
www	:	World Wide Web

ABSTRACT

Full Name : Yazan Mohammad Suliman Al-Rawashdeh
Thesis Title : Robust Behavioral-Control of Multi-Agent Systems
Major Field : Systems and Control Engineering
Date of Degree : March 26, 2018

A unifying framework for designing distributed semi-linear and nonlinear state-dependent protocols to control the behavior of multiagent systems over communication networks is presented. The building blocks of the proposed framework are detailed, and its generality is demonstrated through comparing it to some major results available in the related literature. The applicability of the framework in producing several behaviors that govern the interaction among the connected spatial agents is demonstrated through mainly simulation where proofs of stability and convergence are provided. Various behaviors of the connected agents, mainly on undirected graphs, are achieved by well-designed couplings where consensus, clustering, shape consensus, formation, deployment, containment and escorting are some examples of the achievable behaviors. Composite behaviors stored in a behavior bank can be selected by a suitable behavior selection mechanism that is controlled directly by the agent's embedded artificial intelligence or indirectly through a mission planning utility.

The integration of first integrals and nonlinear eigenvalue problems constitute the core of this framework. Stability issues are mainly handled using properties of M-matrices and Lasalle's principle. The relation between consensus protocols and potential fields are explained and utilized. Simple examples to show the applicability of the proposed framework in designing bounded controls that meet a prescribed performance are also provided.

Using state-dependent parameterization to control the switching between the various behaviors is also presented. The multitude of primitive behaviors are used to build more sophisticated behavioral banks that reside in each agent permitting each agent to choose or follow this chosen behavior. Designing connectivity-preserving protocols is also addressed. The result is a sophisticated distributed coordination motion planner.

The agents under consideration could be maritime, airborne or ground robots. Both kinematical and kino-dynamical trajectory-generator systems are developed and integrated with reactive and hybrid-model-based-reactive intelligent controllers to facilitate interacting with more realistic working spaces. Harmonic potential fields are utilized to model the environment to enable collision-avoidance. Both collision-avoidance and connectivity-preserving behaviors are combined in a simple, yet, efficient way.

Second-order, general linear time-invariant, nonlinear systems, and other models with higher dimensions are presented, where a special representation methodology reveals the usefulness of the proposed framework when dealing with such systems. The design steps presented can be easily upgraded to deal with systems of heterogeneous dynamical features.

The framework provides the means to facilitate building robust behaviors that can handle disturbances, noise and uncertainties in the decision-making process and the involved agents' dynamics. In the latter case, the integration with a robust local controller is made possible under the proposed framework.

ملخص الرسالة

الاسم الكامل: يزن محمد سليمان الرواشده

عنوان الرسالة: التحكم المُحكّم في سلوك الأنظمة متعددة الأطراف

التخصص: هندسة الأنظمة و التحكم

تاريخ الدرجة العلمية: ٢٦ آذار ٢٠١٨

في هذه الدراسة يتم تقديم إطار موحد لتصميم البروتوكولات شبه الخطية وغير الخطية المعتمدة على الدوال للتحكم في سلوك الأنظمة متعددة الأطراف المتصلة عبر شبكات الاتصالات. يتم أيضاً تفصيل اللّبنات الأساسية للإطار المقترح ، ويتم إثبات عموميتها من خلال مقارنتها ببعض النتائج الرئيسية المتوفرة في الأدبيات ذات الصلة. إن قابلية تطبيق الإطار في إنتاج العديد من السلوكيات التي تحكم التفاعل بين الأنظمة المكانية المتصلة يتم إثباتها من خلال المحاكاة بشكل أساسي حيث يتم توفير إثباتات الاستقرار والتقارب. إن السلوكيات المختلفة للأنظمة المتصلة ، وبشكل رئيسي عبر الشبكات ثنائية الإتجاه ، يتم تحقيقها بواسطة متغيرات التوصيل الجيدة التصميم حيث التوافق و التجميع و تكوين الإجماع و التشكيل و الإنتشار و الاحتواء والمرافقة هي بعض الأمثلة على السلوكيات القابلة للتحقيق. يمكن اختيار السلوكيات المركبة و المخزنة في بنك السلوك عن طريق آلية اختيار السلوك المناسبة التي يتم التحكم فيها مباشرة بواسطة الذكاء الاصطناعي الداخلي للنظام أو بشكل غير مباشر من خلال آلية تخطيط المهمة.

ويشكل دمج first integrals و non-linear eigenvalue problems جوهر هذا الإطار. يتم التعامل مع قضايا الاستقرار بشكل رئيسي باستخدام خصائص M-matrices ومبدأ Lasalle. يتم أيضاً شرح العلاقة بين بروتوكولات الإجماع و potential fields واستخدامها. كما يتم تقديم أمثلة بسيطة لإظهار قابلية تطبيق الإطار المقترح في تصميم عناصر التحكم المحدودة التي تفي بأداء محدد.

كما يتم تقديم استخدام الدوال المعتمدة على الحالة للتحكم في التبديل بين السلوكيات المختلفة. يتم استخدام العديد من السلوكيات البدائية لبناء بنوك سلوكية أكثر تطوراً يتم تخزينها في كل نظام مما يسمح لكل نظام باختيار أو اتباع السلوك المختار. يتم أيضاً تصميم البروتوكولات للحفاظ على الاتصال والنتيجة هي مخطط حركة و منسق متطور.

يمكن للأنظمة قيد النظر أن تكون روبوتات بحرية أو جوية أو أرضية. يتم تطوير أنظمة المولدات الكينماتية والكينامية الديناميكية على حد سواء وتكاملها مع وحدات تحكم ذكية تفاعلية ونموذجية مبنية على نماذج تفاعلية لتسهيل التفاعل مع أماكن عمل أكثر واقعية. يتم استخدام harmonic potential fields لنمذجة البيئة للتمكين من تجنب الاصطدام . يتم الجمع بين كل من سلوكيات تجنب التصادم و المحافظة على الإتصال بطريقة بسيطة ، ولكنها فعالة. يتم تقديم النظام الديناميكي من الدرجة الثانية ، و الخطي غير الثابت ، والأنظمة غير الخطية ، والنماذج الأخرى ذات الأبعاد الأعلى ، حيث توضح منهجية التمثيل الخاصة المقدمة مدى فائدة الإطار المقترح عند التعامل مع هذه الأنظمة . يمكن ترقية خطوات التصميم المقدمة بسهولة للتعامل مع الأنظمة الديناميكية غير متجانسة السمات. يوفر الإطار الوسائل لتسهيل بناء سلوكيات قوية و محكمة يمكنها التعامل مع الاضطرابات والضوضاء والشكوك في عملية صنع القرار و الأنظمة الديناميكية المشاركة .في الحالة الأخيرة ، أصبح التكامل مع وحدة محلية ذات تحكم قوي و محكم ممكناً في إطار العمل المقترح.

CHAPTER 1

INTRODUCTION

Due to the desperate needs of human beings to make up for their lack of abilities in doing certain important tasks under some severe conditions, the usage of robots has emerged rapidly to the extent where hundreds or even thousands of these robots are being deployed to achieve the entitled missions. As these missions may vary, the types of robots used must be suitably selected to suit the prescribed goals. For example, soft-robots may serve well over the World Wide Web (www) as agents responsible for collecting or distributing information. However, physical robots, i.e., comprising mechanical and electronics parts, are most suited when physical interaction with the real world is desired. Swarms of submarines, un-manned aerial vehicles (UAVs), un-manned ground vehicles (UGVs) are some good examples of such physical robots.

Usually, the increase in the number of deployed robots comes to compensate for the inability of an individual robot to function properly by itself due to its on-board power limitations or due to its insufficiency to interact with loads because of its limited structural rigidity, for example. Inspired by nature, themes of bio-inspired robots have been brought to peoples' attentions to overcome limited robots' capacities. One inspiration was the cooperation among various creatures, ants for example, to handle a heavy load.

From engineering perspective, describing a collective activity or behavior of hundreds or even thousands of interacting systems may not be easy. However, engineers alleviated that

complexity by establishing the needed frameworks and mathematical tools through bridging various fields together. Classical control, switched-systems control, finite field, and networked control theories are some examples of what has been used to facilitate the diversity of such frameworks. Having that in mind, it will be obvious to the interested reader that dealing with multi-robots collectively is a serious challenge. As a result, covering every aspect concerning this area is far beyond the capacity of a single study.

One way of enabling the multi-agents, i.e., robots, to interact collectively is by establishing a communication network. One should not limit the definition of a network to a wire/less network where the exchange of information is conveyed by communication protocols such as ATX or ZigBee, for instance. In fact, nature tells us more about that; for example, among the fireflies, the light is a way of communication to achieve synchronization among the males to give an uncluttered view for the females to choose their mates during the season, whereas to humans reading, writing, and body gestures are more evolved means of communication. In the engineered systems, i.e., robots, lights, beeps, and even machine vision can serve as good examples of their ability to communicate properly. From the aforementioned discussion, it should be clear that the main entrance to the cooperative robots' world is through the understanding of the underlying communication networks.

Next, some preliminaries needed throughout the thesis are presented.

1.1 Algebraic Graph Theory

In many cases, it is convenient to visually represent the connections among interconnected dynamical systems, or agents, using a graph. A graph consists of a collection of labeled vertices (\mathcal{V}) that are connected via edges (\mathcal{E}). Thus, a graph (\mathcal{G}) can be given as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$.

The vertex \mathcal{V}_i represents the i^{th} agent while \mathcal{E}_{ij} represents the edge connecting both \mathcal{V}_i and \mathcal{V}_j agents. Each edge facilitates the exchange of flow back and forth the two ends. The flow could be unidirectional, i.e., directed, or bidirectional, i.e., undirected, with different magnitudes. Therefore, an edge is represented visually using single-arrow or double-arrow headed line associated explicitly with weights (w_{ij}) reflecting the magnitude of the flow, if desired. Thus, a graph can be given more specifically as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ where \mathcal{W} is the set of the corresponding edges weights in the graph. Several directed and undirected edges may join two vertices which results in a mixed multigraph as shown in Figure 1.1. The flow nature could be anything, based on the application, ranging from a conceptual effect to a real physical one. It is worth noting that the nature of the flow among agents is not necessarily like the nature of the edge through which the flow is conveyed. For example, the spatial speed of one agent may get transferred to another agent via a wireless communication and the vibration experienced by an agent may get propagated to another via a rigid rod. Consequently, it is not surprising to see a graph with real or complex numbers associated with the edges' weights. The use of complex edge weights is useful in many cases as can be found in [2], for example. Moreover, depending on the application, self-loops or buckles, i.e., $w_{ii} \neq 0$, may exist which indicates that an agent is more concerned about its own status than it does about those of other agents. More information about the graph types and their associated theories can be found in the literature, e.g. [3]. In our work, we will be using real weights of simple undirected graphs only and therefore multiple edges and loops are not allowed, unless otherwise explicitly stated.

Note 1.1: Using mainly undirected networks should not limit the applicability of the proposed framework; since we are interested in understanding how agents affect and get

affected by each other. Some examples showing directed links are also included as can be seen in formation related behaviors where a leader or multiple leaders affect the followers. This will introduce the concept of non-cooperating leader by which the leader is not affected by its followers meaning that it will not slow down if the followers are unable to follow or obey its lead, for examples. In contrary, a cooperative leader will consider the states of its followers which means that it is interacting with them over undirected links. In this work, the leaders are selected randomly, and no voting or similar algorithms are used.

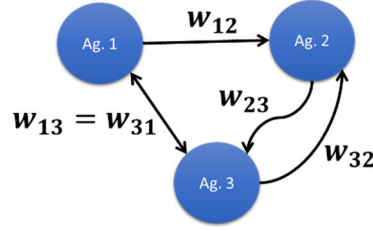


Figure 1.1. A mixed multigraph consists of three connected agents.

Mathematically, a graph \mathcal{G} can be represented using a special type of square matrices called the adjacency matrix [3] ($\mathcal{A} = [w_{ij}]$) that shows the relation among agents where the i^{th} row of the matrix reflects how the i^{th} agent is affected by its in-neighbors while the i^{th} column reflects the effect of the i^{th} agent on its out-neighbors. The interaction among agents is governed by certain rules denoted by the used network protocols. These protocols should not be confused with the communication network protocols usually used in the handshaking of data over a wireless communication link, for instance.

Based on the network protocol used - as will be explained in this thesis-, the analysis of the response, i.e., behavior, of the interconnected agents will be different. The diffusive-coupling network protocol is widely used in the literature (see e.g. [4], [5]) and it deals with the relative differences between the connected agents states over the edges of their associated graph. Using this protocol, another important diagonal matrix known as the in-

degree matrix [3] (\mathcal{D}) is obtained whose i^{th} diagonal element represent the summation of the weighted flows entering the i^{th} agent. The mathematics for this is delayed to the next section. Both \mathcal{A} and \mathcal{D} are used to formulate- as will be shown shortly- a special matrix known as the Laplacian matrix, i.e., $\mathcal{L}_f(\mathcal{G})$, associated with the graph \mathcal{G} whose spectral properties are of significant value in the design and analysis of network protocols, in general. The subscript in \mathcal{L}_f denotes that all w_{ij} in \mathcal{G} are fixed constants.

Note 1.2: It is desired in this thesis to allow for variations in the edges' weights to accommodate variations in context. Note that fixed weights do not allow for such flexibility.

In many applications, the interaction among the connected agents may vary explicitly over time or based on their status and operating conditions. Thus, dealing with varying edge weights is inevitable. In this thesis, mainly state-dependent variable weights will be used as will be explained in the coming section while the case of both explicit time and state-dependent weights will be investigated in a future work.

1.2 Non-negative Matrices

The non-negative matrices, known for short as M -matrices, were used extensively in studying the stability of systems appearing in various fields like in polytopic systems [6], genetic regulatory networks [7]-[8] and most importantly- in our case- the cooperative control of multi-agent systems [9]. In control theory, both M -matrices and Hurwitz matrices relate to each other when studying the stability of feedback systems though the analysis using each type of these matrices is different. An M -matrix may be singular or

nonsingular. Several characterizations of nonsingular M -matrices are available in the literature, such in [10], as well as singular ones, such in [11]. In the proposed framework, herein, it is desired to use the theory of singular M -matrices to formulate the stability of Multi-Agent Systems (MAS) under general network consensus protocols in a unified way. A unified framework to study the stability problem of composite systems using nonsingular M -matrices was presented in [12]. The main difference between the aforementioned frameworks is that in the one presented herein the M -matrix is directly related to the Laplacian matrix representing the underlying communication network connecting agents while the M -matrix presented in [12]- or similar approaches- is related to the coupling weights and it is generally nonsingular.

Several classes of singular M -matrices in $\mathfrak{R}^{N \times N}$ based on their stability types were explained in [11] like being an *s-semistable* (denoted by M), *D-semistable* (denoted by K), *diagonally semistable* (denoted by J), *weak semistable* (denoted by W) and *semistable* (denoted by L) matrices. The following theorem states the relations among the previous classes:

Theorem 1.1: [11] In $\mathfrak{R}^{N \times N}$

$$J \subseteq K \subseteq L \subseteq M \subseteq W$$

while

$$J \cap Z^{N \times N} \subseteq K \cap Z^{N \times N} = L \cap Z^{N \times N} = M \cap Z^{N \times N} = W \cap Z^{N \times N} = P$$

where the inclusion of $J \cap Z^{N \times N}$ in P is strict for $N \geq 2$.

The $Z^{N \times N}$ is a subset of $\mathfrak{R}^{N \times N}$ consisting of the matrices with non-positive off diagonal entries. Thus, the set of all M -matrices is strictly contained in $Z^{N \times N}$ [11]. The matrices denoted by P are a special class of singular M -matrices that shares several important characteristics enjoyed by the nonsingular M -matrices. Such a class is called M -matrices with “*property c*” where the following theorem states their properties.

Theorem 1.2: [13] Let $A \in \mathfrak{R}^{N \times N}$ be a singular, irreducible M -matrix of order N . Then:

1. A has rank $(N - 1)$.
2. There exists a vector $x \gg 0$ such that $Ax = 0$.
3. A has “*property c*”.
4. Each principal submatrix of A other than A itself is a nonsingular M -matrix.
5. A is *almost monotone*, i.e., $Ax \geq 0 \Rightarrow Ax = 0$.

Now, if the Jacobian of the MAS dynamics is acting as a point-wise singular M -matrix in the desired manifold of the state space- as per the proposed framework, it is desirable to investigate which type of stability it exhibits. For example, in [9] and [14], the diagonally semistable property, i.e., J , was used to prove the stability of the involved systems incorporating M -matrices. This property is defined as follows:

Definition 1.1: [11] A matrix $A \in \mathfrak{R}^{N \times N}$ is diagonally semistable, i.e., $A \in J$, if there exists a positive diagonal matrix D such that $AD + DA^T \geq 0$.

A good candidate for establishing the D matrix is to use the left eigenvector elements such that $D = \text{diag}([v_1, \dots, v_N])$. Note that the left and right eigenvectors of a balanced digraph and undirected graphs are the all-one vector, i.e., $\vec{1}$. However, other types of M -matrices stability might also be needed, like the D -semistability, which is defined as follows:

Definition 1.2: [11] A matrix $A \in \mathbb{R}^{N \times N}$ is *D-semistable*, i.e., $A \in K$, if for every positive diagonal matrix D , the matrix AD is positive semidefinite.

If the diagonal matrix D happens to be non-positive, then the MAS may become unstable. Some other issues related to *D-stability* can be found in [15].

1.3 Thesis Motivation

The rapid spread of the multi-agent systems has addressed several challenges on the last decades. Among those important challenges are the agents' cooperation and coordination. The importance of cooperation and coordination may be clearly seen in Search and Rescue Missions (SRM). Figure 1.2 shows a proposed SRM in a devastating environment. Unlike most of the available literature, this dissertation proposes a new perspective of agents' cooperation and coordination. In this perspective, mainly three- may be more- types of agents are assigned different, yet interacting, tasks that they must accomplish. The communication agent (CA) is responsible for maintaining the connectivity of the other types despite their spatial distribution. While a scout agent (SA) is responsible for discovering and mapping the environment which is most likely believed unknown. With the aid of CAs, SAs can pinpoint valuable information about the environment and survivors. The pinpointed targets can then be communicated back to the user who is responsible for dispatching the active agents. Active agents (AA) are a special type of agents under the proposed SRM perspective; because they are equipped with special tools and equipment that enable them to directly interact with the targets.

As in most unknown environments, a devastating one can have static or moving obstacles. In the related literature, this is equivalent to structured and unstructured environments, respectively.

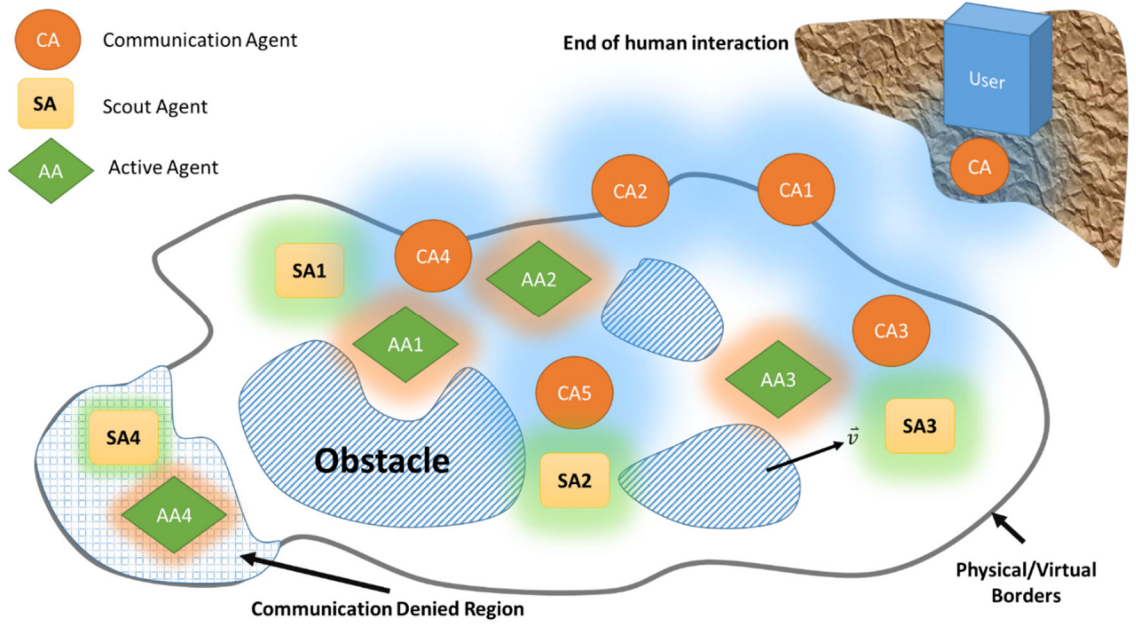


Figure 1.2. Search and Rescue Team in an unknown devastating environment.

These agents are organized in several hierarchical structures with different mathematical complexities. The first structure contains different levels or layers as shown in Figure 1.3.

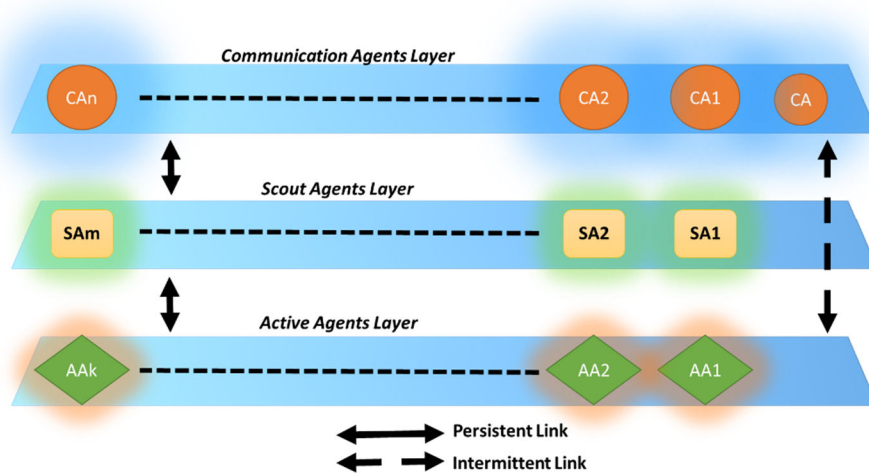


Figure 1.3. First hierarchical structure of agents according to the proposed SRM perspective.

The first structure assumes different numbers of CAs, SAs and AAs. This structure introduces couplings among the dynamic motion of all agents especially after the AAs are deployed by the user. However, this structure offers an increased degree of freedom since all agents can move freely when compared to the second structure discussed next.

A cluster-based hierarchical structure is shown in Figure 1.4 where agents are placed into clusters such that:

- 1- A cluster must contain at least one CA.
- 2- A cluster may contain a SA without AAs.
- 3- A cluster may contain AAs if and only if it contains a SA.
- 4- A cluster may evolve dynamically according to the context.
- 5- In a cluster, the SA is considered as the leader.
- 6- A cluster may allow AAs without a leader to pass through it temporarily.

Remark 1.1: These criteria are subjected to future modifications once the problem is thoroughly addressed since new requirements could take place.

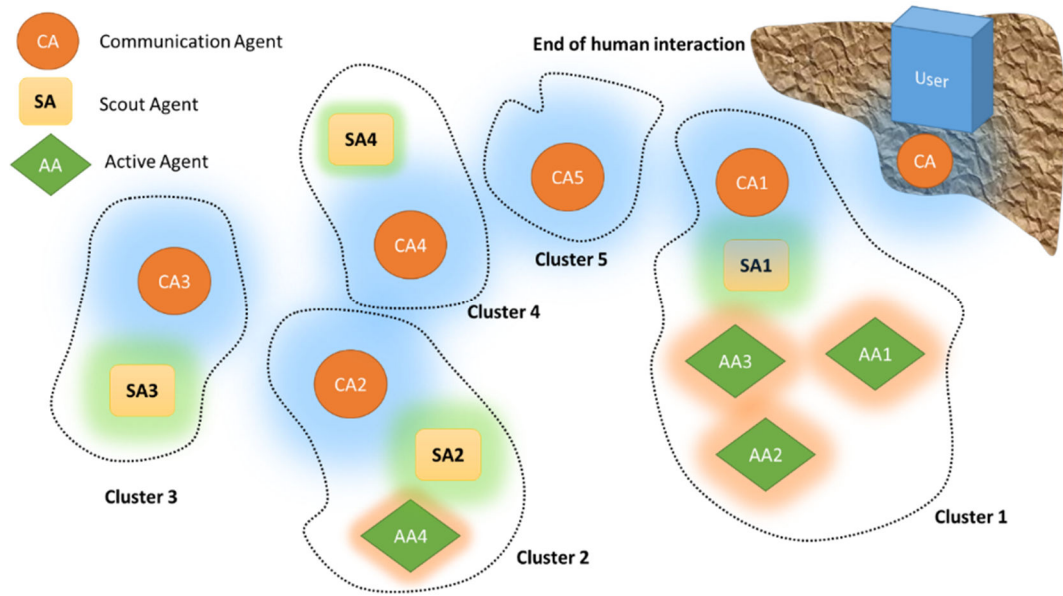


Figure 1.4. Second hierarchical structure of agents according to the proposed SRM perspective.

According to the second structure, the coupling is only introduced at the CAs level. As a result, a reduced degree of freedom is offered by the second structure when compared to the first one.

Another structure is shown in Figure 1.5. This structure assumes dealing with an equal number of SAs and AAs with- may be- a different number of CAs. If the CAs layer is omitted, then this structure will have the same structure as ordinary node-to-node consensus problem formulation if both the SAs and AAs are having the same network structure, see [8] for example.

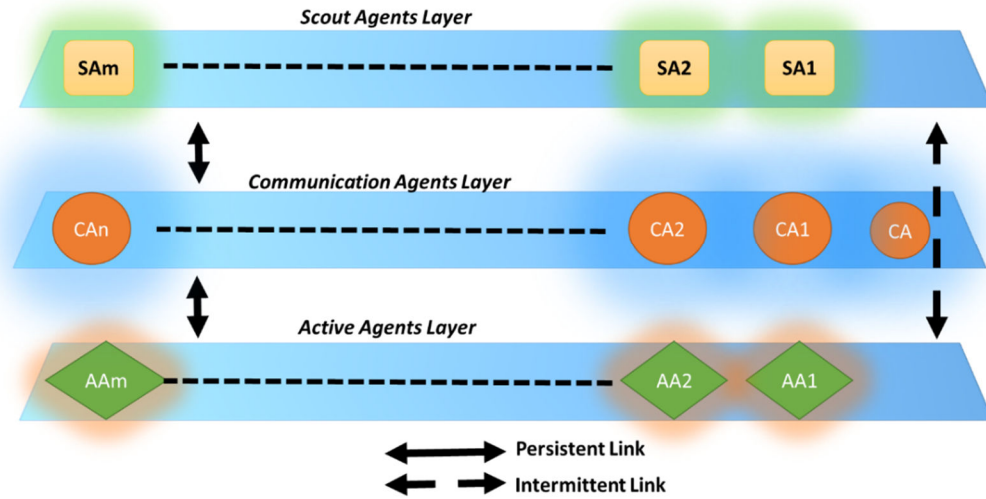


Figure 1.5. Third hierarchical structure of agents according to the proposed SRM perspective.

In the following subsection, various types of controllers that will be utilized mainly in Chapter 5 are presented. Using these controllers, an agent is most likely to situate itself within its surrounding under different working conditions.

1.4 Context-aware Intelligent Controllers

Agents navigating through unknown or dynamic environments require fast reflexes. As in the human body, reflexes take place in several situations almost daily. These reflexes help

in avoiding dangers and responding quickly to threats *unconsciously*. To mimic such a precious aspect of the human body, a Reactive Controller (RC) can be hard-wired into the lowest-level part of the agent, i.e., the actuators.

Figure 1.6 shows the general structure of a RC in which sensors feed their readings to, usually, static activation functions to generate the most appropriate actions or reflexes.

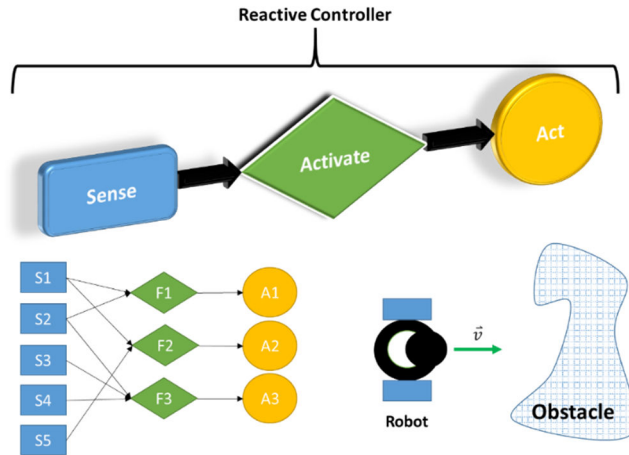


Figure 1.6. General structure of a Reactive Controller.

Imagine a robot in the x - y plane approaching an obstacle. It is intuitive that the robot should turn away from it. This reflex can be implemented simply using a RC. The RC can fit in the control loop as shown in Figure 1.7. Under the shown integration between the robot controller and RC, the latter will be acting as a disturbance to the robot. So, it is essential not to sacrifice the stability of the closed loop system. The stability issue under the integration of RC or behavior control in general motivates another objective of this thesis. Figure 1.7 shows the integration between the robot local and reactive controllers.

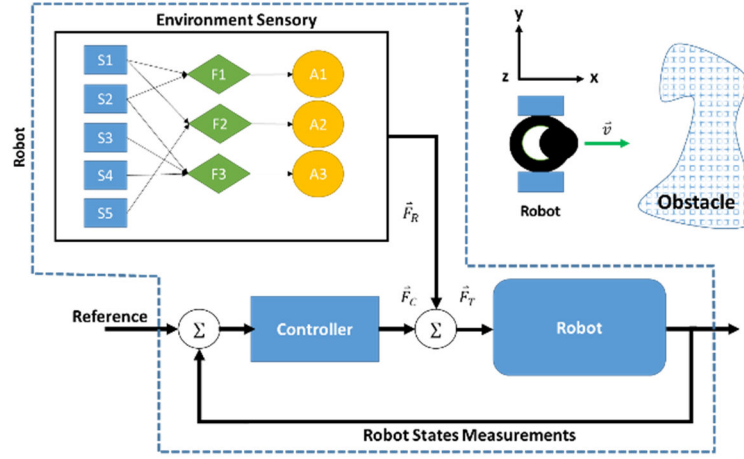


Figure 1.7. Integration between the robot local and reactive controllers.

Note 1.3: The RC is acting as a feedforward controller to avoid disturbances (obstacles).

The mathematical representation of the reactive control signal acting as a disturbance can be obtained, for example, as a function of sensory data approximating the distance between the robot and the obstacle in a primary direction, say x -axis. Figure 1.8 shows the graphical representation of the reactive force used in the RC which simply states robots should move away from the obstacle.

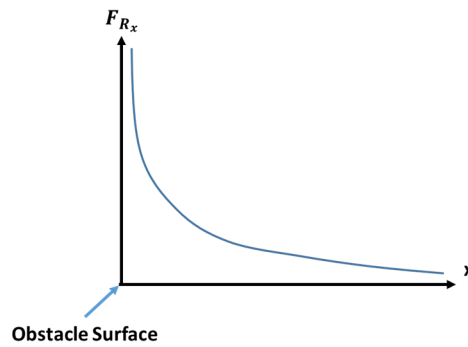


Figure 1.8. Graphical representation of the reactive force used in the reactive controller.

In more complex environments, a richer set of behaviors should be involved to bring the robot closer to context-aware realm. Model-based Motor Schema (MMS) intelligent

controller, shown in Figure 1.9, maps the context obtained using the *Know* step to a set of possible behaviors. Only the allowed behaviors under the current context can be triggered and then the results are passed to the *Plan* step. In this thesis, we are only interested in studying specifically the motion-related behaviors. Some of the covered behaviors are shown in Figure 1.10 where a selection mechanism chooses the most suitable behavior that is going to dominate a given context.

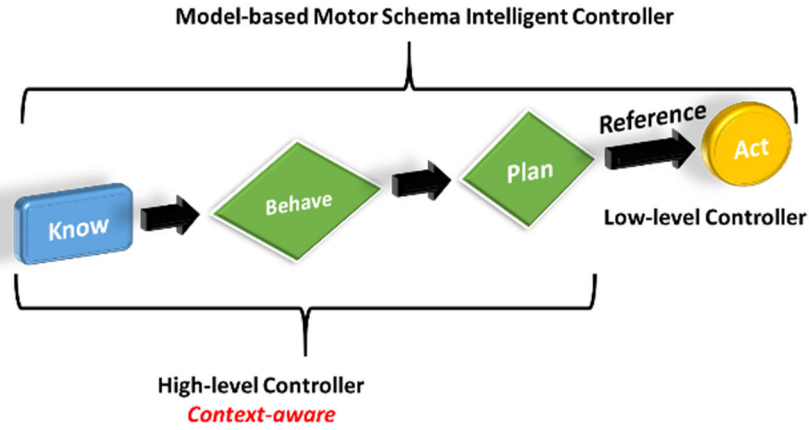


Figure 1.9. Structure of Model-based Motor Schema intelligent controller.

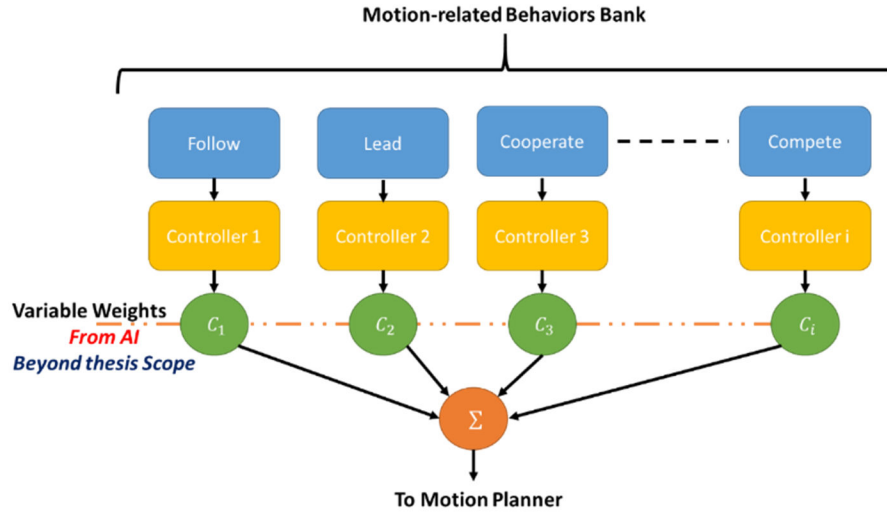


Figure 1.10. Some of motion-related behaviors under MMS intelligent controller.

As example, consider the case where a robot is intended to behave in a way that can be described verbally as follows:

- Move only along the positive x -axis.
- Maintain the distance from the y -axis as small as possible, i.e. $y=0$.

As suggested in Figure 1.11, the needed forces to be applied are shown. The overall force field will exert a behavior command based on the combination of the two primitive actions. The combination of the primitive actions or behaviors is what gives the strength to the MMS intelligent controller structure. Mathematically, we may write:

$$F_x(t) = \text{constant} = a = \dot{x}(t) \quad (1.1)$$

$$F_y(t) = -y(t) = \dot{y}(t) \quad (1.2)$$

where: F_x and F_y are forces resulting as gradients of suitable potential fields. Both x and y denote the states of the trajectory generated using the planner under MMS structure. The planner equation is given as follows:

$$\ddot{\mathbf{X}}(t) = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} a \\ -y \end{bmatrix} \quad (1.3)$$

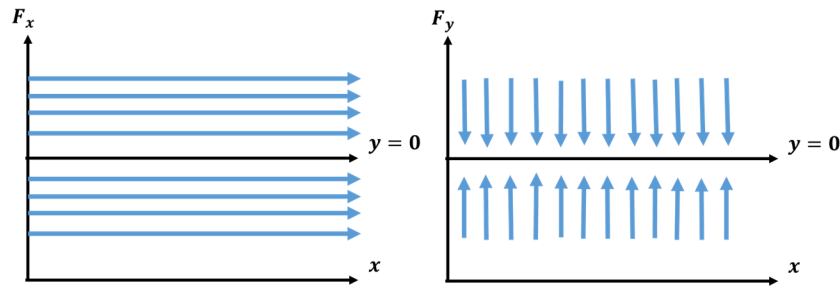


Figure 1.11. Desired force fields along the x and y axes.

Now, the resulting trajectory, i.e., the reference signal to the low-level controller, is shown in Figure 1.12. More details will be provided in Chapter 5.

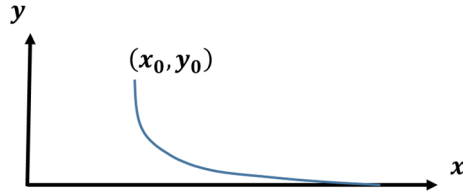


Figure 1.12. Resulting reference signal generated by the motion planner under MMS intelligent controller.

To combine the benefits of both RC and MMS intelligent controllers, the Hybrid Model-based Reactive (HMR) controller is presented as shown in Figure 1.13. The implementation of HMR controller is shown in Figure 1.14 using one preferred configuration.

Under HMR, the known part of the robot context is handled by the MMS while the unknown part of the robot context is handled by the RC. In case of unforeseen situation, the RC will directly affect the robot behavior to keep it safe. This intervention from the RC will result in a change of the overall robot context that is most likely sensed by the *know* part inside the MMS controller when time advances.

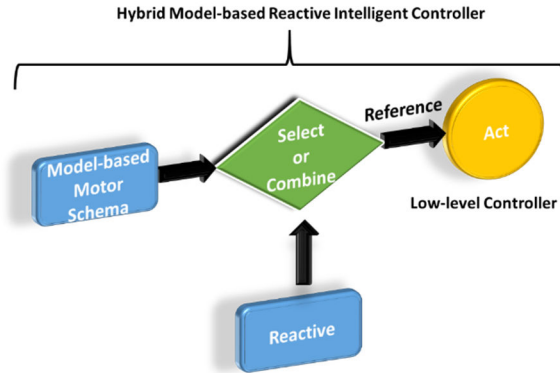


Figure 1.13. Hybrid schema integrating both the RC and the MMS intelligent controllers.

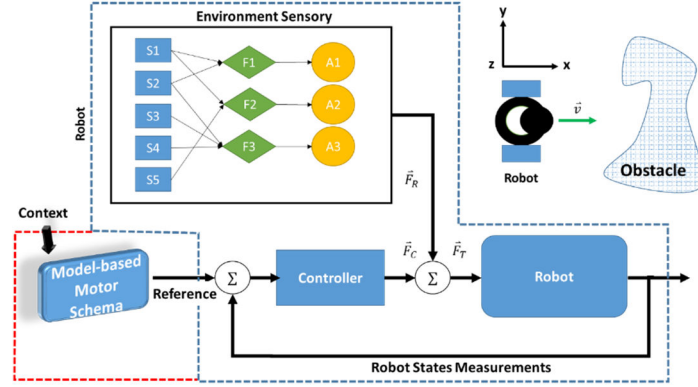


Figure 1.14. The implementation of HMR controller using one preferred configuration.

Now, let us assume that a multi-agent system of a group of robots as shown in Figure 1.15 is used. The main purpose of this gathering is to behave cooperatively. If the working space is obstacle-free- as shown in Figure 1.15, then this will be a typical problem in the related literature. However, if the previous multi-agent system exists in a working space filled with obstacles- as shown in Figure 1.16-, then the problem becomes more involved; since both cooperative and obstacle-avoidance behaviors must co-exist together. So, it is desirable to investigate the collective behavior of agents when operating in a real situation under behavioral control.

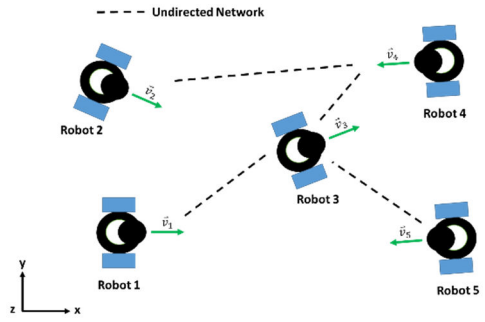


Figure 1.15. Multi-agent system in obstacle-free context.

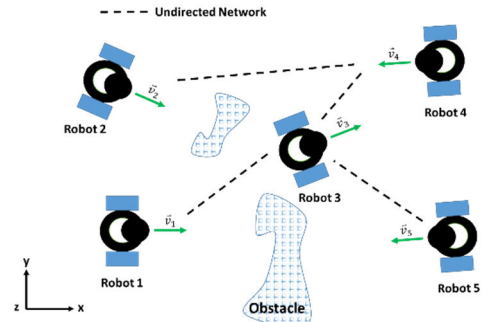


Figure 1.16. Multi-agent system in obstacle context.

Figure 1.17 shows the relation between the distributed control law and a communication channel based on ZigBee protocol. In this thesis, we are targeting the application layer of the used communication protocols.

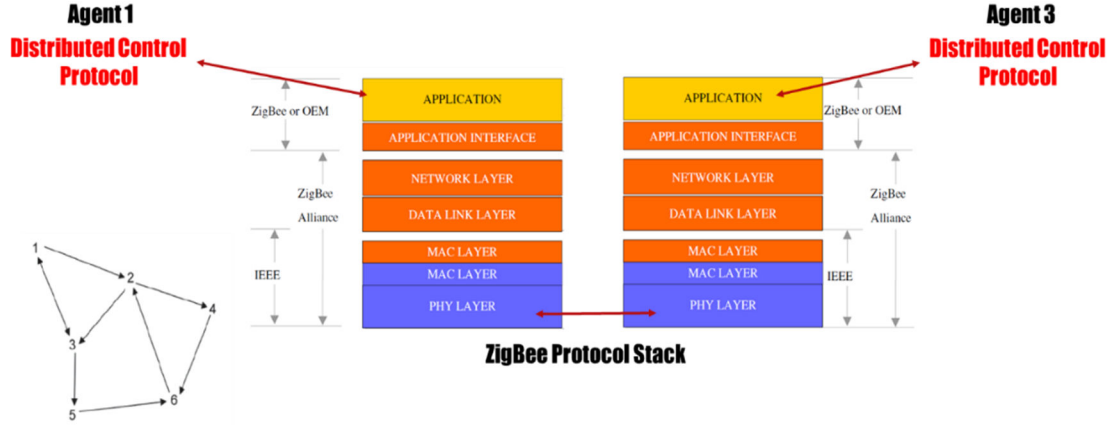


Figure 1.17. Relation between distributed control and communication protocols.

1.5 Thesis Layout

The presentation of this thesis is organized as follows: In Chapter 2, a thorough literature review to cover the needed topics of this thesis are presented. In Chapter 3, the state-dependent semi-linear protocols are introduced to reflect upon the similarities between the notion of state-dependent and constant Laplacian matrices. Then, it will be shown how the state-dependent Laplacian matrix can be modified to achieve a special type of consensus, namely: cluster consensus. After that, various types of consensus available in the literature are upgraded to be state-dependent. Having some of the main results shown, the proposed framework will be presented where the interplay of its pillars is detailed. Next, the applicability of the framework mainly for nonlinear protocols is demonstrated where several protocols appeared in the literature were verified using the framework. Issues related to stability and performance are then handled followed lastly with revealing the relation between consensus protocols and potential fields where it can be used to design and analyze the desired protocols.

In Chapter 4, the applicability of the framework in producing several motion-related behaviors that govern the interaction among the connected agents is demonstrated through

mainly simulation examples where proofs of invariance, stability and convergence are presented when needed. Using state-dependent parameterization to control the switching between the various behaviors is presented. The multitude of primitive behaviors are used to build more sophisticated behavioral banks that reside in each agent permitting each agent to choose or follow this behavior. Designing connectivity-preserving protocols is also addressed. The result is a sophisticated distributed coordination motion planner.

In Chapter 5, more sophisticated behaviors that are mainly directed to mobile agents are provided. These agents could be maritime, airborne or ground robots. Both kinematical and kino-dynamical trajectory-generator systems are developed and integrated with reactive and hybrid-model-based-reactive intelligent controllers to facilitate interacting with more realistic working spaces. Harmonic potential fields are utilized to model the environment such that collision-avoidance is made possible. Both collision-avoidance and connectivity-preserving behaviors are combined in a simple, yet, efficient way. Also, second-order, general linear time-invariant, nonlinear systems, and other models with higher dimensions are presented, where a special representation methodology reveals the usefulness of the proposed framework when dealing with such systems. The design steps presented can be easily upgraded to deal with systems of heterogeneous dynamical features. Using simulation, the strength of the kinematical trajectory-generator is demonstrated by guiding a group of non-holonomic front-wheel steered robots through a working space where obstacles exist. Another example demonstrating the use of the developed kino-dynamical trajectory-generator systems in controlling the behavior of a MAS consisting of six quadrotors is presented. Lastly, In Chapter 6, some of the recommended future work is listed.

The main contributions of this thesis are as follows:

- 1- Presenting a unifying framework under which state-dependent semi-linear and nonlinear consensus protocols can be designed in a systematic manner.
- 2- Providing a clear link between the fixed-weight and state-dependent Laplacian matrices through adopting nonlinear eigenvalue problem under which the fixed-weight protocols proof to be a special case of the proposed framework.
- 3- Focusing mainly on MAS consisting of N single-integrator dynamical agents interacting over undirected graphs.
- 4- Handling the stability, convergence and performance issues using singular M -matrices, Lasalle's principle and vector calculus-related theories.
- 5- Demonstrating the strength of the proposed framework through the possible achievable types of consensus and through the performance metrics applicable.
- 6- Formulating the problem of consensus protocol design as a general system of partial differential equations which relates generally the consensus protocols to potential fields.
- 7- Extending and analyzing some of the related consensus protocols using the proposed framework.
- 8- Designing connectivity-preserving protocols under the proposed framework.
- 9- Designing sophisticated kinematical and kino-dynamical trajectory-generator systems that can provide the guidance to agents while conducting a specific mission in a working space filled with obstacles.

CHAPTER 2

LITERATURE REVIEW

In cooperative systems, knowing the number of the interconnected agents (nodes) on a graph is very important in almost all the distributed control and sensing protocols. In fact, this number can be estimated using dedicated communication protocols, statistical inferences or through system-identification based methods. This number may vary over time, which indicates the state of having new agents joining or leaving the network. Such variation is typical in Mobile Ad hoc networks (MANET) in which even the agents' types may vary significantly resulting in a heterogeneous network structure. Any provided solution that depends on communication to identify the number of agents on a graph will consume about 15 to 35% of the energy available in each agent [16].

A distributed mobile agent based topology discovery framework is presented in [17]. Usually, a centralized approach with a predetermined range of IP addresses that are used to ping the agents on a regular basis is used. This centralized approach is impractical from MANET point of view; since it requires the knowledge of the IP addresses of all agents in advance which may violate the privacy of the interconnected agents. Besides the previous points, such centralized approach uses the available power resources at each mobile agent intensively resulting in a shorter life of the agents and consequently that of the network connecting them. Topology discovery can be done using different approach such as power control approaches which adjusts the power at each node to keep network connectivity or cluster-based approach which emphasizes a certain hierarchical topology organization

where each cluster is managed by an elected head based on different factors such as its communication quality and the available power [16]. Most of these topology discovery approaches are mainly ad-hoc network (ANT) based approaches in which the mobile agents collect and update the information about the visited nodes by traveling from node to node [17]. The ANT based approaches are purely proactive and uses huge bandwidth even if there are no changes occurring in the network. Besides that, the agents under such approaches may get lost once the host node is disconnected. However, the approach presented in [17] ensures that such problems do not exist and it also does not require a predefined set of IPs to be available a priori; because it depends solely on the beacons sent by the nodes over the network and it is also reactive since it sends topology updates when there is a change in the network. This approach divides the network nodes logically into two hierarchical groups, namely cluster heads containing the manager agents and cluster members containing the service agents. Each service agent in a cluster reports to its manager agent in the same cluster. Managers at each cluster got elected based on their ability to communicate with other cluster heads as well as other factors such as their available power. This results in a more efficient and stable topology discovery approach. More details about the system design can be found in [17].

Estimating the network size in a distributed fashion is important in realizing generally all distributed protocols. As it will be clear from the coming chapter, starting with a complete knowledge of the network topology and agents number is a pre-assumption that must be satisfied. Of course, dedicated protocols can be run in the preparation stages of the MAS creation. In [18], a distributed network size estimation that preserves privacy based on system-identification is presented. This approach depends on the locally available

information at the agent level and requires no leader or centralized agent to implement its functionality. Privacy-preserving algorithms are very important for anonymous networks where the connected agents are unable or not willing to provide a unique ID labels telling their identities. Moreover, providing an algorithm that can be suitably used when the agents are with limited resources, and with almost no knowledge of the network structure is very important. Basically the approach presented in [18] requires all agents to run a suitable linear dynamical system and then identify the order of an associated *Hankel matrix* (H) which can be considered as the minimum possible number of connected agents. This lower bound will converge to the exact number of agents when the communication network is strongly connected and when each agent has a self-loop, and the algorithm is implemented using real-valued numbers and when the size of the network is small, i.e. less than few tens. The used dynamical system is shown next:

$$x(k + 1) = Ax(k) \tag{2.1}$$

Where: A is a matrix describing the graph, i.e. $A_{uv} = 0$ if (v, u) is not an edge in it. In real-valued implementation of this approach, the identification of a large size system causes numerical issues resulting from the stability or instability of the A matrix; since the entries of $x(k)$ will become smaller as time passes if A is stable and get larger when A is unstable. Even if A was chosen to be marginally stable, it will not help; since the entries of $x(k)$ will remain in the convex hull of the initial conditions. However, this problem can be solved by implementing this approach using finite field representation of numbers which will then provide a lower bound of the agents' number with a probability less than one given by:

$$\text{prob}(\text{rank } H_n^{(r)} = n) \geq 1 - \frac{n^2}{q} \quad (2.2)$$

as estimated in the r^{th} agent, where: q is the size of the finite field used, i.e. \mathcal{F}_q , and $q \geq n^2$.

Moreover, defining $d = 0.5(n^3 + n)$, if $q \geq d$, then:

$$\text{prob}(\text{rank } H_n^{(r)} = n) \geq 1 - \frac{d}{q}, \quad \forall r = 1, \dots, n \quad (2.3)$$

With a detailed proof available in [19] inspired by [20].

However, this bound requires a field size that is very large when n is medium-size, e.g. when n is a few hundreds, which consequently demands a huge amount of memory quadratically increasing with n . As a result, a slightly modified version of Hankel-Lanczos factorization algorithm [18] was presented which ensures the reusability of the evaluated H matrix in previous steps. This gives the possibility of only using a memory size which is linear in n instead of being quadratic. Simulation results indicated that the algorithm presented in [21] which is based on statistical inference and on max-consensus has a comparable memory requirement as the algorithm provided by [18] and much higher transmission complexity (n times larger). Also, the algorithm presented in [21] has a performance much worse in terms of probability of exact computation of n , slightly better in terms of average n , and much better in terms of average quadratic error once compared to the algorithm presented in [18].

The method proposed in [22] requires two key assumptions to effectively estimate the global network topology in a distributed manner, namely: the network must be slowly varying, and the estimating agent can measure or estimate the states of all its neighbors,

otherwise the global network topology cannot be estimated using only one estimating agent. Mainly, three system-identification approaches were investigated in [22], namely: concurrent, traditional gradient descent method and least squares based approaches. It was noted that regardless of what the identification method used to solve the network discovery problem, there must exist some amount of excitation to ensure convergence to the right solution.

Remark 2.1: In general, persistent excitation is essential in improving the identifiability and convergence of the involved parameters estimates.

After identifying the network topology along with the number of agents sharing it- if they are not already known- one can proceed to have a look at the overall properties associated with the multi-agent system on graph in terms of controllability and observability. *Structural Controllability* and *Structural Observability* of the multi-agent system are used to distinguish them from the ordinary structural controllability and observability at the level of each agents. In the following context, it is desirable to reveal the relationship between these terms and the network topology used and to provide a way by which they can be obtained mathematically using a graph-theoretic perspective under several conditions.

In [23], the problem of weight assignment for point mass agents comprising a single leader and multi followers under weighted undirected and connected graphs was discussed and formulated using the optimal control theory. The weights are assumed to be time-varying and freely assigned as well. The main objective was to minimize the control effort given to the whole system, so that the provided index function was solved with the aid of Hamilton-Jacobi-Bellman (HJB) equation. However, this approach depends only on the

leader to solve the optimal problem and to provide the optimal control and weights to the followers, so the global optimal problem is not solved in a distributed fashion among the agents. Therefore, a delay in the response is most likely to take place.

Now, let us discuss in more details the structural properties of dynamical multi-agent systems on graphs. In [24], the structural controllability of multiple leader multiple followers on a weighted and directed graph was investigated and its conditions are given algebraically and from graph-theoretic perspective. In [25], the relaxed equitable partition was used to investigate structural controllability under un-weighted graphs where the controllable subspace is based on characteristic vector when the system is not completely controllable was identified. However, this method cannot be used with weighted graphs [24]. It was shown in [24] that the system is not completely controllable if the leaders set is not globally reachable from the followers. According to [26], an algebraic characterization of structural controllability is to have none of the left eigenvectors of the adjacency matrix orthogonal to any column of the leader-follower pinning matrix. From a graph-theoretic perspective, it was shown that the system is structurally uncontrollable if it is symmetric with respect to leaders and the controllable subspace [24], i.e. leader symmetry is a sufficient but not necessary condition for system uncontrollability except for a Path graph it is necessary and sufficient condition for uncontrollability [27]. In such cases, the controllability may be restored either by weight assignment of the communication links or by adding more agents if a special condition is satisfied [24].

In [28], the structural controllability of a single leader multi-follower systems under absolute and relative protocols was investigated and some states could have different network topology which can be used to model more general classes of complex systems in

practical applications. The effect of the selected leaders on the overall structural controllability was examined in [29] via a constructive design procedure for an uncontrollable multi-agent system with a given communication graph. Also, in [29], a design procedure under which new communication graphs can be constructed by adding or deleting new nodes and new edges between follower and leader nodes, respectively, was proposed. Such approaches are very important to be used especially when the networks are dynamically created. Considering the previous discussion, it is crucial to study both the structural controllability and observability under state-dependent protocols which is a recommended future of this thesis.

Another approach to extend the concept of structural controllability to the multi-leader configuration was investigated in [27] where network equitable partitions were used. It was also shown, in the case of single leader, if the followers Laplacian matrix associated with the graph of the follower nodes does not have distinct eigenvalues or at least one of the eigenvectors is orthogonal to $\vec{1}$ then the overall system is uncontrollable. In such a case, one can tell that there exists an eigenvector of the leader-follower Laplacian matrix that has a zero entry on the index that corresponds to the leader. Hence; if none of those eigenvectors has a zero component, then the leader-follower system is controllable for any choice of the leader. This result is very important when it comes to select the leader for the system and it will be interesting if such a property of the graph associated with eigenvectors can be investigated in a distributed manner which will give a basis for leader nomination among the agents. Moreover, having more connections among agents may increase the chances of having a symmetric graph with respect to the leader, like the case in a complete graph. However, sometimes this problem can be avoided by keeping the links on the

longest path between a leader and other nodes and deleting the unnecessary communication links which will result in breaking its inherited symmetry. Considering these results, having only one leader in a Ring graph will make it uncontrollable. In a Path graph- whose vertices can be listed in the order (v_1, v_2, \dots, v_N) such that the edges are $\{v_i, v_{i+1}\}$ where $i = 1, 2, \dots, N - 1$ -, the system is always controllable for all choices of the leader if and only if it is of an even order.

Using sampling is widely used when dealing with MAS. As a result, investigating the effect of sampling on mainly the structural controllability is of great value. It was shown in [30] that the structural controllability of multi-agent systems under weighted and undirected graphs is preserved if the corresponding un-sampled system is controllable and the sampling time T is not pathological, i.e. satisfies the following condition:

$$\lambda_j - \mu_k \neq \frac{2k\pi}{T}j, k = \pm 1, \pm 2, \dots \quad (2.5)$$

For every two eigenvalues λ_j, μ_k of the system dynamical matrix.

Assuming all agents are clock-synchronized, and a zero-order-hold is used in the actuation of agents, it can be shown that sampling does not affect the controllability of multi-agent systems with single-integrator, generic systems with scalar state, and generic systems with vector states, i.e. the controllability is invariant under these cases [30].

The structural controllability is needed to build a feasible distributed control protocol that governs the collective behavior of all nodes on the graph. To build this protocol, usually states feedback of the agents is distributed over the network to the neighboring agents. This exchange of information may result in increased size of information needed to implement this protocol which demands an increase in the communication bandwidth which will be

eventually reflected as a decrease in the service life of individual agents. So, there is a need to reduce the size of exchanged information without affecting the used sensing/controlling protocols. Having identical dynamical agents on the graph, i.e. having a homogeneous multi-agent system on a graph, helps in reducing the communication links bandwidth by estimating the needed information to build a distributed sensing/controlling protocols among the agents. One main requirement to implement such solution is to have the interconnected system observable.

A different perspective of multi-agent systems consensus is presented in [31]. According to this perspective, two levels of agents are involved, namely: leaders' and followers' levels. Unlike traditional consensus distributed laws, the law presented in [31] seeks consensus between the agents from the followers level with their corresponding leader from the leaders level. More specifically, it is assumed that the number of leaders is equal to the number of followers. So that each follower is entitled to have consensus with its corresponding leader. Also, the networks in the two levels are assumed identical.

The challenging point in this perspective is that some of the agents, at a time, can have direct connections with their corresponding leaders. So that a suitable choice of those followers can solve the problem. Leaders are only affected by their neighbors in the leaders' level while followers are affected by their neighbors and their corresponding leaders. Necessary and sufficient conditions are also presented.

The consensus problem among several first-order dynamical systems is considered in [32] where agents can cooperate or compete among each other. These behaviors, i.e., competing or cooperating, can be modeled mathematically by positive or negative weights, respectively. Based on the assigned weights, the whole network is sub-divided into mainly

two parts, namely: cooperation and competition sub-networks. By introducing a delayed control input to the competing agents, consensus can be achieved among the two sub-networks. This paper emphasizes the fact that a delay can be sometimes helpful when dealing with multi-agent systems. As a continuation of [31], [33] proved that each follower can get into consensus with its corresponding leader if this follower is influenced either directly or indirectly by a leader from the leaders level even under a switching leader-follower pinning gains. As in [33], the same authors are interested in node-to-node consensus among leaders and their corresponding followers as found in [34]. However, in this paper it is required for a follower to directly sense its corresponding leader to achieve node-to-node consensus.

In [35], the tracking consensus in the mean-square sense under node-to-node paradigm is considered. The interacting agents are nonlinear with non-identical dynamics. Due to the uncertainties that may occur in the sampling devices, the node-to-node consensus is studied under stochastic sampling. Input-delay method is useful in designing the distributed law. Figure 1.5 can be used to visualize the concept of node-to-node consensus when the communication agents' layer is removed. In fact, the node-to-node consensus has interesting applications as can be found in the case where troopers are accompanied with supporting robots.

Sampled data can be utilized to reduce the communication load over the network while achieving node-to-node consensus as can be found in [36]. An algorithm was designed to estimate the maximum value of the sampling period that can be used in the node-to-node consensus distributed law. Note that if such period is violated, then the whole MAS system comprising the two layers will suffer from instability.

Another way to reduce the communication load is achieved using the event-triggered controller approach. In [37], controller's signals are updated only when a triggering condition is fired. This approach can also save a considerable amount of energy when agents are equipped with limited resources. When communication range is limited, agents must stay close enough to cooperate. Additionally, the rendezvous problem is solved in this paper by designing a controller with bounded signals that ensures the connections among the agents to be preserved. Zeno phenomenon is also excluded under this controller so that the control signal will not experience an infinite number of jumps in any finite amount of time. Avoiding this phenomenon is important to maintain stability.

Unlike many distributed event-triggered control systems, [38] presents a self-triggered controller by which it is possible to avoid monitoring the measurements errors continuously. The self-triggered controller depends on the information available locally to determine the next instant to retrigger. Self-triggered controller can also reduce the communication bandwidth used.

Alternatively, reducing the amount of data shared reduces the needed communication bandwidth. In many cases, this can be achieved by relying only on the outputs of the interacting agents instead of using their full states information. Suitable state-observers can also be used. In [39], global robust regulation based on the outputs of a group of nonlinear systems, namely: Lorenz systems, is presented. The main idea behind this work is to use the internal model approach to formulate a global and robust stabilization of agents under switched networks and distributed output feedback controller.

In [40], a dynamic event-triggered control approach is introduced. This approach introduces a dynamic variable that get affected not only by recent agents' errors but also

by those of the previous ones. In contrast to the classical event-triggered controllers, this dynamical event-triggered controller considers the evolution of the errors into its triggering mechanism. Unfortunately, it is centralized and not distributed.

Based on discretized version of general linear and continuous-time dynamical systems, [41] presents a periodic event-triggered consensus controller. The only restriction on the used discretization period is not to be pathological relative to the continuous-time system matrix. This means that the discretization period must be chosen such that the local controllability of the agent is preserved. However, its effect on the overall multi-agent system is not investigated.

Solving the conflicts among agents when shared resources are acquired simultaneously is a must. In [42], a cooperative and decentralized approach that ensures balanced utilization of resources through advanced decision making was presented. This approach is very helpful in establishing the relation between multi-agent behaviors and their mathematical representations. The developed approach was used to solve the conflict between four agents sharing the same highway. In this thesis, we will present the concept of behavior banks that store primitive behaviors that can be selected by any suitable means like the one found in [42].

Unlike many distributed consensus protocols, the one developed in [43] is independent of the global information about the underlying network topology connecting the agents. The developed protocol adopts an adaptive scheme that ensures leader-follower consensus if the graph- representing the communication network- contains a directed spanning tree with the leader as the root node. This is equivalent to running a network discovery protocol in

parallel. Such adaptive approach may prove useful under situations where followers are added or removed dynamically.

Recalling Figure 1.9, any input delay caused by latency in decision making for example can affect the action taken. In [44], an event-triggered controller to achieve consensus between a leader and a group of followers was presented. The followers were assumed to have general linear dynamics subjected to input delay between the actuator and the controller. Such latency is related to timeliness issues considered by the Robotics community.

The containment of several followers by a group of dynamic leaders using a robust controller based on high-frequency feedback was presented in [45]. These agents were assumed to have an unknown- yet bounded- nonlinearities. It was shown in this paper that under the proposed controller that the said multi-agent behavior can be achieved if all followers are accessed directly by at least one leader over a directed communication path. The connections among the followers must be undirected. Such a behavior can be found in autonomous surface vessels acting as coast guards and warships escorting, for examples.

In [46], a group of nonlinear dynamical systems achieve consensus under unreliable communication and switching network topology. A leader-follower structure under the previously stated conditions is brought to consensus in a convergence rate and communication rate adjustable using the feedback gain. Both linear matrix inequality method and Lyapunov theory are used to derive the sufficient conditions of convergence.

As stated previously, in this thesis the concept of behavior banks can be controlled using suitable selection mechanisms where a specific behavior can be paired with a specific network topology. So, by switching the topology then the behavior will also change. In

[47], the concept of state-controlled switching topology is used. Under such switching law, multi-agent states play a significant role in choosing the appropriate network topology. The used states can represent the signal strength, position or velocity of the involved agents. Other network switching laws may depend on time or even on a random process that describes the working environment, for example.

Multi-agent systems are widely used in many applications like computation, exploration, rescuing, sensing and various military applications, just to name few. These connected agents relate to each other via an underlying connectivity means referred to as a communication network. Depending on the application, such networks may vary in nature from being logical, chemical or even up to the extent of being physically realized using rigid or soft links. One example of a physically realizable soft link is the wireless communication where no hard-wires are used. Similarly, one may consider cams or gear chains as kinds of physically realizable rigid links. In the gear chains example, the behavior, i.e., the response, of the attached objects at the ends of the chain will vary depending on the engaged gears, consider a vehicle gear box as an example where the transmission shaft rotational speed varies depending on the engaged gears. Thus, inspired by these facts, engineers sparked the utilization of homogeneous replicas of the same objects or heterogeneous objects to maximize the utilization of the available resources and to speed up meeting timely-manner needs. One interesting example is the Programmable Logic Gates (PLGs) where changing the connections among the available logic units will result in a new response of the used PLGs.

More specifically, in the field of physical MAS over soft links, engineers have studied the relations between the used agents, mainly of dynamical features, and the communication

network used to connect them. Starting from the simple cases to more complex situations, magnificent efforts available in the literature testify to the vast progress in this study. As a result, the spectrum of approaches used to solve mainly designing network protocols to control even the simplest dynamical system, namely: the single-integrator, is wide and scattered. It does not take much time to realize that in each contribution in the stated direction, different approaches to state the stability and convergence conditions do vary significantly.

A group of identical or non-identical agents connected over a graph form what is called a MAS. These agents may have different role to play in the overall MAS behavior. For example, an agent acting as a leader will provide commands to other agents on the graph most likely over unidirectional links. Other agents acting as followers may cooperate or compete over bi-directional links. More details on this will be provided in this thesis.

The inter-agent interaction over the graph is governed by a set of protocols designed specifically to orchestrate certain behaviors. For example in [2], relative positions among sensing agents in the 2-D plane were used to realize self-localization among these agents in a distributed fashion. In the same area, in [48] consensus protocols were designed to build consensus filters that facilitate fusion of the distributed measurements of sensing agents. Distributed frequency and phase consensus among clocks can be found in [49]. A general overview of consensus protocols can be found in [50].

Consensus behavior is widely studied in the research community, both linear and nonlinear consensus protocols were introduced in the related literature. For instance, in [51], [52] nonlinear consensus protocols over directed networks was studied. Other linear consensus protocols can be found in [50], [53]. Several types of consensus have also been studied, for

example, average [54], [55] and geometric [56] types of consensus were handled. Other behaviors like cluster consensus [1], formation [57] and coordination [58] were also studied.

Issues related to the stability of interconnected dynamical systems under the designed network protocols are very important, for instance in [12], a special class of positive matrices was used to study the stability of composite dynamical systems. Even recently, different areas have benefited from the same approach like found in [7], [8], [14]. Convergence of protocols is another concern which was investigated in [59], for example. In [60], bounded control of distributed protocols were designed. Prescribed performance [54], invariance [61] and connectivity preserving [57], [58], [62] were also studied. Introducing state-dependent weights into the network protocols was also handled in [63], [64], for instance.

Behaviors in dynamical systems have received a lot of attention from several communities. Enabling robots to interact with their surroundings is a demand in various applications. Unavoidably, the very basic definition of a behavior can be easily argued, nevertheless, a behavior does relate to the way an action is initiated or conducted while interacting with internal or external stimuli based on collected data using onboard sensory data for example or processed data that was previously available. Being interested in collective behaviors, i.e., of a group of robots or agents comprising a MAS, several tools to model, design, simulate and synthesize behaviors are needed. Depending on the perspective, expertise is usually needed to design and realize behaviors related to actual motion in the working space. Issues like stability, latency, collision avoidance and control effort are usual concerns in the robotics field.

From the modelling point of view, a behavior can be modelled using hierarchical Finite State Machine [65] or based on the interconnections among the involved elements [66]. A behavior can be designed according to specific structures and hierarchies like found in [67]–[70]. Various simulation platforms can be used to simulate a behavior, for example XABSL [71], ROS, OpenRDK and YARP. Detailed comparison among these utilities can be found in [70]. Control aspects under behavioral control were also studied, for example in [60], network connectivity in MASs was controlled using a bounded input signal. Saturated control signals under behavioral control were investigated in [72]. The MAS stability under time-dependent communication links was investigated in [73], [74]. The ability of the MAS to avoid collision, including self-collision, is very important so as to preserve the assets and to ensure performance. Collision avoidance was discussed- for example- in [75], [76].

In [77], coupled dynamics with multiple time-scales over state-dependent graphs has been studied. Two cases were mainly investigated in which the network dynamics act as the fast and slow dynamics in the overall coupled system. Singular perturbation theory has been used to obtain reduced order models helping in stability analysis under such coupling. The stability is ensured by quantitative bounds describing the underlying graph topology.

Comparing the approach presented in [77] to what will be presented in this thesis, it is interesting to note that in [77] the tracking error is also used to affect the consensus protocol. In addition, like what is proposed in this thesis, the same concepts of desired (ideal) and true (actual) states were used in [77]. A simulation example of steering eight unicycle-type robots was provided in [77]. These results can be compared to a similar example provided at the end of this thesis.

The leader-following consensus problem for a special class of nonlinear MASs was studied in [78]. The consensus protocol depends on sampled and delayed version of the output feedback signals. Such technique can reduce the needed communication bandwidth of the communication network. This can also be done using other techniques such as event-triggered as can be found in [79] where a cluster formation behavior was realized using also sampled data. Other types of event-triggered approaches can also be used like the one found in [80] where the containment behavior over directed graphs was realized using event-triggered broadcasting. Additionally, using periodic event-triggering and quantized data was used in [81] to solve the consensus problem among several agents. A thorough review of sampled-data distributed consensus protocols is available in [82]. Recent reviews of distributed protocols can be found in [83] and [84].

No doubt, great efforts were reported in the literature concerning the field of robotics in its variant disciplines. Here, in this thesis, a proposed framework that suits various situations and dynamical behaviors usually addressed by the robotics community is presented. Dynamical behaviors like consensus, clustering, formation, leader-follower, communication-aware, deployment, connectivity-preserving, flocking, shape consensus, containment and escorting under state-dependence are presented in a systematic way that emphasizes upon the strength, effectiveness, correctness, consistency while keeping the simplicity of the proposed framework. Aspects like stability, bounded control signals, connectivity-preservation validation and design, and switched behavior control are also discussed in more details. Using simple hybrid automata in two examples show the framework interface to the behavioral control from algorithmic point of view, on one hand. On the other hand, providing the dynamical aspects like stability, control boundedness,

connectivity preservation and switching stability proofs shows the applicability of the proposed framework under realistic scenarios where obstacles can be stationary or moving and the mission is conducted under varying contexts.

Other details are covered based on the need during the sequel of this thesis.

CHAPTER 3

The Fundamentals of the Proposed Framework

In this chapter, details related to the structure of the framework are provided, where the integration of first integrals and nonlinear eigenvalue problems that constitute its core is presented. It will be clear in the sequel that the nonlinear protocols can be converted into their equivalent semi-linear protocols by applying a simple transformation, in general. However, the needed conditions will be rendered the same in both cases. Consensus protocols- both semi-linear and nonlinear- to achieve arithmetic, geometric, harmonic means are derived. A generalization to the mean-of-order-p is also provided. Stability issues are mainly handled using properties of M-matrices and Lasalle's Invariance principle. The relation between consensus protocols and potential fields are explained and utilized. Simple examples to show the applicability of the proposed framework in designing bounded controls that meet a prescribed performance are provided. Simulation results show the feasibility of this framework.

3.1 Consensus and Clustering of N-identical Single-Integrator MAS over State-dependent Network Protocols

Starting with a MAS system with fixed number of agents whose dynamics are modeled as follows:

$$\dot{x}_i(t) = u_i(t) \in \mathfrak{R}, \quad \forall i = 1, 2, \dots, N \quad (3.1)$$

Let u_i , i.e., the diffusive-coupling network protocol, be given as follows:

$$\begin{aligned}
u_i &= - \sum_{j \in \mathcal{N}_i} g_{ij}(\vec{x}) \{x_i - x_j\} \\
g_{ij}(\vec{x}) &= \begin{cases} g_{ij}(\vec{x}) > 0, & j \in \mathcal{N}_i \\ 0, & j \notin \mathcal{N}_i \end{cases}
\end{aligned} \tag{3.2}$$

where: $g_{ij}(\vec{x}): \mathfrak{R}^N \rightarrow \mathfrak{R}_{>0}$ is a general bounded weighting vector-valued functional of the global state vector $\vec{x} = [x_1, x_2, \dots, x_N]^T \in \mathfrak{R}^N$ and $j \in \mathcal{N}_i$ denotes that the j^{th} agent is an in-neighbor of the i^{th} agent. Note that g_{ij} was used instead of w_{ij} to highlight that the edges' weights are dependent on the agents states and they are not fixed any more. Note also that the used network protocol must be designed in accordance with the prescribed tasks desired, for example protocol (3.2) can be used to achieve mainly the *average consensus* among the connected agents as will be clarified soon.

Remark 3.1: The weighting function given in (3.2), namely $g_{ij}(\vec{x})$, must depend on local information available only at the agent level to make the protocol distributed. Thus, it can be written as $g_{ij}(x_i, x_j)$ instead which indicates that it depends only on the states of the agent itself and its neighbor, i.e., $g_{ij}(\vec{x}): \mathfrak{R}^2 \rightarrow \mathfrak{R}_{>0}$. Note that g_{ij} must be strictly positive to ensure the stability of the MAS as will be explained later in this chapter.

Thus, (3.1) can be rewritten for short- by dropping the explicit time dependency- as follows:

$$\begin{aligned}
\dot{x}_i &= - \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{x_i - x_j\} \\
&= -x_i \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) + \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) x_j
\end{aligned} \tag{3.3}$$

Simplifying (3.3) more, yields:

$$\dot{x}_i = -x_i d_i + \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) x_j = u_i \quad (3.4)$$

where: $d_i = \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j)$ is the in-degree of the i^{th} agent. Taking $\mathcal{D} = \text{diag}([d_1, d_2, \dots, d_N])$, we can write the global dynamics of the connected MAS using the following deterministic system model:

$$\dot{\vec{x}}(t) = -(\mathcal{D}(\vec{x}) - \mathcal{A}(\vec{x}))\vec{x} = -\mathcal{L}(\vec{x})\vec{x}(t) = \vec{f}(\vec{x}) \quad (3.5)$$

where: \vec{f} is a smooth function with $\vec{f}(\vec{0}) = \vec{0}$, $\mathcal{A} = [g_{ij}]$ is the adjacency matrix associated with the graph representing the available communication network among agents, and $\mathcal{L}(\vec{x})$ - or \mathcal{L}_s for short- is the resulting state-dependent Laplacian matrix. Notice that when all g_{ij} functions do not depend on states or time, then the resulting Laplacian matrix will be fixed, i.e., the network topology and the associated edges weights are invariant, with respect to states and time. Therefore, the resulting Laplacian matrix can be denoted by \mathcal{L}_f to differentiate it from the state-dependent one. Later, an interesting relation between the fixed and state-dependent Laplacian matrices will be a corner-stone in the proposed framework presented in this thesis.

Remark 3.2: One of the main objectives of the proposed framework is to establish a relation between the theory of the fixed and state-dependent Laplacian matrices of the same network graph. This allows borrowing some of the available work and theories of the fixed Laplacian matrix to deal with the state-dependent one, on the one hand, and to deal with more realistic and complex situations on the other hand since networks are allowed to depend explicitly on agents' states. Moreover, other parameters representing the working environment for instance can also be incorporated within the structure of the Laplacian

matrix through the g_{ij} functions- as it is the case of communication-aware robotics e.g. [85]- which is beyond the scope of this chapter.

The differential equation given in (3.5) is nonlinear in general and thus a special consideration of its elements must take place since it affects the stability of the MAS. Consensus among agents, is one of the important behaviors investigated in the literature. Motivated by the application, several types of consensus do exist such as arithmetic, geometric and harmonic means. These types of consensus- among others as well- can be encapsulated into a general type called the mean-of-order- p , see [86] for example. Other interesting versions of these means are their corresponding weighted means which are important in understanding the MAS behavior over a special type of communication networks represented by directed graphs, or digraphs for short, which are not thoroughly covered here.

Clustering can be viewed as a special case of consensus where several partial consensuses among some agents within the MAS are established. Thus, several subgroups or clusters can be obtained at steady-state response of the MAS. More technically, the mean difference between consensus and clustering is the nature of the null-space spanned in both cases. In a complete consensus among the agents, the null-space is spanned using the vector associated with $\lambda_1 = 0$, namely: $c\vec{1}$ where $c \neq 0 \in \Re$ which indicates that all agents are cooperative. In contrast, in cluster consensus, the null-space can be spanned by a non-positive vector whose elements are other than 1 which indicate that some agents are cooperative while others are competitive. In such a case, the complete consensus protocol (3.4) is modified as follows:

$$\dot{x}_i = -x_i d_i + \sum_{j \in \mathcal{N}_i} d_{ij} g_{ij}(x_i, x_j) x_j = u_i \quad (3.6)$$

where: $d_{ij} \in \mathcal{R}$ is the fixed competitive or cooperative weight that is usually given as a real number whose sign is positive or negative, respectively. This weight will convert the unsigned network graph into a signed graph [87] which is a special case of a gain graph. To maintain the in-degree, i.e., d_i , and therefore the stability of the resulting MAS, it is required to have $d_{ij} = 1/d_{ji} = \sigma_j/\sigma_i$ where σ_i reflects the desired behavior of the i^{th} agent with respect to the j^{th} agent- either competitive or cooperative. If $\sigma_i = \sigma_j$, then both agents are in the same cluster. Mathematically, the relation between both (3.4) and (3.6) is reflected on their corresponding Laplacian matrices so that the resulting MAS dynamics can be given as follows:

$$\vec{\dot{x}}(t) = -(D^{-1}\mathcal{L}(\vec{x})D)\vec{x}(t) = \vec{h}(\vec{x}) \quad (3.7)$$

where: $D = diag(\sigma_1, \sigma_2, \dots, \sigma_N)$. Note that multiplying a matrix with vectors / matrices with similar signs will not change its sign. To eliminate any notational ambiguity, the complete consensus is referred to as *consensus* while *cluster consensus* will be explicitly used when needed.

To make it possible to compare the proposed framework to the results available in the related literature, it is desired to provide detailed derivations of the previously mentioned consensus types over mainly state-dependent undirected graphs. These derivations will be used later to reveal the link between various types of MAS consensus over state-dependent and fixed communication graphs. It is worth noting that these derivations followed a certain methodology that reflects an important ingredient of the proposed framework.

3.1.1 Arithmetic Mean Consensus

In some applications, like distributed sensor fusion [88], the arithmetic mean is the best estimate of the measured quantity if sensors have identical noise characteristics [9]. If the communication graph is weight-balanced, i.e., weighted in-degree equals weighted out-degree for all agents [9], then the associated Laplacian matrix will have the elements of each row add up to zero. The same can be said about the elements of each column of such graphs if they are undirected. Note that the in-degree of the i^{th} agent is the sum of the i^{th} row while its out-degree is the sum of the i^{th} column. The arithmetic mean (μ_a), i.e., the average, of N measurements, or states, is given as follows:

$$\mu_a(t) = \frac{1}{N} \sum_{i=1}^N x_i(t), \quad \forall t \geq t_0 \quad (3.8)$$

If the average at t_0 is to be achieved and maintained, then (3.8) must be time-invariant.

Thus, it is straight forward to show that:

$$\frac{d\mu_a(t)}{dt} = \frac{1}{N} \sum_{i=1}^N \frac{dx_i(t)}{dt} = 0, \quad \forall t \geq t_0 \quad (3.9)$$

Substituting (3.1), i.e., the system dynamics, into (3.9) yields the following:

$$[1 \ 1 \ \cdots \ 1][u_1, u_2, \cdots u_N]^T = 0 \quad (3.10)$$

where: $[\cdot]^T$ denotes the transpose.

Having in mind that the average value is to be estimated in a distributed fashion, a nontrivial input u_i must be carefully designed such that it utilizes the measurements originating only from the agent itself and from its in-neighbors. One possible state-dependent semi-linear protocol that can achieve, yet to be verified, the desired goal is given

by (3.2). It should be kept in mind that there might be other candidates, nonlinear in principle, that can achieve the same goal as will be shown later.

From (3.4), it is possible to rewrite (3.10) as follows:

$$-x_1 d_1 + \sum_{j \in \mathcal{N}_1} g_{1j}(x_1, x_j) x_j - \cdots - x_N d_N + \sum_{j \in \mathcal{N}_N} g_{Nj}(x_N, x_j) x_j = 0 \quad (3.11)$$

which can be stacked in a more compact form as follows:

$$[1 \cdots 1] \begin{bmatrix} -d_1(x_1, x_{j \in \mathcal{N}_1}) & \cdots & g_{1N}(x_1, x_N) \\ \vdots & \ddots & \vdots \\ g_{N1}(x_N, x_1) & \cdots & -d_N(x_N, x_{j \in \mathcal{N}_N}) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = 0 \quad (3.12)$$

or simply as:

$$\vec{1}^T \mathcal{L}(\vec{x}) \vec{x}(t) = 0, \quad \forall \vec{x}, t \geq t_0 \quad (3.13)$$

A sufficient condition to solve (3.11) *algebraically* is:

$$g_{ij} = g_{ji}, \quad \forall i, j \in \mathcal{N}_i, \forall \vec{x}, t \geq t_0 \quad (3.14)$$

However, solving (3.12) needs more attention, in general. When the graph is fixed, $\mathcal{L}(\vec{x})$ can be given as \mathcal{L}_f and (3.13) can be written as follows:

$$\vec{1}^T \mathcal{L}_f \vec{x}(t) = 0, \quad \forall t \geq t_0 \quad (3.15)$$

which indicates that the $\vec{1}^T = [1 \ 1 \cdots 1]_{N \times 1}$ is simply a left eigenvector of \mathcal{L}_f associated with the zero-eigenvalue resulting from the fact that the row-sum of the Laplacian matrix is always zero [89] if it is weight-balanced. To elaborate more on this point, let the global model of the MAS consisting of N-identical single-integrator agents interacting over a fixed and undirected communication graph be given as follows:

$$\dot{\vec{x}}(t) = -\mathcal{L}_f \vec{x}(t) \quad (3.16)$$

In linear algebra, the left and right eigenvalue problems can be given respectively as follows:

$$\vec{v}_f^T(\lambda_1 I + \mathcal{L}_f) = \vec{0} \quad (3.16.1)$$

$$(\lambda_1 I + \mathcal{L}_f)\vec{e}_f = \vec{0} \quad (3.16.2)$$

where: $I \in \Re^{N \times N}$ is the identity matrix, \vec{v}_f^T and \vec{e}_f are the linear left and right eigenvectors, respectively. For $\lambda_1 = 0$, it is straightforward to see the relation between the left eigenvalue problem (3.16.1) and (3.15). The stability of the linear time-invariant system (3.16) can be guaranteed if and only if $\lambda_1 = 0$ is simple which- from a graph theoretic point of view- requires the underlying graph associated with $\mathcal{L}_f \geq 0$ to be always connected, i.e., $|\lambda_2| > 0$.

Remark 3.3: The similar structure of both (3.13) and (3.15) motivates one design method as proposed in this framework. However, (3.15) is a linear eigenvalue problem while (3.13) is a nonlinear one. In the coming section, the nonlinear eigenvalue problem will be handled. The null-space vector of (3.16) is given as:

$$\vec{x}_{eq} = \mu_a \vec{1}^T, \quad \mu_a \in \Re \quad (3.17)$$

If the Laplacian matrix has a simple, i.e., not repeated, zero-eigenvalue, then its rank will be $N - 1$ and the $\vec{1}^T$ will belong to its null-space, i.e., $\vec{x}_{eq} = \mu_a \vec{1}^T \in \text{Null}(\mathcal{L}_f)$. It was shown in [89] that if the graph is connected, then it has a simple zero-eigenvalue. Moreover, it was also shown in [89] that when \mathcal{L}_f is positive semidefinite and $\mu_a \vec{1}^T$ is the only equilibria set- or point-, then the network protocol (3.2), when g_{ij} functions are all constants with 0-1 weights, will *globally* and *asymptotically* drive the MAS (3.16) to the average-consensus, i.e., the arithmetic mean value, given by (3.8).

In [90], it was shown that a *necessary* and *sufficient* condition for \mathcal{L}_f to have $\vec{1}^T$ as a left eigenvector is that the communication graph must be a balanced digraph, i.e., a graph

whose rows and columns sum up to zeros. In [91], it was shown that \mathcal{L}_f will have a simple zero-eigenvalue *if and only if* the associated communication graph has a spanning tree, i.e., includes all the graph vertices with the least number of edges. However, for such graphs, the average consensus is not straightforward and requires special treatment like the one found in [55]. For a connected undirected graph, the $\vec{1}^T$ is a left eigenvector associated with the zero eigenvalue of \mathcal{L}_f [89] and this will be the main type of graphs treated in this thesis.

3.1.2 Geometric Mean Consensus

In some applications, like in distributed Bayesian consensus filtering [92], the geometric mean plays an important role and it is required to be estimated in a distributed fashion. Following the same approach presented in deriving the arithmetic mean consensus protocol, a geometric mean consensus protocol can be derived knowing that the geometric mean is given as follows:

$$\mu_g(t) = \sqrt[N]{\prod_{i=1}^N x_i(t)}, \quad \forall t \geq t_0, x_i(t) \neq 0 \quad \forall i \quad (3.18)$$

If the geometric mean at t_0 is to be achieved and maintained, then (3.18) must be time-invariant. Thus, it is straightforward to show that $\forall t \geq t_0$:

$$\frac{d\mu_g(t)}{dt} = \overbrace{\left(\prod_{i=1}^N x_i \right)^{\frac{1}{N}-1}}^{\text{non-zero}} \frac{\{\dot{x}_1 x_2 \cdots x_N + \cdots + x_1 x_2 \cdots \dot{x}_N\}}{N} = 0 \quad (3.19)$$

A possible candidate protocol that can be used is given as follows:

$$u_i = -x_i \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{x_i - x_j\} \quad (3.20)$$

Substituting (3.1) and (3.20) into (3.19) yields the following:

$$\left\{ \sum_{j \in \mathcal{N}_1} g_{1j}(x_1, x_j) \{x_1 - x_j\} + \cdots + \sum_{j \in \mathcal{N}_N} g_{Nj}(x_N, x_j) \{x_N - x_j\} \right\} = 0 \quad (3.21)$$

which can be stacked in a more compact form as (12) and (3.13) which indicates that the

$\vec{1}^T = [1 \ 1 \ \cdots \ 1]_{N \times 1}$ is simply a left eigenvector of $\vec{f}(\vec{x})$ given in (3.5).

Remark 3.4: It should be noted that all x_i must have the same sign for the geometric mean to be defined.

3.1.3 Harmonic Mean Consensus

In clock synchronization [49] for example, the distributed estimation of the harmonic mean of some variables is crucial. Following similar steps as done previously, the harmonic mean consensus protocol can be derived knowing that the harmonic mean is given as follows:

$$\mu_h(t) = \frac{N}{\sum_{i=1}^N \frac{1}{x_i(t)}}, \quad \forall t \geq t_0, x_i(t) > 0, \forall i \quad (3.22)$$

If the harmonic mean at t_0 is to be achieved and maintained, then (3.22) must be time-invariant. Thus, it is straight forward to show that $\forall t \geq t_0$:

$$\frac{d\mu_h(t)}{dt} = \frac{N \left(\frac{\dot{x}_1}{x_1^2} + \frac{\dot{x}_2}{x_2^2} + \cdots + \frac{\dot{x}_N}{x_N^2} \right)}{\underbrace{\left(\sum_{i=1}^N \frac{1}{x_i(t)} \right)^2}_{non-zero}} = 0, \quad \forall t \geq t_0 \quad (3.23)$$

A possible candidate protocol obtained by educated guessing is given as follows:

$$u_i = -x_i^2 \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{x_i - x_j\} \quad (3.24)$$

Substituting (3.1) and (3.24) into (3.23) yields the following:

$$\left\{ \sum_{j \in \mathcal{N}_1} g_{1j} \{x_1 - x_j\} + \cdots + \sum_{j \in \mathcal{N}_N} g_{Nj} \{x_N - x_j\} \right\} = 0 \quad (3.25)$$

which can be stacked in a more compact form as (3.12) and (3.13) which, again, indicates that the $\vec{1}^T = [1 \ 1 \ \cdots \ 1]_{N \times 1}$ is simply a left eigenvector of $\vec{f}(\vec{x})$ given in (3.5). In (3.25), $g_{ij}(x_i, x_j)$ was written as g_{ij} for sake of compactness.

3.1.4 Mean-of-order- p Consensus

This mean is a generalization of the previous means plus other types of mean, like minimum and maximum means [86]. It is also known as the Hölder [93] or power mean for only positive values and is given as follows:

$$\mu_p(t) = \sqrt[p]{\frac{1}{N} \sum_{i=1}^N x_i^p(t)}, \quad \forall t \geq t_0, x_i(t) > 0 \quad \forall i \quad (3.26)$$

If the mean-of-order- p - where p is an integer number- at t_0 is to be achieved and maintained, then (26) must be time-invariant. Thus, it is straight forward to show that $\forall t \geq t_0$:

$$\frac{d\mu_p(t)}{dt} = \overbrace{\frac{1}{N} \left(\frac{1}{N} \sum_{i=1}^N x_i^p(t) \right)^{\frac{1}{p}-1}}^{\text{non-zero}} \sum_{i=1}^N x_i^{p-1} \dot{x}_i = 0 \quad (3.27)$$

A possible candidate protocol that can be used is given as follows:

$$u_i = -x_i^{1-p} \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{x_i - x_j\} \quad (3.28)$$

Substituting (3.1) and (3.28) into (3.27) yields an equation like (3.25) which can be stacked in a more compact form as (3.12) and (3.13) which, again, indicates that the $\vec{1}^T = [1 \ 1 \ \dots \ 1]_{N \times 1}$ is simply a left eigenvector of $\vec{f}(\vec{x})$ given in (3.5).

Remark 3.5: It should be noted that even though (3.11) and (3.12) are mathematically equivalent, yet they may direct the mind into two different directions to find a solution. Notice that (3.11) is more general yet less structured once compared to (3.12). Moreover, (3.12) can be brought into a specific structure that will facilitate a systematic way of designing g_{ij} functions and to maintain a visible relation with the already available theories of fixed graphs in the related literature. More discussion regarding these issues will be provided in the coming section.

Observing (3.26), the value of the order- p can be chosen as 1, 0 and -1 such that the arithmetic, geometric and harmonic means are obtained, respectively [86]. Moreover, if the value of the order- p is chosen as 2, ∞ and $-\infty$, then quadratic mean, maximum and minimum of $\vec{x}(t)$ can be obtained, respectively [86].

Remark 3.6: Even though the previous protocols are available in the literature, like [94] for instance, their derivations are extended herein to the case of state-dependent Laplacian matrices.

Remark 3.7: The conditions obtained mainly in (3.11), (3.12), and (3.13) motivates the framework proposed in this thesis where the concept of eigenvalues and eigenvectors were borrowed from linear algebra to provide some important results in the domain of MAS on a graph. Following the same concept, nonlinear eigenvalues and eigenvectors are used herein to extend the work for state-dependent graphs while maintaining a clear and structured approach for the design process.

3.2 The proposed framework

In this section, the proposed framework consisting of a combination of two design steps is presented. The two steps are the nonlinear eigenvalue-based and the first integral design approaches. Using this framework, various semi-linear and nonlinear protocols can be designed in a systematic way while keeping a clear connection between state-dependent and fixed Laplacian matrices associated with the same graph topology. This relation can be realized through the preservation of the eigenvalue and eigenvector concepts between the two types of the Laplacian matrices. This preservation is also projected onto keeping certain properties of the fixed Laplacian matrices even after upgrading them to be state-dependent ones. This implies preserving the *M-matrix* properties which plays a significant role once the stability and convergence of the designed protocols are considered. Other details are provided in the coming subsections.

3.2.1: Nonlinear Eigenvalue-based Design Approach

Here, the main tool used in forming the proposed framework is presented. First, the desired properties of the state-dependent Laplacian matrix to retain are stated followed by the introduction of nonlinear eigenvalues and eigenvectors.

3.2.1.1: Properties of the state-dependent Laplacian matrix

To ease the analysis of the stability of (3.5) and to preserve a clear link with the fixed Laplacian matrices-related theories, the following structure of the state-dependent Laplacian matrix is proposed:

- 1- $\mathcal{L}(\vec{x})$ must be a point-wise, i.e., $\forall \vec{x}(t)$, singular *M-matrix*.
- 2- $\mathcal{L}(\vec{x})$ must have the same spectral properties of the desired fixed Laplacian matrix at consensus and at origin if the latter is included in the equilibria set.

3- $\mathcal{L}(\vec{x})$ must always have the $\lambda_1 = 0$ as a simple eigenvalue irrespective of the states $\vec{x}(t)$ values, i.e., the graph must always be connected which implies also that $|\lambda_2| > 0$, $\forall \vec{x}(t)$.

Definition 3.3: An M -matrix \mathbb{M} is a diagonally dominant matrix with non-positive off-diagonal elements which can be given mathematically as follows [13]:

$$\mathbb{M} = sI - B, \quad s \geq \rho(B), \quad B \geq 0$$

where: I is the identity matrix and $\rho(B)$ is the spectral radius of B . If $s > \rho(B)$, then \mathbb{M} is called a nonsingular M -matrix. If $s = \rho(B)$, then \mathbb{M} is a singular M -matrix. The general set of positive numbers, denoted by $Z^{N \times N}$, has $s > 0$. Thus, $\mathbb{M} \in Z^{N \times N}$.

A Laplacian matrix is an M -matrix having both the rows and columns sum up to zero if it is associated with an undirected graph. Thus, $\mathcal{L}_f \in \mathbb{M} \subset Z^{N \times N}$.

Note 3.1: It will be assumed that the edges of the graph of concern, represented by \mathcal{L}_f , have only 0-1 weights and the g_{ij} functions can be written as linear combinations of the elements of \mathcal{L}_f unless otherwise clearly stated. In other words, $\mathcal{L}(\vec{x})$ can be written as a linear combination of another state-dependent Laplacian matrix and \mathcal{L}_f , i.e., $\mathcal{L}(\vec{x}) = \hat{\mathcal{L}}(\vec{x}) + \sigma \mathcal{L}_f$, if the needed conditions are satisfied. In such a case, it is possible to have $g_{ij} \geq 0$ and to obtain a prescribed connectivity performance under state-dependent networks as will be shown later. Figure 3.1 shows a regular undirected link and Figure 3.2 shows how an edge may look like under such circumstance where the parallel state-dependent edge will increase the strength of coupling between the connected agents if $g_{ij} > 0$ and thus might increase the MAS convergence rate. So, practically speaking, this means that the link is always ensured. However, this might be possible when designing a controller rather than modeling a realistic communication network which may break under some conditions.

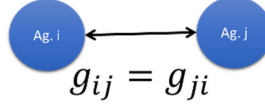


Figure 3.1. A regular undirected link.

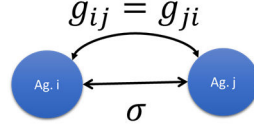


Figure 3.2. The effect of having $\mathcal{L}(\vec{x})$ linearly dependent on \mathcal{L}_f .

In general, the previously stated properties can be achieved using carefully crafted $g_{ij}(x_i, x_j)$ functions in such a way that when time is sufficiently large, the state-dependent Laplacian matrix will converge to the desired fixed Laplacian matrix designed to meet the requirements of a specific task.

Unlike [63] where the desired fixed Laplacian matrix is explicitly entered in the protocol designed, the structure of the proposed state-dependent Laplacian matrix can embed the desired fixed Laplacian matrix through $g_{ij}(x_i, x_j)$ functions implicitly- mainly- if they are nonlinear.

Referring back to (3.5), it is clear that $\vec{f}(\vec{x})$ is a nonlinear vector-valued function that exhibits a semi-linear structure, namely $-\mathcal{L}(\vec{x})\vec{x}(t)$, under the proposed protocol shown in (3.3). So, to obtain the desired properties of the state-dependent Laplacian matrix, one must ensure the following:

- 1- The $g_{ij}(x_i, x_j) \geq 0, \forall i, j$. This is to ensure the properties of *M-matrices*, considering

Note 3.1.

- 2- The $g_{ij}(x_i, x_j)$ must be locally Lipschitz.

- 3- The ones vector, i.e., $\vec{1}$, must be a left eigenvector associated with the simple eigenvalue $\lambda_1 = 0$ of $\vec{f}(\vec{x})$, i.e., each column in the Laplacian matrix must sum up to zero. In other words, the matrix must be symmetric. This agrees with (14).
- 4- The spectrum of $\mathcal{L}(\vec{x})$ must be such that $\lambda_1 = 0 < |\lambda_2| \leq \dots \leq |\lambda_N|, \forall \vec{x} \in \Omega \subseteq \mathbb{R}^N$ which indicates that the underlying undirected graph must always be connected. More details will be provided when the concept of prescribed connectivity is presented.

It is well-known that the concepts of eigenvalues and eigenvectors are associated with linear systems. However, these concepts got extended to include nonlinear functions as can be found in [95]–[99].

According to [95], the nonlinear left eigenvalue problem can be given as follows:

$$\vec{v} = \frac{\partial \vec{v}}{\partial \vec{x}} \vec{f}(\vec{x}) = \lambda(\vec{x}) \vec{v} - \left(\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \right)^T \vec{v} \quad (3.29)$$

where: \vec{v} is the associated left eigenvector and in general it is a function of the states, i.e., $\vec{v}(\vec{x})$. Similarly, the nonlinear right eigenvalue problem can be given as follows [95]:

$$\vec{e} = \frac{\partial \vec{e}}{\partial \vec{x}} \vec{f}(\vec{x}) = -\lambda(\vec{x}) \vec{e} + \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \vec{e} \quad (3.30)$$

Before solving (3.29), the relation between the fixed and state-dependent Laplacian matrices should be made clear. Since a fixed Laplacian matrix associated with a specific graph- whether directed or undirected- is a singular *M-matrix*, the resulting equilibria set, or the agreement space [89], can be characterized as follows:

$$\mathfrak{S} = \{\vec{x}(t) \in \mathbb{R}^N \mid \vec{x} = c\vec{1}\} \quad (3.31)$$

where $c \in \mathbb{R}$ is the *collective decision* of the group of agents [89], which is assumed to be time-invariant.

It is desirable, as per the proposed approach, to establish the relation between the fixed and the state-dependent Laplacian matrices not only about the origin and consensus but also for the whole state space, if applicable. The fixed Laplacian matrix \mathcal{L}_f can be viewed as a special case of the state-dependent Laplacian matrix $\mathcal{L}(\vec{x})$ at the equilibria set. However, certain attributes of the origin under $\mathcal{L}(\vec{x})$ will be more difficult to investigate once compared to the case under \mathcal{L}_f like the region of attraction and stability where initial conditions and g_{ij} functions play significant roles in the consensus problem as will be investigated later. Mathematically, it is desired to have:

$$\left. \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \right|_{\mathfrak{S}} = -\mathcal{L}(\vec{x})|_{\mathfrak{S}} = -\mathcal{L}_f \quad (3.32)$$

where \mathcal{L}_f is imposed originally by the designer based on the ultimate application objectives.

By design, a connected \mathcal{L}_f has $\lambda_1 = 0$ as a simple eigenvalue with $\vec{v}_f = \vec{1}c$ as an associated linear left eigenvector. Considering the previous discussion, (3.29) can be given as follows when generally evaluated at $\lambda_1(\vec{x}) = 0$ and $\vec{v}(\vec{x}) = \vec{1}c \ \forall \vec{x} \in \Omega \subseteq \mathfrak{R}^N$:

$$\left(\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \right)^T \vec{1} = \vec{0} \quad (3.33.1)$$

$$\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \vec{1} = \vec{0} \quad (3.33.2)$$

Note 3.2: If both (3.33.1) and (3.33.2) are satisfied, then $\lambda_1(\vec{x}) = 0$ will be a nonlinear eigenvalue for both in-system and out-system as will be explained later in this chapter. Consequently, both systems will also have $\vec{v}(\vec{x}) = \vec{1}c \ \forall \vec{x} \in \Omega \subseteq \mathfrak{R}^N$ and therefore they will possess the same invariant quantity $\forall \vec{x}$. The in-system is related to the in-neighborhood of agents in the MAS while the out-system is related to the out-neighborhood of agents in the same MAS. In simpler words, the Jacobian will be symmetric and the in-

flow and the out-flow of information at each agent level will be the same. If only (3.33.1) is satisfied, then the out-flow of agents may not reach average consensus in the out-system $\forall \vec{x}$. Similarly, if only (3.33.2) is satisfied, then the in-flow of agents may not reach average consensus in the in-system $\forall \vec{x}$. However, since (3.32) is forced by the proposed framework, the two systems will achieve the same average consensus value inside \mathfrak{S} .

Theorem 3.3: A sufficient condition to have $\vec{v}^T = \vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ is to design the $g_{ij}(x_i, x_j)$ functions as follows:

$$\frac{\partial g_{ij}(x_i, x_j)}{\partial x_i} = \frac{\partial g_{ji}(x_i, x_j)}{\partial x_i}, \quad \forall i, j \in \mathcal{N}_i, \vec{x}(t) \quad (3.34)$$

Proof: Let the dynamics of N single-integrator MAS over a state-dependent graph Laplacian be given as follows, assuming a fully connected complete undirected graph:

$$\dot{\vec{x}}(t) = -\mathcal{L}(\vec{x})\vec{x}(t) = \vec{f}(\vec{x}) \quad (3.34.1)$$

$$\vec{x}(t) = \begin{bmatrix} -x_1 \sum_{j=2}^N g_{1j} + x_2 g_{12} + \cdots + x_N g_{1N} \\ x_1 g_{21} - x_2 \sum_{j=1, j \neq 2}^N g_{2j} + \cdots + x_N g_{2N} \\ \vdots \\ x_1 g_{N1} + x_2 g_{N2} + \cdots - x_N \sum_{j=1}^{N-1} g_{Nj} \end{bmatrix} \quad (3.34.2)$$

So, the Jacobian of $\vec{f}(\vec{x})$ is evaluated as follows:

$$\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \cdots & \frac{\partial f_N}{\partial x_N} \end{bmatrix} \quad (3.34.3)$$

Now, solving (3.33.1) is equivalent to solving the following equations, which states that any j^{th} column in the Jacobian must sum up to zero:

$$\sum_{i=1}^N \frac{\partial f_i}{\partial x_j} = 0, \quad \forall j = 1, 2, \dots, N \quad (3.34.4)$$

Since all columns must satisfy (3.34.4), the first column is only solved as follows:

$$\begin{aligned} \sum_{i=1}^N \frac{\partial f_i}{\partial x_1} = & \left\{ - \sum_{k=2}^N g_{1k} + \left(\sum_{k=2}^N \frac{\partial g_{1k}}{\partial x_1} (x_k - x_1) \right) \right\} + \left\{ g_{21} + \left(\sum_{k=1, k \neq 2}^N \frac{\partial g_{2k}}{\partial x_1} (x_1 - x_k) \right) \right\} + \dots \\ & + \left\{ g_{N1} + \left(\sum_{k=1}^{N-1} \frac{\partial g_{Nk}}{\partial x_1} (x_1 - x_k) \right) \right\} = 0 \end{aligned} \quad (3.34.5)$$

Which can be simplified as follows:

$$\begin{aligned} \sum_{j=1}^N \frac{\partial f_i}{\partial x_j} = & \left\{ - \sum_{k=2}^N g_{1k} + g_{21} + \dots + g_{N1} \right\} + \left\{ \left(\sum_{k=2}^N \frac{\partial g_{1k}}{\partial x_1} (x_k - x_1) \right) + \left(\sum_{k=1, k \neq 2}^N \frac{\partial g_{2k}}{\partial x_1} (x_1 - x_k) \right) + \dots \right. \\ & \left. + \left(\sum_{k=1}^{N-1} \frac{\partial g_{Nk}}{\partial x_1} (x_1 - x_k) \right) \right\} \\ = & 0 \end{aligned} \quad (3.34.6)$$

The first bracket in (3.34.6) is equal to zero- by forcing (3.14). The second bracket however can be made equal to zero $\forall \vec{x}$ only if certain conditions are forced. Since, by design, a given g_{ij} function depends only on the states of the i^{th} and the j^{th} agents, one possible sufficient condition to satisfy (3.34.6) is by forcing:

$$\frac{\partial g_{ij}(x_i, x_j)}{\partial x_i} = \frac{\partial g_{ji}(x_i, x_j)}{\partial x_i}, \quad \forall i, j \in \mathcal{N}_i, \vec{x}(t)$$

■

It is worth noting that under the proposed framework, (3.34) presents a sufficient condition to reach consensus among connected agents of dynamics given in (3.1) under network protocol (3.3). This is totally a different approach once compared to [59] in which a similar problem was handled. In [59], a general class of consensus protocols of single-integrator MAS acting over static and fixed directed graphs under state-dependent combination of relative positions of an agent neighbors is handled. Sufficient conditions for convergent consensus protocols were provided based on the contracting properties of the convex hull initially containing agents. A sufficient condition that can be used to force $\vec{e} = \vec{1}$ as a nonlinear right eigenvalue problem is given by the following theorem.

Theorem 3.4: A sufficient condition to have $\vec{e} = \vec{1}$ as a nonlinear right eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ is to design the $g_{ij}(x_i, x_j)$ functions as follows:

$$\frac{\partial g_{ij}(x_i, x_j)}{\partial x_i} = -\frac{\partial g_{ij}(x_i, x_j)}{\partial x_j}, \quad \forall i, j \in \mathcal{N}_i, \vec{x}(t) \quad (3.35)$$

Proof: Equating the sum of each i^{th} row of the Jacobian given by (3.34.3) to zero, yields:

$$\sum_{j=1}^N \frac{\partial f_i}{\partial x_j} = 0, \quad \forall j = 1, 2, \dots, N \quad (3.35.1)$$

Since all rows must satisfy (3.35.1), the first row is only solved as follows:

$$\begin{aligned} \sum_{j=1}^N \frac{\partial f_1}{\partial x_j} = & \left\{ -\sum_{k=2}^N g_{1k} + \left(\sum_{k=2}^N \frac{\partial g_{1k}}{\partial x_1} (x_k - x_1) \right) \right\} + \left\{ g_{12} + \frac{\partial g_{12}}{\partial x_2} (x_2 - x_1) \right\} \\ & + \dots + \left\{ g_{1N} + \frac{\partial g_{1N}}{\partial x_N} (x_N - x_1) \right\} = 0 \end{aligned} \quad (3.35.2)$$

One sufficient condition to equate (3.35.2) to zero is by forcing (3.35).

■

Each edge on the graph connecting two agents may have a different g_{ij} function. This possibility will enrich the behavior of the MAS in which neighbors may behave differently towards each other not only based on their relative distance for example, but also due to other designed properties like hierarchical roles in a specific application. This is depicted in Figure 3.3 and is given mathematically as follows:

$$u_i = - \sum_{j \in \mathcal{N}_i^1} g_{ij}^1(x_i, x_j) \{x_i - x_j\} - \dots - \sum_{j \in \mathcal{N}_i^m} g_{ij}^m(x_i, x_j) \{x_i - x_j\} \quad (3.36)$$

where: $\mathcal{N}_i^k, k = 1, 2, \dots, m$ are the neighborhoods defined such that $\mathcal{N}_i = \mathcal{N}_i^1 \cup \mathcal{N}_i^2 \cup \dots \mathcal{N}_i^m$. In general, under behavioral control, it might be desirable to directly affect the behavior of an agent towards one of its neighbors based on the information available to the agent not only from that individual neighbor but also from agents in their common neighborhood as well. In such a case, the g_{ij} functions- considering *Note 3.1*- can be written as follows:

$$g_{ij}^k(x_i, x_j, x_{l \in \mathcal{N}_i^k}) = \begin{cases} g_{ij}^k \geq 0, & j \in \mathcal{N}_i^k \\ 0, & j \notin \mathcal{N}_i^k \end{cases} \quad (3.37)$$

where: $g_{ij}^k: \mathcal{R}^{\mathcal{N}_i^k} \rightarrow \mathcal{R}_{\geq 0}$.

Remark 3.8: In the following, the g_{ij} functions are assumed to be given as $g_{ij}: \mathcal{R}^2 \rightarrow \mathcal{R}_{>0}$ unless otherwise explicitly stated.

Note 3.3: The stability of the nonlinear system (3.5) near the equilibria set \mathfrak{F} , given in (3.31), cannot be investigated using the linearized version (3.32) in which the Jacobian evaluated at the equilibria set is negative semi-definite with simple zero eigenvalue as forced by design and thus the stability test using linearization fails.

In a later section, the stability of (3.5) will be investigated using mainly LaSalle's invariance principle and the properties of M -matrices. If the necessary condition, i.e., $g_{ij} > 0, \forall i, j, \vec{x}(t)$, and the sufficient conditions (3.14) are met, then the protocols (3.3), (3.20), (3.24) and (3.28) can locally and asymptotically achieve their corresponding consensus types. Moreover, if the necessary condition (3.32) and the sufficient condition (3.34) are met, then the previous behaviors can be guaranteed globally for any initial conditions- within the domain of interest- under the previously stated state-dependent protocols.

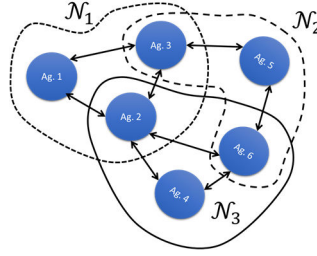


Figure 3.3. MAS with multiple neighborhoods.

Note 3.4: The proposed framework prohibits the state-dependent weights from altering the consensus value, obtained using \mathcal{L}_f , under undirected graphs with 0-1 weights.

Remark 3.9: Even though the nonlinear eigenvalues are variant when their associated nonlinear eigenvectors are multiplied by scalar, i.e., c , it was shown in [95] that as the origin being an equilibrium point, $\lambda(\vec{0})$ will be the same. Moreover, since c is a scalar independent of the states, the eigenvector $\vec{v} = \vec{1}c$ is still associated with $\lambda_1(\vec{x}) = 0$ of \mathcal{L}_s $\forall \vec{x}$ and it is the only left eigenvector associated with it due to the structure of the Jacobian imposed by the framework.

3.2.1.2: Average consensus value of MAS over state-dependent Laplacian matrix

The $\lambda_1 = 0$ eigenvalue and its associated left eigenvector are very important in evaluating the consensus value of the MAS over fixed directed and undirected communication graphs.

Fortunately, due to the proposed framework presented in this thesis, this will be also the case for state-dependent communication graphs.

In this subsection and the one that follows, it is desired to use the concept of eigenvalue and eigenvector to prove the motion-invariance behavior of the MAS under the previously designed protocols and to derive the consensus value achieved by the agents when time is significantly large. This is given by the following theorem.

Theorem 3.5: The initial conditions average value of N single-integrator MAS who is acting over connected undirected graphs and controlled using semi-linear state-dependent consensus protocols is time-invariant if and only if $\vec{1}^T$ is a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ and moreover, the average value is given as follows:

$$\mu_a = (\vec{1}^T \vec{x}(t_0))/N \quad (3.38)$$

Proof: Let $y = \vec{v}^T \vec{x}$ be the motion constant [9] of (5). So:

$$\frac{dy}{dt} = (\vec{v}^T \vec{x} + \vec{v}^T \dot{\vec{x}}) \quad (3.38.1)$$

Substituting (5) and (29) into (38.1), yields:

$$\frac{dy}{dt} = \left\{ \left(\lambda(\vec{x}) \vec{v}(\vec{x}) - \left(\frac{\partial \vec{f}(\vec{x})}{\partial x} \right)^T \vec{v} \right)^T - \vec{v}^T \mathcal{L}(\vec{x}) \right\} \vec{x}(t) = 0 \quad (3.38.2)$$

In state-dependent connected undirected graphs and balance digraphs, each row and column sum up to zero $\forall \vec{x}(t)$ considering (3.14), (3.34) and (3.35). Thus, $\lambda_1 = 0$ is always a simple eigenvalue and one eigenvector associated with it is $\vec{v}^T = c \vec{1}^T$ along which the solution is desired to evolve. So, (3.38.2) can be rewritten as follows:

$$\frac{dy}{dt} = c \vec{1}^T \left\{ -\frac{\partial \vec{f}(\vec{x})}{\partial x} \vec{x}(t) + \vec{f} \right\} = 0 \quad (3.38.3)$$

Which can be solved using the trivial solution $\vec{x}(t) = \vec{0}$, or at consensus $\vec{x}(t) = \vec{1}c, \forall t \geq t_{consensus}$. So, to keep the consensus value c time-invariant $\forall t \geq t_0$, one sufficient condition is given in (3.34) that ensures $\vec{1}^T$ to be a left eigenvector of \vec{f} associated with $\lambda_1 = 0$. Notice that forcing (3.35) is unnecessary. Thus, the value of c can be obtained as follows:

$$y(t_0) = \vec{v}^T(\vec{x}(t_0))\vec{x}(t_0) = c\vec{1}^T\vec{x}(t_0) \quad (3.38.4)$$

$$y(t_\infty) = \vec{v}^T(\vec{x}(t_\infty))\vec{x}(t_\infty) = \vec{1}^T\vec{1}c^2 = Nc^2 \quad (3.38.5)$$

Since $y(t_0) = y(t_\infty)$, solving for c using (3.38.4) and (38.5) yields:

$$c = (\vec{1}^T\vec{x}(t_0))/N \quad (3.38.6)$$

Which equals (3.38). Note that a connected undirected graph has $\lambda_2(\vec{x}) > 0 \forall \vec{x}$ and therefore $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ is simple. ■

Remark 3.10: Under the proposed protocol given by (3.3), the MAS connected via an undirected state-dependent graph has a rich behavior by which consensus value is reached at each time instant; being time-invariant. This behavior enables the agents to accommodate any temporary disturbances acting on a portion of the connected agents in a way that maintain the consensus value.

3.2.1.3: Other types of MAS consensus over state-dependent undirected graphs

The results obtained using protocol (3) can be extended to include other types of consensus-stated in section II- using a modified version of it given as follows:

$$u_i = -\alpha g(x_i) \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{x_i - x_j\} \quad (3.39)$$

where: $\alpha > 0$ is a real and fixed scalar that determines the speed of convergence and $g(x_i)$ is given as $1, x_i, x_i^2$ and x_i^{1-p} to achieve arithmetic, geometric, harmonic and the power mean-of-order- p , respectively. Note that α value will not affect the achieved consensus value.

Thus, the global dynamics of the MAS of N connected single-integrator dynamical systems under (39) can be given as follows:

$$\dot{\vec{x}}(t) = -\alpha G(\vec{x}) \mathcal{L}(\vec{x}) \vec{x}(t) \quad (3.40)$$

where: $G(\vec{x})$ is a positive definite matrix that embeds the consensus type desired and is given as follows:

$$G(\vec{x}) = \begin{bmatrix} g(x_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & g(x_N) \end{bmatrix} \quad (3.41)$$

Since $G(\vec{x})$ is a positive definite matrix- by design requirements explained previously-, it will not change the null space of $\mathcal{L}(\vec{x}) \vec{x} = \vec{f}(\vec{x})$ and its associated eigenvectors thus the same conditions given by (3.34) are sufficient to achieve all the previous types of consensus over a state-dependent undirected graph using the modified protocol (3.39). This is stated by the following Theorem.

Theorem 3.6: The MAS given by (3.40) acting over connected undirected graphs and controlled using semi-linear state-dependent consensus protocols has the arithmetic, geometric, harmonic and the power mean-of-order- p means of the agents initial conditions as time-invariant quantities if and only if $\vec{1}^T$ is a nonlinear left eigenvector of (3.5) associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ and $G(\vec{x})$ is chosen appropriately.

Proof: Consider the geometric mean given in (3.18), which can be rewritten as follows [100]:

$$\mu_g(t) = \exp(\vec{v}^T \vec{z}) \quad (3.42)$$

where \vec{v}^T is the normalized nonlinear left eigenvector of $\vec{f}(\vec{x})$ associated with the simple eigenvalue $\lambda(\vec{x}) = 0$ and $\vec{z} = [\ln(x_1), \ln(x_2), \dots, \ln(x_N)]^T$. The network protocol given in (20) can be used to achieve (3.42).

Taking the time-derivative of (3.42) and equate it to zero, one can show that- using (3.29) and (3.39) with $g_i(x_i) = x_i$ - the geometric mean is time-invariant *if and only if* the $\vec{v}^T = c\vec{1}^T$ is a left eigenvector of (3.5), i.e., $\mathcal{L}(\vec{x})\vec{x}(t) = \vec{f}(\vec{x})$, associated with $\lambda(\vec{x}) = 0$. This can be shown from the following:

$$\frac{d\mu_g}{dt} = \{\vec{v}^T \vec{z} + \vec{v}^T \vec{z}\} \exp(\vec{v}^T \vec{z}) = 0 \quad (3.42.1)$$

$$\frac{d\mu_g}{dt} = \vec{v}^T \left\{ -\frac{\partial \vec{f}(\vec{x})}{\partial x} \vec{z} + \vec{f} \right\} = 0 \quad (3.42.2)$$

Substituting $\vec{v} = c\vec{1}$ in (3.42) and evaluate it at t_0 and t_∞ , the consensus value c can be found as follows:

$$\mu_g(t_0) = \prod_{i=1}^N x_i^c(t_0) \quad (3.42.3)$$

$$\mu_g(t_\infty) = c^{Nc} \quad (3.42.4)$$

Since $\mu_g(t_0) = \mu_g(t_\infty)$, solving for c using (3.42.3) and (42.4) yields:

$$c = \sqrt[N]{\prod_{i=1}^N x_i(t_0)} \quad (3.42.5)$$

Which equals (3.18).

In general, let the weighted power, i.e., generalized, mean-of-order- p be given as follows

$\forall t \geq t_0, x_i(t) > 0 \quad \forall i$:

$$\mu_p(t) = \sqrt[p]{\frac{1}{N} \sum_{i=1}^N v_i x_i^p(t)} = \exp\left(\frac{\ln\left(\frac{\vec{v}^T \vec{z}}{N}\right)}{p}\right) \quad (3.42.6)$$

where: $\vec{v} = [v_1, \dots, v_N]^T$ and $\vec{z} = [x_1^p, \dots, x_N^p]^T$

Taking the time derivative of (3.42.6) and equate it to zero, yields:

$$\frac{d\mu_p}{dt} = N \frac{(\vec{v}^T \vec{z} + \vec{v}^T \dot{\vec{z}})}{p(\vec{v}^T \vec{z})} \mu_p(t) = 0 \quad (3.42.7)$$

Now, by substituting (3.1) and (3.39) into (3.42.7)- after choosing the desired type of consensus in $g(x_i)$ as shown previously-, yields:

$$\dot{\vec{z}} = p\vec{f} \quad (3.42.8)$$

where: \vec{f} is shown in (3.5). Thus, (3.42.7) can be given as:

$$\frac{d\mu_p}{dt} = \vec{v}^T \left\{ -\frac{\partial \vec{f}(\vec{x})}{\partial x} \vec{z} + p\vec{f} \right\} = 0 \quad (3.42.9)$$

Which simply indicates that \vec{v}^T , for the undirected graphs the $\vec{v}^T = c\vec{1}^T$, must be a left eigenvector of \vec{f} which agrees with (3.55). Note that a connected undirected graph has $\lambda_2(\vec{x}) > 0 \forall \vec{x}$ and therefore $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ is simple.

■

3.2.2: First Integral Design Approach

In this subsection, the second designing approach included in the proposed framework is presented. Recalling *Remark 3.5*, this method is considered less structured since it deals directly with the solution of the desired consensus value as being time-invariant, i.e., as a first integral, motion constant or conserved quantity, to facilitate designing the required protocols. Mainly, this method is suitable to deal with situations where energy of the

mechanical systems, for example, is said to be conserved. Mathematically, a first integral of a differential equation is a special function- say $I(\vec{x}): \mathfrak{R}^N \rightarrow \mathfrak{R}$ - once evaluated over the solution of that differential equation it proves to be constant. This last feature of the first integral function makes it suitable to be used as a designing tool as proposed in this framework in a way that we are forcing- rather than identifying- it as a first integral of mainly the N-identical single integrator MAS by choosing certain structures of the desired protocols, i.e., semi-linear or nonlinear, that ensures the problem to be solved in a distributed fashion. When the system of differential equations is of the first order, then a first integral is an algebraic relation that expresses the solution [101]. What was demonstrated earlier in this thesis starting with (3.6) up to (3.28) was a realization of the first integral designing method and the way it can be integrated with the nonlinear eigenvalue-based method under the proposed framework. However, in this subsection, the first integral design approach will be discussed.

In many applications, the first integral and its modified version, i.e., the first integral increment, play significant roles in investigating the behaviors of periodic and chaotic flows in 2-dimensional Hamiltonian systems with a time periodically perturbed vector field once the initial conditions are specified [102]. Also, the use of first integral in studying the behavior of discontinuous dynamical systems in different domains of the state space is evident such that when the first integral quantity is determined for each sub-domain, the flow in the corresponding sub-domains can be measured [103]. In the area of dynamical MAS on graphs, the use of the first integral is not new, see for example [104], and also is widely used, however in most cases referred to implicitly or explicitly as being a time-invariant [53] or a conserved [14] quantity.

In this thesis, a unifying framework that includes various results available in the literature with the ability to extend them further without much effort to study other types of behaviors is presented. Interestingly, the first integral design method can be easily used to deal with fixed, time-varying or state-dependent graph topologies as well as with the design of linear, semi-linear and nonlinear network protocols. For example, In [105] the first integral method was used to deal with non-autonomous cooperative systems. In this thesis, mainly the state-dependent graphs and the design of semi-linear and nonlinear network protocols will be handled. Other cases will be the subject of a future work.

To mathematically demonstrate the first integral design method, let the result given by (3.10) be rewritten as follows:

$$\nabla I(\vec{x}(t))[\dot{x}_1, \dot{x}_2, \dots, \dot{x}_N]^T = 0 \quad (3.43)$$

where: $\nabla I(\vec{x}(t)) = [1 \ 1 \ \dots \ 1]$ denotes the gradient of the desired first integral of the autonomous system $\dot{\vec{x}}(t) = \vec{f}(\vec{x})$.

Definition 3.4: [106] A first integral of an autonomous system is a real-valued function $I(\vec{x}(t))$ which is constant on solutions.

Definition 3.5: [106] A system that does have a nontrivial first integral is a conservative system.

Obviously, any constant function can be considered as a trivial first integral. **Definition 3.4** suggests that all solutions $\vec{x}(t)$ are constrained to move along the same *level set*- defined by the initial conditions- say of $I(\vec{x}(t)) = \mu_a(t_0) \ \forall t \geq t_0$. Due to the invariance property of the first integral, one can write the following using the chain rule:

$$\frac{d}{dt}I(\vec{x}(t)) = \nabla I(\vec{x}(t)) \cdot \dot{\vec{x}}(t) = L_{\vec{f}}\mu_a = 0 \quad (3.44)$$

where: $L_{\vec{f}}\mu_a$ is the Lie derivative, i.e., the directional derivative of $\mu_a(t)$ in the direction of the vector field $\vec{f}(\vec{x})$. $I(\vec{x})$ always exists in a neighborhood of a point where $\vec{f}(\vec{x})$ is Lipschitz. Moreover, if \vec{f} is globally Lipschitz, then $I(\vec{x})$ will be unique and global as well. It should be noted that the first integrals form a subset of the so-called constants of motion; since the former do not depend explicitly on time [107].

3.2.3: Relation between first integral and nonlinear eigenvalue approaches

In general, the nonlinear eigenvalue and eigenvector concepts presented in this thesis are special cases of their general concepts widely used in various applications like nonlinear accessibility/observability of nonlinear systems [108] or solving the differential Riccati equation [109]. Whenever the Jacobian matrix is used in the nonlinear eigenvalue problems, as in (3.29) and (3.30) for example, then they relate to invariant co-distributions [95]. Using the first integral design approach to achieve a desired behavior- namely: mean consensus, it is possible to design the network protocols and therefore the structure of the nonlinear mapping \vec{f} appearing for instance in (3.5). Doing so, the nonlinear eigenvalue approach can be used such that the elements of \vec{f} are customized; to ensure that $\lambda_1(\vec{x}) = 0$ always corresponds to the desired constant of motion and thus the solution $\vec{x}(t)$ evolves as desired along $\vec{v}^T = \vec{1}^T$. The same thing can be found in quantum mechanics, for example, where the eigenvalue equations help in finding the states corresponding to the constants of motion.

One major implication of (3.33.1), if satisfied, is that the consensus type is guaranteed to be achieved globally irrespective of any non-persistent disturbance, thus the notion of *robust behavior* can be used. While depending only on (3.44) to achieve consensus may not be sufficient because even a non-persistent disturbance can force the MAS, or at least

a portion of it, to leave the subset Ω formed by the initial conditions. In other words, (3.33.1) guarantees that the constant of motion is globally associated with $\lambda_1(\vec{x}) = 0$ while (3.44) only guarantees that near the equilibria set. Figure 3.4 shows an example of three agents willing to come to consensus in the 3D space over $\Omega \subseteq \mathbb{R}^N$.

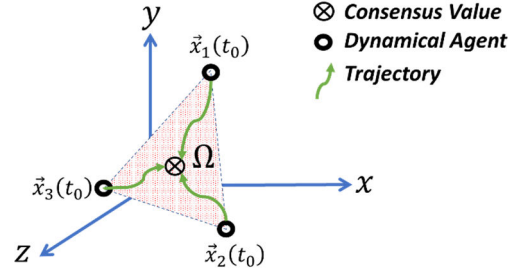


Figure 3.4. First Integral subset formed by the initial conditions of three dynamical agents in 3D space.

Next, the use of the proposed framework to design nonlinear network protocols is demonstrated.

3.2.4: Nonlinear Distributed Consensus Protocols over undirected graphs using the proposed framework

Here, additional nonlinear distributed protocols will be derived. Unlike (3.3), which is a semi-linear protocol, the newly derived protocols will allow for completely nonlinear relations between the states used. Notice that in (3.3) states are used- multiplied by g_{ij} functions- in a linear fashion to facilitate building the state-dependent Laplacian matrix as was shown for example in (3.5). This simply suggests that only the edges weights of the communication graph are, in general, nonlinearly dependent on the states as explained before. However, this should not be the case anymore.

By and large, nonlinear protocols are important due to the true nature of applications that requires nonlinear relationships among parameters and variables. Moreover, nonlinear protocols can be used to facilitate rich behaviors that cannot be- in many cases- achieved using their linear counterparts. Dealing with bounded inputs and generating dynamical

friendly signals might inspire the use of such protocols. In the state-dependent semi-linear network protocols, the concept of Laplacian matrix is preserved, thanks to the diffusive-coupling used, which is not the case in the nonlinear version of these protocols. Therefore, special attention must be taken when dealing with this version of protocols. Next, some major classes of such protocols are suggested. The main goal is to investigate the possibility of achieving different types of consensus using general classes of nonlinear protocols and to show the necessary and sufficient conditions when applicable.

Remark 3.11: Note that the edges weights can be chosen to be state-dependent either separate from or totally fused within the nonlinear functions used. From analysis point of view, the choice made will affect mainly the proofs methodologies.

3.2.4.1: Class 1: Consider the nonlinear consensus protocol given as follows:

$$\dot{x}_i = u_i = \alpha \sum_{j \in \mathcal{N}_i} \varphi_{ij}(x_i, x_j) = f_i \quad (3.45)$$

where: $\alpha > 0$ is a real and fixed scalar that determines the speed of convergence. $\varphi_{ij}(0) = 0$. Such a protocol was used in [110] and [52]. By substituting (3.45) into (3.44), we may write the following:

$$\sum_{j \in \mathcal{N}_1} \varphi_{1j}(x_1, x_j) + \dots + \sum_{j \in \mathcal{N}_N} \varphi_{Nj}(x_N, x_j) = \vec{1}^T \vec{f} = 0 \quad (3.46)$$

where: $\vec{f} = [f_1, f_2, \dots, f_N]^T$ with f_i given by (3.45). Usually, such equations can be solved using physics, educated guessing or luck [106]. Assuming an undirected communication graph, one candidate solution may be characterized as follows, $\forall i, j \in \mathcal{N}_i, \vec{x}(t) \in \Psi \subseteq \mathbb{R}^N$:

$$\varphi_{ij}(x_i, x_j) = -\varphi_{ji}(x_j, x_i), \quad (3.47)$$

which means that the functions used must be anti-symmetric. This is in agreement with what was proposed in [94].

At this point, it is important to notice that either the function used, i.e., φ_{ij} , is selected based on the application or the desired behavior and then Ψ follows or the function used is designed such that it is defined for a specific domain Ψ of interest. In some cases, both conditions must be satisfied as found in [56], for example.

Using the nonlinear eigenvalue-based approach, the necessary conditions needed for (3.45) to achieve the average consensus over Ψ can be given as follows- obtained by solving (3.33.1):

$$\frac{\partial \varphi_{ij}(x_i, x_j)}{\partial x_i} = -\frac{\partial \varphi_{ji}(x_j, x_i)}{\partial x_i}, \quad \forall i, j \in \mathcal{N}_i, \vec{x}(t) \in \Psi \quad (3.48)$$

If the nonlinear right eigenvector $\vec{e} = \vec{1}$ is to be forced as well, then the following conditions must be satisfied- obtained by solving (3.33.2):

$$\frac{\partial \varphi_{ij}(x_i, x_j)}{\partial x_i} = -\frac{\partial \varphi_{ij}(x_i, x_j)}{\partial x_j}, \quad \forall i, j \in \mathcal{N}_i, \vec{x}(t) \in \Psi \quad (3.49)$$

By making use of the following lemma, other types of mean consensus can be achieved using (3.45).

Lemma 3.1: The following Class 1 nonlinear protocol can be used to achieve various types of consensus over connected undirected graphs if and only if $\vec{1}^T$ is a nonlinear left eigenvector of (3.46) associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ and $G(\vec{x})$ is chosen appropriately.

$$u_i = \alpha g(x_i) \sum_{j \in \mathcal{N}_i} \varphi_{ij}(x_i, x_j) \quad (3.50)$$

Proof: See (3.42.1) -(3.42.9) with \vec{f} given as in (3.46).

■

As an example, consider the following entropic protocol that was used to achieve the geometric mean [56]:

$$u_i = -x_i \sum_{j \in \mathcal{N}_i} w_{ij} \ln\left(\frac{x_i}{x_j}\right) \quad (3.51)$$

$$\forall i, j \in \mathcal{N}_i, \vec{x}(t_0) \in \mathbb{R}_{>0}^N, \vec{x}(t) \in \Omega \subseteq \mathbb{R}_{>0}^N$$

where: $w_{ij} = w_{ji}$ is the weight of the edge connecting the i^{th} and j^{th} agents- assumed to be a 0-1 weight- and $\ln(\cdot)$ is the natural logarithm function.

3.2.4.2: Class 2: Consider the nonlinear consensus protocol given as follows:

$$\dot{x}_i = u_i = -\alpha \sum_{j \in \mathcal{N}_i} \{\theta(x_i) - \theta(x_j)\} = f_i \quad (3.52)$$

where: $\alpha > 0$ is a real and fixed scalar that determines the speed of convergence. $\theta(0) = 0$. Such a protocol was used in [52] and [51].

Protocol (3.52) deals with a 0-1 weight for an edge explicitly by depending on \mathcal{N}_i . Now, the derivation of the previous consensus types for a MAS of single-integrator agents connected using (3.52) over mainly an undirected graph is presented.

➤ Different Types of Mean Consensus

Using the first integral approach, it is easy to show the following:

$$\sum_{j \in \mathcal{N}_1} \{\theta(x_1) - \theta(x_j)\} + \dots + \sum_{j \in \mathcal{N}_N} \{\theta(x_N) - \theta(x_j)\} = \vec{1}^T \vec{f} = 0 \quad (3.53)$$

which is valid for a general $\theta(x_i)$ function. However, the function structure must ensure the stability of the MAS. The same result about $\theta(x_i)$ can be obtained by solving (3.33.1) which indicates that the nonlinear left eigenvector associated with $\lambda(\vec{x}) = 0$, i.e., $\vec{v}^T = c\vec{1}^T$, is forced which is not the case for the right eigenvector unless $\theta(x_i) = x_i$ resulting in

the standard protocol used with fixed Laplacian matrices, in general. One example of (3.52) can be found in [56] and is shown here as follows:

$$u_i = \sum_{j \in \mathcal{N}_i} w_{ij} \left(\ln(x_j) - \ln(x_i) \right), \quad \forall i, j, \vec{x}(t) \in \mathfrak{R}_{>0}^N \quad (3.54)$$

which is used to achieve average consensus over an undirected graph. Multiplying and dividing by $(x_j - x_i)$ yields a $g_{ij} = \left(\ln(x_j) - \ln(x_i) \right) / (x_j - x_i), \vec{x} \neq \vec{0} \forall t \geq t_0$ function equal to the inverse of the logarithmic mean which is symmetric and positive in both x_i and x_j and therefore $\vec{v}^T = c \vec{1}^T$ can be forced as a left eigenvector- but not the right eigenvector $\vec{e} = \vec{1}$ - of the resulting $\mathcal{L}(\vec{x})$ in the semi-linear protocol. This simple transformation can be used in most classes of nonlinear protocols presented herein which makes them special cases of the semi-linear protocols in terms of their design and stability analysis. Another way to write (3.52) in a semi-linear form is given as follows:

$$\dot{\vec{x}}(t) = -\alpha \mathcal{L}_f \vec{\theta}(\vec{x}) = \vec{f}(\vec{x}) \quad (3.55)$$

Such a model was presented in [14]. Other types of consensus under (3.52) are guaranteed by the following lemma.

Lemma 3.2: The following Class 2 nonlinear protocol can be used to achieve various types of consensus over connected undirected graphs if and only if $g(x_i)$ is chosen appropriately.

$$u_i = -\alpha g(x_i) \sum_{j \in \mathcal{N}_i} \{ \theta(x_i) - \theta(x_j) \}, \quad \forall i, j, \vec{x}(t) \in \Omega \subseteq \mathfrak{R}^N \quad (3.56)$$

Proof: Let $\vec{f} = [u_1, \dots, u_N]^T$ where u_i is given in (3.52). Clearly, $\vec{1}^T$ is always a left eigenvector of \vec{f} . So, the rest of the proof follows as in (3.42). ■

From (3.56), it is obvious that both (3.51) and (3.54) belong to the same class, namely: Class 2, which is a general structure usually arises in chemical reactions, see [111] for example.

3.2.4.3: Class 3: Consider the nonlinear consensus protocol given as follows:

$$u_i = \alpha \sum_{j \in \mathcal{N}_i} \varphi_{ij} \left(\pi(x_j) - \pi(x_i) \right) \quad (3.57)$$

where: $\alpha > 0$ is a real and fixed scalar that determines the speed of convergence and $\varphi_{ij}(0) = 0$. Such a protocol was used in [94] where $\pi(x_i): \mathfrak{R} \rightarrow \mathfrak{R}$ is differentiable and $d\pi(x_i)/dx_i$ is locally Lipschitz and strictly positive.

A similar version of (3.57) was used in [53] with $\pi(x_i) = x_i$. Now, by using the first integral method, the sufficient conditions for (3.57) to be time-invariant over $\Omega \subseteq \mathfrak{R}^N$ will be given by algebraically solving:

$$\sum_{j \in \mathcal{N}_1} \varphi_{1j} \left(\pi(x_j) - \pi(x_1) \right) + \cdots + \sum_{j \in \mathcal{N}_N} \varphi_{Nj} \left(\pi(x_j) - \pi(x_N) \right) = 0 \quad (3.58)$$

Assuming a 0-1 weighted undirected graph, (3.57) will be a candidate solution if the following conditions are satisfied:

$$\varphi_{ij} \left(\pi(x_j) - \pi(x_i) \right) = -\varphi_{ji} \left(\pi(x_i) - \pi(x_j) \right) \quad (3.59)$$

$$\forall i, j \in \mathcal{N}_i, \vec{x}(t) \in \Omega$$

where φ_{ij} must be odd, continuous, locally Lipschitz, and *strictly increasing* [94]. One widely used example of (3.57) over the open interval $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ is [56]:

$$u_i = \sum_{j \in \mathcal{N}_i} \sin(x_j - x_i) \quad (3.60)$$

The nonlinear protocol (3.60) can be used for phase averaging. Using the nonlinear eigenvalue-based design approach, the needed conditions to force the $\vec{1}$ as a nonlinear left eigenvector associated with $\lambda(\vec{x}) = 0$ are listed as follows- irrespective of $\pi(\cdot) \neq 0$:

$$\frac{\partial \varphi_{ij}(x_i, x_j)}{\partial x_i} = -\frac{\partial \varphi_{ji}(x_i, x_j)}{\partial x_i}, \quad \forall i, j \in \mathcal{N}_i, \vec{x}(t) \in \Omega \quad (3.61)$$

Moreover, to force the $\vec{1}$ as a nonlinear right eigenvector associated with $\lambda(\vec{x}) = 0$, the following conditions are needed when solving (3.33.2) $\forall i, j \in \mathcal{N}_i, \vec{x}(t) \in \mathfrak{R}^N$:

$$\frac{d\pi(x_i)}{dx_i} \cdot \frac{\partial \varphi_{ij}(x_i, x_j)}{\partial x_i} = \frac{d\pi(x_j)}{dx_j} \cdot \frac{\partial \varphi_{ij}(x_i, x_j)}{\partial x_j} \quad (3.62)$$

Noticeably, under (3.60) the resulting Jacobian is symmetric and acts as a negative singular M -matrix over $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ which will ensure the stability of the resulting MAS as will be explained later.

A generalized version of (3.57) to achieve several types of consensus is given by the following lemma.

Lemma 3.3: The following Class 3 nonlinear protocol can be used to achieve various types of consensus over connected undirected graphs if and only if $\vec{1}^T$ is a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ of the dynamics $\vec{f} = [u_1, \dots, u_N]^T$ where u_i is given in (3.57), and $g(x_i)$ is chosen appropriately.

$$u_i = \alpha g(x_i) \sum_{j \in \mathcal{N}_i} \varphi_{ij}(\pi(x_j) - \pi(x_i)) \quad (3.63)$$

Proof: Which can be used if and only if the $\vec{1}^T$ is a left eigenvector of $\vec{f} = [u_1, \dots, u_N]^T$ associated with $\lambda(\vec{x}) = 0$ where u_i is given in (3.57). This can be shown by following the same steps taken to proof (3.42) earlier.

■

3.2.4.4: Class 4: Consider the nonlinear consensus protocol given as follows:

$$u_i = \alpha g(x_i) \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \cdot \varphi_{ij}(x_i, x_j) \quad (3.64)$$

where: $\alpha > 0$ is a real and fixed scalar that determines the speed of convergence and $\varphi_{ij}(0) = 0$. It is easy now to show that (3.64) can be used to achieve several types of consensus discussed previously, using the first integral method and solving (3.33.1), *if and only if* the following conditions are met:

$$\varphi_{ij}(x_i, x_j) = -\varphi_{ji}(x_j, x_i), \quad \forall i, j \in \mathcal{N}_i, \vec{x}(t) \in \Omega \subseteq \mathbb{R}^N \quad (3.65)$$

$$g_{ij}(x_i, x_j) = g_{ji}(x_j, x_i), \quad \forall i, j \in \mathcal{N}_i, \vec{x}(t) \in \Omega \subseteq \mathbb{R}^N \quad (3.66)$$

Obviously, (3.3) is a special case of (3.64) where $\varphi_{ij}(x_i, x_j) = x_j - x_i$. The consensus protocol (3.64) combines state-dependent weights of the communication links as well as a general nonlinear update of the states.

To extend these results of, mainly, the arithmetic average consensus under (3.64) to the whole $\Psi \subseteq \mathbb{R}^N$ space, the nonlinear eigenvalue-based design method is used. The following simple example shows the derivation steps in more detail.

Example 3.1: Assume an undirected link exists among two single-integrator agents as shown in Figure 3.5. The overall dynamics of the MAS can be written as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} g_{12}(x_1, x_2) \cdot \varphi_{12}(x_1, x_2) \\ g_{21}(x_2, x_1) \cdot \varphi_{21}(x_2, x_1) \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \vec{f} \quad (3.67)$$

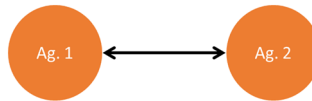


Figure 3.5. two single-integrator agents interacting over an undirected link.

Calculating the Jacobian of \vec{f} in (3.67), yields:

$$\frac{\partial f_1}{\partial x_1} = \frac{\partial g_{12}}{\partial x_1} \cdot \varphi_{12} + g_{12} \cdot \frac{\partial \varphi_{12}}{\partial x_1} \quad (3.67.1)$$

$$\frac{\partial f_1}{\partial x_2} = \frac{\partial g_{12}}{\partial x_2} \cdot \varphi_{12} + g_{12} \cdot \frac{\partial \varphi_{12}}{\partial x_2} \quad (3.67.2)$$

$$\frac{\partial f_2}{\partial x_1} = \frac{\partial g_{21}}{\partial x_1} \cdot \varphi_{21} + g_{21} \cdot \frac{\partial \varphi_{21}}{\partial x_1} \quad (3.67.3)$$

$$\frac{\partial f_2}{\partial x_2} = \frac{\partial g_{21}}{\partial x_2} \cdot \varphi_{21} + g_{21} \cdot \frac{\partial \varphi_{21}}{\partial x_2} \quad (3.67.4)$$

Having evaluated the Jacobian of \vec{f} , the $\vec{1}$ can be forced as a left/right eigenvector of (3.67) associated with $\lambda(\vec{x}) = 0$. This is again to maintain the relation between the nonlinear function and the Laplacian matrix usually used in MAS design. To do so, both the row-sum and column-sum of each row and column in the Jacobian must equate to *exactly* zero.

Remark 3.12: Equating both the row-sum and column-sum of each row and column in the Jacobian to *exactly* zero is possible in single-integrator MAS because there are no actual dynamics associated with the agents. In other words, the agents can accelerate in each direction instantaneously. This will not be the case once real systems are examined, however the resulting protocols herein can be used for coordination [50]. A modified version of the proposed framework will be covered in later to deal with more realistic models of dynamical systems. Later in Chapter 5, an example that demonstrate the use of mainly the first integral method to deal with MAS consisting of N double-integrator systems will be presented where the angular momentum is used as the constant of motion. Doing the Math only for the first row of the Jacobian, i.e., by solving (3.33.2), and column, i.e., by solving (3.33.1), yields the following:

$$\frac{\partial f_1}{\partial x_1} + \frac{\partial f_1}{\partial x_2} = \varphi_{12} \cdot \left\{ \frac{\partial g_{12}}{\partial x_1} + \frac{\partial g_{12}}{\partial x_2} \right\} + g_{12} \cdot \left\{ \frac{\partial \varphi_{12}}{\partial x_1} + \frac{\partial \varphi_{12}}{\partial x_2} \right\} = 0 \quad (3.67.5)$$

$$\frac{\partial f_1}{\partial x_1} + \frac{\partial f_2}{\partial x_1} = \left\{ \frac{\partial g_{12}}{\partial x_1} \cdot \varphi_{12} + \frac{\partial g_{21}}{\partial x_1} \cdot \varphi_{21} \right\} + \left\{ g_{12} \cdot \frac{\partial \varphi_{12}}{\partial x_1} + g_{21} \cdot \frac{\partial \varphi_{21}}{\partial x_1} \right\} = 0 \quad (3.67.6)$$

From (3.67.5), a possible solution that yields a symmetric Jacobian can be given when:

$$\frac{\partial g_{12}}{\partial x_1} = -\frac{\partial g_{12}}{\partial x_2}, \quad \forall \vec{x}(t) \in \mathfrak{R}^2 \quad (3.67.7)$$

$$\frac{\partial \varphi_{12}}{\partial x_1} = -\frac{\partial \varphi_{12}}{\partial x_2}, \quad \forall \vec{x}(t) \in \mathfrak{R}^2 \quad (3.67.8)$$

As for (3.67.6), we may write a generalized solution as follows:

$$g_{12} = g_{21}, \quad \forall \vec{x}(t) \in \mathfrak{R}^2 \quad (3.67.9)$$

$$\frac{\partial \varphi_{12}}{\partial x_1} = -\frac{\partial \varphi_{21}}{\partial x_1}, \quad \forall \vec{x}(t) \in \mathfrak{R}^2 \quad (3.67.10)$$

and either:

$$\frac{\partial g_{12}}{\partial x_1} = -\frac{\partial g_{21}}{\partial x_1}, \quad \forall \vec{x}(t) \in \mathfrak{R}^2 \quad (3.67.11)$$

$$\varphi_{12} = \varphi_{21}, \quad \forall \vec{x}(t) \in \mathfrak{R}^2 \quad (3.67.12)$$

or:

$$\frac{\partial g_{12}}{\partial x_1} = \frac{\partial g_{21}}{\partial x_1}, \quad \forall \vec{x}(t) \in \mathfrak{R}^2 \quad (3.67.13)$$

$$\varphi_{12} = -\varphi_{21}, \quad \forall \vec{x}(t) \in \mathfrak{R}^2 \quad (3.67.14)$$

In case that the state-update function, i.e., φ_{12} , is bounded to certain domain $\Omega \subseteq \Psi$, we may rewrite the previous conditions, (3.67.7) – (3.67.14), by simply replacing \mathfrak{R}^2 with $\Psi \subseteq \mathfrak{R}^2$.

Conditions (3.67.13) and (3.67.14) can be used along with (3.67.9) and (3.67.10) to extend the use of (3.60) to include state-dependent weights under the proposed framework and it is given as follows:

$$u_i = \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \sin(x_j - x_i), \quad \forall i, j, \vec{x}(t) \in \Psi \quad (3.67.15)$$

where $\Psi: \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. □

3.2.4.5: Class 5: Consider the nonlinear consensus protocol given as follows:

$$u_i = \varpi \phi \left(\alpha \sum_{j \in \mathcal{N}_i} w_{ij} (x_j - x_i) \right) \quad (3.68)$$

where: $\alpha, \varpi > 0$ are real and fixed scalars that determine the speed of convergence, $\phi(\vec{0}) = 0$ and $w_{ij} = w_{ji}$ is the weight of the edge connecting the i^{th} and j^{th} agents- assumed to be a 0-1 weight.

This class may represent very useful functions where the used inputs are bounded to certain values for instance. Consider the case when $\phi = \tanh(\cdot)$, i.e., the hyperbolic function, then ϖ can represent the magnitude of the bounded input assuming its bound to be symmetric, i.e., $\pm\varpi$. The $\tanh(\cdot)$ function is usually used as a smooth approximation of the $\text{sign}(\cdot)$ function to get rid of the chattering in the control signal usually available in sliding-mode controllers, in general. Interestingly, if α is very large, then the resulting behavior of (3.68) is equivalent to a bang-bang controller. On the contrary, if it is too small, then the convergence will be too slow, however the control signal will be bounded. Therefore, α should be somehow chosen in an appropriate way which is beyond the scope of this investigation. Using $\phi = \tanh(\cdot)$, (3.68) can be rewritten as follows:

$$u_i = \varpi \tanh \left(\alpha \sum_{j \in \mathcal{N}_i} w_{ij} (x_j - x_i) \right) = f_i \quad (3.69)$$

It is straightforward to show that the Jacobian (J) of (3.69) is given as follows:

$$J = -\alpha \varpi \operatorname{diag}([sech^2(q_1), \dots, sech^2(q_N)]) \mathcal{L}_f \leq 0 \quad (3.70)$$

where: $q_i = \alpha \sum_{j \in \mathcal{N}_i} w_{ij} (x_j - x_i)$, $\forall i = 1, 2, \dots, N$.

Solving (3.33.1) and (3.33.2) using the Jacobian given in (3.70), it is evident that $\vec{e} = \vec{1}$ is a nonlinear right eigenvector associated with $\lambda_1(\vec{x}) = 0$ and not $\vec{v}^T = c \vec{1}^T$, therefore only consensus is possible but not average consensus. The invariant quantity associated with $\lambda(\vec{x}) = 0$ can be found by solving (3.44). Obviously, the average mean- given in (3.8)- is not a constant of motion and therefore $c \neq \mu_a$. To find the value of c , we need to *identify* the motion of constant, i.e., $I(\vec{x})$, so (3.44) can be used to formulate the following needed partial differential equation:

$$\frac{\partial I}{\partial x_1} \tanh(q_1) + \dots + \frac{\partial I}{\partial x_N} \tanh(q_N) = 0 \quad (3.71)$$

whose solution is not attempted in this thesis. Note, however, that once $\alpha \rightarrow 0$, then $sech^2(q_1) \rightarrow 1$ in (3.70), and therefore $J \rightarrow -\alpha \varpi \mathcal{L}_f$ and the average consensus can almost be achieved. Interestingly, the time to consensus is affected by ϖ which helps addressing the finite-time consensus problem as will be shown in a coming section. Another approach to address the average consensus problem with bounded input will be presented later as well.

Remark 3.13: In what proceeded Class 5, all the protocols – both semi-linear and nonlinear- were designed based on the desired constant of motion and afterwards the nonlinear eigenvalue problems were solved to make sure that this constant of motion is

globally associated with $\lambda_1(\vec{x}) = 0$. In Class 5, we have reversed that order deliberately to emphasize the fact that the two design steps in the proposed framework can be used separately based on the designated goals set by the designer.

3.2.4.6: Class 6: Consider the nonlinear consensus protocol given as follows:

$$u_i = -\phi_i\left(\sum_{j \in \mathcal{N}_i} g_{ij}\right)x_i + \sum_{j \in \mathcal{N}_i} g_{ij}x_j \quad (3.72)$$

where: $\phi_i(\vec{0}) = 0$ and \mathcal{N}_i is the i^{th} agent in-neighborhood designed originally by the designer through \mathcal{L}_f . It is assumed that the $g_{ij} \in (0,1] \forall i, j$ otherwise the problem becomes more complicated and Class 6 cannot stand in its current form and a modification is needed. If $-\phi_i(\sum_{j \in \mathcal{N}_i} g_{ij}) = -\sum_{j \in \mathcal{N}_i} g_{ij}$, then the average consensus is achieved; since both (3.33.1) and (3.33.2) are satisfied. In general, for (3.33.2) to be satisfied, the following relation must hold- using (3.35):

$$-\phi_i\left(\sum_{j \in \mathcal{N}_i} g_{ij}\right) + \sum_{j \in \mathcal{N}_i} g_{ij} = 0, \quad \forall \vec{x}(t) \in \Omega \subseteq \mathbb{R}^N, \forall i = 1, 2, \dots, N \quad (3.73)$$

Solving (3.33.1) requires much involved conditions which are not reported herein. Captivatingly, the network protocol (3.72) can be used to control the behavior of the i^{th} spatial agent according to its neighborhood in a special way. Say for example, if it is fully or partially isolated from its neighborhood, then it can individually move towards a specific point in the phase space- usually stationary and determined by the designer (*a rendezvous point* o_r) the origin in our case- to achieve what may be called a trivial consensus [112], [113] which give a chance for agents to dynamically establish connections that allow them to reach a non-trivial consensus cooperatively later on. For example, consider the case

where the g_{ij} functions are used to model the communication signal strength between the i^{th} agent and its in-neighbors. Therefore, $\phi(\cdot)$ can be given as follows:

$$\phi_i(z) = \begin{cases} \tau_i, & z < th_i \\ z, & z \geq th_i \end{cases}, \quad \forall i = 1, 2, \dots, N \quad (3.74)$$

where: τ_i and th_i are fixed, real and positive scalars denoting the self-centered convergence rate and the signal strength threshold- specified by the designer- based on which agents decide to be self-centered or cooperative. Based on (3.74), the condition (3.73) can be met whenever $\sum_{j \in \mathcal{N}_i} g_{ij} \geq th_i$ which indicates that the i^{th} agent can establish reliable connections among its neighbors. Should the i^{th} agent select its neighbors individually based on their reputation, power level, signal strength or the like, then (3.72) can be modified- considering (3.74)- such that each link is assessed separately. Therefore, a modified version of (3.72) can be given- for example- as follows:

$$u_i = -\phi_i \left(\sum_{j \in \mathcal{N}_i} \delta_i(g_{ij}) \right) x_i + \sum_{j \in \mathcal{N}_i} \delta_i(g_{ij}) x_j \quad (3.75)$$

where:

$$\delta_i(z) = \begin{cases} 0, & z < TH_i \\ z, & z \geq TH_i \end{cases} \quad (3.75.1)$$

and $TH_i \leq th_i/|\mathcal{N}_i|$ is a fixed, real and positive scalar reflecting the desired connection quality modelled by the g_{ij} functions and $|\mathcal{N}_i|$ is the i^{th} agent in-neighbors number as imposed by \mathcal{L}_f . In general, only some agents within the MAS may be successful to establish connections with each other when approaching the meeting point while the others are still on their ways. The stability of (3.72) can be guaranteed outside Ω_{nt} -which is an attracting positively invariant set or a trapping region in Ω - by making the gradient of the dynamical system nonpositive. To ensure this, we must have:

$$\tau_c \geq \max(d_{i,f}) \geq \tau_i \geq d_{i,f} \quad (3.75.2)$$

to make the matrices diagonally dominant where τ_c is the common convergence rate and $d_{i,f}$ is the in-degree of the i^{th} agent imposed by the designer through \mathcal{L}_f . Notice that the trajectories of the dynamical system starting outside Ω_{nt} will hit its boundary, i.e., $\partial\Omega_{nt}$, at the same time when (3.73) is satisfied. This moment (t_{impact}), in fact, is the birth moment of this invariant set. Being invariant with also a nonpositive Jacobian, the trajectories of the system after switching from Ω_t into Ω_{nt} will evolve inside Ω_{nt} only $\forall t \geq t_{impact}$. Thus, the switching is happening once, and the Zeno phenomenon will not occur.

Note 3.5: The network protocol (3.3) is a special case of (3.72) when $\phi_i(\sum_{j \in \mathcal{N}_i} g_{ij}) = \sum_{j \in \mathcal{N}_i} g_{ij}$. The main difference between them is that (3.3) deals with the whole universal and connected domain of interest ($\Omega \subseteq \mathbb{R}^N$) to achieve the non-trivial (Ω_{nt}) consensus while (3.72) splits Ω into two accessible sub-domains, namely: trivial (Ω_t) and non-trivial, where the flow of MAS dynamics switches while enjoying time-continuity. For simplicity, a 2-D illustration is depicted in Figure 3.6. Note also that even if the same initial conditions, i.e., $\vec{x}(t_0)$, are used under both protocols the consensus values might be different unless $\vec{x}(t_0)$ originally belong to Ω_{nt} . The usefulness of both (3.72) and (3.75) will be emphasized in Chapter 4.

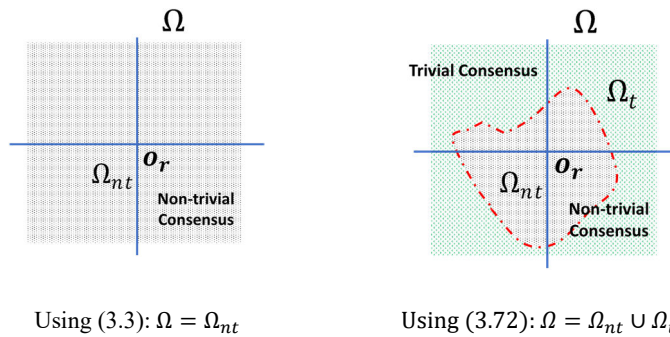


Figure 3.6. Effect of using protocols (3.3) and (3.72) on the domain of interest.

3.2.5: Importance of the nonlinear right eigenvector

As was shown in (3.38.3), forcing the $\vec{1}$ as a right vector associated with $\lambda_1 = 0$ of the nonlinear function (3.5) under network protocol (3.3) is not necessary. This is due the fact that the semi-linear structure of (3.5) claims a symmetric structure owing to the underlying undirected communication graph used. In general, this is not true especially with nonlinear protocols whose Jacobian may not be symmetric $\forall \vec{x}(t) \in \mathfrak{R}^N$, consider (3.54) as an example. The unnecessary of forcing $\vec{1}$ as a nonlinear right eigenvector associated with $\lambda_1 = 0$ is demonstrated as follows:

Let the weighted power, i.e., generalized, mean-of-order- p be given as follows $\forall t \geq t_0, x_i(t) > 0 \ \forall i$:

$$\mu_p(t) = \sqrt[p]{\frac{1}{N} \sum_{i=1}^N v_i x_i^p} = \exp\left(\frac{\ln\left(\frac{\vec{v}^T \vec{z} + \vec{z}^T \vec{v}}{2N}\right)}{p}\right) \quad (3.76)$$

where: $\vec{v} = [v_1, \dots, v_N]^T$ and $\vec{z} = [x_1^p, \dots, x_N^p]^T$.

Taking the time derivative of (3.76) and equate it to zero, yields:

$$\frac{d\mu_p}{dt} = N \frac{(\vec{v}^T \vec{z} + \vec{v}^T \vec{z} + \vec{z}^T \vec{v} + \vec{z}^T \vec{v})}{p(\vec{v}^T \vec{z})} \mu_p(t) = 0 \quad (3.77)$$

Now, by substituting (3.1) and any previous network protocol into (3.77)- after choosing any type of consensus in $g(x_i)$ as shown before-, yields a similar equation to (3.50.4).

Thus, (3.77) can be given as:

$$p\{\vec{v}^T \vec{f} + f^T \vec{v}\} - \left\{ \vec{z}^T \left(\frac{\partial \vec{f}(\vec{x})}{\partial x} \right)^T \vec{v} + \vec{v}^T \frac{\partial \vec{f}(\vec{x})}{\partial x} \vec{z} \right\} = 0 \quad (3.78)$$

Obviously, it is sufficient to have $\vec{v}^T = c\vec{1}^T$ as a left eigenvector of \vec{f} associated with $\lambda_1(\vec{x}) = 0$ to equate both brackets to zero. Thus, forcing $\vec{e} = \vec{1}$ as a right eigenvector is not needed; because the consensus value does depend on the left null space, i.e., the columns space, and not on the null space, i.e., rows space. This will be highlighted shortly.

Remark 3.14: Note that the Jacobian structure is affected by the underlying communication graph and the network protocol used especially if it is nonlinear. Thus, having undirected graph is not sufficient for the Jacobian to be symmetric, in general. The advantage of having a symmetric Jacobian is demonstrated by the following example.

Example 3.2: To understand Remark 3.12, consider for example the Jacobian associated with (3.54) connecting three agents over an undirected graph, as shown in Figure 3.7, and is given as follows:

$$\frac{\partial \vec{f}(\vec{x})}{\partial x} = - \begin{bmatrix} \frac{2}{x_1} & \frac{-1}{x_2} & \frac{-1}{x_3} \\ \frac{-1}{x_1} & \frac{2}{x_2} & \frac{-1}{x_3} \\ \frac{-1}{x_1} & \frac{-1}{x_2} & \frac{2}{x_3} \end{bmatrix} \quad (3.79)$$

Which is asymmetric and have $\vec{1}^T$ as a left but not as a right eigenvector $\forall \vec{x}(t) \in \mathfrak{R}_{>1}^N$ except at the equilibrium point $\vec{x}_{eq} = [1 \ 1 \ 1]^T$ at which the Jacobian will be equal to the Laplacian matrix \mathcal{L}_f . The existence of $\lambda_1 = 0$ is guaranteed by having $c\vec{1}^T$ as a left eigenvector or right eigenvector of \vec{f} . It is possible to understand this in more depth by looking at the problem from an energy perspective. The overall dynamics of (3.5), for example, over a connected undirected graph can be viewed as two, in-neighbor and out-neighbor, subsystems as follows:

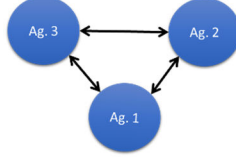


Figure 3.7. three single-integrator agents interacting over an undirected graph under (3.54).

$$\vec{x}_{in}(t) = -\mathcal{L}(\vec{x}_{in})\vec{x}_{in}(t) \quad (3.79.1)$$

$$\vec{x}_{out}(t) = -\mathcal{L}^T(\vec{x}_{out})\vec{x}_{out}(t) \quad (3.79.2)$$

Let $E_{in/out}^i$ be the in/out *generalized energy* [114], i.e., may not be the total mechanical energy, of the i^{th} agent. Thus, the residual energy at this agent will be given as such:

$$E_{residual}^i = E_{in}^i - E_{out}^i \quad (3.79.3)$$

If $E_{residual}^i > 0$ then the i^{th} agent is acting as a *sink* at that time instant. Similarly, if $E_{residual}^i < 0$ then the i^{th} agent is acting as a *source* at that time instant. If $E_{residual}^i = 0$ then the energy is conserved at the i^{th} agent level at that time instant. The system (3.79.2) was used in [14] to achieve consensus among a sensor network. More information about the energy of a graph can be found in [115]. Notice that if the Jacobian is symmetric, i.e., $\mathcal{L} = \mathcal{L}^T, \forall \vec{x}$, - which is indeed the case for an undirected graph and a semi-linear protocol, then $E_{residual}^i$ will be exactly zero.

□

Remark 3.15: Even though forcing the $c\vec{1}$ as a left nonlinear eigenvector of \vec{f} is sufficient to achieve consensus as explained previously, forcing also $c\vec{1}$ as a right eigenvector $\forall \vec{x}$ will make the Jacobian symmetric and thus the energy will be conserved within each agent and not only at the level of the MAS.

For a general nonlinear system and from the perspective of vector fields, the divergence and curl are measures of how the field changes its magnitude and direction about a point

in the phase space, respectively. Knowing that a symmetric Jacobian corresponds to an irrotational or a curl-free field; since its curl is identically zero. Therefore, this vector field is conservative, and the total energy is conserved- assuming it is continuous over a simple connected domain. This is again the advantage of having a symmetric Jacobian structure. In general, if it is impossible to have $\lambda_1 = 0$ using (3.29), then (3.30) must be used to achieve a non-trivial consensus like encountered in (3.69). If both failed, then trivial consensus will be achieved if only the system is stable. So, having only $\vec{e} = \vec{1}$ will ensure consensus, but having only $\vec{v}^T = c\vec{1}^T$ will ensure the average consensus, for example.

Note 3.6: Notice that the eigenvalues of a constant matrix will be equal to those of its transpose. However, this is not necessarily the case for a nonlinear system, in general unless it is diagonalizable [95].

3.2.6: The concept of prescribed connectivity

The algebraic connectivity of the underlying network connecting the dynamical MAS is a vital element in both the stability and the performance of the overall system. It can be usually characterized using the second smallest eigenvalue, i.e., $|\lambda_2|$, of the associated Laplacian matrix which is also called the *Fiedler eigenvalue*. If the graph is not connected, then $\lambda_2 = 0$ and the cooperative behavior is not achieved and might become unstable. Other indicators about the connectivity of a communication network can be given using the end-to-end bit error rate (BER), for example, which is sometimes preferred over the Fiedler metric of connectivity simply because the latter is a high-level measure and does not measure the communication reception quality [85]. However, in this thesis, we will be using the Fiedler eigenvalue as a connectivity metric under the proposed framework.

In general, the algebraic connectivity depends on the edges weights. Consequently, under state-dependency, where the notion of eigenvalues and eigenvectors is still valid, λ_2 will be a function of the states and can be given as $\lambda_2(\vec{x})$. Thus, there is a possibility to lose connectivity within a region of the state space. To avoid such a possibility, the following theorem can be used:

Theorem 3.7: (Theorem 4.1.2) [3] Let $\hat{\mathcal{G}}$ be a graph on N vertices, and let \mathcal{G} be a graph on N vertices created from $\hat{\mathcal{G}}$ by adding a weighted edge joining two nonadjacent vertices in $\hat{\mathcal{G}}$, or by increasing the weight of an existing edge in $\hat{\mathcal{G}}$. Then for all $i = 1, \dots, N$, we have

$$\lambda_i(\mathcal{L}(\hat{\mathcal{G}})) \leq \lambda_i(\mathcal{L}(\mathcal{G})).$$

where $\hat{\mathcal{G}}$ is assumed to be undirected. Considering *Theorem 1* in conjunction with *Note 3.1*, the semi-linear protocols can be designed such that $\mathcal{L}(\vec{x}) = \hat{\mathcal{L}}(\vec{x}) + \sigma \mathcal{L}_f$ with $\sigma > 0 \in \Re$ is a constant which denotes the prescribed connectivity that ensures the algebraic connectivity of $\mathcal{L}(\vec{x})$ to be bounded below $\forall \vec{x}$ if and only if \mathcal{L}_f is associated originally with a connected undirected graph.

The notion of prescribed connectivity proposed herein can be viewed as the D -stability theory [116] used in robust systems to obtain a desired performance irrespective of the uncertainties involved. Figure 3.8 shows a utilization of the previous theorem where additional edges might be added dynamically. The resulting MAS, using the averaging semi-linear protocol, can be given as follows:

$$\dot{\vec{x}}(t) = -(\hat{\mathcal{L}}(\vec{x}) + \sigma \mathcal{L}_f) \vec{x}(t) = -\mathcal{L}(\vec{x}) \vec{x}(t) = \vec{f}(\vec{x}) \quad (3.80)$$

Note 3.7: Notice that if the $c\vec{1}$ was not forced either as a left or right eigenvector, then the resulting state-dependent matrix may not have $\lambda_1 = 0$ as an eigenvalue at all. This justifies

the introduction of nonlinear eigenvalues and eigenvectors when dealing with state-dependent matrices. Moreover, **Theorem 3.1** indicates that the eigenvalues of $\mathcal{L}(\vec{x})$ will increase with respect to $\sigma\mathcal{L}_f$ or remain the same but never decrease *if and only if* both matrices are positive semi-definite [3]. Also, notice that the lower bound is not only a function of the state's initial conditions as it was the case in [64] and [61], but also of $\sigma\mathcal{L}_f$.

The prescribed connectivity will not change the consensus value.

Using such a concept of prescribed connectivity indicates that the communication links represented by $\sigma\mathcal{L}_f$ are persistent, i.e., maintained, irrespective of the working environment or agents' states- positions for example- and more importantly \mathcal{L}_f is assumed initially connected. The initial connectedness of \mathcal{L}_f was an essential assumption in [54], [57], [60], [62] where various controllers were designed to ensure connectivity preservation. Notice that even though the initial connectedness of the graph is required, it might be lost under a dynamically changing graph as was shown in [58]. Therefore, a suitable controller/protocol is needed may be like (3.72) or (3.75). Adopting the same assumption of initial connectedness of \mathcal{L}_f , the herein proposed framework can be used to design the g_{ij} functions such that $\hat{\mathcal{L}}(\vec{x})$ is strictly increasing. Notice that (3.80) may not be applicable especially when dealing with more realistic situations where the notion of communication-aware MAS is inevitable [85] or while navigating through a working space filled with obstacles, for examples. However, it is possible to incorporate the signal-strength in the g_{ij} functions and apply the concept of distance-dependent coordination in the designed controllers to overcome any shortcomings of (3.80) as will be shown in Chapter 4.

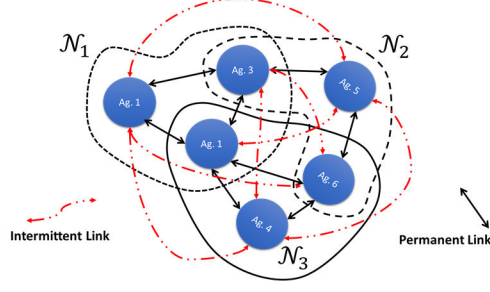


Figure 3.8. A MAS with permanent and intermittent link connecting agents within multiple neighborhoods.

Idea: Figure 3.8 may represent several moving molecules- such that each neighborhood represents one molecule having permanent links or strong forces- connected by weak forces, i.e., intermittent links.

For nonlinear protocols where the resulting MAS of single-integrators can be given as $\vec{\dot{x}}(t) = \vec{f}(\vec{x})$, **Theorem 3.7** can also be used to achieve a prescribed connectivity. In such a case, the overall dynamics of the modified MAS can be given as follows:

$$\vec{\dot{x}}(t) = \vec{f}(\vec{x}) - \sigma \mathcal{L}_f \vec{x}(t) = \vec{f}(\vec{x}) \quad (3.81)$$

The implication of Theorem 1 on the algebraic connectivity of (3.80) and (3.81) can be shown as follows:

Corollary 3.1: Considering **Theorem 3.7** and by taking $i = 2$, all possible values, see **Remark 3.2**, of the modified Fiedler eigenvalue under a prescribed connectivity of σ are given as follows:

$$\lambda_2(\mathcal{L}) \geq \lambda_2(\hat{\mathcal{L}}) + \sigma \hat{\lambda}_2(\mathcal{L}_f) \quad (3.81.1)$$

Proof: Since by design $\hat{\mathcal{L}}(\vec{x})$ is a positive semi-definite matrix $\forall \vec{x}(t)$ its second smallest eigenvalue can either be positive or identically zero if its associated graph get disconnected at any time instant, i.e., $\lambda_2(\hat{\mathcal{L}}) \geq 0, \forall \vec{x}(t)$. So, if \mathcal{L}_f is associated- originally- with a connected undirected graph chosen by the design engineer or achieved by an appropriate

algorithm like [117], then it is straight forward to see that $\lambda_2(\mathcal{L}) \geq \sigma\lambda_2(\mathcal{L}_f) > 0$, $\forall \vec{x}(t)$.

Thus, the MAS will always stay connected while achieving the designated behavior.

Another way to proof both (3.80) and (3.81) can be given as follows:

$$\lambda_2(\vec{f}) = \lambda_2\left(\vec{f}(\vec{x}) - \sigma\mathcal{L}_f\vec{x}(t)\right) \quad (3.81.2)$$

Using (3.29)- or (3.30)- and by finding the extrema points of $\vec{v}_2(\vec{x})$, i.e., by setting $\vec{v}_2(\vec{x}) = \vec{0}$, it is straight forward to show that- when the Jacobian is symmetric:

$$\lambda_2(\vec{x}) = \frac{\vec{v}_2^T(\vec{x}) \left\{ \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} - \sigma\mathcal{L}_f \right\} \vec{v}_2(\vec{x})}{\vec{v}_2^T(\vec{x}) \vec{v}_2(\vec{x})} \quad (3.81.3)$$

where: $\vec{v}_2(\vec{x})$ is the normalized nonlinear left eigenvector associated with $\lambda_2(\vec{x})$ of $\vec{f}(\vec{x})$.

Evaluating (3.81.3) at initial time and at consensus, i.e., at the boundary points $\vec{x} = \vec{x}_0$, $\vec{x} = \vec{1}_c$ using (3.32) and $\vec{v}_2(\vec{1}_c) = \vec{v}_{2f}(\mathcal{L}_f)$, yields:

$$\lambda_2(\vec{x}_0) \geq \lambda_2\left(\hat{\mathcal{L}}(\vec{x}_0)\right) + \sigma\hat{\lambda}_2(\mathcal{L}_f) \quad (3.81.4)$$

$$\lambda_2(\vec{1}_c) = (1 + \sigma)\hat{\lambda}_2(\mathcal{L}_f) = (1 + \sigma) \frac{\vec{v}_{2f}^T \mathcal{L}_f \vec{v}_{2f}}{\vec{v}_{2f}^T \vec{v}_{2f}} \quad (3.81.5)$$

where Weyl's monotonicity theorem [118] was used in (3.81.4). Note that to have both matrices in (3.81.3) positive semi-definite, the Jacobian of $\vec{f}(\vec{x})$ must be symmetric and \mathcal{L}_f is associated with a connected undirected graph. ■

It is important to understand that the evolution of $\lambda_2(\vec{f})$ depends mainly on the Jacobian of $\vec{f}(\vec{x})$ such that if it is increasing then both (3.81.4) and (3.81.5) can be used as lower and upper bounds, respectively. The Jacobian structure depends on the underlying network

topology and the network protocol used. Inspired by the previous proof, one may consider achieving a predetermined evolution of $\lambda_2(\vec{f})$ and then solve (3.32) and/or (3.33) such that $\lambda_1(\vec{f}) = 0, \forall \vec{x}(t), t \geq t_0$ simultaneously with desired $\lambda_2(\vec{f})$. This will result in finding the needed structure of \vec{f} and its Jacobian to achieve the desired connectivity. In general, this problem can be hard to solve, and it is left for a future investigation. However, in Chapter 4, a way to preserve the connectivity will be presented. It is worth mentioning that the prescribed connectivity will not affect the consensus value; since the matrices involved have $\vec{1}$ as a left eigenvector.

Remark 3.16: The concept of prescribed connectivity presented in this section is a special case of connectivity-preserving protocols covered in Chapter 4 where more discussion about how to judge the connectivity of a nonlinear MAS using the concept of eigenvalues, namely $\lambda_2(\vec{x})$, despite being non-unique is presented. Besides, the simulation results shown in this part involving the algebraic connectivity were obtained using the MATLAB *eig*($\mathcal{L}(\vec{x})$) function which is going to be justified in Chapter 4, see *Note 4.12*.

3.2.7: Stability and Convergence Analysis

The stability of the previous continuous-time autonomous nonlinear systems, i.e., $\vec{\dot{x}}(t) = \vec{f}(\vec{x}), \vec{x}(t_0) \in \Omega \subseteq \mathbb{R}^N$, can be investigated in various ways depending on the way it is dealt with. For example, if the whole MAS dynamics is considered as one system, then it is straight forward to think of using the Lyapunov stability methods like first and second Lyapunov methods, the Krasovskii, or other methods as summarized neatly in [119], for example. In general, nonlinear systems may exhibit a set of equilibria or an isolated equilibrium point. If applicable, local or global stability whether exponential or asymptotic are usually of main concern. Recently in [95], a criterion using nonlinear eigenvalues was

proposed to check the global asymptotic stability of a nonlinear system having a unique equilibrium point if it is diagonalizable. It is, by far, difficult to construct or analyze the stability of nonlinear systems in a unified way. So, usually certain classes are proposed, and a general stability criterion can be assigned to each one separately.

From another perspective, it is helpful to treat a nonlinear system, mainly the MAS, as a collection of composite, i.e., interconnected, systems. This will simplify the analysis of stability by constructing a separate Lyapunov function for each system which can then be grouped into a composite Lyapunov function through their weighted sum. In such a case, the problem of finding feasible weights is the main concern [120]. Usually, these weights result in a special type of matrices called the M -matrices.

Now, since $\vec{f}(\vec{x})$ has its Jacobian acting as a point-wise singular M -matrix in the desired manifold of the state space- as per the proposed framework, it is desirable to investigate which type of stability it exhibits. For example, in (3.40), the D -semistability is needed. The conditions stated when (3.18), (3.22) and (3.26) were introduced, certainly to ensure this type of stability. Moreover, since the Laplacian matrix associated with any undirected graph is symmetric, it is easy to show that $G(\vec{x})\mathcal{L}(\vec{x})$ in (3.40) is indeed D -semistable. Thus, there is no need to find a stronger type of stability of the involved singular M -matrices appearing in the protocols presented so far.

In the previous nonlinear functions, the equilibria set can be defined as follows:

$$\mathfrak{S} = \{\vec{x} \in \mathbb{R}^N \mid \vec{f}(\vec{x}) = \vec{0}\} \cup \{\vec{x} \in \mathbb{R}^N \mid \vec{v}^T \vec{f}(\vec{x}) = 0\} \cup \{\vec{x} \in \mathbb{R}^N \mid \vec{f}(\vec{x}) \vec{e} = \vec{0}\} \quad (3.82)$$

where: \vec{v} and \vec{e} are the nonlinear left and right eigenvectors associated with $\lambda(\vec{x}) = 0$ of $\vec{f}(\vec{x})$, respectively.

The stability of the previous continuous-time autonomous nonlinear systems, i.e., $\vec{x}(t) = \vec{f}(\vec{x}), \vec{x}(t_0) \in \mathfrak{R}^N$, can be investigated using the LaSalle's theorem [120] as follows:

Let the following be a candidate Lyapunov function of a general nonlinear protocol:

$$V(\vec{x}) = \vec{f}^T \vec{f} \quad (3.83)$$

Evaluating the time derivative, yields:

$$\dot{V}(\vec{x}) = \vec{f}^T \left\{ \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right)^T + \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right) \right\} \vec{f} \leq 0 \quad (3.84)$$

Per the proposed framework, the Jacobian of $\vec{f}(\vec{x})$ is the negative of a point-wise singular M -matrix. So, from Definitions 4 and 5 it is clear that $\dot{V}(\vec{x}) \leq 0$. Let \mathfrak{S} be the largest invariant set in \mathfrak{R}^N such that $\mathfrak{S} \subseteq E$ where $E = \{\vec{x} \in \mathfrak{R}^N \mid \dot{V}(\vec{x}) = 0\}$. Let Ω be a compact positively invariant set such that $\Omega = \{\vec{x} \in \mathfrak{R}^N \mid V(\vec{x}) \leq b\}, b \in \mathfrak{R}$. Thus, from LaSalle's theorem, every trajectory starting in Ω will converge to \mathfrak{S} asymptotically as time reaches infinity. These findings are global when $V(\vec{x}) \rightarrow \infty$ as $\|\vec{x}\| \rightarrow \infty$, so the design of the network protocol must also ensure that $\|\vec{f}(\vec{x})\| \rightarrow \infty$ as $\|\vec{x}\| \rightarrow \infty$ or more specifically as $\|x_i - x_j\| \rightarrow \infty \forall i, j \in \mathcal{N}_i$. This means that the network protocol must provide sufficient energy to each agent such that the consensus is achieved.

For any protocol that can be written in a semi-linear format, like (3.5) or (3.64) for example, let the following Lyapunov candidate function of (3.5) be used as follows:

$$V(\vec{x}) = \vec{x}^T \vec{x} \quad (3.85)$$

Thus, the time-derivative of (3.85) can be given as follows:

$$\dot{V}(\vec{x}) = -2\vec{x}^T \mathcal{L}(\vec{x}) \vec{x} \leq 0 \quad (3.86)$$

where $\mathcal{L}(\vec{x})$ is associated with a state-dependent undirected graph, and is thus symmetric. For example, let us consider two single-integrator agents connected via an undirected link where both (3.85) and (3.86) are used to study the properties of the network protocol (3.3) used to connect them. In such a case, it is easy to show that (3.86) can be given as follows:

$$\dot{V}(\vec{x}) = -2(x_1 - x_2)^2 g_{12} \leq 0 \quad (3.87)$$

Now, let g_{12} be a distance-dependent weighting function given by:

$$g_{12} = \frac{1}{[1 + (x_1 - x_2)^2]^\beta}, \quad \beta > 0 \quad (3.88)$$

which is the Cucker-Smale (C-S) flocking model [64].

So, for (3.87) to be valid $\forall \vec{x}(t) \notin \mathfrak{J}$ and $\forall t \geq t_0$ where $|x_1(t_0) - x_2(t_0)| \rightarrow \infty$, it is sufficient to have $0 < \beta \leq 1$ which agrees with [64]. A necessary condition for reaching consensus under a general state-dependent network protocol is to have:

$$\dot{V}(\vec{x}) < 0 \text{ as } \|\vec{x}\| \rightarrow \infty, \quad \forall \vec{x}(t) \notin \mathfrak{J}, \quad \forall t \geq t_0 \quad (3.89)$$

Under state-independent network protocol, (3.89) is naturally satisfied. Moreover, all the previous conditions for global stability must be restricted to the domain of interest if it is other than \mathfrak{R}^N . Note that if (3.89) is not satisfied, then the consensus is not achieved while the invariant quantity is achieved. This can be seen from Figure 3.7 in [64] where the arithmetic mean was achieved but not the consensus. This observation- which was not pointed out in [64]- can be justified by the fact that if agents do not have sufficient energy, then they will maintain the invariant quantity by utilizing the available energy in all agents even if it means that they keep their initial values. So, a consensus protocol that is stable might be non-converging.

To analyze the convergence of the connected systems to the general time-invariant consensus value, i.e., c , let the disagreement vector be defined as follows [53]:

$$\vec{x}(t) = \vec{1}c + \vec{\delta}(t) \quad (3.90)$$

where: $x_i - x_j = \delta_i - \delta_j$, and $\sum_i \delta_i = 0$.

So, the disagreement dynamics can be given as follows- after a change of variables:

$$\dot{\vec{\delta}}(t) = \vec{f}(\vec{\delta}) \quad (3.91)$$

Note that for a symmetric Laplacian matrix, (3.91) will also be valid for a semi-linear protocol since $\vec{1}$ is a left and right eigenvector. Let the following be a candidate Lyapunov function for (3.91):

$$V(\vec{\delta}) = \vec{f}^T \vec{f} \quad (3.92)$$

Evaluating the time derivative, yields:

$$\dot{V}(\vec{\delta}) = \vec{f}^T \left\{ \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right)^T + \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right) \right\} \vec{f} \leq 0 \quad (3.93)$$

Intuitively, if (3.93) is satisfied globally, then (3.92) is indeed a Lyapunov function of (3.91) since $V(\vec{\delta}) > 0$ and $V(\vec{0}) = 0$ and thus the consensus is globally asymptotically achieved for all initial conditions *if and only if* (3.89) is satisfied as well. In case (3.93) is satisfied only in a local neighborhood of the origin, then the convergence is also local. In general, for nontrivial dynamics, the consensus may be reached even though the MAS is unstable. This can be clearly seen in connected double-integrator systems as shown in [64] for example. Note that the proposed Lyapunov functions are not unique. So, other Lyapunov candidates could be used to establish the global stability and convergence of any developed network protocols.

3.2.8: Performance Analysis

Specifically, the convergence rate towards consensus can be given as a function of the second smallest eigenvalue associated with the state-dependent Laplacian matrix, i.e., the Fiedler eigenvalue denoted by λ_2 . Using (3.92), this convergence rate can be characterized as follows- assuming a general nonlinear protocol:

$$\dot{V}(\vec{x}) = -\lambda_2 \left(\left(\frac{\partial \vec{f}}{\partial \vec{x}} \right)^T + \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right) \right) V(\vec{x}) \leq 0 \quad (3.94)$$

Which depends on the symmetric part of the Jacobian that needs not be of monotonic type. Similarly, for a semi-linear network protocol, the convergence rate can be given as follows based on (3.92)- assuming a symmetric Jacobian:

$$\dot{V}(\vec{x}) = -\lambda_2 \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right) V(\vec{x}) \leq 0 \quad (3.95)$$

Considering (3.81.1), we may conclude that the convergence rate obtained in (3.94) and (3.95) is at least as fast as $\sigma \hat{\lambda}_2(\mathcal{L}_f)$.

Interestingly, if $\vec{f}(\vec{x})$ is diagonalizable, then both left and right eigenvalues will be the same [95]. Thus, by adding (3.29) and (3.30) and solving for $\lambda(\vec{x})$ assuming a symmetric Jacobian, then $\lambda_2(\vec{x})$, in specific, can be given as follows:

$$\lambda_2(\vec{x}) = \frac{\vec{v}_2^T \left(\left(\frac{\partial \vec{f}}{\partial \vec{x}} \right)^T + \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right) \right) \vec{v}_2}{2 \vec{v}_2^T \vec{v}_2} = \frac{\vec{v}_{2f}^T \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right) \vec{v}_{2f}}{\vec{v}_{2f}^T \vec{v}_{2f}} \quad (3.96)$$

where the left and right nonlinear eigenvectors are element-wise similar but with opposite signs.

Theorem 3.8: If the Jacobian of $\vec{x}(t) = \vec{f}(\vec{x})$ is symmetric, then we may use the linear left and right eigenvectors, i.e., \vec{v}_{2f}^T and \vec{e}_{2f} , associated with the second smallest eigenvalue,

i.e., λ_{2f} , to monitor the connectivity, i.e., $\lambda_2(\vec{x})$ of $\vec{f}(\vec{x})$, $\forall \vec{x}$ in the domain of interest using (3.96).

Proof: The result given by (3.96) can be shown as follows:

Definition 3.5: [99] A system $\dot{\vec{x}}(t) = \vec{f}(\vec{x})$ is said to be in a diagonal form if its Jacobian, i.e., $\partial \vec{f} / \partial \vec{x}$, is given as:

$$\frac{\partial \vec{f}}{\partial \vec{x}} = \text{diag}([\lambda_1, \dots, \lambda_N]) \quad (3.96.1)$$

where: $\lambda_i, i = 1, \dots, N$ are the set of eigenvalues from the field of meromorphic functions of variables \vec{x} denoted by \mathcal{K} . Moreover, we have:

Theorem 3.9: [99] The eigenvalues of $\dot{\vec{x}}(t) = \vec{f}(\vec{x})$ are invariant with respect to a change of coordinates $\xi = \phi(\vec{x})$ where $\phi \in \mathcal{K}$.

So, the transformation of $\vec{f}(\vec{x})$ into the diagonal form, which is a special case of the feedforward form, can be studied through the following:

Theorem 3.10: [99] Given a system $\dot{\vec{x}}(t) = \vec{f}(\vec{x})$, there exists a change of coordinates $\xi = \phi(\vec{x})$ that transforms the system into a diagonal form if and only if there exist N eigenvalues $\lambda_1, \dots, \lambda_N$ associated with N eigenvectors e_1, \dots, e_N such that:

$$T^{-1} = (\vec{e}_1 | \vec{e}_2 | \dots | \vec{e}_N)$$

is nonsingular and $Tdx = (w_1 | w_2 | \dots | w_N)^T$ where the one-form [95] $(w_1 | w_2 | \dots | w_N)$ is exact.

Now, if the $\vec{1}$ is forced as a left and right eigenvector associated with $\lambda(\vec{x}) = 0, \forall \vec{x}$ using (3.33.1) and (3.33.2), then the resulting Jacobian of $\vec{f}(\vec{x})$ is pointwise symmetric in the desired subset of the state-space. Knowing that a real symmetric matrix is always diagonalizable [121], then there exists a coordinate transformation at each point in the

state-space that can be used to transform $\vec{f}(\vec{x})$ into a diagonal form, however calculating these transformations as functions in \mathcal{K} is beyond the scope of this thesis.

As pointed out in [95], it is possible to show that the sets of nonlinear left and right eigenvalues of $\vec{f}(\vec{x})$ are equivalent when the system is in a diagonal form. However, the relations between the left and right eigenvectors can be investigated as follows- where *col* and *row* indicate the column-wise and row-wise stacking operations, respectively:

Let $A = \partial \vec{f} / \partial \vec{x}$. Then [99], using (3.30):

$$\begin{aligned} AT^{-1} &= col(A\vec{e}_1 | A\vec{e}_2 | \cdots | A\vec{e}_N) = col(\lambda_1 \vec{e}_1 + \dot{\vec{e}}_1 | \lambda_2 \vec{e}_2 + \dot{\vec{e}}_2 | \cdots | \lambda_N \vec{e}_N + \dot{\vec{e}}_N) \\ &= T^{-1}\Lambda + \frac{d(T^{-1})}{dt} = T^{-1}\Lambda - T^{-1}\dot{T}T^{-1} \end{aligned} \quad (3.96.2)$$

where: $\Lambda = diag([\lambda_1, \dots, \lambda_N])$. Since T^{-1} is nonsingular, we may write:

$$\Lambda = TAT^{-1} + \dot{T}T^{-1} \quad (3.96.3)$$

Similarly, using (3.29) we may write:

$$MA = row(\vec{v}_1^T A | \vec{v}_2^T A | \cdots | \vec{v}_N^T A) \quad (3.96.4)$$

where: $M = [\vec{v}_1, \vec{v}_2, \dots, \vec{v}_N]^T$. So:

$$MA = row(\vec{v}_1^T \lambda_1 - \dot{\vec{v}}_1^T | \vec{v}_2^T \lambda_2 - \dot{\vec{v}}_2^T | \cdots | \vec{v}_N^T \lambda_N - \dot{\vec{v}}_N^T) = \Lambda M - \dot{M} \quad (3.96.5)$$

Knowing that A is symmetric, Λ is a diagonal matrix, and M^{-1} is nonsingular, we may write:

$$\Lambda = MAM^{-1} + \dot{M}M^{-1} \quad (3.96.6)$$

Since the Jacobian is symmetric, its left and right eigenvectors are identical, i.e., $\vec{v}^T = \vec{e}$, and therefore $M = T$. This also can be seen by pre-multiplying (3.29) and (3.30) with \vec{e}^T and \vec{v}^T , respectively, and then adding them we will have:

$$\vec{e}^T \vec{v} + \vec{v}^T \vec{e} = 0 \quad (3.96.7)$$

Indicating in fact that both vectors are orthogonal, knowing that the left and right eigenvalues are identical.

Taking $\vec{v} = \vec{v}_f$ and $\vec{e} = \vec{e}_f$ - see *Remark 3.9*, as possible eigenvectors of each eigenvalue, consequently, (3.96.7) is still valid and therefore we can monitor, or even design, the connectivity of the nonlinear system based on the variations in the linear signature of the underlying dynamical system involved, mainly λ_{2f} . Remember that in the proposed framework we have built the protocols on top of its linear counterpart.

■

Lemma 3.4: If the Jacobian of $\vec{x}(t) = \vec{f}(\vec{x})$ is asymmetric, then the results of *Theorem 3.5* applies only in a small neighborhood of the equilibria.

Note 3.8: To find the values of $\lambda_2(\vec{x})$ given in (3.96), one needs to solve (3.33.1) to get \vec{v}_2 which is difficult, in general. So, in this part, (3.96) will be used mainly to characterize the values of $\lambda_2(\vec{x})$ rather than finding it analytically. However, a special case where we are interested in finding $\lambda_2(\vec{x})$ is presented in Chapter 4.

3.2.8.1: The concept of prescribed performance revisited

When it comes to performance, one may expect that it is desired to control the evolution of the states over time in a certain manner by imposing suitable constraints as found for example in [54]. However, in our case we are interested in imposing certain invariance and performance metrics on the structures of the designed protocols to achieve desired characteristics- such as prescribed connectivity, divergence, curl and volume- required by various applications. Later in this section, we will show- through a simple example- how

the concept of prescribed performance devised in [54] can be used under the herein proposed framework.

Motivated by (3.96) and **Theorem 3.5**, the following design procedure by which a prescribed performance that ensures the connectivity preservation, can be stated as follows:

Average Consensus Protocol under state-dependent networks with prescribed performance:

Given $\lambda_{2\min}$ and $\lambda_{2\max}$, design $\vec{f}(\vec{x})$ such that the following is true:

$$\lambda_{2\min} \leq \min_i \lambda_i(\vec{x}) \leq \lambda_{2\max}$$

$$\forall \vec{x} \in \Omega \subseteq \Re^N, i = 2, \dots, N.$$

Subjected to:

$$\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}}|_{\mathfrak{I}} = -\mathcal{L}(\vec{x})|_{\mathfrak{I}} = -\mathcal{L}_f \quad (3.97)$$

$$\left(\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \right)^T \vec{1} = \vec{0} \quad \text{and/or} \quad \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \vec{1} = \vec{0}$$

□

If both $\lambda_{2\min}$ and $\lambda_{2\max}$ are linearly dependent, then one way to solve (3.97) is to first design $\sigma \mathcal{L}_f$ - assuming a semi-linear protocol- to make sure about $\sigma \lambda_2(\mathcal{L}_f)$ using [122] for example and making sure that the g_{ij} functions used are strictly increasing. Doing so, then (3.81.4) and (3.81.5) will be the lower and upper bounds of the algebraic connectivity, respectively. Note that by multiplying (3.32) by a constant $\tilde{\sigma} \in \Re$, then $\lambda_{2\max}$ becomes $\lambda_{2\max} = (\tilde{\sigma} + \sigma) \lambda_2(\mathcal{L}_f)$ instead of $(1 + \sigma) \lambda_2(\mathcal{L}_f)$ and therefore can be chosen more freely. Generally, we may write- by dropping \min_i from (3.97):

$$\lambda_{2 \min} \leq \frac{-1}{(N-1)} \text{trace} \left(\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \right) \leq \lambda_{\max} \quad (3.98)$$

under the previously stated conditions where $\lambda_1 = 0$ and λ_{\max} is the upper bound of spectral radius of the Jacobian matrix $\forall \vec{x}$ such that:

$$\lambda_1 = 0 < \lambda_{2 \min} \leq |\lambda_2| \leq |\lambda_3| \leq \dots \leq |\lambda_N| \leq \lambda_{\max}$$

Notice that the $\text{trace}(\partial \vec{f} / \partial \vec{x})$ is nothing but the divergence of \vec{f} ($\text{div}(\vec{f})$) which gives a measure about the change in magnitude of the vector field \vec{f} about a point in the state-space. Note that $\text{div}(\vec{f}) = \text{Vol}(\vec{f})$ which is the volume of the flow of the vector field \vec{f} . Also notice that the diagonal elements of the Jacobian matrix represent the negative of the in-degree of each agent, i.e., $-d_i$, plus the negative of a scalar function of partial differential quantities ($-\Delta(\vec{x})$) when a semi-linear protocol is used as will be explained in the coming section. So, we may look at the problem from a graph theoretic view as bounds on the graph *in-volume* $\text{Vol}(\mathcal{G})$ as follows:

$$0 < \lambda_{2 \min} \leq \frac{-1}{(N-1)} \{ \text{Vol}(\mathcal{G}(\vec{x})) + \Delta(\vec{x}) \} \leq \lambda_{\max} \quad (3.99)$$

where: $\text{Vol}(\mathcal{G}(\vec{x})) = \sum_i d_i(\vec{x})$ and $\text{Vol}(\vec{f}) = \text{Vol}(\mathcal{G}(\vec{x})) + \Delta(\vec{x})$. Since $d_i(\vec{x})$ is affected by the initial conditions, we may use the concept of prescribed connectivity presented in section **E** to avoid this effect as follows- assuming a semi-linear protocol:

$$\lambda_{2 \min} \leq \frac{-1}{(N-1)} \left\{ -\text{trace} \left(\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \right) + \text{trace}(\sigma \mathcal{L}_f) \right\} \leq \lambda_{\max} \quad (3.100)$$

Imposing $\lambda_{2 \min}$ by design requirements, both σ and \mathcal{L}_f can be planned in advance assuming $\text{trace} \left(\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \right) = 0$, for example, while the g_{ij} functions should be designed as

strictly increasing to have $\lambda_{2\max}$ given by (3.81.5) and λ_{\max} could be taken as $(1 + \sigma)\hat{\lambda}_N(\mathcal{L}_f)$.

To see the importance of d_i on the algebraic connectivity- and therefore the importance of (3.99)-, let \mathcal{L}_f be associated with an undirected graph, then its algebraic connectivity is bounded above by [123]:

$$\lambda_2(\mathcal{L}_f) \leq \frac{N}{N-1} d_{\min} \quad (3.101)$$

where: $d_{\min} = \min(d_i)$ is the minimum in-degree or vertex degree in the graph. So, by increasing d_{\min} in $\mathcal{L}(\vec{x})$, consensus might be reached more rapidly. A lower bound of $\lambda_2(\mathcal{L}_f)$ - not necessarily tight- is given as follows [124]:

$$\lambda_2(\mathcal{L}_f) \geq \frac{2N}{2 + N(N-1)d - 2md} \quad (3.102)$$

where: d is the graph diameter, i.e., the maximum distance between any two vertices in the connected graph [3], and m is its size, i.e., the number of edges. Note that equality in (3.102) holds when the graph is complete or P^3 [124]. Other lower bounds on $\lambda_2(\mathcal{L}_f)$ that includes a connected undirected graph in-volume is given as follows [9]:

$$\lambda_2(\mathcal{L}_f) \geq \frac{1}{\text{Vol}(\mathcal{G}_f) d} \quad (3.103)$$

Or [125]:

$$\lambda_2(\mathcal{L}_f) \geq \frac{4}{Nd} \quad (3.104)$$

It is clear from (3.104) that a disconnected graph with $\lambda_2 = 0$ corresponds to an infinite diameter, i.e., $d = \infty$. So, to have a connected graph, i.e., $\lambda_2 > 0$, d must be minimized. A comparison between (3.102) and (3.104) is available in [124] which also includes lower bounds of $\hat{\lambda}_N$.

Relating the vector calculus of flow vector fields to planar digraphs theoretic sense can be traced back to [126]- or even earlier- for example where it showed that any network may be decomposed into three networks, namely: potential component having divergence-free and curl-free, solenoidal component having divergence-free but not curl-free, and irrotational component having curl-free but not divergence-free.

For example, using the semi-linear consensus protocol (3.5) will result in a curl-free flow over undirected graphs with symmetric structure when assuming a simple connected domain, however, the resulting divergence will be varying. Using (3.51) over the same undirected graph, will result in a constant negative divergence however the resulting flow will have vorticity, i.e., rotational. The consensus protocol given in (3.51) is indeed an entropic protocol [56]. Other possibilities will be investigated in the coming section.

Recalling the concept of prescribed performance given in [54], the applicability of the herein proposed framework can be demonstrated as follows:

Let the time-dependent constraint that governs the evolution of the MAS states towards consensus is given as follows:

$$\begin{aligned} \varpi \cdot t + \min_i(x_i(t_0)) &\leq x_i \leq -\varpi \cdot t + \max_i(x_i(t_0)) \\ \forall \vec{x} \in \Omega \subseteq \mathfrak{R}^N, i &= 1, 2, \dots, N \text{ and } t \in [t_0, t_c] \end{aligned} \quad (3.105)$$

where: t denotes the time, t_c denotes the consensus time (unknown a priori) and $\varpi > 0$ is a real constant. After t_c , the inequality in (3.132) flips. Taking the derivative of (3.132) with respect to time, yields:

$$-\varpi \leq u_i \leq \varpi, \quad \forall i = 1, 2, \dots, N. \quad (3.105.1)$$

Or simply as:

$$|u_i| \leq \varpi, \quad \forall i = 1, 2, \dots, N. \quad (3.105.2)$$

Which can be solved using a well-designed version of (3.69). In the future, other time-dependent constraints will be investigated once the proposed framework is upgraded to include both time and state dependencies. If $x_i(t_0) \in [x_{min}, x_{max}] \forall i$, then (3.105) can be relaxed, i.e., there is no need to know $\min_i(x_i(t_0))$ and $\max_i(x_i(t_0))$ a prior, as follows:

$$\varpi \cdot t + x_{min} \leq x_i \leq -\varpi \cdot t + x_{max} \quad (3.106)$$

where: $t \in [t_0, t_c]$ and $t_c \leq t_f$ such that t_f is determined by the designer. In such a case, ϖ can be designed or estimated as follows:

$$\varpi = \frac{x_{max} - x_{min}}{2t_f} \quad (3.106.1)$$

Note that (3.106) may be violated after t_c , i.e., after reaching consensus. The results obtained using (3.69) will be shown in the simulation section.

3.2.9: Analyzing consensus protocols using vector calculus

Treating spatial dynamical agents as particles is an oversimplification that makes building navigation systems or path planners quite easy. Usually, the open environment is modelled as a potential field which the navigation system tries to wander in the most desirable way. Usually, these potential fields are used to generate the velocity field, or the artificial force, that will govern the evolution of the navigation system through time. So far, the dynamics of the navigation systems are given by (3.1), i.e., assuming massless particles that can accelerate instantaneously in every direction.

Considering the previous discussion, what we were trying to do is to find a suitable potential field that can be used to achieve the desired task while imposing certain conditions on it that will ensure the performance and stability of the designed protocols. So, motivated by the requirements, different types of potential fields can be considered as

candidates while issues related to the applicability of the principle of superposition, boundedness, local/global minima/maxima and smoothness must be considered. Being mostly interested in spatial agents having their working space in 2-D or 3-D, we may refer to the fundamental theorem of vector calculus also known as Helmholtz representation or Helmholtz decomposition to develop a methodology that helps finding suitable weighing-functions, i.e., protocols, rather than randomly check them one by one. This of course will be handled according to the framework we are proposing.

Considering *Remark 3.8*, the edges weighting function are assumed to be in $\mathfrak{R}^2 \rightarrow \mathfrak{R}_{>0}$. Therefore, let us have a general force vector field denoted by $\vec{f}(x_1, x_2): \mathfrak{R}^2 \rightarrow \mathfrak{R}^2$ which according to Helmholtz decomposition can be written as a combination of two vector field as follows [127]:

$$\vec{f}(x_1, x_2) = \nabla h(x_1, x_2) + \nabla \times \vec{H}(x_1, x_2) \quad (3.107)$$

where: $\nabla = \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2} \right]$ is the gradient operator, h is a scalar map, i.e., $h: \mathfrak{R}^2 \rightarrow \mathfrak{R}$, \vec{H} is a vector field, i.e., $\vec{H}: \mathfrak{R}^2 \rightarrow \mathfrak{R}^2$, and \times denotes the cross product. If the force field \vec{f} is associated with a conservative field, then its curl, i.e., $\text{curl}(\vec{f}) = \nabla \times \vec{H}(x_1, x_2)$, is zero everywhere in its domain. In such a case, \vec{f} will only be associated with a potential function p such that $h(x_1, x_2) = -p(x_1, x_2)$ and consequently can be written as $\vec{f} = -\nabla p$. This is exactly the equation used to solve the consensus problem as will be shown shortly. Given that, now it is obvious how network protocols do relate to the realm of vector calculus.

To elaborate more on this point in specific, consider the MAS shown in Figure 3.9 consisting only of two dynamical agents modelled by (3.1) connected by two edges whose weighting functions are g_{12} and g_{21} and let a semi-linear protocol be used.

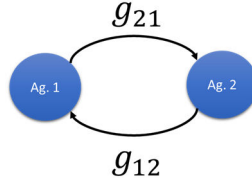


Figure 3.9. A MAS of two agents connected by two links (edges).

Let each node in Figure 3.9 has its own potential function, namely: p^{12} and p^{21} where the superscript denotes the sink/ source node index. Being interested in understanding the mutual effect of these potential functions, let us assign an observer at each node and evaluate the gradient of each potential function in the direction of the other node to establish the in-systems of both agents. Thus, the dynamics of this MAS is given as follows:

$$\vec{\dot{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \vec{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = - \begin{bmatrix} p_{x_1}^{12} \\ p_{x_2}^{21} \end{bmatrix} = - \begin{bmatrix} g_{12} (x_1 - x_2) \\ g_{21} (x_2 - x_1) \end{bmatrix} = - \begin{bmatrix} g_{12} & -g_{12} \\ -g_{21} & g_{21} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.108)$$

where: $p_{x_1}^{12}$ and $p_{x_2}^{21}$ are the partial derivatives of p^{12} and p^{21} in the directions of x_1 and x_2 , respectively. Now, since we have assumed \vec{f} to be conservative, then its curl must equal zero. This last requirement imposes certain conditions on the weighting functions to be designed or selected. So:

$$curl(\vec{f}) = \begin{vmatrix} \bar{x}_1 & \bar{x}_2 & \bar{x}_3 \\ \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_3} \\ p_{x_1}^{12} & p_{x_2}^{21} & 0 \end{vmatrix} = \left(\frac{\partial p_{x_2}^{21}}{\partial x_1} - \frac{\partial p_{x_1}^{12}}{\partial x_2} \right) \bar{x}_3 \quad (3.109)$$

where: \bar{x}_1, \bar{x}_2 and \bar{x}_3 are three unit-vectors in the three-perpendicular direction assuming Cartesian coordinates. Using (3.108) to solve (3.109) we will have the following:

$$curl(\vec{f}) = g_{12} - g_{21} + (x_2 - x_1) \left[\frac{\partial g_{12}}{\partial x_2} + \frac{\partial g_{21}}{\partial x_1} \right] \quad (3.109.1)$$

Which can be made zero $\forall \vec{x}$ if $g_{12} = g_{21}$ and $\frac{\partial g_{12}}{\partial x_2} = \frac{\partial g_{21}}{\partial x_2} = -\frac{\partial g_{21}}{\partial x_1}$ which agrees with both (3.34) and (3.35). Similarly, $\text{div}(\vec{f})$ can be found by taking the trace of the Jacobian as follows:

$$\text{div}(\vec{f}) = -(g_{12} + g_{21}) + (x_1 - x_2) \left[\frac{\partial g_{21}}{\partial x_2} - \frac{\partial g_{12}}{\partial x_1} \right] \quad (3.110)$$

Using (3.34), (3.35) and $g_{12} = g_{21}$, we may write (3.110) as follows:

$$\text{div}(\vec{f}) = -2 \left(g_{12} + (x_1 - x_2) \frac{\partial g_{12}}{\partial x_1} \right) \quad (3.110.1)$$

So, $g_{12} > 0$ must dominate the other term if it happens to be negative.

Note 3.9: The negative divergence is necessary to have a stable MAS system even if it is not convergent to consensus. As a result, for the consensus problem we may have only vector fields with negative divergence whose curl could be zero. That is, we may use conservative and non-conservative force fields to construct the consensus protocols. Note that a vector field whose divergence and curl are zeros is a Laplacian field and it is associated with a harmonic potential function- not necessarily unique. A non-constant harmonic function $p: \mathbb{R}^2 \rightarrow \mathbb{R}$ is unbounded (*Liouville's Theorem*). A deeper analysis of the gradient flow, i.e., $\vec{\dot{x}} = -\nabla p$, of a harmonic function p in \mathbb{R}^3 can be found in [128].

From consensus point of view, having a positive divergence means MAS instability, while having a zero divergence indicates that the MAS is disconnected.

To understand the relation between the divergence and convergence mentioned in **Note 3.7**, let us have the g_{12} function in (3.108) given by (3.88), then the resulting divergence is given as follows- for an arbitrary $\beta > 0$:

$$\text{div}(\vec{f}) = -2 \left\{ \frac{1 + (1 - 2\beta)(x_2 - x_1)^2}{[(x_2 - x_1)^2 + 1]^{\beta+1}} \right\} \quad (3.110.2)$$

Let $e_0 = x_1(t_0) - x_2(t_0)$, then:

$$\lim_{e_0 \rightarrow \infty} \text{div}(\vec{f}) = 0 \quad (3.110.3)$$

even for $\beta = 1$. This result does not contradict with [64]; because what (3.110.3) simply says is that the MAS will have less tendency to leave its current position.

3.2.10: Constructing consensus protocols using the proposed framework:

So far, we have used the proposed framework to test existing consensus protocols available in the literature or constructing major classes using educated guessing. However, in this section, we will show how to utilize the proposed framework to help us constructing the protocols- mainly nonlinear- by solving a system of second-order linear and homogeneous Partial Differential Equations (PDEs) with constant coefficients.

Adopting the principle of superposition and the notation presented in (3.108), let us have a MAS consisting of N single-integrator agents. Therefore, its dynamics can be given as follows:

$$\vec{\dot{x}} = \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_N \end{bmatrix} = \vec{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} = - \begin{bmatrix} \sum_{j \in \mathcal{N}_1} p_{x_1}^{1j} \\ \vdots \\ \sum_{j \in \mathcal{N}_N} p_{x_N}^{Nj} \end{bmatrix} \quad (3.111)$$

Since we are interested in solving the consensus problem in specific, we need to have $\lambda_1 = 0, \forall \vec{x}$. So, we must solve (3.33.2) or both (3.33.1) and (3.33.2) to have average consensus and a curl-free force field. Solving (3.33.1) alone ensures having average consensus but not a curl-free force field. These cases are handled separately as follows- after evaluating the Jacobian:

3.2.10.1: Solving (3.33.2) alone:

This case is equivalent to forcing $\vec{e} = \vec{1}$ as a nonlinear right eigenvector associated with $\lambda_1 = 0$. So, the resulting system of PDEs is:

$$p_{x_i x_i}^{ij} + p_{x_i x_j}^{ij} = 0, \quad \forall i, j \in \mathcal{N}_i \quad (3.112)$$

3.2.10.2: Solving (3.33.1) alone:

This case is equivalent to forcing $\vec{v}^T = \vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1 = 0$. So, the resulting system of PDEs is:

$$p_{x_i x_i}^{ij} + p_{x_j x_i}^{ji} = 0, \quad \forall i, j \in \mathcal{N}_i \quad (3.113)$$

3.2.10.3: Solving both (3.33.1) and (3.33.2):

This case is equivalent to forcing both $\vec{e} = \vec{1}$ and $\vec{v}^T = \vec{1}^T$ as nonlinear right and left eigenvectors associated with $\lambda_1 = 0$. This is also equivalent to solving the following scalar problem:

$$\vec{1}^T \left[\left(\frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \right)^T + \frac{\partial \vec{f}(\vec{x})}{\partial \vec{x}} \right] \vec{1} = 0 \quad (3.114)$$

So, the resulting PDEs are:

$$\sum_{j \neq i \in \mathcal{N}_j} \sum_{i \neq j \in \mathcal{N}_i} \{ 2p_{x_i x_i}^{ij} + p_{x_j x_i}^{ji} + p_{x_i x_j}^{ij} \} = 0 \quad (3.114.1)$$

To simplify (3.114.1), we may manipulate the indices i and j if all $p^{ij}: \mathcal{R}^2 \rightarrow \mathcal{R}$ have the same structure- with similar parameters- such that it is possible to rewrite it as follows:

$$\{ p_{x_i x_i}^{ij} + p_{x_j x_j}^{ji} \} + \{ p_{x_i x_j}^{ij} + p_{x_j x_i}^{ji} \} = 0, \forall i, j \in \mathcal{N}_i \quad (3.114.2)$$

Note that the first bracket is Laplace's equation if it is identically zero implying that the divergence is zero which is not desired when solving the consensus problem as stated

before. Since the stability of the MAS using consensus protocols requires a negative divergence, then the second bracket must be positive for (3.114.2) to be true. More elegantly, (3.114.2) can be written as follows for arbitrary i and j :

$$\frac{\partial^2 p}{\partial x_i^2} + \frac{\partial^2 p}{\partial x_i \partial x_j} + \frac{\partial^2 p}{\partial x_j \partial x_i} + \frac{\partial^2 p}{\partial x_j^2} = 0 \quad (3.114.3)$$

which is the parabolic PDE. All the previous PDE systems should make use of (3.32) and assume general initial conditions. Note that for the semi-linear protocol (3.5), it is sufficient to consider (3.34) and (3.35) instead of (3.114.3). After finding the potential function p , we can construct our consensus protocol as shown in (3.111).

The general form of second-order linear and homogeneous PDE with constant coefficients is given as follows:

$$A \frac{\partial^2 p}{\partial x_i^2} + 2B \frac{\partial^2 p}{\partial x_i \partial x_j} + C \frac{\partial^2 p}{\partial x_j^2} + D \frac{\partial p}{\partial x_i} + E \frac{\partial p}{\partial x_j} + F = 0 \quad (3.115)$$

which can be characterized using the following matrix:

$$Z = \begin{bmatrix} A & B \\ B & C \end{bmatrix} \quad (3.115.1)$$

If $Z > 0$, then (3.115) is an elliptic PDE. Laplace's equation is a simple example of such PDE. If $Z < 0$, then (3.115) is a hyperbolic PDE. The wave equation is an example of hyperbolic PDEs. If $Z = 0$, then (3.115) is parabolic which is indeed; since $A = B = C = 1$ with $D = E = F = 0$. Heat conduction equation and other diffusion equations are examples of parabolic PDEs. The difference among the three types of PDEs is that the elliptic PDEs describes steady-state processes while the other two described time-evolving processes. Note that (3.114.3) do fit into the general form (3.115); because the curl-free

conditions- imposed by the left and right nonlinear eigenvectors associated with $\lambda_1 = 0$ -
guarantee that $\frac{\partial^2 p}{\partial x_i \partial x_j}$ and $\frac{\partial^2 p}{\partial x_j \partial x_i}$ are equal.

Before solving (3.114.3), one must ensure that the problem is well-posed. Issues related to the existence and uniqueness of analytic solutions of (3.114.3) are beyond the scope of the current investigation.

Note 3.10: Handling (3.114.3) in more than two variables is related to solving the consensus problem when considering (3.37). The response of the resulting MAS is delayed to Chapter 4.

3.2.11: Analyzing the C-S model using the proposed framework

Designed semi-linear protocols, in general, can satisfy (3.114.3) so easily and therefore are considered good candidates to solve the average consensus problem and others as well. For reasons that will be revealed through the sequel, we are interested in using the C-S model and hence this section is devoted to analyzing it considering the proposed protocol. A parameterized version of the C-S model presented in [64] is given as follows:

$$g_{ij}(x_j, x_i) = \frac{H}{\left(\gamma(x_j - x_i)^2 + \delta\right)^\beta} \quad (3.116)$$

where: $H, \gamma, \delta, \beta > 0$.

Using (3.108) and (3.116), (3.114.3) is satisfied with:

$$\frac{\partial^2 p}{\partial x_i \partial x_j} = \frac{\partial^2 p}{\partial x_j \partial x_i} = r \quad (3.116.1)$$

$$\frac{\partial^2 p}{\partial x_i^2} = \frac{\partial^2 p}{\partial x_j^2} = -r \quad (3.116.2)$$

where:

$$r = H \left\{ \frac{\delta + \gamma(1 - 2\beta)(x_2 - x_1)^2}{[\gamma(x_2 - x_1)^2 + 1]^{\beta+1}} \right\} \quad (3.116.3)$$

To understand the effect of parameters existing in (3.116), Figure 3.10 shows the control signal $\dot{x}_1 = u_1$ in (3.108) for different values of $\beta > 0$ when $H = \gamma = \delta = 1$. Interestingly, setting $\beta = 0.5$ results in a special response of u_1 .

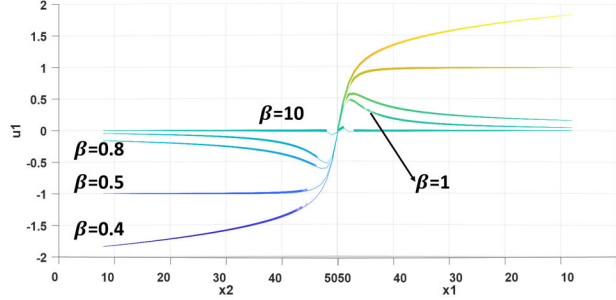


Figure 3.10. The control signal of (3.108) when g_{12} being the C-S model with different values of β .

Note 3.11: Observing the response of $-1 \leq u_1 \leq 1$ when $\beta = 0.5$ inspires us to reconsider the problem of average consensus with bounded input. But before that, we should understand the effects of the other parameters appearing in (3.116).

Let us- in a simplified way- investigate the limiting response of u_1 when $e_{12} = x_2 - x_1 \rightarrow \pm\infty$ and $e_{12} \rightarrow 0$ to have a glimpse about the contribution of the involved parameters. This could be done as follows:

$$\lim_{e_{12} \rightarrow \pm\infty} u_1 = \frac{H e_{12}}{\sqrt{\gamma e_{12}^2 + \delta}} = \pm \frac{H}{\sqrt{\gamma}} \quad (3.117)$$

So, by always keeping the ratio between H and $\sqrt{\gamma}$ equal to ± 1 , we can easily adjust the limits of the control signal as desired. Similarly, the limiting decaying rate of u_1 is related to its gradient (∇u_1). So:

$$\lim_{x_2 - x_1 \rightarrow 0^+} \nabla u_1 = \lim_{x_2 - x_1 \rightarrow 0^-} \nabla u_1 = \frac{H}{\sqrt{\delta}} (-\bar{x}_1 + \bar{x}_2) \quad (3.118)$$

From which the relation between H and δ is clear. Figure 3.11 shows the response of u_1 when different values of H, γ and δ are used with $\beta = 0.5$ in all cases.

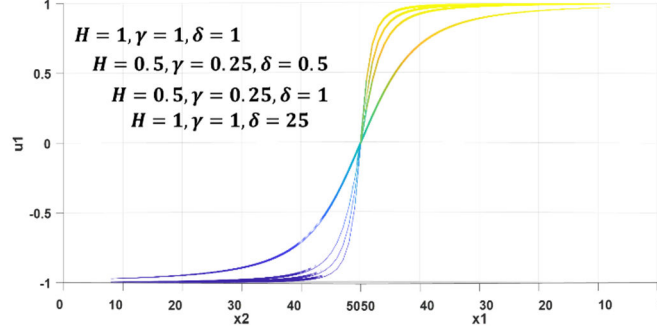


Figure 3.11. The control signal of (3.108) when g_{12} being the C-S model with different values of H, γ and δ are used with $\beta = 0.5$ in all cases. The values are given for the upper part of the graph from left to right.

3.2.11.1.1: The average consensus problem with bounded input

After understanding the effect of the parameters appearing in the C-S model on the response of u_1 in (3.108), it is time to propose a bounded control strategy whose bounds can easily be adjusted. The idea is to use a semi-linear protocol with $\vec{e} = \vec{1}$ and $\vec{v}^T = \vec{1}^T$ as right and left nonlinear eigenvectors associated with $\lambda_1 = 0$. Moreover, the g_{ij} functions used should be the C-S model with $\beta = 0.5$ and appropriately designed parameters such that $H/\sqrt{\gamma} = 1$. The resulting MAS dynamics are governed by the following system:

$$\dot{\vec{x}}(t) = -\frac{\varpi \mathcal{L}(\vec{x}) \vec{x}(t)}{\max_i d_{i,f}} = \vec{f}(\vec{x}) \quad (3.119)$$

where: $\pm\varpi$ are the symmetric control bounds and $\max_i d_{i,f}$ is the maximum in-degree of \mathcal{L}_f all known a-priori. Note that in (3.119), we mainly control the convergence rates of \vec{f} such that the control bounds are not debased. Note also that dividing by $\max_i d_{i,f}$ will normalize all control signals with respect to the largest signal. This will benefit in maintaining mainly $\vec{v}^T = \vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1 = 0$ on one

hand, and on the other to activate the desired control bounds $\forall \vec{x}, t \geq t_0$. The use of $\max_i d_{i,f}$ can be seen from the following relation between the control signal, its bounds, the in-degree of an agent and the maximum in-degree of agents in the MAS:

$$|u_i| \leq \varpi \frac{d_i(\vec{x})}{\max_i d_{i,f}}, \quad \forall i = 1, 2, \dots, N \quad (3.119.1)$$

where: $d_i(\vec{x}) \leq d_{i,f} \leq \max_i d_{i,f}$. So, $d_i(\vec{x})/\max_i d_{i,f} \leq 1, \forall \vec{x}$ and therefore the control limits are not violated using (3.119).

Note 3.12: Dividing each control signal by its in-degree, i.e., $d_{i,f}$, instead of $\max_i d_{i,f}$ will result in a left eigenvector associated with $\lambda_1 = 0$ other than $\vec{v}^T = c\vec{1}^T$ and thus the average consensus will not be achieved. Considering this, variants of (3.119) cannot be used to achieve other types of consensus with bounded inputs unless it is redesigned, if possible. Should the parameters H, γ and δ be selected according to some criteria in general, we may benefit from the knowledge of the associated potential field given as follows:

$$h(x_1, x_2) = -p(x_1, x_2) = -\frac{H}{\gamma} \sqrt{\gamma(x_1 - x_2)^2 + \delta} + const. \quad (3.120)$$

Additionally, solving the average consensus problem using bounded input with prescribed performance is also possible and it will follow the same steps presented previously.

The benefit of having (3.120) can be appreciated once we know that path planning, and consensus problems can be studied more thoroughly using the concept of potential fields as will be shown in other parts of this thesis.

3.2.12: Different Types of g_{ij} Functions

In general, these functions can be used to model the underlying communication network such as its signal strength, or to exert a calculated behavior among agents based on the

available context. One example of the latter case is the distance-dependent functions [64] which can be used to reduce the controller effort needed to achieve a certain objective. However, these functions should not create new equilibrium points unless it is desired to create systems with multiple invariant sets that might be used in realizing a mode-switching control strategy or like what is found in formation control, in general. The g_{ij} functions could be polynomials, rational or trigonometric functions, for examples. Consider also the following logistic activation function used in neural networks (NN) which can be useful in designing a consensus protocol as depicted partially in Figure 3.12:

$$g_{ij}(x_j, x_i) = \frac{1}{1 + e^{\lambda_{ij} \|x_j - x_i\|^2}} \quad (3.121)$$

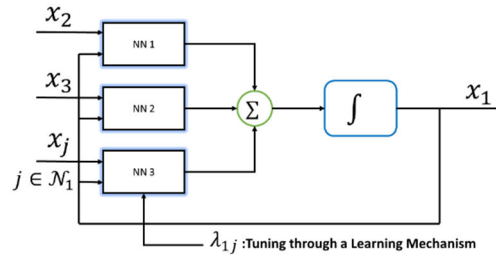


Figure 3.12. Visualization of g_{ij} being as an activation function used in NN.

♦ g_{ij} as Polynomials:

The general form can be given as follows:

$$g_{ij}(x_j, x_i) = w_{ij} \sum_{k=0}^n \alpha_k (x_j - x_i)^{2k} \quad (3.1.122)$$

where: α_i is positive coefficient with $\alpha_0 = 1$ and n is the desired order of the weighting function. It should be clear that, for the same initial values and order, if α_i is too large then g_{ij} may become infeasible while if it is too small then g_{ij} may become negligible.

In Chapter 4, various forms of g_{ij} will be used to realize behaviors directly related to the relative distance among agents. Next, simulation results obtained during the evaluation of the previous consensus protocols are presented.

3.3 Simulation Results

In this part, the results obtained while simulating various semi-linear and nonlinear consensus protocols over a complete graph of six agents are displayed. For the semi-linear protocols, several g_{ij} functions are used. All the results were run under the same initial conditions when applicable so that their effects on the MAS response can be made clear. A general undirected graph, shown in Figure 3.13, will also be used to demonstrate the generality of the proposed framework and to highlight the effect of the underlying communication network on the MAS behavior when the same protocol is used.

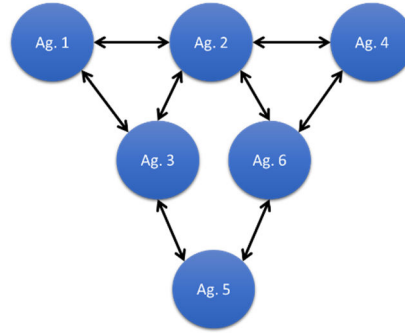


Figure 3.13. A general undirected graph comprising six agents.

3.3.1: Behaviors using semi-linear network protocols

The following examples were originally designed to work mainly with the semi-linear consensus protocol (3.4) and their applicability under the proposed framework can be easily checked. These examples deal with the distances among agents.

3.3.1.1: The g_{ij} as Rational Functions

The general form can be given as follows:

$$g_{ij}(x_j, x_i) = \frac{p(x_j - x_i)}{q(x_j - x_i)} \quad (3.123)$$

One example is given as follows:

$$g_{ij}(x_j, x_i) = \frac{w_{ij} \left\{ \gamma_1 (x_j - x_i)^a + e^{\theta(x_j - x_i)^2} \right\}}{(\gamma_2 (x_j - x_i)^2 + 1)^\beta} \quad (3.124)$$

where: $a = 2 \geq 0$ must be even, $\theta = -0.8 \leq 0$ and $\gamma_1 = 0.1, \gamma_2 = 0.02, \beta = 4 \geq 0$.

Observe the eigenvalue interlacing shown in Figure 3.14 under the distance-dependent protocol. This is in fact what justifies the slower states response and the lower control effort once compared to the results obtained using the fixed Laplacian matrix. Figure 3.15 shows the results of various types of means and the effect on the algebraic connectivity obtained using (3.124) and (3.39).

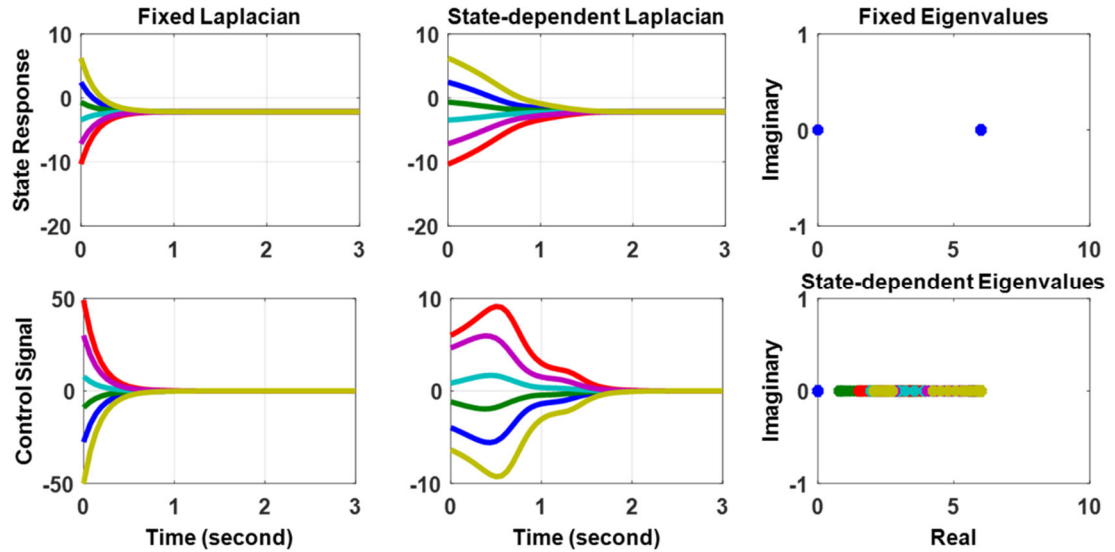
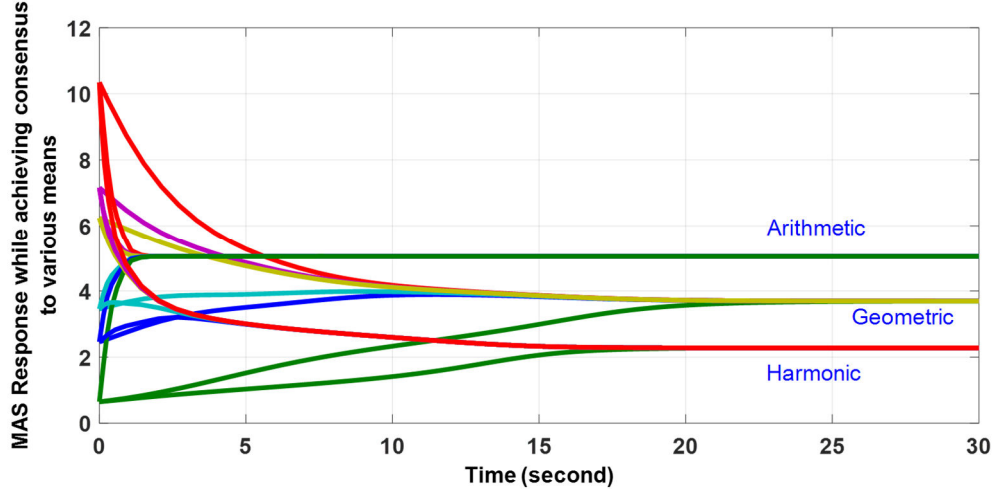
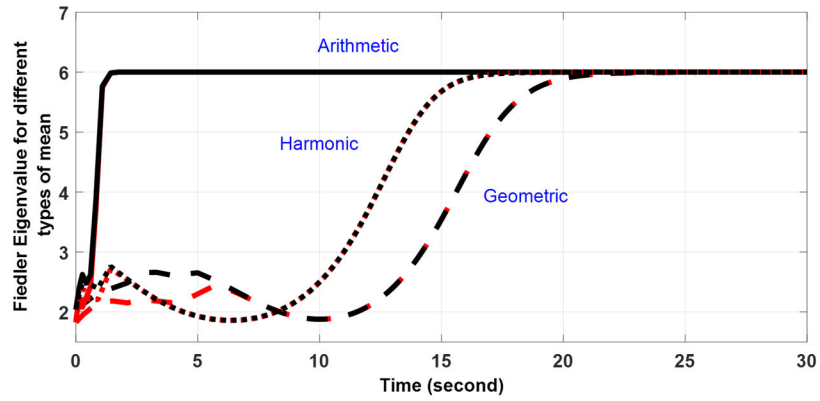


Figure 3.14. Arithmetic mean consensus among agents using protocol (3.5) utilizing (3.124). The first row from left to right: states responses, control signal and eigenvalues. The second row is their corresponding distance-dependent quantities, where eigenvalue interlacing is discussed in Remark 3.16.

Note 3.13: The effect of state-dependent weights can be seen by observing the slower MAS response when compared to the fixed weights. However, the control effort needed under state-dependent weights is considerably lower than that needed under fixed weights.



(a)



(b)

Figure 3.15. Different types of mean consensus among agents using protocol (3.39) utilizing (3.124). (a): states responses. (b): Algebraic connectivity $|\lambda_2|$, in red, with upper bound, in black, evaluated using (3.101) at each time instance for different types of means. The eigenvalue evolution is discussed in Remark 3.16.

Using the network graph depicted in Figure 3.13 and the same initial conditions, arithmetic, geometric and harmonic means are obtained using (3.39) utilizing (3.124) as shown in Figure 3.16. Even though (3.124) seems physically meaningless, it was provided to

emphasize the fact that the g_{ij} functions could generally be anything if the needed conditions are met. One famous example of g_{ij} being a rational function is the Cucker-Smale (C-S) flocking model [64].

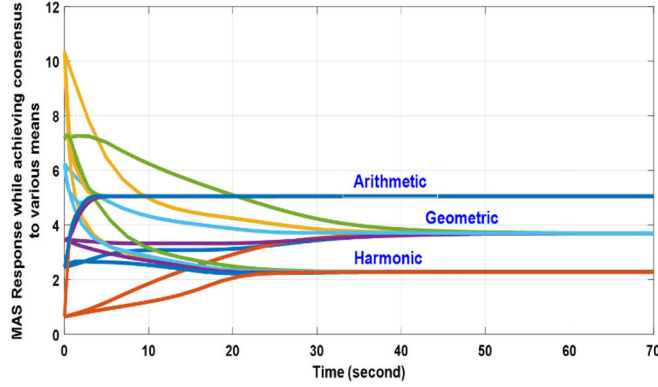


Figure 3.16. States responses while achieving consensus to different types of mean using protocol (3.39) utilizing (3.124) and the graph shown in Figure 3.13.

3.2.1.2: The g_{ij} as the Activation Function used in Neural Networks (NN)

One example can be given as shown in (3.121) where the simulation results are shown in Figure 3.17. The resulting convergence rates are lower than shown in Figure 3.16.

3.2.1.3: Clustering

The distance-dependent protocol (3.6)- utilizing (3.88) with $\beta = 1$ - is used to solve the simulation example found in [1] under the same initial conditions and network graph. This shows the applicability of the proposed framework to clustering application by which the original Laplacian matrix must be designed before applying the desired cooperative-competitive weights. The results are shown in Figure 3.18.

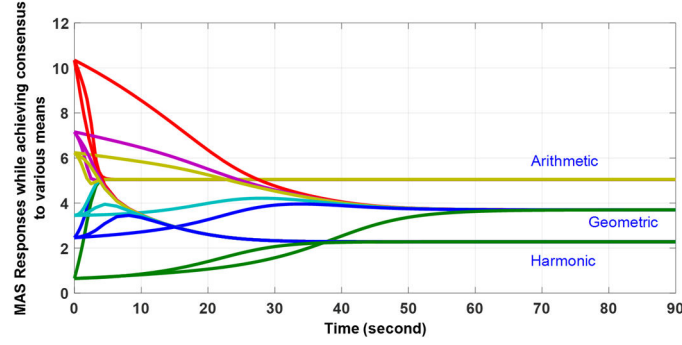


Figure 3.17. States responses while achieving different types of mean consensus among agents using protocol (3.39) utilizing (3.121).

3.2.1.4: The Prescribed Connectivity using semi-linear protocols

As stated before, adjusting the controller structure by introducing an additional term reflecting the worst allowed connectivity level helps improving the response of the MAS. Figure 3.19.a shows the effect of the prescribed connectivity- on the response of the MAS- despite its tiny values, while Figure 3.19.b justifies the necessity of using a prescribed connectivity if large initial conditions are expected during the interaction of agents. It is worth noting that by introducing the concept of prescribed connectivity, the resulting g_{ij} functions are treated as controller gains rather than as signal strength models of the underlying communication network. Doing so helps in avoiding the estimation of the domain of attraction of the equilibria set. However, the agents should be initially connected or another protocol must be run to make them connected, like the nearest neighbor protocol found in [117], (3.72) or (3.75) for examples.

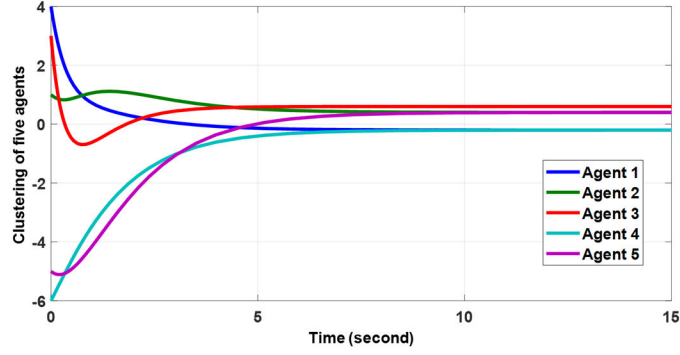
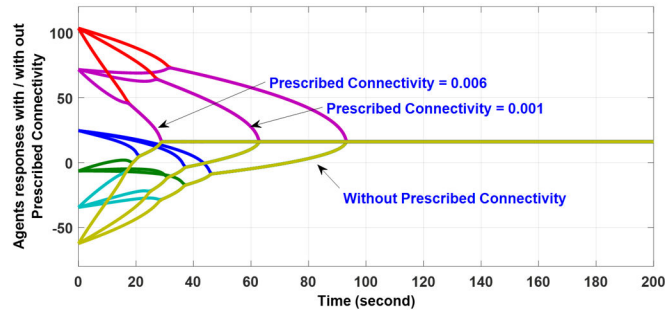


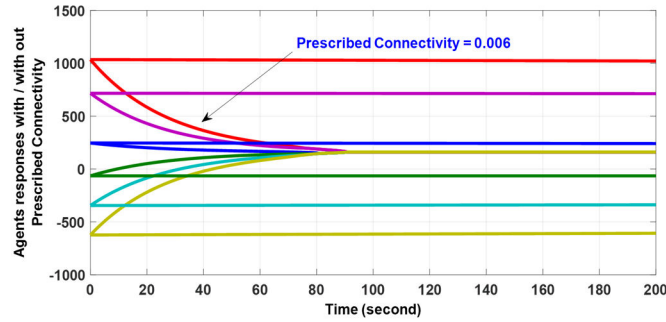
Figure 3.18. Clustering example available in [1] solved here using protocol (3.6) utilizing (3.88) with $\beta = 1$.

3.3.2: Behaviors using nonlinear network protocols

In this part, the simulation results obtained while simulating the nonlinear protocols (3.54) and (3.67.15) are presented. Under the proposed framework, it is obvious that both (3.51) and (3.54) refer to the same family of nonlinear protocols, namely: Class 2, whose results are shown in Figure 3.20. Note that achieving the harmonic mean was not reported in [56] which makes the results reported therein a special case of the proposed framework presented in this thesis.



(a)



(b)

Figure 3.19. MAS response with or without prescribed connectivity levels over a complete graph and under (a): relatively small initial conditions and (b): large initial conditions cause the expected time for consensus to occur as $t \rightarrow \infty$.

The results of using the nonlinear protocol (3.54) over the graph shown in Figure 3.13 are shown in Figure 3.21. Notice the difference from the ones shown in Figure 3.20.

The simulation results of protocol (3.67.15) are shown in Figure 3.22 over a complete graph.

The effect of the underlying communication network and the protocol- mainly the g_{ij} functions- used can be understood by comparing the results shown in figures 1.15, 1.16 and 1.17 while fixing the initial conditions. Notice that in figures 1.15 and 1.17, the same complete graph was used however with different g_{ij} functions while similar g_{ij} functions were used to obtain figures 1.15 and 1.16 however with different network topologies.

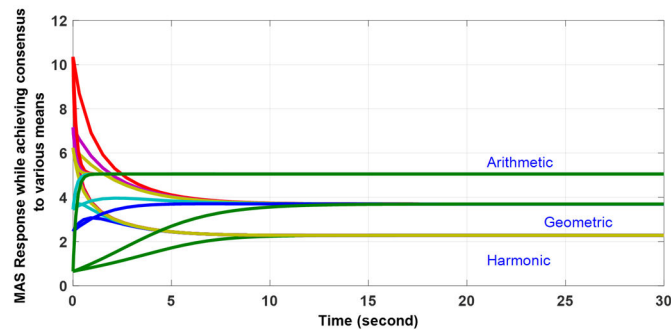


Figure 3.20. States responses while achieving different types of mean consensus among agents using nonlinear protocol (3.54) over a complete graph.

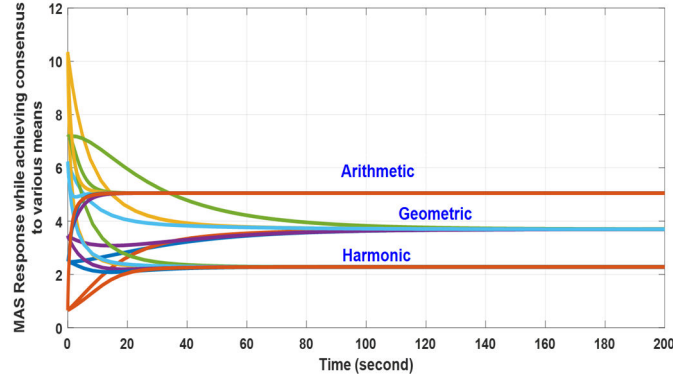


Figure 3.21. States responses while achieving different types of mean consensus among agents using nonlinear protocol (3.54) over the graph shown in Figure 3.13.

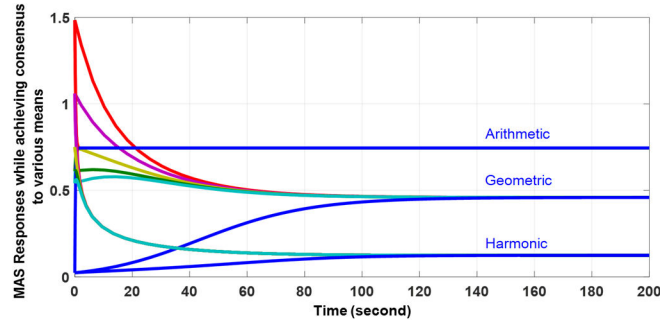


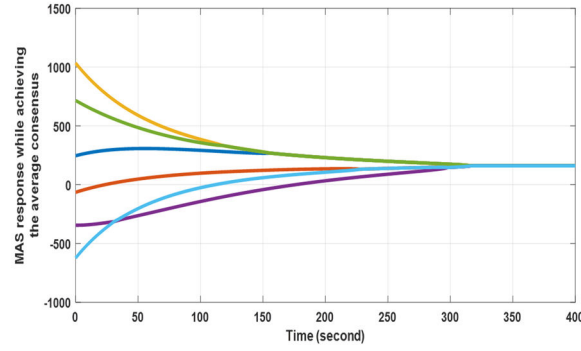
Figure 3.22. States responses while achieving different types of mean consensus among agents using nonlinear protocol (3.67.15) over a complete graph. To achieve the harmonic mean, the convergence rate was 30 times larger than what was used with the other two types.

Clearly, the network shown in Figure 3.13 has a lower algebraic connectivity compared to the one obtained using a complete graph and hence the slower rate of consensus obtained using the undirected graph shown in Figure 3.13. Interestingly, using the complete graph and protocol (3.39) utilizing (3.121) gave a slower consensus rate once compared to the case where the incomplete graph- shown in Figure 3.13- was used with different g_{ij} functions. Despite these differences, the same value of consensus for each mean type was reached in the three cases. The same discussion is also valid for the nonlinear protocols as shown in figures 1.20 and 1.21. Consequently, it should be kept in mind that there might be different combinations of network topologies and protocols that can be used to achieve

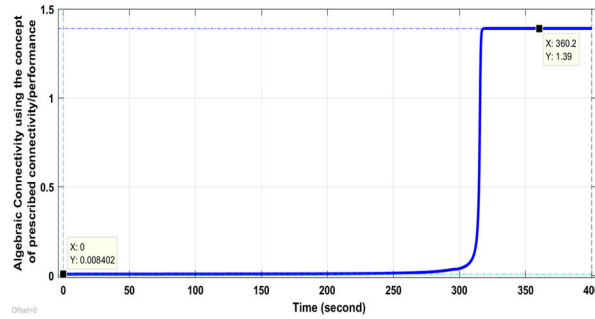
the same behavior and it is up to the designer to decide what to use provided that the design requirements are met.

Note 3.14: The differences in colors of the compared figures are insignificant; since the initial conditions for each agent were the same.

Figure 3.23.a shows the MAS response over the graph shown in Figure 3.13 under the semi-linear protocol utilizing the C-S model with $\beta = H = \gamma = \delta = 1$ and a prescribed connectivity of 0.006 while Figure 3.23.b shows the evolution of $|\lambda_2|$ where (3.81.4) and (3.81.5) served as lower and upper values, respectively.



(a)



(b)

Figure 3.23. (a): States responses while achieving the average consensus among agents using the graph shown in Figure 3.13 utilizing a semi-linear protocol with C-S model and a prescribed connectivity of 0.006 with $\beta = H = \gamma = \delta = 1$. (b): Point-wise calculation of $\lambda_2(\mathcal{L}(\vec{x}))$ to have an insight about the propagation of the algebraic connectivity of the MAS as will be justified in Part II. See Remark 3.16.

Figure 3.24 shows the MAS response over the graph shown in Figure 3.13 under the action of the semi-linear protocol (3.68) while achieving consensus using a bounded control signal as given in (3.105). The protocol (3.68) utilizing the sign function (top) and the tanh function (bottom)- with $\varpi = 0.3$ and $\alpha = 0.1$ - were used where the two black dashed lines represent the constraints. Note the differences in the control signals and the consensus values where the one achieved using the sign function was 2.066 and 2.054 using the tanh function and compare these to the arithmetic mean value which was 1.6046. The constraints were slightly violated just before the consensus and that is why their consensus values were different. Note that since the $\vec{1}$ vector is not a left eigenvector associated with $\lambda_1 = 0$, the average consensus was not achieved in both cases. The consensus protocol realized using the sign function introduced chattering which is undesirable especially in a reference signal generator whose other dynamical systems are going to track. Thus, the benefit of using an approximate version of the sign function, namely the tanh function, is obvious.

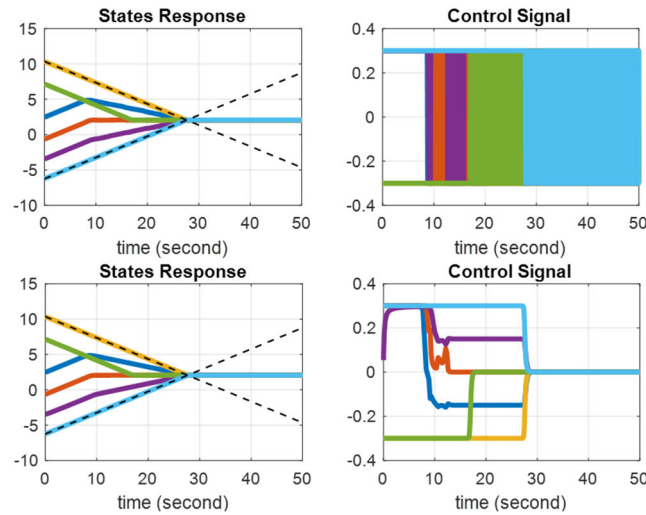


Figure 3.24. States responses while achieving consensus among agents using the graph shown in Figure 3.13 utilizing (3.68) with sign function (top) and tanh function (bottom), with $\varpi = 0.3$ and $\alpha = 0.1$, where the two black dashed lines represent the constraints.

The average consensus protocol with bounded inputs, i.e., (3.119), will be utilized in Part II to design mainly bounded connectivity-preserving protocols.

3.4 Conclusion

In this chapter, a unifying framework for synthesizing continuous-time distributed semi-linear and nonlinear state-dependent consensus network protocols in a systematic way was presented. In this framework, the use of both the first integrals and nonlinear eigenvalues was set into interplay such that the needed protocols can be designed. This design process kept a clear link to what is available in the literature regarding the static-gain protocols usually related to constant Laplacian matrices. General stability proofs using M -matrices were also presented which make designing new protocols with guaranteed stability a straightforward process under the proposed framework. Several network protocols available in the literature were shown to be special cases of the general classes proposed herein. Besides stability, performance issues were also handled where the concepts of prescribed connectivity and prescribed performance were introduced. Considering bounded inputs in conjunction with finite-time consensus problem was also detailed. The relation between consensus protocols and potential fields was highlighted. The utilization of the proposed framework in designing new protocols that will generate interesting behaviors of the MAS will be covered in the next chapter of this thesis.

CHAPTER 4

State/Distance-dependent Behaviors

In Chapter 3, a unifying framework for distributed semi-linear and nonlinear state-dependent protocols to control multiagent systems over communication networks was presented. The building blocks of the proposed framework were also covered in Chapter 3 where its generality was demonstrated through comparing it to some major results available in the related literature. In this chapter, the applicability of the framework in producing several motion-related behaviors that govern the interaction among the connected agents is demonstrated through totally simulation examples where proofs of invariance, stability and convergence are presented when needed. Using state-dependent parameterization to control the switching between the various behaviors is presented. The multitude of primitive behaviors are used to build more sophisticated behavioral banks that reside in each agent permitting each agent to choose or follow the selected behavior. Designing connectivity-preserving protocols is also addressed. The result is a sophisticated distributed coordination motion planner.

4.1 The framework Utilization in state-dependent and motion-related behaviors realization

In Chapter 3, we have developed a semi-linear protocol to control the behavior of a MAS system with fixed number of agents whose dynamics are modeled as follows:

$$\dot{x}_i(t) = u_i(t) \in \mathfrak{R}, \quad \forall i = 1, 2, \dots, N \quad (4.1)$$

with u_i , i.e., the network protocol, given as follows:

$$u_i = - \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{x_i - x_j\} \quad (4.2)$$

$$g_{ij}(x_i, x_j) = \begin{cases} g_{ij}(x_i, x_j) > 0, & j \in \mathcal{N}_i \\ 0, & j \notin \mathcal{N}_i \end{cases}$$

where: $g_{ij}(x_i, x_j): \mathfrak{R}^2 \rightarrow \mathfrak{R}_{>0}$ is a general and usually bounded weighting vector-valued functional and $j \in \mathcal{N}_i$ denotes that the j^{th} agent is a neighbor of the i^{th} agent. Let the global state vector $\vec{x} = [x_1, x_2, \dots, x_N]^T \in \mathfrak{R}^N$, thus, (4.1) can be rewritten for short- by dropping the explicit time dependency- as follows:

$$\dot{x}_i = - \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{x_i - x_j\} = -x_i \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) + \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) x_j \quad (4.3)$$

Simplifying (4.3) more, yields:

$$\dot{x}_i = -x_i d_i + \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) x_j = u_i \quad (4.4)$$

where: $d_i = \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j)$ is the in-degree of the i^{th} agent. Taking $\mathcal{D} = \text{diag}([d_1, d_2, \dots, d_N])$, we can write the global dynamics of the connected MAS as follows:

$$\dot{\vec{x}}(t) = -(\mathcal{D}(\vec{x}) - \mathcal{A}(\vec{x}))\vec{x} = -\mathcal{L}(\vec{x})\vec{x}(t) = \vec{f}(\vec{x}) \quad (4.5)$$

where: $\vec{f}(\vec{0}) = \vec{0}$ and $\mathcal{A} = [g_{ij}]$ is the adjacency matrix associated with the graph resembling the available communication network among agents, and $\mathcal{L}(\vec{x})$ - or \mathcal{L}_s for short- is the resulting state-dependent Laplacian matrix.

To achieve clustering among agents in a MAS, the consensus protocol (4.4) is modified as follows:

$$\dot{x}_i = -x_i d_i + \sum_{j \in \mathcal{N}_i} d_{ij}^x g_{ij}(x_i, x_j) x_j = u_i \quad (4.6)$$

where: $d_{ij}^x \in \mathcal{R}$ is the fixed competitive or cooperative weight that is usually given as a real number whose sign is $\{-1, 1\}$, respectively. This weight will convert the unsigned network graph into a signed graph [87] which is a special case of a gain graph. To maintain the in-degree, i.e., d_i , of the original network graph, it is required to have $d_{ij}^x = 1/d_{ji}^x = \sigma_j/\sigma_i$ where σ_i is the desired behavior of the i^{th} agent. Clearly, if $\sigma_i = \sigma_j$, then both agents are in the same cluster. This relation will preserve the stability of the resulting MAS under cluster consensus. Mathematically, the relation between both graphs as reflected on their corresponding Laplacian matrices, so the resulting MAS dynamics can be given as follows:

$$\dot{\vec{x}}(t) = -(D^{-1}\mathcal{L}(\vec{x})D)\vec{x}(t) = \vec{f}(\vec{x}) \quad (4.7)$$

where: $D = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_N)$.

Practically speaking, certain applications demand the usage of different communication networks for different states; may be due to security issues or simply because some agents do not require direct measurements of other agents of specific states. Nevertheless, by extending the agents states dimensions from \mathfrak{R} to \mathfrak{R}^3 for example, we may write the overall MAS dynamics in 3-D as follows:

$$\dot{\vec{z}}(t) = - \begin{bmatrix} \alpha_1 G_1(\vec{z}_1) \mathcal{L}_1(\vec{z}_1, \vec{z}_2, \vec{z}_3) \vec{z}_1(t) \\ \alpha_2 G_2(\vec{z}_2) \mathcal{L}_2(\vec{z}_1, \vec{z}_2, \vec{z}_3) \vec{z}_2(t) \\ \alpha_3 G_3(\vec{z}_3) \mathcal{L}_3(\vec{z}_1, \vec{z}_2, \vec{z}_3) \vec{z}_3(t) \end{bmatrix} \quad (4.8)$$

where: $\vec{z} = [\vec{z}_1^T, \vec{z}_2^T, \vec{z}_3^T]^T$ and $\vec{z}_i = [x_{1i}, x_{2i}, \dots, x_{Ni}]^T$ while $\vec{x}_j = [x_{j1}, x_{j2}, x_{j3}]^T, j = 1, \dots, N, i = 1, 2, 3$. The state-dependent Laplacian matrices $\mathcal{L}_i, \alpha_i > 0$ and G_i may not be similar. The $G_i(\vec{z}_i)$ are given as follows:

$$G_i(\vec{z}_i) = \begin{bmatrix} g_i(x_{1i}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & g_i(x_{Ni}) \end{bmatrix} \quad (4.9)$$

Where $g_i(x_{ji})$ is given as $1, x_{ji}, x_{ji}^2$ and x_{ji}^{1-p} to achieve arithmetic, geometric, harmonic and power mean-of-order- p , respectively.

In this chapter, the proposed framework is utilized to generate various behaviors of interacting spatial-agents using mainly the semi-linear protocol with g_{ij} being distance-dependent. Note that the g_{ij} functions used may represent a communication link model or a part of the behavior controller. Each behavior is assumed to be fully dominant, i.e., no combinations among different behaviors are allowed. This dominance ensures that the null-space of the corresponding matrix representing a behavior is preserved. Mathematically, this can be seen from the fact that the addition, for example, of two general matrices cannot guarantee the preservation of their null-spaces if they were distinct. Note that each behavior may enjoy different spatial dimensions as shown in (4.8), for example.

Having several behaviors residing in a local behavior bank- which is identical in each agent- as shown in Figure 4.1, an intelligent agent can decide individually or collectively how to react to different situations or contexts by switching among these embedded behaviors. However, as far as this thesis is concerned, the switching mechanism among the proposed set of behaviors is designed to be time-dependent where a central monitoring unit, a human-in-the-loop for example, issues the switching command to all agents. Though, practically speaking, this mechanism is not suitable for various situations, it is used herein to convey the concept, mainly. Other behavior-selection mechanism that are event-driven (including self-triggering), Finite-State-Automata [129] or sequential-based mechanisms can also be utilized.

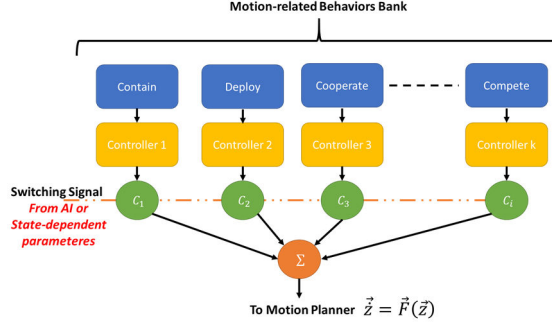


Figure 4.1. An agent behavior-bank consists of k -motion-related behaviors.

4.1.1: Consensus using the common neighborhood approach

The merits behind this behavior are the ability of agents interaction not only at the edge level but also on the neighborhood level that might represent a preferred selection of an operator controlling the MAS behavior. An example is depicted in Figure 4.2 in which an operator is creating two complete neighborhoods and a partial one. We refer to a complete neighborhood as such all agents are fully-connected like \mathcal{N}_1 and \mathcal{N}_3 in Figure 4.2 while \mathcal{N}_2 is a partial neighborhood where these types can be decided dynamically at agents level or imposed by the operator using a supervisory front-end software, for instance. It is permissible for neighborhoods to overlap.

Having the neighborhoods selected- as in Figure 4.2-, it is essential to decide next what the task is. Let us consider achieving the average consensus among the agents. Therefore, the corresponding control signals are given as follows:

$$\begin{aligned}
 \dot{x}_1 &= g_{123}\{x_2 + x_3 - 2x_1\} \\
 \dot{x}_2 &= g_{123}\{x_1 + x_3 - 2x_2\} + g_{246}\{x_4 + x_6 - 2x_2\} \\
 \dot{x}_3 &= g_{123}\{x_1 + x_2 - 2x_3\} + g_{35}\{x_5 - x_3\} \\
 \dot{x}_4 &= g_{246}\{x_2 + x_6 - 2x_4\} \\
 \dot{x}_5 &= g_{53}\{x_3 - x_5\} + g_{56}\{x_6 - x_5\}
 \end{aligned} \tag{4.10}$$

$$\dot{x}_6 = g_{246}\{x_2 + x_4 - 2x_6\} + g_{65}\{x_5 - x_6\}$$

where:

$$\begin{aligned} g_{123} &= 1/[(x_1 - x_2)^2 + (x_1 - x_3)^2 + (x_2 - x_3)^2 + 1] \\ g_{246} &= 1/[(x_2 - x_4)^2 + (x_2 - x_6)^2 + (x_4 - x_6)^2 + 1] \\ g_{35} &= 1/[(x_5 - x_3)^2 + 1] \\ g_{65} &= 1/[(x_5 - x_6)^2 + 1] \end{aligned} \tag{4.10.1}$$

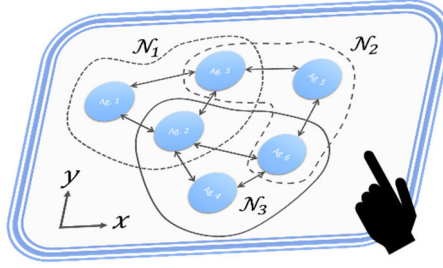


Figure 4.2. MAS with multiple neighborhoods created by the operator.

Solving (3.33.1) and (3.33.2), the MAS given by (4.10) is indeed capable of achieving the average consensus as desired $\forall \vec{x}$. The results shown in Figure 4.3 can be used to compare the response of (4.10) with what was shown in Figure 3.23.a where both systems used a prescribed connectivity of 0.006 and the same underlying network. The slower version of (4.10) is due to the additional elements appearing at the denominator of g_{123} - acting on the complete neighborhood level- compared to g_{65} acting at the edge level, for example.

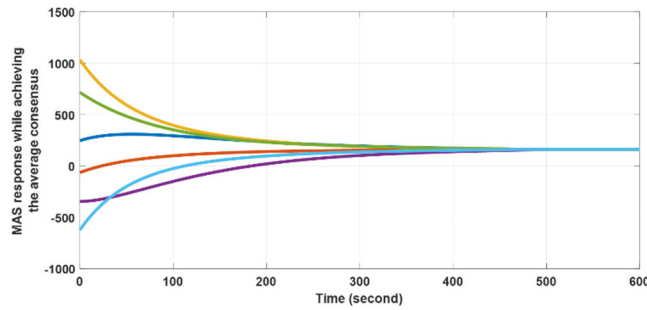


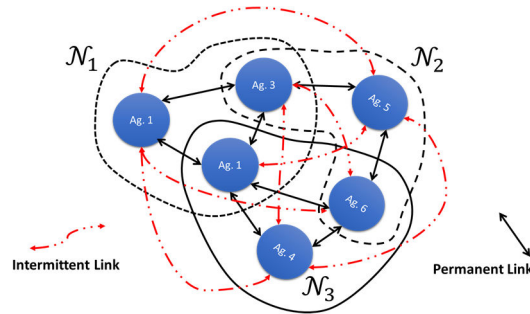
Figure 4.3. Response with prescribed connectivity 0.006 using the concept of multi-neighborhood. Compare it with Figure 3.23.a.

Having random initial conditions, there might be a possibility of physical neighboring between two agents that are not originally designed to be neighbors on the graph associated with the communication network. Therefore, enabling dynamic edge creation is a plus for the overall cooperative behavior of MAS especially if it consists of spatial agents. Figure 4.4.a shows a possible integration among permanent links forced by the designer and intermittent links dynamically created on the fly.

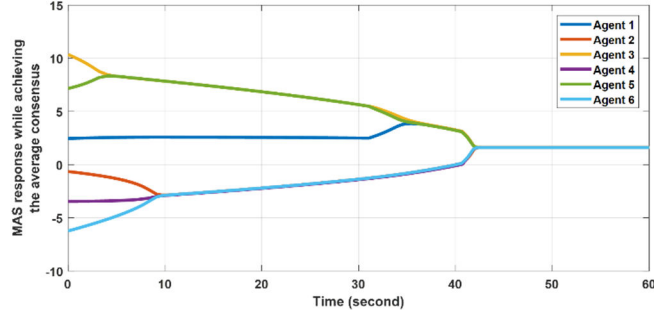
Interestingly, the intermittent links creation could be due to physical nearness of agents and therefore can be realized based on onboard sensory measurements, for example. The effect of the intermittent links on the overall performance of MAS can be seen through comparing figures 2.4.b and 2.4.c where the effect of the intermittent link established between agents 1 and 5 is clear. This can be read from **Theorem 3.7** available in Chapter 3. Although the creation of intermittent links could be controlled in many ways, the ones used in Figure 4.4.a are created using the following function:

$$g_{ij} = \delta \left(\left[(x_j - x_i)^2 + 1 \right] - TH \right) / \left[(x_j - x_i)^2 + 1 \right] \quad (4.11)$$

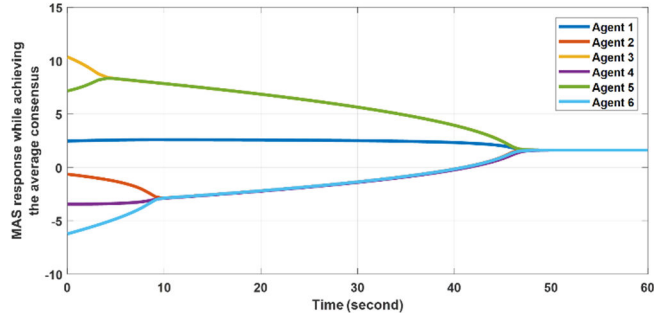
where: $\delta(\cdot)$ is the unit-step function and TH is a threshold set as 0.1.



(a)



(b)



(c)

Figure 4.4. A MAS with permanent and intermittent link connecting agents within multiple neighborhoods and no prescribed connectivity. (a): illustration. (b): response with intermittent links enabled with threshold equals 0.1 and (c): disabled.

Let all agents be well-informed about each other, i.e., agents relate to each other over a complete undirected graph, then there will be one global neighborhood which is also complete. In such a case, the overall MAS dynamics can be given as follows:

$$\begin{aligned}
 \vec{x}(t) &= -\alpha \left(\frac{1}{1 + \sum_{i,j,i \neq j}^N (x_j - x_i)^2} + \sigma \right) \mathcal{L}_f \vec{x}(t) \\
 &= -\alpha (c(\vec{x}) + \sigma) \begin{bmatrix} h_1 \\ \vdots \\ h_N \end{bmatrix} = -\alpha (c(\vec{x}) + \sigma) \vec{h}(\vec{x}) = \vec{f}(\vec{x})
 \end{aligned} \tag{4.12}$$

where: $\sum_{i,j,i \neq j}^N (x_j - x_i)^2$ denotes the summation of all possible and unique combinations n from \vec{x} , \mathcal{L}_f is the fixed Laplacian matrix imposed originally by the designer, and $\alpha > 0$

is a fixed and real scalar controlling the convergence rate and the prescribed connectivity is denoted by σ . Figure 4.5 shows the results obtained using (4.12) when the graph shown in Figure 4.4.a is complete. Note that in (4.12), $\alpha = 1$ was used with prescribed connectivity $\sigma = 0.006$. Note also the hump in the control signal right before consensus is achieved. The effect of the global neighborhood can be seen by comparing Figure 3.19.a with Figure 4.5.

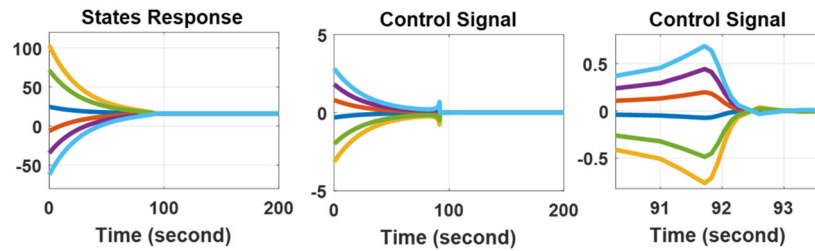


Figure 4.5. A MAS response while achieving the average consensus under the global neighborhood.

Let us reconsider the problem described in Figure 4.2 and solve the average consensus problem in an actual 2-D working space assuming a global neighborhood. This can be simply done by introducing another dynamical system in the other direction, say the y -direction, as given in (4.8). The results are shown in Figure 4.6. Note that agents responded in a linear fashion due to equal efforts, i.e., control signal magnitude. Note also that the control signal in both Cartesian coordinates were totally decoupled as suggested in (4.8).

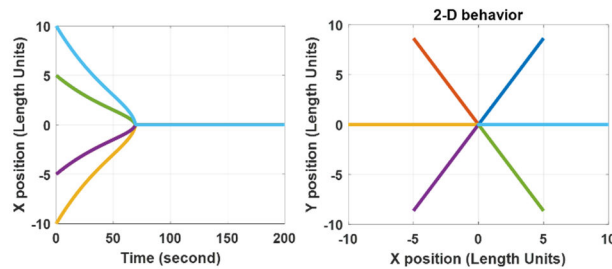


Figure 4.6. A MAS response under the global neighborhood while achieving the average consensus on 2-D working space.

Note 4.1: In (4.12), the relation between the evolution of the MAS eigenvalues under state-dependent convergence rate and the constant eigenvalues of \mathcal{L}_f - imposed by the designer- is interesting, which motivates the following sub-section.

4.1.1.1: Linear dynamics contribution in the global neighborhood response

In Chapter 3, **Theorem 3.7** is used to present the concept of prescribed connectivity mainly by scaling the desired \mathcal{L}_f using a scalar, namely: σ , and then add a states-dependent weight for each edge in \mathcal{L}_f through the used g_{ij} functions. So, the previous concept of prescribed connectivity is operating at the edge level and not on the MAS level. Inspired by (4.12), the whole \mathcal{L}_f matrix can be scaled by a well-designed scalar functional such that the evolution of the connectedness of agents in MAS is pre-controlled or pre-known under state-dependence.

Recalling **Theorem 3.8**, the *modified Fiedler eigenvalue* of (4.12) for a system of N single-integrator systems can be given as follows:

$$\lambda_2(\vec{x}) = 2\alpha c^2(\vec{x}, t) \|\vec{v}_{2f}^T \vec{h}\|_2^2 - \alpha c(\vec{x}, t) \lambda_{2f} \quad (4.13)$$

where: $c(t, \vec{x}) > 0$ is a scalar function that might depend on both time and states, and λ_{2f} is the Fiedler eigenvalue of the fixed Laplacian matrix \mathcal{L}_f . Because time is not supported yet in the proposed framework, we assume that c is time-independent, i.e., $c(\vec{x})$. Which leads to the following:

Proposition 4.1: The *modified Fiedler eigenvalue* of (4.12) for a system of N single-integrators is given by (4.13).

Proof: The Jacobian of (4.12) is given as follows assuming $\sigma = 0$:

$$\frac{\partial \vec{f}}{\partial \vec{x}} = \begin{bmatrix} h_1 \nabla c(\vec{x}) + c(\vec{x}) \nabla h_1 \\ \vdots \\ h_N \nabla c(\vec{x}) + c(\vec{x}) \nabla h_N \end{bmatrix} \quad (4.13.1)$$

where: ∇ is the row vector gradient operator. Solving (3.33.1) and (3.33.2) to force a symmetric Jacobian resulted in the following condition:

$$\nabla c(\vec{x}) \cdot \vec{1} = 0 \quad (4.13.2)$$

Therefore, the Jacobian will act as a point-wise singular *M-matrix* whose sign nearby the equilibria set can be examined as will be shown in the coming subsection. Note that for a complete graph where agents are all connected to each other, we have:

$$\sum_{i=1}^N h_i = 0 \quad (4.13.3)$$

Having a symmetric Jacobian, we may use (3.96) so that the second smallest eigenvalue of (4.12) is given as follows:

$$\lambda_2(\vec{x}) = \frac{\vec{v}_2^T \left(\frac{\partial \vec{f}}{\partial \vec{x}} \right) \vec{v}_2}{\vec{v}_2^T \vec{v}_2} \quad (4.13.4)$$

where: $\vec{v}_2 = g(c(\vec{x}), \vec{x}) \vec{v}_{2f}$ with $\vec{v}_{2f}^T \vec{v}_{2f} = 1$ and $g(c(\vec{x}), \vec{x}) > 0$ is a general scalar function of the nonlinear scaling $c(\vec{x})$.

In general, a nonlinear system might have many eigenvalues- possibly infinitely- which are not unique [99]. Each eigenvalue among those might have several eigenvectors as well [99]. Being interested in observing the signature of the linear mode, i.e., λ_{2f} , we choose $\vec{v}_2 = c(\vec{x}) \vec{v}_{2f}$ with $\vec{v}_{2f}^T \vec{v}_{2f} = 1$, among many, as a nonlinear unity left eigenvector- yet to be verified- associated with $\lambda_2(\vec{x})$. This relation between the eigenvectors is desired such that $c(\vec{x}) > 0$ will only change the length of the vector \vec{v}_{2f} and not its direction so that the contribution of the linear dynamics in the MAS response can be extracted. So:

$$\begin{aligned}
\lambda_2(\vec{x}) &= \alpha \vec{v}_{2f}^T \left(\begin{bmatrix} h_1 \nabla c(\vec{x}) + c(\vec{x}) \nabla h_1 \\ \vdots \\ h_N \nabla c(\vec{x}) + c(\vec{x}) \nabla h_N \end{bmatrix} \right) \vec{v}_{2f} \\
&= \alpha \vec{v}_{2f}^T \begin{bmatrix} h_1 \nabla c(\vec{x}) \\ \vdots \\ h_N \nabla c(\vec{x}) \end{bmatrix} \vec{v}_{2f} - \alpha c(\vec{x}) \vec{v}_{2f}^T \mathcal{L}_f \vec{v}_{2f} \\
&= \alpha \vec{v}_{2f}^T \begin{bmatrix} h_1 \nabla c(\vec{x}) \\ \vdots \\ h_N \nabla c(\vec{x}) \end{bmatrix} \vec{v}_{2f} - \alpha c(\vec{x}) \lambda_{2f} \\
&= \alpha \vec{v}_{2f}^T \vec{h} \nabla c(\vec{x}) \vec{v}_{2f} - \alpha c(\vec{x}) \lambda_{2f}
\end{aligned} \tag{4.13.5}$$

Now, knowing that the Jacobian is symmetric and after calculating its needed elements, we find that $\forall i, j$:

$$\begin{aligned}
\frac{\partial c}{\partial x_i} h_i &= 2c^2(\vec{x}) h_i^2 \\
\frac{\partial c}{\partial x_j} h_i &= 2c^2(\vec{x}) h_i h_j
\end{aligned} \tag{4.13.6}$$

Therefore, (4.13.5) can be rewritten as follows:

$$\lambda_2(\vec{x}) = 2\alpha c^2(\vec{x}) \vec{v}_{2f}^T \vec{h} \vec{h}^T \vec{v}_{2f} - \alpha c(\vec{x}) \lambda_{2f} \tag{4.13.7}$$

or:

$$\lambda_2(\vec{x}) = 2\alpha c^2(\vec{x}) \|\vec{v}_{2f}^T \vec{h}\|_2^2 - \alpha c(\vec{x}) \lambda_{2f} \tag{4.13.8}$$

which equals (4.13).

■

Note that the $c(\vec{x})$ used in (4.12) does indeed satisfy (4.13.2). Now, to prohibit the scalar function $c(\vec{x})$ from changing the direction of \vec{v}_{2f} , the latter should be selected from the generalized eigenspace associated with λ_{2f} whose algebraic multiplicity over a complete undirected graph is greater than 1. This should be done at the initial time, t_0 , using the

known initial conditions. Doing so, \vec{v}_{2f} will, almost, be perpendicular to $\nabla c(\vec{x})$ and therefore we will have:

$$\lambda_2(\vec{x}) \approx -\alpha c(\vec{x}) \lambda_{2f} \quad (4.14)$$

Otherwise, (4.12) is true.

Note 4.2: Finding all possible values of $\lambda_2(\vec{x})$ is beyond the scope of this work. Notice that (4.13) should not give the indication that the algebraic connectivity will be lost; because this relation is introduced to investigate the contribution of λ_{2f} in the MAS response.

Remark 4.1: In Chapter 3, the linear modes contributions found in the figures were investigated by calculating, in a point-wise fashion, the eigenvalues of $\mathcal{L}(\vec{x})$ in the semi-linear protocols using the MATLAB function $\text{eig}(\mathcal{L}(\vec{x}))$. More details will be provided later.

Remark 4.2: Investigating the linear signatures in nonlinear systems can be found in many fields, e.g., [130] in which the linear signature is used to interpret the plasma turbulence using pseudospectra.

4.1.1.2: Stability analysis of global neighborhood

It was shown in (3.83) that if the Jacobian is a negative singular M -matrix outside the equilibria set then this equilibria-set will be attractive, and the MAS is stable. Therefore, the sign of (4.13.1) is examined as follows:

$$\frac{\partial \vec{f}}{\partial \vec{x}} = 2\alpha c^2(\vec{x}) \vec{h} \vec{h}^T - \alpha c(\vec{x}) \mathcal{L}_f \quad (4.15)$$

So:

$$\vec{x}^T \frac{\partial \vec{f}}{\partial \vec{x}} \vec{x} = 2\alpha c^2(\vec{x}) \vec{x}^T \vec{h} \vec{h}^T \vec{x} - \alpha c(\vec{x}) \vec{x}^T \mathcal{L}_f \vec{x} \quad (4.15.1)$$

$$= 2\alpha^{-1}\vec{x}^T \vec{f} \vec{f}^T \vec{x} + \vec{x}^T \vec{f}$$

And we have:

$$\vec{x}^T \vec{f} = -c(\vec{x})R^2 \quad (4.15.2)$$

where:

$$R^2 = \sum_{i,j,i \neq j}^N (x_j - x_i)^2 \quad (4.15.3)$$

So, we have:

$$0 < c(\vec{x}) = \frac{1}{1 + R^2} \leq 1 \quad (4.15.4)$$

and therefore, it will compress the length of the vector \vec{v}_{2f} . We may write (4.15.1) as follows:

$$\vec{x}^T \frac{\partial \vec{f}}{\partial \vec{x}} \vec{x} = 2\alpha^{-1} \frac{R^4}{(1 + R^2)^2} - \frac{R^2}{1 + R^2} = \frac{[(2 - \alpha)R^2 - \alpha]R^2}{\alpha(1 + R^2)^2} \quad (4.15.5)$$

Now, evaluating (4.15.5) in the near vicinity of the equilibria set, i.e., $\mathfrak{S} = \{\vec{x} \in \mathbb{R}^N | \vec{x} = q\vec{1}\}, q \geq 0$, results in a negative value, and therefore the equilibria set is attractive, and the MAS is stable.

Before concluding this section, let us first revisit the concept of the global neighborhood which is defined as the neighborhood of all complete neighborhoods. A generalization of this type of neighborhoods is given as follows:

Definition 4.1: A universal neighborhood is the one that includes all partial, complete and global neighborhoods.

A universal neighborhood is depicted in Figure 4.7. The universal neighborhood itself may belong to a constellation of universes, and so on. These types of neighborhoods will aid

significantly in constructing structured Laplacian matrices that will be helpful in analyzing the behavior of interacting systems in the whole.

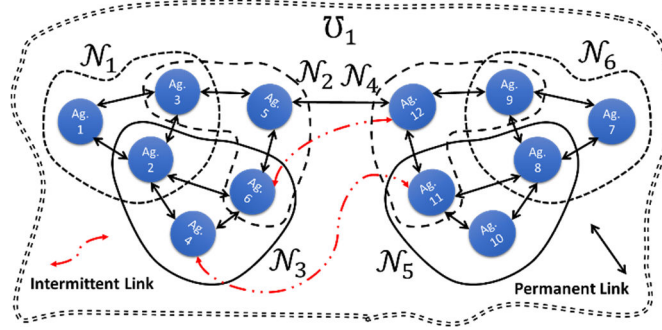


Figure 4.7. A universal neighborhood comprising multiple complete and partial neighborhoods among which permanent and persistent links may exist.

4.1.2: Formation in obstacle-free working environment

4.1.2.1: The first formation protocol

In various situations, a spatial MAS is expected to maintain specific arrangements or formations in 2-D or 3-D while interacting with the working environment. This behavior can be realized for example in the x-direction using the steepest decent law [131] given as follows:

$$u_i = \sum_{j \in \mathcal{N}_i} (d_{ij}^{*2} - d_{ij}^2) \{x_i - x_j\} \quad (4.16)$$

where: d_{ij}^{*2} is the desired Euclidean distance between the i^{th} and the j^{th} agents. Comparing (4.16) to (4.3), this consensus protocol is indeed semi-linear, moreover $g_{ij} = (d_{ij}^2 - d_{ij}^{*2})$ satisfies both (3.33.1) and (3.33.2) and therefore fits within the framework proposed in Chapter 3. However, note that g_{ij} used in (4.16) introduces a new equilibria set besides (3.31). Thus, several non-isolated equilibria may result under formation control, in general. Among these equilibria, the correct equilibrium is the one at which the specified distances are attained and is always attractive [131]. This fact will make the analysis of formation

control much involved when attractive incorrect equilibria do exist or under arbitrary formation switching; since no single globally attractive equilibria exists among the resulting switched systems, in general. Treatment of such systems can be found in [132] and [133], for examples. Using (4.16), the overall MAS dynamics in the x -direction can be given as follows:

$$\vec{\dot{x}}(t) = -\mathcal{L}(\vec{x}, \vec{y}, \vec{z}, \vec{s})\vec{x}(t) = \vec{f}^x(\vec{x}, \vec{y}, \vec{z}, \vec{s}) \quad (4.17)$$

where: \vec{s} is a vector whose elements are all possible d_{ij}^{*2} .

Similarly, the MAS dynamics in the y and z directions can be given as follows:

$$\begin{aligned} \vec{\dot{y}}(t) &= -\mathcal{L}(\vec{x}, \vec{y}, \vec{z}, \vec{s})\vec{y}(t) = \vec{f}^y(\vec{x}, \vec{y}, \vec{z}, \vec{s}) \\ \vec{\dot{z}}(t) &= -\mathcal{L}(\vec{x}, \vec{y}, \vec{z}, \vec{s})\vec{z}(t) = \vec{f}^z(\vec{x}, \vec{y}, \vec{z}, \vec{s}) \end{aligned} \quad (4.18)$$

Both (4.17) and (4.18) introduce couplings among agents when $d_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2$ is used; however, if $d_{ij}^2 = (x_i - x_j)^2$ in (4.16)- the same goes for d_{ij}^{*2} as well-, then there will be no coupling between (4.17) and (4.18). Either way, (4.8) follows. The difference between the two cases is the quality of the generated reference trajectory such that in the first case a smother trajectory is most likely to happen while in the second case a zig-zag like trajectory with sharp corners might happen which is not welcomed from the tracking agents point of view. This will be obvious when d_{ij} is used instead of d_{ij}^2 and therefore (4.16) becomes:

$$\begin{aligned} u_i &= \sum_{j \in \mathcal{N}_i} (d_{ij}^* - |x_i - x_j|)\{x_i - x_j\} \\ &= \sum_{j \in \mathcal{N}_i} \{d_{ij}^*(x_i - x_j) - \text{sign}(x_i - x_j)(x_i - x_j)^2\} \end{aligned} \quad (4.19)$$

where the *sign* function introduces a jump at its zero which may compromise the smoothness usually required in reference trajectory signal. So, introducing couplings in (4.8) is sometimes inevitable. Various inter and intra couplings in (4.8) result in interesting behaviors as will be introduced in the coming chapter.

To examine the effect of the internal coupling in a more depth, let $d_{ij}^{*2} = \{\Delta_{ij,x}^2, \Delta_{ij,y}^2, \Delta_{ij,z}^2\}$, then (4.17), for example, can be rewritten as follows:

$$\vec{x}(t) = [A_f^x - \mathcal{L}(\vec{x})]\vec{x}(t) + [A_f^y - \mathcal{L}(\vec{y})]\vec{y}(t) + [A_f^z - \mathcal{L}(\vec{z})]\vec{z}(t) \quad (4.20)$$

where: $A_f^{x,y,z} \in \mathbb{R}^{N \times N}$ denotes the fixed matrix comprising all $\Delta_{ij,x,y,z}^2$ elements in the three Cartesian coordinates x, y and z, respectively. We refer to A_f as the *shape matrix* associated with the *shape graph* [131]. Note that $A_f^{x,y,z}$ follows the structure of the Laplacian matrix \mathcal{L} associated with an undirected graph, generally, with different entries.

If $d_{ij}^{*2} = c_{ij}$ is provided instead, then (4.17) can be given as:

$$\vec{x}(t) = \{A_f - \mathcal{L}(\vec{x}, \vec{y}, \vec{z})\}\vec{x}(t) \quad (4.21)$$

which can be written as (4.17). Note that (4.21) reveals the link between consensus and formation control; in a way that (4.21) introduces new equilibria through A_f besides those introduced by $\mathcal{L}(\vec{x}, \vec{y}, \vec{z})$. The overall equilibria set is defined, for the x-direction, as follows:

$$\begin{aligned} \mathfrak{J}^x = & \{\vec{x} \in \mathbb{R}^N \mid \vec{x} = q\vec{1}\} \cup \{\vec{x} \in \mathbb{R}^N \mid \vec{v}^T \vec{f}(\vec{x}) = \vec{0}\} \\ & \cup \{\vec{x} \in \mathbb{R}^N \mid \vec{f}(\vec{x})\vec{e} = \vec{0}\} \cup \{\vec{x} \in \mathbb{R}^N \mid \vec{f}(\vec{x}) = \vec{0}\} \end{aligned} \quad (4.21.1)$$

where $q \geq 0 \in \mathbb{R}$, $\vec{v}^T(\vec{x})$ and $\vec{e}(\vec{x})$ are the nonlinear left and right eigenvectors, respectively. Then, the overall equilibria set is:

$$\mathfrak{J} = \mathfrak{J}^x \cup \mathfrak{J}^y \cup \mathfrak{J}^z \quad (4.21.2)$$

So, as stated in [131], several non-isolated equilibria may result under formation control (4.21) and the correct equilibrium is the one at which the specified distances are attained and is always attractive [131]. However, consider for example the case when all agents start exactly at the same initial conditions, i.e., $\vec{x} = c\vec{1}$, then they will stay there forever unless some sufficient perturbation will get them out this state.

Interestingly, a parameterized version of (4.21) can be used to control the behavior of the MAS such that it switches between consensus and formation. This observation reveals the relation between formation and consensus which can be thought of as the general case. Moreover, the formation is achieved about its moving coordinate system that is attached to the average value of agents' initial conditions. Therefore, (4.21) can be parameterized as follows:

$$\vec{x}(t) = \{c_f A_f - \mathcal{L}(\vec{x}, \vec{y}, \vec{z})\} \vec{x}(t) \quad (4.21.3)$$

where: $c_f \geq 0 \in \mathfrak{R}$ is the parameter used. Note that c_f and other existing parameters are usually controlled by a centralized, decentralized or distributed decision-making mechanism that governs the behavior of the MAS from a global perspective. Such parametrization enables the MAS to be semi or fully-autonomous while interacting with its surrounding. In general, analyzing the MAS behavior from global perspective allows grasping the big picture of interacting MAS systems, if they are many.

Clearly, c_f can be used to realize average consensus and formation among the connected agents; such that when $c_f > 0$ - usually 1-, then formation is achieved, and when it is zero, then the consensus is achieved. Shape uniform-scaling can also be controlled using c_f if it is other than zero or one. This procedure is equivalent to creation and destruction of the additional equilibria set introduced by the formation protocol.

Note 4.3: In (4.21.3), when switching occurs from formation to consensus, one might expect that it is impossible to switch back; because at consensus all states will be $\vec{x} = q\vec{1}$ and therefore (4.21.3) will stay in that equilibria. However, the time for consensus using (4.21.3) is too large and consequently switching back to formation might be possible unless consensus is already reached.

Enlightened by (4.21.3), another modified version of (4.21) can be developed in which switching between multiple formations is possible. It is given as follows:

$$\dot{\vec{x}}(t) = \{c_{f1}A_{f1} + \dots + c_{fk}A_{fk} - \mathcal{L}(\vec{x}, \vec{y}, \vec{z})\}\vec{x}(t) \quad (4.21.4)$$

where: $c_{fk} > 0$ is the parameter of the k^{th} formation.

Remark 4.3: The parameters in (4.21.4) can be used to assign a preferred formation to a specific situation or context occurring at the MAS or working environment levels, for example. To avoid stability or performance problems, only one c_{fk} parameter is nonzero at any given time instance so that there will be only one attractive equilibria at a given time.

This is related to the indication function

Recalling Chapter 3, if the formation is desired to have its coordinate system attached to a point other than that corresponding to the average value of its initial conditions, then (4.21.4) can be modified as follows:

$$\begin{aligned} \dot{\vec{x}}(t) &= G(\vec{x})\{c_{f1}A_{f1} + \dots + c_{fk}A_{fk} - \mathcal{L}(\vec{x}, \vec{y}, \vec{z})\}\vec{x}(t) \\ &= G(\vec{x})\vec{f}^x(\vec{x}, \vec{y}, \vec{z}, \vec{s}) \end{aligned} \quad (4.21.5)$$

where: $G(\vec{x})$ is like G_i given in (4.9).

Theorem 4.1: As $t \rightarrow \infty$, the leaderless parameterized MAS given in (4.21.5) will achieve the desired formation in the working space about the point μ_c corresponding to the desired mean type selected by $G(\vec{x})$ along with appropriate initial conditions if and only if

$\vec{f}^x(\vec{x}, \vec{y}, \vec{z}, \vec{s})$ has $\vec{v}^T = \vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0$, $\forall \vec{x}(t)$.

Proof: Since the shape matrices involved, i.e., A_{fi} , follows the same structure of \mathcal{L} , so they are symmetric and- along with their parameters- which are state-independent and generally constant, then the conditions (3.34) and (3.35) are still valid, and therefore the Jacobian of $\vec{f}(\vec{x})$ in (4.21.5) will have $\vec{1}$ as both nonlinear left and right eigenvectors associated with $\lambda_1(\vec{x}) = 0$, $\forall \vec{x}(t)$. So, (4.21.5) can achieve the desired formation about the desired consensus point in the working space. For more details, see (3.42).

Note that $|g_{ij}| = |d_{ij}^{*2} - d_{ij}^2| \rightarrow 0$ as $|x_i - x_j| \rightarrow 0$, therefore the control signal becomes smaller and smaller and convergence to right formation is expected to occur as $t \rightarrow \infty$.

■

Note 4.4: *Theorem 4.1* applies also to y and z directions.

Lemma 4.1: Let the conditions specified by *Theorem 4.1* be satisfied, then the parameters c_{f1} to c_{fk} can all be nonzero at the same time provided that $\sum_i c_{fi} = 1, i = 1, 2, \dots, k$. So, the resulting formation will be centered at the same point μ_c and contained within the convex hull of all formations.

Proof: Let A_f in (4.21) be given such that:

$$A_f = \sum_{i=1}^k c_{fi} A_{fk}$$

Then, the equilibrium points of (4.21) are in the convex hull of all A_{fk} .

■

Proposition 4.2: By slowly varying the parameters such that $\sum_i c_{fi} = 1, \forall t$, then a smooth morphing among formations can happen. This will suppress the abrupt changes in the

control signal which results in a smoother trajectory to be followed. In this case, the parameters will be sliding on a hyperplane in \mathfrak{R}^k where k is the number of formations. Other relations among the parameters are also possible.

□

Remark 4.4: By carefully examining the formation protocol given by (4.16), when $d_{ij}^{*2} - d_{ij}^2 \equiv d_{ij}^{*2} - (x_i - x_j)^2 - (y_i - y_j)^2 - (z_i - z_j)^2 = 0$ is used, then the equilibria set is basically the solution of all points residing on spheres surfaces centered at (x_j, y_j, z_j) that can be used to realize the desired formation. This is equivalent to the graphical procedure followed to draw polygons on a paper using compass. These points are determined based on agents' initial conditions and that is what really makes the formations realized using (4.16) geometrically flexible where reflection and rotation can occur. Therefore, more geometrical constraints may be required to have the desired formation.

4.1.2.2: The second formation protocol

Another formation control protocol that can be used is given as follows:

$$u_i = - \sum_{j \in \mathcal{N}_i} g_{ij} \{x_i - x_j - \Delta_{ij}^x\} \quad (4.22)$$

where: g_{ij} can be constant as in [9] or state-dependent as explained previously in Chapter 3, and Δ_{ij}^x is the fixed relative distance between the two agents- in the x-direction- according to the desired formation. Note that in (4.22), the elements in the equilibria set given in (3.31) are shifted by the corresponding Δ_{ij} specified and no additional equilibria are introduced. Thus, studying the formation convergence of (4.22) seems more appealing than (4.16). However, a comparison between the stated formation protocols will be provided in the coming subsection.

In fact, it is straight forward to examine the stability and convergence of the overall MAS under (4.22) following the same steps presented in Chapter 3.

The essence of using protocol (4.22) to achieve formation starts by adopting a moving or shifting reference frame, i.e., coordinate system, based on which the relative distances among agents comprising the desired formation are determined. This is illustrated in Figure 4.8.

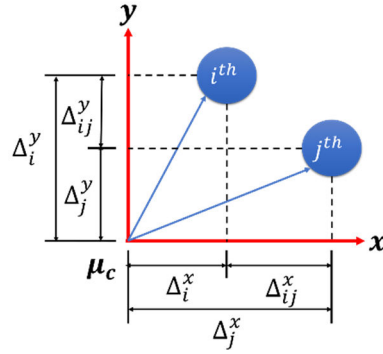


Figure 4.8. An illustration of the essence of formation protocol (4.22) showing two agents and the relative distances among them using a moving reference frame.

In Figure 4.8, the relative distance among both agents in the x -direction for example is $\Delta_{ji}^x = \Delta_j^x - \Delta_i^x$ which equals $-\Delta_{ij}^x$. Consequently, the overall MAS dynamics in the x -direction can be given as follows:

$$\vec{\dot{x}}(t) = -\mathcal{L}(\vec{x}, \vec{y}, \vec{z})\vec{x}(t) + B(\vec{x})\vec{1} \quad (4.23)$$

where: $B(\vec{x})$ is a skew-symmetric matrix with zero diagonal elements.

A modified version of (4.23) can be introduced such that the formation is realized around a point μ_c corresponding to the desired mean type. This can be given as follows for k formations:

$$\begin{aligned} \vec{\dot{x}}(t) &= G(\vec{x})\{-\mathcal{L}(\vec{x}, \vec{y}, \vec{z})\vec{x}(t) + c_{f1}B_{f1}(\vec{x}, \vec{y}, \vec{z})\vec{1} + \dots + c_{fk}B_{fk}(\vec{x}, \vec{y}, \vec{z})\vec{1}\} \\ &= G(\vec{x})\vec{f}^x(\vec{x}, \vec{y}, \vec{z}) \end{aligned} \quad (4.24)$$

Theorem 4.2: The leaderless parameterized MAS given in (4.24) will achieve the desired formation in the working space about the point μ_c corresponding to the desired mean type selected by $G(\vec{x})$ along with appropriate initial conditions if and only if $\vec{f}^x(\vec{x}, \vec{y}, \vec{z}, \vec{s})$ has $\vec{v}^T = \vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0, \forall \vec{x}(t)$. Moreover, the convergence to the desired formation will occur in a finite-time if the g_{ij} functions used C-S model with $0 < \beta \leq 1$.

Proof: Let $G(\vec{x}) = I$, i.e., the identity matrix of appropriate size, then for \vec{f}^x to have $\vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0, \forall \vec{x}(t)$, the g_{ij} functions in (4.22) must verify (3.34). Also, note that $\vec{1}^T B_{fi} \vec{1} \equiv 0, \forall i = 1, 2, \dots, k$; since B_{fi} is a skew-symmetric matrix. For other types of means, i.e., $G(\vec{x}) \neq I$, the proof follows as in (3.42). It was shown in (3.88) that a convergent g_{ij} utilizing the C-S model occurs when $0 < \beta \leq 1$ and therefore a finite-time convergence to the desired formation is expected to happen with $t < \infty$. ■

Note 4.5: *Theorem 4.2* applies also to y and z directions.

4.1.2.3: Comparison between the two formation protocols

The main differences among the two formation protocols can be summarized as follows:

- 1- The protocol given in (4.16) is geometrically more flexible than (4.22).
- 2- Switching from formation to consensus, agents controlled using (4.21.5) will not be able to switch back to formation if consensus is reached which is not the case with (4.24). Note that once consensus is approached under (4.21.5), the control signal will remain zero afterwards.

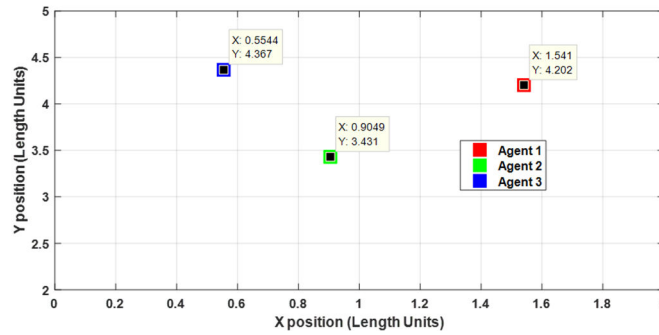
- 3- Because of the above, the switching- if possible- using (4.21.5) will occur about a slightly different point in the working space while this is not the case with (4.24).
- 4- Knowing that each behavior whether formation or consensus takes its initial conditions from the previous one, protocol (4.24) will fail to maintain the same reference point if behaviors are not fully achieved.
- 5- The control signal under protocol (4.22) is directly proportional to $(d_{ij}^{*2} - d_{ij}^2)$, while the control signal under (4.24) is inversely proportional to the relative distances between agents when a C-S model is used.

The results of formation control of three single-integrator agents to form an equilateral triangle based on (4.16) and (4.22) are shown in Figure 4.9. The triangle has a side length of 1 in length units. The C-S model was used to build the g_{ij} functions in (4.22). The undirected communication network shown in Figure 3.7 is used. Both the communication and formation graphs were identical. Notice that in Figure 4.9.a, the equilateral triangle is rotated about the average value of initial conditions, which were $\vec{x}(t_0) = [2, 3, -2]^T$ and $\vec{y}(t_0) = [3, 1, 8]^T$, while the equilateral triangle in Figure 4.9.b followed the desired orientation- set at the design stage of formation- about the average point of initial conditions. These results demonstrate the level of geometric flexibility of formations under the two protocols especially in 3-D where additional constraints are needed when using (4.16). In general, other behaviors might accompany the formation behavior like leader following, collision avoidance and preserving connectivity as found in [134]. However, the example provided herein showed the relation between the proposed framework and the formation problem. In a coming subsection, the relation of leader following while preserving the formation under the proposed framework is discussed in more detail.

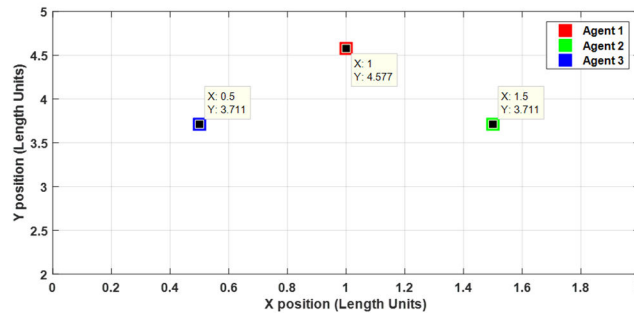
Note 4.6: Switching back and forth between consensus and formation can be found in nature where the petals of *Mimosa pudica* opens and closes based on the environmental conditions. In robotics engineering, such behavior might be useful in controlling the actuators responsible for the robot movement in the robotic jellyfish, for instance.

4.1.2.4: State-dependent parameters in formation protocols

Here, we extend the previous two formation protocols, namely: (4.21.5) and (4.24), to include state-dependent parameters that can be used to semi-autonomously or autonomously switch among formations- including complete consensus- based on a given context. This will be done using the proposed framework of Chapter 3.



(a)



(b)

Figure 4.9. Formation control of three single-integrator agents interacting over an undirected graph under: (a) protocol (4.16). (b) protocol (4.22).

A short version of (4.21.3) under state-dependent parameters will be used by which it is desired to investigate the behavior of MAS in a 3-D global neighborhood, i.e., in which all agents are interacting over a complete undirected graph in 3-D. This version is given as follows:

$$\begin{bmatrix} \vec{x}(t) \\ \vec{y}(t) \\ \vec{z}(t) \end{bmatrix} = \begin{bmatrix} Q & [0]_{N \times N} & [0]_{N \times N} \\ [0]_{N \times N} & Q & [0]_{N \times N} \\ [0]_{N \times N} & [0]_{N \times N} & Q \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{y}(t) \\ \vec{z}(t) \end{bmatrix} \quad (4.25)$$

$$Q = c_{f1}(\vec{x}, \vec{y}, \vec{z})A_{f1} + \dots + c_{fk}(\vec{x}, \vec{y}, \vec{z})A_{fk} - c(\vec{x}, \vec{y}, \vec{z})\mathcal{L}_f$$

Or as:

$$\begin{bmatrix} \vec{x}(t) \\ \vec{y}(t) \\ \vec{z}(t) \end{bmatrix} = \begin{bmatrix} Q_1 & [0]_{N \times N} & [0]_{N \times N} \\ [0]_{N \times N} & Q_2 & [0]_{N \times N} \\ [0]_{N \times N} & [0]_{N \times N} & Q_3 \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{y}(t) \\ \vec{z}(t) \end{bmatrix} \quad (4.25.1)$$

$$Q_i = c_{f1}(\vec{x}, \vec{y}, \vec{z})A_{f1} + \dots + c_{fk}(\vec{x}, \vec{y}, \vec{z})A_{fk} - c_i(\vec{a}_i)\mathcal{L}_f$$

where: $\vec{a}_i = \{\vec{x}, \vec{y}, \vec{z}\}$ and $[0]_{N \times N}$ is a block of $N \times N$ zeros with N as the number of agents in the global neighborhood.

The main difference between (4.25) and (4.25.1) is the nature of the weighting function used, i.e., c ; where in (4.25) it depends on the states in all directions of the 3-D space and in (4.25.1), it depends only on the states in one direction. As a result, the MAS moves in the working space in a way that suits all agents in the involved directions.

Theorem 4.3: The parameterized MAS given in (4.24) will achieve the desired formation in the working space about the point μ_c corresponding to the desired mean type selected by $G(\vec{x})$ for all state-dependent switching parameters $c_{fh}(\vec{x}, \vec{y}, \vec{z})$.

Proof: Working in the x-direction, let us analyze the formation protocol (4.24) to check the conditions needed for the state-dependent parameters to work under the proposed framework. The Jacobian of (4.24) is given as follows:

$$\frac{\partial \vec{f}^x(\vec{x}, \vec{y}, \vec{z})}{\partial \vec{x}} = \sum_{h=1}^k J_h + J_{\mathcal{L}} \quad (4.26)$$

where: $J_{\mathcal{L}}$ is the Jacobian associated with $-\mathcal{L}(\vec{x}, \vec{y}, \vec{z})\vec{x}(t)$ which has been dealt with extensively in Chapter 3 and therefore the needed conditions to force the $\vec{1}$ as a nonlinear left and right eigenvector associated with $\lambda_1(\vec{x}) = 0, \forall \vec{x}(t)$ are given in (3.34) and (3.35), respectively. The J_h is given as follows:

$$J_h = (B_{fh}(\vec{x}, \vec{y}, \vec{z})\vec{1})\nabla c_{fh} + c_{fh}\nabla(B_{fh}(\vec{x}, \vec{y}, \vec{z})\vec{1}) \quad (4.26.1)$$

Let $B_{fh}(\vec{x}, \vec{y}, \vec{z})\vec{1} = \vec{w}_h$ such that:

$$\begin{aligned} \vec{w}_h &= \left(\begin{bmatrix} 0 & b_{12} & b_{13} & \cdots & b_{1N} \\ -b_{12} & 0 & b_{23} & \cdots & b_{2N} \\ -b_{13} & -b_{23} & 0 & \ddots & b_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -b_{1N} & -b_{2N} & -b_{3N} & \cdots & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right)_h \\ &= [w_1 \quad w_2 \quad w_3 \quad \cdots \quad w_N]_h^T \end{aligned} \quad (4.26.2)$$

where: $b_{ij} = g_{ij}\Delta_{ij}^x$ with Δ_{ij}^x a real fixed scalar denoting the required relative distance in the formation among the i^{th} and j^{th} agents and $g_{ij} = H/[\gamma(x_j - x_i)^2 + \delta]^\beta$ is the C-S model with its parameters designed as was shown in (3.116); however, here they are all assumed to be 1.

So, we may write J_h as follows:

$$J_h = \begin{bmatrix} w_1 \frac{\partial c_f}{\partial x_1} + c_f \frac{\partial w_1}{\partial x_1} & w_1 \frac{\partial c_f}{\partial x_2} + c_f \frac{\partial w_1}{\partial x_2} & \cdots & w_1 \frac{\partial c_f}{\partial x_N} + c_f \frac{\partial w_1}{\partial x_N} \\ w_2 \frac{\partial c_f}{\partial x_1} + c_f \frac{\partial w_2}{\partial x_1} & w_2 \frac{\partial c_f}{\partial x_2} + c_f \frac{\partial w_2}{\partial x_2} & \cdots & w_2 \frac{\partial c_f}{\partial x_N} + c_f \frac{\partial w_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ w_N \frac{\partial c_f}{\partial x_1} + c_f \frac{\partial w_N}{\partial x_1} & w_N \frac{\partial c_f}{\partial x_2} + c_f \frac{\partial w_N}{\partial x_2} & \cdots & w_N \frac{\partial c_f}{\partial x_N} + c_f \frac{\partial w_N}{\partial x_N} \end{bmatrix}_h \quad (4.26.3)$$

Now, to force the $\vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0$, $\forall \vec{x}(t)$, we need to solve the following:

$$\vec{1}^T \frac{\partial f^x(\vec{x}, \vec{y}, \vec{z})}{\partial \vec{x}} = \sum_{h=1}^k \vec{1}^T J_h + \vec{1}^T J_L = \vec{0} \quad (4.26.4)$$

In Chapter 3, we have established the conditions to have $\vec{1}^T J_L = \vec{0}$ and here we turn our attention to solve the following equation:

$$\left(\sum_{h=1}^k \vec{1}^T J_h \right)_j = \sum_{h=1}^k \left(\frac{\partial c_f}{\partial x_j} \sum_{i=1}^N w_i + c_f \frac{\partial}{\partial x_j} \sum_{i=1}^N w_i \right)_h = 0 \quad (4.26.5)$$

where: $j = 1, 2, \dots, N$ denotes the j^{th} column in J_h . Knowing that \vec{w}_h is associated with a skew-symmetric matrix as given in (4.26.2), then the sum of its elements is zero and hence (4.26.5) is true. Therefore, $\vec{1}^T$ is naturally a left eigenvector of $\lambda_1(\vec{x}) = 0$ irrespective of the structure of the state-dependent switching parameters used. ■

Should $\vec{1}$ be a nonlinear right eigenvector associated with $\lambda_1(\vec{x}) = 0$, $\forall \vec{x}(t)$ as explained in Chapter 3, then certain conditions on the structure of the state-dependent switching parameters used in (4.24) must be imposed. This is detailed in the following lemma.

Lemma 4.2: Utilizing a C-S model to build the g_{ij} functions, a MAS whose agents are interacting over a global neighborhood under the formation control (4.24) can switch formation using state-dependent parameters while preserving the $\vec{1}$ as a nonlinear right eigenvector of $\lambda_1(\vec{x}) = 0, \forall \vec{x}(t)$ if

$$\nabla c_{fh} \cdot \vec{1} = 0, \forall h = 1, 2, \dots, k$$

Proof: Forcing the $\vec{1}$ as a nonlinear right eigenvector of $\lambda_1(\vec{x}) = 0$ is equivalent to solving the following equation:

$$\frac{\partial \vec{f}^x(\vec{x}, \vec{y}, \vec{z})}{\partial \vec{x}} \vec{1} = \sum_{h=1}^k J_h \vec{1} + J_{\mathcal{L}} \vec{1} = \vec{0} \quad (4.26.6)$$

In Chapter 3, we have established the conditions to have $J_{\mathcal{L}} \vec{1} = \vec{0}$ and here we turn our attention to solve the following equation:

$$\left(\sum_{h=1}^k J_h \vec{1} \right)_i = \sum_{h=1}^k \left(w_i \sum_{j=1}^N \frac{\partial c_f}{\partial x_j} + c_f \sum_{j=1}^N \frac{\partial w_i}{\partial x_j} \right)_h = 0 \quad (4.26.7)$$

where: $i = 1, 2, \dots, N$ denotes the i^{th} row in J_h . Knowing that the elements of \vec{w}_h are utilizing the C-S model which verifies the conditions given in (3.35), then it is straightforward to verify that:

$$\left(\sum_{j=1}^N \frac{\partial w_i}{\partial x_j} \right)_h = 0, \quad \forall i = 1, 2, \dots, N \quad (4.26.8)$$

Therefore, (4.26.7) becomes as follows:

$$\left(\sum_{h=1}^k J_h \vec{1} \right)_i = \sum_{h=1}^k \left(w_i \sum_{j=1}^N \frac{\partial c_f}{\partial x_j} \right)_h = \sum_{h=1}^k (w_i \nabla c_f \cdot \vec{1})_h = 0 \quad (4.26.9)$$

One possible solution is given as follows:

$$\nabla c_{fh} \cdot \vec{1} = 0, \forall h = 1, 2, \dots, k \quad (4.26.10)$$

■

Having only one formation active at a given context is very crucial; so that the MAS can respond- as planned- to the variation in the working conditions. This requires imposing activation rules on the state-dependent switching parameters. This is explained as follows:

Proposition 4.3: The parameterized MAS given in (4.24) will have only one formation active at any given instant if and only if the state-dependent switching parameters satisfy the following generalized composition condition:

$$(c_{fi} \circ c_{fj})(\vec{x}, \vec{y}, \vec{z}) = 0, \quad \forall i, j \neq i$$

Indicating that both parameters cannot be active at the same instant.

□

The simplest realization of **Proposition 4.3** is the Heaviside unit-step function such that:

$$\begin{aligned} c_{fi} &\stackrel{\text{def}}{=} u_i(g(\vec{x}, \vec{y}, \vec{z}) - a) = \begin{cases} 0, & g(\vec{x}, \vec{y}, \vec{z}) < a \\ 1, & g(\vec{x}, \vec{y}, \vec{z}) \geq a \end{cases} \\ c_{fj} &\stackrel{\text{def}}{=} u_j(a - g(\vec{x}, \vec{y}, \vec{z})) = \begin{cases} 1, & g(\vec{x}, \vec{y}, \vec{z}) < a \\ 0, & g(\vec{x}, \vec{y}, \vec{z}) \geq a \end{cases} \end{aligned} \quad (4.27)$$

where: $g(\vec{x}, \vec{y}, \vec{z})$ is a general function of states that might model a certain variable in the working environment and a is a fixed constant. Relation (4.27) can be extended to any number of state-dependent switching parameters. A future work will be directed towards investigating the use of mainly orthogonal polynomials to realize the state-dependent parameters. Another possibility is to have $(c_{fi} \oplus c_{fj})(\vec{x}, \vec{y}, \vec{z}) = 1, \forall i, j \neq i$ which can be realized by having $c_{fi}(1 - c_{fj}) + (1 - c_{fi})c_{fj} = 1$.

Note 4.7: In (4.27), finite state-machines can be used instead of the Heaviside unit-step functions as will be demonstrated in the following subsection.

4.1.2.5: Communication-aware Formation control

In this subsection, mainly the utilization of the formation protocol (4.24) with $G(\vec{x}) = I$ is demonstrated by autonomously switching the formation based on the communication signal strength in the working space using the state-dependent parameters. Finite state-machine is used in conjunction with the state-dependent parameters to decide about the suitable formation used in any region in the working space. The communication signal strength was modeled based on the 2-D simulator available in [85]. We assume a stationary router- residing at the origin of the x - y plane- through which agents are communicating.

The communication signal strength ($SS_i(\vec{x}, \vec{y})$) model sensed by the i^{th} agent includes both the path loss and shadowing effects. Because agents are interacting over a global neighborhood, the decision variable SS_T of the total SS sensed by the MAS is used to autonomously decide between complete consensus and a circle formation. Figure 4.10 shows the communication SS in the working space with 30 dBm shift along the figure z -axis.

The decision variable $SS_T \in [0,1]$ is given mathematically as follows:

$$SS_T = \left(\sum_{i=1}^N \tanh(SS_i - TH) + 8 \right) / 16 \quad (4.28)$$

where: $TH = -30 \text{ dBm}$ is the minimum SS_i threshold and $N = 8$ indicates the number of single-integrator agents involved. Figure 4.11 shows the finite state-machine with continuous states- forming what is called a hybrid automaton- used in decision making based on which the $c_{f1} = \{0,1\}$ is set, while the evolution of SS_T over time is shown in Figure 4.12.

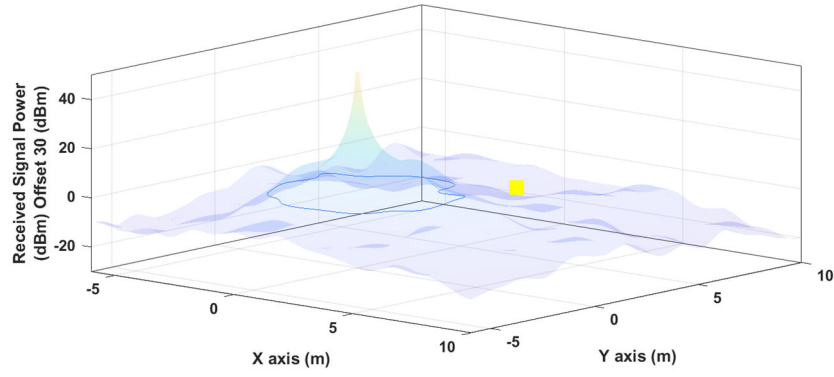


Figure 4.10. Communication signal power in the working space.

Figures 4.11-4.13 are related to the case where the agents initial conditions are $IC_x = [-6, 14, -2, 6, -3, 4, 6, 5]^T$ and $IC_y = [3, 14, 8, 3, 8, 1, 2, -5]^T$. Note that the average value of these initial conditions, i.e., (3, 4.25), is outside the prescribed region of desired SS_i and therefore consensus outside this region is achieved.

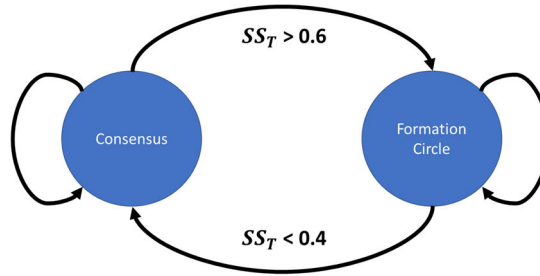


Figure 4.11. A hybrid automaton used in autonomous decision making.

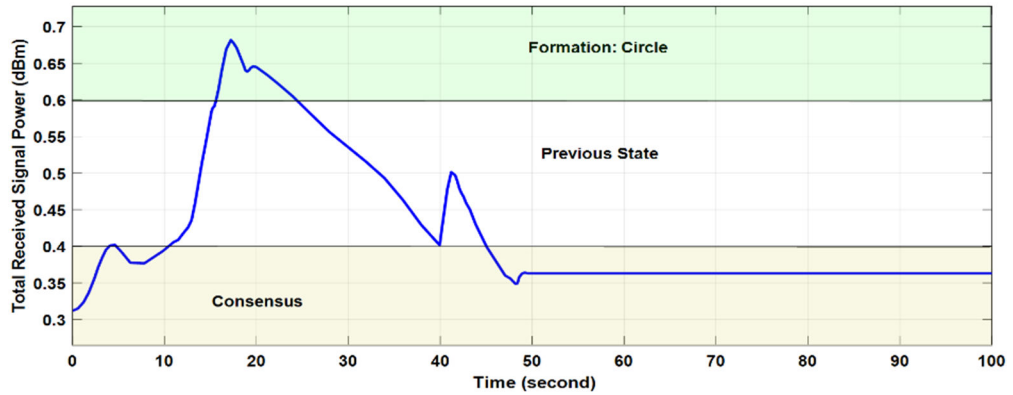


Figure 4.12. Evolution of SS_T over time using the finite state-machine shown in Figure 4.11.

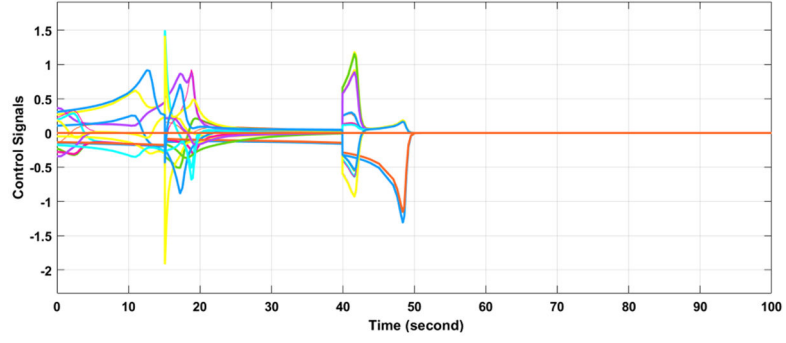


Figure 4.13. Control signal of the overall MAS during the whole switching.

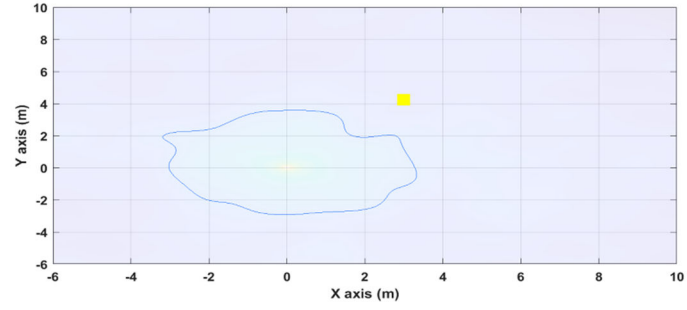


Figure 4.14. Final behavior of the MAS is consensus; since the average value of agents initial condition was outside the desired region of $SS_i(\vec{x}, \vec{y})$.

The result of another case where the previous initial conditions are scaled down by 0.2 is shown in Figure 4.15.

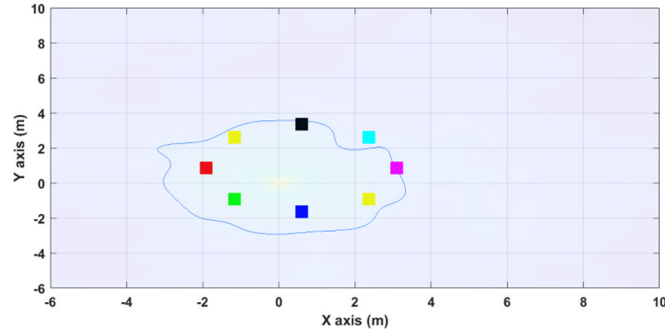


Figure 4.15. Final behavior of the MAS is formation; since the average value of agents initial condition was inside the desired region of $SS_i(\vec{x}, \vec{y})$.

So, using the finite state-machine shown in Figure 4.11, agents are unable to tell how formation could be possible inside the prescribed region of $SS_i(\vec{x}, \vec{y})$ and that is left for the initial conditions to decide, which is unacceptable, in general.

To solve this problem, we introduce another simple behavior that drives agents towards a rendezvous point, i.e., the origin of the plane, while monitoring the SS_i values $\forall i$ before switching to the formation behavior. The related finite state-machine is shown in Figure 4.16.

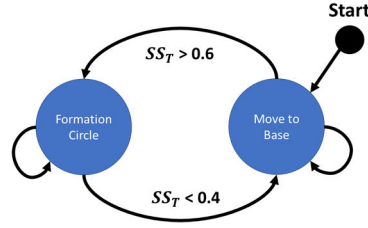


Figure 4.16. Finite state-machine used to drive agents towards desired region of $SS_i(\vec{x}, \vec{y})$ where formation is required.

The *Move to Base* behavior could be realized by any suitable linear Hurwitz matrix or a stable nonlinear one involving damping. Figure 4.17 shows the time evolution of SS_T using the finite state-machine shown in Figure 4.16. Note that the initial conditions were $IC_x = [-6, 14, -2, 6, -3, 4, 6, 5]^T$ and $IC_y = [3, 14, 8, 3, 8, 1, 2, -5]^T$. The MAS- under the finite state-machine shown in Figure 4.16- achieved formation in the desired region of $SS_i(\vec{x}, \vec{y})$ about a point other than the average value of the initial conditions; because of the *Move to Base* behavior used.

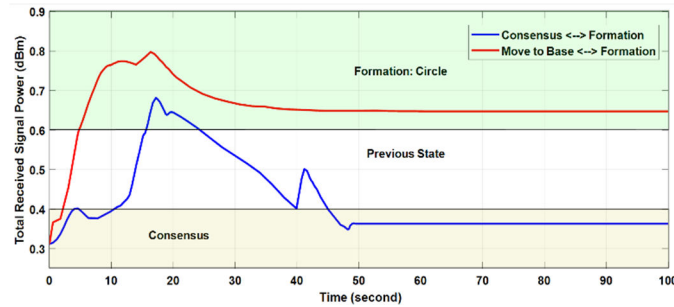


Figure 4.17. Evolution of SS_T over time using the finite state-machine shown in Figure 4.16.

More information about the leader following effects on formation control is detailed in the following subsection.

Note 4.8: At the formation design stage, i.e., when assigning the values for $\Delta_{ij}^{x,y}$, of the previous examples, it was known in a-priori that the formation will fit within the desired region of $SS_i(\vec{x}, \vec{y})$. This knowledge may not be available all the time, so there should be a way to enable agents to decide on the fly about the- most- suitable formation among possible formations that can fit within a prescribed region, if they are many, or to scale down the dimensions of the only existing formation.

4.1.2.6: Formation control while following a leader

In real robotics applications, at least one leader- whether a real or a virtual one- is introduced to the MAS and it is required to follow that leader while maintaining the designed formation. These competing requirements may not be always satisfied along the course of the planned mission. Therefore, in this section, it is desired to investigate the effect of introducing a, generally state-dependent, pinning gain on both the formation quality and the equilibria set when the MAS is interacting over a global interaction neighborhood.

Let us start by first examining the formation protocol (4.24) under the influence of a leader, state-dependent formation switching parameter and one possible formation. Mathematically, this is given as follows- in the x-direction:

$$\dot{x}_i = u_i^x = \gamma_{i,f} u_{i,f}^x + \gamma_{i,l} u_{i,l}^x \quad (4.29)$$

where: $u_{i,f}^x$ and $u_{i,l}^x$ are the i^{th} agent formation and following control signals, respectively and $\gamma_{i,f}, \gamma_{i,l} > 0$ are tuning parameters by which the preference of the i^{th} agent towards formation or following the leader is displayed. The control signals in both y and z directions follow the same approach. The control signals in (4.29) are given as follows:

$$u_{i,f}^x = - \sum_{j \in \mathcal{N}_i} g_{ij} \{x_i - x_j - c_{f1}(\vec{x}, \vec{y}, \vec{z}) \Delta_{ij}^x\} \quad (4.29.1)$$

$$u_{i,l}^x = -p_i(x_i - x_l)$$

where: p_i is the pinning gain assumed to be state-dependent. The overall dynamics of the MAS under (4.29) is given as follows- using short-hand notations about state-dependence:

$$\vec{\dot{x}}(t) = -\{\Gamma_p + \Gamma_f \mathcal{L}_s\} \vec{x}(t) + c_{f1} \Gamma_f B_{f1} \vec{1} + \Gamma_p \vec{1} x_l(t) \quad (4.29.2)$$

where: \mathcal{L}_s is the state-dependent Laplacian matrix, $\Gamma_p = \text{diag}([\gamma_{1,l} p_1, \dots, \gamma_{N,l} p_N])$ and $\Gamma_f = \text{diag}([\gamma_{1,f}, \dots, \gamma_{N,f}])$.

Mathematically, when $\text{trace}(\Gamma_p) > 0$, then $\Gamma_p + \Gamma_f \mathcal{L}_s > 0$, $\forall \vec{x}$ is a non-singular *M-matrix* and therefore the notion of null-space does not exist anymore and $\lambda_1(\vec{x}) > 0$, $\forall \vec{x}(t)$.

Therefore, $\vec{1}$ does not belong to the equilibria set and the formation is achieved about the leader position rather than a mean of the agents initial conditions. This also applies to the first formation protocol which can be given under a leader action as follows:

$$\vec{\dot{x}}(t) = \left\{ \Gamma_f \left(c_{f1} A_{f1} - \mathcal{L}(\vec{x}, \vec{y}, \vec{z}) \right) - \Gamma_p \right\} \vec{x}(t) + \Gamma_p \vec{1} x_l(t) \quad (4.29.3)$$

Let the pinning gain in 3-D be given as follows:

$$p_i = \frac{p_{i,s}}{(x_l - x_i)^2 + (y_l - y_i)^2 + (z_l - z_i)^2 + 1} \quad (4.29.4)$$

or:

$$p_i = p_{i,s} [(x_l - x_i)^2 + (y_l - y_i)^2 + (z_l - z_i)^2 + 1] \quad (4.29.5)$$

where: $p_{i,s} > 0 \in \mathfrak{R}$ is a fixed constant, set by the designer, which determines the value of the pinning gain when tracking error is zero. Note that this pinning gain approach is like the proportional gain found in a regular proportional-derivative-integral (PID) controller. So, for any pinning gain, i.e., $p_i = p_{i,s}$, and slowly varying leader, then the forced equilibria

can be used to approximate the deformation in the formation due to the leader pinning gain.

This is stated in the following theorem.

Theorem 4.4: For any given positive pinning gain matrix Γ_p , uniform Γ_f and a slowly varying leader, the deformation in the formation ΔS under (4.29.2) and (4.29.3) can be predicted, to a certain accuracy, at any position of the leader in the working space by solving the following problem:

$$\begin{aligned}\vec{1}^T \Gamma_p (\vec{x}_{eq} - \vec{1}x_l) &= 0 \\ \vec{1}^T \Gamma_p (\vec{y}_{eq} - \vec{1}y_l) &= 0 \\ \vec{1}^T \Gamma_p (\vec{z}_{eq} - \vec{1}z_l) &= 0\end{aligned}\tag{4.30}$$

Then, $\Delta S = -\mathcal{L}(\vec{x}_{eq}, \vec{y}_{eq}, \vec{z}_{eq})\vec{x}_{eq} + B_{f1}(\vec{x}_{eq}, \vec{y}_{eq}, \vec{z}_{eq})\vec{1}$ under (4.29.2) and $\Delta S = A_{f1} - \mathcal{L}(\vec{x}_{eq}, \vec{y}_{eq}, \vec{z}_{eq})$ under (4.29.3).

Proof: By evaluating the forced equilibria of (4.29.2), we end up having the following- in the x-direction:

$$\begin{aligned}\vec{1}^T \vec{0} &= -\vec{1}^T \{\Gamma_p + \gamma_{1,f} \mathcal{L}_s\} \vec{x}_{eq} + c_{f1} \gamma_{1,f} \vec{1}^T B_{f1} \vec{1} + \vec{1}^T \Gamma_p \vec{1}x_l(t_{eq}) \\ &= \vec{1}^T \Gamma_p (\vec{x}_{eq} - \vec{1}x_l) = 0\end{aligned}\tag{4.30.1}$$

where: B_{f1} is a skew-symmetric matrix. Doing the same with (4.29.3), yields:

$$\begin{aligned}\vec{1}^T \vec{0} &= \vec{1}^T \{\gamma_{1,f} (c_{f1} A_{f1} - \mathcal{L}_s) - \Gamma_p\} \vec{x}_{eq} + \vec{1}^T \Gamma_p \vec{1}x_l(t_{eq}) \\ &= \vec{1}^T \Gamma_p (\vec{x}_{eq} - \vec{1}x_l) = 0\end{aligned}\tag{4.30.2}$$

Note that $\vec{x}_{eq} - \vec{1}x_l = \vec{0}$ is not a solution; because under a leader pinning gain the $\vec{1}$ is no more in the equilibria set, i.e., $\lambda_1(\vec{x}) > 0$, $\forall \vec{x}(t)$. The evaluation of ΔS is straightforward.

■

Under a leaderless situation, agents still have the possibility to issue their own pinning gain; however, in such a situation it is called a self-loop by which an agent tends to the origin about which the formation is achieved in a similar manner under a leader. In other words, a self-loop is equivalent to a virtual leader at the origin.

4.1.2.7: Should I follow or abandon the leader?

In the previous subsection, the pinning gain Γ_p is distance-dependent with a static gain, i.e., $p_{i,s}$, set by the designer. Under strict conditions, it might be desirable to empower agents with the choice to follow or abandon the leader autonomously. Consider for instance the case where the leader is moving towards a communication denied region, then agents should not indulge themselves in that region simply to avoid getting lost. This is most clear when the leader is an actual pursuit or unfriendly agent. Nevertheless, agents may still be able to monitor the leader even without taking actions, i.e., separating the awareness from action.

Note that in (4.29.2) and (4.29.3) the value of c_{f1} is decided for at the MAS level as explained before; however, for Γ_p it is mainly determined at the agent level. Let the leader motion be independent from the state of all agents, so the conditions needed for $\Gamma_p \vec{1}x_l(t)$ to have $\vec{1}^T$ as a left eigenvector associated with $\lambda_1(\vec{x}) = 0$, $\forall \vec{x}(t)$ are given as follows- see (4.26) where the superposition principle is used and recalled here:

$$J_{\Gamma_p} = \frac{\partial}{\partial \mathbf{x}} (\Gamma_p \vec{1}x_l) = \vec{1}x_l \nabla \Gamma_p \quad (4.31)$$

$$= \begin{bmatrix} \gamma_{1,l} \frac{\partial p_1}{\partial x_1} & \gamma_{1,l} \frac{\partial p_1}{\partial x_2} & \cdots & \gamma_{1,l} \frac{\partial p_1}{\partial x_N} \\ \gamma_{2,l} \frac{\partial p_2}{\partial x_1} & \gamma_{2,l} \frac{\partial p_2}{\partial x_2} & & \gamma_{2,l} \frac{\partial p_2}{\partial x_N} \\ \vdots & & \ddots & \vdots \\ \gamma_{N,l} \frac{\partial p_N}{\partial x_1} & \gamma_{N,l} \frac{\partial p_N}{\partial x_2} & \cdots & \gamma_{N,l} \frac{\partial p_N}{\partial x_N} \end{bmatrix}$$

Now, to force the $\vec{1}^T$ as left eigenvector associated with $\lambda_1(\vec{x}) = 0$, $\forall \vec{x}(t)$ when the whole MAS is not pinned to any leader, we need to make sure that the following is true for each j^{th} column $\forall \vec{x}(t)$:

$$\left(\vec{1}^T J_{\Gamma_p} \right)_j = \frac{\partial}{\partial x_j} \sum_{i=1}^N \gamma_{i,l} p_i = 0 \quad (4.31.1)$$

Which can be controlled by setting all γ_i to zero. Similarly, to force $\vec{1}$ as a right eigenvector of $\lambda_1(\vec{x}) = 0$, $\forall \vec{x}(t)$ the following should always be true for each i^{th} row $\forall \vec{x}(t)$:

$$\left(J_{\Gamma_p} \vec{1} \right)_i = \gamma_i \nabla p_i \vec{1} = 0 \quad (4.31.2)$$

Now, to appreciate the usefulness of having state-dependent pinning gains parameters, let us have the following mission where agents are required to follow the leader, stay connected with the base station, i.e., the wireless router, and to change formation based on the signal strength in the region they are navigating. Such competing requirements can be tackled in a systematic way under the proposed framework.

This mission is demonstrated through two examples. In *Example 4.1*, the mission is simplified such that the leader will stay nearby the base station and therefore the situation whether agents should decide to follow the leader or not is never encountered. On the opposite, *Example 4.2* will emphasize this situation and will explain how such contradicting requirements can be addressed simultaneously. In both examples, the

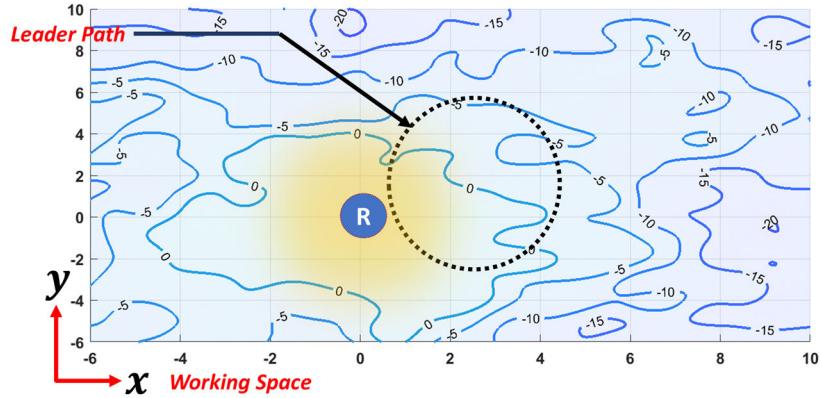
following modified version of (4.16) is used due to its associated geometrical flexibility and it is given as follows:

$$u_i = \sum_{j \in \mathcal{N}_i} \frac{(d_{ij}^{*2} - d_{ij}^2)}{(d_{ij}^{*2} - d_{ij}^2)^2 + 1} \{x_i - x_j\} \quad (4.32)$$

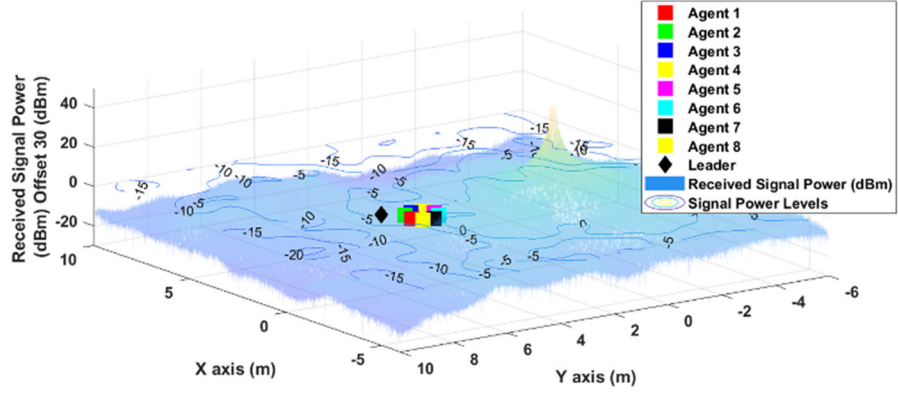
The main advantage of this version is the reduction of the control signal when $d_{ij}^{*2} - d_{ij}^2$ is too large. It can be shown that (4.36) satisfies both (3.34) and (3.35).

Example 4.1: Autonomous leader-follower and adaptive formation scaling based on communication signal strength

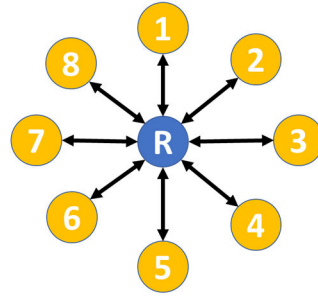
In Figure 4.18, the working space and the leader path along with the available signal strengths levels are shown. Agents should decide autonomously about the best formation scale to have in any region. This is a realization of **Proposition 4.2**. Agent 2 is pinned to the leader while others maintain the formation. It is assumed that agents are interacting over a complete interaction neighborhood.



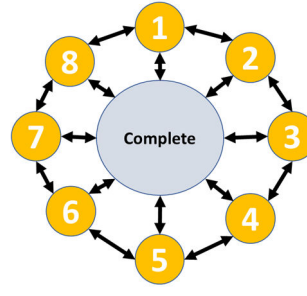
(a)



(b)



(c)



(d)

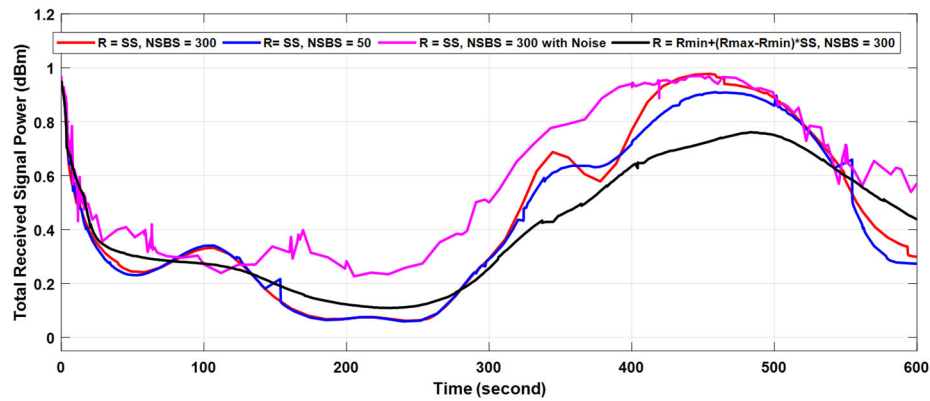
Figure 4.18. Example 4.1: (a) the 2-D working space and the leader path along with the available signal strengths contours with zero level at -30 dBm. (b): 3D view of the working space overlapped with the received signal power including path loss, shadowing and multipath fading effects. (c): communication network topology connecting agents through the router. (d): the global interaction neighborhood, i.e., shape, graph used.

Unlike (4.28), using the wireless channel simulator [135] with the multipath fading effect involved results in a noisy received signal strength, therefore $SS_T \in [0,1]$ will inherit this noise. Consequently, depending on the noisy SS_T to determine the scale of the formation for instance is not recommended; since changes will be demanded almost at each time instant and the MAS will act hesitant about the radius to achieve.

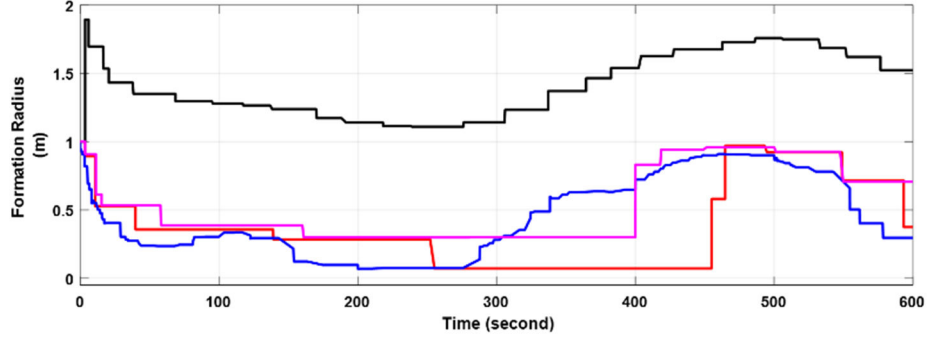
To gain some level of confidence about the decision to make, i.e., to have a robust behavior within sufficient margin, the SS_T calculated using (4.28) is filtered over some period to make sure that the decision taken is convenient. This resulted in $\overline{SS_T}$.

Remark 4.5: Besides the robustness in decision making, other features are usually required in any intelligent controller such as completeness, i.e., to know in advance that the behavior taken is sufficient for the current context and therefore can be fully conducted, and timeliness of the decision such that latency between decision making and action taken is minimum. Unfortunately, these requirements are not addressed in this thesis; since they require sufficient level of perception which is beyond the scope of the current work.

To filter this noisy signal, we have used the average mean of a collected number of samples over a period. Using other means such as the median is also possible. Figure 4.19.a shows the results obtained when averaging different number- of-samples-before-scaling-or-switching ($NSBS$) of SS_T with/out the multipath effects (noise) being active. The formation radius (R) was taken either as $R = \overline{SS_T}$ or $R = R_{min} + (R_{max} - R_{min}) \overline{SS_T}$ as shown in Figure 4.19.b. The pinning gain given in (4.29.5) was used with $p_{2,s} = 0.5$.



(a)



(b)

Figure 4.19. Example 4.1: (a) the value of $\overline{SS_T}$ obtained for different NSBS with/out incorporating the multipath fading effect as a noise source. (b) the corresponding formation radii.

The effect of introducing $\overline{SS_T}$ in R gives the advantage of using onboard proximity sensors to connect agents together in case the wireless link with the base station is cut.

□

Remark 4.6: To ensure the connectivity of the MAS connected via a wireless router as used in *Example 4.1*, then the magnitude of the modified Fiedler eigenvalue, i.e., $\lambda_2(\vec{x})$, associated with must be greater than zero. More specifically, one must examine $\lambda_2(\vec{x})$ associated with the graph given in Figure 4.18.c, i.e., the communication graph, and not that given in Figure 4.18.d, i.e., the shape graph. This is generally applicable for all cases where different behaviors are possible over a common communication network.

Example 4.2: Autonomous leader-follower and adaptive formation scaling based on communication signal strength with the ability to follow or abandon the leader

In this example, the leader is designed to move along a trajectory where $\overline{SS_T}$ falls down a predetermined value set by the designer. This value acts as a constraint on the connectivity of the whole MAS. Here, agent 2 is pinned to both the leader and the base station and therefore has two contradicting goals to achieve, namely: follow the leader and stay

connected to the base station. Furthermore, we will challenge the MAS to decide for the formation radius based on a noisy $\overline{SS_T}$ resulting from incorporating the multipath fading in the simulated wireless channel. The pinning gains connecting agent 2 to the base station and the leader are given as follows:

$$\begin{aligned}
p_{2-base,x} &= \frac{(\tanh(k(\overline{SS_T} - TH)) - 1)(x_2 - x_{base})}{\sqrt{(x_2 - x_{base})^2 + 1}} \\
p_{2-base,y} &= \frac{(\tanh(k(\overline{SS_T} - TH)) - 1)(y_2 - y_{base})}{\sqrt{(y_2 - y_{base})^2 + 1}} \\
p_{2-leader,x} &= \frac{(x_{leader} - x_2)}{\sqrt{(x_{leader} - x_2)^2 + 1}} \\
p_{2-leader,y} &= \frac{(y_{leader} - y_2)}{\sqrt{(y_{leader} - y_2)^2 + 1}}
\end{aligned} \tag{4.33}$$

Where we have used the results obtained in (3.116) about the C-S model [64]. $TH \geq 0$ denotes the required signal strength threshold. So, agent 2 is responsible for solving the contradicting requirements while other agents are focused on maintain the formation according to the radius determined by the whole MAS. It is known that the $\tanh(\cdot) \in [-1,1]$, therefore by selecting $k > 0$ large enough, then the pinning to the base station is active only when $\overline{SS_T}$ is below TH . On the contrary, when k is selected small enough, then the pinning to the base station is always active. These two cases result in an interesting behavior of the MAS where in the first case the MAS will be acting as being reactive while in the second case it will act as being proactive. The two cases are explained shortly. Note that the value of k is specified according to the application.

Case I: The MAS being reactive

Here, it is assumed that the states used in decision making are noise-free, i.e., deterministic, and thus can be used directly without extra precautions. So, in (4.33), $k = 50$ was used.

Case 2: The MAS being proactive

Here, it is assumed that the available information, i.e., states, are uncertain, noisy or the like. So, the MAS must cope with this situation assuming worst case scenario, i.e., be robust. Here, the noisy \overline{SS}_T is our decision variable and therefore we have chosen $k = 0.5$ to be used.

In both cases, we have $TH = 0.4$, $NSBS = 300$ and $R = R_{min} + (R_{max} - R_{min}) \overline{SS}_T$ with $R_{min} = 1 \text{ meter}$ and $R_{max} = 2 \text{ meters}$. The working space and the leader path are shown in Figure 4.20.

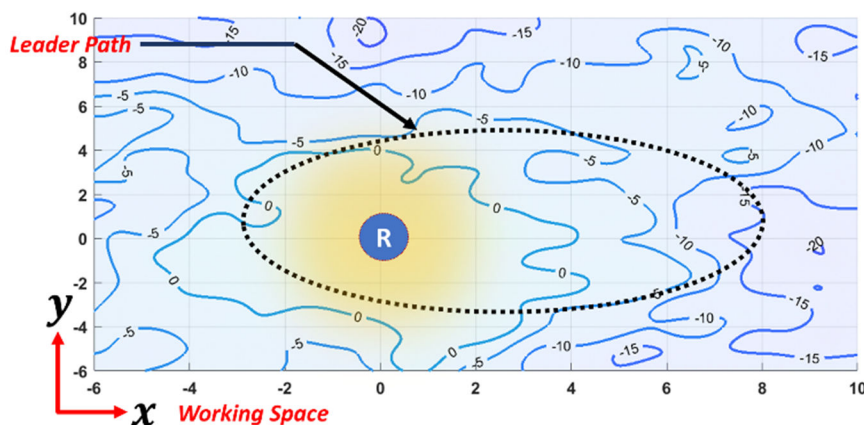
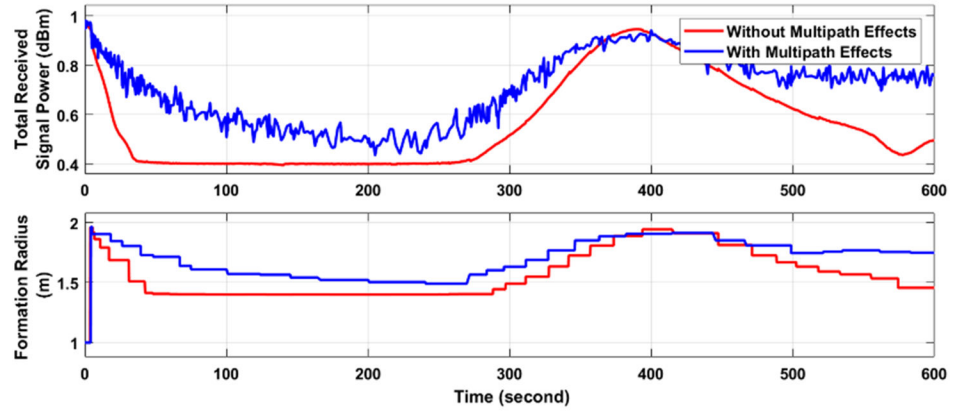
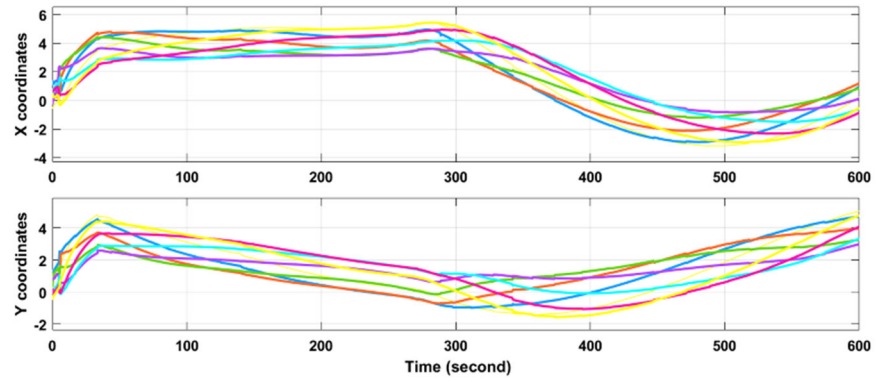


Figure 4.20. Example 4.2: the 2-D working space and the leader path along with the available signal strengths contours with zero level at -30 dBm.

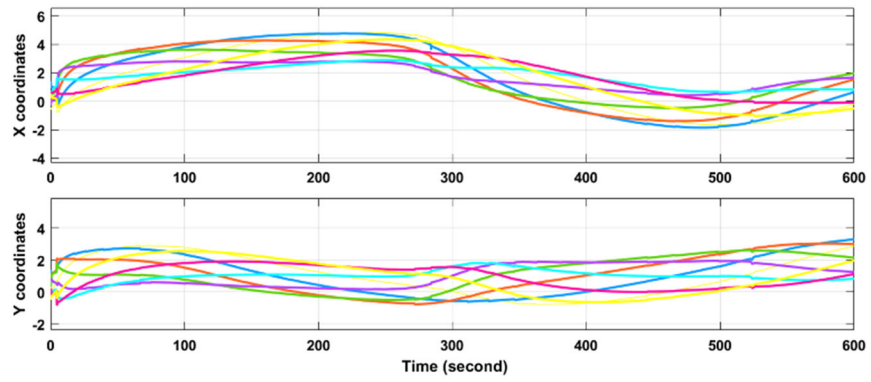
Figure 4.21 shows the results obtained when simulating the previous two cases from which one may conclude that having a proactive MAS will result in better dynamical-friendly trajectories and therefore a reduced control signal chattering.



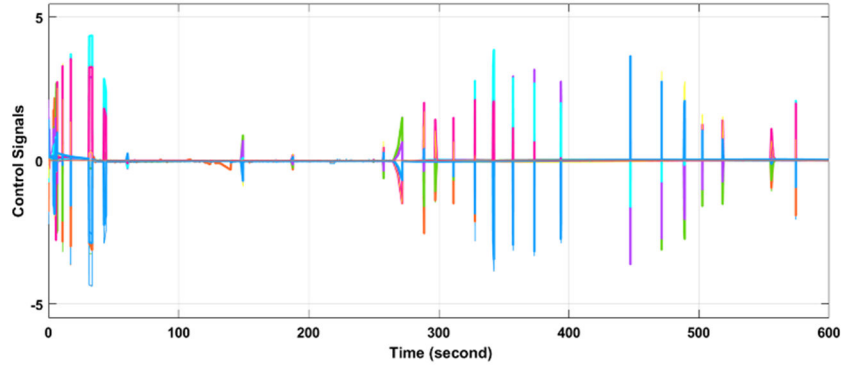
(a)



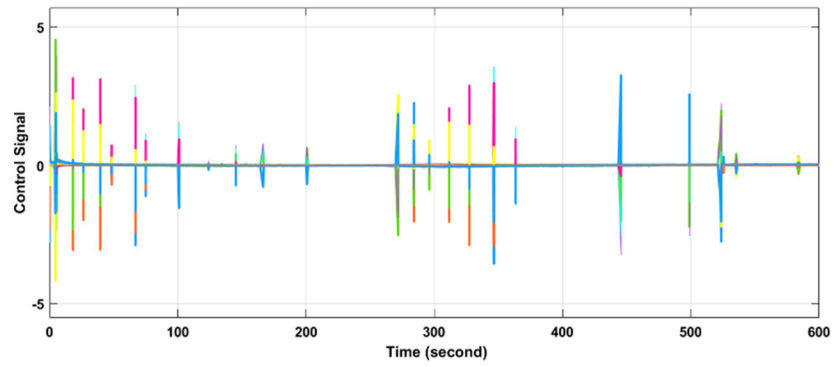
(b)



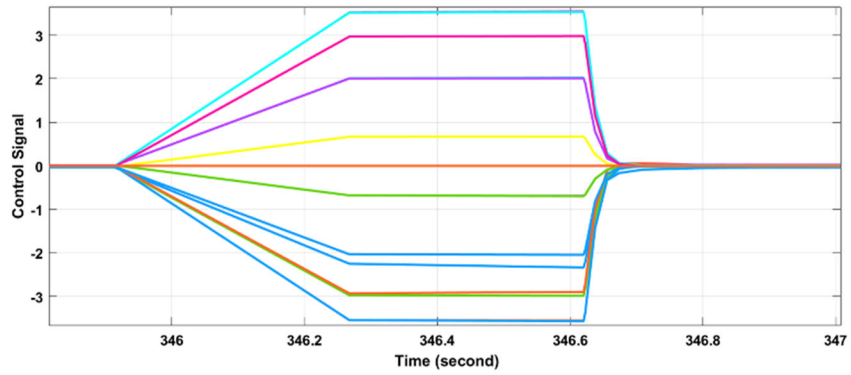
(c)



(d)



(e)



(f)

Figure 4.21. Example 4.2: (a) the total received signal power and the resulting formation radius. (b)-(e) agents positions and corresponding control signals without and with incorporating the multipath fading effects as noise. (f) a close view on (e) at 346 seconds which looks like the control signal in (d).

Note 4.9: When online measurements are used in the state-dependent parameters, the trace of their variance could be used to reflect upon the value of k in case agents have to decide its value autonomously. General filters can also be used as found in [136].

□

4.1.3: Deployment in obstacle-free 3-D space

In this section, the deployment behavior of the MAS in an obstacle-free 3-D space is introduced. Such a behavior is crucial in many applications where agent may represent communication routers, repeaters or mobile sensing nodes that can be used in the working environment. During deployment, each agent should maximize the Euclidian distance between itself and its neighbors while preserving the connectivity that can be modelled as a function of the signal strength among agents or as a distance threshold that should not be exceeded. Unlike the previous subsections, in this section, the wireless channel simulator presented in [135] cannot be used; because it is only applicable to 2-D. Therefore, the specified threshold will be a function of the maximum agents' separation allowed.

Assuming a fully-connected undirected graph, the state-dependent network graph responsible for achieving the deployment in the x-direction for example can be given as follows- utilizing the concept of *hysteresis switching* [137]:

$$\begin{aligned}\dot{x}_i &= -\alpha \sum_{j \in \mathcal{N}_i} g_{ij}(x_j, x_i) \{x_i - x_j\} \\ g_{ij}(x_j, x_i) &= \tanh \left(k(TH - \hat{g}_{ij}) \right) \hat{g}_{ij} \\ \hat{g}_{ij} &= \frac{H}{\left[\gamma(x_j - x_i)^2 + \delta \right]^\beta}, \quad \gamma, H, \delta, \beta > 0\end{aligned}\tag{4.34}$$

where: $\alpha > 0 \in \mathfrak{R}$ is a fixed constant to control the speed of convergence, $TH \in (0, +\infty)$ and $k > 0$. Note that $\hat{g}_{ij} \in (0, H/\delta^\beta]$ and therefore it is more useful to limit TH by the same bound, i.e., $TH \in (0, H/\delta^\beta)$. The dynamics of the i^{th} agent in the other Cartesian coordinates can be given in a similar manner. For now, let $\gamma = H = \delta = \beta = 1$.

Note 4.10: The \hat{g}_{ij} function used in (4.34) could be any function in general if it satisfies the condition (3.34) and (3.35).

In (4.34), new equilibria set is produced besides the one already existing in the simplest semi-linear consensus protocol (4.5). This is like (4.16) in which new equilibria set is intentionally introduced to (4.5). So, the stability of the MAS resulting from (4.34) is guaranteed by the following theorem.

Theorem 4.5: The 3-D MAS resulting from the deployment protocol (4.34) is always stable and the average value of agents positions is time-invariant.

Proof: The new equilibria set- a lamina $\mathfrak{S}_{\{i,j\}}^x$ - of each edge, i.e., $\{i, j\} \in \mathcal{E}$, introduced by the term $\tanh\left(k(TH - \hat{g}_{ij})\right)$ is given as follows:

$$\mathfrak{S}_{\{i,j\}}^x = \left\{ x_i, x_j \mid x_j \leq x_i \pm \sqrt{\frac{1}{TH} - 1} \right\} \quad (4.34.1)$$

Which is depicted in Figure 4.22. Note that when $TH = 0$, then the MAS becomes unstable, and when $TH = 1$, then a complete average consensus is achieved.

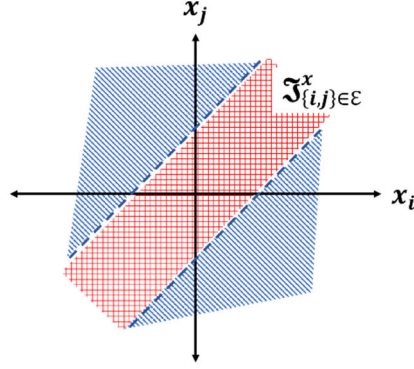


Figure 4.22. The equilibria set (the lamina in red) introduced by (4.34).

The overall equilibria set of all edges are given by the union of equilibria sets at edge level.

Mathematically, it is given as follows:

$$\mathfrak{J}_{Deployment}^x = \bigcup_{\{i,j\} \in \mathcal{E}} \mathfrak{J}_{\{i,j\}}^x \quad (4.34.2)$$

Similarly, the $\mathfrak{J}_{Deployment}^y$ and $\mathfrak{J}_{Deployment}^z$ can be found. Note that $x_i = x_j \in \mathfrak{J}_{\{i,j\}}^x$ for each edge, and therefore the $\vec{1} \in \mathfrak{J}_{Deployment}^x$ and is stable/unstable if $\mathfrak{J}_{Deployment}^x$ is stable/unstable, respectively.

By examining the Jacobian of the resulting MAS utilizing (4.34), it is straightforward to see that once all agents are on or outside the boundary of the overall equilibria set, i.e., $\partial \mathfrak{J}_{Deployment}^x$, then the Jacobian is a negative *M-matrix*, i.e., a *Metzler-matrix* usually used in stability analysis of time-delayed differential equations and positive linear dynamical systems. Therefore, the stability of the MAS resulting from (4.34) can follow the steps found in (3.83).

The time-invariance of the MAS under (4.34) can be easily seen from the fact that the resulting state-dependent Laplacian matrix has $\vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ which is sufficient for the MAS to preserve its average value for $\forall t \geq t_0$ as explained in (3.13) and the discussion that followed.

■

If the application requires the deployment to preserve another type of mean other than the average value of the agents initial conditions, then this will be possible using a modified version of (4.34) according to the following theorem.

Theorem 4.6: The 3-D MAS resulting from the following deployment protocol can have any type of mean as time-invariant with $g(x_i)$ given as $1, x_i, x_i^2$ and x_i^{1-p} to achieve arithmetic, geometric, harmonic and power mean-of-order- p , respectively.

$$\dot{x}_i = -\alpha g(x_i) \sum_{j \in \mathcal{N}_i} g_{ij}(x_j, x_i) \{x_i - x_j\} \quad (4.35)$$

Proof: Having (4.34) modified as given in (4.35), it is possible to proof Theorem 4.6 by following the steps shown in (3.39) – (3.42.9) which simply requires (4.34) to have $\vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ which is indeed the case since g_{ij} in (4.34) satisfies both (3.34) and (3.35).

Note that the remaining Euclidean directions follow in the same manner.

■

Before examining the effect of the deployment protocol (4.34) on the connectivity of the overall MAS, it is convenient to first have a look- from an agent perspective- on the contribution that each connected edge affords to the overall connectivity of the MAS. This is the subject of the following subsection.

4.1.3.1: An edge contribution in MAS connectivity preservation

Recalling Figure 3.5, the edge connecting agents 1 and 2 is undirected and it is how the action-reaction between the two agents is displayed. Among all edges available in a graph, the so-called critical links [138]- which are not unique [139]- are decisive in studying the

structural controllability of a MAS under leader-follower setup that suffers from concurrent failure of both agents and links. According to [140], a MAS is structurally controllable if and only if its fixed graph is connected. Knowing that λ_{2f} is a measure of graph connectivity [85], it is incorporated in various studies of MAS like structural controllability and observability and- most importantly in our case- the state-dependent networks.

Recently, in [61], a robust connectivity preservation control law for MAS has been presented. In that study, as well as in [64], it was shown that the modified Fiedler eigenvalue associated with the state-dependent Laplacian matrix is bounded from below as follows:

$$\lambda_2(\vec{x}) = \alpha(\vec{x}(t_0)) \lambda_{2f} \quad (4.36)$$

where: $\alpha(\vec{x}(t_0))$ is the value of the state-dependent weights evaluated at initial conditions such that [61]:

$$0 < \alpha(\vec{x}(t_0)) \leq \max_{\{i,j\} \in \mathcal{E}} \alpha(|x_i - x_j|) \quad (4.36.1)$$

Clearly, for (4.36) to be true $\forall \vec{x}$, $\alpha(|x_i - x_j|)$ must be increasing. This is similar to what was proposed in (3.97) like which semidefinite programming in [141] was used to maximize $\lambda_2(\vec{x})$ of a state-dependent graph. Compare (4.36) to (3.81.1) which is given as a linear combination of a scaled λ_{2f} and initial state-dependent weight.

Motivated by the previous discussion, we would like to examine the role each edge has in the overall MAS connectivity under state-dependent semi-linear behavioral protocols $\vec{f}(\vec{x})$.

Recalling **Theorem 3.8**, it is straight forward to show that $\lambda_2(\vec{x})$ of $\vec{f}(\vec{x})$, consider (3.67) with $\varphi_{12}(x_1, x_2) = x_2 - x_1$, is given as follows:

$$\vec{\dot{x}}(t) = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \alpha \begin{bmatrix} g_{12}(x_1, x_2)\{x_2 - x_1\} \\ g_{12}(x_1, x_2)\{x_1 - x_2\} \end{bmatrix} = \vec{f}(\vec{x}) \quad (4.36.2)$$

Therefore, the Jacobian of $\vec{f}(\vec{x})$ is given as:

$$\frac{\partial \vec{f}}{\partial \vec{x}} = \alpha \begin{bmatrix} \frac{\partial g_{12}}{\partial x_1}(x_2 - x_1) - g_{12} & \frac{\partial g_{12}}{\partial x_2}(x_2 - x_1) + g_{12} \\ \frac{\partial g_{21}}{\partial x_1}(x_1 - x_2) + g_{21} & \frac{\partial g_{21}}{\partial x_2}(x_1 - x_2) - g_{21} \end{bmatrix} \quad (4.36.3)$$

As forced by the proposed framework, we have $g_{12} = g_{21}$ - by the first integral method- and $\frac{\partial g_{12}}{\partial x_1} = -\frac{\partial g_{12}}{\partial x_2}$ and $\frac{\partial g_{12}}{\partial x_1} = \frac{\partial g_{21}}{\partial x_1}$ both by the nonlinear eigenvalue approach. So, the edge contribution can be found by calculating the following:

$$\lambda_2^{\{1,2\}} = \vec{v}_{2f}^T \frac{\partial \vec{f}}{\partial \vec{x}} \vec{v}_{2f} \quad (4.36.4)$$

After simplification, $\lambda_2^{\{1,2\}}$ is given as follows:

$$\lambda_2^{\{1,2\}}(\vec{x}) = \alpha \left\{ \frac{\partial g_{12}}{\partial x_1} \{x_2 - x_1\} - g_{12} \right\} (v_1 - v_2)^2 \quad (4.37)$$

where: $v_{1,2}$ are the elements of the linear left eigenvector associated with $\lambda_{2f} = -2$.

Having N agents in the MAS, (4.37) can be extended to include the contribution of these graph edges in the overall MAS connectivity as follows- assuming all edges are using the same parameters and structure of g_{ij} function:

$$\begin{aligned} \lambda_2(\vec{x}) &= \alpha \sum_{\{i,j\} \in \mathcal{E}} \lambda_2^{\{i,j\}} \\ &= \alpha \sum_{\{i,j\} \in \mathcal{E}} \left\{ \frac{\partial g_{ij}}{\partial x_i} \{x_j - x_i\} - g_{ij} \right\} (v_i - v_j)^2 \end{aligned} \quad (4.38)$$

This result is given by the following theorem.

Theorem 4.7: A MAS dynamical system whose Jacobian is symmetric has its $\lambda_2(\vec{x})$ given by (4.38).

Proof: Having $N \times N$ symmetric Jacobian is guaranteed if and only if both (3.34) and (3.35) are satisfied. Therefore:

$$\lambda_2(\vec{x}) = \alpha \vec{v}_{2f}^T \frac{\partial \vec{f}}{\partial \vec{x}} \vec{v}_{2f} \quad (4.38.1)$$

can be written as follows:

$$\lambda_2(\vec{x}) = \alpha \sum_{i=1} v_i \sum_{j=1} v_j \frac{\partial f_j}{\partial x_i}, \quad \forall \{i, j\} \in \mathcal{E} \quad (4.38.2)$$

where: $\vec{v}_{2f}^T = [v_1, v_2, \dots, v_N]$. Dealing with undirected graphs, then we may break down (4.38.2) into pairs indicating that we came back to the edge level instead of the MAS level. Therefore, (4.38.2) can be rearranged and written with respect to any edge $\{i, j\} \in \mathcal{E}$ as follows:

$$\lambda_2 = \alpha \sum_{\forall \{i, j\} \in \mathcal{E}} \left\{ v_i^2 \frac{\partial f_i}{\partial x_i} + v_i v_j \frac{\partial f_i}{\partial x_j} + v_i v_j \frac{\partial f_j}{\partial x_i} + v_j^2 \frac{\partial f_j}{\partial x_j} \right\} \quad (4.38.3)$$

From (3.34) and (3.35), we find that:

$$\begin{aligned} \frac{\partial f_i}{\partial x_i} &= \frac{\partial f_j}{\partial x_j} = -\frac{\partial f_i}{\partial x_j} \\ \frac{\partial f_i}{\partial x_j} &= \frac{\partial f_j}{\partial x_i} \end{aligned} \quad (4.38.4)$$

So, (4.38.3) becomes:

$$\lambda_2(\vec{x}) = \alpha \sum_{\{i, j\} \in \mathcal{E}} \lambda_2^{\{i, j\}} = \alpha \sum_{\{i, j\} \in \mathcal{E}} \frac{\partial f_i}{\partial x_i} (v_i - v_j)^2 \quad (4.38.5)$$

which is valid for general nonlinear protocols. If a semi-linear protocol is used, then (4.38.5) can be written as follows:

$$\alpha \sum_{\{i, j\} \in \mathcal{E}} \lambda_2^{\{i, j\}} = \alpha \sum_{\{i, j\} \in \mathcal{E}} \left\{ \frac{\partial g_{ij}}{\partial x_i} \{x_j - x_i\} - g_{ij} \right\} (v_i - v_j)^2 \quad (4.38.6)$$

and consequently (4.38) follows. ■

If the g_{ij} function uses the C-S model (3.116) with $H = \gamma = \delta = \beta = 1$, then g_{ij} will be strictly increasing and consequently (4.36) is indeed sufficient to proof the connectivity of the MAS under the resulting semi-linear state-dependent protocol. Moreover, (4.37) can be given as follows:

$$\lambda_2^{\{1,2\}}(\vec{x}) = 2\alpha \frac{(x_2 - x_1)^2 - 1}{[(x_2 - x_1)^2 + 1]^2} \quad (4.39)$$

where: $v_1 = 1/\sqrt{2}$ and $v_2 = -1/\sqrt{2}$. The result is shown in Figure 4.23 where $\alpha = 1$.

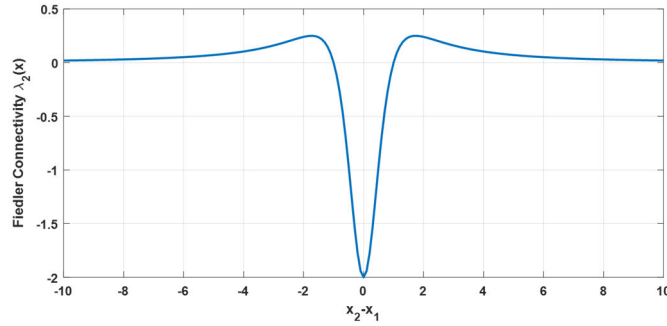


Figure 4.23. The value of $\lambda_2^{\{1,2\}}(\vec{x})$ of edge $\{1,2\} \in \mathcal{E}$ using C-S model.

In general, we are more interested in the invariance of the $\lambda_2^{\{i,j\}}(\vec{x})$ sign than in its value to prove edge connectivity. If the sign does not change, then $\lambda_2^{\{i,j\}}(\vec{x})$ is never zero and the edge remains connected. However, the value of $\lambda_2^{\{1,2\}}(\vec{x})$ can be related to the stability of edge $\{1,2\} \in \mathcal{E}$, for example, as follows:

$$\frac{\partial \vec{f}}{\partial \vec{x}} = 0.5 \begin{bmatrix} \lambda_2^{\{1,2\}} & -\lambda_2^{\{1,2\}} \\ -\lambda_2^{\{1,2\}} & \lambda_2^{\{1,2\}} \end{bmatrix} \quad (4.39.1)$$

From which when $\vec{x} = c\vec{1}, c \geq 0 \in \mathfrak{R}$, then (4.39.1) will be negative semi-definite and the edge is stable, and when $(x_2 - x_1)^2 = 1$, then the edge becomes inactive or disconnected and agents will reside stationary unless at least one of them is connected to a third agent

over an active edge. Figure 4.24 shows the phase portrait of (4.36.3) with g_{ij} function uses the C-S model (3.116) with $H = \gamma = \delta = \beta = 1$.

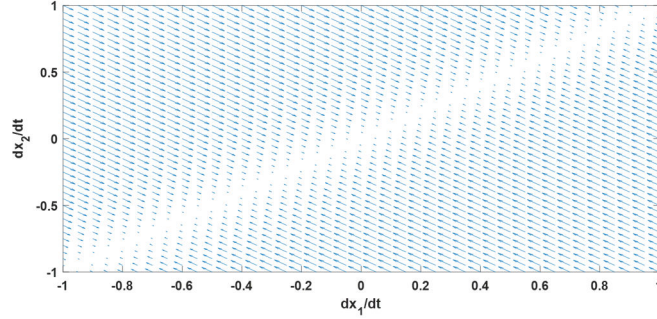


Figure 4.24. The phase portrait of (4.36.3) with C-S model used in g_{ij} functions.

At this stage, it is good to examine the parameterized version of the C-S model given in (3.116). This will give a clear insight about how the parameters involved get to interplay with the connectivity of an edge. Such knowledge helps us optimizing the parameters to maximize the connectivity, if desired. Doing the Math, yields:

$$\lambda_2^{\{1,2\}}(\vec{x}) = 2H\alpha \frac{\gamma(2\beta - 1)(x_2 - x_1)^2 - \delta}{[\gamma(x_2 - x_1)^2 + \delta]^{\beta+1}} \quad (4.39.2)$$

Clearly, setting $\beta = 0.5$ will cancel the zero from (4.39.2) and therefore the edge will always remain active or connected. The result is shown in Figure 4.25. This leads us to the following theorem.

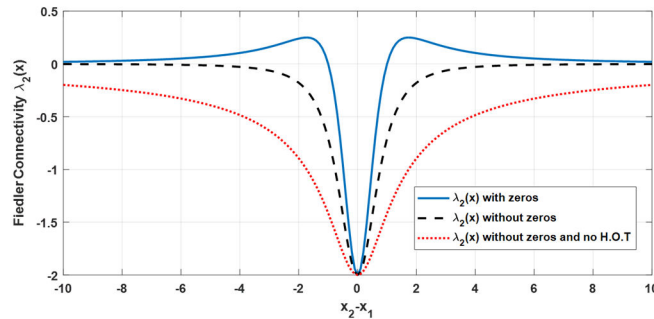


Figure 4.25. The value of $\lambda_2^{\{1,2\}}(\vec{x})$ of edge $\{1,2\} \in \mathcal{E}$ using C-S models, (solid): (4.39), (dashed): (4.39.2) with $\beta = 0.5$, (dotted): (4.39.2) with $\beta = 0.5$ and neglecting the higher-order-terms (H.O.T), all with $\gamma = H = \delta = \alpha = 1$.

Theorem 4.8: A MAS controlled using a semi-linear protocol comprising the parameterized C-S model with $\beta = 0.5$ will always remain connected if and only if the initial graph is connected.

Proof: The state-dependent graph connectivity is measured by $\lambda_2(\vec{x})$ given by (4.38) as a sum of all edges contributions using their own connectivity measures. Governed by (4.39.2) with $\beta = 0.5$, then all edges are always active and therefore $|\lambda_2(\vec{x})| > 0, \forall \vec{x}(t)$ if and only if $\forall i, j \in \mathcal{E}$ we have $(v_i - v_j) \neq 0$. ■

Generalizing (4.39.2), we may calculate in advance the expected value of algebraic connectivity at average consensus. This is given as follows:

$$\lambda_2^{\{1,2\}}(\vec{1}) = -\frac{\gamma H \alpha}{\delta \beta} (v_i - v_j)^2 \quad (4.39.3)$$

and the parameters can be chosen as desired. Note that (3.32) is a special case in the sense of (4.39.3).

Relaxing the conservative condition given by **Theorem 4.8**, the following lemma can be used to ensure the connectivity of a MAS utilizing the same protocol. It reads as follows:

Lemma 4.3: Borrowing the definition of critical links from [138], a MAS controlled using a semi-linear protocol comprising the parameterized C-S model with $\beta = 0.5$ over unsigned graphs will always remain connected if and only if all critical links remain active.

Proof: Using (4.39.2), an edge has its $\lambda_2^{\{i,j\}} > 0$ when it is active, and $\lambda_2^{\{i,j\}} = 0$ when it is inactive. So, focusing only on the contributions of all critical links to be greater than zero allows us to have an insight about the true value of $\lambda_2(\vec{x})$ whether being zero or not. Mathematically, this can be done by taking the geometric mean of the connectivity of all critical edges as follows:

$$\lambda_{2g}(\vec{x}) = \sqrt[k]{\prod_{\{i,j\} \in \mathcal{E}} |\lambda_2^{\{i,j\}}(\vec{x})|} \quad (4.40)$$

where: k is the number of critical edges in the graph. Thus, if $\lambda_{2g}(\vec{x}) > 0$, then consequently $\lambda_2(\vec{x}) > 0$ and the MAS is connected, else the MAS is disconnected. ■

Note 4.10: Designing a distributed protocol to achieve (4.40) is beyond the scope of this investigation.

Proposition 4.4: The signature of the linear eigenvalues of the fixed Laplacian matrix, i.e., \mathcal{L}_f , can be used along with their associated linear left or right eigenvectors, i.e., v_f^T, \vec{e}_f , to monitor their nonlinear eigenvalues counterparts in the state-dependent Laplacian matrix, i.e., $\mathcal{L}_s, \forall \vec{x}(t), t \geq t_0$, as follows:

$$\begin{aligned} \lambda_r(\vec{x}) &= \alpha \sum_{\{i,j\} \in \mathcal{E}} \lambda_r^{\{i,j\}} \\ &= \alpha \sum_{\{i,j\} \in \mathcal{E}} \left\{ \frac{\partial g_{ij}}{\partial x_i} \{x_j - x_i\} - g_{ij} \right\} (v_i - v_j)_r^2 \end{aligned} \quad (4.41)$$

where: $r = \{1, 2, \dots, N\}$ denotes the eigenvalue index. In (4.41), it is assumed that all edges are using the same parameters and structure of the g_{ij} function. If this is not the case, then each different edge should be treated separately. ■

Note 4.11: Note that in (4.41), when $r = 0$, then we have $v_f^T = \vec{1}^T$ and $\vec{e}_f = \vec{1}$, and therefore $\lambda_1(\vec{x}) = 0 \forall \vec{x}(t), t \geq t_0$, as expected.

Note 4.12: In Part I, figures 2.14 and 2.23.b displayed the point-wise evolution of $\lambda_2(\vec{x})$ using the $eig(\mathcal{L}(\vec{x}))$ MATLAB function. This can be justified knowing the fact that the

g_{ij} functions used were all according to C-S model, i.e., strictly increasing. Therefore, $\text{eig}(\mathcal{L}(\vec{x}))$ can be considered- in worst cases- as a lower bound of the actual $\lambda_2(\vec{x})$ according to (4.36).

Remark 4.7: Having **Note 4.12** in mind, one may wonder about the reason why we have discussed in more details the contribution of an individual edge in $\lambda_2(\vec{x})$ if we have (4.36) in hand from the first place. Well, among other benefits that might emerge in future investigations, the most immediate response to such a question arises from the need to understand the behavior of $\lambda_2(\vec{x})$ when its evolution is not strictly increasing, i.e., when (4.36) fails to lower bound $\lambda_2(\vec{x})$. Such cases do exist in applications, consider the case in Figure 3.15 where the evolution of $\lambda_2(\vec{x})$ was not increasing $\forall \vec{x}(t)$ while achieving the harmonic and geometric means. So, there is a need to understand the situation for a general g_{ij} - like (4.34)- and mean type- or general $g(x_i)$ function that might be needed- such that a lower bound of $\lambda_2(\vec{x})$ can be certain.

Under (4.35), the evolution of $\lambda_r(\vec{x})$ is given as follows:

$$\lambda_r(\vec{x}) = \alpha \sum_{\{i,j\} \in \mathcal{E}} \lambda_r^{\{i,j\}} = \alpha \sum_{\{i,j\} \in \mathcal{E}} \left\{ \frac{\partial g_{ij}}{\partial x_i} \{x_j - x_i\} - g_{ij} \right\} m_r^1 + g_{ij} \{x_j - x_i\} m_r^2 \quad (4.42)$$

where:

$$m_r^1 = v_{i,r}^2 g(x_i) - v_{i,r} v_{j,r} \{g(x_i) + g(x_j)\} + v_{j,r}^2 g(x_j) \quad (4.42.1)$$

$$m_r^2 = v_{i,r}^2 \frac{\partial g(x_i)}{\partial x_i} - v_{j,r}^2 \frac{\partial g(x_j)}{\partial x_j} \quad (4.42.2)$$

Now, we are in position to state the following corollary:

Corollary 4.1: A MAS controlled using a semi-linear protocol (4.35) comprising the parameterized C-S model with $\beta = 0.5$ will always remain connected while achieving the arithmetic, geometric and harmonic means if and only if the initial graph is connected.

Proof: Utilizing (4.42), it is straight forward to see that when $g(x_i) = g(x_j) = 1$, then the MAS will achieve the average consensus while always being connected as a direct consequence of **Theorem 4.8**. The general form of (4.42) under (4.35) when $\beta = 0.5$ is given as follows:

$$\lambda_2^{\{1,2\}}(\vec{x}) = -\alpha H \frac{\gamma(x_1^{1-p} + x_2^{1-p})\delta + m_3}{\sqrt{[\gamma(x_2 - x_1)^2 + \delta]^3}} \quad (4.43)$$

where:

$$m_3 = 0.5(x_2 - x_1)(1 - p)(x_2^{1-p} - x_1^{1-p})[\gamma(x_2 - x_1)^2 + \delta] \quad (4.43.1)$$

For the geometric mean, we have $p = 0$ and therefore (4.43) can be given as follows:

$$\lambda_2^{\{1,2\}}(\vec{x}) = -\alpha H \frac{\gamma(x_2 + x_1)\delta}{\sqrt{[\gamma(x_2 - x_1)^2 + \delta]^3}} \quad (4.43.2)$$

Considering (3.18) and *Remark 1.4*, it is clear that (4.43.2) has no zeros, and therefore an edged will stay connected while achieving the geometric mean under (4.35) when $\beta = 0.5$.

For the harmonic mean, we have $p = -1$ and therefore (4.43) can be given as follows:

$$\lambda_2^{\{1,2\}}(\vec{x}) = -\gamma\alpha H \frac{(x_1^2 + x_2^2)\delta + m_4}{\sqrt{[\gamma(x_2 - x_1)^2 + \delta]^3}} \quad (4.43.3)$$

with $m_4 = (x_2 - x_1)^2[\gamma(x_2 - x_1)^2 + \delta]$ and $\delta/\gamma = 1$. Considering (3.22), it is clear that (4.43.3) has no zeros, and therefore an edge will stay connected while achieving the harmonic mean under (4.35) when $\beta = 0.5$.

■

Remark 4.8: Considering **Corollary 4.8**, the conditions needed, if possible, to achieve the general power mean-of-order- p can be investigated using the same steps shown earlier in (4.43).

4.1.3.2: The deployment protocol and MAS connectivity

In the previous subsection, we have investigated in detail the contribution of an edge in the overall MAS connectivity during consensus. During consensus, agents must come closer to each other as time passes till they reach the consensus. On the contrary, during the deployment protocol, agents may come closer or move apart before reaching their steady states behaviors. Therefore, unlike the case with consensus, edges are expected to have some periods where their dynamics can be labeled unstable and $\lambda_2^{\{i,j\}}$ could be positive, negative or instantly zero.

To avoid repetition of mathematical derivation, here we will depend on the graphical visualization to grasp on the differences between the deployment and consensus protocols. Of course, the needed mathematical equations to facilitate understanding will be presented. Figure 4.26 depicts a mechanical analogy of the deployment protocol. In this analogy, the massless spring whose stiffness is nonlinear in general plays the role of the deployment protocol where its two ends represent the connected agents over that edge. When the spring is initially extended, it is expected that the two ends, i.e., agents, will move inwardly to consume the initial energy put into the system. On the opposite, when the two ends are nearby each other, then it is expected to observe outward movement of the two ends till they come back to the natural length of the spring. It happens that the parameters in a way, yet to be explained, specify this length.

The dynamics of the edge $\{i, j\}$ shown in Figure 4.26 are governed by $\vec{f}(\vec{x})$ given in (4.36.3), and g_{ij} functions given by (4.34). Its phase portrait is shown in Figure 4.27.

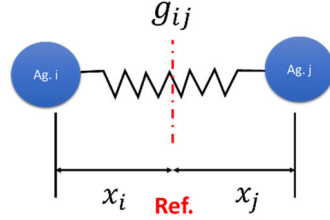
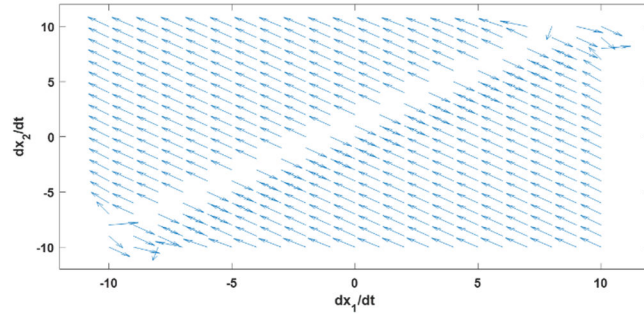
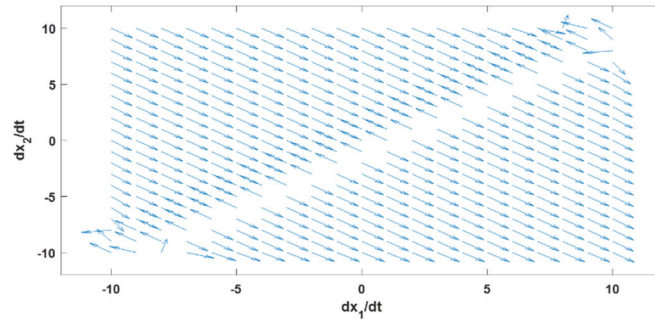


Figure 4.26. Mechanical analogy of edge $\{i, j\}$ under deployment protocol with the reference exactly at the average value of the two agents.



(a)



(b)

Figure 4.27. Phase portrait of: (a) the first agent. (b) the second agent both under deployment protocol (4.34) with $TH = 0.1$, $H = \delta = \gamma = 1$ and $k = 10$.

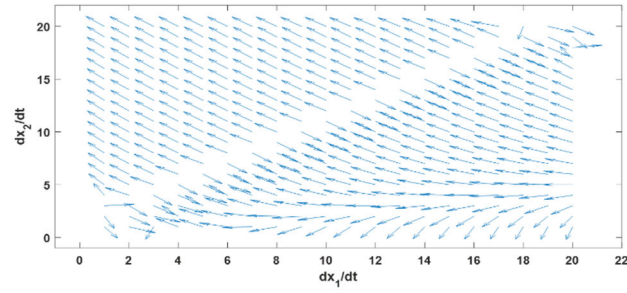
Overlapping figures 2.27.a and 2.27.b yields the lamina-shown in Figure 4.22- whose width (Δ) is controlled via the parameters available in (4.34) and can be given as follows:

$$\Delta = 2 \sqrt{\frac{1}{\gamma} \left(\frac{H}{TH} - \delta \right)} \quad (4.44)$$

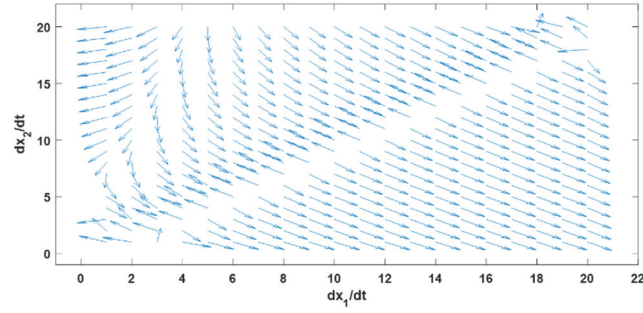
where the borders of the lamina at which agents will settle is given as follows:

$$x_j = x_i \pm 0.5\Delta \quad (4.44.1)$$

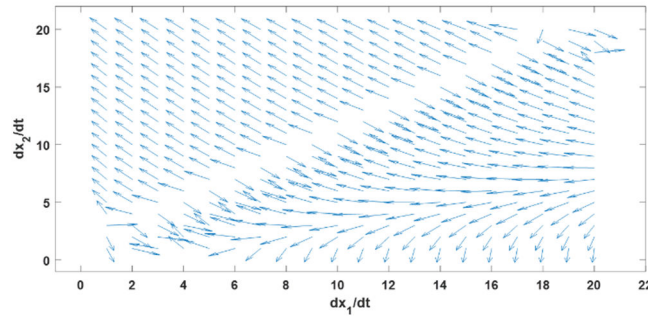
Note 4.13: Choosing the parameters in (4.44) such that $\Delta = 0$ results in a complete consensus among agents. Note also that the $\vec{1}$ is inside the lamina.



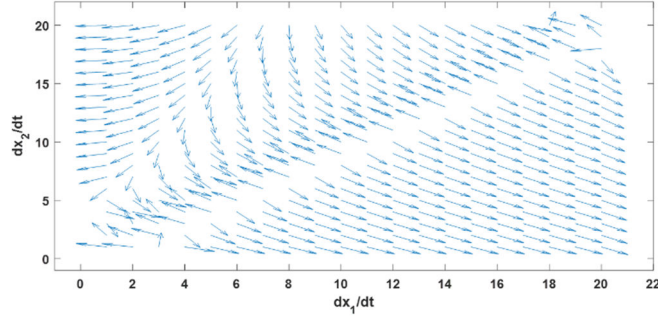
(a)



(b)



(c)



(d)

Figure 4.28. Phase portrait of: (a) the first agent. (b) the second agent both under deployment protocol (4.35) with $g(x_i) = x_i$, i.e., geometric mean deployment. (c) the first agent. (d) the second agent both under deployment protocol (4.35) with $g(x_i) = x_i^2$, i.e., harmonic mean deployment all with $TH = 0.1$, $k = 10$ and $H = \delta = \gamma = 1$.

The phase portrait under (4.35) is given in Figure 4.28 where it is obvious how $g(x_i)$ shapes the lamina while achieving the geometric and harmonic deployment, i.e., these means are preserved under the final positions of the deployed agents in the working space. Empowered by this knowledge, here, in this section, the effect of the deployment protocol (4.34) on the overall MAS connectivity can be unswervingly addressed. Using (4.35) with the g_{ij} functions given in (4.34), the contribution of the edge $\{1,2\}$ in $\lambda_2^{\{i,j\}}(\vec{x})$ of $\vec{f}(\vec{x})$ given in (4.36.3), can be given as follows:

$$\lambda_r(\vec{x}) = \alpha \sum_{\{i,j\} \in \mathcal{E}} \lambda_r^{\{i,j\}} = \alpha \sum_{\{i,j\} \in \mathcal{E}} n^1 m_r^1 + \frac{H \tanh(\cdot)(x_j - x_i)}{d^\beta} m_r^2 \quad (4.45)$$

where:

$$n^1 = \frac{2\gamma\beta H(x_j - x_i)^2}{d^{\beta+1}} \left\{ \frac{kH \operatorname{sech}^2(\cdot)}{d^{2\beta-1}} + \tanh(\cdot) \right\} - \frac{H \tanh(\cdot)}{d^\beta}$$

$$m_r^1 = v_{i,r}^2 g(x_i) - v_{i,r} v_{j,r} \{g(x_i) + g(x_j)\} + v_{j,r}^2 g(x_j) \quad (4.45.1)$$

$$m_r^2 = v_{i,r}^2 \frac{\partial g(x_i)}{\partial x_i} - v_{j,r}^2 \frac{\partial g(x_j)}{\partial x_j} \quad (4.45.2)$$

$$d = \gamma(x_j - x_i)^2 + \delta$$

The basis function of an edge connectivity, i.e., $\lambda_2^{\{i,j\}}$, as a function of the relative distance $x_j - x_i$ is shown in Figure 4.28. An edge connectivity basis function is given by n^1 in (4.49).

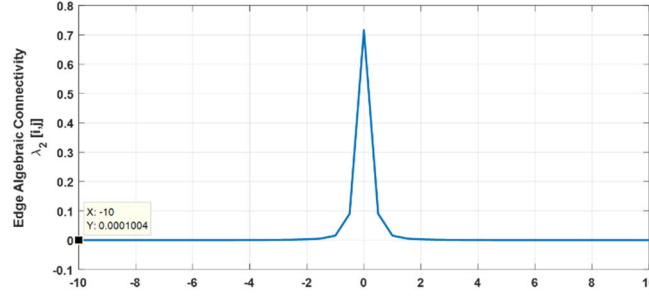


Figure 4.29. The basis function of $\lambda_2^{\{i,j\}}(\vec{x})$ of edge $\{i,j\} \in \mathcal{E}$ using (4.49) with $TH = 0.1$, $H = \delta = k = \beta = 1$ and $\gamma = 10$.

Using $H = \delta = k = \beta = 1$, $\alpha = 20$ and $\gamma = 10$ in (4.45) results in a lamina width of $\Delta = 1.8974$. Notice that, unlike the C-S model, in (4.45) we are unable to attain further simplification due to the complexity involved and the parameters must be tuned manually or via dedicated equation solving routines such that $\lambda_r^{\{i,j\}} > 0, \forall x_i, x_j$ in the domain of interest. Here, we have tuned the parameters manually. As a result, the connectivity of the MAS in 3-D under the deployment protocol given in (4.34) is always ensured. This is stated in the following theorem.

Theorem 4.9: The 3-D MAS resulting from the deployment protocol (4.34) is always connected if and only if its associated graph is initially connected.

Proof: Having an initially connected graph results in $\lambda_{2f} > 0$ and therefore the associated eigenvectors of λ_{2f} , i.e., \vec{v}_{2f} is other than the $\vec{1}$. Along with \vec{v}_{2f} , a good parameterized deployment protocol (4.34) can be used when the involved parameters are tuned such that

(4.45) is positive in the domain of interest. Thus, each edge whether critical or not will always stay connected. This will result in $\lambda_2(\vec{x}) > 0, \forall \vec{x}$ and therefore, from a graph theoretic point of view, the MAS will stay connected. ■

Note 4.14: Using \vec{v}_{2f} to monitor the evolution of $\lambda_2(\vec{x})$ is not restrictive. In fact, we may keep the general nonlinear eigenvector $\vec{v}_2(\vec{x})$ and use it in place of \vec{v}_{2f} and still obtain the same deduction about the connectivity of the MAS when $g(x_i) = 1$. Otherwise, if for any reason $\vec{v}_2(\vec{x})$ must be used, then solving the nonlinear differential equations (3.29) and (3.30) is unavoidable.

Observing the similarity between (4.16) and (4.34), modified versions of both (4.16) and (4.32) can be used to deal with situations where a prescribed performance is required, or bounded control is inevitable. This is given as follows:

Proposition 4.5: Inspired by (4.34), the formations controls (4.16) and (4.32) can be modified to accommodate the requirements of bounded control and prescribed performance as follows:

$$u_i = \sum_{j \in \mathcal{N}_i} \varpi \tanh(k(d_{ij}^{*2} - d_{ij}^2)) \frac{\{x_j - x_i\}}{\sqrt{(d_{ij}^{*2} - d_{ij}^2)^2 + 1}} \quad (4.16')$$

$$u_i = \sum_{j \in \mathcal{N}_i} \varpi \tanh\left(\frac{k(d_{ij}^{*2} - d_{ij}^2)}{(d_{ij}^{*2} - d_{ij}^2)^2 + 1}\right) \frac{\{x_j - x_i\}}{\sqrt{(d_{ij}^{*2} - d_{ij}^2)^2 + 1}} \quad (4.32')$$

where: k is appropriately small- to avoid chattering nearby the equilibria- and $|u_i| \leq \varpi \in \mathbb{R}$.

Proof: By examining the conditions (3.34) and (3.35), we can straightforwardly see the applicability of (4.16') and (4.36') to be further used in (3.105) and (3.106). Therefore,

both (4.16') and (4.32') can be used when control effort is bounded or when a prescribed performance is required.

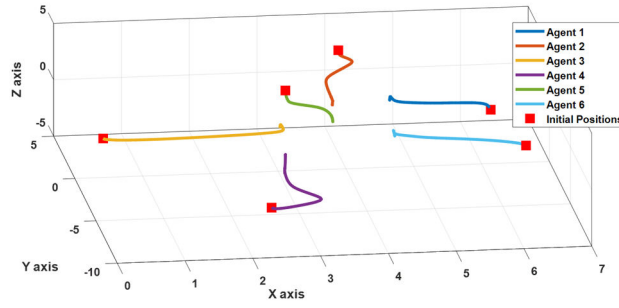
■

In (4.34), the \tanh function was used as a smooth approximation of the sign function to avoid chattering in the time-domain response of the MAS [142]. The i^{th} agent will slide along the sliding surface formed by the i^{th} and j^{th} agents. The resulting MAS can be classified as a variable structure system (VSS) that uses a distributed sliding mode controller (SMC) to slide along the surface given by $TH - \hat{g}_{ij}$. A detailed survey of VSS and SMC for motion control is available in [143]. A recent work on distributed SMC of MAS can be found in [144]. Note that (4.34) meets both (3.34) and (3.35).

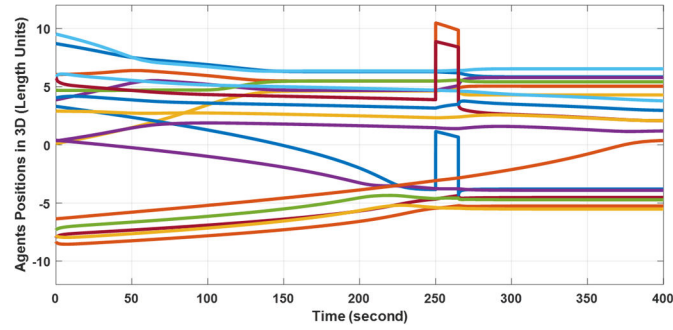
When agents are deployed in 3-D, they are expected to maximize their separations in a way that ensures their connectivity and average value to be maintained. Thus, the optimization of these separations is constrained only by the variable TH that reflects the desired connectivity level. When obstacles exist in the working space, then they will act as additional constraints to the optimization problem solved in a distributed fashion using (4.34). Figure 4.30 shows the deployment results of six single-integrator agents with $TH = 0.1$ and $\rho = 0.01$ where the communication network used is shown in Figure 3.13. Depending on the initial conditions, agents will move inwards- as shown in Figure 4.30.a- or outwards- as shown in Figure 4.30.d- with respect to a ball containing all the initial conditions while maintaining their average value and connectivity. If other means are desired, other than the arithmetic mean, then (4.8) can be used. In this deployment example, $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ were the identity matrix and $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ were identical. The final connectivity level reflected by $\lambda_2(\vec{x}(t))$ will be the same irrespective of the agents' initial positions as

shown in Figure 4.30.i when $t \rightarrow \infty$. However, the agents are assumed to be initially connected or another protocol must be run to make them connected, like the nearest neighbor protocol found in [117], for example.

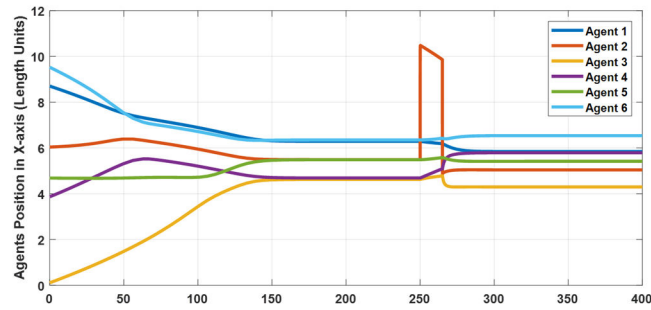
Note 4.15: In Figure 4.30.e, the final value of $\lambda_2(\vec{x})$ is not equal to $\lambda_{2f} = 6$ of the complete undirected graph used simply because this is not a consensus problem.



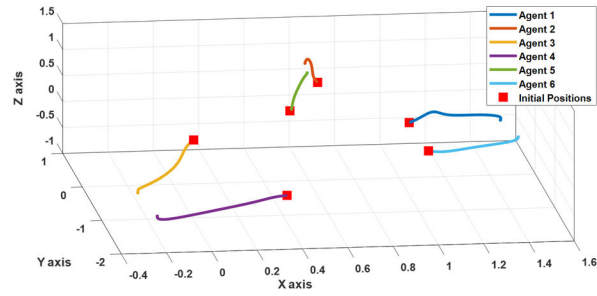
(a)



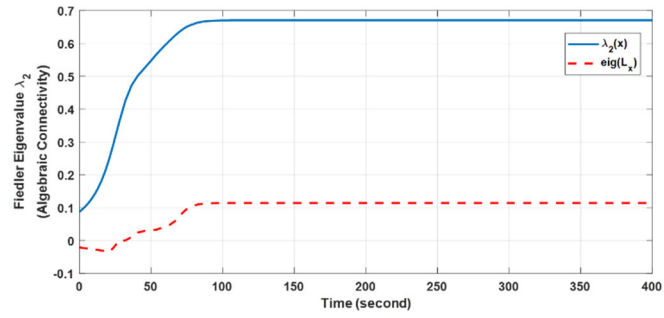
(b)



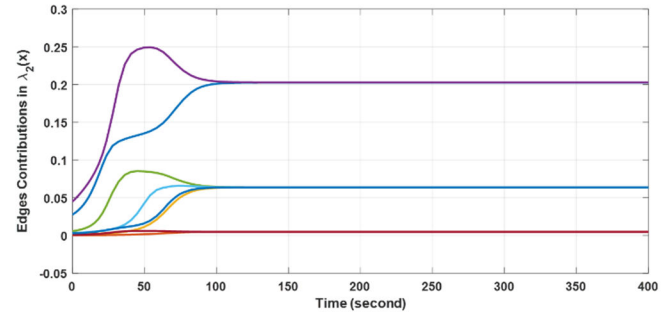
(c)



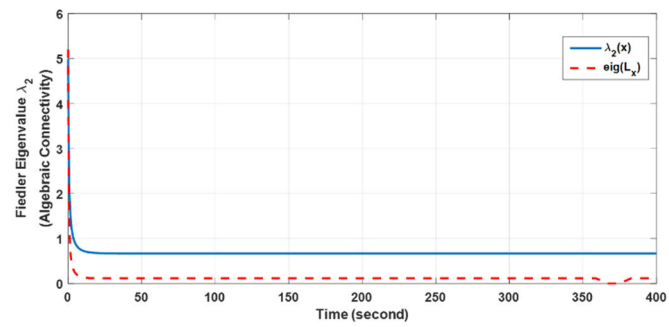
(d)



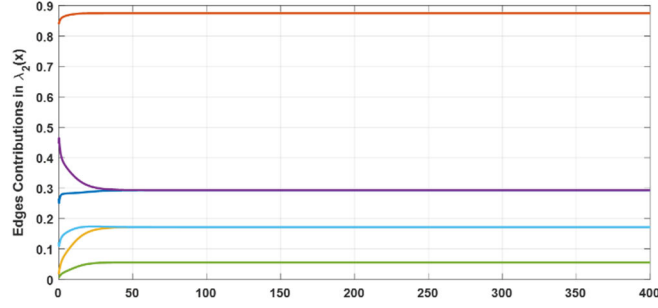
(e)



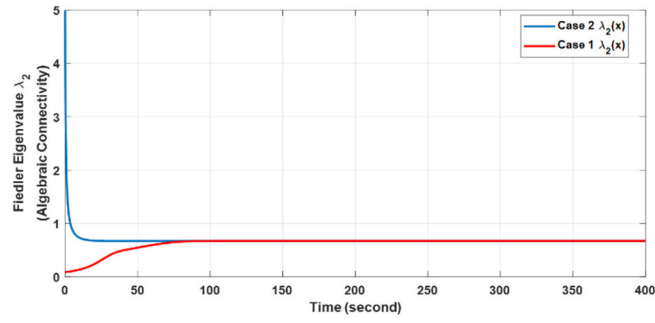
(f)



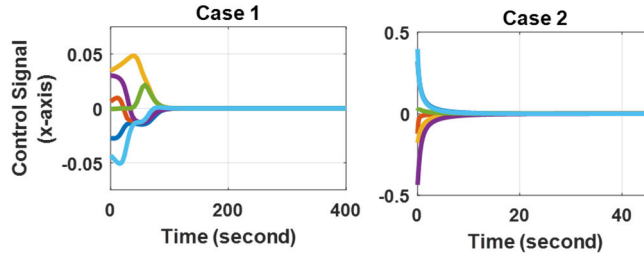
(g)



(h)



(i)

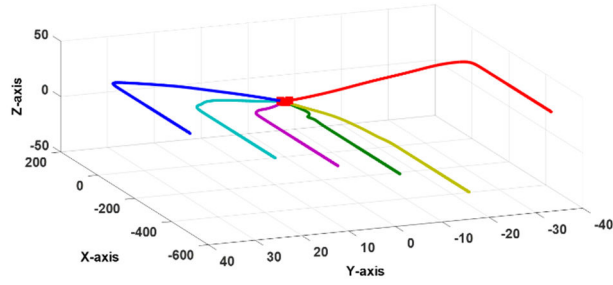


(j)

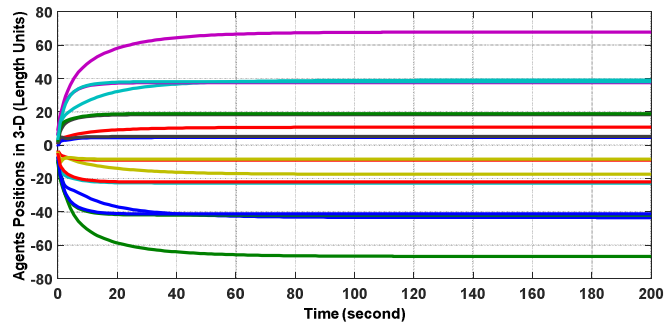
Figure 4.30. Deployment of six single-integrator agents in obstacle-free 3-D space. (a): Case 1 without disturbance. (b)-(c): Disturbance acting only on agent 1 at 100 second for 15 second duration under Case 1. (d): Case 2 deployment response when agents are initially close to each other. (e): the corresponding overall connectivity of Case 1. (f): edges contributions in the overall connectivity of Case 1. (g): the corresponding overall connectivity of Case 2. (h): edges contributions in the overall connectivity of Case 2. (i) connectivity of the two cases. (j) control signals in the two cases.

Note that in (4.34), if $TH = 0$ - or negative, then the MAS will become unstable and if $TH \rightarrow +\infty$, then the desired type of consensus is achieved. Thus, we may conclude that the

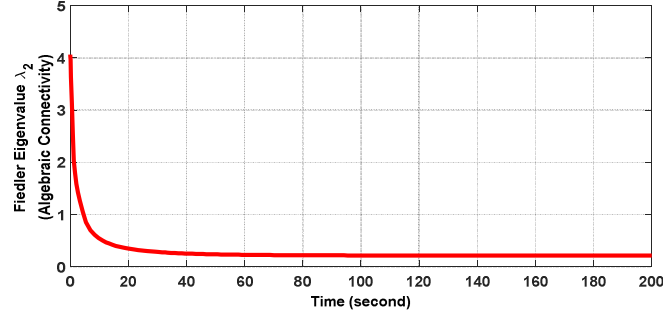
consensus behavior of the MAS is affected by TH since it changes the equilibria set and its characteristics. It was observed through numerical simulation that if each agent uses a different value of TH , then the MAS will behave in an interesting way as shown in Figure 4.31.a simply because the left eigenvector, i.e., $\vec{1}$, is not preserved. This behavior is believed to be related to the bifurcation phenomenon occurs within nonlinear systems, in general. If the same was done to agents in the y-direction, then a heading for the whole MAS can be specified. However, studying such behaviors is beyond the scope of this thesis. If different values of TH were used such that the $\vec{1}$ remains as a left eigenvector under the same communication network, then more separation can be achieved while maintaining the connectivity of agents and their desired mean value as shown in Figure 4.31.b and c.



(a)



(b)



(c)

Figure 4.31. MAS response when agents have different TH values in (4.34) along the x-axis. The $eig(\mathcal{L}(\vec{x}))$ MATLAB function was used.

In the previous section, the formation control was introduced in which less detailed geometry was given using (4.16) resulting in more geometric flexibility once compared to (4.23) in which detailed geometry must be provided. In this section, the deployment process can be viewed as an additional type of formation in which even lesser geometric constraints are provided once compared to (4.16). Both (4.16) and (4.34) impose geometric constraints on agents' separations; however, in a following section, the formation problem will be viewed from a different perspective in which the resulting formation, i.e., shape consensus, is constrained in the direction of separation rather than in its magnitude.

4.1.4: Connectivity-preserving protocols

In the previous section, the issue of MAS connectivity preservation was highlighted and useful equations that can be used to examine whether the protocols are connectivity-preserving or not were introduced. In this section, more details about the conditions needed for a behavioral protocol to be connectivity-preserving are provided.

Recall that a general single-edge system whose dynamics are given in (4.36.2) and the edge connectivity is given by (4.36.4) which can be other than zero $\forall \vec{x}$ if the edge weighting function used, i.e., g_{12} , is designed according to the following theorem.

Theorem 4.10: A behavioral semi-linear protocol satisfying both (3.34) and (3.45) and acting over an initially connected network graph is connectivity-preserving if and only if the following is true $\forall \vec{x}$ in the domain of interest.

$$\frac{\partial g_{ij}}{\partial x_i} \{x_j - x_i\} - g_{ij} \neq 0, \quad \forall i, j, \vec{x} \quad (4.46)$$

Proof: Having an initially connected graph results in $\lambda_{2f} > 0$ and therefore the associated eigenvectors of λ_{2f} , i.e., \vec{v}_{2f} is other than the $\vec{1}$. So, in (4.41), $(v_i - v_j)_2^2 \neq 0$. Thus, each edge whether critical or not will always stay connected if and only if (4.46) is true. This will result in $\lambda_2(\vec{x}) > 0, \forall \vec{x}$ and therefore, each edge will stay connected. Having g_{ij} that satisfies both (3.34) and (3.35) results in a symmetric Jacobian that made it easy to derive (4.46) as a sufficient condition for connectivity preserving under general semi-linear protocols. ■

Note 4.16: For asymmetric Jacobians resulting from semi-linear and nonlinear protocols, the same steps can be followed as shown in the previous section, and emphasized here, to decide whether such protocols can be connectivity-preserving or not.

To demonstrate the strength of **Corollary 4.1** and **Theorem 4.10**, the following subsections are provided in which deployment, formation and consensus protocols are revisited from a connectivity preserving point of view.

4.1.4.1: A connectivity-preserving deployment-formation-consensus-collision-avoidance (*universal*) behavioral protocol

Consider a deployment-formation-consensus protocol that is given as follows:

$$\dot{x}_i = \alpha \left[\frac{TH_{ij}}{(x_j - x_i)^2} - 1 \right] \{x_i - x_j\} = \alpha g_{ij} \{x_i - x_j\} \quad (4.47)$$

where: $\alpha > 0$ is a scalar that determines the speed of convergence and $TH_{ij} \geq 0$ is a real scalar parameter that plays a significant role in the behavior of the behavioral protocol (4.47). Obviously, $x_j = x_i$ is a singular point in (4.47) that can be removed if $TH_{ij} = 0$. Then, (4.47) becomes a consensus protocol. When $TH_{ij} \neq 0$, then collision is avoided whenever $x_j - x_i \rightarrow 0$. When $TH_{ij} = TH \forall i, j$, then (4.47) becomes a deployment protocol with separation at maximum equals \sqrt{TH} . Lastly, when TH_{ij} is specified for each edge $\{i, j\} \in \mathcal{E}$, then (4.47) becomes a formation protocol. It is straightforward to show that g_{ij} in (4.47) satisfies both (3.34) and (3.35) and therefore according to **Theorem 4.10** it can be used as a connectivity-preserving protocol.

According to (4.37), the edge $\{i, j\} \in \mathcal{E}$ connectivity is given as follows:

$$\lambda_2^{\{i,j\}}(\vec{x}) = \alpha \left\{ \frac{TH_{ij}}{(x_j - x_i)^2} + 1 \right\} (v_i - v_j)^2 > 0, \forall \vec{x} \quad (4.47.1)$$

Observe the similarities between (4.47) and the formation protocol (4.16), and between (4.47) and the deployment protocol (4.34). As $x_j - x_i \rightarrow 0$, the value of $\lambda_2^{\{i,j\}}(\vec{x})$ in (4.52.1) increases drastically, which is logical since the maximum connectivity occurs when the two agents reside on top of each other.

Being concerned about having bounded control signal, let us modify (4.47) by introducing the C-S model and generalize it to a MAS of N agents as follows:

$$\dot{x}_i = \alpha H \sum_{j \in \mathcal{N}_i} \frac{\left[\frac{TH_{ij}}{(x_j - x_i)^2} - 1 \right]}{\left[\gamma (x_j - x_i)^2 + \delta \right]^\beta} \{x_i - x_j\} \quad (4.47.2)$$

where: $\gamma, H, \delta, \beta > 0$. According to (4.47.2), Figure 4.32 shows the response of a single-edge MAS system and the corresponding control signal when $TH_{ij} = \{1, 0, 0.6\}$ varied according to a multistep function filtered through a suitable low-pass filter and $\alpha = H = 1$, $\gamma = 0.01$, $\beta = 0.5$ and $\delta = 1000$.

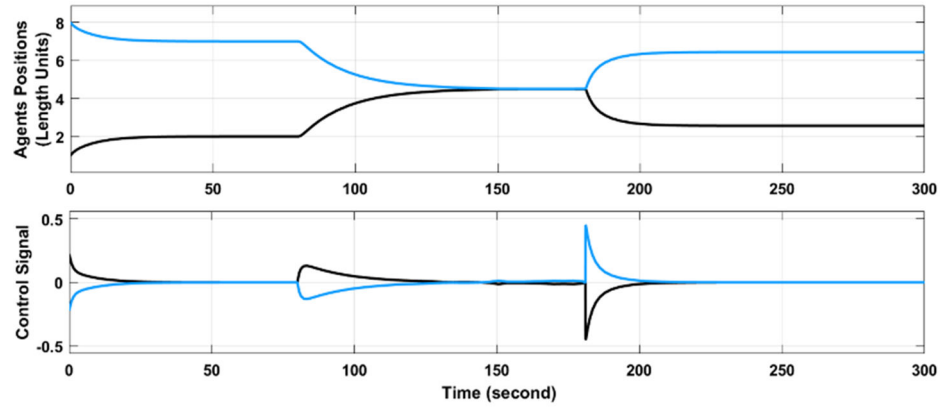


Figure 4.32. Single-edge response due to a multistep input under (4.52.2).

Examining the structure of (4.47.2), we may rewrite it in the following form:

$$\dot{x}_i = \alpha \sum_{j \in \mathcal{N}_i} r_{ij}(x_i, x_j) \{x_i - x_j\} - H \frac{\{x_i - x_j\}}{[\gamma(x_j - x_i)^2 + \delta]^\beta} \quad (4.47.3)$$

where:

$$r_{ij}(x_i, x_j) = H \frac{\left[\frac{TH_{ij}}{(x_j - x_i)^2} \right]}{[\gamma(x_j - x_i)^2 + \delta]^\beta} \quad (4.47.4)$$

Clearly, the numerator of $r_{ij}(x_i, x_j)$ is related to a repulsive potential field, i.e., a source, whose strength is localized using the parameters involved as will be explained shortly. When $TH_{ij} \neq 0$, it is expected from the edge $\{i, j\} \in \mathcal{E}$ - along with its two connected agents- to undergo a deployment or formation behavior. This behavior, however, cannot

be realized except about the average consensus value of agents' initial conditions. Now, it is possible to rewrite (4.47.3) as follows:

$$\dot{x}_i = \alpha \sum_{j \in \mathcal{N}_i} r_{ij}(x_i, x_j) \{x_i - x_j\} - \hat{g}_{ij}(x_i, x_j) \{x_i - x_j\} \quad (4.47.5)$$

where: \hat{g}_{ij} is given by the parameterized C-S model given in (4.34). Then, the overall dynamics of the MAS can be given as follows:

$$\vec{\dot{x}}(t) = \alpha \{R_r(\vec{x}) - \mathcal{L}(\vec{x})\} \vec{x}(t) \quad (4.47.6)$$

where: $R_r(\vec{x}) = [r_{ij}(x_i, x_j)]$ and has the same structure as the state-dependent Laplacian matrix $\mathcal{L}(\vec{x})$. As a result, comes the following corollary.

Corollary 4.2: A MAS, which acts over an initially connected network graph, under the universal protocol (4.47.2) with $\beta = 0.5$ is connectivity-preserving $\forall \vec{x}$ in the domain of interest.

Proof: Following the same discussion presented in proofing **Theorem 4.10**, and after doing the Math, we end up having the following condition $\forall i, j, \vec{x}$:

$$(\gamma TH_{ij} + \delta + 2\gamma\beta TH_{ij})e^2 + \gamma(1 - 2\beta)e^4 + \delta TH_{ij} \neq 0 \quad (4.47.7)$$

where: $e = x_j - x_i$. Taking $\beta = 0.5$ results in a positive quantity for all state-independent parameterization of γ, δ and TH_{ij} and therefore all edges remain connected.

■

At this point, one cannot resist the temptation of introducing the $g(x_i)$ function; so that other types of consensus can be used instead of the average consensus. In this way, we can handle a more sophisticated situation of agents' deployment with less mathematical burden compared to (4.45), in which parameters tuning to maintain edge connectivity seems challenging when $g(x_i) \neq 1$. In such a case, agents are deployed in the working space

while always maintaining the desired type of mean. The resulting MAS dynamics is given as follows:

$$\vec{x}(t) = \alpha G(\vec{x}) \{R_r(\vec{x}) - \mathcal{L}(\vec{x})\} \vec{x}(t) \quad (4.47.8)$$

where: $G(\vec{x})$ is given by (4.9). Other Euclidean directions follow the same steps. This gives rise to the following corollary.

Corollary 4.3: A MAS, which acts over an initially connected network graph, whose dynamics are given by (4.47.8) with $\beta = 0.5$ is connectivity-preserving $\forall \vec{x}$ in the domain of interest while achieving a collision-free average, geometric, harmonic and mean-of-order- p deployment, formation and consensus in the working space with appropriate choice of TH_{ij} values.

Proof: Following the same discussion presented in proofing **Theorem 4.10**, **Corollary 4.2** and (3.42), it is straight forward to see that **Corollary 4.3** is true. Note also that both $R_r(\vec{x})$ and $\mathcal{L}(\vec{x})$ have the $\vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$, therefore all the needed conditions to proof the connectedness of the MAS and achieving the desired type of mean are satisfied if TH_{ij} values are adequately chosen as discussed before.

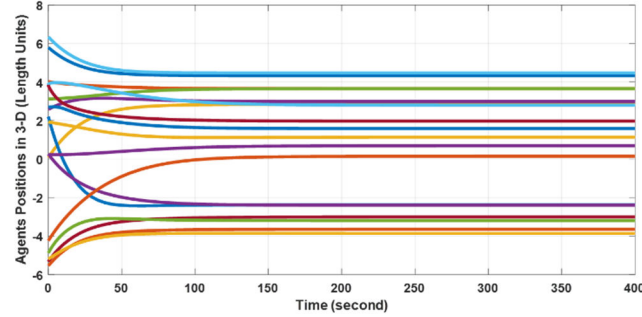
■

The proof of agents' consensus under (4.47.8) is straightforward, and the similarity between (4.47.8) and the formation protocol (4.21.5) is obvious. The following figures show the results obtained when using (4.47.8) to achieve average, geometric and harmonic deployment of six agents in the 3-D space using the network graph shown in Figure 3.13. Observe the difference between agents' responses under (4.34) and (4.47.8) while achieving the average deployment task. This can be read from figures 4.33.b and 4.33.c.

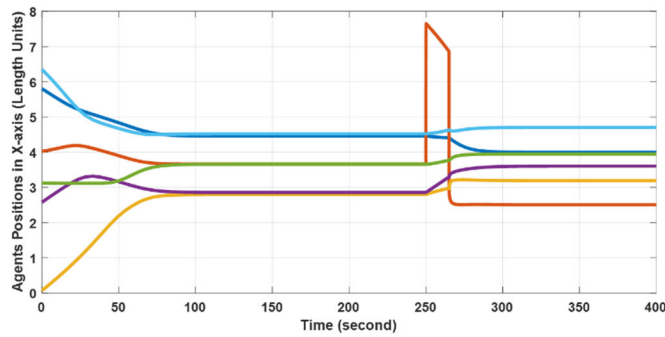
To address the formation control in 3-D using (4.47.8), it is best to include all the Euclidean directions in $R_r(\vec{x})$, i.e., to have $R_r(\vec{x}, \vec{y}, \vec{z})$ instead without affecting the C-S part of r_{ij} . This will inherit the easiness and geometric flexibility offered by (4.16). Therefore, the overall MAS dynamics in 3-D under the modified version of (4.47.8) is given as follows:

$$\begin{aligned}\vec{x}(t) &= \alpha_1 G_1(\vec{x}) \{R_{rx}(\vec{x}, \vec{y}, \vec{z}) - \mathcal{L}(\vec{x})\} \vec{x}(t) \\ \vec{y}(t) &= \alpha_2 G_2(\vec{y}) \{R_{ry}(\vec{x}, \vec{y}, \vec{z}) - \mathcal{L}(\vec{y})\} \vec{y}(t) \\ \vec{z}(t) &= \alpha_3 G_3(\vec{z}) \{R_{rz}(\vec{x}, \vec{y}, \vec{z}) - \mathcal{L}(\vec{z})\} \vec{z}(t)\end{aligned}\tag{4.48}$$

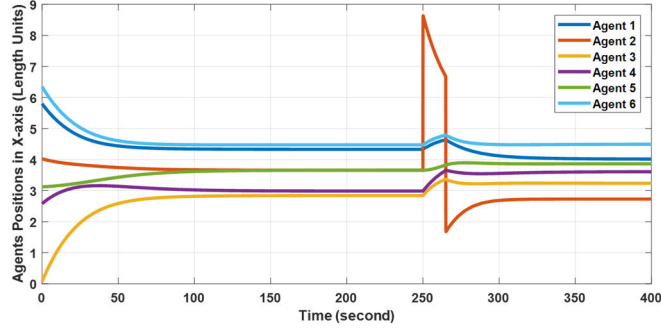
where: G_i is given in (4.9) and therefore could be different. Figure 4.43.a shows the simulation results obtained during formation control of six agents using (4.48) with $G_i(\vec{x}) = I$, $\alpha_i = 10$, $i = 1, 2, 3$ and complete consensus along the z-axis, i.e., $R_{rz}(\vec{x}, \vec{y}, \vec{z}) = [0]$.



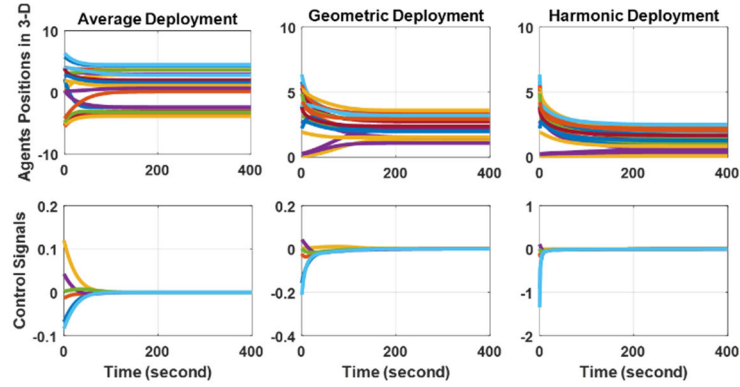
(a)



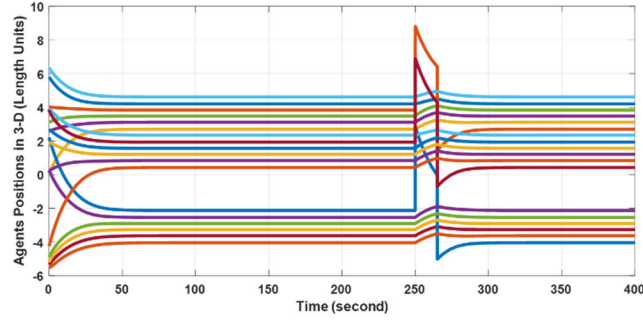
(b)



(c)



(d)



(e)

Figure 4.33. Simulation results of agents' deployment in 3-D using the universal protocol (4.47.8). (a) agents' positions under average deployment. (b) average deployment using (4.34) with disturbance enables- only agents positions along the x-axis are shown. (c) another experiment of average deployment using (4.47.8) with disturbance enables- only agents positions along the x-axis are shown. (d) the three types of mean deployment shown along with the needed control signals with 20 times increase in the convergence rate for the last two types. (e) average deployment of the same agents using (4.47.8) over a complete graph with identical TH_{ij} values.

The communication network used matched the shape graph and both were complete undirected graphs. The intended separation between two successive agents was 2 length units; however, there was some scaling as expected, in general. Figure 4.34.b shows the minimum distance among agents during the whole formation sequence and it is evident that there was no collision among agents while navigating the 3-D working space. In addition, if agents have finite sizes, then it can be reflected directly into $R_r(\vec{x}, \vec{y}, \vec{z})$ as follows:

$$r_{ij}(x_i, x_j, \Delta_{ij}^x) = H \frac{\left[\frac{TH_{ij}}{(x_j - x_i - \Delta_{ij}^x)^2} \right]}{\left[\gamma(x_j - x_i - \Delta_{ij}^x)^2 + \delta \right]^\beta} \quad (4.48.1)$$

and the C-S part is modified in the same way. Therefore, the overall control structure becomes:

$$\dot{x}_i = \alpha \sum_{j \in \mathcal{N}_i} \{r_{ij}(x_i, x_j, \Delta_{ij}^x) - \hat{g}_{ij}(x_i, x_j, \Delta_{ij}^x)\} \{x_i - x_j - \Delta_{ij}^x\} \quad (4.48.2)$$

Interestingly, after accommodating the physical dimensions of agents into (4.47.3), its similarity with the second formation protocol, i.e., (4.22), becomes obvious. Despite this similarity, in (4.48.2), $\Delta_{ij}^x = -\Delta_{ji}^x$ denotes the physical dimension rather than the desired formation distance between the two agents.

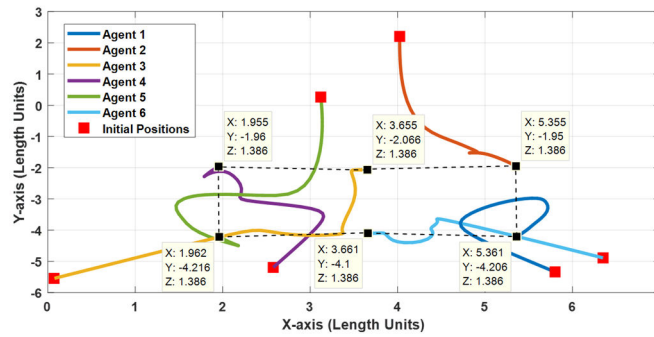
Accidentally, while simulating (4.48) we obtained orbiting-like behaviors of agents while keeping the formation shown in Figure 4.34.a. It turned out that changing a specific undirected edge weights, see Figure 3.9, or making it directed is the cause behind such behaviors over a general network graph. Figure 4.35 shows some of these results with a brief explanation provided in a coming subsection. Examining such behaviors in detail will

be a subject of a future investigation. Note, however, that this approach is different from the one which depends on the conservation of angular momentum.

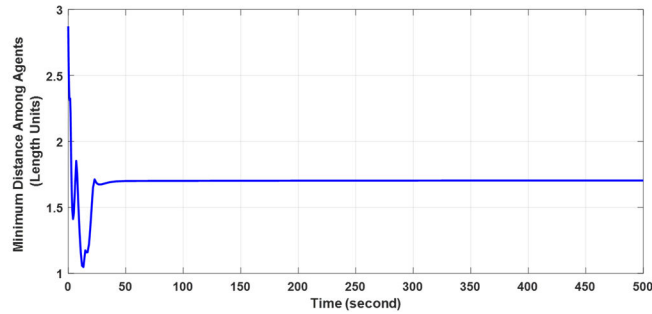
The ability to change TH_{ij} at the edge level makes the formation protocol (4.21.5) a special case of (4.53). The conditions needed to parameterize TH_{ij} will be the subject of the following subsection.

4.1.4.2: State-dependent parameterization at edge-level using the universal behavioral protocol

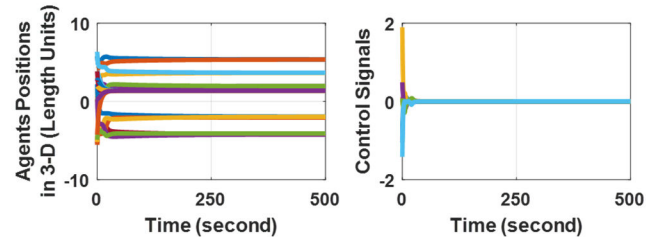
Working at the edge-level gives more in-depth understanding about the various ways a MAS behavior could be controlled.



(a)

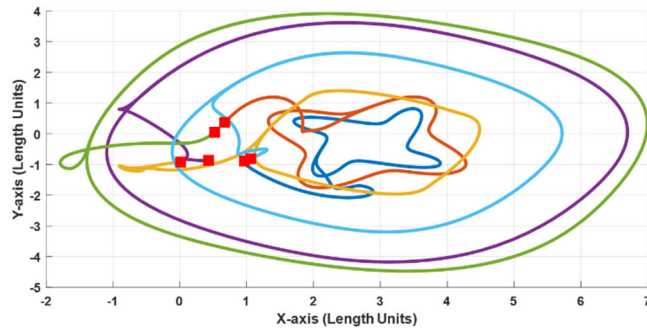


(b)

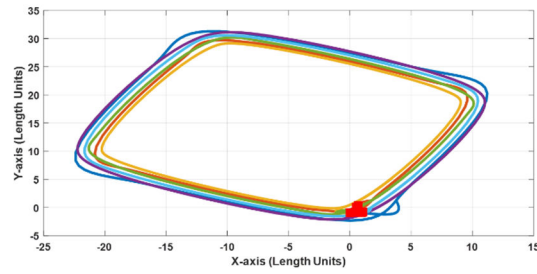


(c)

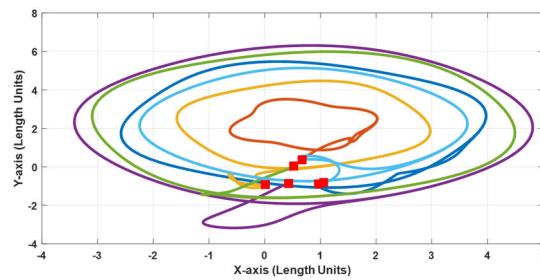
Figure 4.34. Simulation results of agents' formation in 3-D using the universal protocol (4.47.8). (a) trajectories. (b) 3-D minimum distance among agents during the whole course of formation. (c) agents' positions in 3-D and the needed control signals.



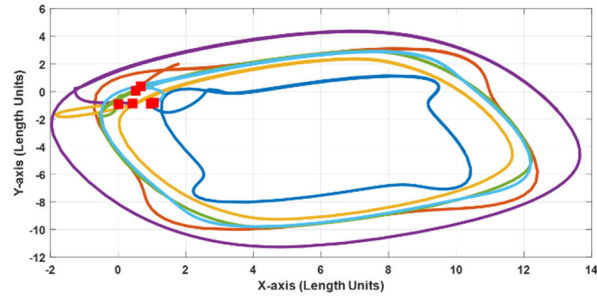
(a)



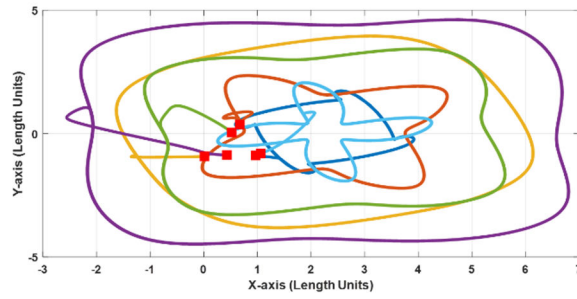
(b)



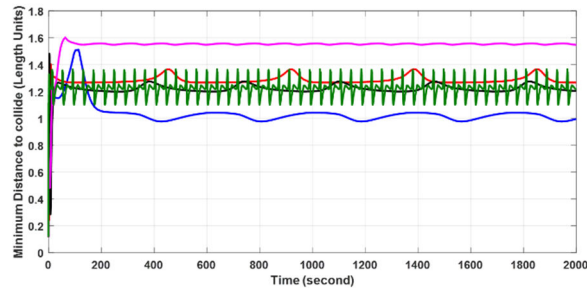
(c)



(d)



(e)



(f)

Figure 4.35. Simulation results of agents' formation in 2-D using the universal protocol (4.48) over a general network while changing the undirected weights, i.e., unbalanced, or making an edge directed. (a)-(e) agents' positions. (f) minimum distance to collide.

For example, introducing the partial, complete and global neighborhoods enables controlling the behavior of a MAS from a holistic view, so that individual agents' contribution is not much of a concern from the designer point of view whose presumption of full-submission of agents is of no doubt. Having this in mind, we study the individual

agents' participation in achieving the overall behavior and also enable them with self-awareness- up to an extent- so that they consciously participate with other agents.

Using (4.48), the overall MAS dynamics in the x-direction can be given as follows where state-dependent parameterization at edge-level is introduced:

$$\dot{\vec{x}}(t) = \alpha_1 G_1(\vec{x}) \{R_{rx}(\vec{x}, \vec{y}, \vec{z}, \vec{s}) - \mathcal{L}(\vec{x})\} \vec{x}(t) \quad (4.49)$$

where: \vec{s} is a vector whose elements are all possible values of TH_{ij} . Similarly, the MAS dynamics in the y and z directions follow.

Note 4.17: When $TH_{ij}(s_{ij}(\vec{x}, \vec{y}, \vec{z})) = TH(s(\vec{x}, \vec{y}, \vec{z}))$, then the whole MAS is affected by one state-dependent parameterization. This is exactly what happens in complete, global and- to an extent- partial neighborhoods.

Since (4.49) is a semi-linear protocol, we may use the conditions stated in Part I, i.e., (3.34) and (3.35), so that we design the structure of $TH_{ij}(s_{ij}(\vec{x}, \vec{y}, \vec{z}))$. Working mainly with the x-direction, we have the following lemma:

Lemma 4.4: The MAS given in (4.49) with $G_1(\vec{x}) = I$ will achieve the formation-deployment-consensus about a point in the working space corresponding to the arithmetic mean of agents' initial values if and only if the initial graph is connected and the following is true:

$$TH_{ij} > 0 \quad (4.49.1)$$

$$\frac{\partial TH_{ij}}{\partial q_i} = -\frac{\partial TH_{ij}}{\partial q_j} \quad (4.49.2)$$

$$\frac{\partial TH_{ij}}{\partial q_i} = \frac{\partial TH_{ji}}{\partial q_i} \quad (4.49.3)$$

Moreover, it will be always connected if and only if for $\beta = 0.5$ the following is true:

$$\frac{\partial TH_{ij}}{\partial q_i}(q_j - q_i) \geq 0 \quad (4.49.4)$$

where: $q_i = \{x_i, y_i, z_i\}$.

Proof: Recalling (3.34) and (3.35), it is straight forward to show, by direct differentiation of (4.47.4), that the needed conditions to satisfy both (3.34) and (3.35)- using r_{ij} - is to have (4.49.1) and (4.49.2) true. Note that the other term in (4.49), i.e., the C-S model, already satisfies (3.34) and (3.35).

In addition to (4.47.7), the additional term that ensures the connectivity of $\lambda_2^{\{i,j\}}(\vec{x}) \forall \vec{x}$ is given by (4.49.4). This is shown as follows:

Given g_{ij} as follows:

$$g_{ij} = \frac{\left[\frac{TH_{ij}(x_i, x_j)}{(x_j - x_i)^2} - 1 \right]}{\left[\gamma(x_j - x_i)^2 + \delta \right]^\beta} \quad (4.49.5)$$

Then, it is easy to find $\partial g_{ij} / \partial x_i$ which is then used in (4.38). So, we have:

$$\begin{aligned} \lambda_2^{\{i,j\}}(\vec{x}) &= \left\{ \frac{\partial g_{ij}}{\partial x_i} \{x_j - x_i\} - g_{ij} \right\} (v_i - v_j)^2 \\ &= \frac{(v_i - v_j)^2 \left\{ \frac{\partial TH_{ij}}{\partial x_i} [\gamma(x_j - x_i)^3 + \delta(x_j - x_i)] + m_4 \right\}}{(x_j - x_i)^2 [\gamma(x_j - x_i)^2 + \delta]^{\beta+1}} \end{aligned} \quad (4.49.6)$$

where: $m_4 = (\gamma TH_{ij} + \delta + 2\gamma\beta TH_{ij})e^2 + \gamma(1 - 2\beta)e^4 + \delta TH_{ij}$. This equation can be made positive $\forall \vec{x}$ and all parameterization if $(v_i - v_j)^2 \neq 0 \forall i, j$ indicating that the initial graph is connected, and by taking $0 < \beta \leq 0.5$ - here we prefer $\beta = 0.5$, and by making sure that (4.49.4) is satisfied.

■

4.1.4.3: Designing connectivity-preserving protocols

In previous subsections, we mainly were interested in examining the nature of a given protocol whether it is connectivity-preserving or not. In this subsection, we will continue from that point on and show how a connectivity-preserving protocol can be designed and localized such that it helps other behaviors to complete their tasks with minimum intervening once agents are connected. The idea behind this approach is to have a strategy that ensures the connectivity of agents under various behaviors. Once a behavior is acting against the well-being of connectedness, then the connectivity-preserving protocol will take over assuming behaviors with known bounded control signals irrespective of what the actual behavior is. It is at this point that the reason why being interested in behaviors with bounded control signals becomes more obvious.

In Chapter 3, the concept of prescribed connectivity was presented and displayed to be always persistent and therefore will be only useful in situations where complete or partial consensus is required. However, that concept is indeed a special case of the connectivity preservation concept introduced here.

In [145], a Lyapunov function was used to design a connectivity-preserving flocking behavior of a group of mobile robots assuming that $\lambda_2(\vec{x})$ is a nondecreasing function of each edge weight. In this subsection, we mainly depend on **Corollary 4.1** and **Theorem 4.10** to design a semi-linear control signal that utilizes the gradient of the $\lambda_2(\vec{x})$ in a general yet systematic fashion once compared to [145]. Nonlinear connectivity-preserving protocols will be a subject of future work.

Starting with any g_{ij} that- preferably- yields a connectivity-preserving protocol, $\lambda_2^{\{i,j\}}(\vec{x})$ can be found using (4.41) or (4.42). In order to maximize the edge connectivity, we

suggest- as also done in [145]- to move along the gradient of $\lambda_2^{\{i,j\}}(\vec{x})$. Being connected to several edges, an agent should move in a way that satisfies all its neighbors, and therefore the agent should move towards the instantaneous center of its neighborhood. This can be clearly seen from Figure 4.36.

Among the various possibilities existing, let us use mainly (4.41) to design our connectivity-preserving protocols. Therefore, one component of the gradient connectivity of an edge can be given as follows:

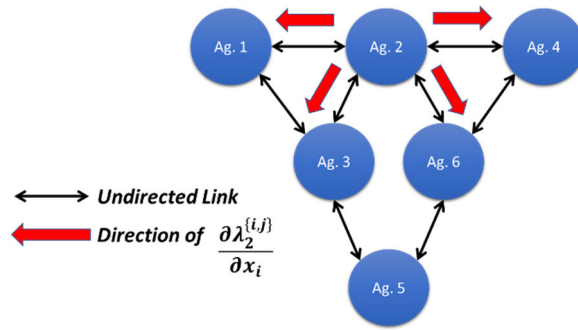


Figure 4.36. A communication graph showing the links and edges connectivity gradients from the perspective of agent 2.

$$\frac{\partial \lambda_2^{\{i,j\}}}{\partial x_i} = \left\{ \frac{\partial^2 g_{ij}}{\partial x_i^2} \{x_j - x_i\} - 2 \frac{\partial g_{ij}}{\partial x_i} \right\} (v_i - v_j)_2^2 \quad (4.50)$$

The other component of gradient can be found similarly.

Having (4.50) as a template, various semi-linear based connectivity-preserving protocols can be designed. Recalling specifically the C-S model and the result obtained earlier in (4.39.2) with $\beta = 0.5$, we may design our desired protocol as follows:

$$\frac{\partial \lambda_2^{\{i,j\}}}{\partial x_i} = \frac{3\gamma H \alpha \delta (x_i - x_j)}{[\gamma (x_j - x_i)^2 + \delta]^{5/2}} (v_i - v_j)_2^2 = h_{ij}(x_i, x_j) \{x_i - x_j\} \quad (4.50.1)$$

which has its maxima $e^* = (x_1^* - x_2^*)$ at:

$$e^* = \sqrt{\frac{\delta}{4\gamma}} \quad (4.50.2)$$

The peak value of (4.55.1) at this point is given as follows:

$$P(e^*) = 3\alpha H (v_i - v_j)_2^2 \sqrt{\frac{4^4\gamma}{5^5\delta^2}} \quad (4.50.3)$$

It is desired to bound (4.50.1) such that its value is always less or equal to 1. So, by normalizing it with respect to the convergence rate used and the elements of the eigenvector involved, i.e., $\alpha(v_i - v_j)_2^2$, we have:

$$H = \frac{1}{3} \sqrt{\frac{5^5\delta^2}{4^4\gamma}} \quad (4.50.4)$$

At this point, it is the responsibility of the design engineer to decide the values of e^* and to build as many as needed of these connectivity-preserving protocols to cover the expected range of agents' spatial separations. Figure 4.37 shows three of such protocols along with a consensus protocol realized using the C-S model with $\beta = 2$ with a convergence rate $\alpha = 100$ and the contribution of eigenvector elements was ignored.

Figure 4.37 recalls the radial basis functions (RBF) used in neural networks. In fact, the radial basis function has been used in connectivity preservation as can be found in [146], for example. However, the RBF are centered bell-shaped curved and not like those in Figure 4.37.

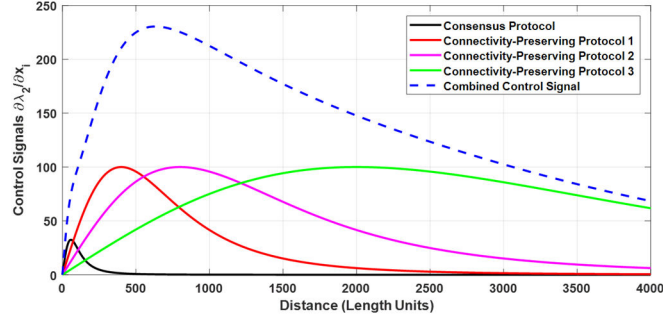


Figure 4.37. The control signals under multi connectivity-preserving protocols running in parallel with another consensus protocol realized using the C-S model with $\beta = 2$ with a convergence rate $\alpha = 100$ and the contribution of eigenvector elements was ignored.

Figure 4.38 shows the simulation results of the MAS depicted in Figure 4.36 with and without the connectivity-preserving protocols. The maximum separation was between agents 1 and 3 along the x-axis and was equal to 955 length units, $e^* = \{400, 800, 2000\}$ and $\delta = 64$ for all bases functions of connectivity-preserving protocols.

Note 4.18: The multi connectivity-preserving protocols are running in parallel at agent level.

Clearly, the following is true:

$$\frac{\partial \lambda_2^{\{i,j\}}}{\partial x_i} = -\frac{\partial \lambda_2^{\{i,j\}}}{\partial x_j} \quad (4.50.5)$$

The overall dynamics of the MAS shown in Figure 4.36 is given as follows- in the x-direction:

$$\vec{x}(t) = -\alpha \{C_\lambda^T(\vec{x})\vec{1} + \mathcal{L}(\vec{x})\vec{x}(t)\} \quad (4.50.6)$$

where:

$$C_{\lambda}^T(\vec{x}) = \begin{bmatrix} 0 & \frac{\partial \lambda_2^{\{1,2\}}}{x_2} & \dots & \frac{\partial \lambda_2^{\{1,N\}}}{\partial x_N} \\ \frac{\partial \lambda_2^{\{1,2\}}}{\partial x_1} & 0 & \dots & \frac{\partial \lambda_2^{\{2,N\}}}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \lambda_2^{\{1,N\}}}{\partial x_1} & \frac{\partial \lambda_2^{\{2,N\}}}{\partial x_2} & \dots & 0 \end{bmatrix} \quad (4.50.7)$$

Other Euclidean directions follow in a similar way.

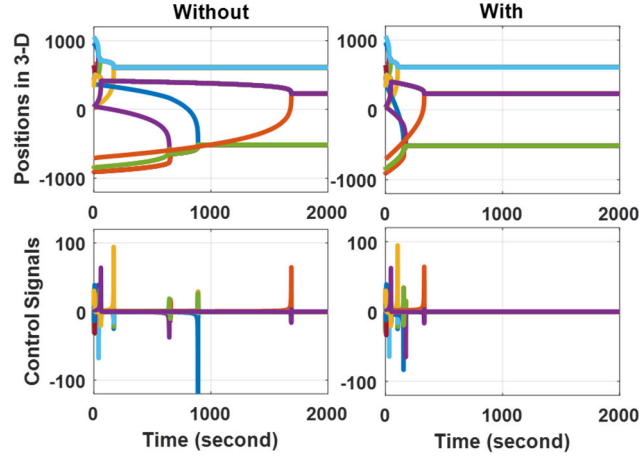


Figure 4.38. The simulation results of the MAS shown in Figure 4.36 with and without the connectivity-preserving protocols while achieving the average consensus. The maximum separation was between agents 1 and 3 along the x-axis and was equal to 955 length units.

Note 4.19: When there is no edge between the i^{th} and the j^{th} agents, then $\partial \lambda_2^{\{i,j\}} / \partial x_i = 0$.

Note also that $C_{\lambda}^T = -C_{\lambda}$.

Clearly, in (4.50.6), the average consensus will be achieved. By introducing the $G_1(\vec{x})$ matrix- as we have done in (4.49) for example- we may achieve various types of consensus.

Therefore, (4.50.6) can be written as follows:

$$\vec{x}(t) = -\alpha G_1(\vec{x}) \{C_{\lambda}(\vec{x}) \vec{1} + \mathcal{L}(\vec{x}) \vec{x}(t)\} \quad (4.51)$$

where $G_1(\vec{x})$ is given in (4.9). This is stated by the following lemma.

Lemma 4.5: The 3-D MAS under (4.51) will stay connected, maintain and converge to, average, geometric, harmonic and the general mean-of-order- p at consensus if at least one of the following is true:

- 1- The g_{ij} functions used in the consensus protocol are originally connectivity-preserving and achieve (3.34) and the initial network graph is connected.
- 2- The maximum initial separations between agents is within the connectivity-preserving protocols range of action.

Proof: If $\mathcal{L}(\vec{x})$ is constructed from connectivity-preserving functions, then all its edges will remain connected if they were in the first place. Moreover, if the g_{ij} functions used achieve (3.34), then $\vec{1}^T$ will be associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ and therefore the desired type of consensus selected by $G_1(\vec{x})$ is achieved, see (3.42).

If $\mathcal{L}(\vec{x})$ is constructed from non-connectivity-preserving functions, and single or multi connectivity-preserving protocols are used, then the MAS will be acted upon by these functions if and only if the agents maximum separation is within their range of actions.

■

In Chapter 3, we have presented the concept of prescribed connectivity which turned to be both global and persistent based on the current argument. Moreover, that concept of prescribed connectivity is a special case of the connectivity-preserving concept presented here. This is shown by the following lemma.

Lemma 4.6: The protocol with prescribed connectivity presented in Chapter 3 is a special type of connectivity-preserving protocols.

Proof: Let us have the following g_{ij} function:

$$g_{ij}(x_i, x_j) = H(x_j - x_i)^n \quad (4.52)$$

where: $n > 0$. Now, it should be easy to show that:

$$\frac{\partial \lambda_2^{\{i,j\}}}{\partial x_i} = n(n+1)H(v_i - v_j)_2^2 (x_j - x_i)^{n-1} \quad (4.52.1)$$

indicating that (4.52) is a connectivity-preserving function when the edge is connected and $x_j \neq x_i$. Taking $n = 2$, yields:

$$\frac{\partial \lambda_2^{\{i,j\}}}{\partial x_i} = 6H(v_i - v_j)_2^2 (x_j - x_i) \quad (4.52.2)$$

Writing (4.52.2) $\forall \{i, j\} \in \mathcal{E}$ edges and evaluating $-C_\lambda^T(\vec{x})\vec{1} = C_\lambda(\vec{x})\vec{1}$, we obtain the following:

$$u_i^{\lambda_2} = \sum_{j \in \mathcal{N}_i} \frac{\partial \lambda_2^{\{i,j\}}}{\partial x_i} = \sigma \sum_{j \in \mathcal{N}_i} h_{ij} \{x_j - x_i\} \quad (4.52.3)$$

where: $\sigma = 6H$, $h_{ij} = (v_i - v_j)_2^2$ and $u_i^{\lambda_2}$ is i^{th} agent connectivity-preserving control signal. Note that (4.52.3) is like (4.2). Following the steps taken in (4.3) to (4.5), we may write the global connectivity-preserving control signal as follows:

$$\vec{u}^{\lambda_2} = -\sigma \tilde{\mathcal{L}}_f \vec{x} \quad (4.52.4)$$

In Chapter 3, we have taken $\tilde{\mathcal{L}}_f$, i.e., the fixed Laplacian matrix with weighted edges, as \mathcal{L}_f , i.e., the fixed Laplacian matrix with $\{0,1\}$ weights for simplicity. So, (4.51) can be rewritten as follows when $G_1(\vec{x}) = I$:

$$\vec{\dot{x}}(t) = -\alpha \{\sigma \tilde{\mathcal{L}}_f + \mathcal{L}(\vec{x})\} \vec{x}(t) \quad (4.52.5)$$

which agrees with (3.80). ■

Remark 4.9: Note that not every connectivity-preserving protocol can be written in a semi-linear form, i.e., as $H(\vec{x})\vec{x}$, unless it has $\pm(x_j - x_i)$ as a factor. This is stated by the following lemma.

Lemma 4.7: A connectivity-preserving protocol that maintains the time constant of the MAS, i.e., its average value for example, can be written in a semi-linear form $\vec{u}^{\lambda_2} = -H(\vec{x})\vec{x}$ if and only if it has $\pm(x_j - x_i)$ as a factor and the resulting $H(\vec{x}) = [h_{ij}(x_i, x_j)]$ matrix has $\vec{1}$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$.

Proof: Following the same steps shown in (4.52) to (4.52.4), we will end up with the following:

$$u_i^{\lambda_2} = \sum_{j \in \mathcal{N}_i} \frac{\partial \lambda_2^{\{i,j\}}}{\partial x_i} = \sigma \sum_{j \in \mathcal{N}_i} h_{ij}(x_i, x_j) \{x_j - x_i\} \quad (4.53)$$

Which after simplification can be written as:

$$\vec{u}^{\lambda_2} = -\sigma H(\vec{x})\vec{x} \quad (4.53.1)$$

which maintains the MAS invariance if and only if it has $\vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$. This of course can be checked using (3.34).

■

According to **Lemma 4.7**, we may put (4.50.1) into a semi-linear form.

Remark 4.10: Having a desired evolution of an edge connectivity, one may find the corresponding needed weighting functions, i.e., g_{ij} , by solving (4.50) if it is well-posed. It is true that general functions that are not related to the diffusive-coupling g_{ij} functions can be used; however, may lose the clear picture of the relation between the MAS connectivity and the functions used, or at best the picture will be too complicated to grasp. Using non-diffusive coupling under the proposed framework is a subject of future work.

4.1.4.4: The universal protocol and the Solar system

Looking carefully at the universal protocol in 3-D, it is possible to recognize its similarity to the differential equation that models the motion of the planets in our Solar system.

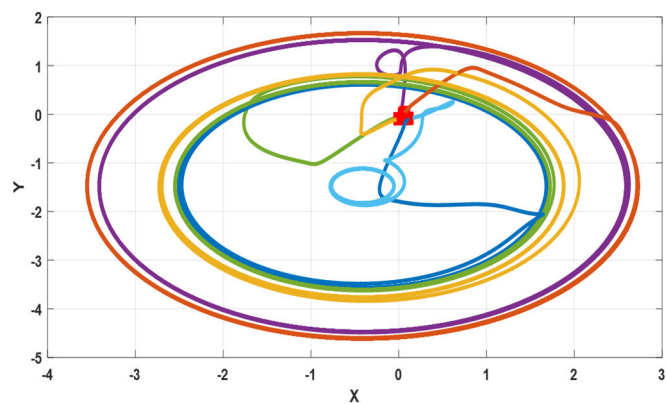
Taking $\beta = 0.5$, yields:

$$g_{ij} = H \frac{\left[\frac{TH_{ij}}{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} - 1 \right]}{\left[\gamma(q_j - q_i)^2 + \delta \right]^{0.5}} \quad (4.54)$$

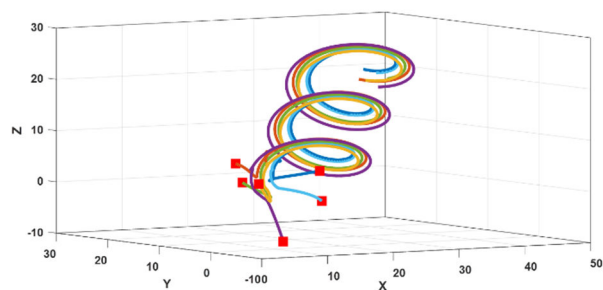
where: $q_i = \{x_i, y_i, z_i\}$. Describing the motion along the x-axis, let $\delta = (y_j - y_i)^2 + (z_j - z_i)^2$ and $\gamma = 1$. Compare this to the following law of motion of planets [147] when $\beta = 0.5$:

$$m_i \frac{d^2 \vec{r}_i}{dt^2} = \bar{G} m_i \sum_{j \neq i} m_j \frac{\vec{r}_j - \vec{r}_i}{\|\vec{r}_j - \vec{r}_i\|^3} \quad (4.54.1)$$

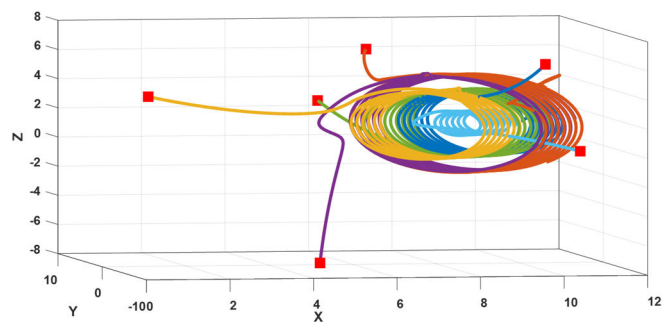
where: \bar{G} is the universal constant of gravitation, the masses are denoted by $m_{i,j}$ and \vec{r}_i is the position vector in 3-D space. It is straightforward to see the similarity; however, in (4.54) there is an additional term responsible for achieving consensus among the agents. The following figures were generated using a mixed multigraph- whose structure and relation to physics are yet to be understood- with $\delta = 1$, $\gamma = 10^{-4}$ and $H = 10^{-2}$ and the formation shown in Figure 4.34.a.



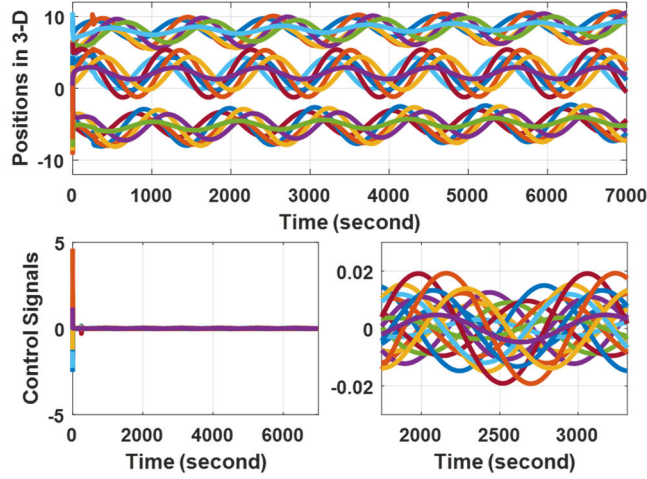
(a)



(b)



(c)



(d)

Figure 4.39. Simulation results of agents formation in 3-D using the universal protocol (4.54) over a general network while changing the undirected weights, i.e., unbalanced, or making an edge directed. (a) -(c) agents positions. (d) agents positions versus time and the needed control signal for (c).

4.1.5: Shape consensus and clustering behaviors

Under various protocols of formation and deployment, additional equilibria sets are usually introduced to the MAS dynamics to facilitate its behavior. In this section, the original equilibria set will be modified such that agents at steady-state will be partitioned into clusters in which agents will reach consensus. That is why we refer to clustering as partial consensus in this work.

4.1.5.1: The clustering behavior

Partial consensus can be looked at as a generalization of the complete consensus among agents. The main difference appears in the way weights are assigned when building the consensus or clustering protocols. In (4.6), the d_{ij} weights determine the relation between the i^{th} and j^{th} agents; such that when $d_{ij} > 0$ then the two agents will cooperate and when $d_{ij} < 0$ then the two agents will compete. It is when $d_{ij} = 1 \forall i, j$, then the whole MAS

will reach the complete consensus, or consensus for short. Therefore, we may define d_{ij} in the x-direction as follows:

$$d_{ij}^x \equiv d_{ij}^x(x_i, x_j): \mathbb{R}^2 \rightarrow \mathbb{R}/\{0\} \quad (4.55)$$

Examining (4.6), the equilibria given in (3.31) is affected by the value of d_{ij}^x as follows:

$$0 = \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{d_{ij}^x x_j - x_i\}, \forall i = 1, 2, \dots, N \quad (4.55.1)$$

Knowing that $g_{ij} > 0$ - see *Remark* PI-8, satisfying at least (3.35) and by assuming that $x_i \neq 0$, (4.55.1) can be rewritten as follows:

$$0 = \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{d_{ij}^x x_j/x_i - 1\}, \forall i = 1, 2, \dots, N \quad (4.55.2)$$

from which it is obvious that the ratio $x_j/x_i = 1/d_{ij}^x$ which- unlike the complete consensus case- might be different $\forall i, j$. This ratio governs the elements of the equilibria set. In other words, the additional weights, i.e., d_{ij}^x , steer the line of interaction between the i^{th} and j^{th} agents. At this point, we may equip the agents with the ability to autonomously decide for themselves about the way they should interact with their neighbors based on some criterion. So, for the i^{th} agent, we may write the following:

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{d_{ij}^x(x_i, x_j) x_j - x_i\} \quad (4.55.3)$$

where: $d_{ij}^x(x_i, x_j) \in \mathbb{R}/\{0\}$ denotes the criterion based on which the i^{th} agent decides the level of cooperation or competition with the j^{th} agent. Note that (4.55.3) is closely related to the general class given by (3.72) and (3.75).

Let us consider the case shown in Figure 4.40 in which an operator has created two clusters- may be with different types of assets in each- and assigned a virtual leader / target in the working space. Agent 1 is pinned to that virtual leader.

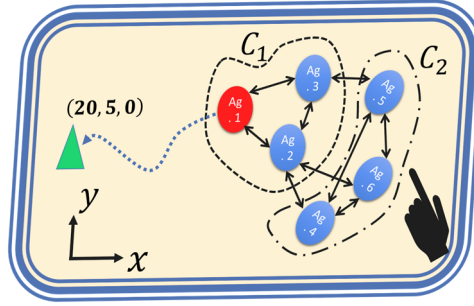


Figure 4.40. Two clusters C_1 and C_2 with agent 1 pinned to the virtual leader / target point (triangle) selected by an operator.

In such a scenario, we are faced with three tasks that the MAS must be able to simultaneously conduct, namely: clustering, leader following and maintaining a sufficient separation between agents in the same cluster to avoid collision. Not as easy as it may seem, once a cluster starts moving, its size starts changing. This effect can be read from the equilibria element that governs the i^{th} and j^{th} agents, i.e., $1/d_{ij}$, along which the movement is not constrained, see Figure 4.41.e. Fortunately, (4.55.3) can be modified such that the previous tasks can be satisfied at once. This can be done as follows using *clustering size control*- written in a short notation:

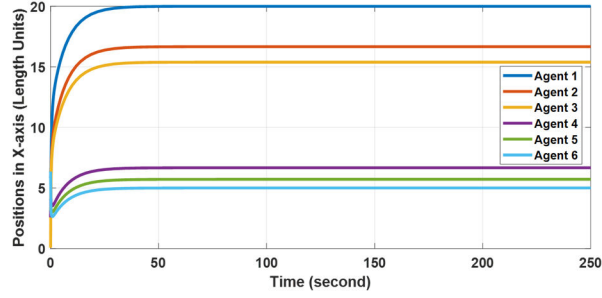
$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} g_{ij} \{d_{ij}^x x_j - x_i\} + p_i \{x_l - x_i\} \quad (4.55.4)$$

where: $p_i \geq 0$ is the pinning gain and x_l is the leader state, and d_{ij}^x is given as follows:

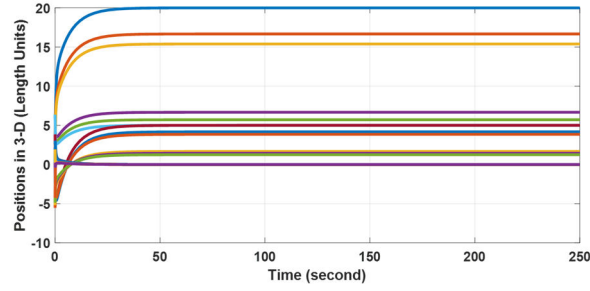
$$d_{ij}^q = C_{ij}^w(d_{ij,f}^q, q_i, q_{j \in \mathcal{N}_i}) = \frac{1}{2} d_{ij,f}^q (1 + \tanh(\arg)) + \frac{1}{2} (1 - \tanh(\arg)) \quad (4.55.5)$$

where: C_{ij}^w denotes a general scalar function, real or complex, q_i could be one or more of $\{x_i, y_i, z_i\}$, $d_{ij,f}^x$ is the fixed d_{ij}^x value given when the cluster is designed, the $\arg =$

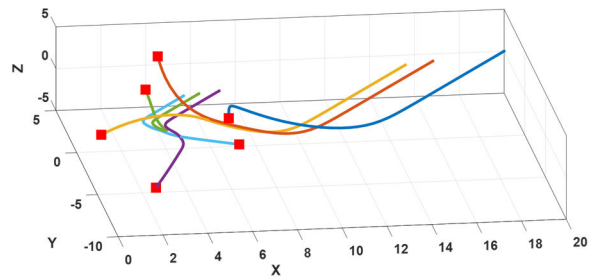
$k \left(TH - \|r_{ij}\|^2 \right)$ with $k \gg 1, TH > 0$ and, in general, $\|r_{ij}\|^2 = (x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2$. TH denotes the maximum desired distance between intra-cluster neighbors. The simulation results of this case are shown in Figure 4.41 where: $D = \text{diag}(1, 1.2, 1.3, 3, 3.5, 4)$, see (4.7).



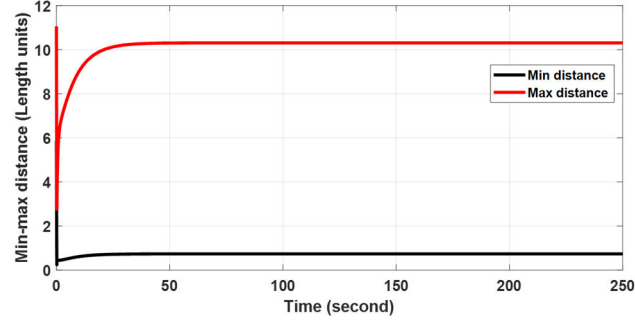
(a)



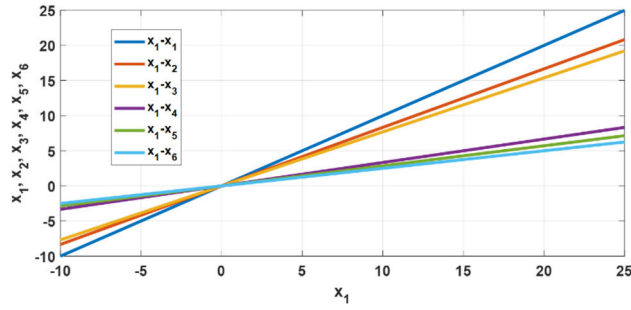
(b)



(c)



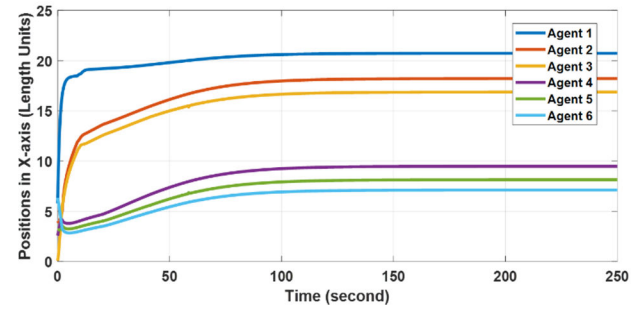
(d)



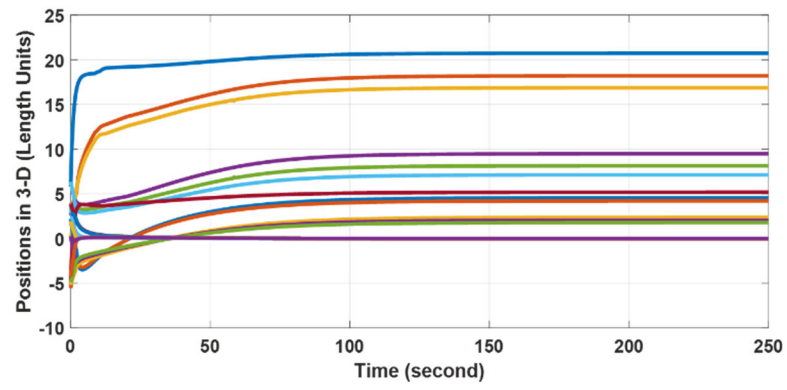
(e)

Figure 4.41. The simulation results of the example shown in Figure 4.40 using the 3-D version of the C-S model with $\gamma = 1$, $H = 10$, $\delta = 0$ and $\beta = 0.5$. (a) agents' positions along the x-axis. (b) agents' positions in 3-D versus time. (c) agents' positions in the working space. (d) min-max distance among neighboring agents on the graph shown in Figure 4.36. (e) the equilibria of (4.55.4) where all equilibria can be easily written in terms of x_1 starting with x_2 and x_3 .

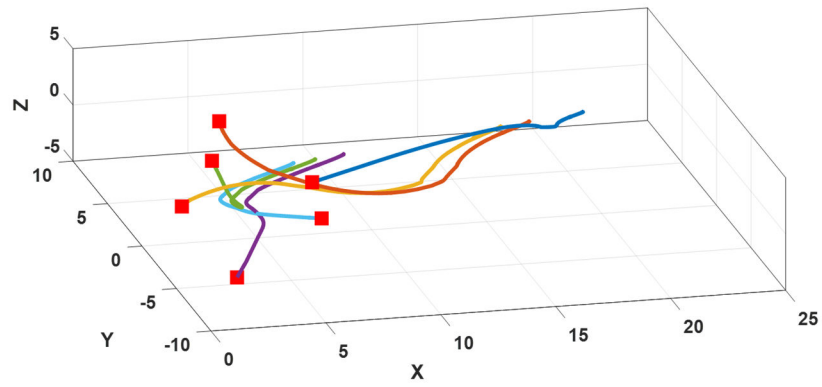
The same scenario is simulated using (4.55.5) to reflect upon the advantage of the state-dependent clustering weights. Figure 4.42 shows the results obtained when $TH = 81$ where the final intra cluster separation is \sqrt{TH} .



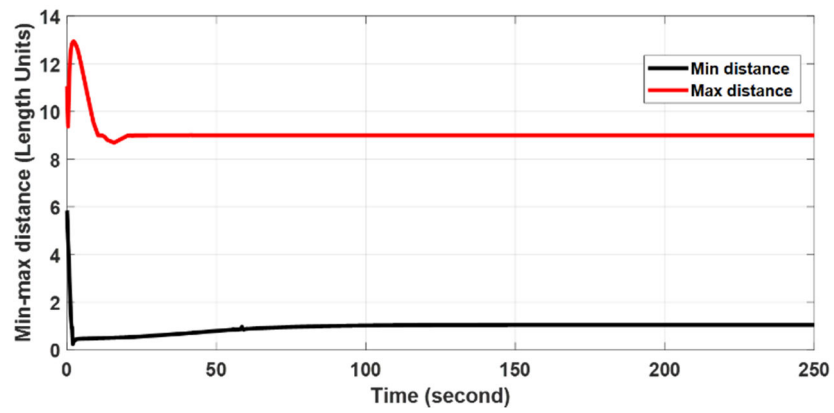
(a)



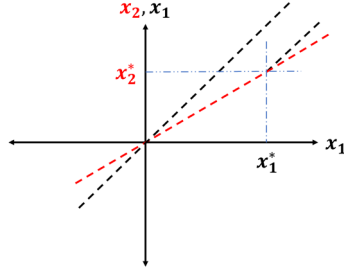
(b)



(c)



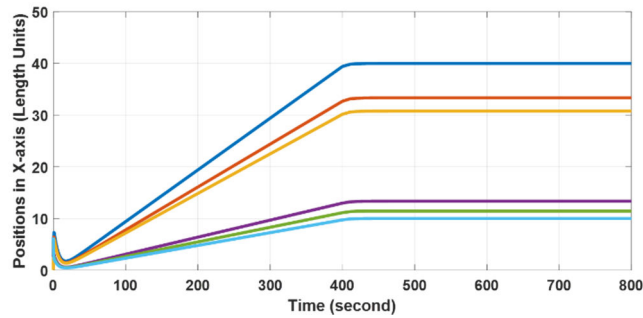
(d)



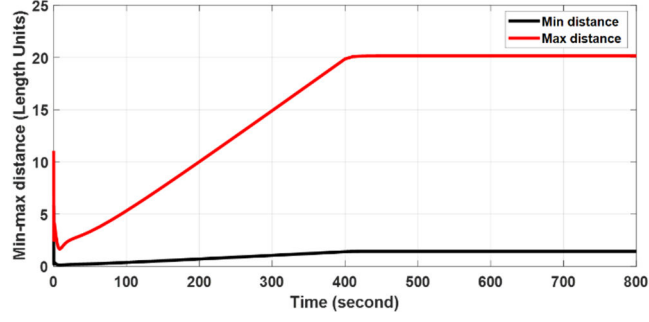
(e)

Figure 4.42. The simulation results of the example shown in Figure 4.40 using the 3-D version of the C-S model with $\gamma = 1$, $H = 10$, $\delta = 0$ and $\beta = 0.5$ and the modified clustering weights given by (4.55.5). (a) agents' positions along the x-axis. (b) agents' positions in 3-D versus time. (c) agents' positions in the working space. (d) min-max distance among neighboring agents on the graph shown in Figure 4.40. (e) the equilibria of edge $\{1,2\} \in \mathcal{E}$ written in terms of x_1 and x_2 when (4.55.5) is used.

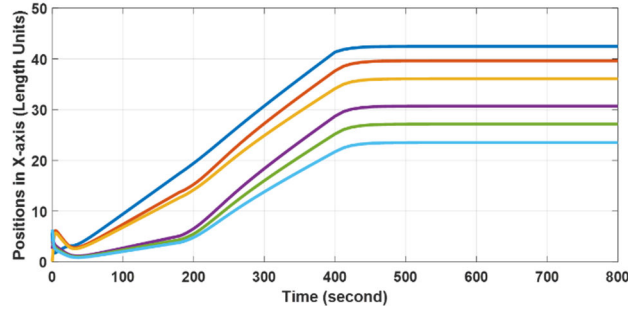
To appreciate the modified clustering weights given by (4.55.5), let us have the scenario shown in Figure 4.40 however with the modification of ramp command input issued by the virtual leader. Figure 4.43 shows the results obtained with and without the cluster sizing control offered by (4.55.5). It is easy to read from the figures that if the ramp continues for a longer period, i.e., the target point is far away, then the agents will become increasingly distant from each other and therefore there is a possibility to lose the connectivity of the MAS. However, this is not the case under (4.55.5).



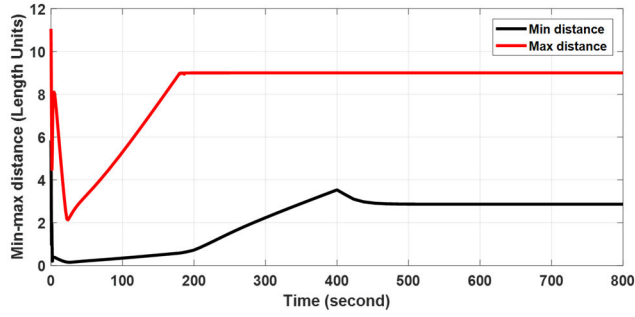
(a)



(b)



(c)



(d)

Figure 4.43. The simulation results of the example shown in Figure 4.40 using the 3-D version of the C-S model with $\gamma = 1$, $H = 10$, $\delta = 0$ and $\beta = 0.5$ and the modified clustering weights given by (4.55.5). (a)-(b) and (c)-(d) agents positions along the x-axis and the minimum and maximum distance among neighboring agents on the graph shown in Figure 4.40 without and with the clustering size control with $TH = 81$ respectively, and $p_i = 1$.

The modified clustering weights, i.e., $C_{ij}^w(d_{ij,f}^x, q_i, q_{j \in \mathcal{N}_i})$, given in (4.55.5) can be generalized to meet various application requirements as follows:

$$C_{ij}^w(d_{ij,f}^x, q_i, \quad j \in \mathcal{N}_i) = e^{\pm(h(q_i, q_j) - \mu_i)^2} \quad (4.55.6)$$

where: $h(q_i, q_j)$ is the criterion to be minimized or maximized relative to a specified threshold μ_i .

To bridge the link between consensus and clustering, let us have the following affine-in-control-input scalar nonlinear system as a representative of an agent dynamics. Then, (4.55.6) can be seen into the picture as follows:

$$\dot{x}_i = f(x_i) + g(x_i)u_i, \quad g(x_i) \neq 0 \quad \forall x_i \quad (4.56)$$

with an exact feedback linearization [148] controller:

$$u_i = g^{-1}(x_i) \left\{ -\tilde{C}_{ij}^w f(x_i) + \sum_{j \in \mathcal{N}_i} g_{ij} \{ C_{ij}^w x_j - x_i \} \right\} \quad (4.56.1)$$

where: \tilde{C}_{ij}^w denotes the overall in-neighborhood criteria evaluation and has a structure like C_{ij}^w .

To submit itself to the team, the i^{th} agent must be assured about the benefits gained upon teaming up with the rest. These guarantees are reflected in \tilde{C}_{ij}^w . Being a member in the team, i.e., $\tilde{C}_{ij}^w = 1$, the i^{th} agent can determine how to react to other agents in the same team, i.e., cooperatively or competitively, based on C_{ij}^w . When $C_{ij}^w = 1 \quad \forall i, j$, then a consensus is achieved when at least g_{ij} satisfies (3.35). Note that (4.56.1) is closely related to the general class given by (3.72) and (3.75).

Remark 4.11: The universal protocol can be also used under the clustering behavior.

Back to (4.55.5), and by having Figure 4.42.e in mind, the effect of the cluster size control- as seen from Figure 4.44- is equivalent to switching between consensus and formation;

where the desired shape is controlled by Δ_{ij}^x , see (4.22). Therefore, the relations among consensus, clustering and formation are now obvious.

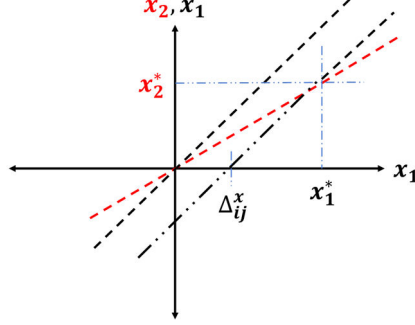


Figure 4.44. Effect of cluster size control approach on clustering behavior.

4.1.5.2: The shape consensus behavior

Unlike the formation control approach, shape consensus can achieve a desired shape only by affecting the way the left eigenvector associated with $\lambda_1(\vec{x}) = 0$ is spanning the column space, i.e., the left null-space, of the Laplacian matrix. In the average consensus for example, the left null-space is spanned by the $\vec{1}^T$ vector while in shape consensus it is spanned using a probably signed vector whose elements are not necessarily 1. However, the resulting Laplacian matrix under shape control is still positive semi-definite and therefore stable- as explained previously-, but it is weight-unbalanced, i.e., the row-sum does not equate to zero. Inspired by [1] and utilizing (4.8), a shape control shown in Figure 4.45 was achieved in 2-D by manually tuning the cooperative-competitive weights. One feature of the proposed shape control is its flexibility to scaling and reflection due to the agents' initial positions. It was verified through numerical simulation that the translation also affects the scale of the resulting shape. So, the used weights could be adaptively adjusted to overcome these issues if they are undesirable, may be using (4.55.5).

Remark 4.12: What motivated the shape control proposed herein is our vision to provide a way for agents to relate their preferred shape with the underlying communication network connecting them which enables switching the shape whenever the underlying network switches.

The underlying communication network used to achieve the shape shown in Figure 4.45 is depicted in Figure 4.46 and the cooperative-competitive weight matrices used were as follows- see [1] for more details:

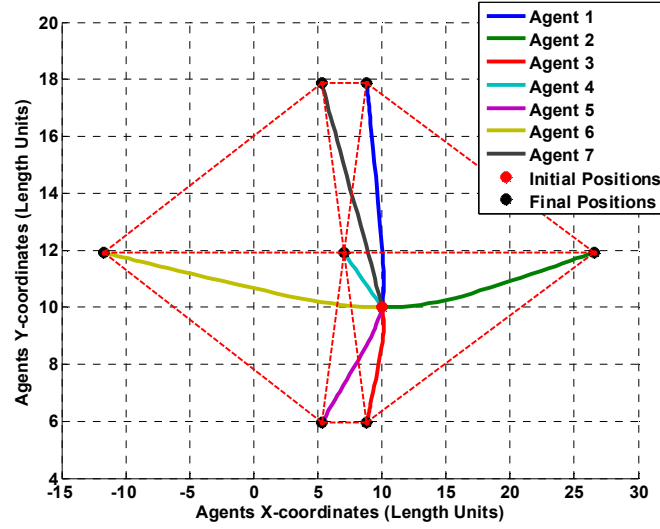


Figure 4.45. Shape control results under C-S distance-based protocol where agents $\{6\}$, $\{2\}$, $\{4\}$, $\{1,3\}$ and $\{5,7\}$ form five clusters in the x-direction while agents $\{3,5\}$, $\{2,4,6\}$ and $\{1,7\}$ form three clusters in the y-direction. Agent 4 is the center of this shape.

$$w_x = \begin{bmatrix} [0] & 1/d_{x1} & [0] \\ d_{x1} & 0 & d_{x2} \\ [0] & 1/d_{x2} & [0] \end{bmatrix} \quad (4.57)$$

$$w_y = \begin{bmatrix} [0] & 1/d_{y1} & [0] \\ d_{y1} & 0 & d_{y2} \\ [0] & 1/d_{y2} & [0] \end{bmatrix} \quad (4.57.1)$$

where:

$d_{x1} = [0.8, 0.2655, 0.8]$, $d_{x2} = [1.3332, -0.6, 1.3332]$, $d_{y1} = [0.6667, 1.0, 2.0]$, $d_{y2} = [2.0, 1.0, 0.6667]$ and $[0]$ is 3×3 zero matrix. These design variables can be directly related to the desired shape and the left and right eigenvectors associated with $\lambda_1 = 0$ of the new resulting graph. This will help in creating a systematic method to implement the shape consensus as presented herein. This will be a subject of a future work. Note that similar approach was presented in [149].

The shape controller in the x-direction is given as follows- see (4.8):

$$\vec{z}_1 = ((w_x + I_{7 \times 7}) \circ \mathcal{L}(\vec{z}_1)) \vec{z}_1 \quad (4.57.2)$$

where: \vec{z}_1 is given in (4.8), and \circ denotes the Hadamard product of two matrices, i.e., an element wise product, and I is the identity matrix. The shape controller in the y-direction can be given in a similar manner.

Once the shape control proposed in [131] is compared to the one proposed herein, it is straightforward to notice that both shape control approaches do require an explicit similarity between the communication network and the resulting shape. So, they give a way to *get-what-you-see* which will be highly appreciated by the design engineers. After simplification, the cluster consensus value reported in [1] can be given as follows:

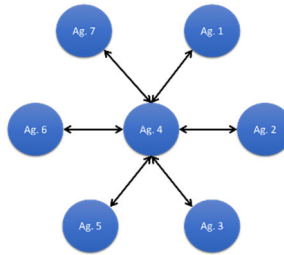


Figure 4.46. Network topology used in 2-D shape control shown in Figure 4.38.

$$\vec{c} = \left(\frac{1}{N} \sum_{i=1}^N \frac{x_i(t_0)}{e_i} \right) \vec{e} \quad (4.57.3)$$

where: \vec{e} is the right eigenvector associated with $\lambda_1 = 0$. Note that under the proposed shape control, the elements of the left eigenvector associated with $\lambda_1 = 0$ are the reciprocals of their corresponding elements in the right eigenvector. It is also worth noting that the control effort will be zero once the shape consensus is achieved. Even though a state-dependent version is applicable as stated previously, it was not used to obtain the results shown in Figure 4.45. Other cooperative-competitive behaviors can be realized similar to what was reported in [150]. More information about cooperative and competitive systems described by ordinary differential equations can be found in [151] and [152].

4.1.6: Containment and Escorting in obstacle-free 2-D plane

In some applications, agents must conduct special kinds of behaviors like containing a target or valuable assets while escorting them to a desired location. This can be found in autonomous surface vessels acting as coast guards and warships escorting, for instance. Although such behaviors can be viewed from formation control point of view, it is intended herein to avoid such approaches and to provide a more flexible approach that does not require hardcoded information to maintain the connectivity while achieving the desired task objectives.

Let the behavior be realized using the following directed communication network:

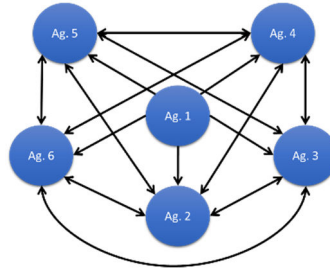


Figure 4.47. Network topology used in containment and escorting example.

As suggested by this communication network, it is desired to contain agent 1 and escort it while it is moving in the 2-D plane. Agent 1 acts as a leader to the other agents, however

if it only has inwardly directed links then it will be a follower for multiple leaders. Geometrically, there will be two concentric regions- preferably circular regions- inside which the desired behavior is to be achieved while having the maximum separation among the surrounding agents whose connectivity must be maintained. Figure 4.48 shows the containment region and the available forces among agents.

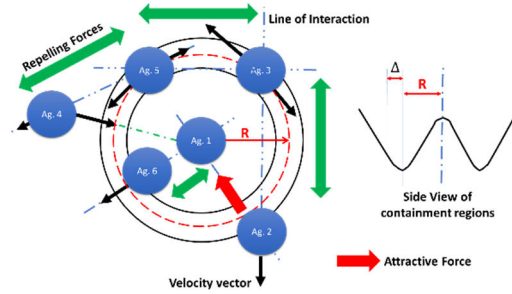


Figure 4.48. Desired containment region, its approximate side view and the available forces among agents.

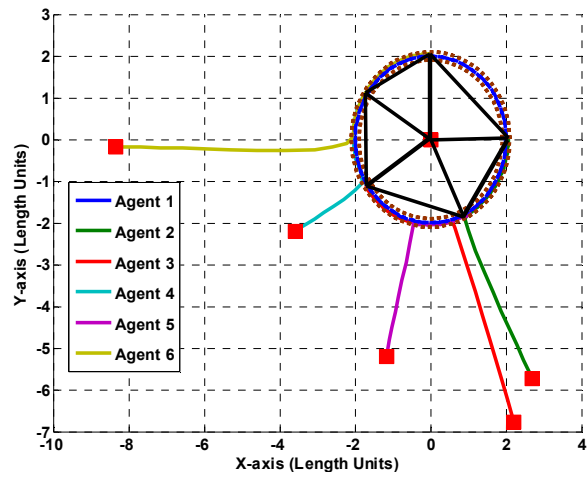
Thus, the functions used for the x-axis motion can be given as follows

$$p_{i1}(x_1, x_i) = \gamma_{i1} \{ \tanh(k(r_{i1} - \Delta - R)) + \tanh(k(r_{i1} + \Delta - R)) \} \quad (4.58)$$

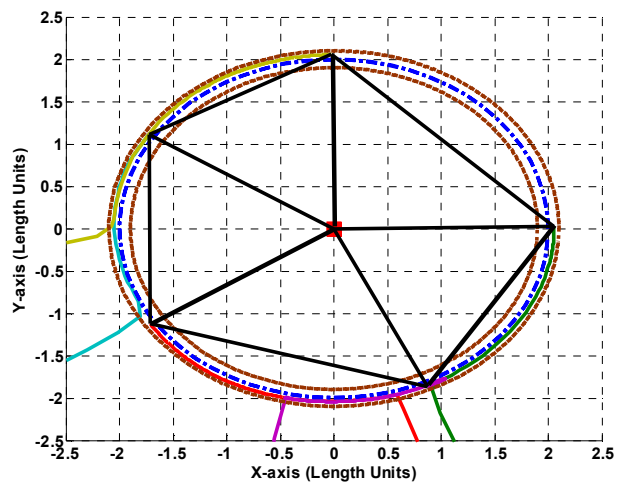
$$g_{ij}(x_j, x_i) = \gamma_{ij} \tanh(k(TH - h_{i1})) \hat{g}_{ij}, \quad i, j \neq 1 \quad (4.58.1)$$

where: p_{i1} is the pinning gain, \hat{g}_{ij} is given in (4.34), $\gamma_{i1} > 0$ and $h_{i1} = 1/(1 + \alpha r_{i1}^2)$ with $\alpha > 0$ and $r_{i1} = \sqrt{(x_i - x_1)^2 + (y_i - y_1)^2}$. For the y-axis motion, simply replace x_i in \hat{g}_{ij} with y_i .

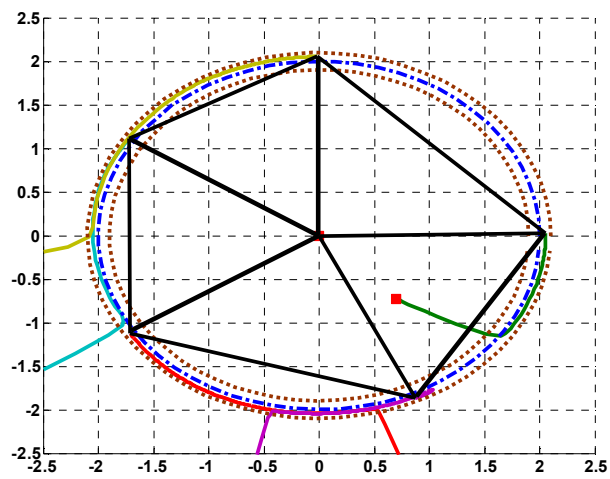
The simulation results for both containment and escorting are shown in Figure 4.49. One main difference between this behavior and the formation behavior explained previously is that agents in the former behavior decide online in a distributed fashion the best formation that meet the requirements of both containment and connectivity preservation. So, in general one should not expect to have almost regular polygons for any R and TH values. The values used to generate Figure 4.49 were as follows: $R = 2$, $TH = 0.1$ and $\Delta = 0.1$.



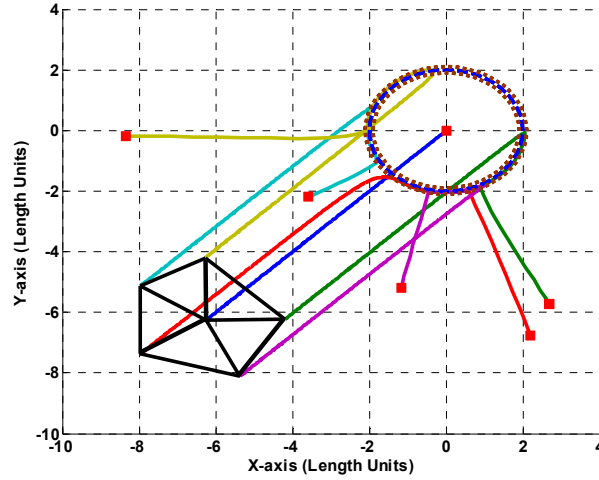
(a)



(b)



(c)



(d)

Figure 4.49. Containment and escorting results. (a): containment phase. (b): close view. (c): Agent 2 is initially close to agent 1. (d): escorting phase where all agents are connected to the virtual leader.

4.1.7: Switching behaviors

As, the name implies, this behavior consists of several sub-behaviors among which the agents switch based on a switching signal generated maybe using a behavior selection mechanism in a centralized or a distributed fashion or even as a response to triggered events. This mechanism may depend on a distributed or collective artificial intelligence or it may have a human-in-the-loop who is responsible for deciding what to do. The collection of sub-behaviors consist what is referred to as a behavior bank as depicted in Figure 4.1.

In general, the switching among the previously mentioned behaviors and others as well must be stable to make sure that the overall mission is achieved. In switching systems theory, the working space, say \mathbb{R}^3 in the 3-D Euclidean space, is partitioned into several operating regions (OPR) separated by a family of switching surfaces. During the transition from one OPR to another, the states might experience a jump-like response referred to as the impulse effect. To overcome this effect, a reset map is devised. When the impulse effects are ignored, one may argue that the resulting system is a discontinuous system and

not hybrid [137]. Nevertheless, a stability approach for switching systems will be used to analyze the stability of the proposed behaviors. Following the approach presented in [90] and by finding a valid common Lyapunov function [137], the stability of the MAS with switched behaviors can be studied as follows:

Let $\vec{q} \in \mathbb{R}^N$ be continuous and let Γ_N be a finite set of all graphs involved, i.e., \mathcal{G}_i , from which a behavior $b_n \in \mathcal{B}_m$ is selected at a given time instant by an arbitrary discrete switching signal $s(t)$. \mathcal{B}_m denotes the finite set of all possible behaviors available in the behavior bank. It is assumed that all graphs are strongly connected or simply connected- in the case of undirected graphs- in which case $\lambda_1(\vec{q}) = 0 \ \forall \vec{q}$. Consequently, $\vec{1}^T$ is a nonlinear left eigenvector associated with $\lambda_1(\vec{q}) = 0$ indicating that the MAS has an invariant quantity. Note that this automatically will exclude the clustering behavior- at least from the coming discussion-; because the clustering behavior simply does not have $\vec{1}^T$ as a left eigenvector. Mathematically, Γ_N is given as follows:

$$\Gamma_N = \{\mathcal{G}_l(\vec{q}, b_n) = (\mathcal{V}, \mathcal{E}, \mathcal{W}(\vec{q}, b_n)) : \text{rank}(\mathcal{L}_s(\mathcal{G}_l)) = N - 1, \vec{1}^T \mathcal{L}_s = \vec{0}\} \quad (4.59)$$

where: $\vec{q} = \{\vec{x}, \vec{y}, \vec{z}\}$ assuming no coupling among the spatial directions. Γ_N is finite because the number of vertices, i.e., N , involved is fixed resulting at max in $N(N - 1)$ different graphs [90]. The overall dynamics of the MAS can therefore be given as follows:

$$\dot{\vec{q}}(t) = \vec{f}_{s(t)}(\vec{q}(t)) \quad (4.59.1)$$

where: $\vec{f}_{s(t)}$ is mainly constructed using the previously shown semi-linear protocols comprising $\mathcal{L}_s(\mathcal{G}_k)$, $s(t) = \{l, n\} > 0 \in \mathbb{N}$ are the indices of the graph $\mathcal{G}_l \in \Gamma_N$ and the behavior $b_n \in \mathcal{B}_m$. Note that $\vec{f}_{s(t)}$ could also result from a nonlinear protocol as was shown in Chapter 3.

Remark 4.13: A single graph may facilitate several behaviors and several graphs could facilitate the same behavior.

The stability of the behavioral-switched MAS can be guaranteed by the following theorem.

Theorem 4.11: The behavioral-switched MAS given in (4.59.1) is stable for all arbitrary switching, moreover its invariant quantity is globally achieved if and only if all its switched instances are connected and having $\vec{1}^T$ as a nonlinear left eigenvector associated with $\lambda_1(\vec{q}) = 0 \forall \vec{q}$.

Proof: Let the MAS consists of a fixed number of agents, and a finite number of behaviors acting over a finite number of connected graphs, so at each switching, a new behavior and/or network appears in the MAS dynamics. If each graph is connected under the active behavior at that specified switching, then $\lambda_1(\vec{q}) = 0$ is a simple eigenvalue to which the MAS invariant quantity is related. Moreover, designing the g_{ij} functions for each behavior $b_n \in \mathcal{B}_m$ such that both (3.34) and (3.35) are satisfied, will ensure that $\vec{1}^T$ is a nonlinear left eigenvector associated with $\lambda_1(\mathcal{L}_{s(t)}(\mathcal{G}_l)) = 0 \forall \vec{q}$. This means that for semi-linear protocols especially those resulting in symmetric Jacobians, we may define the disagreement vector $\vec{\varepsilon}(t)$ as [90]- see also (3.90):

$$\vec{q}(t) = c\vec{1} + \vec{\varepsilon}(t) \quad (4.59.2)$$

Taking the derivative with respect to time, we may write the disagreement dynamics as follows:

$$\dot{\vec{q}}(t) = -\mathcal{L}_{s(t)}(\vec{q})\vec{q} = \vec{\varepsilon}(t) \quad (4.59.3)$$

Substituting (4.59.2) into (4.59.3), yields:

$$\vec{\varepsilon}(t) = -\mathcal{L}_{s(t)}(c\vec{1} + \vec{\varepsilon}(t))\{c\vec{1} + \vec{\varepsilon}(t)\} \quad (4.59.4)$$

Which by assuming that the g_{ij} functions used in all $\mathcal{L}_{s(t)}(\vec{q})$ depends on the relative error between states, i.e., $g_{ij}(q_j - q_i)$, and knowing that- from (3.35)- $\vec{1}$ is a nonlinear right eigenvector associated with $\lambda_1(\vec{q}) = 0 \forall \vec{q}$, one can easily show that (4.59.4) is equal to:

$$\vec{\varepsilon}(t) = -\mathcal{L}_{s(t)}(\vec{\varepsilon}(t))\vec{\varepsilon}(t) \quad (4.59.5)$$

Now, let the following positive-definite and smooth function be a common Lyapunov candidate function:

$$V(\vec{\varepsilon}) = \frac{1}{2} \vec{\varepsilon}^T \vec{\varepsilon} \quad (4.59.6)$$

So, by taking the time-derivative and from (4.59.5), we will have the following:

$$\frac{dV}{dt} = -\vec{\varepsilon}^T \mathcal{L}_{s(t)} \vec{\varepsilon} < 0, \quad \forall \vec{\varepsilon} \neq \vec{0} \quad (4.59.7)$$

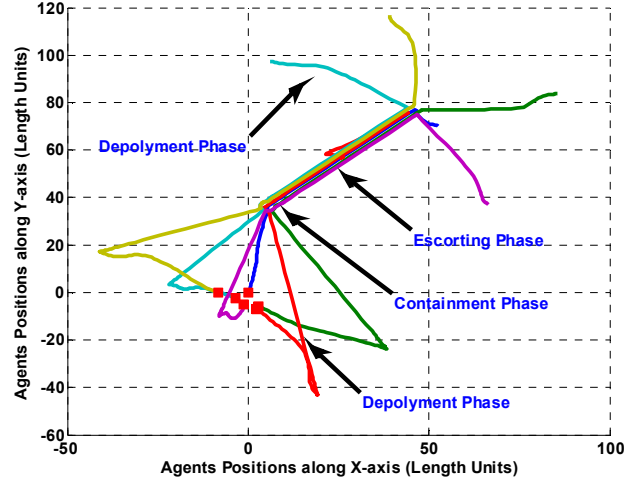
This result can be seen from the fact that $\mathcal{L}_{s(t)}$ is an *M-matrix* $\forall \vec{\varepsilon}$. Since $V(\vec{\varepsilon}) > 0, \forall \vec{\varepsilon} \neq \vec{0}$ and $V(\vec{\varepsilon}) = 0$ when $\vec{\varepsilon} = \vec{0}$, therefore (4.59.6) is indeed a common Lyapunov function and therefore the behavioral-switch MAS (4.59.1) is stable for arbitrary switching.

■

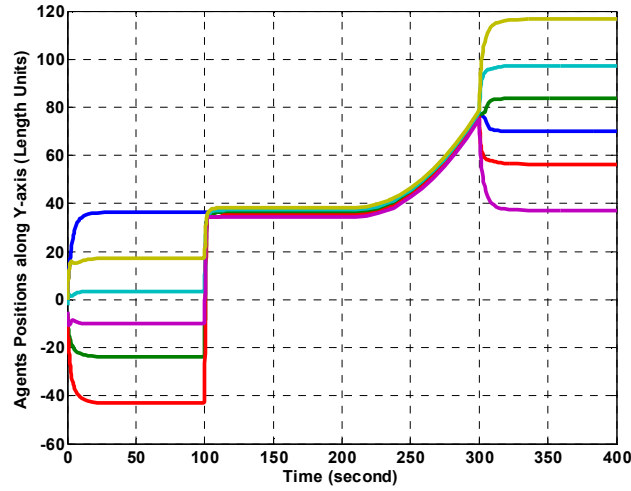
The following example is a demonstration of the previous theorem. Consider a MAS entitled to search a working space to find specific targets, once an agent finds a target, then all agents will contain it along with the target found and escort them to a specified place. Once the target is delivered, then the cycle starts again.

To simulate this behavior, a switching signal based on time- to avoid further complications and to stay focused on the objective- is used to switch in a planned manner between three main behaviors, namely: deployment, containment and escorting all in 2-D obstacle-free plane. During escorting, all agents followed an externally generated command to direct

them to the delivery point after which agents commenced again their search mission. These phases are shown in Figure 4.50.



(a)



(b)

Figure 4.50. The switching behaviors results. (a): in 2-D. (b): Agents y-positions.

4.1.8: Agents with scalar and nonlinear affine-in-control-input

To push further the applicability of the proposed framework, let us reconsider the agent dynamics given in (4.56). So, under the semi-linear protocol given in (4.2), we may write (4.56) as follows-in the x -direction assuming no coupling with other directions:

$$\dot{x}_i = f(x_i) + g(x_i) \sum_{j \in \mathcal{N}_i} g_{ij} \{x_j - x_i\} \quad (4.60)$$

under which the overall MAS dynamics can be written as:

$$\dot{\vec{x}}(t) = F(\vec{x}) - G(\vec{x})\mathcal{L}(\vec{x})\vec{x}(t) = \vec{f}_s(\vec{x}) \quad (4.60.1)$$

Moreover, if $F(\vec{x})$ can be written or approximated as $F(\vec{x}) = A(\vec{x})\vec{x}$, then (4.60.1) becomes:

$$\dot{\vec{x}}(t) = \{A(\vec{x}) - G(\vec{x})\mathcal{L}(\vec{x})\}\vec{x}(t) = \vec{f}(\vec{x}) \quad (4.60.2)$$

Similarly, if a nonlinear protocol is used, say (3.45), then it is easy to show that (4.56) can be written as follows:

$$\dot{\vec{x}}(t) = F(\vec{x}) + G(\vec{x})\Phi(\vec{x}) = \vec{f}_n(\vec{x}) \quad (4.60.3)$$

If $F(\vec{x}) = A(\vec{x})\vec{x}$, then (4.60.3) can be given as:

$$\dot{\vec{x}}(t) = A(\vec{x})\vec{x} + G(\vec{x})\Phi(\vec{x}) = \vec{f}_n(\vec{x}) \quad (4.60.4)$$

which is the same setup used when dealing with state-feedback in the control community.

In this point of view, the state-feedback controller is $\vec{U}(\vec{x}) = \Phi(\vec{x})$. However, the affine-in-control-input nonlinear system given by (4.60.4) follows a certain structure imposed by the underlying communication network. Therefore, aspects like structural controllability [153] and structural observability [154] are discussed.

Applying the proposed framework to (4.60.1) or (4.60.2), one may find out the conditions, if possible, to design the g_{ij} functions to achieve certain behaviors. The same goes for (4.60.3) and (4.60.4) as well. Mainly, for (4.60), we may use the following theorem.

Theorem 4.12: Considering the proposed framework, the exact feedback linearization is a special solution for the synchronization problem among agents in a MAS when $g_{ij}(x_i, x_j) \equiv g_{ij}(x_i)$ is a non-diffusive weighting function.

Proof: After finding the Jacobian of (4.60), we may force $\vec{e} = \vec{1}$ as a nonlinear right eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ by solving the following partial differential equation (PDE) $\forall i, j \in \mathcal{E}$ as pairs- if possible:

$$\begin{aligned} \frac{\partial f(x_i)}{\partial x_i} + \{x_j - x_i\} \left\{ g(x_i) \left[\frac{\partial g_{ij}}{\partial x_i} + \frac{\partial g_{ij}}{\partial x_j} \right] + \frac{\partial g(x_i)}{\partial x_i} g_{ij} \right\} &= 0 \\ \frac{\partial f(x_j)}{\partial x_j} + \{x_i - x_j\} \left\{ g(x_j) \left[\frac{\partial g_{ji}}{\partial x_i} + \frac{\partial g_{ji}}{\partial x_j} \right] + \frac{\partial g(x_j)}{\partial x_j} g_{ji} \right\} &= 0 \end{aligned} \quad (4.60.5)$$

Similarly, to force $\vec{v}^T = \vec{1}^T$ as a nonlinear right eigenvector associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$ by solving the following PDE $\forall i, j \in \mathcal{E}$ as pairs - if possible:

$$\begin{aligned} \frac{\partial f(x_i)}{\partial x_i} + \{x_j - x_i\} \left\{ g(x_i) \frac{\partial g_{ij}}{\partial x_i} - g(x_j) \frac{\partial g_{ji}}{\partial x_i} + \frac{\partial g(x_i)}{\partial x_i} g_{ij} \right\} + \{g(x_j)g_{ji} - g(x_i)g_{ij}\} &= 0 \\ \frac{\partial f(x_j)}{\partial x_j} + \{x_i - x_j\} \left\{ g(x_j) \frac{\partial g_{ji}}{\partial x_j} - g(x_i) \frac{\partial g_{ij}}{\partial x_j} + \frac{\partial g(x_j)}{\partial x_j} g_{ji} \right\} + \{g(x_i)g_{ij} - g(x_j)g_{ji}\} &= 0 \end{aligned} \quad (4.60.6)$$

If we are interested only in synchronization of agents, i.e., without preserving any time-invariant quantity like average mean for example, then we may only force $\vec{e} = \vec{1}$ only by solving (4.60.5) for the i^{th} and j^{th} agents, separately such that $x_j - x_i \rightarrow 0$ as $t \rightarrow \infty$. Let $f(x_i)$ in (4.60) be stable, then $\partial f(x_i)/\partial x_i$ will vanish as $t \rightarrow \infty$. So, by taking:

$$g_{ij}(x_i, x_j) = g^{-1}(x_i), \quad g(x_i) \neq 0 \quad (4.60.7)$$

Therefore, we have:

$$\frac{\partial g_{ij}}{\partial x_i} = -\frac{\partial g(x_i)}{\partial x_i} \frac{1}{g^2(x_i)} \quad (4.60.8)$$

So, (4.60.5) is solved $\forall i, j \in \mathcal{E}$. Note that (4.60.7) is the exact feedback linearization approach.

■

Remark 4.14: In (4.60.5) and (4.60.6), if $g(x_i) = g(x_j) = 1$ and we take $g_{ij} = g_{ji}$ both being diffusive weighting functions, then by forcing (3.34) and (3.35) the synchronization between the two agents will be achieved assuming $f(x_i)$ and $f(x_j)$ to be stable.

To force both $\vec{e} = \vec{1}$ and $\vec{v}^T = \vec{1}^T$ as nonlinear right and left eigenvectors associated with $\lambda_1(\vec{x}) = 0 \forall \vec{x}$, i.e., to have a symmetric Jacobian, we need to solve the following system of PDEs at once- say for (4.60.1) in the x-direction assuming no coupling with other directions:

$$\vec{1}^T \frac{\partial \vec{f}_s(\vec{x})}{\partial \vec{x}} \vec{1} = 0, \quad \forall \vec{x} \quad (4.60.9)$$

given: $\vec{x}(t_0) = \vec{x}_0$ such that:

$$\begin{aligned} \frac{\partial \vec{f}_s(\vec{x})}{\partial \vec{x}} &< 0, \quad \forall \vec{x} \notin \mathfrak{S}^x \\ \frac{\partial \vec{f}_s(\vec{x})}{\partial \vec{x}} &= 0, \quad \forall \vec{x} \in \mathfrak{S}^x \end{aligned} \quad (4.60.10)$$

If a solution does exist, then it will be stable and invariant.

When $\partial f(x_i)/\partial x_i = \partial f(x_j)/\partial x_j = 0$, we may solve (4.60.9) for a single-edge MAS- whose agents dynamics are given in (4.60)- by simply adding the two equations available in (4.60.5) or (4.60.6) resulting in:

$$\begin{aligned} \{x_j - x_i\} \{ \nabla(g(x_i)g_{ij})\vec{1} - \nabla(g(x_j)g_{ji})\vec{1} \} &= 0 \\ \{x_j - x_i\} \{ L_{\vec{1}}(g(x_i)g_{ij}) - L_{\vec{1}}(g(x_j)g_{ji}) \} &= 0 \end{aligned} \quad (4.60.11)$$

where: $L_{\vec{1}}$ denotes the directional derivative, i.e., Lie derivative. (4.60.11) can be satisfied

$\forall x_j \neq x_i$ by forcing the following condition:

$$L_{\bar{1}}(g(x_i)g_{ij}) = L_{\bar{1}}(g(x_j)g_{ji}) \quad (4.60.12)$$

for both diffusive and non-diffusive g_{ij} weighting function.

Using **Theorem 4.12** to avoid complications, and despite being different from the one given in (4.9), the $G(\vec{x})$ matrix in (4.60.1) and (4.60.3) can be used under exact feedback linearization and a semi-linear protocol as follows- assuming $g(x_i) \neq 0$ in the domain of interest:

$$u_i = g^{-1}(x_i) \left\{ -f(x_i) + \sum_{j \in \mathcal{N}_i} g_{ij} \{x_j - x_i\} \right\} \quad (4.60.13)$$

The resulting MAS system dynamics under a semi-linear and a nonlinear protocol can be then given as follows:

$$\begin{aligned} \vec{\dot{x}}(t) &= -\mathcal{L}(\vec{x})\vec{x}(t) = \vec{f}_s(\vec{x}) \\ \vec{\dot{x}}(t) &= \Phi(\vec{x}) = \vec{f}_n(\vec{x}) \end{aligned} \quad (4.60.14)$$

which brings us back to our proposed framework.

To conclude, despite the applicability of our framework under the general case given in (4.56), we have preferred to focus on realizing coordination protocols that can be considered as command signals to be tracked by dedicated local agent controllers. The advantages gained under this philosophy will be obvious in the next chapter.

4.2 Conclusion

In this chapter, several examples of motion-related behaviors were presented. Unlike most of the reported results in the related literature, the framework proposed herein gave a systematic and holistic view of designing motion-related behaviors for multiagent systems on graphs. Stability proofs were presented also in a systematic way, thanks to the inherited *M-matrices* properties. Moreover, the framework enables introducing state-dependent

parameters and dynamics into the behaviors making process. The needed conditions to keep both invariance and stability under such dependency were obtained using the same framework. Therefore, we see – so far- the proposed framework as a sophisticated coordination motion planner that provides the foundation of further extensions in which, including the time as an additional dimension, distributed measurements and intelligence, navigation and guidance are made possible. This will result in a sophisticated semi or fully-autonomous mission orchestrating platform.

CHAPTER 5

ADVANCED BEHAVIORS

In this chapter, we continue our work by providing more sophisticated behaviors that are mainly directed to mobile agents. These agents could be maritime, airborne or ground robots. Both kinematical and kino-dynamical trajectory-generator systems are developed and integrated with reactive and hybrid-model-based-reactive intelligent controllers to facilitate interacting with more realistic working spaces. Harmonic potential fields are utilized to model the environment to enable collision-avoidance. Both collision-avoidance and connectivity-preserving behaviors are combined in a simple, yet efficient, way. Also, second-order, general linear time-invariant, nonlinear systems, and other models with higher dimensions are presented, where a special representation methodology reveals the usefulness of the proposed framework when dealing with such systems. The design steps presented can be easily upgraded to deal with systems of heterogeneous dynamical features. The strength of the kinematical trajectory-generator is demonstrated by guiding a group of non-holonomic front-wheel steered robots through a working space where obstacles exist. Another example demonstrating the use of the developed kino-dynamical trajectory-generator systems in controlling the behavior of a MAS consisting of six quadrotors is also presented.

In this part, more sophisticated behaviors that suit various types of robots and missions are provided. Mainly, in the previous parts, agents with simple dynamics were considered. In this part, dynamics with higher dimensions are used to build the trajectory-generator

systems. These systems can be of kinematical or kino-dynamical natures. As the name implies, a kinematical trajectory-generator system does not consider the steered agent dynamics explicitly. On the contrary, the kino-dynamical version considers the dynamics of the steered agent explicitly. Therefore, this version can be extended to include any order of dynamics in a lower-triangle hierarchical structure, which facilitate studying the stability in a top-down manner as will be discussed later.

Capturing the working environment is very important in guidance and path planning, in general. Sensory inputs and environment models can be combined to increase the awareness of the trajectory-generator system about the context evolving. This awareness, when combined with reasoning, opens the way to intelligent behavior mechanisms to handle almost every situation.

In this part, we will depend on harmonic potential fields to model the known parts of the environment and show how to mechanize the sensory inputs such that interesting behaviors can be generated. In Chapter 4, behavior banks were proposed in which mainly one behavior can be active at any given instant; however, in this part, this should not be the case any longer. Both sequential and parallel behaviors are addressed herein, and some examples are shown to emphasize their importance. For example, the collision-avoidance and connectivity-preserving behaviors can run simultaneously such that when an obstacle is avoided a preferable escape direction is nominated at which connectivity among agents will not decrease.

5.1 MAS utilizing trajectory-generators

In this section, and without loss of generality, a MAS consisting of N identical or nonidentical scalar dynamical systems is analyzed. Let us start with a fixed number, i.e., N , of scalar systems whose dynamics are modeled as follows:

$$\dot{x}_i(t) = a_i x_i(t) + b_i u_i(t) \in \mathfrak{R}, \quad \forall i = 1, 2, \dots, N \quad (5.1)$$

where: a_i, b_i are certain constants and could be different for each agent.

Let u_i be given as a combined input such that:

$$u_i = u_i^l + u_i^t \quad (5.2)$$

where:

$$u_i^l = -k_{1i} x_i$$

$$u_i^t = \frac{k_{2i}}{b_i} \left\{ (\Gamma_i - x_i) + \int_{t_0}^t (\Gamma_i - x_i) d\tau \right\}$$

and u_i^l is the local controller- which can be used to stabilize the dynamics if not originally stable or to meet certain performance criteria- and u_i^t is the tracking controller of the i^{th} agent. The trajectory signal of the i^{th} agent is generated using a single-integrator dynamical system -existing within the agent- as shown in the previous parts, and is given as follows:

$$\dot{\Gamma}_i = \alpha \sum_{j \in \mathcal{N}_i} g_{ij}(\Gamma_i, \Gamma_j) \{\Gamma_j - \Gamma_i\} \quad (5.3)$$

Where: $\Gamma_i(t_0) = x_i(t_0)$ and $g_{ij}(\Gamma_i, \Gamma_j): \mathfrak{R}^2 \rightarrow \mathfrak{R}_{>0}$ is a general weighting vector-valued functional that achieves the desired behavior, and Γ_j is the reference signal of the j^{th} agent who is neighboring the i^{th} agent. System (5.3) will make use of what was previously presented in Chapters 3 and 4.

Remark 5.1: Note that in (5.3), the $g_{ij}(\Gamma_i, \Gamma_j)$ can be $g_{ij}(x_i, x_j)$, $g_{ij}(x_i, \Gamma_j)$ or $g_{ij}(\Gamma_i, x_j)$ instead which indicates that actual scalar system states are used after being filtered through

the dynamics of their corresponding systems. However, this will introduce a direct nonlinear coupling- in general- between the system and its trajectory-generator co-system. So, the ideal reference signal generated, i.e., Γ_i , can be affected by both the actual state, i.e., x_i , and Γ_i . This will reduce the tracking error; because the reference signal generator is aware of the status of the actual system given in (5.1). In such a case, (5.3) can be generally considered as a nonlinear parameter varying system where the parameters include $x_i, \forall i = 1, 2, \dots, N$.

Remark 5.2: In Chapter 3, the g_{ij} functions were mainly of dynamical nature. In general, the implementation of these functions can also be realized using algorithmic approaches if it follows the conditions stated in Parts I and II. This will allow including the algorithms into the overall dynamical system and the stability can be analyzed more easily. This combination of algorithmic and dynamical approaches allows including more behaviors that depend more on logic. Consider implementing these functions using Fuzzy Logic, for example.

From a distributed MAS point of view, having a self-loop feedback should not be interpreted as if the agents are completely self-interested rather it should be looked at as an advantage added to the team since every agent can evaluate the global objective of the team in a distributed fashion and willingly accepts it based on the task assigned to it. This combination of self-awareness and teaming allows agents to change their behavior in a way that ensures their safety, stability and optimality for example while submitting themselves to the mission assigned to the whole team by the designer. It will be evident, through the sequel of this thesis, that enabling agents with some level of freedom to decide for themselves during mission conduction- based on their embedded artificial intelligence and

the planned actions- will enrich their behaviors and make them context-aware besides being *almost* autonomous. Moreover, this will ease the problem of designing the MAS, in general, when more realistic dynamical models are involved.

Having $\Gamma_i(t_0) = x_i(t_0)$ in (5.3) indicates that the agent- in a distributed manner- will try to minimize the tracking error, i.e., e_i , between its idealized behavior resembled by Γ_i and its actual behavior x_i for all times. This is depicted in Figure 5.1. Note that the local controller, namely: k_{1i} , can be designed such that the i^{th} agent is stabilized if not originally stable. The local controller can be designed using the standard techniques like optimal or pole placement in case it is certain. Notice that the local controller should not be used to cancel the system dynamics to bring it back to the single-integrator system; simply because an unstable local controller must be used to do so if the system is originally stable. This should be the case even if the system is originally unstable; since canceling its dynamics might reduce its robustness, significantly. If the system (5.1) happen to include uncertainty, then robust techniques should be used to design k_{1i} , if needed. If the uncertainty in (5.1) includes b_i , then b_i must be sign definite and the tracking controller should use the nominal value of b_i , i.e., \hat{b}_i . In general, the tracking controller, namely: k_{2i} , can be designed using a suitable fixed gain, adaptive or optimal techniques.

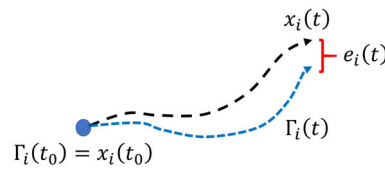


Figure. 5.1. The tracking error of the i^{th} agent.

Justified by the application, the local controller ensures the stability of an agent such that it can accommodate any parametric uncertainties involved in the agent dynamics, while the tracking controller follows the desired behavior generated by the trajectory-generator while

rejecting external disturbances. These relations are depicted in Figure 5.2. Note that a slowly varying trajectory signal will result in a dynamical friendly response of the MAS. The scalar system is like the single-integrator with added self-loop if viewed from a graph theoretic perspective. Figure 5.3 shows a MAS of three scalar systems where the self-loop is evident.

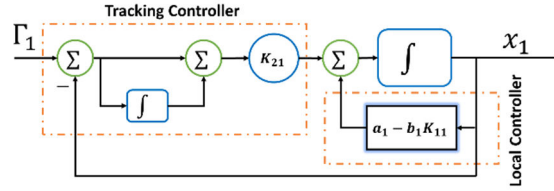


Figure. 5.2. The local and tracking controllers of the 1st agent.

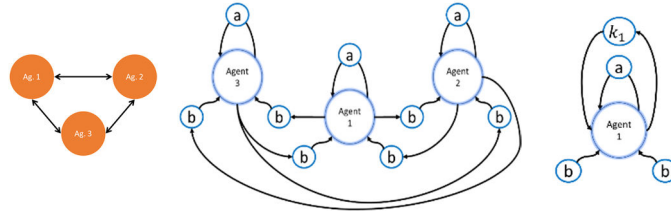


Figure. 5.3. Three identical agents with scalar dynamics building a MAS over an undirected graph without tracking controller. Left: Network topology. Middle: Signal flow diagram. Right: Effect of local controller.

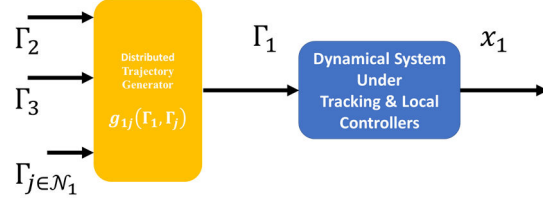
Recalling **Remark 5.1**, we may rewrite (5.3) as one of the following:

$$\dot{\Gamma}_i = \alpha \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, \Gamma_j) \{\Gamma_j - \Gamma_i\} \quad (5.4)$$

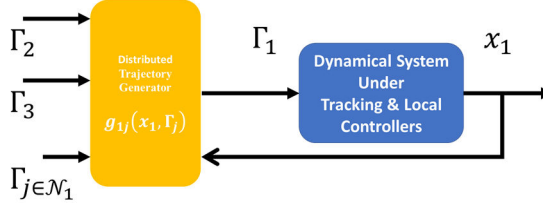
$$\dot{\Gamma}_i = \alpha \sum_{j \in \mathcal{N}_i} g_{ij}(\Gamma_i, x_j) \{\Gamma_j - \Gamma_i\} \quad (5.5)$$

$$\dot{\Gamma}_i = \alpha \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{\Gamma_j - \Gamma_i\} \quad (5.6)$$

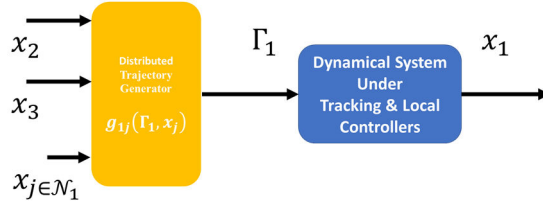
The effects of using (5.4) – (5.6) can be identified from Figure 5.4.



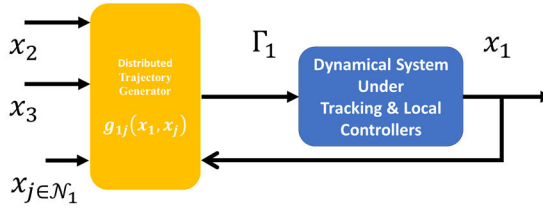
(a)



(b)



(c)



(d)

Figure. 5.4. The effect of using different couplings in g_{ij} functions. (a) $g_{ij}(\Gamma_i, \Gamma_j)$. (b) $g_{ij}(x_i, \Gamma_j)$. (c) $g_{ij}(\Gamma_i, x_j)$. (d) $g_{ij}(x_i, x_j)$. Examples are shown specifically for agent 1.

So, the overall MAS dynamics can be written into mainly three categories as follows- in the x-direction:

$$\vec{\Gamma} = -\alpha \mathcal{L}(\vec{\Gamma}) \vec{\Gamma} = \vec{f}^{\Gamma} \quad (5.7)$$

$$\vec{\Gamma} = -\alpha \mathcal{L}(\vec{x}, \vec{\Gamma}) \vec{\Gamma} = \vec{f}^{\Gamma, x} \quad (5.8)$$

$$\vec{\Gamma} = -\alpha\mathcal{L}(\vec{x})\vec{\Gamma} = \vec{f}^x \quad (5.9)$$

$$\vec{\Gamma} = -\alpha\mathcal{L}(\vec{\Gamma}, \vec{x})\vec{\Gamma} = \vec{f}^{x,\Gamma} \quad (5.10)$$

In fact, each connection between the actual system dynamics and the corresponding trajectory-generator is very interesting. In Figure 5.4.a, the trajectory-generator is neither aware of the actual system it is driving nor its neighbors and therefore generates a trajectory that suits the context viewed from the ideal states perspective. In Figure 5.4.b, the trajectory-generator of agent 1 tries to situate the agent based on the ideal state of its neighbors considering its own actual state, which is exactly the opposite to what is happening under the connection shown in Figure 5.4.c. The last connection shown in Figure 5.4.d indicates that the trajectory-generator of agent 1 is concerned about its actual state as well as its neighbors. A visualization of the previous effects can be read from Figure 5.5.

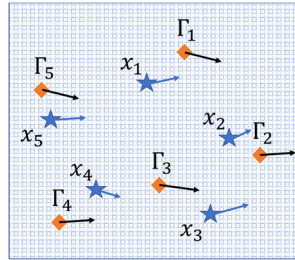


Figure. 5.5. Visualization of the effect of using different couplings in g_{ij} functions where stars denote the actual systems states and the squares denote the states of their trajectory-generator systems.

Applying the proposed framework conditions on the (5.7) – (5.9), and after solving (3.33.1) and (3.33.2) for each case, we find that (5.8) and (5.10) do satisfy (3.33.2) but not (3.33.1).

So, only the nonlinear right eigenvector $\vec{e} = \vec{1}$ is associated with $\lambda_1(\vec{\Gamma}) = 0 \forall \vec{\Gamma}$ but not $\vec{v}^T = \vec{1}^T$. Therefore, we will be using (5.7) and (5.9) or one of them in the coming discussion. Note that both (5.7) and (5.9) have $\vec{e} = \vec{1}$ and $\vec{v}^T = \vec{1}^T$ as nonlinear right and

left eigenvectors associated with $\lambda_1(\vec{\Gamma}) = 0 \forall \vec{\Gamma}$. In words, this means that the invariance properties are preserved.

The differences between (5.7) and (5.9) are listed as follows:

- 1- (5.7) is semi-linear system, while (5.9) is linear parameter varying (LPV) system.
- 2- Since (5.9) depends on the actual agent dynamics, it is aware of the agent status, while (5.7) is not; since it only depends on the ideal states of the trajectory-generator system.
- 3- (5.7) is a cascade connection while (5.9) is a loop connection.

The frequency domain transfer function of the i^{th} agent can be given as follows- assuming $x_i(t_0) = 0 \forall i$:

$$\frac{X_i(s)}{\Gamma_i(s)} = \frac{k_{2i}(s + 1)}{s^2 + (k_{2i} + b_i k_{1i} - a_i)s + k_{2i}} \quad (5.11)$$

where: s is the Laplacian operator. The stability of (5.11) is straightforward if and only if $k_{2i} + b_i k_{1i} - a_i > 0$ is satisfied -as can be obtained using Routh-Hurwitz criterion- and Γ_i is bounded.

Now, since the desired set-point value- under the current formulation- is mainly a function of the original system initial values, i.e., $\vec{x}(t_0)$, the initial value of $\vec{\Gamma}(t)$ must also be equal to $\vec{x}(t_0)$. When $g_{ij}(\Gamma_i, \Gamma_j)$ or $g_{ij}(\Gamma_i, x_j)$ is used, then a cascade connection is established, and when $g_{ij}(x_i, x_j)$ or $g_{ij}(x_i, \Gamma_j)$ is used, then a loop is introduced. Therefore, studying the stability of the augmented system, i.e., command generator and agent dynamics, is inevitable.

5.1.1: Stability of augmented systems

5.1.1.1: Stability of cascade-connected augmented systems

Let the general dynamics of the i^{th} agent and the trajectory-generator co-system be given as follows:

$$\vec{x}_i(t) = \vec{f}_i(t, \vec{x}_i, \vec{\Gamma}_i) \quad (5.12)$$

$$\vec{\Gamma}_i = \vec{h}_i(t, \vec{x}_i, \vec{\Gamma}_i) \quad (5.13)$$

where: $\vec{f}_i: [0, \infty) \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ and $\vec{h}_i: [0, \infty) \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ are piecewise continuous in the time, i.e., t , and locally Lipschitz in $\vec{w} = [\vec{x}_i, \vec{\Gamma}_i]^T$. Suppose that $\vec{x}_i(t) = \vec{f}_i(\vec{x}_i, \vec{0})$ and (5.13) have globally asymptotically stable equilibrium points at their respective origins. Then, the stability of the origin, i.e., $\vec{w} = \vec{0}$, of the cascade-augmented systems shown in figures 3.4.a and 3.4.c follows from the coming lemma where $\vec{\Gamma}_i$ is the input.

Lemma 5.1: (Lemma 4.7) [148] *Under the stated assumptions, if the system (5.12), with $\vec{\Gamma}_i$ as input, is input-to-state stable (ISS) and the origin of (5.13) is globally uniformly asymptotically stable, then the origin of the cascade system (5.12) and (5.13) is globally uniformly asymptotically stable.*

Proof: See [148] page 180. ■

Since (5.13) is in fact an autonomous system, we may use **Lemma 5.1** by focusing on the global asymptotic stability of the origin of (5.13). This is possible by following (3.83) through (3.86). Moreover, \vec{h}_i can be written as $\vec{h}_i(\vec{\Gamma}_i)$ to cover (5.7). According to the proposed framework, the command generator signal, i.e., Γ_i is or can be made bounded.

Consider for example the C-S model given in (3.116) whose value $\in (0, H/\delta^\beta]$, which means it is bounded even if the relative distance between states becomes unstable.

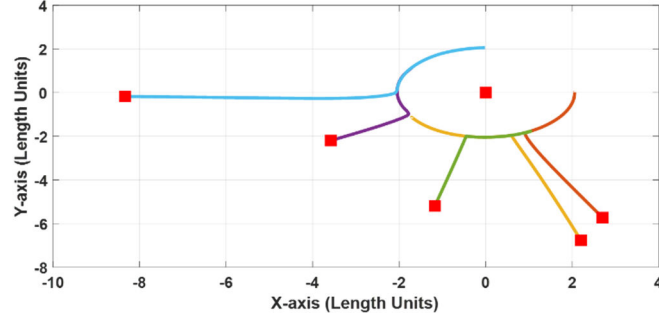
Proposition 5.1: *Considering the previous discussion, we may split the augmented system design process into two steps, namely: first designing ISS agent dynamics and then the bounded globally asymptotically stable trajectory-generator system.*

■

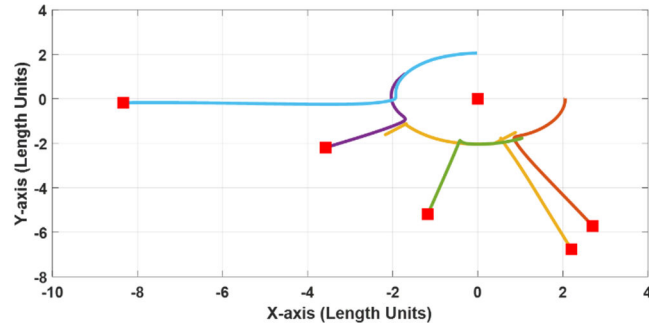
Figure 5.6 shows the result of six augmented systems that consist of agents' dynamics given by (5.1) under (5.2) and a trajectory-generator system given by (5.3) where the g_{ij} functions are given by (4.58) and (4.58.1). A disturbance was acting on agent 3 between 5-20 seconds. Note that the trajectory-generator system was unaware of this disturbance as can be read from Figure 5.6.a.

The difference between the responses of (5.7) and (5.9) is clearly shown in Figure 5.6. a and Figure 5.7.a. The tracking error while following the trajectory generated using (5.9) is depicted in Figure 5.8.

Remark 5.3: In figures 3.7.c and 3.7.d, the needed control signals reflect the difference between crafting the invariance, i.e. the behavior, intrinsically in the agent dynamics and following a behavior that is externally or internally generated, as it is the case in our proposed philosophy. In a future work, the former view point will be elaborated in more depth considering agents with non-trivial dynamics.



(a)



(b)

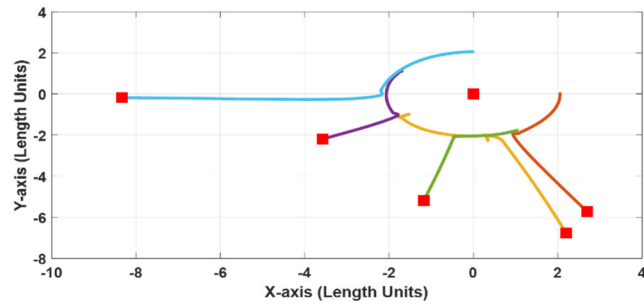
Figure. 5.6. The containment behavior of six nonidentical agents with scalar dynamics when $g_{ij}(\Gamma_i, \Gamma_j)$ under a disturbance acting on agent 3 between 5-20 seconds. (a) reference trajectory. (b) actual system response.

5.1.1.2: Stability of tip-to-tail connected systems

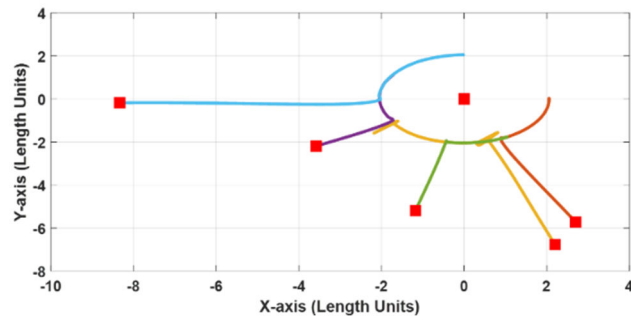
As stated before, when $g_{ij}(x_i, x_j)$ or $g_{ij}(x_i, \Gamma_j)$ is used, a loop is introduced. The resulting system under this interconnection could be viewed as an activator-inhibitor system [155] or sometime known as reaction-diffusion system. In this perspective, the trajectory-generator, i.e., the activator, stimulates its own production via autocatalysis as well as the production of the inhibitor, i.e., the agent dynamics. The inhibitor in turn suppresses the production of the activator [156].

The stability of this combined system is an immediate result of **Lemma 5.1**; since both systems are ISS. This can be seen by cutting the loop into two loops from tip-to-tail.

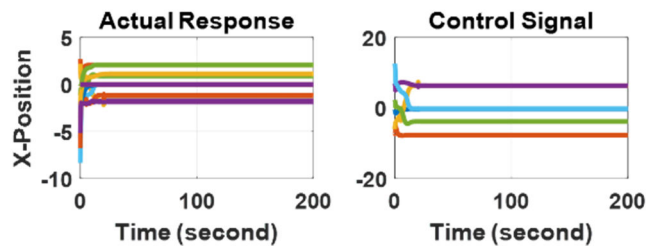
The simulation results obtained while achieving the containment behavior, explained in Chapter 3, using six nonidentical scalar dynamical systems are shown in figures 3.7 and 3.8. The distributed trajectory generation protocol was running over the network graph shown in Figure 4.47.



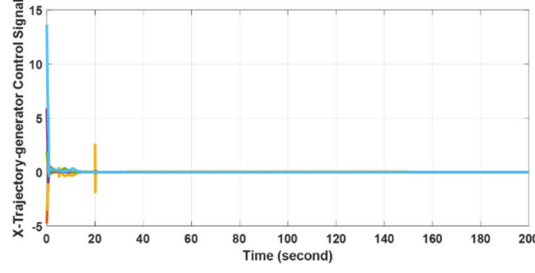
(a)



(b)



(c)



(d)

Figure. 5.7. The containment behavior of six nonidentical agents with scalar dynamics under a disturbance acting on agent 3 between 5-20 seconds: ideal trajectories denoted by solid lines and actual trajectories denoted by dashed lines, when (5.3) uses: (a)-(b) reference trajectory and actual system response when $g_{ij}(x, x_j)$. (c) actual systems position along the x-axis and the needed control signals when $g_{ij}(x_i, x_j)$. (d) control signal of the trajectory-generator along the x-direction when $g_{ij}(x_i, x_j)$.

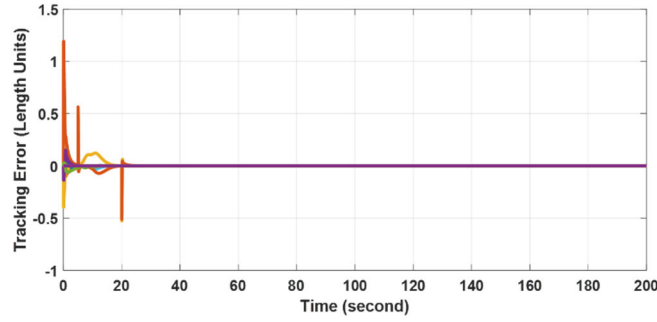


Figure. 5.8. The tracking error of all agents during the containment behavior using (5.9).

5.1.2: Effect of agents dynamics on the MAS connectivity

In Chapter 4, an edge contribution in the overall MAS connectivity was discussed, and the results obtained are still applicable under (5.7). However, under (5.9), the matter deserves investigation.

Following the same steps taken to derive **Theorem 4.6**, we may write the overall MAS connectivity as follows:

$$\lambda_2(\vec{x}) = \alpha \sum_{\{i,j\} \in \mathcal{E}} \lambda_2^{\{i,j\}} = \alpha \sum_{\{i,j\} \in \mathcal{E}} \frac{\partial f_i^x}{\partial \Gamma_i} (v_i - v_j)^2 \quad (5.15)$$

which is valid for general nonlinear protocols. If a semi-linear protocol is used, then (5.15) can be written as follows:

$$\lambda_2(\vec{x}) = \alpha \sum_{\{i,j\} \in \mathcal{E}} g_{ij}(x_i, x_j) (v_i - v_j)^2 \quad (5.16)$$

This leads us to the following theorem.

Theorem 5.1: For $g_{ij}(x_i, x_j) > 0 \forall x_i, x_j$, an edge $\{i, j\} \in \mathcal{E}$ will remain connected under (5.6) if and only if it is initially connected.

Proof: Obviously, when the edge is initially connected, we have $v_i - v_j \neq 0$. So, by having $g_{ij}(x_i, x_j) > 0 \forall x_i, x_j$, then $\lambda_2^{\{i,j\}} > 0$ and therefore the edge is connected $\forall x_i, x_j$. ■

Corollary 5.1: An initially connected edge will be disconnected when the diffusive weighting function $g_{ij}(x_i, x_j)$ approaches 0.

Proof: Clearly, when $g_{ij}(x_i, x_j) \rightarrow 0$, then $\lambda_2^{\{i,j\}} \rightarrow 0$ as well and the edge could be deemed disconnected. Consider for example the C-S model given in (3.116), if $x_j - x_i \rightarrow \infty$, then $g_{ij}(x_i, x_j) \rightarrow 0$. ■

Proposition 5.2: Under (5.6), a healthy link can be characterized according to a threshold percentage between the state-dependent edge-weight and its fixed weight in the fixed Laplacian matrix of the underlying communication graph.

Proof: Let the contribution of the edge $\{i, j\} \in \mathcal{E}$ in the fixed Laplacian matrix be $\lambda_{2f}^{\{i,j\}}$, and let the accepted change threshold be $\lambda_{2,TH}^{\{i,j\}} > 0$. Then, we may write the following:

$$\lambda_{2,TH}^{\{i,j\}} = \frac{\lambda_2^{\{i,j\}}}{\lambda_{2f}^{\{i,j\}}} = \frac{g_{ij}(x_i, x_j)(v_i - v_j)^2}{w_{ij}(v_i - v_j)^2} = \frac{g_{ij}(x_i, x_j)}{w_{ij}} > 0 \quad (5.17)$$

where: w_{ij} is the fixed edge-weight.

■

A connectivity-preserving protocol can be designed by taking the gradient of $\lambda_2(\vec{x})$. So, we have:

$$\frac{\partial \lambda_2(\vec{x})}{\partial x_i} = \alpha \sum_{\{i,j\} \in \mathcal{E}} \frac{\partial g_{ij}}{\partial x_i}(x_i, x_j)(v_i - v_j)^2 \quad (5.18)$$

Which is exactly similar to what was proposed in [145]. As a result, we may modify both (5.3) and (5.6) as follows:

$$\dot{\Gamma}_i = \alpha \sum_{j \in \mathcal{N}_i} g_{ij}(\Gamma_i, \Gamma_j) \{\Gamma_j - \Gamma_i\} + \frac{\partial \lambda_2^{\{i,j\}}}{\partial \Gamma_i} \quad (5.19)$$

$$\dot{\Gamma}_i = \alpha \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{\Gamma_j - \Gamma_i\} + \frac{\partial \lambda_2^{\{i,j\}}}{\partial x_i} \quad (5.20)$$

The effect of adding the connectivity-preserving protocol to the trajectory-generator at edge $\{1,2\} \in \mathcal{E}$ is depicted in Figure 5.9 assuming a consensus protocol is carried out in 2-D working space.

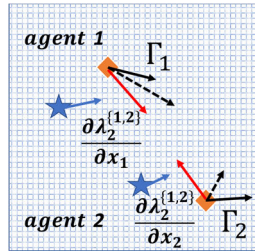


Figure. 5.9. A visualization of the effect of adding a connectivity-preserving controls to (5.6) in (red). Stars denote the actual positions of agents and actual directions of motion (blue), while squares denote the ideal trajectory to follow (black). The dashed lines are the resultants directions to follow.

Now, to see the effects of the agents dynamics on the edge connectivity under (5.6), we may simply take the change of $\lambda_2^{\{i,j\}}$ with respect to time, which yields:

$$\frac{d\lambda_2^{\{i,j\}}}{dt} = \alpha(v_i - v_j)^2 \left[\frac{\partial g_{ij}}{\partial x_i}, \frac{\partial g_{ij}}{\partial x_j} \right] \begin{bmatrix} \dot{x}_i \\ \dot{x}_j \end{bmatrix} \quad (5.21)$$

Let the g_{ij} functions satisfy (3.35), and the i^{th} agent closed-loop dynamics be given as follows:

$$\dot{x}_i(t) = f_i(x_i) + g_i(x_i)u_i(x_i, \Gamma_i) \in \mathfrak{R} \quad (5.22)$$

Then, the time-change in $\lambda_2^{\{i,j\}}$ can be given as follows:

$$\frac{d\lambda_2^{\{i,j\}}}{dt} = \alpha(v_i - v_j)^2 \frac{\partial g_{ij}}{\partial x_i} (\dot{x}_i - \dot{x}_j) \quad (5.23)$$

Which is valid even if the connected agents are heterogeneous. Note that (5.23) depends on the tracking controller used; for example, if an exact state feedback linearization is used as given in (4.60.13), then (5.23) will be given as follows:

$$\frac{d\lambda_2^{\{i,j\}}}{dt} = \alpha(v_i - v_j)^2 \frac{\partial g_{ij}}{\partial x_i} (\dot{\Gamma}_i - \dot{\Gamma}_j) \quad (5.23.1)$$

Considering (5.6), the time change in $\lambda_2^{\{i,j\}}$ depends also on neighbors' dynamics of both i^{th} and j^{th} agents. Note that the time change in $\lambda_2^{\{i,j\}}$ under (5.3) does not involve agents' dynamics at all.

B.1: Introducing the tracking error into the trajectory-generator kinematics as a multiplicative term

So far, we have not considered the tracking error, i.e., $e_i = \Gamma_i - x_i$, effect on the connectivity of agents. To do so, let the convergence rate $\alpha > 0 \in \mathfrak{R}$ be given as a function of the tracking error at the agent level. Thus, it can be given as follows:

$$\begin{aligned}\alpha_i &\equiv \alpha_i(e_i) > 0, \quad \forall i = 1, 2, \dots, N \\ \alpha_i(e_i(t_0)) &= \alpha, \forall i\end{aligned}\tag{5.23.2}$$

Previously, we used to have $\alpha_i = \alpha, \forall i, t \geq t_0$ as a real constant. Considering (5.21), we may rewrite (5.19) and (5.20) as follows- while neglecting the connectivity-preserving controls:

$$\dot{\Gamma}_i = \alpha_i(e_i) \sum_{j \in \mathcal{N}_i} g_{ij}(\Gamma_i, \Gamma_j) \{\Gamma_j - \Gamma_i\} \tag{5.23.3}$$

$$\dot{\Gamma}_i = \alpha_i(e_i) \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{\Gamma_j - \Gamma_i\} \tag{5.23.4}$$

Therefore, it is straightforward to modify (5.7) and (5.9) as follows:

$$\vec{\dot{\Gamma}} = -A(\vec{x}, \vec{\Gamma}) \mathcal{L}(\vec{\Gamma}) \vec{\Gamma} = \vec{f}^\Gamma \tag{5.24}$$

$$\vec{\dot{\Gamma}} = -A(\vec{x}, \vec{\Gamma}) \mathcal{L}(\vec{x}) \vec{\Gamma} = \vec{f}^x \tag{5.25}$$

where: $A(\vec{x}, \vec{\Gamma}) = \text{diag}([\alpha_i(e_i), \dots, \alpha_N(e_N)])$, $\alpha_i(e_i) > 0$ and $e_i = \Gamma_i - x_i$.

Let $\alpha_i(e_i)$ be given as follows- other forms are also possible:

$$\alpha_i(e_i) = \left(\frac{\beta_f}{\rho_i |e_i| + 1} \right), \quad \beta_f, \rho_i > 0 \tag{5.26}$$

Note the similarity between (5.23.3) and (4.35). Now, the effect of introducing the tracking error (5.26) on the edge connectivity under (5.23.3) using the parameterized C-S model (3.116) with $\beta = 0.5$ is stated without proof as follows- see (4.42) and replace $g(x_i)$ with $\alpha_i(e_i)$ given in (5.26):

$$\lambda_2^{\{i,j\}} = m_r^1 + m_r^2 + m_r^2 \tag{5.27}$$

where:

$$m_r^1 = c_{ij}^1 (v_i - v_j)^2$$

$$\begin{aligned}
m_r^2 &= c_{ij}^1 \{ \rho_j |e_j| v_i^2 - v_i v_j [\rho_i |e_i| + \rho_j |e_j|] + \rho_i |e_i| v_j^2 \} \\
m_r^3 &= c_{ij}^2 \{ \rho_j v_j^2 \alpha_j^2 \operatorname{sgn}(e_j) - \rho_i v_i^2 \alpha_i^2 \operatorname{sgn}(e_i) \} \\
c_{ij}^1 &= \frac{-\delta H \alpha_i \alpha_j}{\beta_f [\gamma (\Gamma_j - \Gamma_i)^2 + \delta]^{3/2}} \\
c_{ij}^2 &= \frac{H \{\Gamma_j - \Gamma_i\}}{\beta_f [\gamma (\Gamma_j - \Gamma_i)^2 + \delta]^{1/2}} \tag{5.27.1}
\end{aligned}$$

Similarly, the effect of introducing the tracking error (5.26) on the edge connectivity under (5.23.4) using a general g_{ij} weighting function is stated without proof as follows- see (4.42) and replace $g(x_i)$ with $\alpha_i(e_i)$ given in (5.26):

$$\lambda_2^{\{i,j\}} = m_r^1 + m_r^2 + m_r^3 \tag{5.28}$$

where:

$$\begin{aligned}
m_r^1 &= c_{ij}^3 (v_i - v_j)^2 \\
m_r^2 &= c_{ij}^3 \{ \rho_j |e_j| v_i^2 - v_i v_j [\rho_i |e_i| + \rho_j |e_j|] + \rho_i |e_i| v_j^2 \} \\
m_r^3 &= c_{ij}^4 \{ \rho_j v_j^2 \alpha_j^2 \operatorname{sgn}(e_j) - \rho_i v_i^2 \alpha_i^2 \operatorname{sgn}(e_i) \} \\
c_{ij}^3 &= \frac{-\alpha_i \alpha_j}{\beta_f} g_{ij}(x_i, x_j) \\
c_{ij}^4 &= \frac{\{\Gamma_j - \Gamma_i\}}{\beta_f} g_{ij}(x_i, x_j) \tag{5.28.1}
\end{aligned}$$

In (5.28.1), c_{ij}^4 cannot be bounded using $g_{ij}(x_i, x_j)$. On contrary, c_{ij}^2 in (5.27.1) is bounded and is equal to $H/(\beta_f \sqrt{\gamma})$ as explained in (3.117). In general, $\lambda_2^{\{i,j\}}$ in (5.27) can be made negative for all Γ_i, Γ_j and bounded e_i, e_j by selecting ρ_i sufficiently small. Therefore, the trajectory-generator under (5.22) will be mostly used in this investigation.

In the literature, there are many methods by which guaranteed tracking performance can be achieved. For example, in [157] dynamics surface control (DSC) method was used in conjunction with neural networks to guarantee the L_∞ tracking performance for a class of uncertain nonlinear systems under output feedback. In [158] asymptotic tracking for a class of nonlinear systems, which globally can be transformed into systems in generalized output feedback canonical forms, is presented. Note that in [157], the initial value of the tracking error is set to zero; so that the L_∞ norm of the tracking error can be guaranteed, i.e., $|e_i|_\infty \leq \zeta$, $\zeta \geq 0$. This is like what is depicted in Figure 5.1.

B.2: The effect of introducing the tracking error- as a multiplicative term- on the invariant quantity

Clearly, when $\rho_i|e_i| = 0$, then the average value, i.e., the arithmetic mean, is achieved.

When $\rho_i|e_i| \neq 0$, then a deviation from the average value starts taking place. This is stated in the following theorem.

Theorem 5.2: *The invariant quantity, i.e., the arithmetic mean, of a MAS under (5.23.3) and (5.23.4) with bounded tracking errors $|e_i| \leq \zeta$, $\zeta \geq 0$, is asymptotically achieved. Moreover, for the same ζ , the rate of change from the invariant quantity is upper bounded by the ideal-states' instantaneous weighted average.*

Proof: Let the following approximate the arithmetic mean μ_a under (5.23.3) and (5.23.4):

$$\mu_a(t) = \frac{1}{N} \sum_{i=1}^N \alpha_i^{-1}(e_i) \Gamma_i(t), \forall t \geq t_0 \quad (5.29)$$

Taking the derivative of (4.29) with respect to time, yields:

$$\frac{d\mu_a(t)}{dt} = \frac{1}{N} \frac{d}{dt} \sum_{i=1}^N \alpha_i^{-1}(e_i) \Gamma_i(t) = \frac{1}{N} \sum_{i=1}^N \left\{ \frac{d\alpha_i^{-1}}{dt} \Gamma_i + \alpha_i^{-1} \frac{d\Gamma_i}{dt} \right\} \quad (5.29.1)$$

Using (5.23.3) and (5.26), it is straightforward to rewrite (4.29.1) as follows:

$$\frac{d\mu_a(t)}{dt} = -\frac{\vec{1}^T}{N} \left\{ \mathcal{L}(\vec{\Gamma}) + \frac{d}{dt} A^{-1}(\vec{x}, \vec{r}) \right\} \vec{\Gamma} \quad (5.29.2)$$

where: $\frac{d}{dt} A^{-1}(\vec{x}, \vec{r}) = \text{diag} \left(\left[\frac{d\alpha_i^{-1}}{dt}, \dots, \frac{d\alpha_N^{-1}}{dt} \right] \right)$.

Since $\vec{1}^T \mathcal{L}(\vec{\Gamma}) \vec{\Gamma} = 0, \forall \vec{\Gamma}$, we are left with:

$$\frac{d\mu_a(t)}{dt} = -\frac{\vec{1}^T}{N} \frac{d}{dt} A^{-1}(\vec{x}, \vec{r}) \vec{\Gamma} \quad (5.29.3)$$

Which can be written in element-wise notation as follows:

$$\frac{d\mu_a(t)}{dt} = -\frac{1}{N} \sum_{i=1}^N (\rho_i \dot{e}_i \text{sgn}(e_i)) \Gamma_i \quad (5.29.4)$$

Let $|\dot{e}_i| \leq \eta_i$ be guaranteed by the tracking controller for some sufficiently large constants $\eta_i \geq 0$, then:

$$\left| \frac{d\mu_a(t)}{dt} \right| \leq \frac{1}{N} \sum_{i=1}^N \{\rho_i \eta_i\} \Gamma_i(t) \quad (5.29.5)$$

Which is upper bounded by the ideal-states' instantaneous weighted average. Note that if $\rho_i = 0$, then the average consensus will be achieved.

■

Table 5.1 and Table 5.2 display the consensus behavior results in 3-D of six nonidentical scalar dynamical systems over the network (**N.**) shown in Figure 2.36 (**#1**) and an undirected 6-cycle (**#2**). The trajectory-generator systems (5.23.3) and (5.23.4) were used with different values of ρ_i . The g_{ij} functions utilized the parameterized C-S model (3.116) with $\beta = 0.5$. The systems were controlled using (5.2) and therefore with no specific bounds on the tracking errors being imposed. A disturbance was acting on agent 3 between 5-20 seconds. All results were generated using the same simulation time and initial conditions whose average values are $\{-1.1875, 1.8872, -1.0266\}$. The actual MAS

consensus value is denoted by $\{\bar{x}, \bar{y}, \bar{z}\}$ and the trajectory-generator achieved consensus value is denoted by $\{\bar{\Gamma}_x, \bar{\Gamma}_y, \bar{\Gamma}_z\}$.

Clearly, from tables 5.1 and 5.2, we may deduce that the best response is obtained when $\rho_i \rightarrow 0 \forall i$ and the tracking responsibility is left for the local controller. So, all the previous results obtained in the previous parts are still valid without any modification and now we are in position to propose more sophisticated behaviors.

Table 5. 1: Consensus Tracking using (5.23.3) when $k_2 = 20$.

ρ	0.01		1		100	
N.	#1	#2	#1	#2	#1	#2
\bar{x}	-1.1848	-1.1842	-0.9283	-0.8776	0.2050	0.3332
\bar{y}	1.8890	1.8881	2.0775	2.0273	3.6670	3.5641
\bar{z}	-1.0265	-1.0264	-1.0189	-1.0100	-0.8434	-0.8044
$\bar{\Gamma}_x$	-1.1848	-1.1843	-0.9284	-0.8777	0.2053	0.3332
$\bar{\Gamma}_y$	1.8890	1.8881	2.0775	2.0272	3.6673	3.5641
$\bar{\Gamma}_z$	-1.0265	-1.0264	-1.0189	-1.0100	-0.8434	-0.8044

Table 5. 2: Consensus Tracking using (5.23.4) when $k_2 = 20$.

ρ	0.01		1		100	
$N.$	#1	#2	#1	#2	#1	#2
\bar{x}	-1.1848	-1.1844	-0.9266	-0.8830	0.2022	0.3349
\bar{y}	1.8890	1.8880	2.0803	2.0267	3.6710	3.5646
\bar{z}	-1.0265	-1.0264	-1.0188	-1.0100	-0.8434	-0.8043
$\bar{\Gamma}_x$	-1.1848	-1.1844	-0.9266	-0.8827	0.2025	0.3350
$\bar{\Gamma}_y$	1.8891	1.8881	2.0803	2.0271	3.6713	3.5647
$\bar{\Gamma}_z$	-1.0265	-1.0264	-1.0188	-1.0100	-0.8434	-0.8043

In the previous discussion, we have seen that entering the self-awareness as a multiplicative variable might not be the best idea especially from the connectivity-preserving perspective; however, it helped in slowing down the evolution of the trajectory-generator systems and resulted in a reduced tracking error.

B.3: Introducing the tracking error into the trajectory-generator kinematics as an additive term

Still, the self-awareness can be introduced to the trajectory-generator system as an additive instead of multiplicative disturbance. As a result, both (5.23.3) and (5.23.4) can be modified as follows:

$$\dot{\Gamma}_i = \beta \sum_{j \in \mathcal{N}_i} g_{ij}(\Gamma_i, \Gamma_j) \{\Gamma_j - \Gamma_i\} + \alpha_i(e_i) \quad (5.30)$$

$$\dot{\Gamma}_i = \beta \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{\Gamma_j - \Gamma_i\} + \alpha_i(e_i) \quad (5.31)$$

where: $\beta > 0$ denotes the convergence rate. Clearly, in both (5.30) and (5.31), the $\alpha_i(e_i)$ is acting as a self-pinning gain once viewed from a leader-follower perspective. Under this setup, the advantages of (5.31) over (5.30) are irrefutable; simply because (5.31) offers explicitly the self-awareness along with neighbors-awareness. Also, from a connectivity preservation perspective, (5.31) is straightforward as given in (5.16). Proving the invariance properties under (5.30) and (5.31) is easy when $\alpha_i(e_i)$ is a non-persistent disturbance.

Let $\alpha_i(e_i)$ be given as follows- utilizing the C-S model:

$$\alpha_i(e_i) = \frac{\beta_f e_i}{[\rho_i e_i^2 + 1]^{\beta_e}} \quad (5.32)$$

where: $e_i = x_i - \Gamma_i$, $\beta_e > 0$. Recalling (3.117), when $\beta_e = 0.5$, then $|\alpha_i(e_i)| \leq \beta_f / \sqrt{\rho_i}$. Note that other forms of α_i can also be used, however, (5.32) was suggested because of the boundedness of the error term when $\beta_e = 1/2$.

At this point, it is good to consider the situation when the i^{th} agent is pinned to M external leaders- other than the self. So, we may rewrite (5.30) and (5.31) as follows:

$$\dot{\Gamma}_i = \beta \sum_{j \in \mathcal{N}_i} g_{ij}(\Gamma_i, \Gamma_j) \{\Gamma_j - \Gamma_i\} + \alpha_i(e_i) + \sum_{n=1}^M l_i^n \{x_n^l - \Gamma_i\} \quad (5.33)$$

$$\dot{\Gamma}_i = \beta \sum_{j \in \mathcal{N}_i} g_{ij}(x_i, x_j) \{\Gamma_j - \Gamma_i\} + \alpha_i(e_i) + \sum_{n=1}^M l_i^n \{x_n^l - \Gamma_i\} \quad (5.34)$$

where: l_i^n is the pinning gain of the i^{th} agent to the n^{th} leader and it can be given as in (4.29.4) or (4.33).

Interestingly, (5.33) and (5.34) can be used to realize *Node-to-Node* behaviors as can be found in [159], [160], for examples. Figure 5.10 shows one realization of such a concept in view of the current investigation.

Considering (5.32), the results of a leaderless consensus behavior under (5.33) and (5.34) are shown in Table 5.3 where the local controller is given by (5.2). The same simulation setup used to generate the results displayed in tables 5.1 and 5.2 is used to fill Table 5.3. Note that $k_{2i} = k_2, \forall i$. Both the actual MAS consensus value and the trajectory-generator achieved consensus value were identical. The average values of the initial conditions are $\{-1.1875, 1.8872, -1.0266\}$. Since the results obtained using (5.34) were identical to those obtained using (5.33), only the latter are reported here.

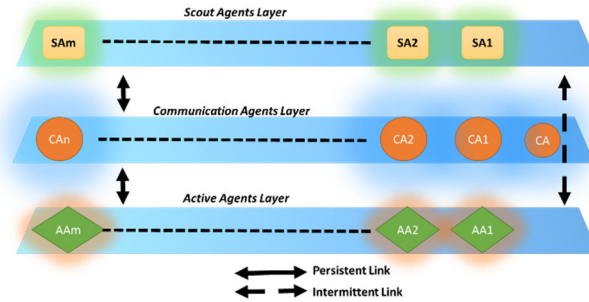


Figure. 5.10. A realization of the concept of Node-to-Node behavior in view of the current investigation. SA denotes the scout agents. CA denotes the communication agents. AA denotes the active agents.

Table 5. 3:Consensus Tracking using (5.33).

k_2	50		100		200	
$N.$	#1	#2	#1	#2	#1	#2
\bar{x}	-1.1787	-1.1788	-1.1836	-1.1836	-1.1853	-1.1854
\bar{y}	1.8736	1.8735	1.8799	1.8799	1.8838	1.8837
\bar{z}	-1.0191	-1.0191	-1.0229	-1.0229	-1.0247	-1.0247

As can be seen from Table 5.3, the average consensus under (5.30) and (5.31) is achieved asymptotically with a bounded error directly proportional to that tracking error. Notice that

in (5.2), the tracking error is directly proportional to the used value of k_{2i} in the PD-controller at the agent level.

To conclude this section, using a trajectory-generator co-system and local stabilizing and tracking controllers simplify the problem to the behavior engineers such that the controllers take care of the underlying dynamical nature of the agent while the trajectory-generator is acting like the brain. So, we will continue in presenting other dynamical behaviors needed in many applications.

5.2 Kino-dynamics of Trajectory-generator systems

In this section, it is intended to generate higher-order trajectory-related signals. Such signals are usually needed when controlling the motion of dynamical systems, in general. It is true that some slowly-moving dynamical systems, say ground robots, may interact with other dynamical systems, say aerial vehicles, under some applications. Such interaction yields dynamical systems with different time-scales in which position, i.e., the zero-derivative with respect to time, alone is not sufficient. Therefore, more information about the dynamics, or specifically, the kinematics, of this interaction is needed beyond velocity and acceleration including jerk, snap and higher derivatives [161]. The dynamics and kinematics of robots motion are combined to what is known as Kino-Dynamics [162]–[164].

5.2.1: Introducing acceleration signals into the trajectory-generator kinematics

By taking the needed number of time derivatives of both (5.33) and (5.34), we may formulate the kinematics of trajectory-generator systems, and most importantly provide a window to introduce explicitly the agents' dynamics into the making of the trajectory signals. This in fact will enable us to introduce models of the involved agents especially if

they are uncertain, which will make designing robust trajectory-signals and dynamically-friendly trajectories feasible.

It is well-known in the art that most of the dynamical systems can be approximated as being second-order dynamical nature. Therefore, the acceleration signal is needed. Taking one more time derivative of (5.34) yields the following:

$$\begin{aligned}\ddot{\Gamma}_i = & \beta \sum_{j \in \mathcal{N}_i} \{\Gamma_j - \Gamma_i\} L_{\vec{f}_{ij}} g_{ij} + g_{ij} \{\dot{\Gamma}_j - \dot{\Gamma}_i\} + \nabla \alpha_i(e_i) \begin{bmatrix} \dot{\Gamma}_i \\ \dot{x}_i \end{bmatrix} \\ & + \sum_{n=1}^M \nabla l_i^n \begin{bmatrix} \dot{x}_n^l \\ \dot{\Gamma}_i \end{bmatrix} \{x_n^l - \Gamma_i\} + l_i^n \{\dot{x}_n^l - \dot{\Gamma}_i\}\end{aligned}\quad (5.35)$$

where: $L_{\vec{f}_{ij}}$ is the directional derivative, and $\vec{f}_{ij} = [\dot{\Gamma}_i, \dot{\Gamma}_j]^T$. Doing the same for all agents, yields:

$$\ddot{\vec{\Gamma}} = -\left\{\beta \mathcal{L}(\vec{\Gamma}, \vec{\Gamma}) + P_1(\cdot)\right\} \ddot{\vec{\Gamma}} - \left\{\beta \mathcal{L}(\vec{\Gamma}) - \Xi_1(\vec{\Gamma}, \vec{x}) + P_2(\cdot)\right\} \ddot{\vec{\Gamma}} + \Xi_2(\vec{\Gamma}, \vec{x}) \ddot{\vec{x}} + \Psi_1(\cdot) \ddot{x}^l + \Psi_2(\cdot) \ddot{x}^l \quad (5.36)$$

where: \mathcal{L} is the state-dependent Laplacian matrix, and:

$$P_1(\cdot) = \text{diag} \left(\left[\sum_{n=1}^M \nabla l_1^n \begin{bmatrix} \dot{x}_n^l \\ \dot{\Gamma}_1 \end{bmatrix}, \dots, \sum_{n=1}^M \nabla l_N^n \begin{bmatrix} \dot{x}_n^l \\ \dot{\Gamma}_N \end{bmatrix} \right] \right)$$

$$P_2(\cdot) = \text{diag} \left(\left[\sum_{n=1}^M l_1^n, \dots, \sum_{n=1}^M l_N^n \right] \right)$$

$$\Xi_1(\cdot) = \text{diag} \left(\left[\frac{\partial \alpha_1}{\partial \Gamma_1}, \dots, \frac{\partial \alpha_N}{\partial \Gamma_N} \right] \right)$$

$$\Xi_2(\cdot) = \text{diag} \left(\left[\frac{\partial \alpha_1}{\partial x_1}, \dots, \frac{\partial \alpha_N}{\partial x_N} \right] \right)$$

$$\begin{aligned}
\Psi_1(\cdot) &= \begin{bmatrix} \nabla l_1^1 \begin{bmatrix} \dot{x}_1^l \\ \dot{\Gamma}_1 \end{bmatrix} & \cdots & \nabla l_1^M \begin{bmatrix} \dot{x}_M^l \\ \dot{\Gamma}_1 \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \nabla l_N^1 \begin{bmatrix} \dot{x}_1^l \\ \dot{\Gamma}_N \end{bmatrix} & \cdots & \nabla l_N^M \begin{bmatrix} \dot{x}_M^l \\ \dot{\Gamma}_N \end{bmatrix} \end{bmatrix} \\
\Psi_2(\cdot) &= \begin{bmatrix} l_1^1 & \cdots & l_1^M \\ \vdots & \ddots & \vdots \\ l_N^1 & \cdots & l_N^M \end{bmatrix} \\
\vec{x}^l &= [x_1^l, \dots, x_M^l]^T
\end{aligned} \tag{5.36.1}$$

Similarly, we may write the overall trajectory-generator systems state under (5.33) as follows:

$$\vec{\Gamma} = -\{\beta\mathcal{L}(\vec{\Gamma}) + P_2(\cdot)\}\vec{\Gamma} + A(\vec{\Gamma}, \vec{x}) + \Psi_2(\cdot)\vec{x}^l \tag{5.37}$$

where: $A(\vec{\Gamma}, \vec{x}) = [\alpha_1(e_1), \dots, \alpha_N(e_N)]^T$. If $\alpha_i(e_i)$ is given by (5.32), then we may rewrite (5.37) as follows:

$$\vec{\Gamma} = -\{\beta\mathcal{L}(\vec{\Gamma}) + \tilde{A}(\vec{\Gamma}, \vec{x}) + P_2(\cdot)\}\vec{\Gamma} + \tilde{A}(\vec{\Gamma}, \vec{x})\vec{x} + \Psi_2(\cdot)\vec{x}^l \tag{5.38}$$

$$\text{where: } \tilde{A}(\vec{\Gamma}, \vec{x}) = \text{diag} \left(\left[\frac{\beta_f}{[\rho_1 e_1^2 + 1]^{\beta_e}}, \dots, \frac{\beta_f}{[\rho_N e_N^2 + 1]^{\beta_e}} \right] \right).$$

Consequently, the overall trajectory-generator state-space model can be given as follows:

$$\begin{bmatrix} \vec{\Gamma} \\ \vec{\Gamma} \end{bmatrix} = \begin{bmatrix} \Pi_1 & [0] \\ \Pi_2 & \Pi_3 \end{bmatrix} \begin{bmatrix} \vec{\Gamma} \\ \vec{\Gamma} \end{bmatrix} + \begin{bmatrix} \tilde{A} & [0] \\ [0] & \Xi_2 \end{bmatrix} \begin{bmatrix} \vec{\Gamma} \\ \vec{\Gamma} \end{bmatrix} + \begin{bmatrix} \Psi_2 & [0] \\ \Psi_1 & \Psi_2 \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{x}^l \end{bmatrix} \tag{5.39}$$

$$\begin{bmatrix} \vec{\Gamma} \\ \vec{\Gamma} \end{bmatrix} = \Sigma_s \begin{bmatrix} \vec{\Gamma} \\ \vec{\Gamma} \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{5.39.1}$$

where: $[0]$ is a block of $N \times N$ zeros, Π_1, Π_2 and Π_3 can be found from there corresponding equations.

Remark 5.4: If the tracking error, i.e., $|e_i| = |\Gamma_i - x_i| = 0 \forall i$, then both Π_1 and Π_3 are equal.

The stability of (5.39) is ensured by the following theorem:

Theorem 5.3: *The trajectory-generator systems given in (5.39) is stable if and only if the agents dynamics under their corresponding tracking controllers are stable and leaders signals are always bounded.*

Proof: When at least one agent is connected to a leader, then both Π_1 and Π_3 are always negative nonsingular M -matrices, similarly will be their Jacobians. Moreover, when agents are stable under their corresponding tracking controllers, then the tracking errors will be bounded and both Π_1 and Π_3 will be equal if these errors are zeros.

When the leaders' signals, namely: \vec{x}^l and $\dot{\vec{x}}^l$ are bounded, then $P_2(\cdot)$ in (5.36.1) will be bounded. Consequently, Π_2 in (5.39) will be bounded which is a sufficient condition for the stability of the system matrix in (5.39) when the Jacobians of both Π_1 and Π_3 are negative [4] which is indeed the case.

To see it clearly, let us define the following transformation:

$$\Phi = \begin{bmatrix} I & [0] \\ [0] & \epsilon I \end{bmatrix}, \quad \epsilon \in \Re \quad (5.39.2)$$

Pre-multiplying (5.39.1) by Φ and post-multiplying it by Φ^{-1} , yields:

$$\begin{bmatrix} I & [0] \\ [0] & \epsilon I \end{bmatrix} \begin{bmatrix} \Pi_1 & [0] \\ \Pi_2 & \Pi_3 \end{bmatrix} \begin{bmatrix} I & [0] \\ [0] & \epsilon^{-1} I \end{bmatrix} = \begin{bmatrix} \Pi_1 & [0] \\ \epsilon \Pi_2 & \Pi_3 \end{bmatrix} \quad (5.39.3)$$

Now, since both Π_1 and Π_2 are the negatives of nonsingular M -matrices and Π_2 is bounded, then (5.39.3) is negative definite for sufficiently small ϵ .

When leaders do exist, then the stability of (5.39) can be tackled using the contraction analysis. In Chapter 3, specifically in (3.107) and the discussion that followed, we have

shown the relation between the semi-linear protocols per the proposed framework and the potential functions. According to [165], the dynamics of a system utilizing gradient decent method is contracting if the associated cost function is strictly convex, i.e., the system:

$$\dot{\vec{x}} = -\frac{\partial V}{\partial \vec{\Gamma}} \quad (5.39.4)$$

has contracting dynamics whenever:

$$\frac{\partial^2 V}{\partial \vec{\Gamma}^2} > 0 \quad (5.39.5)$$

is valid uniformly. This can be seen from our previous discussion in (3.107), see also Figure 2.25. So, we may consider Σ_s in (5.39.1)- from contraction analysis point of view- as the Jacobian of the following dynamical system [166]:

$$\frac{d}{dt} \begin{bmatrix} \delta z_1 \\ \delta z_2 \end{bmatrix} = \begin{bmatrix} \Pi_1 & [0] \\ \Pi_2 & \Pi_3 \end{bmatrix} \begin{bmatrix} \delta z_1 \\ \delta z_2 \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad (5.39.6)$$

where: d_1 and d_2 are disturbances that can be formulated from the terms appearing in (5.39). So, the relative velocities between a desired trajectory c_1 and a system trajectory c_2 verify:

$$\frac{d}{dt} \int_{c_1}^{c_2} \|\delta z_1\| + |\lambda_{1,max}| \int_{c_1}^{c_2} \|\delta z_1\| \leq \|d_1\| \quad (5.39.7)$$

$$\frac{d}{dt} \int_{c_1}^{c_2} \|\delta z_2\| + |\lambda_{2,max}| \int_{c_1}^{c_2} \|\delta z_2\| \leq \|d_2\| + \int_{c_1}^{c_2} \Pi_2 \delta z_1 \quad (5.39.8)$$

where: $\lambda_{i,max}$ is the largest eigenvalue of $\Pi_i, i = 1, 2$.

When both d_1 and d_2 are bounded, then exponential convergence to the desired trajectory is asymptotically achieved. The boundedness of Π_2 can be seen from (5.39.3).

■

Under (5.39), the invariance properties, or behaviors in general, embedded to the kinematical trajectory-generators (5.33) and (5.34) are still preserved. This is stated in the following theorem.

Theorem 5.4: *The behaviors realized using the kinematical trajectory-generators (5.33) can be asymptotically and dynamically realized with q -order systems using (5.39).*

Proof: In a leaderless situation, Σ_s given in (5.39.3) still have both $\vec{v}^T = \vec{1}^T$ and $\vec{e} = \vec{1}$ as nonlinear left and right eigenvectors associated with $\lambda_1(\vec{\Gamma}) = 0 \forall \vec{\Gamma}$. Let the average value, i.e., μ_a , be the invariant quantity of concern. To test its kinematics under (5.39), we need to take its derivative with respect to time twice. This is demonstrated as follows, let:

$$\mu_a(t) = \frac{1}{N} \sum_{i=1}^N \Gamma_i(t), t \geq t_0 \quad (5.40)$$

Taking the derivatives with respect to time, yields:

$$\frac{d\mu_a}{dt} = \frac{1}{N} \sum_{i=1}^N \frac{d\Gamma_i}{dt} \quad (5.40.1)$$

$$\frac{d^2\mu_a}{dt^2} = \frac{1}{N} \sum_{i=1}^N \frac{d^2\Gamma_i}{dt^2} \quad (5.40.2)$$

Equating both time derivatives to zero, yields:

$$\frac{d\mu_a}{dt} = \frac{\beta}{N} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} g_{ij}(\Gamma_i, \Gamma_j) \{\Gamma_j - \Gamma_i\} = 0 \quad (5.40.3)$$

$$\frac{d^2\mu_a}{dt^2} = \frac{\beta}{N} \sum_{i=1}^N \left(\frac{\partial g_{ij}}{\partial x_i} \dot{\Gamma}_i + \frac{\partial g_{ij}}{\partial x_j} \dot{\Gamma}_j \right) \{\Gamma_j - \Gamma_i\} + g_{ij} \{\dot{\Gamma}_j - \dot{\Gamma}_i\} = 0 \quad (5.40.4)$$

As was shown in (3.11), we know that (5.40.3) is satisfied. As for (5.40.4), by recalling (3.12), (3.34) and (3.35), it will be straightforward to show that (5.40.3) is also true; since:

$$\frac{\partial g_{ij}}{\partial \Gamma_i} \dot{\Gamma}_i + \frac{\partial g_{ij}}{\partial \Gamma_j} \dot{\Gamma}_j = \frac{\partial g_{ji}}{\partial \Gamma_i} \dot{\Gamma}_i + \frac{\partial g_{ji}}{\partial \Gamma_j} \dot{\Gamma}_j \quad (5.40.5)$$

So, the average value is achieved as $t \rightarrow \infty$.

Recalling **Theorem 5.2**, then the average value under leaders' influence will be asymptotically and dynamically achieved under (5.39).

In general, classical graph algorithms facilitates combining several systems that can be systematically decomposed into sub systems under some hierarchies where input-equivalence, i.e., invariance, can be verified top-down [4]. Therefore, (5.39) can be extended to work with q -order systems.

■

Likewise, one can find the kinematical version of (5.34) by following the previously stated steps used when deriving (5.39). This simply can be done by replacing $g_{ij}(\Gamma_i, \Gamma_j)$ with $g_{ij}(x_i, x_j)$ and $\dot{\Gamma}_{i,j}$ with $\dot{x}_{i,j}$ in (5.40.4).

Remark 5.5: The kinematical part of (5.39) is dealing explicitly with the dynamics of a general scalar nonlinear system (5.22) as a disturbance. While in the system resulting from (5.34) after introducing the acceleration, the dynamics given in (5.22) appears explicitly as a disturbance and implicitly as a weighting factor in the used g_{ij} functions.

Remark 5.6: According to **Theorem 5.4**, we may extend the Kino-dynamical features of the proposed trajectory-generator systems presented in this subsection to work with realistic robots whose dynamics are of higher-order.

Remark 5.7: Recalling Chapter 3, and 4, all the developed behaviors embedded in the g_{ij} functions can be used in the Kino-dynamical trajectory-generator system presented in this subsection, however, a slight modification might be needed.

Remark 5.8: The trajectory-generator systems in the other two spatial directions can be found similarly.

5.2.2: Designing harmonic potential fields using the proposed framework

In this subsection, the utilization of the proposed framework in designing the navigation and planning signals is briefly demonstrated. By and large, the use of potential fields and more specifically the harmonic potential fields (HPF) has offered enormous advantages in the field of planning for rigid robots. Being an excellent goal-seeking planner, the HPF approach induces guidance vectors in a dense collection of guidance vectors covering the robot admissible space (Ω) [167]. The smooth induced guidance vectors do avoid being trapped in local minima usually encountered in ordinary potential fields.

The HPF is a solution of the Laplace's equation which is considered as a simple example of elliptic PDEs. Recalling (3.115), the Laplace's equation is obtained when the characteristic of (3.115) given by (3.115.1) is positive definite.

In Chapter 3, specifically in section I, we have shown how the vector calculus can be made useful in designing, mainly, the consensus protocols. Once more, we will use the vector calculus to facilitate our findings in this section.

Let us start with a semi-linear protocol in 1-D working space that is related to a conservative potential field p . The protocol is then given as follows:

$$\begin{bmatrix} \dot{\Gamma}_i \\ \dot{\Gamma}_j \end{bmatrix} = -\frac{\partial p(\Gamma_i, \Gamma_j)}{\partial \vec{\Gamma}} = -\nabla p = \vec{f} \quad (5.41)$$

To have a HPF, the divergence of the vector field must be zero, i.e., we need to have the following

$$-\nabla \cdot \nabla p = -\nabla^2 p = -\frac{\partial^2 p}{\partial \vec{\Gamma}^2} \cdot \vec{1} = 0 \quad (5.41.1)$$

where: ∇^2 is the Laplacian operator. From a vector calculus perspective, (5.41.1) is equivalent to having a divergence-free vector field, and because of having \vec{f} as a

conservative field, the curl of that field will also be zero. So, a HPF is a vector field that is both curl-free and divergence-free.

According to the proposed framework, we have shown the needed conditions to have a conservative field, i.e., curl-free, by forcing (3.34) and (3.35), simultaneously. Therefore, the invariant quantity is preserved in the whole domain of interest. So, now we also need to force another condition on the designed protocols, namely: to produce a divergence-free vector field.

Recalling (4.36.3), and working mainly on one-edge MAS with two connected agents in 1-D, the divergence associated with the designed vector field is given as follows:

$$\text{div}(\vec{f}) = \frac{\partial g_{12}}{\partial \Gamma_1}(\Gamma_2 - \Gamma_1) - g_{12} + \frac{\partial g_{21}}{\partial \Gamma_2}(\Gamma_1 - \Gamma_2) - g_{21} = 0 \quad (5.41.2)$$

which is exactly the trace of the Jacobian in (4.36.3). The curl of a vector field is zero, i.e., that vector field is curl-less, if and only if it is continuous over a simple connected domain. This will yield a symmetric Jacobian. According to the proposed framework, we need the following to be true as well, see (3.109.1):

$$\text{curl}(\vec{f}) = \frac{\partial g_{21}}{\partial \Gamma_1}(\Gamma_1 - \Gamma_2) + g_{21} - \frac{\partial g_{12}}{\partial \Gamma_2}(\Gamma_2 - \Gamma_1) - g_{12} = 0 \quad (5.41.3)$$

Adding both (5.41.2) and (5.41.3), yields the following PDE:

$$\left\{ \frac{\partial g_{21}}{\partial \Gamma_1} + \frac{\partial g_{12}}{\partial \Gamma_2} - \frac{\partial g_{12}}{\partial \Gamma_1} + \frac{\partial g_{21}}{\partial \Gamma_2} \right\} (\Gamma_1 - \Gamma_2) = 2g_{12} \quad (5.41.4)$$

Or after imposing (3.34) and (3.35):

$$\frac{\partial g_{12}}{\partial \Gamma_1} \{\Gamma_2 - \Gamma_1\} = g_{12} \quad (5.41.5)$$

Remark 5.9: From connectivity-preserving perspective, a HPF acting alone on an edge makes it disconnected, i.e., $\lambda_2^{\{i,j\}}(\vec{\Gamma}) = 0$, see (4.37).

One may verify that the following is a candidate- if not the only- solution of (5.41.5) in 1-D:

$$g_{ij} = \frac{H}{(\Gamma_j - \Gamma_i)}, \quad H \neq 0 \quad (5.41.6)$$

Substituting (5.41.6) into the semi-linear protocol, yields:

$$\begin{bmatrix} \dot{\Gamma}_i \\ \dot{\Gamma}_j \end{bmatrix} = \begin{bmatrix} H \\ -H \end{bmatrix} = -\nabla p = \vec{f} \quad (5.41.7)$$

Which makes Γ_1 unbounded, when $H > 0$. So, we need to consider higher dimensions to facilitate building the needed HPFs, where things become more interesting and involved.

Remark 5.10: From now on, we need to introduce superscripts on the ideal trajectory signals Γ to denote the spatial directions in the working space, i.e., $\Gamma^{x,y,z}$.

Definition 5.1: In robotics applications, a working space can be defined as the space where the physical action is projected or realized. On contrary, a configuration space can be defined as the space where the action is planned. Both spaces could be of equal dimensions. In a 2-D working space, we need to write the trajectory-generator kinematics in 2-D and to introduce coupling in the two directions, as follows:

$$\begin{aligned} \begin{bmatrix} \dot{\Gamma}_i^x \\ \dot{\Gamma}_j^x \end{bmatrix} &= \begin{bmatrix} g_{12}(\vec{\eta})\{\Gamma_j^x - \Gamma_i^x\} \\ g_{21}(\vec{\eta})\{\Gamma_i^x - \Gamma_j^x\} \end{bmatrix} = \vec{f}^x \\ \begin{bmatrix} \dot{\Gamma}_i^y \\ \dot{\Gamma}_j^y \end{bmatrix} &= \begin{bmatrix} g_{12}(\vec{\eta})\{\Gamma_j^y - \Gamma_i^y\} \\ g_{21}(\vec{\eta})\{\Gamma_i^y - \Gamma_j^y\} \end{bmatrix} = \vec{f}^y \end{aligned} \quad (5.42)$$

Where: $\vec{\eta} = [\Gamma_i^x, \Gamma_j^x, \Gamma_i^y, \Gamma_j^y]^T$. Let $\vec{f} = [\vec{f}^x^T, \vec{f}^y^T]^T$, and its Jacobian is given as follows:

$$\frac{\partial \vec{f}}{\partial \vec{q}} = \begin{bmatrix} \frac{\partial \vec{f}^x}{\partial \vec{\Gamma}^x} & \frac{\partial \vec{f}^x}{\partial \vec{\Gamma}^y} \\ \frac{\partial \vec{f}^y}{\partial \vec{\Gamma}^x} & \frac{\partial \vec{f}^y}{\partial \vec{\Gamma}^y} \end{bmatrix} \quad (5.42.1)$$

where: $\vec{\Gamma}^x = [\Gamma_i^x, \Gamma_j^x]^T$ and $\vec{\Gamma}^y = [\Gamma_i^y, \Gamma_j^y]^T$.

Now, we require to have a HPF, i.e., we need to have a curl-free and divergence-free field.

This can be realized by forcing the Jacobian symmetry and achieving the Laplace's equation. This can be ensured if and only if the following conditions are met:

$$\frac{\partial g_{ij}}{\partial \Gamma_i^x} \{\Gamma_j^y - \Gamma_i^y\} = \frac{\partial g_{ij}}{\partial \Gamma_i^y} \{\Gamma_j^x - \Gamma_i^x\} \quad (5.42.2)$$

$$\frac{\partial g_{ji}}{\partial \Gamma_i^x} \{\Gamma_j^y - \Gamma_i^y\} = \frac{\partial g_{ij}}{\partial \Gamma_i^y} \{\Gamma_j^x - \Gamma_i^x\} \quad (5.42.3)$$

$$\frac{\partial g_{ji}}{\partial \Gamma_i^x} = -\frac{\partial g_{ij}}{\partial \Gamma_j^x}, \quad \frac{\partial g_{ji}}{\partial \Gamma_i^y} = -\frac{\partial g_{ij}}{\partial \Gamma_j^y} \quad (5.42.4)$$

$$g_{ij} = g_{ji} \quad (5.42.5)$$

As expected, these conditions are like what was presented in Chapter 3, namely: (3.14) and (3.34); however, by including two dimensions instead of only one. As stated before, forcing a symmetric Jacobian ensures that the $\vec{e} = \vec{1}$ is a nonlinear right eigenvector associated with $\lambda_1 = 0 \forall \vec{\eta}$. These conditions are as follows:

$$\frac{\partial g_{ij}}{\partial \Gamma_i^x} = -\frac{\partial g_{ij}}{\partial \Gamma_j^x}, \quad \frac{\partial g_{ij}}{\partial \Gamma_i^y} = -\frac{\partial g_{ij}}{\partial \Gamma_j^y} \quad (5.42.6)$$

As a result, the Laplace's equation is given as follows:

$$\frac{\partial g_{ij}}{\partial \Gamma_i^x} \{\Gamma_j^x - \Gamma_i^x\} + \frac{\partial g_{ij}}{\partial \Gamma_i^y} \{\Gamma_j^y - \Gamma_i^y\} = 2g_{ij} \quad (5.42.7)$$

A candidate solution is the following:

$$g_{ij}(\vec{r}) = \frac{H}{(\Gamma_j^x - \Gamma_i^x)^2 + (\Gamma_j^y - \Gamma_i^y)^2} = \frac{H}{\|r_j - r_i\|^2} \quad (5.42.8)$$

So, the HPF associated with \vec{f} is given as follows:

$$p(\|r\|) = H \log(\|r_j - r_i\|) \quad (5.42.9)$$

which is indeed a HPF with spherical symmetry in 2-D [168]. The strength of the field is given by H . Note that the Laplace's equation is invariant under translation, therefore we may choose the origin of the HPF freely.

Remark 5.11: As seen from the above discussion, if more dimensions are involved the size of the Jacobian gets larger. However, it is sufficient to work at the edge-level rather than working on the MAS level. This helps avoiding the curse of dimensionality usually encountered under similar situations.

Note 5.1: Comparing (5.42.8) with the C-S in 2-D, the C-S model, even when $\beta = 0.5$, does not satisfy the Laplace's equation and therefore there will be no problem in preserving connectivity in q -dimensions, in general.

Following the previous steps, the needed conditions to build a HPF in 3-D using the proposed framework can be obtained. Therefore, the Laplace's equation is given as follows:

$$\frac{\partial g_{ij}}{\partial \Gamma_i^x} \{\Gamma_j^x - \Gamma_i^x\} + \frac{\partial g_{ij}}{\partial \Gamma_i^y} \{\Gamma_j^y - \Gamma_i^y\} + \frac{\partial g_{ij}}{\partial \Gamma_i^z} \{\Gamma_j^z - \Gamma_i^z\} = 3g_{ij} \quad (5.42.10)$$

A candidate solution is the following:

$$g_{ij} = \frac{H}{\left[(\Gamma_j^x - \Gamma_i^x)^2 + (\Gamma_j^y - \Gamma_i^y)^2 + (\Gamma_j^z - \Gamma_i^z)^2 \right]^{3/2}} = \frac{H}{\|r_j - r_i\|^3} \quad (5.42.11)$$

So, the HPF associated with \vec{f} is given as follows:

$$p(\|r\|) = H/\|r_j - r_i\| \quad (5.42.12)$$

which is indeed a HPF with spherical symmetry in 3-D [168]. Note that this results in the law of motion of planets given in (4.54.1).

Before concluding this subsection, let us have the following definition:

Definition 5.2: A 1-D edge is an edge connecting two agents interacting along one dimension of the working/configuration space. A q -D edge is defined in a similar fashion.

The Jacobian of a 3-D edge is given as follows:

$$\frac{\partial \vec{f}}{\partial \vec{q}} = \begin{bmatrix} \frac{\partial \vec{f}^x}{\partial \vec{\Gamma}^x} & \frac{\partial \vec{f}^x}{\partial \vec{\Gamma}^y} & \frac{\partial \vec{f}^x}{\partial \vec{\Gamma}^z} \\ \frac{\partial \vec{f}^y}{\partial \vec{\Gamma}^x} & \frac{\partial \vec{f}^y}{\partial \vec{\Gamma}^y} & \frac{\partial \vec{f}^y}{\partial \vec{\Gamma}^z} \\ \frac{\partial \vec{f}^z}{\partial \vec{\Gamma}^x} & \frac{\partial \vec{f}^z}{\partial \vec{\Gamma}^y} & \frac{\partial \vec{f}^z}{\partial \vec{\Gamma}^z} \end{bmatrix} \quad (5.42.13)$$

Recalling (4.8), the g_{ij} functions reflect the interconnection between agents in specific spatial directions. So, if they are connected along the z -direction using g_{ij} functions that depend only on the agents z -coordinates, then we have:

$$\frac{\partial \vec{f}}{\partial \vec{q}} = \begin{bmatrix} \frac{\partial \vec{f}^x}{\partial \vec{\Gamma}^x} & \frac{\partial \vec{f}^x}{\partial \vec{\Gamma}^y} & 0 & 0 \\ \frac{\partial \vec{f}^y}{\partial \vec{\Gamma}^x} & \frac{\partial \vec{f}^y}{\partial \vec{\Gamma}^y} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.42.14)$$

Which means that the motion along the z -direction is independent from the motion in the other two directions. This is useful when the MAS is operating over distinct communication networks in the working space as will be shown later in this thesis.

In this subsection, a systematic method to construct HPFs on q -D was presented. In the coming subsection, enabling the trajectory-generator with awareness of its working environment will be explained.

5.2.3: Modeling the environment into the Kino-dynamics of trajectory-generator systems

Remarkably, the addition of gradients is a natural consequence of superposition property enjoyed by HPF which follows from the linearity of the Laplace's equation [168]. According to Helmholtz decomposition, a vector field can be decomposed into potential components having divergence-free and curl-free, solenoidal component having divergence-free but not curl-free, and irrotational component having curl-free but not divergence-free. So, we may define a potential component that is responsible for collision avoidance, and an irrotational field responsible for steering the MAS towards the target point, namely: the average value of agents' initial conditions.

Taking advantage of the property of super position enjoyed by the HPF, we may define several HPFs to map the environment. Treating obstacles as points of singularity in the working space allows defining a HPF at each obstacle. More sophisticated representations of obstacles can be found in [168].

Figure 5.11 shows a working environment with two stationary obstacles each approximated by a point-of-singularity, or simply as a point obstacle. In this sense, neighboring agents on the graph are treated as mobile obstacles, which helps agents avoiding collision with each other.

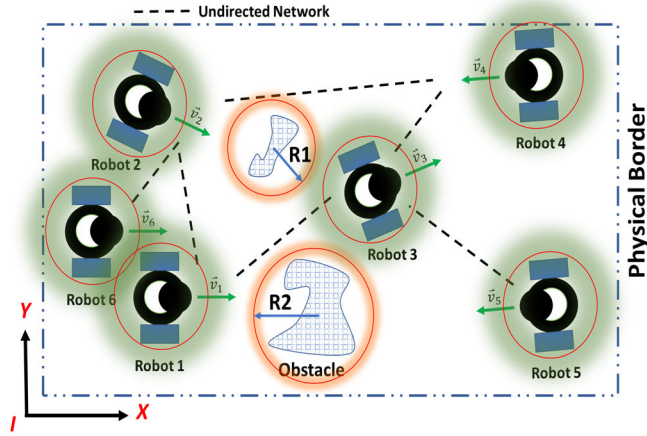


Figure. 5.11. A visualization of a MAS system navigating a working space with obstacles approximated as point obstacles whose field strength is shown in red. Each agent is building a source potential field to avoid colliding with its neighbors on the graph.

Let the combined HPFs gradients of k stationary obstacles from the perspective of the i^{th} agent be given as follows:

$$\begin{aligned}
 O_{R_x^i} &= \sum_{j=1}^k r_{i,o_j}^x = \sum_{j=1}^k \frac{R_j(\Gamma_i^x - O_j^x)}{(\Gamma_i^x - O_j^x)^2 + (\Gamma_i^y - O_j^y)^2} \\
 O_{R_y^i} &= \sum_{j=1}^k r_{i,o_j}^y = \sum_{j=1}^k \frac{R_j(\Gamma_i^y - O_j^y)}{(\Gamma_i^x - O_j^x)^2 + (\Gamma_i^y - O_j^y)^2}
 \end{aligned}
 \tag{5.43}$$

where: R_j decides the strength of the source, i.e., repulsive, field. To avoid dead-lock while in motion, a tangential component can be added to the trajectory-generator system such that dead-lock situations are avoided. This is depicted in Figure 5.12 where agents roll over the boundary of an obstacle in a way that favors the increase of connectivity among agents. Clearly, from Figure 5.12, two tangential directions are possible for agents to take when confronted by obstacles. This is given as follows:

$$\begin{bmatrix} t_{i,o_j}^x \\ t_{i,o_j}^y \end{bmatrix} = \begin{bmatrix} -r_{i,o_j}^x \\ r_{i,o_j}^y \end{bmatrix} \quad (5.43.1)$$

$$\begin{bmatrix} t_{i,o_j}^x \\ t_{i,o_j}^y \end{bmatrix} = \begin{bmatrix} r_{i,o_j}^x \\ -r_{i,o_j}^y \end{bmatrix} \quad (5.43.2)$$

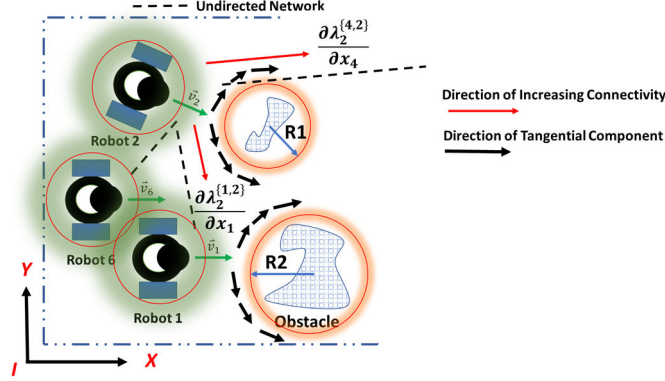


Figure. 5.12. Rolling around the obstacles boundaries in favor to increase connectivity with other graph neighboring agents.

By incorporating the connectivity-preserving concept in choosing one of the previous relations, we may write the following:

$$\begin{bmatrix} t_{i,o_j}^x \\ t_{i,o_j}^y \end{bmatrix} = b_{\lambda_2} \begin{bmatrix} -r_{i,o_j}^x \\ r_{i,o_j}^y \end{bmatrix} \quad (5.43.3)$$

$$b_{\lambda_2} = \text{sign} \left(\left[\sum_{j \in \mathcal{N}_i} \frac{\partial \lambda_2^{\{i,j\}}}{\partial \Gamma_i^x} \right] \cdot \begin{bmatrix} -r_{i,o_j}^x \\ r_{i,o_j}^y \end{bmatrix} \right) \neq 0 \quad (5.43.4)$$

$$b_{\lambda_2} = 1, \quad \text{otherwise}$$

Note that other criteria can be used to facilitate the preferred tangential direction. Figure 5.13 captures a situation where both tangents are perpendicular to the connectivity-preserving gradient in 2-D.

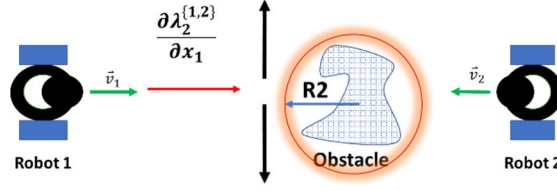


Figure. 5.13. A situation where the connectivity-preserving gradient and the two tangents are perpendicular.

Remark 5.12: A 3-D version of the previous derivation can be realized in a straightforward fashion.

Remark 5.13: In (5.43.3), the decision can be made based on one or some selected edges instead of all neighbors.

By utilizing the smoothness of the HPFs, the obstacles kinematics can be incorporated into the Kino-dynamics of the trajectory-generator systems by differentiating (5.43) and treating the obstacles velocities as disturbances. Moreover, to avoid chattering problems especially after differentiation, the *sign* function in (5.43.4) can be replaced by the *tanh* function as demonstrated in previous parts.

5.2.4: Sensing the environment using on-board sensors

Usually, the agents' onboard sensors can sense the environment and form a *sensing link* with the surrounding objects. If the agent must avoid these parts of the environment, i.e., they are obstacles, then the sensing link should be *repulsive*, and it should be *attractive* if these parts of the environments are targets, way points or the like. For example, a proximity sensor can be modelled as follows:

$$F_{R_{x,i}} = T_i F_{R_0} / |x_i - O| \quad (5.44)$$

where: $F_{R_{x,i}}$ is the sensory input for the i^{th} agent, i.e., robot, in the x-direction, T_i is an appropriate transformation matrix and F_{R_0} is a real proportionality constant. For example, let $F_{R_{x,i}}$ be the attractive/repulsive *soft-force* generated by the sensor due to an obstacle,

maybe another agent, obstructing the agent from proceeding in the x-direction. When the sensors are all the same, then the coefficient F_{R_0} is the same for all sensors. One example of this force is depicted in Figure 5.14.

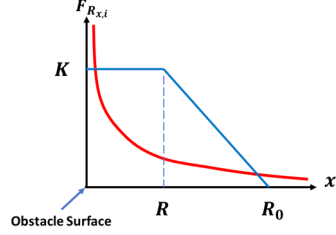


Figure. 5.14. The soft-force as a function of an agent position with respect to an obstacle surface. The non-localized form in red and the localized form in blue where K is a real constant.

The overall sensory information of s -sensors in the x -direction is given as $F_{R_x^i} = \sum_{k=1}^s F_{R_{x,k}}$. Similarly, $F_{R_y^i}$ and $F_{R_z^i}$ can be found.

Remark 5.14: Using sensory inputs increases the level of awareness of the trajectory-generator system. As a result, the trajectory-generator can deal with more realistic situation where unmodeled obstacles are expected in the working space. This will be demonstrated by an example where a group of attackers attack an asset guarded by escorting agents.

Consider the wheeled-robot, shown in Figure 5.15, having four identical proximity sensors arranged symmetrically onboard. This plurality of sensory devices generates a soft-force that can be modelled as follows:

$$\begin{bmatrix} F_{R_x^i} \\ F_{R_y^i} \end{bmatrix} = F_{R_0} \sum_{k=1}^4 s_{k,i} \begin{bmatrix} \cos(45^\circ + \theta_i + 90^\circ(k-1)) \\ \sin(45^\circ + \theta_i + 90^\circ(k-1)) \end{bmatrix} \quad (5.45)$$

where: $s_{k,i}$ is the reading of the k^{th} sensor and θ_i is the actual orientation of the i^{th} agent with respect to the global inertial coordinate frame.

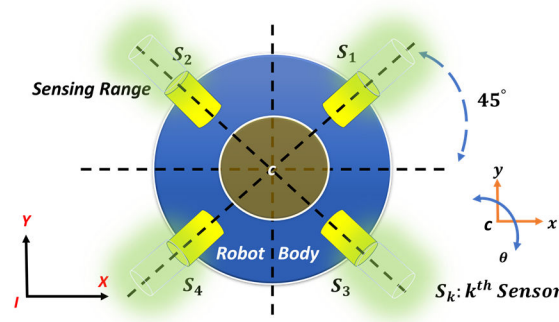


Figure. 5.15. A wheeled-robot having four symmetrically distributed identical proximity sensors onboard. The global inertial coordinate frame is IXY and the local coordinate frame is cxy . The orientation of the i^{th} agent is given with respect to IXY .

Like (5.43), the sensor-based navigation can be modified by introducing tangential components. The sensor information is crucial in realizing what is known as a reactive controller, see Figure 5.12.b.

Therefore, combining the model-based and sensor-based navigation capabilities into the trajectory-generator Kino-dynamics given in (5.39.1) results in the following- in any direction:

$$\begin{bmatrix} \ddot{\vec{\Gamma}} \\ \ddot{\vec{\Gamma}} \end{bmatrix} = \Sigma_s \begin{bmatrix} \ddot{\vec{\Gamma}} \\ \ddot{\vec{\Gamma}} \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} + \begin{bmatrix} \vec{F}_R + \vec{O}_{R_-} \\ \vec{d}_3 \end{bmatrix} \quad (5.46)$$

where: $\vec{F}_R = [F_{R_x^1}, \dots, F_{R_x^N}]^T$, $\vec{O}_{R_-} = [O_{R_x^1}, \dots, O_{R_x^N}]^T$ and $\vec{d}_3 = \frac{d}{dt} \{\vec{F}_R + \vec{O}_{R_-}\}$ or can be set to $[0]_{N \times 1}$.

Considering (5.43), the contribution of $O_{R_x^i}$ can appear directly in Σ_s instead of being considered as a disturbance.

5.2.5: The Complete picture of trajectory-generator systems

In the previous subsections, we have introduced the needed modifications and components to build a sophisticated trajectory-generator system. This included the physical agent

dynamics through the tracking errors, and the context-awareness through modeling and sensing the environment. The relations among these building blocks is given in Figure 5.16 from which the complete picture of Kino-dynamic trajectory-generator can be understood. A similar approach for joint planning and control of nonholonomic mobile robots was presented in [167]; however, here we are addressing the behavior control jointly with navigation assuming a local tracking controller to be already available.

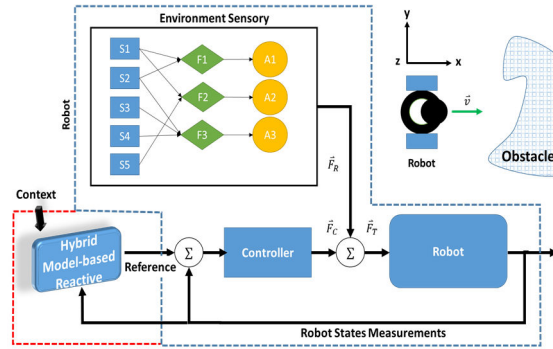
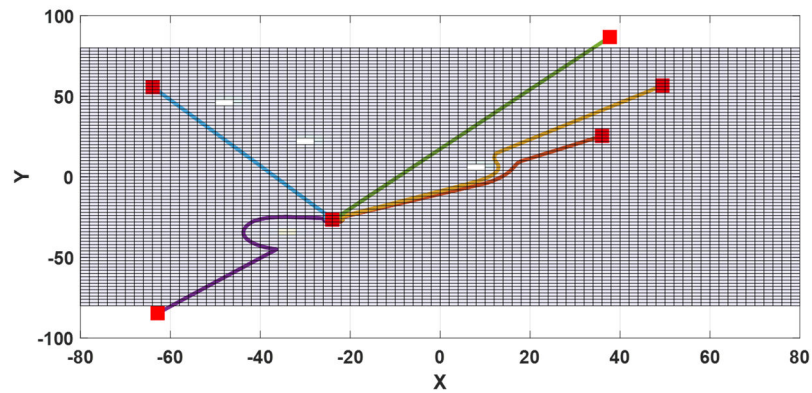
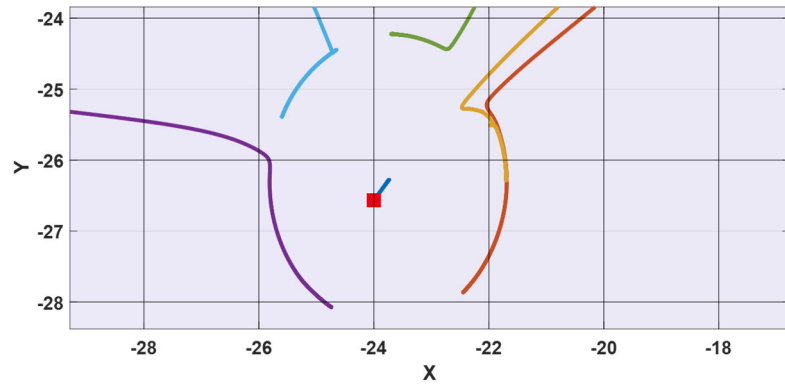


Figure. 5.16. The overall structure of the proposed trajectory-generator system.

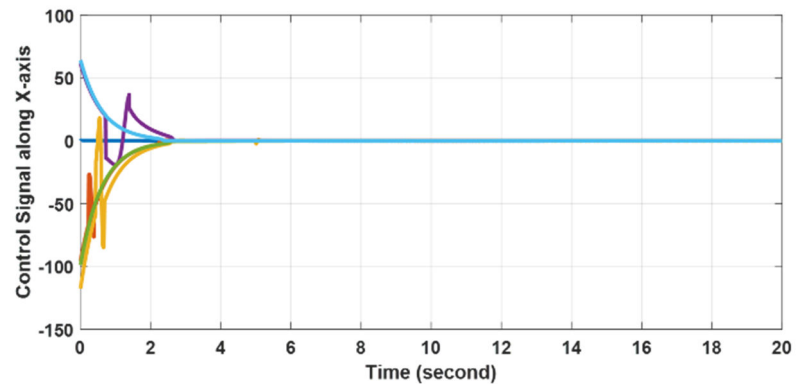
Next, the simulation results of two behaviors, namely: containment and consensus, in a work space that contains obstacles are provided. The kinematical trajectory-generator (5.33) was used to generate these behaviors and (5.43) is used to realize the HPF introduced to (5.33).



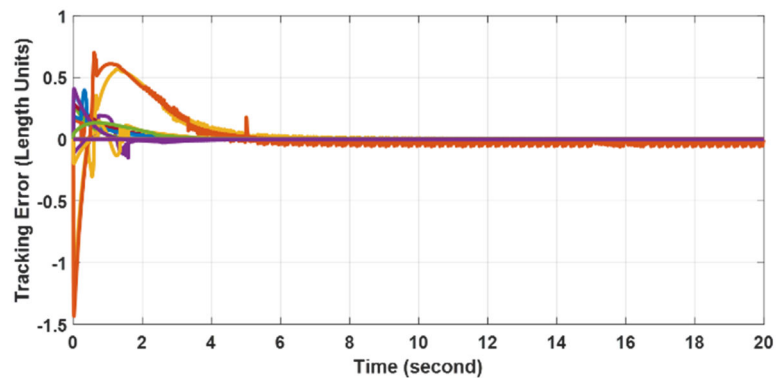
(a)



(b)

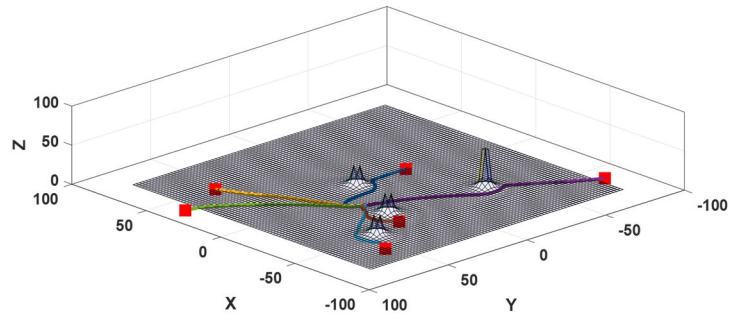


(c)

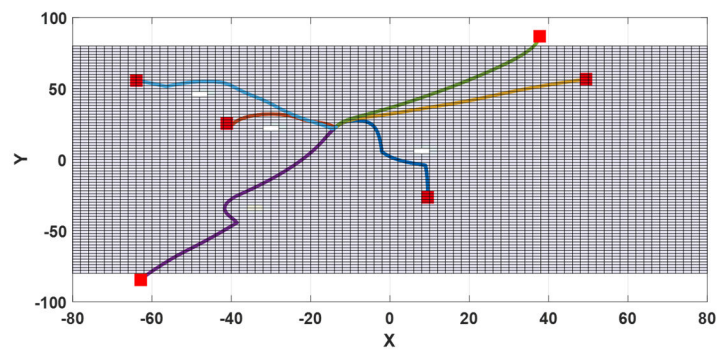


(d)

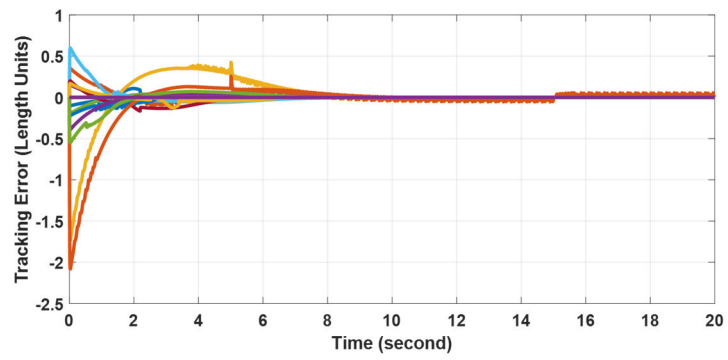
Figure. 5.17. Simulation results of the 2-D containment behavior realized by six heterogeneous linear and scalar systems where stationary obstacles exist in the working space. The red squares represent the initial positions (a) an overview of the complete behavior. (b) a close view of the containment behavior. (c) the needed control signals in the x-direction. (d) the tracking error during the behavior.



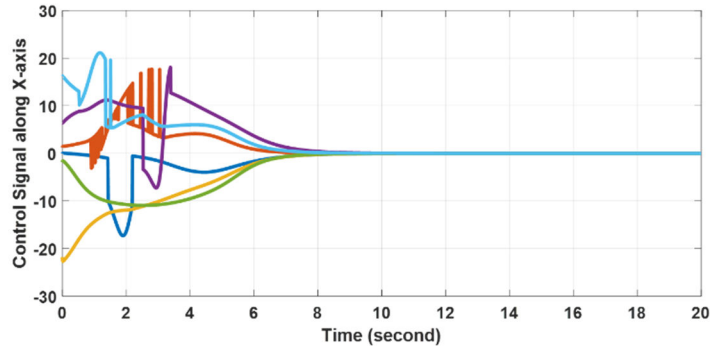
(a)



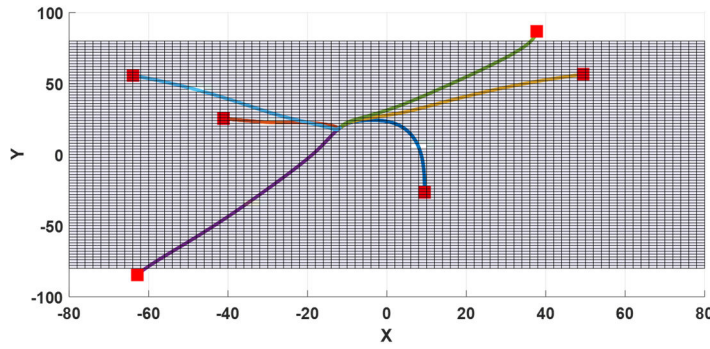
(b)



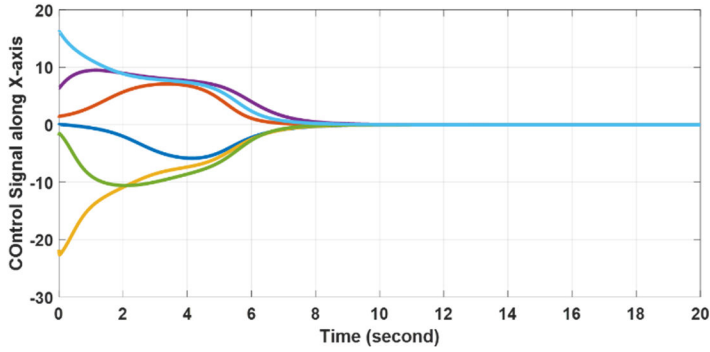
(c)



(d)



(e)



(f)

Figure. 5.18. Simulation results of the 2-D consensus behavior realized by six heterogeneous linear and scalar systems where stationary obstacles exist in the working space. The red squares represent the initial positions (a) a 3-D overview of the complete behavior. (b) a close 2-D view of the working space and the trajectory of agents during behavior conduction. (c) the tracking errors. (d) the control signals in the x-direction. (e) the same behavior when obstacle avoidance is inactive. (f) the needed control signals in the x-direction when working in an obstacle-free working space.

The consensus behavior is shown in Figure 5.18 where the working space was modelled using HPFs. In Figure 5.14.b, the agents' achieved consensus value was $\{-13.6240, 22.8937, 0\}$ while the ideal value achieved by the trajectory-generators was $\{-13.6241, 22.8935, 0\}$. The average value of the initial conditions was $\{-11.8753, 18.8720, 0\}$. The differences between the last two are due to the tracking error contribution in (5.33).

The effectiveness of the proposed approach can be seen from the results shown in figures 5.17 and 5.18.

In the coming sections, agents with higher-order dynamical models will be considered.

5.3 Multi-agent Systems of N -Identical double-integrator systems

In general, most mechanical systems can be modeled using second order systems whether linear or nonlinear. Thus, the double-integrator systems represent the simplest nontrivial holonomic dynamical systems that can be used in many applications, in general. Usually, this requires both the position and velocity command signals to be available.

Generally, these signals could be communicated over distinct communication networks due to security reasons or application constraints. For example, an agent that is almost stationary does not need to communicate its velocity to agents moving significantly faster with respect to it. In this section, we will consider double-integrator systems as building blocks of the trajectory generator systems.

The model of the i^{th} double-integrator trajectory-generator system can be given by:

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix}_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i \quad (5.47)$$

The state-dependent protocol needed to build a distributed version of (5.47) is given as follows:

$$u_i = -[\beta_1, \beta_2] \sum_{j \in \mathcal{N}_i} \begin{bmatrix} g_{ij}(x_i, x_j) & 0 \\ 0 & \hat{g}_{ij}(v_i, v_j) \end{bmatrix} \begin{bmatrix} x_i - x_j \\ v_i - v_j \end{bmatrix} \quad (5.48)$$

which can be simplified as follows:

$$\begin{aligned} u_i = \beta_1 \left\{ - \left(\sum_{j \in \mathcal{N}_i} g_{ij} \right) x_i + \left(\sum_{j \in \mathcal{N}_i} g_{ij} x_j \right) \right\} + \\ \beta_2 \left\{ - \left(\sum_{j \in \mathcal{N}_i} \hat{g}_{ij} \right) v_i + \left(\sum_{j \in \mathcal{N}_i} \hat{g}_{ij} v_j \right) \right\} \end{aligned} \quad (5.48.1)$$

Or collectively, the input of the overall trajectory-generator systems can be given as follows:

$$\vec{u} = -\beta_1 \mathcal{L}(\vec{x}) \vec{x} - \beta_2 \hat{\mathcal{L}}(\vec{v}) \vec{v} \quad (5.48.2)$$

where: $\vec{x} = [x_1, x_2, \dots, x_N]^T$ and $\vec{v} = [v_1, v_2, \dots, v_N]^T$ with $\beta_1, \beta_2 > 0$, yet to be determined.

Thus, the overall dynamics can be written as follows:

$$\dot{\vec{Z}} = \begin{bmatrix} \dot{\vec{x}} \\ \dot{\vec{v}} \end{bmatrix} = \begin{bmatrix} [0] & I \\ -\beta_1 \mathcal{L}(\vec{x}) & -\beta_2 \hat{\mathcal{L}}(\vec{v}) \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{v} \end{bmatrix} = A_c(\vec{Z}) \vec{Z} = \vec{f}(\vec{Z}) \quad (5.49)$$

where: $\vec{Z} = [\vec{x}^T, \vec{v}^T]^T$. Clearly, (5.49) is a nonlinear system given in a semi-linear form whose equilibria set \mathfrak{Z} is given such that $\mathfrak{Z} = \{(\vec{x}, \vec{v}) \in \mathfrak{R}^{2N} \mid (\vec{x}, \vec{v}) = (\vec{0}, \vec{0}) \text{ or } (\vec{x}, \vec{v}) = (\vec{1}, \vec{0})\}$ assuming connected undirected graphs associated with \mathcal{L} and $\hat{\mathcal{L}}$. The Jacobian of (5.49) once evaluated at the equilibria set can be given as follows:

$$\frac{\partial \vec{f}}{\partial \vec{Z}} = \begin{bmatrix} [0] & I \\ -\beta_1 \mathcal{L}_f & -\beta_2 \hat{\mathcal{L}}_f \end{bmatrix} \quad (5.49.1)$$

where: \mathcal{L}_f and $\hat{\mathcal{L}}_f$ are the fixed Laplacian matrices associated with connected undirected graphs selected originally by the designer.

In (5.49), the velocity of all agents, i.e., \vec{v} , reaches a consensus value of the arithmetic mean of the velocities initial values if and only if \hat{g}_{ij} functions are designed properly as proposed in Chapter 3. Thus, the positions of all agents, i.e., \vec{x} , will also reach consensus that might be increasing with time assuming the consensus protocol to be convergent, see (3.87) and the discussion that followed. Considering this, it is important to study the stability of (5.49) under similar and different \mathcal{L} and $\hat{\mathcal{L}}$ matrices assuming their associated graphs to be connected undirected graphs.

Using (5.49.1), the eigenvalue concept can be used to check the stability of (5.49). Thus, the system eigenvalues can be found by solving the following equation:

$$|s^2I + \beta_2\hat{\mathcal{L}}_f s + \beta_1\mathcal{L}_f| = 0 \quad (5.49.2)$$

where: s is the differential Laplace operator. Since (5.49.2) is difficult to be solved in general, let us consider diagonalizing both \mathcal{L}_f and $\hat{\mathcal{L}}_f$ Laplacian matrices by the following similarity transformations:

$$\mathcal{L}_f = MJ_1M^{-1} \quad (5.49.3)$$

$$\hat{\mathcal{L}}_f = NJ_2N^{-1} \quad (5.49.4)$$

where: J_1 and J_2 are diagonal matrices containing the eigenvalues of both \mathcal{L}_f and $\hat{\mathcal{L}}_f$ matrices, respectively. The matrices M and N consist of the right eigenvectors, as columns, of both \mathcal{L} and $\hat{\mathcal{L}}$ matrices while M^{-1} and N^{-1} consist of the left eigenvectors, as rows, of both \mathcal{L} and $\hat{\mathcal{L}}$ matrices, respectively. Next, the cases where agents positions and velocity are communicated over distinct or alike communication networks will be investigated.

5.3.1: Similar communication graphs

Since both networks are alike, let (5.49.3) be used as follows:

$$\bar{A}_c = \begin{bmatrix} M^{-1} & [0] \\ [0] & M^{-1} \end{bmatrix} \begin{bmatrix} [0] & I \\ -\beta_1 \mathcal{L}(\vec{x}) & -\beta_2 \mathcal{L}(\vec{x}) \end{bmatrix} \begin{bmatrix} M & [0] \\ [0] & M \end{bmatrix} \quad (5.49.5)$$

So, we will have:

$$\bar{A}_c = \begin{bmatrix} [0] & I \\ -\beta_1 J_1 & -\beta_2 J_1 \end{bmatrix} \quad (5.49.6)$$

Thus, (5.11) can be solved easily as follows:

$$|s^2 I + \beta_2 J_1 s + \beta_1 J_1| = \prod_{i=1}^N (s^2 + \beta_2 \lambda_i s + \beta_1 \lambda_i) = 0 \quad (5.49.7)$$

Therefore, the eigenvalues of \bar{A}_c are given by:

$$s_i = -\frac{1}{2} \left(\beta_2 \lambda_i \pm \sqrt{(\beta_2 \lambda_i)^2 - 4\beta_1 \lambda_i} \right) \quad (5.49.8)$$

Now, since we are mainly interested in dealing with connected and undirected graphs, the values of β_1 and β_2 could be any positive value since λ_i will always be positive real for all \vec{x} and \vec{v} . Since an undirected graph is represented by a symmetric Laplacian matrix, its eigenvalues are always real. However, if oscillation in the MAS response is not desired, i.e., s_i must be real, then β_1 value should be designed such that the following inequality is satisfied:

$$0 < \beta_1 \leq \frac{\beta_2^2 \min(\lambda_i)}{4}, \quad \lambda_i > 0 \quad (5.49.9)$$

It is worth noting that there are two eigenvalues of \bar{A}_c for each eigenvalue λ_i of \mathcal{L} [9].

Similar results can also be obtained when we use $\mathcal{L}(\vec{v})$ or $\mathcal{L}(\vec{x}, \vec{v})$.

5.3.2: Distinct communication graphs

$$\bar{A}_c = \begin{bmatrix} M^{-1} & [0] \\ [0] & N^{-1} \end{bmatrix} \begin{bmatrix} [0] & I \\ -\beta_1 \mathcal{L}(\vec{x}) & -\beta_2 \hat{\mathcal{L}}(\vec{v}) \end{bmatrix} \begin{bmatrix} M & [0] \\ [0] & N \end{bmatrix} \quad (5.49.10)$$

or:

$$\bar{A}_c = \begin{bmatrix} N^{-1} & [0] \\ [0] & M^{-1} \end{bmatrix} \begin{bmatrix} [0] & I \\ -\beta_1 \mathcal{L}(\vec{x}) & -\beta_2 \hat{\mathcal{L}}(\vec{v}) \end{bmatrix} \begin{bmatrix} N & [0] \\ [0] & M \end{bmatrix} \quad (5.49.11)$$

So, we will have:

$$|s^2 I + \beta_2 J_2 s + \beta_1 N^{-1} \mathcal{L} N| = 0 \quad (5.49.12)$$

or:

$$|s^2 I + \beta_2 M^{-1} \hat{\mathcal{L}} M s + \beta_1 J_1| = 0 \quad (5.49.13)$$

which are difficult to generalize unless a suitable relation between the two communication graphs are imposed like being *co-spectral* for example. Both (5.49.12) and (5.49.13) suggest that the eigenvectors can be used as means to design one of the networks if the other is desired by specifications. However, the effect of general distinct Laplacian matrices is still worthy of investigation.

5.3.3: Studying the invariant quantity of MAS comprising N -identical double-integrator systems

Under state-dependent Laplacian matrices, we can find the consensus value of (5.49) as follows:

Let the arithmetic mean of the MAS consisting of N -identical double-integrator systems be given as follows:

$$\vec{\mu}_a(t) = \begin{bmatrix} \bar{x}(t) \\ \bar{v}(t) \end{bmatrix} = \frac{1}{N} \sum_{i=1}^N \begin{bmatrix} x_i(t) \\ v_i(t) \end{bmatrix} \quad (5.50)$$

Taking the time-derivative and equating it to zero, yields:

$$\frac{d\vec{\mu}_a(t)}{dt} = \frac{1}{N} \sum_{i=1}^N \begin{bmatrix} \dot{x}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \frac{1}{N} \sum_{i=1}^N \begin{bmatrix} v_i(t) \\ u_i(t) \end{bmatrix} \neq \vec{0}, \forall v_i(t), t \geq t_0 \quad (5.50.1)$$

Using (5.47) into (5.50.1), yields:

$$\dot{\vec{v}}(t) = [1 \ 1 \ \cdots \ 1] [u_1, u_2, \cdots, u_N]^T = \vec{1}^T \vec{u} = 0$$

$$\forall x_i(t), v_i(t), t \geq t_0 \quad (5.50.2)$$

If and only if the $\vec{1}^T$ is a nonlinear left eigenvector of both $\mathcal{L}(\vec{x})$ and $\hat{\mathcal{L}}(\vec{v})$ appearing in (5.48.2). Integrating (5.50), yields:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i(0) + t\bar{v} \quad (5.50.3)$$

Where:

$$\bar{v} = \frac{1}{N} \sum_{i=1}^N v_i(0) \quad (5.50.4)$$

where: t denotes the time. From (5.50.3) and (5.50.4), consensus of (5.49) to the arithmetic mean of the agents' initial values is achieved if the graphs used were connected undirected and both the g_{ij} and \hat{g}_{ij} are designed as per Chapter 3. Clearly, if $\bar{v} \neq 0$ then \bar{x} will be time-varying and consequently (5.49) is deemed unstable. Although this behavior is desired in some cases, like rendezvous for example, it might be difficult to use in realizing the behaviors shown in Chapter 4. As a remedy, one may consider generating the needed velocity signal in the trajectory-generator system using (5.33) or (5.34) instead.

5.4 Multi-agent Systems of N-Identical coupled-integrators systems

In this section, an extension of the previous results to include general time-invariant coupling between purely integrator systems is presented. This section will show how the input matrix affects the designed consensus protocols through allowing several inputs to affect one state in the system simultaneously. This will help in understanding how the proposed communication protocol can be used to ease both analysis and design of linear time-invariant (LTI) MAS. Suppose we have the following coupled-integrators, not necessarily a trajectory-generator, system given by:

$$\vec{\dot{x}}_i = B\vec{u}_i(t) \quad (5.51)$$

where: $\vec{x}_i \in \mathbb{R}^n$ and $\vec{u}_i \in \mathbb{R}^m$.

It is important to consider the cases where we have:

- Over actuated systems, i.e., $m > n$.
- Under actuated systems, i.e., $m < n$.
- Squared systems, i.e., $m = n$.

to gain more understanding about the coupling resulting from the input matrix being other than the identity matrix. In MAS consisting of single-integrators, both m and n were equal, while in the double-integrator based MAS $m < n$. Thus, it is intended in this section to investigate the case where $m > n$. The main objective is to convert the over/under actuated systems into squared systems using full-state feedback consensus protocols.

To do this, a modified version of the protocol given in (5.48) is used to build the needed MAS of the specified systems and is given as follows:

$$\vec{u}_i = -K \sum_{j \in \mathcal{N}_i} \text{diag}([g_{ij}^1(x_{i1}, x_{j1}), \dots, g_{ij}^n(x_{in}, x_{jn})]) \begin{bmatrix} x_{i1} - x_{j1} \\ \vdots \\ x_{in} - x_{jn} \end{bmatrix} \quad (5.51.1)$$

where: $K \in \mathbb{R}^{m \times n}$ is a gain to be designed and g_{ij}^l is a weight function associated with the l^{th} state where $l = 1, 2, \dots, n$.

Remark 5.15: Notice that the gain used, i.e., K , is chosen to be identical for all agents. This will dramatically make the analysis and design of the corresponding system more obvious. In case agents are different, i.e., heterogeneous MAS, then the input matrix will be different so should the gain used. This fact will make the design process more involved, in general. Now, to make things more appealing, let us consider the following example to show how (5.51) can be used in (5.51.1).

Example 5.1: Assume undirected links exist among three coupled-integrators agents as shown in Figure 5.3 where the agent dynamics are described as follows:

$$\begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \end{bmatrix}_i = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}_i \quad (5.52)$$

Let K be given as follows:

$$K = \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \\ k_5 & k_6 \end{bmatrix} \quad (5.52.1)$$

Thus, using (5.51.1) and (5.52.1) in (5.52), the overall dynamics of the first agent can be given as follows:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}_1 = -K \left\{ \begin{bmatrix} g_{12}^1(x_{11}, x_{21}) & 0 \\ 0 & g_{12}^2(x_{12}, x_{22}) \end{bmatrix} \begin{bmatrix} x_{11} - x_{21} \\ x_{12} - x_{22} \end{bmatrix} + \begin{bmatrix} g_{13}^1(x_{11}, x_{31}) & 0 \\ 0 & g_{13}^2(x_{12}, x_{32}) \end{bmatrix} \begin{bmatrix} x_{11} - x_{31} \\ x_{12} - x_{32} \end{bmatrix} \right\} \quad (5.52.2)$$

$$\begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \end{bmatrix}_1 = - \left\{ \begin{bmatrix} c_1(g_{12}^1 + g_{13}^1) & -c_1 g_{12}^1 & -c_1 g_{13}^1 \\ c_3(g_{12}^1 + g_{13}^1) & -c_3 g_{12}^1 & -c_3 g_{13}^1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} + \begin{bmatrix} c_2(g_{12}^2 + g_{13}^2) & -c_2 g_{12}^2 & -c_2 g_{13}^2 \\ c_4(g_{12}^2 + g_{13}^2) & -c_4 g_{12}^2 & -c_4 g_{13}^2 \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \end{bmatrix} \right\} \quad (5.52.3)$$

$$\begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \\ k_5 & k_6 \end{bmatrix} \quad (5.52.4)$$

$$= \begin{bmatrix} b_1 k_1 + b_2 k_3 + b_3 k_5 & b_1 k_2 + b_2 k_4 + b_3 k_6 \\ b_4 k_1 + b_5 k_3 + b_6 k_5 & b_4 k_2 + b_5 k_4 + b_6 k_6 \end{bmatrix}$$

Similarly, the overall dynamics for the other two agents are given as follows:

$$\begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \end{bmatrix}_2 = - \left\{ \begin{bmatrix} -c_1 g_{21}^1 & c_1(g_{21}^1 + g_{23}^1) & -c_1 g_{23}^1 \\ -c_3 g_{21}^1 & c_3(g_{21}^1 + g_{23}^1) & -c_3 g_{23}^1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} + \begin{bmatrix} -c_2 g_{21}^2 & c_2(g_{21}^2 + g_{23}^2) & -c_2 g_{23}^2 \\ -c_4 g_{21}^2 & c_4(g_{21}^2 + g_{23}^2) & -c_4 g_{23}^2 \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \end{bmatrix} \right\} \quad (5.52.5)$$

$$\begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \end{bmatrix}_3 = - \left\{ \begin{bmatrix} -c_1 g_{31}^1 & -c_1 g_{32}^1 & c_1(g_{31}^1 + g_{32}^1) \\ -c_3 g_{31}^1 & -c_3 g_{32}^1 & c_3(g_{31}^1 + g_{32}^1) \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} + \begin{bmatrix} -c_2 g_{31}^2 & -c_2 g_{32}^2 & c_2(g_{31}^2 + g_{32}^2) \\ -c_4 g_{31}^2 & -c_4 g_{32}^2 & c_4(g_{31}^2 + g_{32}^2) \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \end{bmatrix} \right\} \quad (5.52.6)$$

Let $\vec{Z}_1 = [x_{11}, x_{21}, x_{31}]^T$ and $\vec{Z}_2 = [x_{12}, x_{22}, x_{32}]^T$. So, by rearranging the states of the agents carefully, then the MAS dynamics can be described as follows:

$$\begin{bmatrix} \vec{Z}_1 \\ \vec{Z}_2 \end{bmatrix} = - \underbrace{\begin{bmatrix} c_1 I & c_2 I \\ c_3 I & c_4 I \end{bmatrix}}_{\text{Coupling Matrix}} \begin{bmatrix} \mathcal{L}(\vec{Z}_1) & [0] \\ [0] & \hat{\mathcal{L}}(\vec{Z}_2) \end{bmatrix} \begin{bmatrix} \vec{Z}_1 \\ \vec{Z}_2 \end{bmatrix} \quad (5.52.7)$$

where $\mathcal{L}(\vec{Z}_1)$ and $\hat{\mathcal{L}}(\vec{Z}_2)$ are the state-dependent Laplacian matrices associated with the network topologies used to communicate the first and second states among the agents, respectively.

□

Remark 5.16: Notice the difference between (4.8) and (5.52.7) to identify the effect of the input matrix when it is other than the identity matrix.

Remark 5.17: One important role of the gain K appearing in (5.52.1) is its use in decoupling the inputs, appearing in (5.52), if possible. In other words, if it is possible to find a gain K such that the coupling matrix is a positive diagonal matrix, then the states are decoupled. In such a case, the stability of the decoupled system requires mainly that all the networks used to communicate the states be connected and undirected.

Let B in (5.51) be given as follows:

$$B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad (5.52.8)$$

If K is designed as:

$$K = \begin{bmatrix} 1 & 2 \\ 2 & -2 \\ -2 & 5 \end{bmatrix} \quad (5.52.9)$$

Then, (5.52.7) will be decoupled, and therefore can be rewritten as follows:

$$\begin{bmatrix} \vec{Z}_1 \\ \vec{Z}_2 \end{bmatrix} = - \begin{bmatrix} 3\mathcal{L}(\vec{Z}_1) & [0] \\ [0] & 3\hat{\mathcal{L}}(\vec{Z}_2) \end{bmatrix} \begin{bmatrix} \vec{Z}_1 \\ \vec{Z}_2 \end{bmatrix} \quad (5.52.10)$$

Now, to study the stability of the MAS consisting of N -identical coupled-integrators systems, we need to have the involved matrices as positive definite/semi-definite. For example, in (5.52.7), we want:

$$A_{cl} = \begin{bmatrix} c_1 I & c_2 I \\ c_3 I & c_4 I \end{bmatrix} \begin{bmatrix} \mathcal{L}(\vec{Z}_1) & [0] \\ [0] & \hat{\mathcal{L}}(\vec{Z}_2) \end{bmatrix} \quad (5.52.11)$$

to be always positive semi-definite.

Thus, by making sure that $c_4 > 0$ and $c_1 - c_2 c_4^{-1} c_3 > 0$, then A_{cl} can be made positive semi-definite; since the multiplication of a positive and semi-positive matrices is a positive semi-definite matrix [169].

From trajectory-generator point of view, the time-invariant coupling matrix can be used to generate interesting behaviors in the MAS. In Chapter 4, behaviors were generated by switching the underlying communication network and/or the g_{ij} functions used. Using (5.51) instead, behaviors can be generated by switching the coupling matrix between pre-designed gains that suit the desired behavior without changing the underlying communication network or the g_{ij} functions used. Consider for example six agents, whose dynamics are given by (5.51), with $B \in \mathbb{R}^{2 \times 2}$. So, to design an x-y trajectory-generator system using (5.51), with $m = 2$, let:

$$B = \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix}, \quad K = \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix} \quad (5.53)$$

Thus, the coupling matrix can be given as follows:

$$\begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix} = BK = \begin{bmatrix} b_1k_1 + b_2k_3 & b_1k_2 + b_2k_4 \\ b_3k_1 + b_4k_3 & b_3k_2 + b_4k_4 \end{bmatrix} \quad (5.53.1)$$

Mainly, two interesting behaviors can be generated using the coupling matrix by setting it as follows:

$$Beh_1 \Rightarrow \begin{bmatrix} 0 & -c_2 \\ c_3 & 0 \end{bmatrix}, Beh_2 \Rightarrow \begin{bmatrix} c_1 & 0 \\ 0 & c_4 \end{bmatrix} \quad (5.53.2)$$

After the designer picks the needed values of c_i for each behavior, (5.53.1) can be solved to satisfy both behaviors by selecting an appropriate common B matrix, and solving for K that achieves the intended behaviors. For example, let (5.53.2) be:

$$Beh_1 \Rightarrow \begin{bmatrix} 0 & -5 \\ 5 & 0 \end{bmatrix}, Beh_2 \Rightarrow \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad (5.53.3)$$

Then, the following matrices can be used to achieve the desired behaviors:

$$B = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, K_{Beh_1} = \begin{bmatrix} 5 & 5 \\ -5 & -10 \end{bmatrix}, K_{Beh_2} = \begin{bmatrix} -2 & 2 \\ 4 & -2 \end{bmatrix} \quad (5.53.4)$$

Clearly, Beh_1 is an orbiting behavior, while Beh_2 is a consensus behavior.

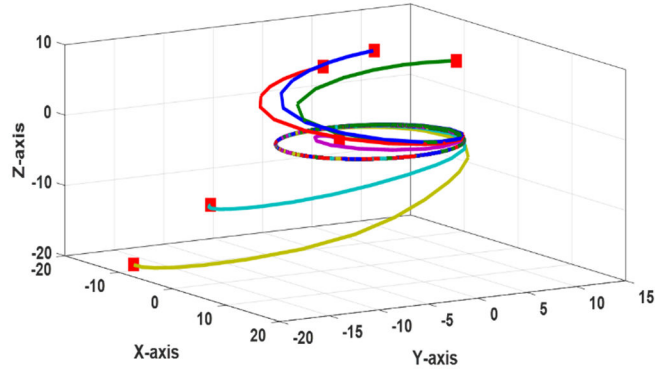
If (5.51) is originally designed as a trajectory-generator system, then the previous behaviors can be straightforwardly achieved by directly manipulating the B matrix as follows:

$$B = \begin{bmatrix} 0 & -5 \\ 5 & 0 \end{bmatrix} \text{ or } B = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad (5.53.5)$$

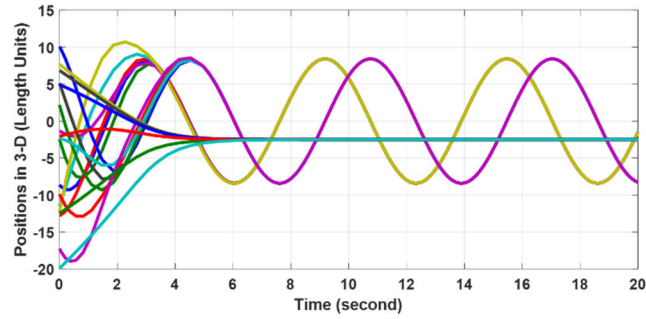
Figure 5.19 shows an orbiting behavior in 2-D and a consensus behavior along the z -axis.

The B matrix can then be given as follows- with real constants $a, b > 0$:

$$B = \begin{bmatrix} 0 & -a & 0 \\ a & 0 & 0 \\ 0 & 0 & b \end{bmatrix} \quad (5.53.6)$$



(a)



(b)

Figure. 5.19. The orbiting behavior. (a): 3-D view of trajectories of all agents. (b): synchronization of trajectories.

Note that the orbiting will happen about the average value of initial conditions in the x - y plane.

5.5 Multi-agent Systems of N-Identical general LTI systems

In this section, an extension of the previous results to include general linear time-invariant systems will be presented. It will be made clear how the proposed communication protocol can be used to ease both analysis and design of LTI MAS. Suppose we have the following LTI system given by:

$$\dot{\vec{x}}_i = A\vec{x}_i(t) + B\vec{u}_i(t) \quad (5.54)$$

where: $\vec{x}_i \in \mathbb{R}^n$ and $\vec{u}_i \in \mathbb{R}^m$.

The dynamics provided in (5.54) reflects more realistic models that real existing dynamical systems may exhibit. Usually in the related literature, such models are stacked as blocks, which helps in visualizing the effects of both the dynamics and the communication network when looking at the MAS overall dynamics. However, this stacking comes along with a preferred realization of MAS dynamics if a common communication networks is used to transmit the data, i.e., the agent states, among neighbors. The resulting MAS dynamics may appear more appealing to study than the case where states are communicated over distinct networks especially when they are state-dependent, see (5.52.7), for example.

Recently, a new representation of the Laplacian matrix was proposed in [170] by which individual states are given a matrix-weight arguments and not scalar weights as usually used in the standard literature. However, [170] makes the resulting representation less obvious to be described as states being communicated via distinct networks. Having the states communicated via distinct networks can be inspired by security reasons or more generally by hierarchical structures associated with the MAS, and the assigned mission.

One main goal in this section is to show the equivalency between our proposed representation and the standard representation when all states are communicated via the same network. This will be done through an example. In this example, a leader agent is assumed to be existing and the follower agents are entitled to reach consensus with its states.

Example 5.2: Assume undirected links exist among three 2nd-order single-input and single output (SISO) LTI agents, given by (5.54), as shown in Figure 5.20. Let the used communication protocol be given as follows:

$$u_i = -c[k_1, k_2] \left\{ \sum_{j \in \mathcal{N}_i} \left\{ \begin{bmatrix} g_{ij}(x_{1j}, x_{1i}) & 0 \\ 0 & \hat{g}_{ij}(x_{2j}, x_{2i}) \end{bmatrix} \begin{bmatrix} x_{1j} - x_{1i} \\ x_{2j} - x_{2i} \end{bmatrix} \right\} + \begin{bmatrix} h_{i1}(x_{10}, x_{1i}) & 0 \\ 0 & h_{i2}(x_{20}, x_{2i}) \end{bmatrix} \begin{bmatrix} x_{10} - x_{1i} \\ x_{20} - x_{2i} \end{bmatrix} \right\} \quad (5.54.1)$$

Let the 2nd-order SISO LTI agent dynamics be given as follows:

$$\vec{\dot{x}}_i = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_i \quad (5.54.2)$$

To ease the derivation, let us assume fully-connected networks, and by using (5.54) in (5.54.1) the overall dynamics of the first agent can be given as follows:

$$\dot{x}_{11} = [u_{11}, u_{12}, u_{13}] \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \end{bmatrix} + [v_{11}, v_{12}, v_{13}] \begin{bmatrix} x_{21} \\ x_{22} \\ x_{23} \end{bmatrix} + [ch_{11}b_1k_1, ch_{12}b_1k_2] \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

$$u_{11} = a_1 - cg_{12}b_1k_1 - cg_{13}b_1k_1 - ch_{11}b_1k_1$$

$$u_{12} = cg_{12}b_1k_1$$

$$u_{13} = cg_{13}b_1k_1$$

$$v_{11} = a_2 - c\hat{g}_{12}b_1k_2 - c\hat{g}_{13}b_1k_2 - ch_{12}b_1k_2$$

$$v_{12} = c\hat{g}_{12}b_1k_2$$

$$v_{13} = c\hat{g}_{13}b_1k_2$$

$$\dot{x}_{21} = [u_{21}, u_{22}, u_{23}] \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \end{bmatrix} + [v_{21}, v_{22}, v_{23}] \begin{bmatrix} x_{21} \\ x_{22} \\ x_{23} \end{bmatrix} + [ch_{11}b_2k_1, ch_{12}b_2k_2] \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

$$u_{21} = a_3 - cg_{12}b_2k_1 - cg_{13}b_2k_1 - ch_{11}b_2k_1$$

$$u_{22} = cg_{12}b_2k_1$$

$$u_{23} = cg_{13}b_2k_1$$

$$v_{21} = a_4 - c\hat{g}_{12}b_2k_2 - c\hat{g}_{13}b_2k_2 - ch_{12}b_2k_2$$

$$\begin{aligned}
v_{22} &= c\hat{g}_{12}b_2k_2 \\
v_{23} &= c\hat{g}_{13}b_2k_2
\end{aligned}
\tag{5.54.3}$$

Similarly, the overall dynamics of the second agent can be given as follows:

$$\begin{aligned}
\dot{x}_{12} &= [\bar{u}_{11}, \bar{u}_{12}, \bar{u}_{13}] \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \end{bmatrix} + [\bar{v}_{11}, \bar{v}_{12}, \bar{v}_{13}] \begin{bmatrix} x_{21} \\ x_{22} \\ x_{23} \end{bmatrix} + [ch_{21}b_1k_1, ch_{22}b_1k_2] \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} \\
\bar{u}_{11} &= cg_{21}b_1k_1 \\
\bar{u}_{12} &= a_1 - cg_{21}b_1k_1 - cg_{23}b_1k_1 - ch_{21}b_1k_1 \\
\bar{u}_{13} &= cg_{23}b_1k_1 \\
\bar{v}_{11} &= c\hat{g}_{21}b_1k_2 \\
\bar{v}_{12} &= a_2 - c\hat{g}_{21}b_1k_2 - c\hat{g}_{23}b_1k_2 - ch_{22}b_1k_2 \\
\bar{v}_{13} &= c\hat{g}_{23}b_1k_2 \\
\dot{x}_{22} &= [\bar{u}_{21}, \bar{u}_{22}, \bar{u}_{23}] \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \end{bmatrix} + [\bar{v}_{21}, \bar{v}_{22}, \bar{v}_{23}] \begin{bmatrix} x_{21} \\ x_{22} \\ x_{23} \end{bmatrix} + [ch_{21}b_2k_1, ch_{22}b_2k_2] \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} \\
\bar{u}_{21} &= cg_{21}b_2k_1 \\
\bar{u}_{22} &= a_3 - cg_{21}b_2k_1 - cg_{23}b_2k_1 - ch_{21}b_2k_1 \\
\bar{u}_{23} &= cg_{23}b_2k_1 \\
\bar{v}_{21} &= c\hat{g}_{21}b_2k_2 \\
\bar{v}_{22} &= a_4 - c\hat{g}_{21}b_2k_2 - c\hat{g}_{23}b_2k_2 - ch_{22}b_2k_2 \\
\bar{v}_{23} &= c\hat{g}_{23}b_2k_2
\end{aligned}
\tag{5.54.4}$$

Also, the overall dynamics of the third agent can be given as follows:

$$\dot{x}_{13} = [\bar{u}_{11}, \bar{u}_{12}, \bar{u}_{13}] \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \end{bmatrix} + [\bar{v}_{11}, \bar{v}_{12}, \bar{v}_{13}] \begin{bmatrix} x_{21} \\ x_{22} \\ x_{23} \end{bmatrix} + [ch_{31}b_1k_1, ch_{32}b_1k_2] \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

$$\bar{u}_{11} = cg_{31}b_1k_1$$

$$\bar{u}_{12} = cg_{32}b_1k_1$$

$$\bar{u}_{13} = a_1 - cg_{31}b_1k_1 - cg_{32}b_1k_1 - ch_{31}b_1k_1$$

$$\bar{v}_{11} = c\hat{g}_{31}b_1k_2$$

$$\bar{v}_{12} = c\hat{g}_{32}b_1k_2$$

$$\bar{v}_{13} = a_2 - c\hat{g}_{31}b_1k_2 - c\hat{g}_{32}b_1k_2 - ch_{32}b_1k_2$$

$$\dot{x}_{23} = [\bar{u}_{21}, \bar{u}_{22}, \bar{u}_{23}] \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \end{bmatrix} + [\bar{v}_{21}, \bar{v}_{22}, \bar{v}_{23}] \begin{bmatrix} x_{21} \\ x_{22} \\ x_{23} \end{bmatrix} + [ch_{31}b_2k_1, ch_{32}b_2k_2] \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix}$$

$$\bar{u}_{21} = cg_{31}b_2k_1$$

$$\bar{u}_{22} = cg_{32}b_2k_1$$

$$\bar{u}_{23} = a_3 - cg_{31}b_2k_1 - cg_{32}b_2k_1 - ch_{31}b_2k_1$$

$$\bar{v}_{21} = c\hat{g}_{31}b_2k_2$$

$$\bar{v}_{22} = c\hat{g}_{32}b_2k_2$$

$$\bar{v}_{23} = a_4 - c\hat{g}_{31}b_2k_2 - c\hat{g}_{32}b_2k_2 - ch_{32}b_2k_2$$

(5.54.5)

Now, let $\vec{Z}_1 = [x_{11}, x_{12}, x_{13}]^T$, $\vec{Z}_2 = [x_{21}, x_{22}, x_{23}]^T$, $\vec{Z}_{10} = x_{10}\vec{1}_N^T$ and $\vec{Z}_{20} = x_{20}\vec{1}_N^T$. So, we can write the MAS dynamics as follows:

$$\begin{bmatrix} \vec{Z}_1 \\ \vec{Z}_2 \end{bmatrix} = \begin{bmatrix} a_1I_N - cb_1k_1(\mathcal{L}_1 + \mathcal{H}_1) & a_2I_N - cb_1k_2(\mathcal{L}_2 + \mathcal{H}_2) \\ a_3I_N - cb_2k_1(\mathcal{L}_1 + \mathcal{H}_1) & a_4I_N - cb_2k_2(\mathcal{L}_2 + \mathcal{H}_2) \end{bmatrix} \begin{bmatrix} \vec{Z}_1 \\ \vec{Z}_2 \end{bmatrix} \\ + c \begin{bmatrix} b_1k_1I_N & b_1k_2I_N \\ b_2k_1I_N & b_2k_2I_N \end{bmatrix} \begin{bmatrix} \mathcal{H}_1 & [0] \\ [0] & \mathcal{H}_2 \end{bmatrix} \begin{bmatrix} \vec{Z}_{10} \\ \vec{Z}_{20} \end{bmatrix}$$

(5.54.6)

where: \mathcal{L}_1 and \mathcal{L}_2 are the associated state-dependent Laplacian matrices that convey the first and second states of each agent to its neighbors, respectively. \mathcal{H}_1 and \mathcal{H}_2 denotes the pinning gains of the first and second states of the leader with the corresponding first and second states of each follower, respectively. Also, we have $\mathcal{H}_d = \text{diag}([h_{1d}, h_{2d}, h_{3d}])$, $d = 1, 2$.

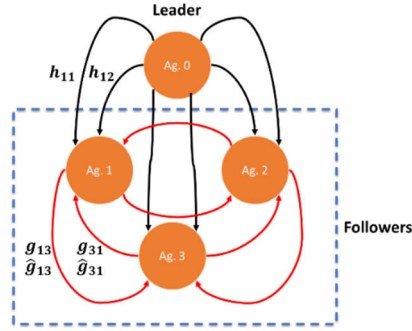


Figure 5.20. Three 2nd-order LTI agents with a single leader where each follower state is connected via a different network topology. The same can be said about the pinning gains denoted by the letter h .

Using the Kronecker product operator, (5.54.6) can be simplified as follows:

$$\begin{aligned}
 \vec{\dot{Z}} = \begin{bmatrix} \vec{\dot{Z}}_1 \\ \vec{\dot{Z}}_2 \end{bmatrix} &= \left(A \otimes I_N - c(BK \otimes I_N) \begin{bmatrix} \mathcal{L}_1 + \mathcal{H}_1 & [0] \\ [0] & \mathcal{L}_2 + \mathcal{H}_2 \end{bmatrix} \right) \begin{bmatrix} \vec{Z}_1 \\ \vec{Z}_2 \end{bmatrix} \\
 &\quad + c(BK \otimes I_N) \begin{bmatrix} \mathcal{H}_1 & [0] \\ [0] & \mathcal{H}_2 \end{bmatrix} \begin{bmatrix} \vec{Z}_{10} \\ \vec{Z}_{20} \end{bmatrix} \\
 &= \overline{A}_c \left(\vec{Z}(t), \vec{Z}_0(t) \right) \vec{Z}(t) + \overline{B}_c \left(\vec{Z}(t), \vec{Z}_0(t) \right) \vec{Z}_0(t) = \vec{f}(\vec{x})
 \end{aligned}
 \tag{5.54.7}$$

where: $K = [k_1, k_2]$ and $\vec{Z}_0 = [\vec{Z}_{10}^T, \vec{Z}_{20}^T]^T$.

□

Remark 5.18: It is easy to extend the results shown in (5.54.7) for agents with state vector $\vec{x}_i \in \mathbb{R}^n$ since (5.54.7) provides a systematic way of doing this by simply introducing additional networks.

Now, we would like to show the equivalency between our representation and the standard one available in the literature when the communication networks are alike, i.e., when $\mathcal{L}_1 = \mathcal{L}_2$ and $\mathcal{H}_1 = \mathcal{H}_2$, see [171] for example. So, by rearranging (5.54.3), we will end up having the following:

$$\begin{aligned}
\dot{x}_{11} &= [\tilde{u}_{11}, \tilde{u}_{12}] \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} + [\tilde{v}_{11}, \tilde{v}_{12}] \begin{bmatrix} x_{12} \\ x_{22} \end{bmatrix} + [\tilde{w}_{11}, \tilde{w}_{12}] \begin{bmatrix} x_{13} \\ x_{23} \end{bmatrix} + [ch_{11}b_1k_1, ch_{12}b_1k_2] \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} \\
\tilde{u}_{11} &= a_1 - cg_{12}b_1k_1 - cg_{13}b_1k_1 - ch_{11}b_1k_1 \\
\tilde{u}_{12} &= a_2 - cg_{12}b_1k_2 - cg_{13}b_1k_2 - ch_{12}b_1k_2 \\
\tilde{v}_{11} &= cg_{12}b_1k_1 \\
\tilde{v}_{12} &= cg_{12}b_1k_2 \\
\tilde{w}_{11} &= cg_{13}b_1k_1 \\
\tilde{w}_{12} &= cg_{13}b_1k_2 \\
\dot{x}_{21} &= [\tilde{u}_{21}, \tilde{u}_{22}] \begin{bmatrix} x_{11} \\ x_{21} \end{bmatrix} + [\tilde{v}_{21}, \tilde{v}_{22}] \begin{bmatrix} x_{12} \\ x_{22} \end{bmatrix} + [\tilde{w}_{21}, \tilde{w}_{22}] \begin{bmatrix} x_{13} \\ x_{23} \end{bmatrix} + [ch_{11}b_2k_1, ch_{12}b_2k_2] \begin{bmatrix} x_{10} \\ x_{20} \end{bmatrix} \\
\tilde{u}_{21} &= a_3 - cg_{12}b_2k_1 - cg_{13}b_2k_1 - ch_{11}b_2k_1 \\
\tilde{u}_{22} &= a_4 - cg_{12}b_2k_2 - cg_{13}b_2k_2 - ch_{12}b_2k_2 \\
\tilde{v}_{21} &= cg_{12}b_2k_1 \\
\tilde{v}_{22} &= cg_{12}b_2k_2 \\
\tilde{w}_{21} &= cg_{13}b_2k_1 \\
\tilde{w}_{22} &= cg_{13}b_2k_2
\end{aligned} \tag{5.54.8}$$

Similarly, both (5.54.4) and (5.54.5) can be rearranged. So, the overall MAS of three identical 2nd-order general LTI systems can be given as follows:

$$\begin{aligned}\vec{x}(t) &= [I_N \otimes A - c\{\mathcal{L}_s(\vec{x}) + \mathcal{H}_s(\vec{x}, \vec{x}_0)\} \otimes BK] \vec{x}(t) + [c\mathcal{H}_s(\vec{x}, \vec{x}_0) \otimes BK] \vec{x}_0(t) \\ &= A_c(\vec{x}(t), \vec{x}_0(t)) \vec{x}(t) + B_c(\vec{x}(t), \vec{x}_0(t)) \vec{x}_0(t) = \vec{f}(\vec{x})\end{aligned}\tag{5.54.9}$$

where: $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix and \otimes is the Kroneker product of two matrices.

$\vec{x} = [\vec{x}_1^T, \vec{x}_2^T, \dots, \vec{x}_N^T]^T$ and $\vec{x}_0 = \vec{1}_N \otimes [x_{10}, x_{20}]^T$ where: $\vec{1}_N = [1 \ 1 \ \dots \ 1]^T$.

The nonlinear systems $\vec{f}(\vec{x})$ and $\vec{f}(\vec{x})$ in (5.54.7) and (5.54.9) are given in a semi-linear state-dependent form. The stability of these systems can be ensured by a careful design of the networks, and the static values of both c and K used. However, since the Laplacian matrices under study are state-dependent, as well as the pinning matrices, static values of both c and K can be used to ensure stability of the overall MAS if and only if we restrict the states to evolve within a prescribed region in the state-space that includes their initial conditions as well. This restriction can be described as conservative and might not be applicable in many situations.

Considering the above discussion, it seems appealing to use a variable state-feedback gain, i.e., $\bar{K}_i(\vec{x}_i, \vec{x}_j)$, instead of a static one, i.e., K . Under such variable gain, we might simply keep the value of c to be static or simply set it to exactly 1. A self-loop can be added at each follower agent as depicted in Figure 5.21. These loops will enable agents to maintain their stability irrespective of the changes occurring at the state-dependent Laplacian matrix and at the same time they will facilitate the agents with self-awareness and the ability to situate themselves within the evolving contexts. This is equivalent to adding a stabilizing local controller.

Revisiting **Example 5.2**, the modified communication protocol is given as follows:

$$u_i = -c[k_1, k_2] \left\{ \sum_{j \in \mathcal{N}_i} \left\{ \begin{bmatrix} g_{ij}(x_{1j}, x_{1i}) & 0 \\ 0 & \hat{g}_{ij}(x_{2j}, x_{2i}) \end{bmatrix} \begin{bmatrix} x_{1j} - x_{1i} \\ x_{2j} - x_{2i} \end{bmatrix} \right\} + \begin{bmatrix} h_{i1}(x_{10}, x_{1i}) & 0 \\ 0 & h_{i2}(x_{20}, x_{2i}) \end{bmatrix} \begin{bmatrix} x_{10} - x_{1i} \\ x_{20} - x_{2i} \end{bmatrix} \right\} - \bar{K}_i(\vec{x}_i, \vec{x}_{j \in \mathcal{N}_i}, \vec{x}_0) \vec{x}_i \quad (5.55)$$

where: $\bar{K}_i = [\bar{k}_{1i}, \bar{k}_{2i}]$.

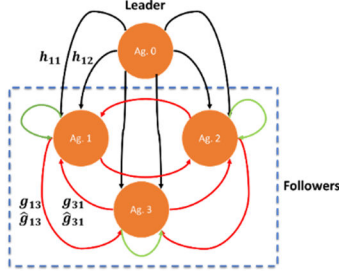


Figure 5.21. Three 2nd-order LTI agents with a single leader where each followers state is connected via a different network topology where the self-loops added are evident.

In (5.55), the role of the gain K is to reflect the changes within the neighborhood, while \bar{K} is responsible for, mainly, maintaining the stability of the agent irrespective of these changes. Thus, by using \bar{K} , the agent becomes self-aware and situated within its context. It is worth noting that the gain K could be the same for all agents while \bar{K}_i is assumed to be different and varying.

Under (5.55), the dynamics for the first agent- for example- can be given as follows:

$$\vec{x}_1 = A_1(\vec{x}_0, \vec{x}_1, \vec{x}_2, \vec{x}_3) \vec{x}_1 + B_{12}(\vec{x}_1, \vec{x}_2) \vec{x}_2 + B_{13}(\vec{x}_1, \vec{x}_3) \vec{x}_3 + B_{10}(\vec{x}_0, \vec{x}_1) \vec{x}_0 \bar{K}_1 \vec{x}_1 \quad (5.55.1)$$

where:

$$A_1 = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix} = A - (B_{12} + B_{13} + B_{10})$$

$$z_{11} = (a_1 - cg_{12}b_1k_1 - cg_{13}b_1k_1 - ch_{11}b_1k_1)$$

$$\begin{aligned}
z_{12} &= (a_2 - c\hat{g}_{12}b_1k_2 - c\hat{g}_{13}b_1k_2 - ch_{12}b_1k_2) \\
z_{21} &= (a_3 - cg_{12}b_2k_1 - cg_{13}b_2k_1 - ch_{11}b_2k_1) \\
z_{22} &= (a_4 - c\hat{g}_{12}b_2k_2 - c\hat{g}_{13}b_2k_2 - ch_{12}b_2k_2) \\
B_{12} &= cBK \begin{bmatrix} g_{12} & 0 \\ 0 & \hat{g}_{12} \end{bmatrix}, \quad B_{13} = cBK \begin{bmatrix} g_{13} & 0 \\ 0 & \hat{g}_{13} \end{bmatrix}, \\
B_{10} &= cBK \begin{bmatrix} h_{11} & 0 \\ 0 & h_{12} \end{bmatrix}
\end{aligned}$$

Thus, the MAS overall dynamics can be written in another format as follows- in which the communication networks effects are implicit:

$$\begin{aligned}
\vec{\dot{x}}(t) &= \begin{bmatrix} \vec{\dot{x}}_1 \\ \vec{\dot{x}}_2 \\ \vec{\dot{x}}_3 \end{bmatrix} = \begin{bmatrix} A_1 - B\bar{K}_1 & B_{12} & B_{13} \\ B_{21} & A_2 - B\bar{K}_2 & B_{23} \\ B_{31} & B_{32} & A_3 - B\bar{K}_3 \end{bmatrix} \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vec{x}_3 \end{bmatrix} + \begin{bmatrix} B_{10} \\ B_{20} \\ B_{30} \end{bmatrix} (\vec{1}_3 \otimes \vec{x}_0) \\
&= \bar{\bar{A}}_c(\vec{x}(t), \vec{x}_0(t))\vec{x}(t) + \bar{\bar{B}}_c(\vec{x}(t), \vec{x}_0(t))(\vec{1}_3 \otimes \vec{x}_0)
\end{aligned} \tag{5.55.2}$$

where: $\vec{1}_3 = [1 \ 1 \ 1]^T$.

If the communication networks used are undirected, then the stability of (5.55.2) can be easily studied as will be discussed shortly. The network topology appears implicitly under (5.55.2) such that when only $B_{ij} \neq [0]$, then a link exists.

Remark 5.19: Clearly, if the system matrix A is originally Hurwitz, then the stability of (5.55.2) is straightforwardly guaranteed as will be discussed shortly, and there will be no need for a local stabilizing controller. However, if the system matrix A is originally unstable, then a stabilizing controller for each agent is inevitable. This could be done using adaptive, robust or optimal techniques, for examples. Since the A_i matrices are state-dependent, one possible approach to design the needed \bar{K}_i gains in (5.52.2) could be

through the solution of state-dependent Riccati equation associated with Linear Quadratic Regulator (LQR) at each point of the state-space of concern.

Figure 5.22 depicts a schematic diagram of the resulting first agent dynamics under (5.55.2).

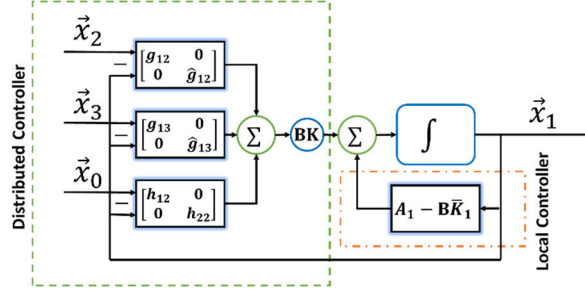


Figure. 5.22. The resulting dynamics of the first 2nd-order LTI agent under (5.55).

Notice that in $\overline{\overline{A_c}}(\vec{x}(t), \vec{x}_0(t))$, each row and column always sum up to $A - B\overline{K}_i$. So, $\overline{\overline{A_c}}$ is acting as a non-singular non-negative block M -matrix. Interestingly, if all g_{ij} and h_{ij} functions used utilize the same structure of the parameterized C-S model given in (3.116) with $\beta = 0.5$, then their values will be bounded and $\in (0, H_{ij}/\sqrt{\delta_{ij}}]$.

For simplicity, having all g_{ij} with similar H_{ij} and δ_{ij} values, $\overline{\overline{A_c}}$ will be equivalent to the following LTI matrix at full-coupling, i.e., when all agents are connected and all g_{ij} functions converge to $H/\sqrt{\delta}$ including the pinning gains:

$$\overline{\overline{A_c}} = \begin{bmatrix} A_1 - B\overline{K}_1 & cBK H/\sqrt{\delta} & cBK H/\sqrt{\delta} \\ cBK H/\sqrt{\delta} & A_2 - B\overline{K}_2 & cBK H/\sqrt{\delta} \\ cBK H/\sqrt{\delta} & cBK H/\sqrt{\delta} & A_3 - B\overline{K}_3 \end{bmatrix} \quad (5.56)$$

where: $A_i = A - 3cBK H/\sqrt{\delta}$ if A_i is pinned to the leader, and $A_i = A - 2cBK H/\sqrt{\delta}$ if it is not. Under the previous assumptions, and for a general MAS comprising N -identical general LTI systems, we may extend the previous results by writing: $A_i = A - c(|\mathcal{N}_i| + 1)BK H/\sqrt{\delta}$ if A_i is pinned to the leader, and $A_i = A - c|\mathcal{N}_i|BK H/\sqrt{\delta}$ if it is

not. $|\mathcal{N}_i|$ denotes the in-degree size of the i^{th} agent. Note that when the i^{th} agent becomes distant from the remaining of the MAS, then we will have $A_i \cong A$.

To understand the relation between the controller's gain K , the value of the coupling c and agents' stability, let us consider fixed weighted links among the agents themselves as well as the leader. Also, let us ignore the local stabilizing controller, i.e., by having $\bar{K}_i = [0]$, and let the neighborhood controller gain, i.e., K , be designed independently from the network topology using local algebraic Riccati equation (ARE) as follows:

Let the pair (A, B) be stabilizable, and $K = R^{-1}B^TP$ where P is the unique positive definite solution of ARE given as:

$$A^TP + PA + Q - PBR^{-1}B^TP = 0 \quad (5.57)$$

where: $Q \in \Re^{n \times n}$ and $R \in \Re^{m \times m}$ are positive definite matrices. After finding P , we need to determine the suitable value of the coupling c appearing in (5.56) with $H/\sqrt{\delta} = 1$. This can be done as follows [9]:

$$A_i^TP + PA_i + Q - PBR^{-1}B^TP = 0 \quad (5.57.1)$$

Considering $K = R^{-1}B^TP$ and (5.57), we may find the following after a simple calculation:

$$(A - c\lambda_i BK)^*P + P(A - c\lambda_i BK) = -Q + \{-2\alpha_i c + 1\}K^TRK \quad (5.57.2)$$

Where the superscript $*$ denotes the conjugate transpose of a complex matrix, $\lambda_i = \alpha_i + j\beta_i$ with $\alpha_i > 0, \beta_i \in \Re$ is the i^{th} complex eigenvalue of the matrix $\mathcal{L}_f + \mathcal{H}_f$ with \mathcal{L}_f is the fixed Laplacian matrix that can be associated with a digraph. According to Lyapunov theory, if $(A - c\lambda_i BK)^*P + P(A - c\lambda_i BK) < 0$, then the system is stable. So, solving for c in (5.57.2) by forcing the bracket to be negative, yields:

$$c \geq \frac{1}{2\alpha_i} \quad (5.57.3)$$

Since we want this to be valid for every agent, we need to have the following:

$$c \geq \frac{1}{2\min(\alpha_i)}, \quad \forall i = 1, 2, \dots, N \quad (5.57.3)$$

It is obvious that the properties of the underlying communication network appeared only when choosing the coupling value c , which facilitates reducing the design process of the complete MAS under consideration.

Choosing c such that (5.57.3) is valid guarantees unbounded synchronization region [9].

The previous results can also be used when the involved matrices are LTI where the locally designed controller, i.e., $K = R^{-1}B^T P$, minimizes the following local performance index:

$$J_i = \frac{1}{2} \int_0^\infty (\vec{x}_i^T Q \vec{x}_i + \vec{u}_i^T R \vec{u}_i) dt \quad (5.57.4)$$

and not the global performance of the MAS, which will be investigated in a future work.

Now, if an agent is far away from its neighbors, then the neighborhood contribution in stabilizing that agent will be minor, and therefore the agent may become unstable if it is not originally stable. Therefore, the local controller is needed, i.e., $\bar{K}_i \neq [0]$. This justifies our proposition of adding a local controller in (5.55.1).

When state-dependent weights are used, a balance between being a self-centered and a cooperative member in the team, an agent must situate itself within the current context. This can be realized using a state-dependent version of ARE known as state-dependent Riccati equation (SDRE) (cf. [172], [173]) as will also be investigated in a future work.

5.6 Multi-agent Systems of N-Identical nonlinearly-coupled higher-order integrator systems

In this section, an extension of the previous results to include N-identical nonlinearly-coupled higher-order integrator systems as kinematical trajectory-generator systems will

be presented. A set of heterogeneous agents of scalar dynamics given by (5.1) are used in conjunction with the developed trajectory-generator systems.

Suppose we have the following systems given by:

$$\vec{\dot{q}}_i = B(\vec{q}_i)\vec{u}_i(t) \quad (5.58)$$

where: $\vec{q}_i \in \mathbb{R}^n$ and $\vec{u}_i \in \mathbb{R}^m$.

Using the identical nonlinearly-coupled higher-order integrators helps in generating more sophisticated behaviors. For example, the containment-escorting behavior, shown in Chapter 4, can be extended to create a dynamic version of it such that agents will move on a spherical surface, generally an ellipsoid, without colliding while conducting the containment-escorting behavior in 3-D obstacle-free Euclidean space. This dynamic behavior is implemented using dynamical approach and not algorithmic approach. For example, the trajectory-generator utilizing angular momentum in 3-D is given as follows:

$$\vec{\dot{q}}_i = \begin{bmatrix} \dot{\Gamma}_i^x \\ \dot{\Gamma}_i^y \\ \dot{\Gamma}_i^z \end{bmatrix} = \begin{bmatrix} 0 & -\Gamma_i^z & \Gamma_i^y \\ \Gamma_i^z & 0 & -\Gamma_i^x \\ -\Gamma_i^y & \Gamma_i^x & 0 \end{bmatrix} \begin{bmatrix} u_{xi} \\ u_{yi} \\ u_{zi} \end{bmatrix} \quad (5.58.1)$$

Let u_{xi} be given as follows:

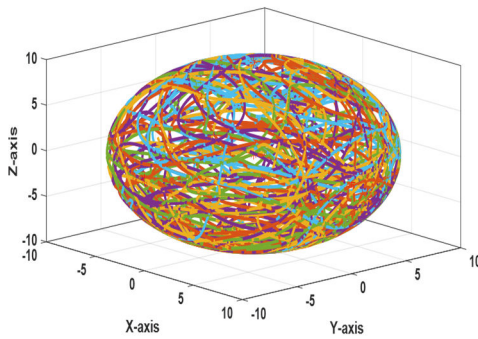
$$u_{xi} = \sum_{j \in \mathcal{N}_i^x} g_{ij}^x(\cdot, \cdot) \{\Gamma_j - \Gamma_i\} \quad (5.58.2)$$

where: the superscript x denotes the x -direction of the Euclidean space under consideration. Note that it is possible, by design, to limit the agent neighbors to certain agents in any direction even if they can communicate. Also g_{ij}^x can be other than g_{ij}^y or g_{ij}^z , in general. Thus, the overall dynamics of the MAS consisting of N trajectory-generator systems is given as follows:

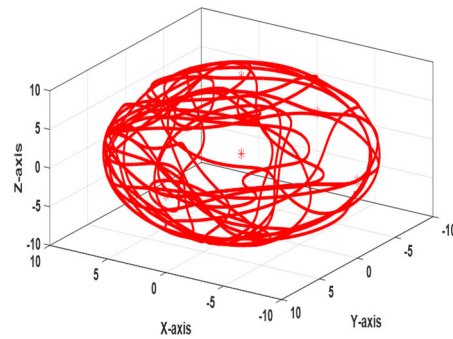
$$\vec{Q} = -\beta_f \begin{bmatrix} [0] & -D^z & D^y \\ D^z & [0] & -D^x \\ -D^y & D^x & [0] \end{bmatrix} \begin{bmatrix} \mathcal{L}^x(\vec{\Gamma}^x) & [0] & [0] \\ [0] & \mathcal{L}^y(\vec{\Gamma}^y) & [0] \\ [0] & [0] & \mathcal{L}^z(\vec{\Gamma}^z) \end{bmatrix} \begin{bmatrix} \vec{\Gamma}^x \\ \vec{\Gamma}^y \\ \vec{\Gamma}^z \end{bmatrix} \quad (5.58.3)$$

where: $\beta_f > 0$ is a real constant and scalar that determines the rate of convergence and $D^\tau = \text{diag}(\vec{\Gamma}^\tau)$, $\tau = \{x, y, z\}$. Figure 5.23 shows the simulation results of 3-D containment behavior over the network shown in Figure 2.47 and the protocol is designed according to (5.58.3) where the ideal trajectory signals were used. Note that the algebraic connectivity of the network in the z-direction is initially higher than the other two networks; simply because agents are initially assumed to be at the quadrants of a semi-sphere sliced by the x-y plane, so five agents share initially the same plane. This also justifies the connectivity evolution during approximately the first 40 seconds after which 3-D motion starts taking place.

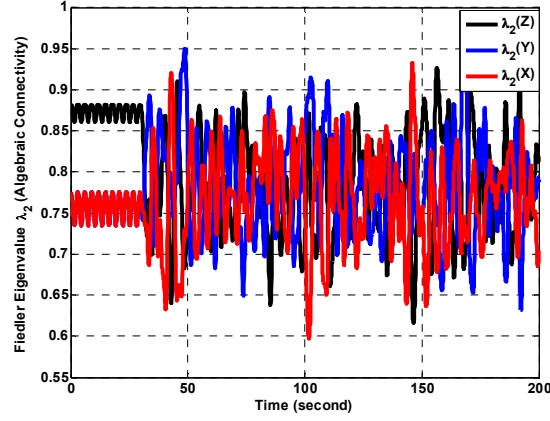
Now, let us examine the implications of **Remark 5.1**. If $g_{ij}^{x,y,z}$ in (5.58.2) depends totally on the actual driven system state $\vec{q}_i = [x_i, y_i, z_i]^T$, i.e., $g_{ij}^x(x_i, x_j)$, then the resulting MAS trajectory-generator will be a linear parameter-varying system and its resulting behavior is orbiting with the exception that agents will move along a spherical surface to approach the orbiting behavior as can be seen from Figure 5.24.



(a)

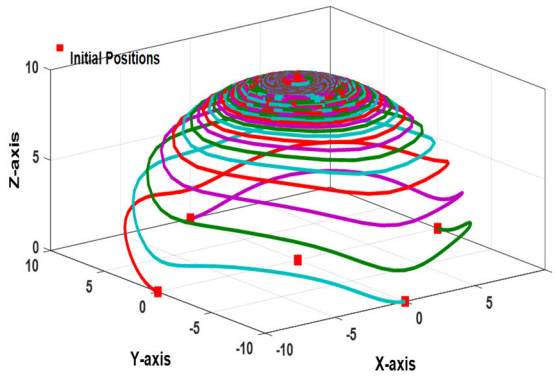


(b)

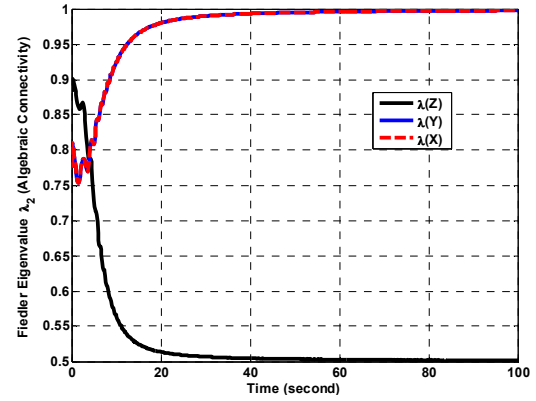


(c)

Figure. 5.23. The 3-D containment behavior with $g_{ij}^{x,y,z}(\Gamma_i, \Gamma_j)$. (a) ideal trajectories of all agents. (b) ideal trajectory of agent 3. (c) algebraic connectivity of all networks using the MATLAB $eig(\cdot)$ function.



(a)



(b)

Figure. 5.24. The 3-D containment behavior with $g_{ij}^{x,y,z}(x_i, x_j)$ in all networks. (a) ideal trajectories of all agents. (b) algebraic connectivity of all networks using the MATLAB $eig(\cdot)$ function.

Similarly, when $g_{ij}^{x,y,z}$ in (5.58.2) depends partially on the actual driven system state and the ideal state generated by the trajectory-generator systems of its neighbors, i.e., $g_{ij}^x(x_i, \Gamma_j^x)$, then the same results shown in Figure 5.24 are obtained. However, if we choose $g_{ij}^x(\Gamma_i^x, x_j)$, then the results are shown in Figure 5.25. These changes must be concurrently done to the other two networks, i.e., g_{ij}^y and g_{ij}^z , as well. In Figure 5.24, agents

will be orbiting on the surface of a semi-sphere with maximum likelihood of their existence in a certain position- not characterized in this study- while maintaining a distance among themselves, so collision is avoided as can be seen from (4.58.1). The difference between the first and last cases is obvious by observing Figures 3.23.b and 3.25.b. Figure 5.25 shows the orbiting behavior that evolves about a point moving on the spherical surface containing the agents' initial positions.

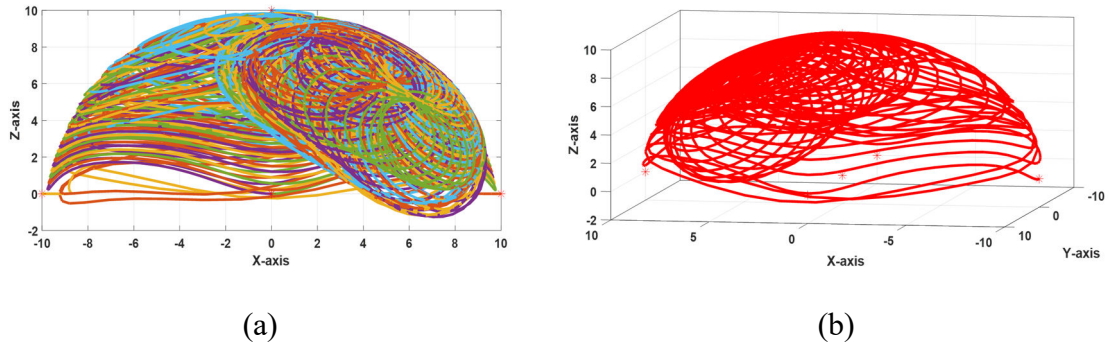


Figure. 5.25. The 3-D dynamic containment behavior with $g_{ij}(\Gamma_i^x, x_j)$ for all networks. (a): ideal trajectories of all agents. (b): ideal trajectory of agent 3.

5.6.1: Dynamic containment-escorting behaviors

In many situations, composite behaviors may evolve within a context during mission conduction. In Chapter 4, we have discussed the switching mechanism among agents residing in a behavior bank. This switching in behaviors reflects the fact that a single behavior may fully-dominate a certain situation over a certain period of time.

In this section, we will provide simulation examples by which we demonstrate the ability of the proposed framework to prove that it is efficient in designing composite behaviors. A composite behavior may consist of several primitive behaviors that evolve concurrently or sequentially.

A set of parallel behaviors consists of several primitive behaviors running concurrently at the same time-space dimensions. This allows cooperative-competitive behaviors within the

same set to be aware of each other and even to evolve on top of each other, and in many cases to be active at the same time.

Note 5.2: It is intended here to make things easy, so extracting an individual behavior contribution in tracking errors, prioritizing and solving conflicts among behaviors are all reserved for a future investigation. Some results are available in [72] where it utilizes the concept of null-space to mainly prioritize behaviors. Here, we assume that behaviors are concatenated linearly as can be found in regular examples where a team of agents synchronizes with a leader.

Sequential behaviors happen one after another based on time, event or task completion such that they do not share the same time interval. The initial conditions of Beh_{i-1} comes from Beh_i if they are aware of each other, or from the last memory of Beh_{i-1} if they are not aware of each other. As a result, there might be a jump. This is depicted in Figure 5.26.

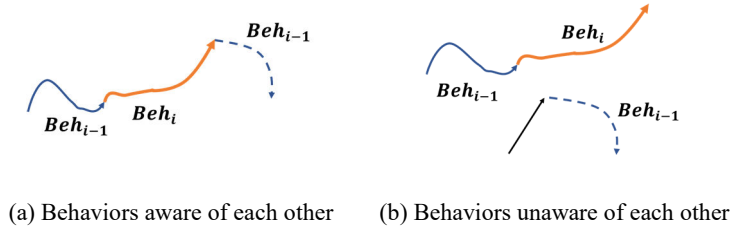


Figure. 5.26. Difference between parallel and sequential behaviors.

Remark 5.20: In the coming examples, we assume that all agents are pinned to the leader so that they are all aware of its position, and the leader tracking error is ignored. Moreover, the tracking errors used are related only to the containment behavior.

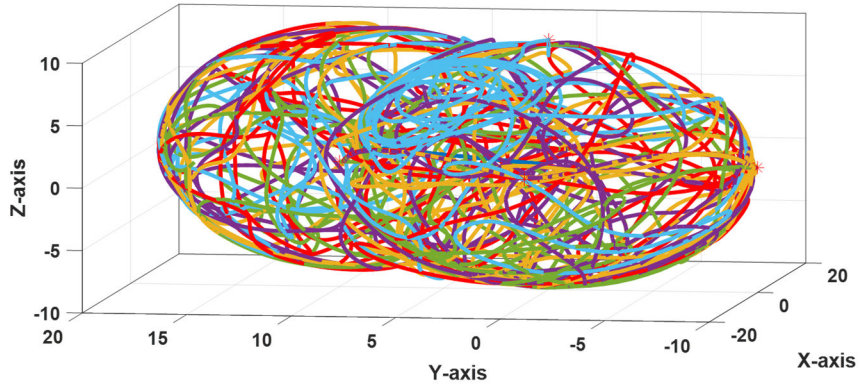
5.6.1.1: Dynamic containment-escorting behaviors

Interestingly, if (5.58.3) is parameterized using several parameters, i.e., $\{a_{xx}, a_{yy}, a_{zz}, a_x, a_y, a_z\} \geq 0 \in \mathbb{R}$, that correspond to a set of behaviors, then it can be given as follows:

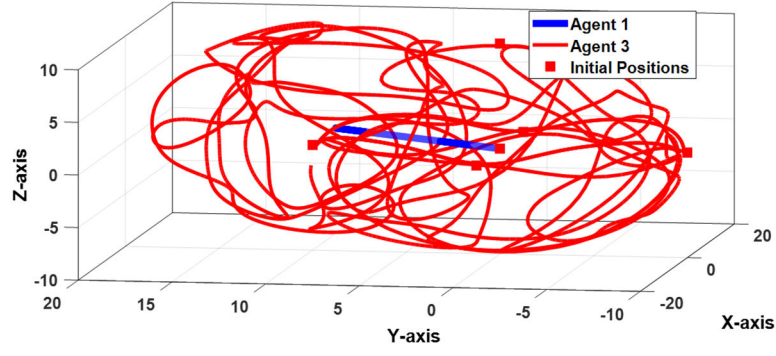
$$\vec{\dot{Q}} = - \begin{bmatrix} a_{xx}I & -a_z D^z & a_y D^y \\ a_z D^z & a_{yy}I & -a_x D^x \\ -a_y D^y & a_x D^x & a_{zz}I \end{bmatrix} \begin{bmatrix} \beta_f^x \mathcal{L}^x(\vec{\Gamma}^x) \vec{\Gamma}^x \\ \beta_f^y \mathcal{L}^y(\vec{\Gamma}^y) \vec{\Gamma}^y \\ \beta_f^z \mathcal{L}^z(\vec{\Gamma}^z) \vec{\Gamma}^z \end{bmatrix} \quad (5.58.4)$$

where: $\beta_f^x, \beta_f^y, \beta_f^z > 0$ are real constant scalars that determines the rate of convergence in each direction and $D^\tau = \text{diag}(\vec{\Gamma}^\tau)$, $\tau = \{x, y, z\}$.

If $\{a_{xx}, a_{yy}, a_{zz}\} = 0$ and $\{a_x, a_y, a_z\} = 1$, then dynamic containment-escorting behavior will happen as shown in Figure 5.27. If $\{a_{xx}, a_{yy}, a_{zz}, a_z\} = 0$ and $\{a_x, a_y\} = 1$, then dynamic containment-escorting behavior will happen while agents 2,3,4 and 5 move on a cylindrical surface about the z-axis as shown in Figure 5.28. If $\{a_{xx}, a_{yy}, a_{zz}, a_x\} = 0$ and $\{a_y, a_z\} = 1$, then dynamic containment-escorting behavior will happen while only agents 4 and 5 move as shown in Figure 5.29.

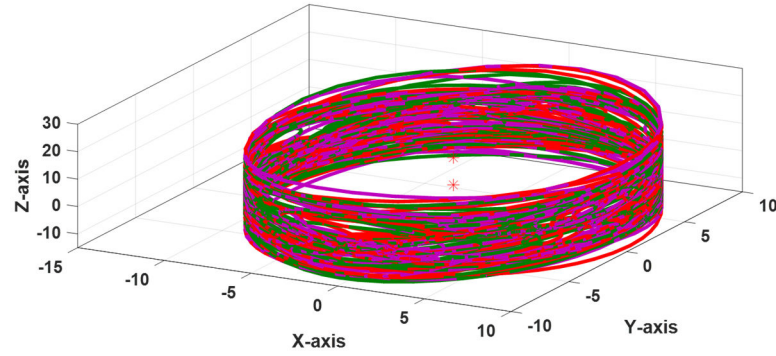


(a)

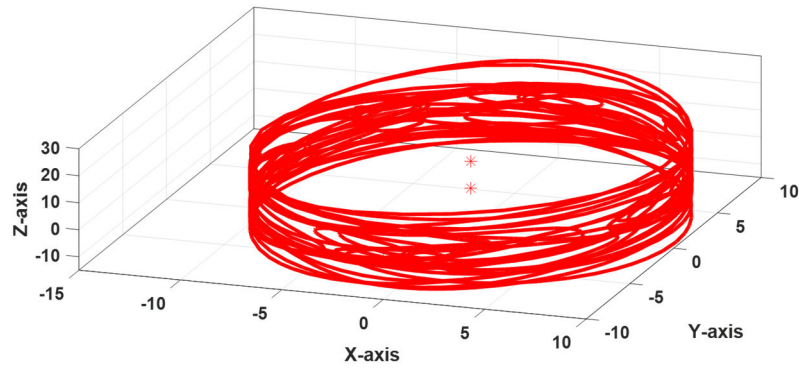


(b)

Figure. 5.27. The 3-D dynamic containment-escorting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ for all networks. (a): ideal trajectories of all agents. (b): ideal trajectories of agent 3 while escorting agent 1 combining both containment and escorting behaviors.

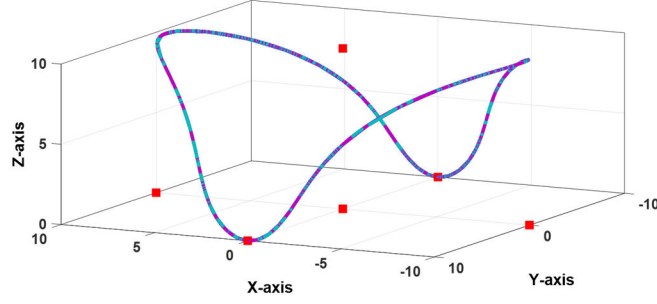


(a)

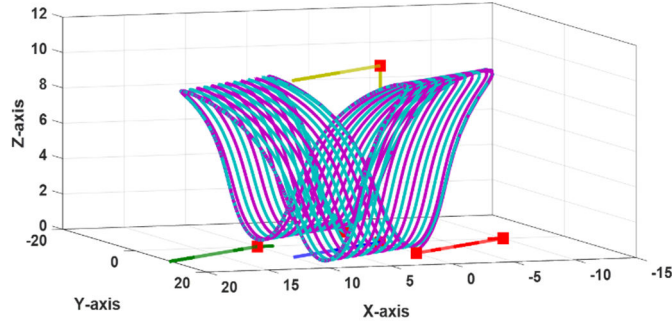


(b)

Figure. 5.28. The 3-D dynamic containment-escorting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ for all networks $\{a_{xx}, a_{yy}, a_{zz}, a_z\} = 0$ and $\{a_x, a_y\} = 1$. (a) ideal trajectories of all agents. (b) ideal trajectories of agent 3.



(a)



(b)

Figure. 5.29. Alternating ideal trajectories of agents 4 and 5 during the 3-D dynamic containment-escorting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ for all networks $\{a_{xx}, a_{yy}, a_{zz}, a_x\} = 0$ and $\{a_y, a_z\} = 1$. (a) ideal trajectory signals. (b) actual agents trajectories.

In general, many other behaviors can be generated using (5.58.4) by allowing $\{a_{xx}, a_{yy}, a_{zz}\} \in \{0, 1\}$ and $\{a_x, a_y, a_z\} \in [0, 1]$ and they are not reported here due to compactness issues.

It is obvious that once **Remark 5.1** was implemented, the modified g_{ij} functions affected the generated behaviors. However, there is a need to have some feedback from the driven agent such that the trajectory-generator system can accommodate the agent dynamical limitations. So, modifying (5.26), we may rewrite (5.58.2) as follows:

$$u_{xi} = \left(\frac{\beta_{fi}^x}{\rho_i |e_i^x| + 1} \right) \sum_{j \in \mathcal{N}_i^x} g_{ij}^x(\cdot, \cdot) \{\Gamma_j - \Gamma_i\} \quad (5.58.5)$$

where: $e_i^x = \Gamma_i - x_i$ is the tracking error from the previous time instance where $e_i^x(t_0) = 0$, see Figure 5.1, and $\beta_{fi}^x > 0$ is a real constant. The parameter $\rho_i \geq 0$ can be used by the i^{th} agent itself or a *performance monitoring unit* to remove the agent from the team; to avoid severe deterioration in the overall performance of the MAS while giving the chance to the erroneous agent to track the changes- if it is connected- and to pick the instance to rejoin the team once recovered. Consequently, ρ_i will be reset to an appropriate value. Both u_{yi} and u_{zi} can be modified accordingly.

Remark 5.21: The added tracking error term in (5.58.5) is inversely-proportioned to the tracking error in the previous time instance and thus changes the speed of the currently generated trajectory. This indirectly makes all agents aware of the slowest agent in their neighborhood. Note that this awareness does not require agents to know the actual dynamics of the slowest one which could be of different dynamics, in general.

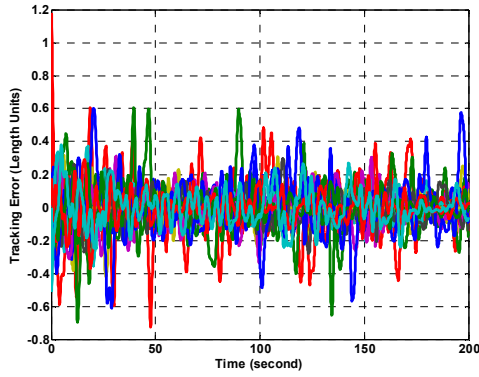
Using (5.58.5), we can rewrite (5.58.4) as follows:

$$\vec{Q} = - \begin{bmatrix} a_{xx}I & -a_zD^z & a_yD^y \\ a_zD^z & a_{yy}I & -a_xD^x \\ -a_yD^y & a_xD^x & a_{zz}I \end{bmatrix} \begin{bmatrix} \beta^x \mathcal{L}^x(\vec{\Gamma}^x) \vec{\Gamma}^x \\ \beta^y \mathcal{L}^y(\vec{\Gamma}^y) \vec{\Gamma}^y \\ \beta^z \mathcal{L}^z(\vec{\Gamma}^z) \vec{\Gamma}^z \end{bmatrix} \quad (5.58.6)$$

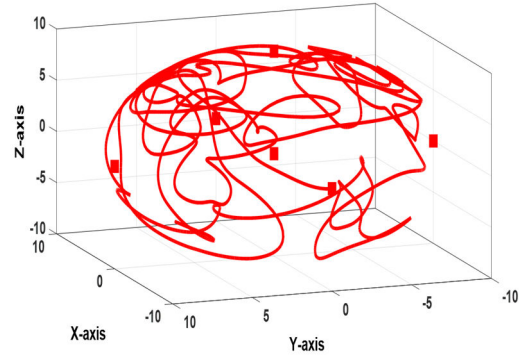
where: $\beta^\tau = \text{diag} \left(\left[\frac{\beta_{f1}^\tau}{\rho_1 |e_1^\tau| + 1}, \dots, \frac{\beta_{fN}^\tau}{\rho_N |e_N^\tau| + 1} \right] \right) > 0$ is acting as a damping term and $D^\tau = \text{diag}(\vec{\Gamma}^\tau)$ with $\tau = \{x, y, z\}$. Note that due to the available coupling, an agent motion in one direction will be affected by its inability to track the ideal trajectory in the remaining directions.

Remark 5.22: Note that the tracking error of the dynamic containment behavior was easy to derive since we assume the states to be fully available.

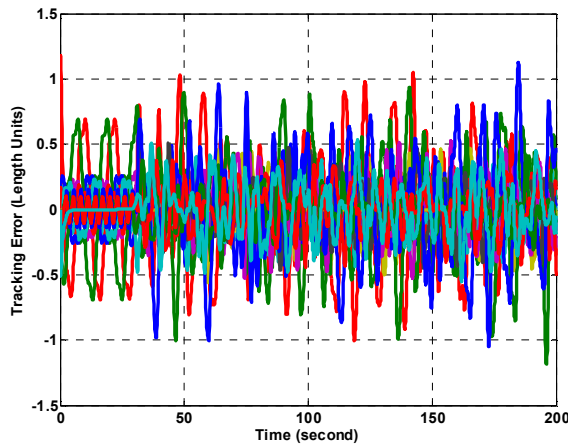
Now, let us assume six nonidentical scalar systems given by (5.1) under the control of (5.58.4) and (5.58.6) to assess the difference. Let the agents dynamics given by (5.1) such that $\{a_1, a_2, a_3, a_4, a_5, a_6\} = \{-2, -1, -3, -0.4, -0.8, -1.2\}$ and $\{b_1, b_2, b_3, b_4, b_5, b_6\} = \{0.2, 0.8, 0.5, 0.2, 0.3, 0.1\}$. The local controllers' gains were all zeros; since agents are stable, while the tracking gains were all set to 20. Figure 5.30 shows the effect of the damping term used in (5.58.5) where the tracking error was reduced as can be seen from Figure 5.29.a and c. However, agent 3 was unable to cover the same regions of the working space as can be seen from figures 5.30.b and 5.23.b.



(a)



(b)



(c)

Figure. 5.30. Resulting trajectories during containment behavior. (a) tracking error of all agents under (5.58.6). (b) actual trajectories of agent 3 under (5.58.6). (c) tracking errors of all agents under (5.58.4).

Remark 5.23: The previous example should not be considered as a deficiency in the proposed approach. Rather it should be used to emphasize the necessity for a careful selection of agents, mainly their dynamics, such that the intended behavior is feasible. If specific initial conditions are needed before a behavior is started, then a formation behavior should precede that behavior such that the needed agents' positions are achieved. This is an example of sequential behaviors.

5.7 Multi-agent Systems of N-Identical affine-in-input nonlinear systems

In this section, a MAS consisting of N-identical non-scalar affine-in-input nonlinear systems will be presented. Suppose that we have the following systems given by:

$$\dot{\vec{q}}_i = \vec{f}(\vec{q}_i) + B(\vec{q}_i)\vec{u}_i(t) \quad (5.59)$$

where: $\vec{q}_i \in \mathbb{R}^n$ and $\vec{u}_i \in \mathbb{R}^m$.

If $\vec{f}(\vec{q}_i)$ is given or desired in a LTI form, and the control input is designed as shown in (5.58.5), then a modified version of (5.58.6) that includes the additional drift terms, i.e., $\vec{f}(\vec{q}_i)$, can be given as follows:

$$\vec{Q} = - \left\{ A + B \begin{bmatrix} \beta^x \mathcal{L}^x(\vec{\Gamma}^x) & [0] & [0] \\ [0] & \beta^y \mathcal{L}^y(\vec{\Gamma}^y) & [0] \\ [0] & [0] & \beta^z \mathcal{L}^z(\vec{\Gamma}^z) \end{bmatrix} \right\} \begin{bmatrix} \vec{\Gamma}^x \\ \vec{\Gamma}^y \\ \vec{\Gamma}^z \end{bmatrix} \quad (5.59.1)$$

where:

$$A = \begin{bmatrix} [0] & -\hat{a}_z I & \hat{a}_y I \\ \hat{a}_z I & [0] & -\hat{a}_x I \\ -\hat{a}_y I & \hat{a}_x I & [0] \end{bmatrix}, B = \begin{bmatrix} a_{xx} I & -a_z D^z & a_y D^y \\ a_z D^z & a_{yy} I & -a_x D^x \\ -a_y D^y & a_x D^x & a_{zz} I \end{bmatrix}$$

with $D^\tau = \text{diag}(\vec{\Gamma}^\tau)$ with $\tau = \{x, y, z\}$, and $\{\hat{a}_x, \hat{a}_y, \hat{a}_z\} \in \{0, 1\}$ are constants. When $\{\hat{a}_x, \hat{a}_y, \hat{a}_z\}$ are taken as $\{0, 0, 1\}$, then orbiting about the z-axis is achieved.

5.7.1: Dynamic containment-escorting-orbiting behaviors

Note that if A in (5.59.1) is Hurwitz, then the trivial consensus, i.e., agents will approach the origin, will be achieved while agents moving along spiral trajectories as shown in Figure 5.31. If agent 1 is initially moving or at a location other than the origin, then the same behavior can be achieved by simply subtracting its initial position from the positions of other agents once the behavior is initiated, see **Remark 5.20**. The result is shown in Figure 5.31. The results shown in figures 3.31 and 3.32 were obtained using $\{a_{xx}, a_{yy}, a_{zz}\} = -1$, $\{a_x, a_y, a_z\} = 0$, and $\{\hat{a}_x, \hat{a}_y, \hat{a}_z\} = \{0, 0, 1\}$.

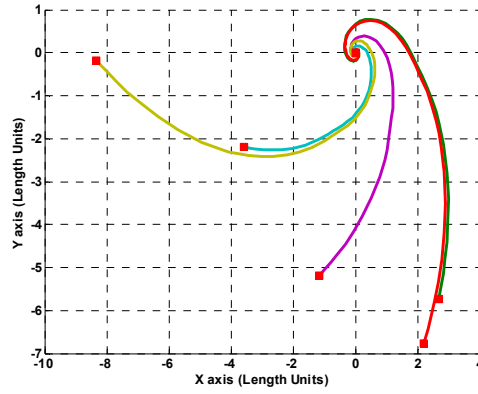
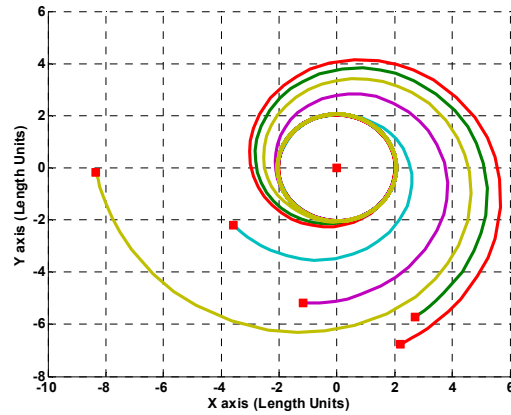
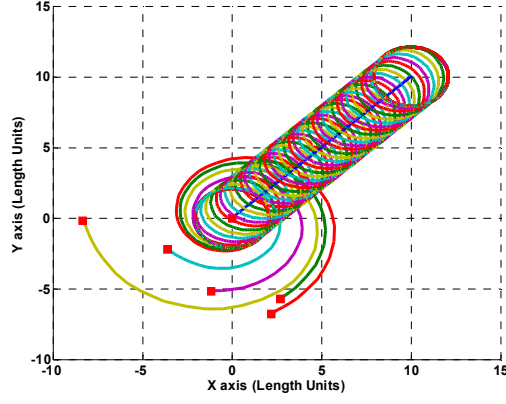


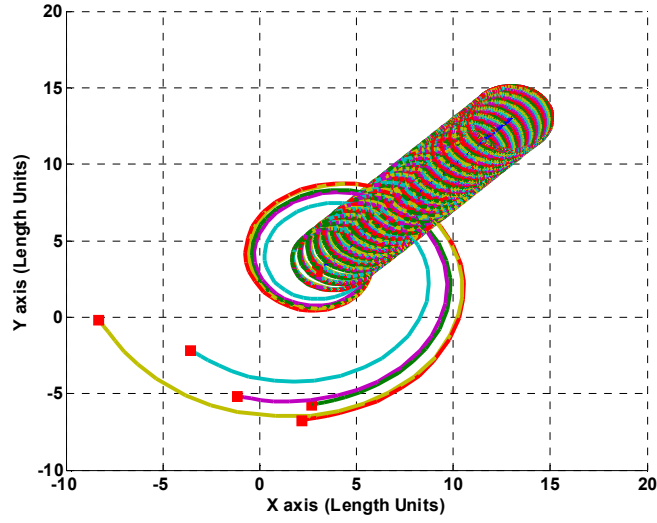
Figure. 5.31. The 3-D dynamic containment-escorting-orbiting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ when A in (5.59.1) is Hurwitz.



(a)



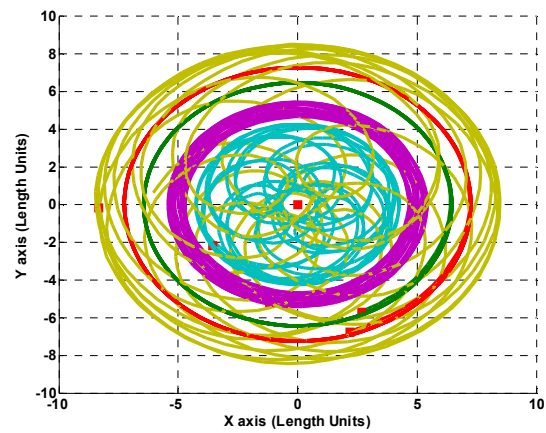
(b)



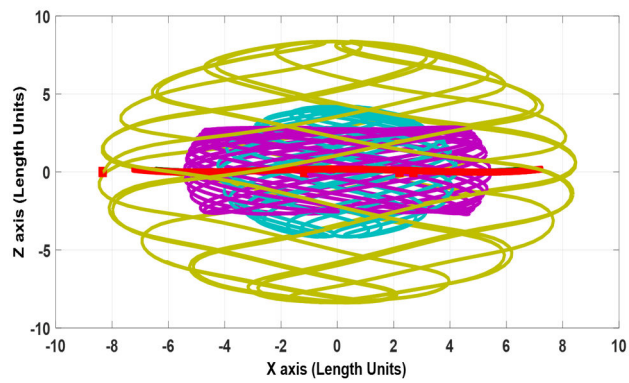
(c)

Figure. 5.32. The 3-D dynamic containment-escorting-orbiting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ and additional damping given by (5.59.1). (a) containment-orbiting trajectory-generator signals. (b) actual trajectory of all agents during the complete behavior when agent 1 is initially at the origin. (c) actual trajectory of all agents during the complete behavior when agent 1 is initially away from the origin.

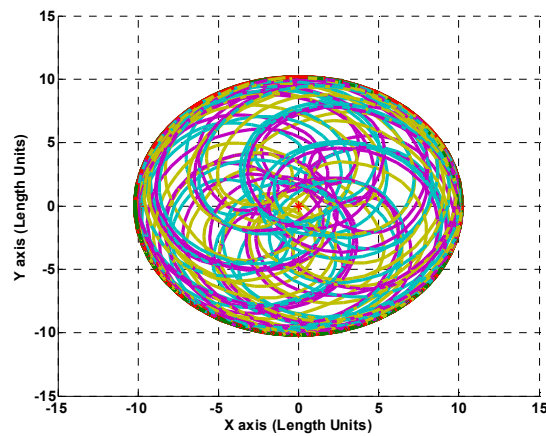
Figure 5.33 shows the results of the dynamic containment-orbiting behavior when the parameters are set as follows: $\{a_{xx}, a_{yy}, a_{zz}\} = 0$, $\{a_x, a_y, a_z\} = 1$, and $\{\hat{a}_x, \hat{a}_y, \hat{a}_z\} = \{0, 0, 1\}$. Note that there was no collision among agents during conducting this behavior, and the minimum distance among agents was 14.0288.



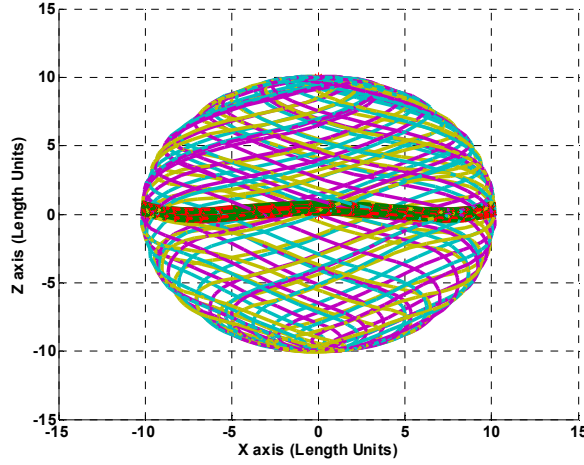
(a)



(b)



(c)



(d)

Figure. 5.33. The 3-D dynamic containment-orbiting behavior with $g_{ij}(\Gamma_i, \Gamma_j)$ and additional damping given by (5.59.1) when agents are spaced on a hemisphere. (a)-(b) not equally or ellipsoid like. (c)-(d) equally or circle like.

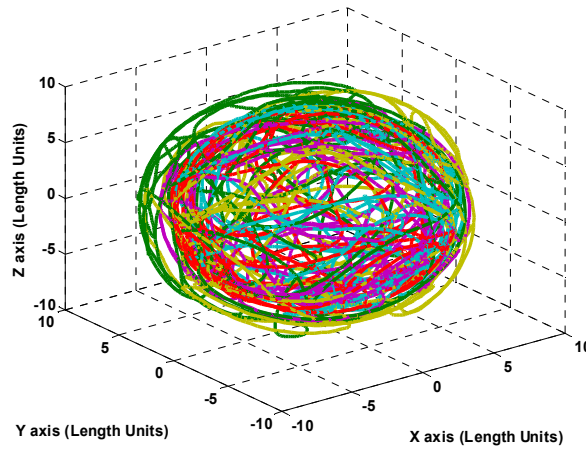
Remark 5.24: The effect of the drift term in (5.59) can be observed by comparing the results shown in figures 5.23.a and 5.33.

5.7.1.1: Shaping the Dynamic containment-escorting-orbiting behaviors using sensory-input

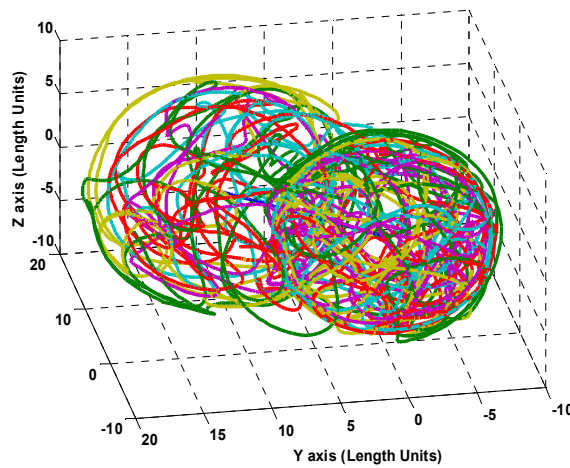
In a previous section, we have discussed mapping the environment into the trajectory-generator Kino-dynamics using HPFs, and two examples where provided. Here, in this subsection, we will demonstrate the usefulness of the sensory-input in shaping the overall behaviors and in triggering a background-running behavior to respond to certain situations. The kinematic version of the trajectory-generator systems will be used in the example provided.

Now, it is desired to show how using (5.43) can upgrade the behavior obtained using (5.59.1)- for example- such that agents avoid colliding with obstacles based on sensory-input available onboard. Figure 5.34 shows the 3-D dynamic containment-escorting-orbiting behavior while being affected by a moving obstacle wandering in the working

space. In Figure 5.34, onboard sensors provide readings about the obstacle position for the agent and fed them into an on-the-fly modelling capability that converts these positions into forces related to HPFs- as shown in (5.43)- which can then be localized as shown in Figure 5.14. However, if these readings are related to a target, then instead of avoiding that object it is intended to target it. This is demonstrated as follows:



(a)

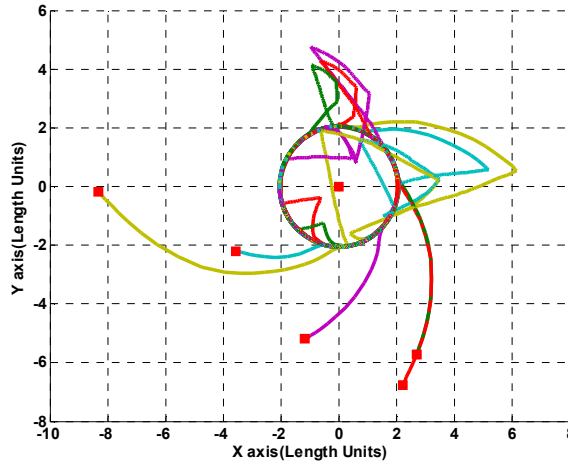


(b)

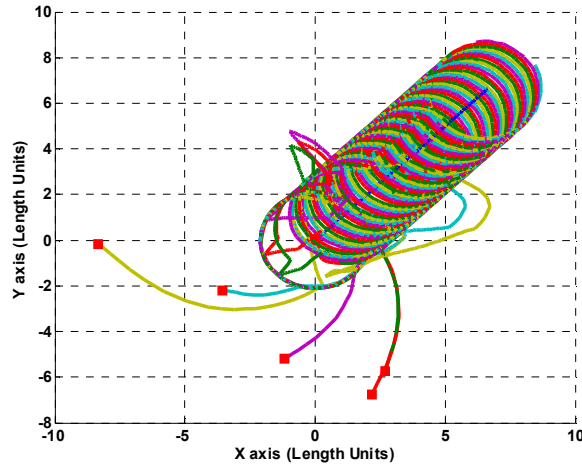
Figure. 5.34. Results of dynamic containment-escorting-orbiting during dynamical single-point-obstacle avoidance. (a) ideal trajectories of all agents due to containment-orbiting trajectory-generator system. (b) actual trajectory of all agents during the complete behavior.

Figure 5.35 shows an attack-protect scheme where J -cooperating or non-cooperating point attackers attack an asset in an unplanned manner, i.e., no attacking plan is used. The attackers try to hit the asset and at the same time evade the guardians whose responsibilities are to protect, contain and escort the asset to a safe zone. Guardians start by performing the 3-D dynamic containment which will be abandoned by a guardian once an attacker is detected. The idea here is to have at least two evolving behaviors within the trajectory-generator co-system within a guardian such that switching among them result in the overall response to certain context. In contrast to the attackers, guardians utilize attractive artificial forces that ensure capturing the attackers while attackers utilize repulsive artificial forces to evade the guardians.

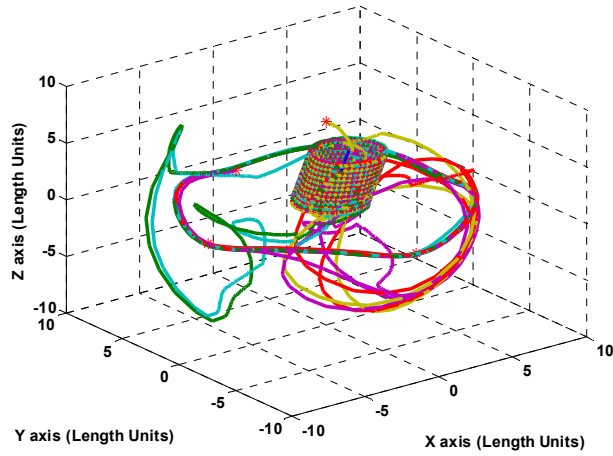
Remark 5.25: In Figure 5.35, attacker-guardian pairs are established based on minimum separation. However, more intelligence can be introduced to control the attacker-guardian relations especially if there are more attackers than guardians.



(a)



(b)



(c)

Figure. 5.35. Results of unplanned attack of four agents targeting an asset protected by dynamic containment-escorting-orbiting utilizing five agents. (a): ideal trajectories in 2-D. (b): actual trajectories in 2-D. (c): actual trajectories of 3-D followed by 2-D dynamic containment-escorting-orbiting behaviors based on the time elapsed.

5.7.2: Non-holonomic wheeled robots

In this subsection, we present the results obtained in guiding a group of six front-wheel steered robots (FSR) to achieve average consensus in a working space containing stationary obstacles. The kinematical trajectory-generator (5.33) was used with the following synchronizing error:

$$\alpha_i(e_i) = \beta_f e_i = \beta_f (x_i - \Gamma_i), \quad \beta_f \gg 1 \quad (5.60)$$

The synchronizing error signal plays a significant role in stimulating the trajectory-generator system to consider the robot motion and capabilities, and therefore generates suitable trajectories that the robot can follow.

A simplified model for FSR robots is given as follows [117]:

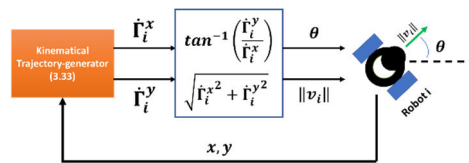
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (5.61)$$

Where: θ is the heading, v is the linear velocity and ω is the angular velocity of the wheels.

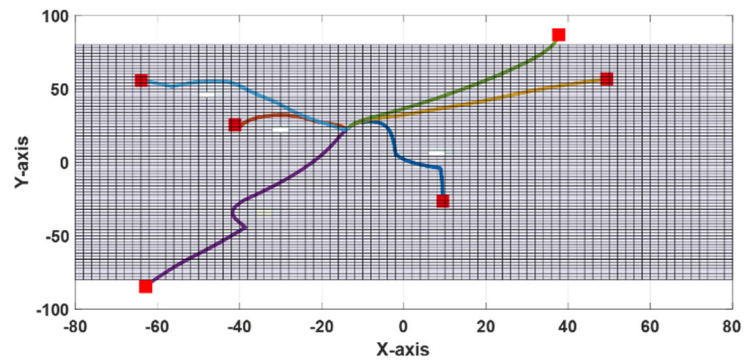
The integration between the trajectory-generator system and the i^{th} FSR is depicted in Figure 5.36.a where a simple nonlinear transformation is used to generate the input signals needed for the FSR, namely: the speed and heading. This setup is like what one may experience when commanding a radio-controlled ground vehicle where the operator acts like the trajectory-generator system in our setup presented while the radio kit is like the nonlinear transformation used. The results are shown in Figure 5.36, in which the following consensus value was obtained: $\{-13.6564, 22.9838, 0\}$, while the actual average value of robots' initial conditions was $\{-11.8753, 18.8720, 0\}$.

Remarkably, the nonlinear transformation used in Figure 5.36.a is another example of possible coupling and transformations that can be used in the designed trajectory-generator systems to facilitate their functionality. Compare it to the coupling shown in (5.58.6), for example. The chattering in the control signal shown in Figure 5.36.e is mainly due to the localization of obstacles potential field.

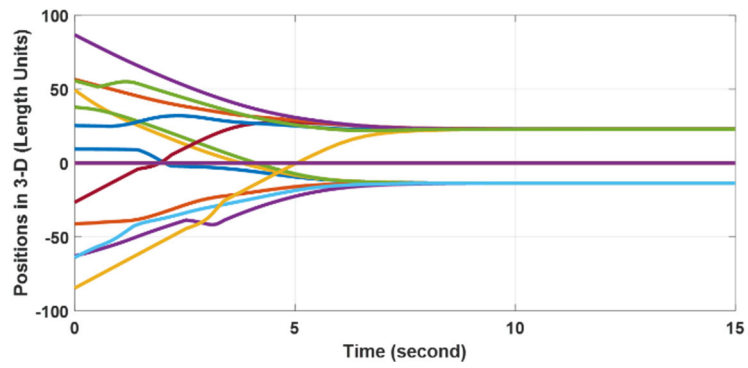
Note 5.3: It is worth noting that other behaviors besides the ones presented here are also feasible. See, Chapter 4 for more information.



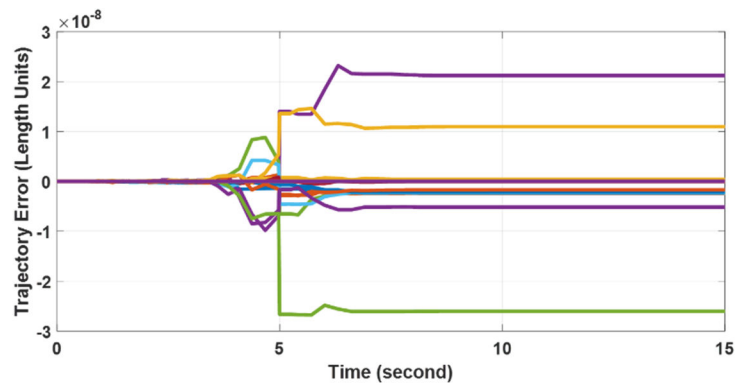
(a)



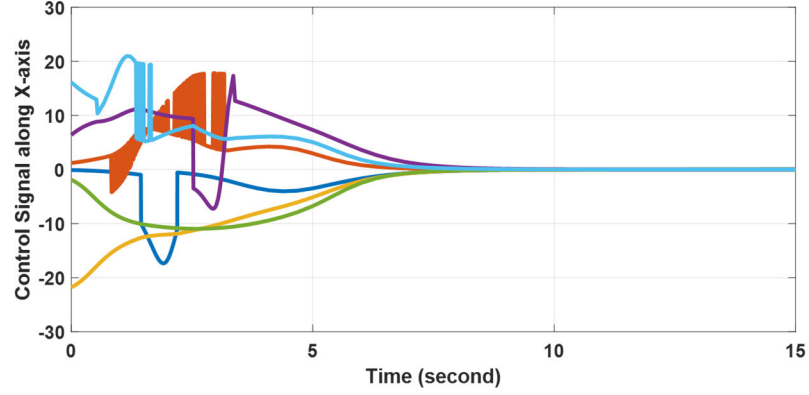
(b)



(c)



(d)



(e)

Figure. 5.36. Simulation results of guiding a group of six FSR in a working space that contains stationary obstacles. (a) trajectory-generator, nonlinear transformation and FSR integration. (b) top-view of working space and FSR actual trajectories. (c) FSR positions versus time. (d) tracking error of order 10^{-8} . (e) control signal along the x-axis.

5.7.3: Under-actuated Flying robots

In this subsection, we present the results obtained in guiding a group of six under-actuated flying robots, namely: quadrotors. In this example we will emphasize on using the kinodynamical trajectory-generator system as an ultimate result of the proposed framework in this thesis. Also, this example will aid in appreciating our philosophy which entails using an idealized trajectory-generator system to steer various realistic systems.

Previously, in equation (5.59.1), the position command signal was generated. However, in more realistic situations, the velocity command is also needed; since most realistic robots can be approximated or are of second order dynamics. So, taking the time-derivative of (5.59.1) yields the following- when $\beta^{x,y,z}$ are fixed constants:

$$\begin{bmatrix} \vec{\dot{Q}} \\ \vec{\ddot{Q}} \end{bmatrix} = \begin{bmatrix} M_1 & [0] \\ M_2 & M_1 \end{bmatrix} \begin{bmatrix} \vec{Q} \\ \vec{\dot{Q}} \end{bmatrix} \quad (5.62)$$

where: M_1 is given in (5.59.1) and M_2 is given as follows:

$$M_2 = \begin{bmatrix} -a_{xx}\beta^x \bar{\mathcal{L}}^x(\vec{\Gamma}^x) & a_z\beta^y \{D^z \mathcal{L}^y + D^z \bar{\mathcal{L}}^y\} & -a_y\beta^z \{D^y \mathcal{L}^z + D^y \bar{\mathcal{L}}^z\} \\ -a_z\beta^x \{D^z \mathcal{L}^x + D^z \bar{\mathcal{L}}^x\} & -a_{yy}\beta^y \bar{\mathcal{L}}^y(\vec{\Gamma}^y) & a_x\beta^z \{D^x \mathcal{L}^z + D^x \bar{\mathcal{L}}^z\} \\ a_y\beta^x \{D^y \mathcal{L}^x + D^y \bar{\mathcal{L}}^x\} & -a_x\beta^y \{D^x \mathcal{L}^y + D^x \bar{\mathcal{L}}^y\} & -a_{zz}\beta^z \bar{\mathcal{L}}^z(\vec{\Gamma}^z) \end{bmatrix}$$

With: $D^\tau = \text{diag}(\tau)$, $\tau = \{x, y, z\}$ and $\bar{\mathcal{L}}^\tau = [d(g_{ij}^\tau)/dt]$.

Now, we need to ensure the stability of the kino-dynamical trajectory-generator system.

Recalling **Theorem 5.3**, we need to ensure that M_2 is bounded.

By re-examining the structure of (5.62), another representation for the x -acceleration signals can be given as follows:

$$\begin{aligned} \ddot{\Gamma}^x = & -a_{xx}\beta^x \left\{ \bar{\mathcal{L}}^x \vec{\Gamma}^x + \mathcal{L}^x \vec{\Gamma}^x \right\} + a_z\beta^y D^z \mathcal{L}^y \vec{\Gamma}^y + a_z\beta^y D^z \left\{ \bar{\mathcal{L}}^y \vec{\Gamma}^y + \mathcal{L}^y \vec{\Gamma}^y \right\} \\ & + \hat{a}_z \vec{\Gamma}^y - \hat{a}_y \vec{\Gamma}^z - a_y\beta^z D^y \mathcal{L}^z \vec{\Gamma}^z - a_y\beta^z D^y \left\{ \bar{\mathcal{L}}^z \vec{\Gamma}^z + \mathcal{L}^z \vec{\Gamma}^z \right\} \end{aligned} \quad (5.63)$$

Now, we would like to simplify the terms $\left\{ \bar{\mathcal{L}}^\tau \vec{\Gamma}^\tau + \mathcal{L}^\tau \vec{\Gamma}^\tau \right\}$ such that the boundedness of M_2 is clear. This can be done by designing the g_{ij}^τ weighting functions as follows:

Using the parameterized C-S model given in (3.116), we may write the following:

$$\dot{\Gamma}_i^\tau = \sum_{j \in \mathcal{N}_i} g_{ij}^\tau \{\tau_j - \tau_i\} \quad (5.63.1)$$

where the coupling among the spatial directions is ignored for simplicity. So, we may find the second time derivative as follows:

$$\ddot{\Gamma}_i^\tau = \sum_{j \in \mathcal{N}_i} \{\tau_j - \tau_i\} \frac{dg_{ij}^\tau}{dt} + \sum_{j \in \mathcal{N}_i} g_{ij}^\tau \{\dot{\tau}_j - \dot{\tau}_i\} \quad (5.63.2)$$

where:

$$\frac{dg_{ij}^\tau}{dt} = \frac{\partial g_{ij}^\tau}{\partial \tau_i} \dot{\tau}_i + \frac{\partial g_{ij}^\tau}{\partial \tau_j} \dot{\tau}_j \quad (5.63.3)$$

Recalling (3.35) imposed by the framework, we may rewrite (5.63.3) as follows:

$$\frac{dg_{ij}^\tau}{dt} = \frac{\partial g_{ij}^\tau}{\partial \tau_i} \{\dot{\tau}_i - \dot{\tau}_j\} = \frac{2\gamma\beta H(\tau_j - \tau_i)(\dot{\tau}_i - \dot{\tau}_j)}{[\gamma(\tau_j - \tau_i)^2 + \delta]^{\beta+1}} \quad (5.63.4)$$

Substituting (5.63.4) into (5.63.2) and doing the simplification, yields:

$$\ddot{\Gamma}_i^\tau = H \sum_{j \in \mathcal{N}_i} \left\{ \frac{\gamma(1-2\beta)(\tau_j - \tau_i)^2 + \delta}{[\gamma(\tau_j - \tau_i)^2 + \delta]^{\beta+1}} \right\} (\dot{\tau}_j - \dot{\tau}_i) \quad (5.63.5)$$

Taking $\beta = 0.5$, yields:

$$\ddot{\Gamma}_i^\tau = H \sum_{j \in \mathcal{N}_i} \left\{ \frac{\delta}{[\gamma(\tau_j - \tau_i)^2 + \delta]^{1.5}} \right\} (\dot{\tau}_j - \dot{\tau}_i) = \sum_{j \in \mathcal{N}_i} \bar{g}_{ij}^\tau (\dot{\tau}_j - \dot{\tau}_i) \quad (5.63.6)$$

Consequently, the global dynamics are given as follows:

$$\ddot{\Gamma}^\tau = -\{\bar{\mathcal{L}}^\tau \dot{\Gamma}^\tau + \mathcal{L}^\tau \ddot{\Gamma}^\tau\} = -\bar{\mathcal{L}}^\tau \dot{\Gamma}^\tau \quad (5.63.7)$$

where: $\bar{\mathcal{L}}^\tau = [\bar{g}_{ij}^\tau]$.

Therefore:

$$\begin{aligned} \ddot{\Gamma}^x = & -a_{xx}\beta^x \bar{\mathcal{L}}^x \dot{\Gamma}^x + a_z\beta^y D^z \mathcal{L}^y \dot{\Gamma}^y + a_z\beta^y D^z \bar{\mathcal{L}}^y \dot{\Gamma}^y + \hat{a}_z \dot{\Gamma}^y - \hat{a}_y \dot{\Gamma}^z \\ & - a_y\beta^z D^y \mathcal{L}^z \dot{\Gamma}^z - a_y\beta^z D^y \bar{\mathcal{L}}^z \dot{\Gamma}^z \end{aligned} \quad (5.63.8)$$

The acceleration signals in the other spatial directions follow in the same manner.

So, we may rewrite M_2 as follows:

$$\bar{M}_2 = \begin{bmatrix} [0] & a_z\beta^y D^z \mathcal{L}^y & -a_y\beta^z D^y \mathcal{L}^z \\ -a_z\beta^x D^z \mathcal{L}^x & [0] & a_x\beta^z D^x \mathcal{L}^z \\ a_y\beta^x D^y \mathcal{L}^x & -a_x\beta^y D^x \mathcal{L}^y & [0] \end{bmatrix} \quad (5.63.9)$$

Recalling the preservation of angular momentum demonstrated in (5.58.1) and the boundedness of the control input under the C-S model with $\beta = 0.5$, it is straightforward

to conclude that \bar{M}_2 is indeed bounded. Therefore, the kino-dynamical trajectory generator is also stable.

5.7.3.1: Introducing the error signals

Previously, when deriving (5.62) we assumed $\beta^{x,y,z}$ to be fixed constants. However, in previous sections we have demonstrated the usefulness of introducing the tracking error signals to activate or inhibit the idealized trajectory-generator signals such that they accommodate the dynamics of the steered agents. Special importance arises when considering the multiplicative tracking error given in (5.58.5); since its time derivative will appear in the acceleration signal generated in the kino-dynamical trajectory-generator proposed and therefore may affect its stability if the local tracking controller fails to bound the magnitude of the tracking error. This is investigated as follows:

Taking the time derivative of β^τ given in (5.58.6), yields:

$$\dot{\beta}^\tau \equiv \frac{d\beta^\tau}{dt} = -\rho \beta_f^\tau \text{diag} \left(\frac{\text{sign}(e_i^\tau) \dot{e}_i^\tau}{(\rho |e_i^\tau| + 1)^2} \right) \quad (5.64)$$

where: $\rho = \rho_i \forall i$ and $\beta_f^\tau = \beta_{fi}^\tau \forall i$ are all fixed constants.

Therefore, the contribution of the multiplicative tracking error can be included in \bar{M}_2 as follows:

$$\bar{\bar{M}}_2 = \begin{bmatrix} [0] & a_z\{\beta^y D^z \mathcal{L}^y + \dot{\beta}^y D^z \mathcal{L}^y\} & -a_y\{\beta^z D^y \mathcal{L}^z + \dot{\beta}^z D^y \mathcal{L}^z\} \\ -a_z\{\beta^x D^z \mathcal{L}^x + \dot{\beta}^x D^z \mathcal{L}^x\} & [0] & a_x\{\beta^z D^x \mathcal{L}^z + \dot{\beta}^z D^x \mathcal{L}^z\} \\ a_y\{\beta^x D^y \mathcal{L}^x + \dot{\beta}^x D^y \mathcal{L}^x\} & -a_x\{\beta^y D^x \mathcal{L}^y + \dot{\beta}^y D^x \mathcal{L}^y\} & [0] \end{bmatrix} \quad (5.64.1)$$

Similarly, considering the additive tracking error can be investigated as follows:

Let the additive tracking error be given as such:

$$ae_i^\tau = \frac{\alpha(\tau_j - \tau_i)}{(\varrho(\tau_j - \tau_i)^2 + 1)^{0.5}} \quad (5.65)$$

where: $\alpha, \varrho > 0$ are fixed and real constants. So, the time derivative of the tracking error is given as follows:

$$\dot{ae}_i^\tau = \frac{\alpha(\dot{\tau}_j - \dot{\tau}_i)}{(\varrho(\tau_j - \tau_i)^2 + 1)^{1.5}} \quad (5.65.1)$$

Therefore, the overall dynamics of the kino-dynamical trajectory-generator system can be given as follows:

$$\begin{bmatrix} \vec{\dot{Q}} \\ \vec{\ddot{Q}} \end{bmatrix} = \begin{bmatrix} M_1 & [0] \\ \bar{M}_2 & M_1 \end{bmatrix} \begin{bmatrix} \vec{Q} \\ \vec{\dot{Q}} \end{bmatrix} + \begin{bmatrix} \overrightarrow{AE} \\ \overrightarrow{\dot{AE}} \end{bmatrix} \quad (5.66)$$

where: $\overrightarrow{AE} = [\overrightarrow{ae}^x, \overrightarrow{ae}^y, \overrightarrow{ae}^z]^T$ with $\overrightarrow{ae}^\tau = \text{diag}(ae_1^\tau, \dots, ae_N^\tau), \tau = \{x, y, z\}$.

Remark 5.26: During the numerical simulation of the dynamic containment behavior under the kino-dynamical trajectory-generator system given in (5.66), it was found that numerical problems were inevitable; because it was difficult to find a suitable solver/step-size combination that uniformly solves the stiff problem over the intended time interval.

Considering **Remark 5.26** and to avoid numerical problems, it is possible to rewrite (5.66) as follows:

$$\begin{bmatrix} \vec{\dot{Q}} \\ \vec{\ddot{Q}} \end{bmatrix} = \begin{bmatrix} [0] & I \\ \bar{M}_2 & M_1 \end{bmatrix} \begin{bmatrix} \vec{Q} \\ \vec{\dot{Q}} \end{bmatrix} + \begin{bmatrix} \overrightarrow{AE} \\ \overrightarrow{\dot{AE}} \end{bmatrix} \quad (5.67)$$

which was found numerically stable. The identity matrix is denoted by I .

Remark 5.27: The initial conditions of the velocity signals must be evaluated using (5.59.1).

The following example demonstrates the dynamic containment behavior under (5.67) where a group of six quadrotors are used.

Example 5.3: Consider a MAS comprising five quadrotors where they must contain the sixth robot dynamically in the 3-D obstacle-free space. Both the interaction and communication graphs are depicted in Figure 5.37. The local nonlinear tracking controllers are designed according to [174] and the Simulink model is shown in Figure 5.38.

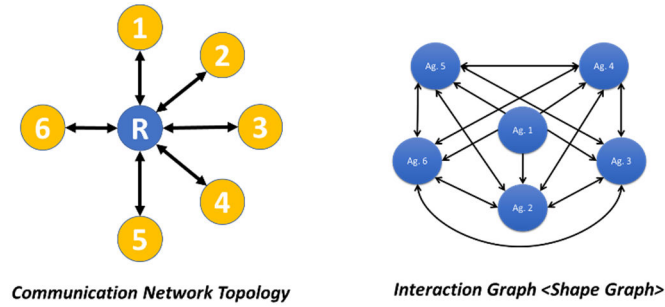
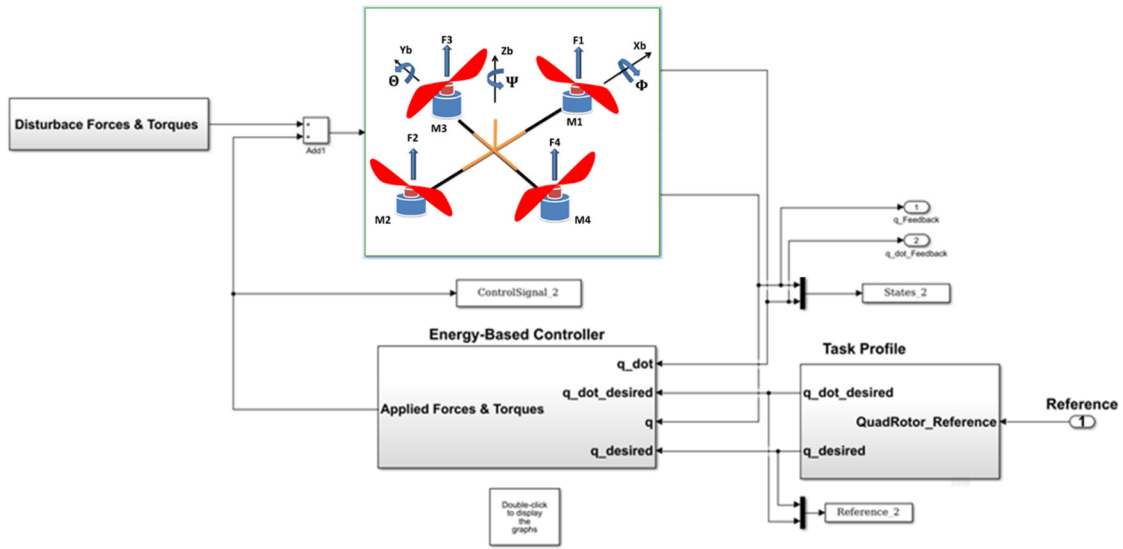
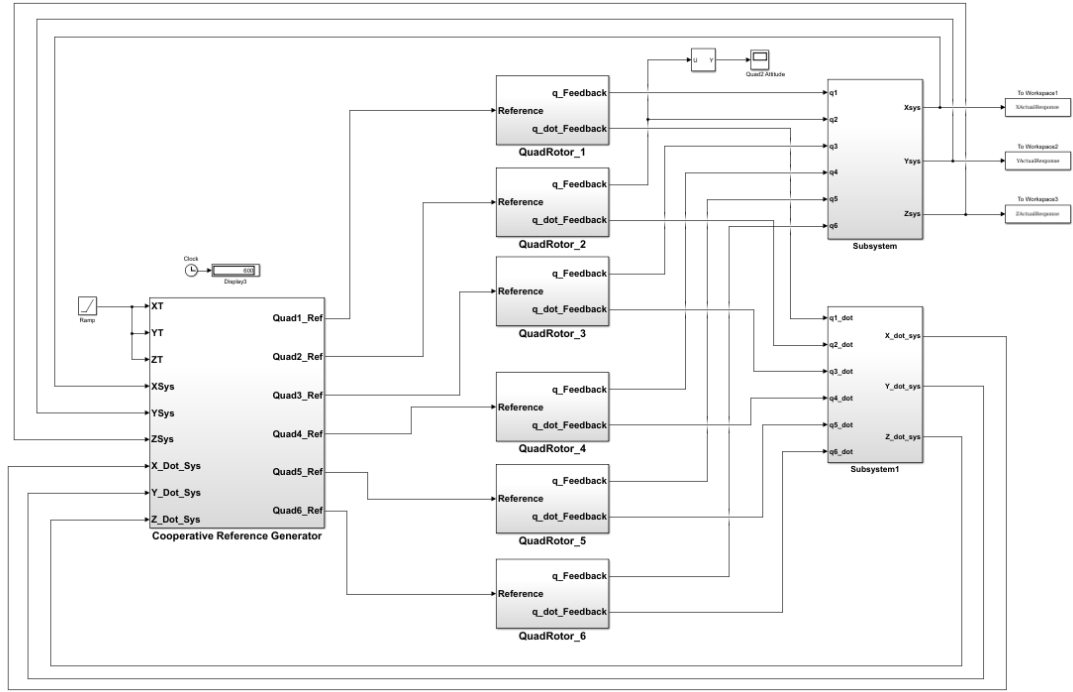


Figure. 5.37. The communication and interaction graph used in quadrotors' simulation example.



(a)



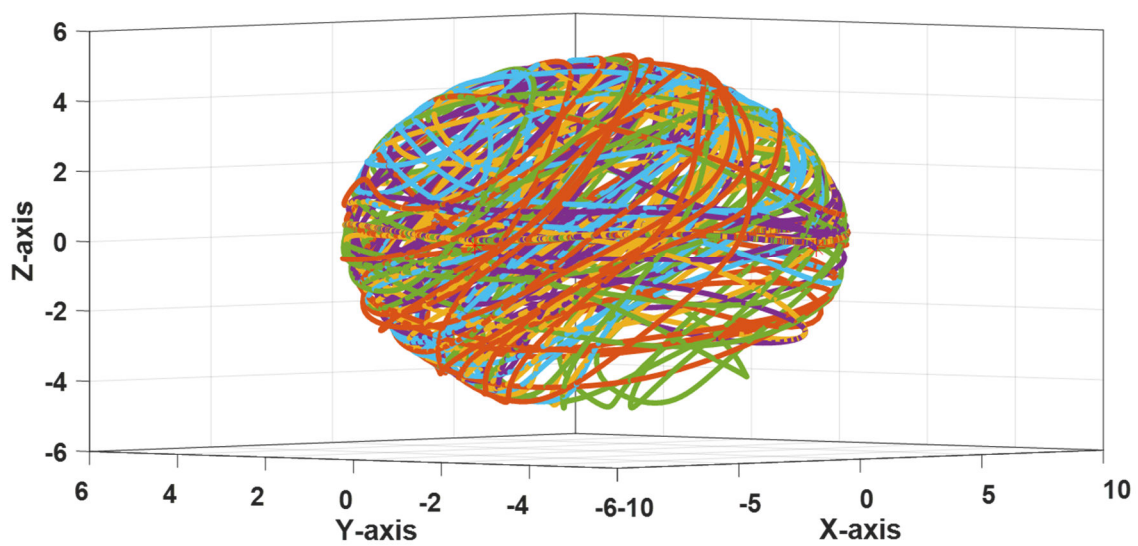
(b)

Figure. 5.38. The Simulink model used in simulating the dynamic containment behavior under (5.67).

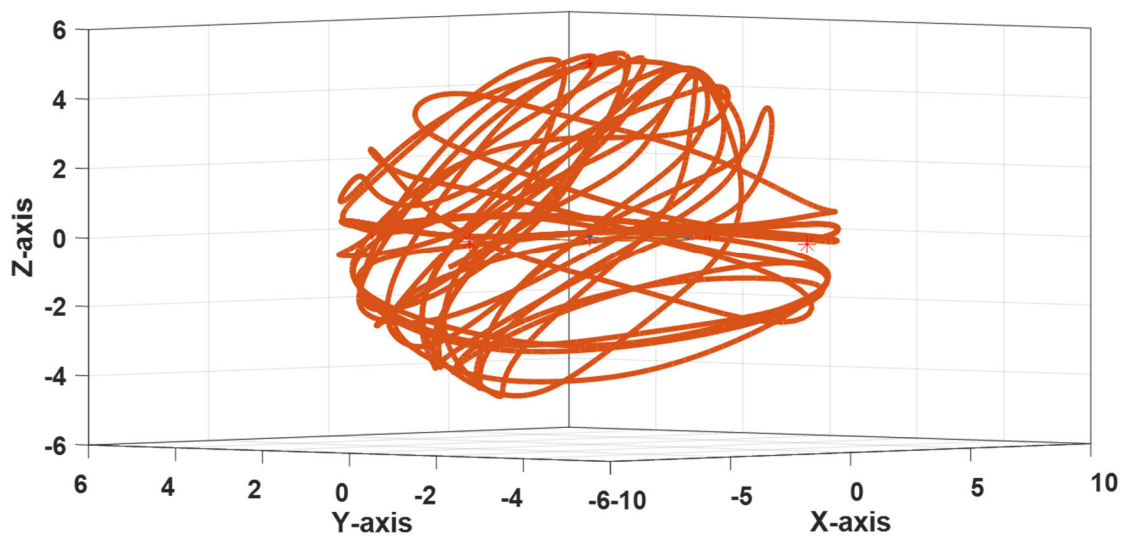
Next, the simulation results of the above setup will be presented for mainly four cases, namely: without tracking errors involved, with additive tracking error only, with multiplicative tracking error only and finally with both types of tracking errors involved. The effect of introducing the tracking errors can be read from generated trajectories and the magnitude of the tracking errors involved.

Case 1: No tracking errors are involved:

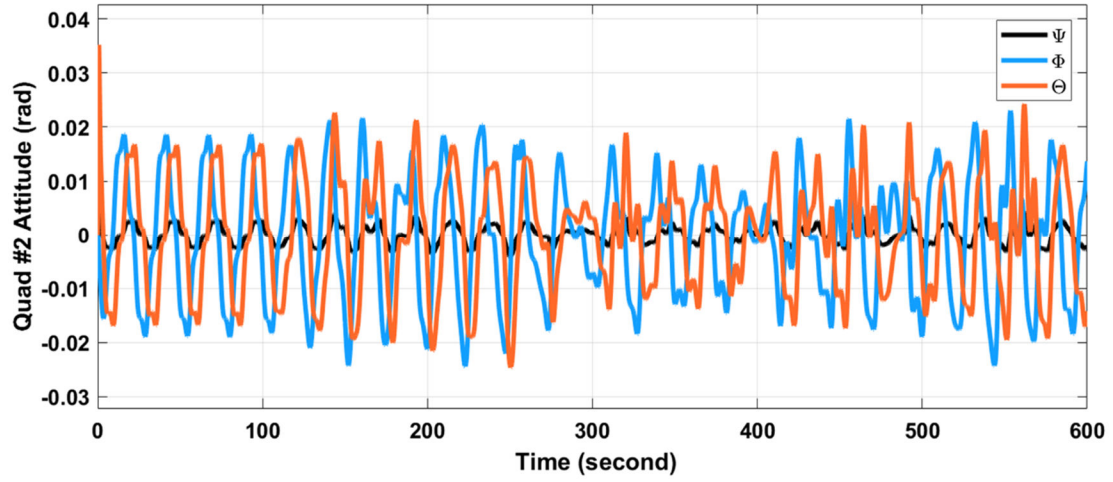
The simulation results of this case are shown in figures 5.39 and 5.40.



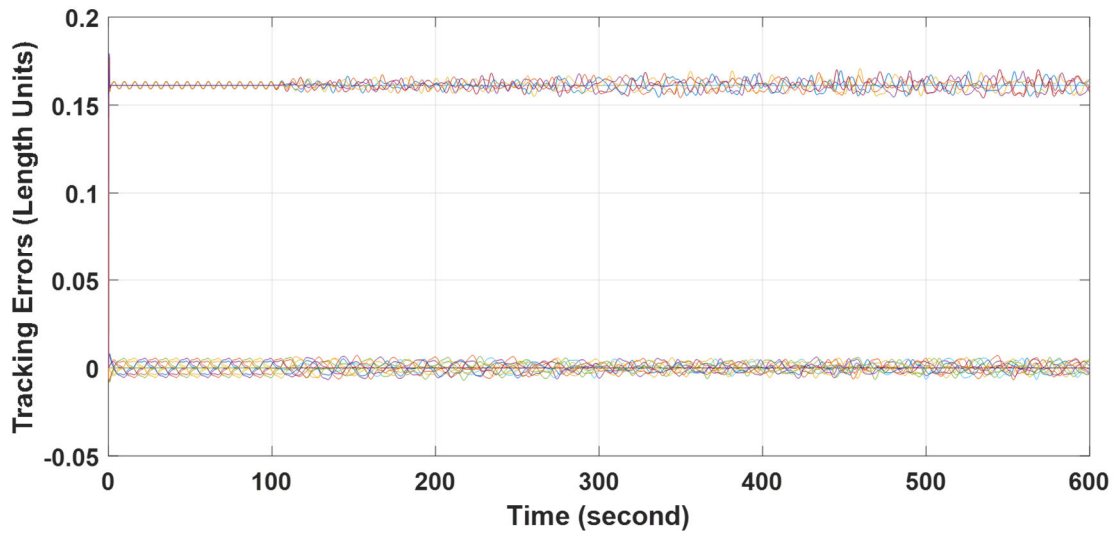
(a)



(b)

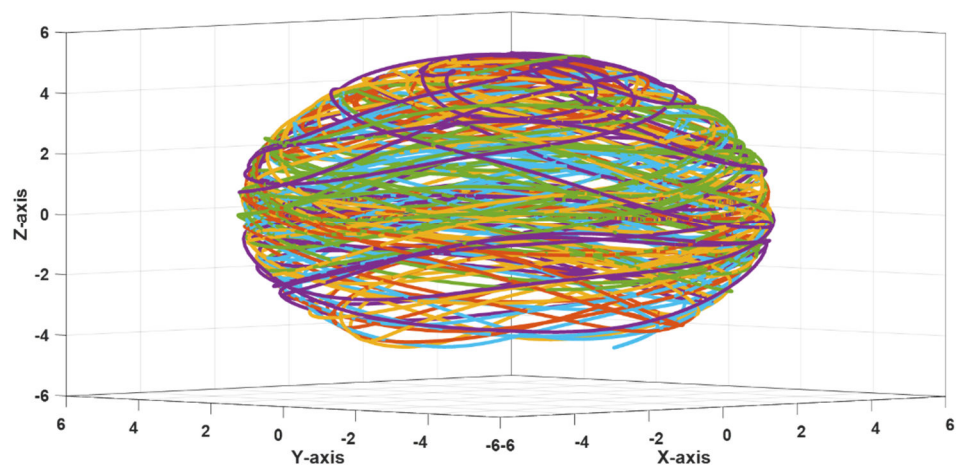


(c)

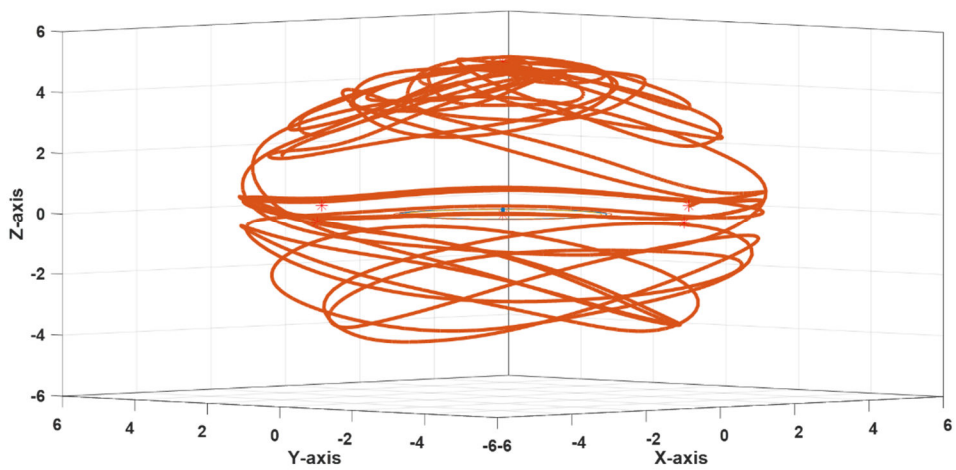


(d)

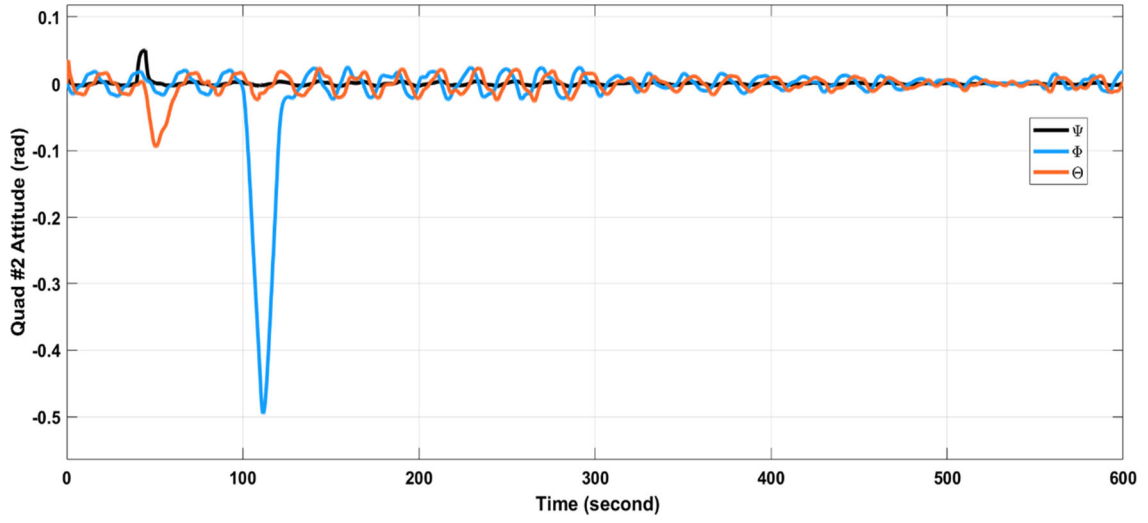
Figure. 5.39. The simulation results for case 1 without disturbances acting on quadrotor 2. (a) Actual 3-D trajectories for all agents. (b) Actual 3-D trajectories for the second quadrotor. (c) its attitude. (d) the tracking errors.



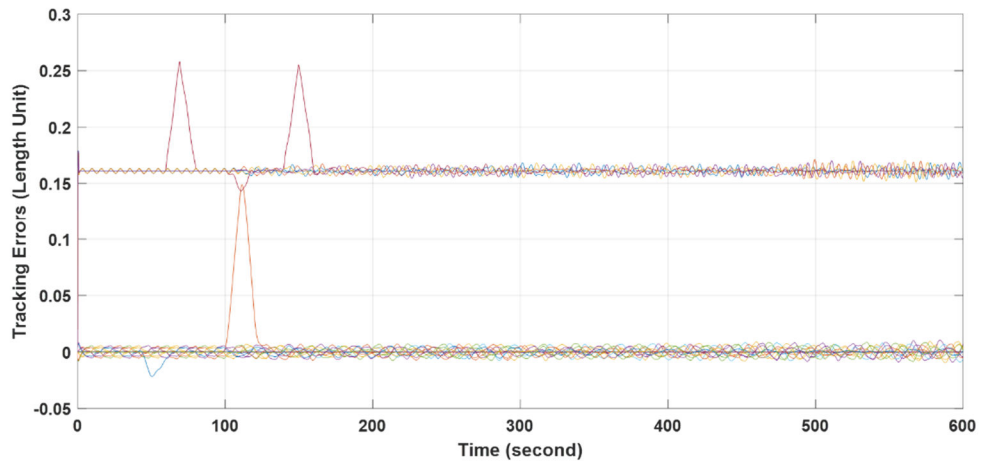
(a)



(b)



(c)

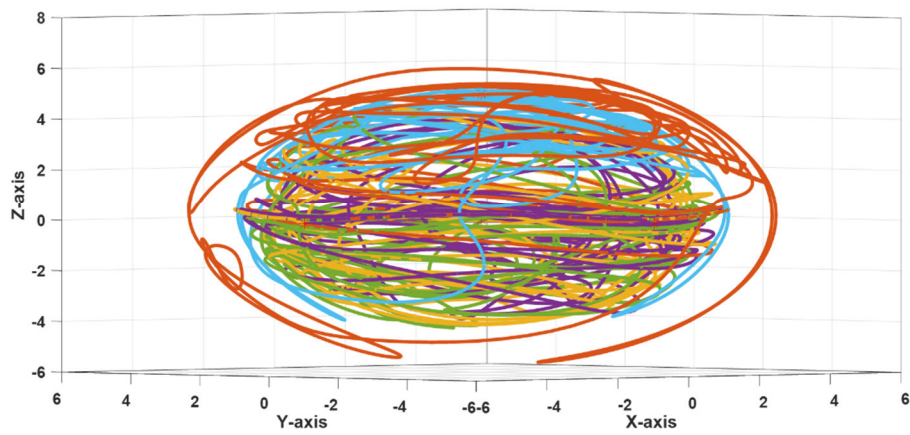


(d)

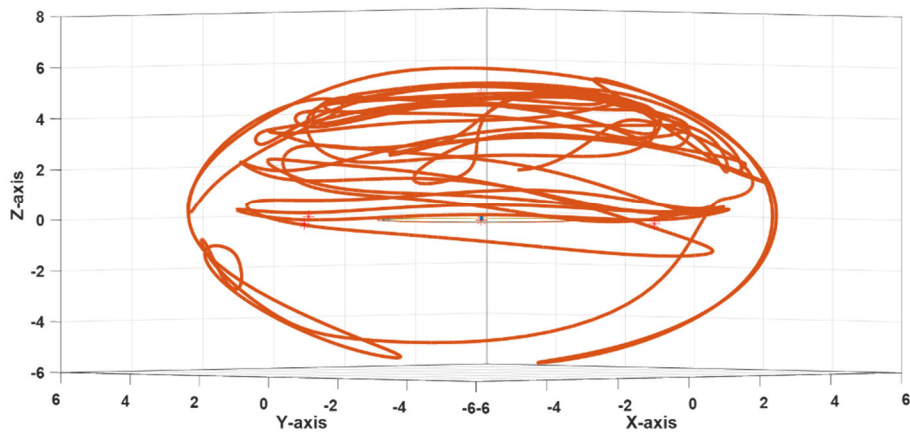
Figure. 5.40 The simulation results for case 1 with disturbances acting on quadrotor 2. (a) Actual 3-D trajectories for all agents. (b) Actual 3-D trajectories for the second quadrotor. (c) its attitude. (d) the tracking errors.

Case 2: With additive tracking error only:

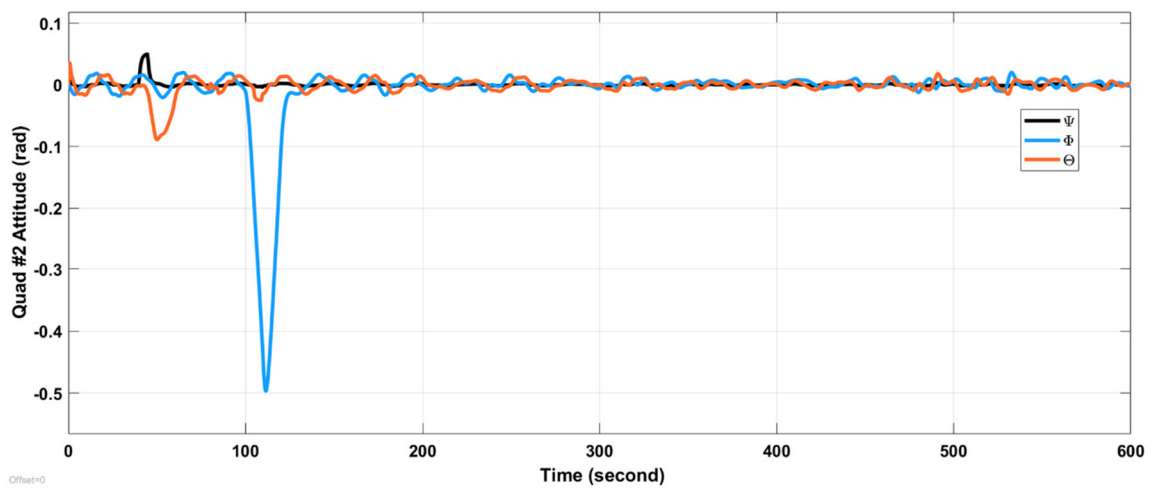
The simulation results of this case are shown in Figure 5.41.



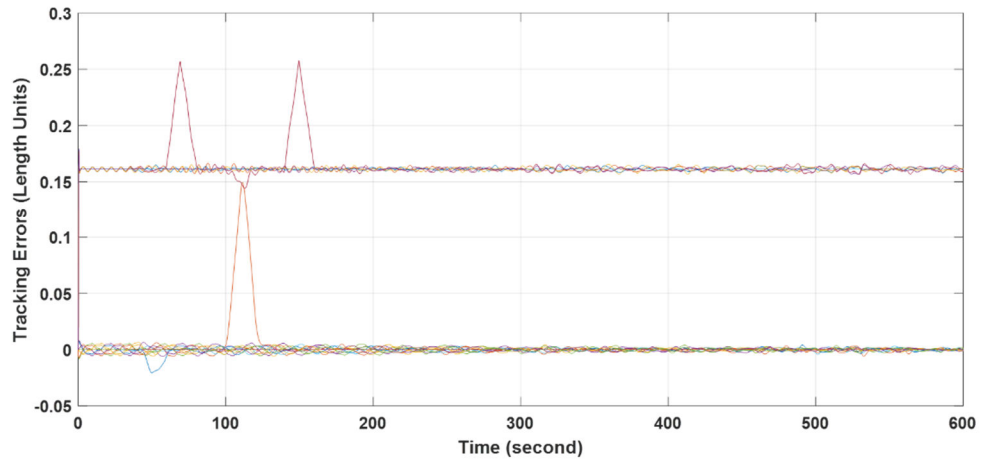
(a)



(b)



(c)

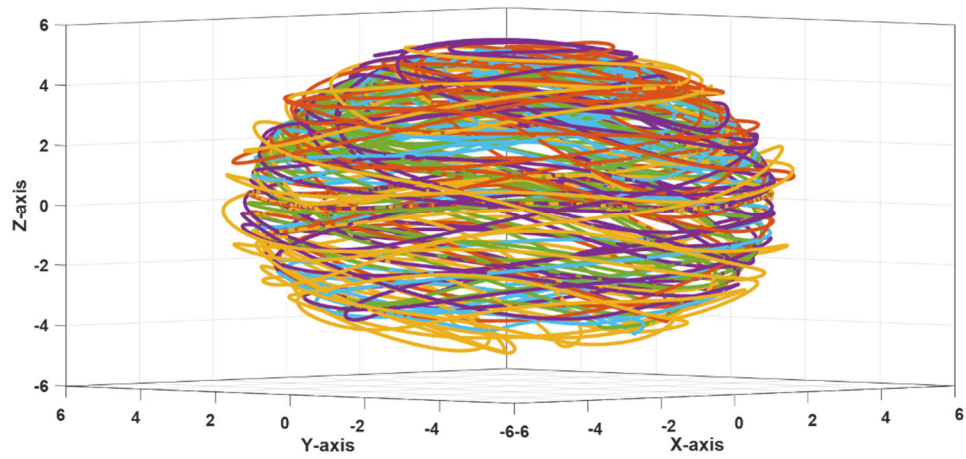


(d)

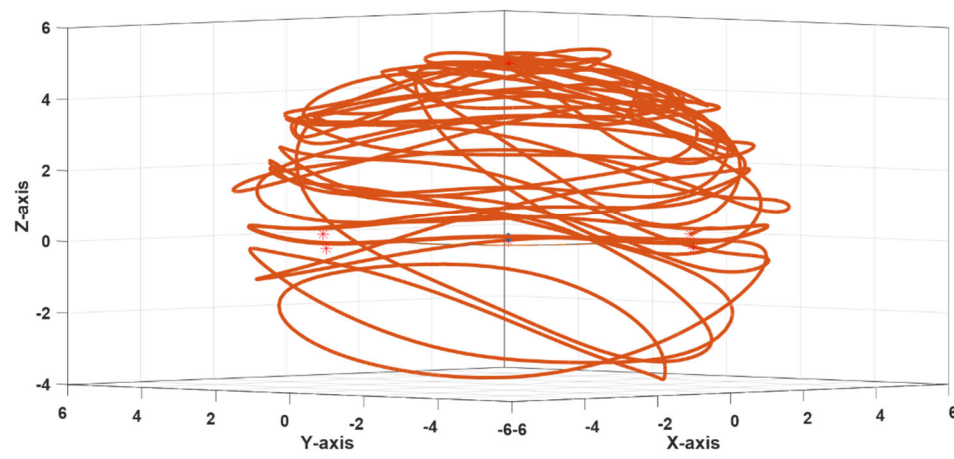
Figure. 5.41 The simulation results for case 2 with disturbances acting on quadrotor 2. (a) Actual 3-D trajectories for all agents. (b) Actual 3-D trajectories for the second quadrotor. (c) its attitude. (d) the tracking errors.

Case 3: With multiplicative tracking error only:

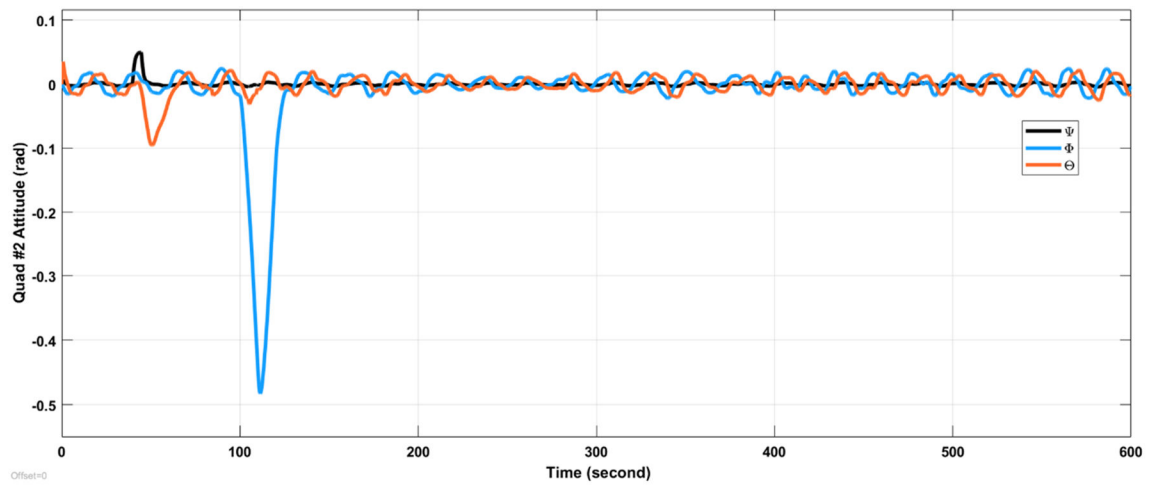
The simulation results of this case are shown in Figure 5.42.



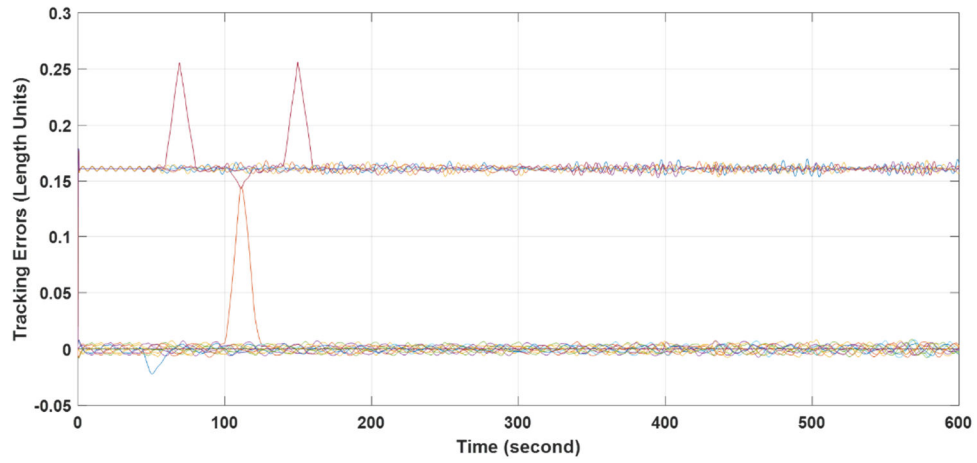
(a)



(b)



(c)

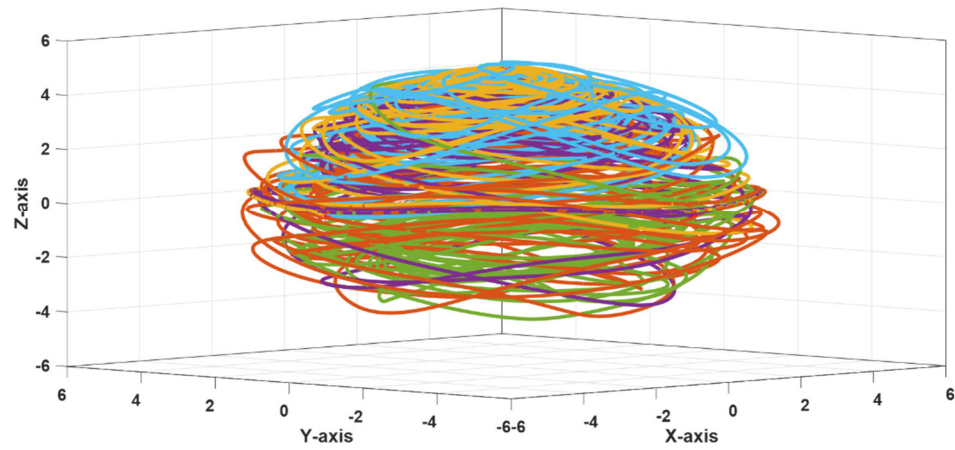


(d)

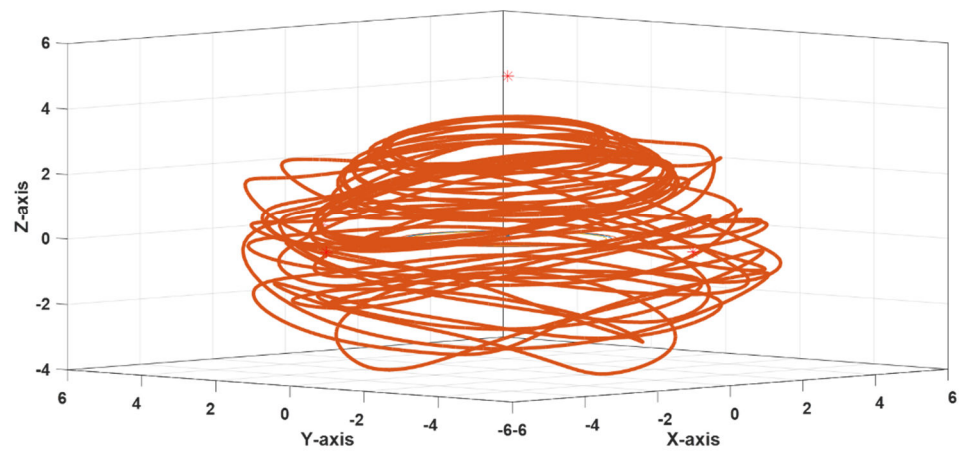
Figure. 5.42 The simulation results for case 3 with disturbances acting on quadrotor 2. (a) Actual 3-D trajectories for all agents. (b) Actual 3-D trajectories for the second quadrotor. (c) its attitude. (d) the tracking errors.

Case 4: With both types of tracking errors involved:

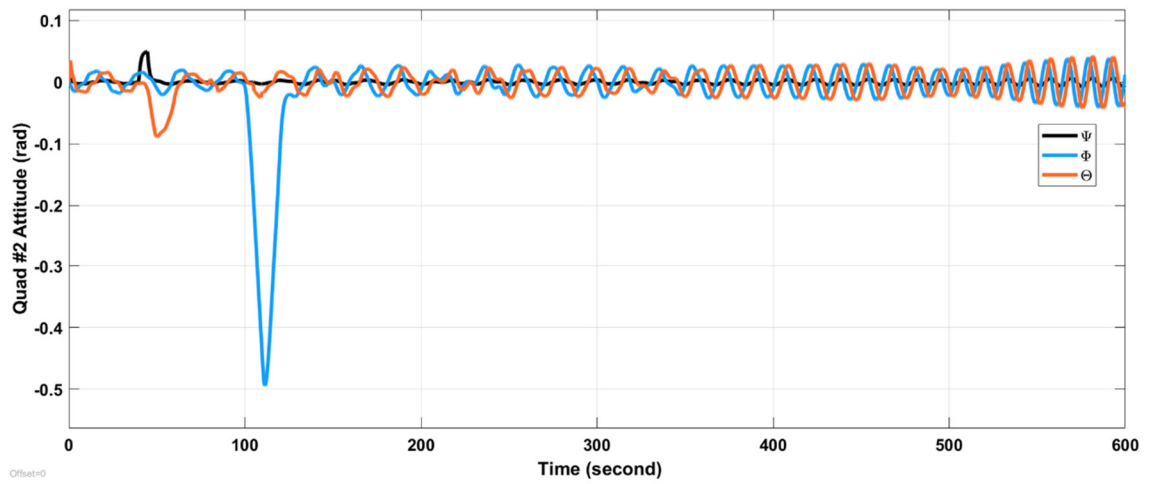
The simulation results of this case are shown in Figure 5.43.



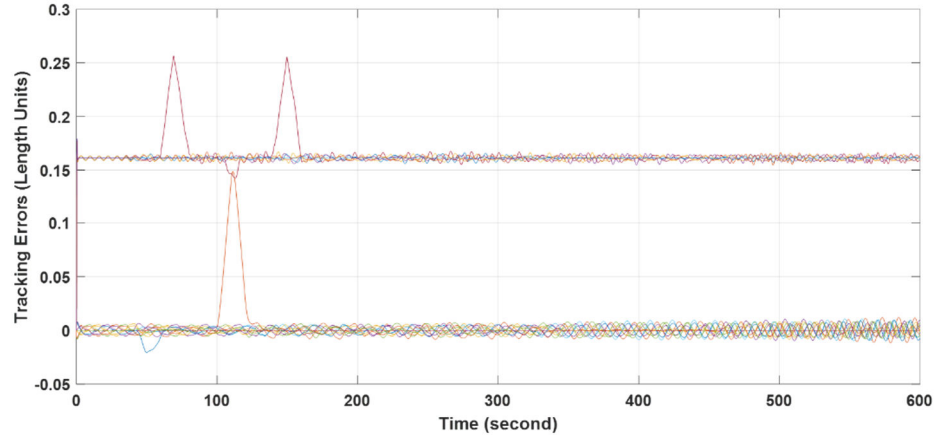
(a)



(b)



(c)



(d)

Figure. 5.43 The simulation results for case 4 with disturbances acting on quadrotor 2. (a) Actual 3-D trajectories for all agents. (b) Actual 3-D trajectories for the second quadrotor. (c) its attitude. (d) the tracking errors.

□

From the previous example, the contribution of the tracking controllers, both the additive and multiplicative, is very important in capturing the real capabilities of the driven agents which will reduce the tracking errors and helps in modifying the idealized trajectories generated so that they become more dynamic-friendly.

Remark 5.28: One main difference between the two types of the tracking controllers is that the additive controller is very sensitive to any fixed offset in the tracking error which may cause the trajectories to drift over time.

Considering **Remark 5.28**, the additive tracking component along the z -axis was ignored because of the dc-offset experienced under the action of the local controller used. This dc-offset can be seen clearly from Figure 5.43 (d), for example.

At this point, we would like to mention that the previous work is inspired by the case study shown in Figure 5.36 that we will consider in more detail in a future work.

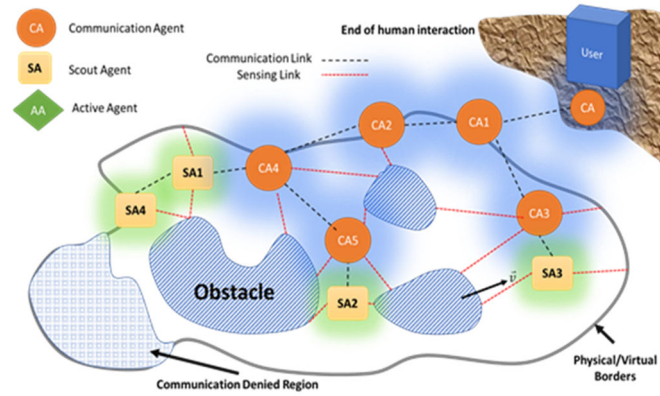


Figure. 5.44. A rescue-team working in a devastated environment.

5.8 Conclusion

In this chapter, a set of interesting behaviors generated using dynamical systems with higher dimensions was presented. The usefulness of the proposed framework in realizing these behaviors were demonstrated under a specialized representation of the involved dynamical systems. Both kinematical and kino-dynamical trajectory-generator systems were designed. Intelligent controllers that depend on both environment models and sensory inputs were realized. Harmonic potential fields were the main tool to model the environment, while the sensory inputs were the pillars of the reactive controllers presented. Sequential and parallel behaviors were discussed were the importance of the latter was demonstrated through an example. Combining both collision-avoidance and connectivity-preserving protocols was highlighted. Finally, a simple, yet efficient, control scheme to guide a group of non-holonomic front-wheel steered robots through an environment filled with obstacles was provided.

In the future, more focus will be needed on utilizing the kino-dynamical trajectory-generator system to deal with more demanding situations- mainly where heterogeneous agents are collaborating to resolve a situation in a devastated region.

CHAPTER 6

The Possible Extensions

In this thesis, several issues related to motion-related behaviors of multi-agent system on graphs have been handled. A framework that can be used to build such behaviors was proposed where motion-invariance and stability analysis can be ensured if the given conditions are observed. Consensus, clustering, formation, deployment are some examples of the covered behaviors. The concept of behavior banks was utilized in which several primitive behaviors can be stored and then be selected by a suitable behavior selection mechanism that is controlled by the agent's embedded artificial intelligence or an external mission planning utility. Issues related to connectivity preservation and collision avoidance were handled under the proposed framework. The ultimate result was a sophisticated distributed coordination and motion planning with both kinematical and kino-dynamical versions were derived in detail. However, more improvements can find its way towards the proposed framework. In this chapter, many possible future extensions which are directly related to the case study which motivated this thesis are stated as follows:

- 1- Studying the non-diffusive couplings under the proposed framework.

This will broaden the possibilities of dealing with almost all systems under absolute state coupling with neighboring agents on the graph. Remember that a symmetrical system can be achieved also using absolute state feedback, as stated in this thesis.

- 2- Considering the same work presented in this thesis under nonlinear protocols.

This step will open the way to considering mainly partial state-feedback or output-feedback. This will relax the assumptions imposed so far by which a full-state-

feedback is required to facilitate the functionality of the proposed behaviors. Suitable state-observers can also be utilized.

3- Solving the state-dependent Riccati equation (SDRE).

This will introduce the optimality at the agent level such that the needed local stabilizing controller can be optimal under a certain situation of operation.

4- Introducing complex weighting functions.

As stated by Complex Analysis Theorem, any function can be written as a sum of two functions, i.e., one real and another that is complex. This will make the Laplacian matrix used in the semi-linear protocols complex. It is expected to be able to introduce more complex behaviors if complex values were used at both states and weighting functions.

5- Increasing the potential of the proposed framework by covering new behaviors.

Driven by the application, new behaviors may emerge other than those already investigated in this thesis. For example, pattern formation and multi-leader multi-follower schemes are interesting behaviors to investigate.

6- Developing a dedicated joint design and simulation utility.

The purpose of this utility is to facilitate building and analyzing the intended behaviors. It is true that there are available robotic simulation software available both commercially and for free; however, building a dedicated utility that matches the capabilities of the proposed framework is highly recommended.

7- Investigating the possibility of building the behaviors within the agents' dynamics, intrinsically.

In this thesis, we have developed sophisticated trajectory-generator co-systems that generate the reference signals. Agents are supposed to follow these signals and as a result they will generally deviate from their preferred state, i.e., least energy at equilibrium, to achieve the desired behaviors. However, by intrinsically crafting the behaviors within agents' dynamics, then the executed behaviors will be their new norms. This was clearly demonstrated in Chapter 3 where the agents' dynamics were approximated as single-integrator dynamics.

8- Introducing the time as an independent variable.

In this thesis, we only focused on using autonomous systems, i.e., independent of time. In real applications; however, time is very critical. So, by introducing the time as an independent variable, the proposed framework can cover more applications.

9- Addressing the issue of prescribed performance in more detail.

In Chapter 3, we only handled the simplest case that can be encountered in real applications. So, imposing more constraints on the behaviors is needed when considering agents with limited resources and timely-applications.

10- Studying the connectivity-preserving under absolute and relative nonlinear protocols.

11- Building decision-making utilities within the system that resolve conflicts among existing behaviors.

12- Addressing the proposed search and rescue mission thoroughly.

13- Study the controllability and observability under state-dependent protocols.

14- Tackling the issue of dealing with emerging behaviors.

In general, the behaviors stored in the behavior bank can adapt or learn the best way to deal with changing contexts. This can be handled by incorporating techniques such as deep learning or the like in the decision-making process.

References

- [1] M. Zhou, J. Zhan, and X. Li, "Cluster Consensus in Networks of Agents With Weighted Cooperative-Competitive Interactions Via Nonlinear Protocols," in *Proceedings of the 34th Chinese Control Conference*, 2015, pp. 7108–7112.
- [2] Z. Lin, M. Fu, and Y. Diao, "Distributed Self Localization for Relative Position Sensing Networks in 2D Space," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3751–3761, 2015.
- [3] J. J. Molitierno, *applications of combinatorial matrix theory to Laplacian matrices of graphs*. CRC Press, 2012.
- [4] Q. Pham and J. Slotine, "Stable concurrent synchronization in dynamic system networks," *Neural Networks*, vol. 20, pp. 62–77, 2007.
- [5] N. Monshizadeh, H. L. Trentelman, and M. K. Camlibel, "Projection-Based Model Reduction of Multi-Agent Systems Using Graph Partitions," *Trans. Control Netw. Syst.*, vol. 1, no. 2, pp. 145–154, 2014.
- [6] D. M. Stipanovic and D. D. Siljak, "stability of polytopic systems via convex M-matrices and parameter-dependent lyapunov functions," *Nonlinear Anal.*, vol. 40, pp. 589–609, 2000.
- [7] X. Zhang, Y. Han, L. Wu, and J. Zou, "M-matrix-based globally asymptotic stability criteria for genetic regulatory networks with time-varying discrete and unbounded distributed delays," *Neurocomputing*, vol. 174, pp. 1060–1069, 2016.
- [8] X. Zhang, L. Wu, and J. Zou, "Globally Asymptotic Stability Analysis for Genetic Regulatory Networks with Mixed Delays : An M-matrix-Based Approach,"

- IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 13, no. 1, pp. 135–147, 2016.
- [9] Frank L. Lewis, H. Zhang, and K. H.-M. A. Das, *Cooperative Control of Multi-Agent Systems*. Springer London, 2014.
 - [10] R. J. Plemmons, “M-Matrix Characterizations .I -Nonsingular M-Matrices,” *Linear Algebra Appl.*, vol. 18, pp. 175–188, 1977.
 - [11] M. Neumann and R. J. Plemmons, “M-matrix characterization II: General M-matrices,” *Linear Multilinear Algebr*, vol. 9, pp. 211–225, 1980.
 - [12] M. Araki, “Application of M-matrices to the Stability Problems of Composite Dynamical Systems,” *J. Math. Anal. Appl.*, vol. 52, pp. 309–321, 1975.
 - [13] A. Berman and R. J. Plemmons, *Nonnegative matrices in the mathematical sciences*. Academic Press, 1979.
 - [14] A. Chapman and M. Mesbahi, “Stability Analysis of Nonlinear Networks via M-matrix Theory : Beyond Linear Consensus,” in *American Control Conference*, 2012, pp. 6626–6631.
 - [15] J. Bierkens and A. Ran, “A singular M-matrix perturbed by a nonnegative rank one matrix has positive principal minors ; is it D-stable ?,” *Linear Algebra Appl.*, vol. 457, no. 270327, pp. 191–208, 2014.
 - [16] S. Basagni, M. Conti, G. Silvia, and I. Stojmenovic, *Mobile AD HOC Networking*. 2004.
 - [17] A. Ahmed and B. Far, “Performance of Mobile Agent based network topology discovery,” in *Canadian Conference on Electrical and Computer Engineering*, 2007, pp. 66–69.
 - [18] F. Garin and Y. Yuan, “Distributed privacy-preserving network size computation :

- A system-identification based method,” in *52nd IEEE Conference on Decision and Control*, 2013, pp. 5438–5443.
- [19] F. Garin and Y. Yuan, “Distributed privacy-preserving network size computation: A system-identification based method,” 2013.
 - [20] J. Zarei, E. Shokri, and H. R. Karimi, “Convergence Analysis of Cubature Kalman Filter,” in *2014 European Control Conference (ECC)*, 2014, pp. 1367–1372.
 - [21] H. Terelius, D. Varagnolo, and K. H. Johansson, “Distributed Size Estimation of Dynamic Anonymous Networks,” 2013.
 - [22] G. Chowdhary, M. Egerstedt, and E. N. Johnson, “Network Discovery : An Estimation Based Approach.”
 - [23] M. Zamani and H. Lin, “Weights ’ Assignment in Multi-Agent Systems Under a Time-Varying Topology,” in *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2009, pp. 746–751.
 - [24] L. O. U. Youcheng and H. Yiguang, “Controllability and Controllable Subspace of Multi-agent Systems with Directed Interconnection,” in *Proceedings of the 30th Chinese Control Conference*, 2011, pp. 6030–6035.
 - [25] S. Martini, M. Egerstedt, and A. Bicchi, “Controllability analysis of multi-agent systems using relaxed equitable partitions,” *Int. J. Syst. Control Commun.*, vol. 2, pp. 100–121, 2010.
 - [26] T. Kailath, *Linear Systems*. Englewood Cliffs, NY: Prentice-Hall, 1980.
 - [27] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, “Controllability of Multi-Agent Systems from a Graph-Theoretic Perspective,” *SIAM J. Control OPTIM*, vol. 48, no. 1, pp. 162–186, 2009.

- [28] G. Yongqiang, J. I. Zhijian, Z. Lin, and W. Long, "Structural Controllability of Higher-Order Multi-Agent Systems under Absolute and Relative Protocols," in *Proceedings of the 32nd Chinese Control Conference*, 2013, pp. 6832–6837.
- [29] Z. Ji, H. Lin, and J. Gao, "Eigenvector Based Design of Uncontrollable Topologies for Networks of Multiple Agents," in *Proceedings of the 32nd Chinese Control Conference*, 2013, pp. 6797–6802.
- [30] J. I. Zhijian, C. Tongwen, and Y. U. Haisheng, "Controllability of Sampled-Data Multi-Agent Systems," in *Proceedings of the 33rd Chinese Control Conference*, 2014, pp. 1534–1539.
- [31] K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, "Multi-agent differential graphical games : Online adaptive learning solution for synchronization with optimality," *Automatica*, vol. 48, no. 8, pp. 1598–1611, 2012.
- [32] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. 1999.
- [33] G. Wen, W. Yu, Y. Zhao, and J. Cao, "Node-to-Node Consensus of Networked Agents with General Linear Node Dynamics," in *Proceedings of the 2013 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, 2013, pp. 24–29.
- [34] H. Hu, W. Yu, Q. Xuan, L. Yu, and G. Xie, "Consensus of multi-agent systems in the cooperation – competition network with inherent nonlinear dynamics : A time-delayed control approach," *Neurocomputing*, vol. 158, pp. 134–143, 2015.
- [35] G. Wen, W. Yu, J. Wang, D. Xu, and J. Cao, "Node-to-Node Consensus of Multi-Agent Systems with Switched Pinning Links," in *Control and Decision*

- Conference (2014 CCDC)*, 2014, pp. 269–273.
- [36] W. E. N. Guanghui, Y. U. Wenwu, and H. U. Guoqiang, “Distributed Node-to-Node Consensus of Multi-Agent Systems with Lur ’ e Node Dynamics,” in *Proceedings of the 32nd Chinese Control Conference*, 2013, pp. 7314–7319.
 - [37] Y. Fan and G. Hu, “Connectivity-Preserving Rendezvous of Multi-Agent Systems with Event-Triggered Controllers,” in *2015 IEEE 54th Annual Conference on Decision and Control (CDC)*, 2015, pp. 234–239.
 - [38] W. Hu, L. Liu, S. Member, and G. Feng, “Consensus of Linear Multi-Agent Systems by Distributed Event-Triggered Strategy,” *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 148–157, 2016.
 - [39] W. Liu and J. Huang, “Cooperative Global Robust Output Regulation for a Class of Nonlinear Multi-Agent Systems with Switching Network,” in *Proceedings of the 33rd Chinese Control Conference*, 2014, no. 1, pp. 1184–1189.
 - [40] S. Wang, P. Zhang, and Y. Fan, “Centralized event-triggered control of multi-agent systems with dynamic triggering mechanisms,” in *27th Chinese Control and Decision Conference (CCDC)*, 2015, pp. 2183–2187.
 - [41] D. Yang, X. Liu, and W. Chen, “Periodic event / self-triggered consensus for general continuous-time linear multi-agent systems under general directed graphs,” *IET Control Theory Appl.*, vol. 9, no. November 2014, pp. 428–440, 2015.
 - [42] P. Pascheka and M. Düring, “Advanced Cooperative Decentralized Decision Making using a Cooperative Reward System,” 2015.
 - [43] Z. Li, G. Wen, Z. Duan, and W. Ren, “Designing Fully Distributed Consensus Protocols for Linear Multi-Agent Systems With Directed Graphs,” *IEEE Trans.*

- Automat. Contr.*, vol. 60, no. 4, pp. 1152–1157, 2015.
- [44] W. Zhu and Z. Jiang, “Event-Based Leader-following Consensus of Multi-Agent,” *IEEE Trans. Automat. Contr.*, vol. 60, no. 5, pp. 1362–1367, 2015.
- [45] Z. Liu, Q. Jin, and Z. Chen, “Distributed containment control for bounded unknown second-order nonlinear multi-agent systems with dynamic leaders,” *Neurocomputing*, vol. 168, pp. 1138–1143, 2015.
- [46] B. Cui, C. Zhao, T. Ma, and C. Feng, “Leader-following consensus of nonlinear multi-agent systems with switching topologies and unreliable communications,” *Neural Comput. Appl.*, 2015.
- [47] K. Chen, J. Wang, Y. Zhang, and Z. Liu, “Consensus of second-order nonlinear multi-agent systems under state-controlled switching topology,” *Nonlinear Dyn.*, vol. 81, no. 4, pp. 1871–1878, 2015.
- [48] R. Olfati-saber and J. S. Shamma, “Consensus Filters for Sensor Networks and Distributed Sensor Fusion,” in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005, pp. 6698–6703.
- [49] E. Mallada and A. Tang, “Distributed Clock Synchronization : Joint Frequency and Phase Consensus,” in *50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 6742–6747.
- [50] W. Ren and R. Beard, “Overview of Consensus Algorithms in Cooperative Control,” in *Distributed Consensus in Multi-vehicle Cooperative Control Theory and Applications*, 2008, pp. 1–22.
- [51] Y. Li, H. Voos, M. Darouach, and C. Hua, “Nonlinear protocols for distributed

- consensus in directed networks of dynamic agents,” *J. Franklin Inst.*, vol. 352, no. 9, pp. 3645–3669, 2015.
- [52] X. Liu, T. Chen, and W. Lu, “Consensus problem in directed networks of multi-agents via nonlinear protocols,” *Phys. Lett. A*, vol. 373, no. 35, pp. 3122–3127, 2009.
 - [53] R. O. Saber and R. M. Murray, “Consensus Protocols for Networks of Dynamic Agents,” in *American Control Conference*, 2003.
 - [54] Y. Karayiannidis, D. V Dimarogonas, and D. Kragic, “Multi-agent average consensus control with prescribed performance guarantees,” in *51st IEEE Conference on Decision and Control*, 2012, pp. 2219–2225.
 - [55] A. Priolo, A. Gasparri, E. Montijano, and C. Sagues, “A distributed algorithm for average consensus on strongly connected weighted digraphs,” *Automatica*, vol. 50, no. 3, pp. 946–951, 2014.
 - [56] H. Mangesius, D. Xue, S. Member, and S. Hirche, “Consensus Driven by the Geometric Mean,” *Trans. Control Netw. Syst.*, vol. 5870, no. i, pp. 1–11, 2016.
 - [57] Z. Kan, A. P. Dani, J. M. Shea, and W. E. Dixon, “Network Connectivity Preserving Formation Stabilization and Obstacle Avoidance via a Decentralized Controller,” *IEEE Trans. Automat. Contr.*, vol. 57, no. 7, pp. 1827–1832, 2012.
 - [58] M. Ji and M. Egerstedt, “Distributed Coordination Control of Multiagent Systems While Preserving Connectedness,” *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, 2007.
 - [59] A. Ajorlou, A. Momeni, and A. G. Aghdam, “Sufficient conditions for the convergence of a class of nonlinear distributed consensus algorithms,” *Automatica*,

- vol. 47, no. 3, pp. 625–629, 2011.
- [60] K. H. Dimarogonas, D. V. Johansson, “Bounded control of network connectivity in multi-agent systems,” *IET Control Theory Appl.*, vol. 4, no. 8, pp. 1330–1338, 2010.
 - [61] D. Boskos and D. V Dimarogonas, “Robustness and Invariance of Connectivity Maintenance Control for Multiagent Systems,” *SIAM J. Control OPTIM*, vol. 55, no. 3, pp. 1887–1914, 2017.
 - [62] M. M. Zavlanos and G. J. Pappas, “Potential Fields for Maintaining Connectivity of Mobile Networks,” *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 1–5, 2007.
 - [63] F. Chen, Z. Chen, Z. Liu, and Z. Yuan, “Control A State-Dependent Dynamic Graph To A Pre-Specified Structure,” *KYBERNETIKA*, vol. 45, no. 5, pp. 801–808, 2009.
 - [64] G. Jing, Y. Zheng, and L. Wang, “Consensus of Multiagent Systems With Distance-Dependent Communication Networks,” *IEEE Trans. NEURAL NETWORKS Learn. Syst.*, pp. 1–15, 2016.
 - [65] M. Risler, “Behavior Control for Single and Multiple Autonomous Agents Based on Hierarchical Finite State Machines,” 2009.
 - [66] J. C. Willems, “The Behavioral Approach to Modeling and Control of Dynamical Systems.” .
 - [67] P. Allgeuer and S. Behnke, “Hierarchical and State-based Architectures for Robot Behavior Planning and Control,” in *In Proceedings of 8th workshop on Humanoid Soccer Robots*, 2013.
 - [68] M. WAHDE, “Behavioral organization,” in *Introduction to autonomous robots*,

- Goteborg, Sweden: Chalmers University of Technology, 2007, pp. 85–108.
- [69] E. Ferrante, “The Design of a Modular Behavior-based Architecture,” *Universite Libre De Bruxelles*, 2009.
 - [70] P. Iñigo-blasco, F. Diaz-del-rio, M. C. Romero-ternero, D. Cagigas-muñiz, and S. Vicente-diaz, “Robotics software frameworks for multi-agent robotic systems development,” *Rob. Auton. Syst.*, vol. 60, no. 6, pp. 803–821, 2012.
 - [71] M. Loetzsch, M. Risler, and J. Matthias, “XABSL - A Pragmatic Approach to Behavior Engineering,” in *International conference on Intelligent Robots and Systems*, 2006, pp. 5124–5129.
 - [72] F. Arrichiello, S. Chiaverini, G. Indiveri, and P. Pedone, “The Null-Space-based Behavioral Control for Mobile Robots with Velocity Actuator Saturations,” *Int. J. Rob. Res.*, pp. 1–21, 2010.
 - [73] L. Moreau, “Stability of Multiagent Systems With Time-Dependent,” *IEEE Trans. Automat. Contr.*, vol. 50, no. 2, pp. 169–182, 2005.
 - [74] S. Lee, H. Kim, H. Myung, and X. Yao, “Cooperative Coevolutionary Algorithm-Based Model Predictive Control Guaranteeing Stability of Multirobot Formation,” *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 37–51, 2015.
 - [75] J. Alonso-mora, M. Rufli, R. Siegwart, and P. Beardsley, “Collision Avoidance for Multiple Agents with Joint Utility Maximization,” in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 2833–2838.
 - [76] C. A. Kitts, K. Stanhouse, and P. Chindaphorn, “Cluster Space Collision Avoidance for Mobile Two-Robot Systems,” in *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1941–1948.

- [77] A. Awad, A. Chapman, E. Schoof, A. Narang-siddarth, and M. M. Fellow, “Time-Scale Separation in Networks: State-Dependent Graphs and Consensus Tracking,” *Trans. Control Netw. Syst.*, vol. 5870, no. c, pp. 1–10, 2018.
- [78] F. Li, Y. Shen, D. Zhang, X. Huang, and Y. Wang, “Leader-following consensus for upper-triangular multi-agent systems via sampled and delayed output feedback,” *Neurocomputing*, vol. 275, pp. 1441–1448, 2018.
- [79] W. Wang, C. Huang, J. Cao, and F. E. Alsaadi, “Event-triggered control for sampled-data cluster formation of multi-agent systems,” *Neurocomputing*, vol. 267, pp. 25–35, 2017.
- [80] K. Liu, Z. Ji, G. Xie, and R. Xu, “Event-based broadcasting containment control for multi-agent systems under directed topology,” *Int. J. Control*, vol. 89, no. 11, pp. 2360–2370, 2016.
- [81] H. Zhang, P. Hu, Z. Liu, and L. Ding, “Consensus analysis for multi-agent systems via periodic event-triggered algorithms with quantized information,” *J. Franklin Inst.*, vol. 354, no. 14, pp. 6364–6380, 2017.
- [82] X. Ge, Q. Han, D. Ding, X. Zhang, and B. Ning, “A survey on recent advances in distributed sampled-data cooperative control of multi-agent systems,” *Neurocomputing*, vol. 275, pp. 1684–1701, 2018.
- [83] A. C. Review, “Multi-Agent Cooperative Control Consensus: A Comparative Review,” *electronics*, vol. 7, no. 22, pp. 1–20, 2018.
- [84] X. Ge, F. Yang, and Q. Han, “Distributed networked control systems: A brief overview,” *Inf. Sci. (Ny)*, vol. 380, pp. 117–131, 2017.
- [85] Y. Yan and Y. Mostofi, “Robotic Router Formation in Realistic Communication

- Environments,” *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 810–827, 2012.
- [86] J. Cortés, “Distributed algorithms for reaching consensus on general functions,” *Automatica*, vol. 44, pp. 726–737, 2008.
- [87] T. Zaslavsky, “Matrices in the Theory of Signed Simple Graphs,” pp. 1–20, 2010.
- [88] R. Olfati-saber and J. S. Shamma, “Consensus Filters for Sensor Networks and Distributed Sensor Fusion,” in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005, pp. 6698–6703.
- [89] B. R. Olfati-saber, J. A. Fax, and R. M. Murray, “Consensus and Cooperation in Networked Multi-Agent Systems,” *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [90] R. Olfati-Saber and R. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Trans. Automat. Contr.*, vol. 49, pp. 1520–1533, 2004.
- [91] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*, 1st ed. Springer-Verlag London, 2008.
- [92] S. Bandyopadhyay and S.-J. Chung, “Distributed Estimation using Bayesian Consensus Filtering,” in *American Control Conference*, 2014, pp. 634–641.
- [93] O. Slučiak and M. Rupp, “Consensus Algorithms With State-Dependent Weights,” *IEEE Trans. SIGNAL Process.*, vol. 64, no. 8, pp. 1972–1985, 2016.
- [94] D. Bauso, L. Giarre, and R. Pesenti, “Non-linear protocols for optimal distributed consensus in networks of dynamic agents,” *Syst. Control Lett.*, vol. 55, pp. 918–928, 2006.
- [95] Y. Kawano and T. Ohtsuka, “Stability criteria with nonlinear eigenvalues for

- diagonalizable nonlinear systems,” *Syst. Control Lett.*, vol. 86, pp. 41–47, 2015.
- [96] P. Lindqvist, “A Nonlinear Eigenvalue Problem.” Department of Mathematics, Norwegian University of Science and Technology, pp. 173–199, 2000.
- [97] M. Hal, “Eigenvalues for a nonlinear time-delay system,” in *Proceedings of the 18th International Conference on System Theory, Control and Computing*, 2014, pp. 291–296.
- [98] D. Robert, “Non Linear Eigenvalue Problems.” Departement de Mathematiques, Universite de Nantes, pp. 109–127, 2004.
- [99] M. Halas and C. H. Moog, “Definition of eigenvalues for a nonlinear system,” in *9th IFAC Symposium on Nonlinear Control Systems*, 2013, pp. 600–605.
- [100] M. J. Crawley, *Statistics: An Introduction using R*. London: John Wiley & Sons, Ltd, 2005.
- [101] C. J. Papachristou, “Aspects of Integrability of Differential Systems and Fields.” 2015.
- [102] A. C. J. Luo and V. Afraimovich, *Hamiltonian Chaos Beyond the KAM Theory*. Springer London, 2010.
- [103] A. C. J. Luo, *Discontinuous Dynamical Systems*. Springer London, 2012.
- [104] J. MIERCZYNSKI, “Strictly Cooperative Systems With A First Integral,” *SIAM J. MATH. ANAL.*, vol. 18, no. 3, pp. 642–646, 1987.
- [105] B. TANG, Y. KUANG, and H. SMITH, “Strictly nonautonomous cooperative system with a first integral,” *SIAM J. Control OPTIM*, vol. 24, no. 5, pp. 1331–1339, 1993.
- [106] P. J. OLVER, “Nonlinear Ordinary Differential Equations,” in *Introduction to*

Partial Differential Equations, 2012, pp. 1081–1142.

- [107] J. Binney and S. Tremaine, *Galactic Dynamics*, 2nd ed. Princeton University Press, 2008.
- [108] Y. Kawano and T. Ohtsuka, “PBH tests for nonlinear systems,” *Automatica*, vol. 80, pp. 135–142, 2017.
- [109] Y. Kawano and T. Ohtsuka, “Nonlinear Eigenvalue Approach to Differential Riccati Equations for Contraction Analysis,” *Trans. Autom. Control*, vol. 62, no. 12, pp. 6497–6504, 2017.
- [110] Q. Hui and W. M. Haddad, “Distributed nonlinear control algorithms for network consensus,” *Automatica*, vol. 44, pp. 2375–2381, 2008.
- [111] J. W. Helton, V. Katsnelson, and I. Klep, “Sign patterns for chemical reaction networks,” *J Math Chem*, vol. 47, pp. 403–429, 2010.
- [112] J. Zhu, Y. Tian, and J. Kuang, “On the general consensus protocol of multi-agent systems with double-integrator dynamics,” *Linear Algebra Appl.*, vol. 431, pp. 701–715, 2009.
- [113] P. Wieland, “From Static to Dynamic Couplings in Consensus and Synchronization among Identical and Non-Identical Systems,” Universit at Stuttgart, 2010.
- [114] J. T. Xing, *Energy Flow Theory of Nonlinear Dynamical Systems with Applications*. Springer London, 2015.
- [115] V. Nikiforov, “Beyond graph energy : norms of graphs and matrices,” *Linear Algebra Appl.*, vol. 506, pp. 82–138, 2016.
- [116] M. Chilali, P. Gahinet, P. Apkarian, and A. Member, “Robust Pole Placement in

- LMI Regions,” *IEEE Trans. Automat. Contr.*, vol. 44, no. 12, pp. 2257–2270, 1999.
- [117] A. A. Masoud, “Nearest Neighbor-Based Rendezvous for Sparsely Connected Mobile Agents,” *J. Dyn. Syst. Meas. Control*, vol. 137, no. December, pp. 1–18, 2015.
- [118] R. Bhatia, *Matrix Analysis*. 1997.
- [119] G. Chen, “Stability of Nonlinear Systems,” in *Wiley Encyclopedia of Electrical and Electronics Engineering (John Wiley & Sons, Inc., New York)*, 2001.
- [120] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2001.
- [121] K. S. Lam, *Topics in Contemporary Mathematical Physics*, 2nd ed. World Scientific, 2015.
- [122] S. Boyd, “Convex optimization of graph laplacian eigenvalues,” in *Proceedings of the international congress of Mathematicians*, 2006, pp. 1311–1319.
- [123] C. W. WU, *synchronization in complex networks of nonlinear dynamical systems*. World Scientific, 2007.
- [124] M. Lu, L. Zhang, and F. Tian, “Lower bounds of the Laplacian spectrum of graphs based on diameter,” *Linear Algebra Appl.*, vol. 420, pp. 400–406, 2007.
- [125] B. Mohar, “Eigenvalues , Diameter , and Mean Distance in Graphs,” *Graphs Comb.*, vol. 7, pp. 53–64, 1991.
- [126] K. Gustafson, “The Curl of Graphs and Networks,” *Math. Model.*, vol. 6, pp. 145–155, 1985.
- [127] D. Petrascheck and R. Folk, “Helmholtz decomposition theorem and Blumenthal’s extension by regularization,” *Condens. Matter Phys.*, vol. 20, no. 1, pp. 1–11,

2017.

- [128] P. Goldstein, “Gradient Flow of a Harmonic Function In R^3 ,” *J. Differ. Equ.*, vol. 247, no. 9, pp. 2517–2557, 2000.
- [129] A. Marino, L. Parker, G. Antonelli, and F. Caccavale, “Behavioral Control for Multi-Robot Perimeter Patrol : A Finite State Automata approach,” in *IEEE International Conference on Robotics and Automation*, 2009, pp. 831–836.
- [130] D. R. Hatch, F. Jenko, A. B. Navarro, V. Bratanov, P. W. Terry, and M. J. Pueschel, “Linear signatures in nonlinear gyrokinetics : interpreting turbulence with pseudospectra,” *New J. Phys.*, 2016.
- [131] M. Deghat, B. D. O. A. Anderson, and Z. Lin, “Combined Flocking and Distance-Based Shape Control of Multi-Agent Formations,” vol. 61, no. 7, pp. 1824–1837, 2016.
- [132] M. Dorothy and S. Chung, “Switched systems with multiple invariant sets,” *Syst. Control Lett.*, vol. 96, pp. 103–109, 2016.
- [133] R. Guo and Y. Wang, “Region Stability Analysis for Switched Nonlinear Systems with Multiple Equilibria,” vol. 15, no. X, pp. 1–8, 2017.
- [134] Z. Han, Z. Lin, Z. Chen, and M. Fu, “Formation Maneuvering with Collision Avoidance and Connectivity Maintenance,” 2015.
- [135] A. Gonzalez-Ruiz, A. Ghaffarkhah, and Y. Mostofi, “A Comprehensive Overview and Characterization of Wireless Channels for Networked Robotic and Control Systems,” *J. Robot.*, vol. 5, 2011.
- [136] G. Indiveri, “An Outlier Robust Filter for Maritime Robotics Applications,” *PALADYN J. Behav. Robot. Res.*, vol. 4, no. 4, pp. 196–203, 2013.

- [137] D. Liberzon, *Switching in Systems and Control*. 1973.
- [138] M. A. Rahimian and A. G. Aghdam, “Structural Controllability of Multi-Agent Systems subject to Simultaneous Failure of Links and Agents,” in *50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 1443–1448.
- [139] S. Jafari, A. Ajorlou, A. G. Aghdam, and S. Tafazoli, “On the Structural Controllability of Multi-Agent Systems Subject to Failure : A Graph-Theoretic Approach,” in *49th IEEE Conference on Decision and Control*, 2010, pp. 4565–4570.
- [140] M. Zamani and H. Lin, “Structural Controllability of Multi-agent Systems,” pp. 5743–5748, 2009.
- [141] Y. Kim and M. Mesbahi, “On Maximizing the Second Smallest Eigenvalue of a State-dependent Graph Laplacian,” in *2005 American Control Conference*, 2005, pp. 99–103.
- [142] D. Lane, “Symbolic Computing Tools for Nonsmooth Dynamics and Control,” in *Variable Structure Systems, Sliding Mode and Nonlinear Control*, no. x, K. D. Young and O. Ozguner, Eds. Springer-Verlag, 1999, pp. 237–262.
- [143] A. Sabanovic, “Variable Structure Systems With Sliding Modes in Motion Control — A Survey,” *IEEE Trans. Ind. INFORMATICS*, vol. 7, no. 2, pp. 212–223, 2011.
- [144] Y. Wu, S. E. Li, Y. Zheng, and J. K. Hedrick, “Distributed Sliding Mode Control for Multi-vehicle Systems with Positive Definite Topologies,” in *Decision and Control*, 2016, pp. 5213–5219.
- [145] B. Erfianto, R. T. Bambang, H. Hindersah, and I. Muchtadi-alamasyah, “Design of

- Connectivity Preserving Flocking Using Control Lyapunov Function,” *J. Robot.*, vol. 2016, pp. 1–12, 2016.
- [146] M. Hongwen, L. Derong, and W. Ding, “distributed control for nonlinear time-delayed multi-agent systems with connectivity preservation using neural networks,” in *22nd International Conference, ICONIP 2015*, 2015, pp. 34–42.
- [147] R. Malhotra, M. Holman, and T. Ito, “Chaos and stability of the solar system,” in *third annual Japanese–American Frontiers of Science symposium*, 2000, pp. 12342–12343.
- [148] H. K. Khalil, P. Hall, and U. S. River, “Nonlinear Systems.”
- [149] L. Wang, Z. Lin, and M. Fu, “Affine Formation of Multi-Agent Systems over Directed Graphs,” in *53rd IEEE Conference on Decision and Control*, 2014, pp. 3017–3022.
- [150] J. Hu and W. Xing, “Emergent collective behaviors on coopetition networks,” *Phys. Lett. A*, vol. 378, no. 26–27, pp. 1787–1796, 2014.
- [151] H. A. L. L. Smith, “Is my system of odes cooperative?,” 2012.
- [152] H. L. Smith, *Dynamical Systems, an introduction to the theory of competitive and cooperative systems*. Rhode Island: American Mathematical Society, 1995.
- [153] R. W. Shields and J. B. Pearson, “Structural Controllability of Multiinput Linear Systems,” *IEEE Trans. Automat. Contr.*, vol. AC-21, no. 2, pp. 203–212, 1976.
- [154] S. Sundaram and C. N. Hadjicostis, “Structural Controllability and Observability of Linear Systems Over Finite Fields With Applications to Multi-Agent Systems,” *IEEE Trans. Automat. Contr.*, vol. 58, no. 1, pp. 60–73, 2013.
- [155] P. G. Ghomsi, F. M. M. Kakmeni, T. C. Kofane, and C. Tchawoua,

- “Synchronization dynamics of chemically coupled cells with activator – inhibitor pathways,” *Phys. Lett. A*, vol. 378, no. 38–39, pp. 2813–2823, 2014.
- [156] S. Pilari, “Activator-Inhibitor Systems Behaviour of Activator-Inhibitor Systems.” pp. 1–10, 2007.
- [157] X. Yu, X. Sun, Y. Lin, and W. Dong, “Output feedback adaptive DSC for nonlinear systems with guaranteed L-infinity tracking performance,” *J Control Theory Appl*, vol. 10, no. November 2010, pp. 124–131, 2012.
- [158] P. Krishnamurthy, F. Khorrami, and Z. P. Jiang, “Global Output Feedback Tracking for Nonlinear Systems in Generalized Output-Feedback Canonical Form,” *IEEE Trans. Automat. Contr.*, vol. 47, no. 5, pp. 814–819, 2002.
- [159] Y. Wan, G. Wen, J. Cao, and W. Yu, “Distributed node-to-node consensus of multi-agent systems with stochastic sampling,” *Int. J. Robust Nonlinear Control*, 2015.
- [160] G. Wen, W. Yu, J. Wang, D. Xu, and J. Cao, “Neurocomputing Distributed node-to-node consensus of multi-agent systems with time-varying pinning links,” *Neurocomputing*, vol. 149, pp. 1387–1395, 2015.
- [161] D. Eager, A. Pendrill, and N. Reistad, “Beyond velocity and acceleration : jerk , snap and higher derivatives,” *Eur. J. Phys.*, vol. 37, pp. 1–11, 2016.
- [162] A. A. Masoud, “Kinodynamic Motion Planning,” *IEEE Robotics & Automation Magazine*, no. March, pp. 85–99, 2010.
- [163] A. A. Masoud, “A Harmonic Potential Field Approach for Planning Motion of a UAV in a Cluttered Environment with a Drift Field,” in *50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 7665–7671.

- [164] K. Motonaka, K. Watanabe, and S. Maeyama, “3-Dimensional Kinodynamic Motion Planning for an X4-Flyer Using 2-Dimensional Harmonic Potential Fields,” in *14th International Conference on Control, Automation and Systems*, 2014, pp. 1181–1184.
- [165] J. E. Slotine and W. Lohmiller, “Modularity , evolution , and the binding problem : a view from stability theory,” *Neural Networks*, vol. 14, pp. 137–145, 2001.
- [166] W. Lohmiller and J. E. Slotine, “On Contraction Analysis for Non-linear Systems *,” *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.
- [167] A. A. Masoud, “A harmonic potential field approach for joint planning and control of a rigid , separable nonholonomic , mobile robot,” *Rob. Auton. Syst.*, vol. 61, no. 6, pp. 593–615, 2013.
- [168] J. Kim and P. K. Khosla, “Real-Time Obstacle Avoidance Using Harmonic Potential Functions,” *IEEE Trmrrctioni Robot. md Autom.*, pp. 1–28, 1992.
- [169] A. R. Meenakshi and C. Rajian, “On a product of positive semidefinite matrices,” vol. 295, pp. 3–6, 1999.
- [170] G. Zhai, “A Generalization Of The Graph Laplacian With Application To A Distributed Consensus Algorithm,” *Int. J. Appl .Math. Comput. Sci.*, vol. 25, no. 2, pp. 353–360, 2015.
- [171] H. Zhang, F. L. Lewis, and Z. Qu, “Lyapunov , Adaptive , and Optimal Design Techniques for Cooperative Systems on Directed Communication Graphs,” vol. 59, no. 7, pp. 3026–3041, 2012.
- [172] M. H. Korayem and S. R. Nekoo, “State-dependent differential Riccati equation to track control of time- varying systems with state and control nonlinearities,” *ISA*

Trans., vol. 57, pp. 117–135, 2015.

- [173] F. Liccardo, S. Strano, and M. Terzo, “Optimal Control Using State-dependent Riccati Equation (SDRE) for a Hydraulic Actuator,” vol. III, pp. 9–13, 2013.
- [174] L. E. Munoz, O. Santos, P. Castillo, and I. Fantoni, “Energy-based nonlinear control for a quadrotor rotorcraft,” in *2013 American Control Conference*, 2013, vol. 1, pp. 1177–1182.

|

Vitae

Name :Yazan Mohammad Suliman Al-Rawashdeh |

Nationality :Jordanian |

Date of Birth :5/21/2015|

Email :yazanmrawashdeh@gmail.com|

Address :Amman-Jordan, 11141|

Academic Background

February 2006: University of Jordan, Amman, Jordan
B.Sc. in Mechatronics Engineering

B.Sc. in Mechatronics Engineering from the University of Jordan with “Very Good” evaluation, 3.34 out of 4.00, and get the 7th place out of 64 in the year 2006.

June 2014: King Fahd University of Petroleum and minerals, Dhahran, Saudi Arabia

M.Sc. in Control and Instrumentation, Systems Engineering

M.Sc. in Control and Instrumentation from KFUPM with GPA 3.875 out of 4.00

March 2018: King Fahd University of Petroleum and minerals, Dhahran, Saudi Arabia

Ph.D. in Systems and Control Engineering

From KFUPM with GPA 3.844 out of 4.00

Professional Experience

September 2006 – April 2008: Engineer

University of Jordan

Amman, Jordan

Mechatronics Labs Supervisor, Experiments Designer.

Engineer

April 2008 – August 2009:

Institute of Applied Technology IAT

Al Ain, UAE

Engineering Labs Teacher

The work involved supporting the work of Applied Technology teachers and their students. Also, providing technical support, ensuring that equipment is functioning properly and ready to use, and that the right materials are available for lessons. Sometimes working closely with students demonstrating experiments; helping teachers with a class; or supporting individual students on projects.

August 2009-August 2010: Teaching Assistant

University of Jordan

Amman, Jordan

Mainly teaching Automation Lab

October 2010-February
2011:

Engineer

Integrated Control and Electrical Systems (ICES)

Part-time employee.

Amman, Jordan

February 2011-September 2012: CNC Machine Maintenance

Personal Work.

Publications

(Journals)

(J1): Al-Rawashdeh, Yazan Mohammad, Moustafa Elshafei, and Mohammad Fahad Al-Malki. "In-Flight Estimation of Center of Gravity Position Using All-Accelerometers." *Sensors* 14.9 (2014): 17567-17585.

(J2): Al-Rawashdeh, Yazan M., and Elshafei, M. "Filtering Techniques for Estimating the Angular Motion Using All-accelerometers." *Applied Mechanics & Materials* 829 (2016).

(Conferences)

(C1): Al-Rawashdeh, Yazan M., Elshafei, M. and El-Ferik, S. "Passive attitude estimation using gyroscopes and all-accelerometer IMU," 2016 7th International Conference on Mechanical and Aerospace Engineering (ICMAE), London, 2016, pp. 368-376.

(C2): Al-Rawashdeh, Yazan M., Muhammad F. Mysorewala, and Moustafa Elshafei. "In-flight alignment of distributed all-accelerometer network." *Sensors Applications Symposium (SAS)*, 2017 IEEE. IEEE, 2017.

(Patents)

(P1): Method and Apparatus for Estimation of Center of Gravity Using Accelerometers
US 9568320 B2.

(P2): Improved Navigation System using all-accelerometer, "Patent Pending"
US15/940223.