Advancements in Adversarially-Resilient Consensus and Safety-Critical Control for Multi-Agent Networks

by

James Bryan Usevitch

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2021

Doctoral Committee:

Associate Professor Dimitra Panagou, Chair
Associate Professor Anouck R. Girard
Assistant Professor Jean-Baptiste Jeannin
Associate Professor Vijay G. Subramanian

James Bryan Usevitch

usevitch@umich.edu

ORCID iD: 0000-0002-1230-7304

I dedicate this to my wife Catherine, whose support and encouragement made this dissertation possible.

Acknowledgments

Everyone knows that Neil Armstrong was the first person on the moon. But how many people helped him get there? Of course there was the rest of the crew: Buzz Aldrin and the oft-forgotten Michael Collins. Then, just like in the movies, there were the dozens of worried-looking mission-control staff on the ground, and notables like Wernher von Braun—intellectual forces who drove the entire program....Armstrong's success required contributions from an entire metropolis worth of people, not including the millions of taxpayers who paid the bills, and the president who challenged a nation to believe. Neil Armstrong is a household name only because his contribution was the most visible. However, the most visible contribution isn't necessarily the most significant.

—Scott Berkun, *The Myths of Innovation* [1]

My path to a PhD would not have been possible without the contributions, influence, and support of many people who have influenced my life. I am grateful for the help and assistance I have received on my path to earning this degree.

I was undecided on whether or not I wanted to pursue a PhD until my senior undergraduate year in college. Four individuals inspired me to finally pursue this path. The first is my father Bryan Usevitch who earned his PhD in Electrical Engineering from the University of Illinois and was a faculty member at the University of Texas at El Paso. I will always be grateful for his attempts to pique my interest as a child in programming with Python, soldering electronics, and working with microcontrollers. The second is my grandfather George Wilson McConkie IV who as a faculty member at Cornell University and the University of Illinois performed groundbreaking experiments in how the eyes and mind process information obtained through reading. His ingenious computer eye-tracking research performed in the 1970's continues to astound me to this day. The third and fourth people are my brother David Usevitch and my cousin Nathan Usevitch, who both entered or were accepted into PhD programs at roughly the same time as myself. I would be less than honest if I didn't admit to feeling at least a little bit of friendly competition to not be outdone by them.

I am deeply indebted to my advisor, Professor Dimitra Panagou, who was willing to take a chance on me and offer me a PhD position that I did not fully deserve. I have had a fantastic experience learning from and working with her, and have appreciated her style of mentoring and teaching as I've earned my degree. I am also grateful for the positive experiences I've had with all past and present members of the DASC lab. Much of what I have learned has been through collaborations and discussions with members of the lab, and I've enjoyed the opportunities to work with each of them. I would like to thank two members in particular for their influence on my PhD: Will Bentz and Kunal Garg. Will was incredibly helpful as I navigated the early stages of my PhD, and offered valuable advice as I prepared for graduation. Kunal has been a huge influence on my PhD experience and technical ability ever since we started the program together. I have appreciated his tutoring, willingness to collaborate and discuss ideas, and his ability to cowrite papers in an unbelievably short amount of time.

I owe much of my success in my work to my mother and father, Heather and Bryan Usevitch. Without their influence I would not be the man I am today. They are an incredible duo who sacrificed part of their present to build a bright future for their children and beyond. All that I know about hard work, honesty, sacrifice, and faith I owe to them. I'd also like to acknowledge the many people from local congregations of The Church of Jesus Christ of Latter-Day Saints who have extended love, service, and friendship to my family during the past 5 years.

Most of all, I would like to thank my wonderful and incredible wife Catherine for her love, optimism, and cheerful encouragement throughout this PhD. Words will never be able to describe how much happiness she has brought me during our experience here at the University of Michigan.

To close, I would like to acknowledge the support of the Automotive Research Center (ARC) in accordance with Cooperative Agreement W56HZV-14-2-0001 U.S. Army TARDEC in Warren, MI; Cooperative Agreement W56HZV-19-2-0001 U.S. Army CCDC Ground Vehicle Systems Center (GVSC) Warren, MI; and the Award No W911NF-17-1-0526. Part of this dissertation has been funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF Award No. 1738714 along with significant contributions from C-UAS industrymembers.

Table of Contents

Dedication . . . . . **ii**

Acknowledgments . . . . . **iii**

List of Figures . . . . . **viii**

Abstract . . . . . **xiv**

Chapter

1 Introduction **1**

1.1 Motivation 1
1.2 Literature Review 2
1.2.1 Origins of the Resilient Agreement Problem 2
1.2.2 MSR Algorithms for Resilient Consensus 5
1.2.3 Resilient Safety and Control of Dynamical Systems 6
1.3 Contributions 9
1.4 Outline 13
1.5 Notation 13

2 Resilient Consensus Algorithms **15**

2.1 Introduction 15
2.2 Preliminaries on Resilient Consensus and MSR Algorithms 18
2.3 Resilient Leader-Follower Consensus to Arbitrary Reference Values in Time-Varying Graphs 23
2.3.1 Problem Formulation 24
2.3.2 Resilient Leader-Follower Consensus in Time-Varying Graphs 28
2.3.3 Adversarial Implications 34
2.3.4 Simulations 35
2.4 Finite-Time Leader-Follower Consensus: Formation Control 38
2.4.1 Notation and Problem Definition 39
2.4.2 Continuous-time System 42
2.4.3 Discrete-time System 49
2.4.4 Simulations 53
2.5 Resilient Finite-Time Consensus: A Discontinuous Systems Perspective 57

2.5.1 Problem Formulation . . . 58
2.5.2 Justification for Discontinuous Systems Approach . . . 62
2.5.3 Review of Discontinuous Systems Theory . . . 62
2.5.4 Main Results . . . 66
2.5.5 Discussion . . . 72
2.5.6 Simulations . . . 73
2.6 Discussion . . . 74
2.7 Conclusion . . . 76

3 Determining r- and (r, s)-Robustness for Design and Analysis of Resilient Networks . 77

3.1 Introduction . . . 77
3.2 Problem Formulation . . . 79
3.3 Robustness of k-Circulant Digraphs . . . 83
3.3.1 Strong r-Robustness of k-Circulant Graphs . . . 89
3.3.2 Implementation of W-MSR Algorithm on k-Circulant Digraphs . . . 90
3.4 Determining r- and (r, s)-Robustness of Digraphs using Mixed Integer Programming . . . 95
3.4.1 Determining r-Robustness using Mixed Integer Linear Programming . . . 95
3.4.2 Determining (r, s)-Robustness using Mixed Integer Linear Programming 102
3.4.3 Approximate Bounds on $\mathbf{r}_{\max}(\mathcal{D})$ . . . 109
3.4.4 Discussion . . . 115
3.4.5 Comparison of MILP Robustness Determination with Prior Methods . . . 116
3.5 Conclusion . . . 121
3.6 Appendix: Description of Algorithm Implementations . . . 121

4 Resilient Broadcast . . . 126

4.1 Introduction . . . 126
4.2 Preliminaries on Resilient Broadcasting . . . 127
4.3 Notation and Problem Formulation . . . 128
4.3.1 Problem Formulation . . . 129
4.4 Sensitivity Analysis . . . 134
4.4.1 Sensitivity to Clock Synchronization Errors . . . 134
4.4.2 Sensitivity to Differences in Parameters . . . 138
4.4.3 Combined Clock and Parameter Perturbation Errors . . . 141
4.5 Resilient Parameter Propagation . . . 142
4.5.1 Synchronous propagation without parameter perturbations . . . 142
4.5.2 Propagation with Time-Varying Graphs . . . 146
4.5.3 Incorporating Parameter Perturbations . . . 148
4.6 Simulations . . . 153
4.6.1 Incorporating Formational Offsets . . . 155
4.6.2 Simulation 1 . . . 157
4.6.3 Simulation 2 . . . 158
4.6.4 Hardware Experiments . . . 163
4.7 Conclusion . . . 168
4.8 Appendix . . . 169

4.8.1 Bernstein Polynomials and Bezier Curves . . . 169

5 Adversarial Resilience for Sampled-Data Systems under High-Relative-Degree Safety Constraints . . . 171

5.1 Introduction . . . 171
5.2 Overview of Control Barrier Function Methods . . . 172
5.3 Adversarial Resilience in Sampled-Data Systems Under Safety Constraints . . . 175
5.3.1 Notation and Problem Formulation . . . 176
5.3.2 Problem Formulation . . . 176
5.4 Safe Set Functions with Relative Degree 1 . . . 179
5.4.1 Preliminaries . . . 179
5.4.2 Synchronous Sampling Times . . . 182
5.4.3 Asynchronous Sampling Times . . . 186
5.4.4 Maximum Safety-Preserving Control Action . . . 188
5.5 Safe Set Functions with High Relative Degree . . . 190
5.5.1 Discussion . . . 196
5.6 Simulations . . . 197
5.6.1 Unicycle Agents in $\mathbb{R}^2$ . . . 197
5.6.2 Double Integrators in $\mathbb{R}^3$ . . . 199
5.7 Conclusion . . . 208

6 Conclusions and Future Work . . . 209

6.1 Conclusions . . . 209
6.2 Future Work . . . 211
6.2.1 Final Discussion . . . 211

Bibliography . . . 213

List of Figures

Figure

2.1 A pictorial representation of how the W-MSR algorithm operates (Algorithm 2.1). . . 20
2.2 A network of agents running the normal W-MSR algorithm with $n = 20,\ k = 15$. The dotted red lines represent adversarial agents, while the solid lines represent normally behaving agents. 22
2.3 A pictorial representation of the Sliding Window Mean-Subsequence-Reduced (SW-MSR) algorithm (Algorithm 2.2). The chief difference between the SW-MSR and W-MSR algorithms are that the SW-MSR considers the most recently received information over a sliding time window to mitigate the effects of the network graph being time-varying. 27
2.4 Time-varying graphs used in the last two simulations. In each graph $\mathcal{G}_j$, $\forall i \in \mathcal{V}$ each agent i sends its state information to the agents depicted. The terms $i + p$ for $p \in \{1, \ldots, 7\}$ are shorthand for $(i + p) \mod n$, where n is the total number of agents. 37
2.5 Leader-follower simulation using the SW-MSR algorithm with a constant reference value in the presence of 2 malicious agents. 37
2.6 Leader-follower simulation using the SW-MSR algorithm with a time-varying reference value in the presence of 3 malicious agents. Note that the normal agents track the reference signal even when the behavior of the malicious agents may be unbounded. . 38
2.7 Visual depiction of the vectors $\boldsymbol{p}_i$, $\boldsymbol{\xi}_i$, and $\boldsymbol{\tau}_i$ used in this section. The formation is achieved when agents' $\boldsymbol{\tau}$ vectors come to consensus. 40
2.8 An example of a Resilient Directed Acyclic Graph (RDAG) with parameter $r = 3$. . . 41
2.9 Diagram of the filtering and state update control law used for the continuous time system in this section. Note that as per Algorithm 2.3, the filtered set $\mathcal{K}_i(t)$ is updated only at time instances $t = m\epsilon_d$ where $\epsilon_d > 0$, $m \in \mathbb{Z}_{\geq 0}$. The reasons for this behavior are discussed below. 43
2.10 Norm $\|\boldsymbol{\tau}_i(t) - \boldsymbol{\tau}_L\|$ of a subset of the normal agents in the continuous time case. For sake of clarity, only a few normal nodes from each set $\mathcal{S}_p$ are shown. 54
2.11 Path of the agents in the continuous time case. All normal and adversarial agents start from the centre of the circle marked by red dots. The leaders are denoted by the star points $\mathrm{p_L}$ and the non-adversarial agents are denoted by $\mathrm{p_N}$. 55
2.12 Norm $\|\boldsymbol{u}_i(t)\|$ of a subset of the normal agents in the continuous-time case, demonstrating that their input magnitudes never exceed the bound $u_M = 1$. The rest of the network is not shown for sake of clarity. 55

2.13 *Discrete Time*: Norm of formational position differences $\|\boldsymbol{\tau}_i[k] - \boldsymbol{\tau}_L\|$ of a subset of the normal agents in the discrete time case. For sake of clarity, only a few normal nodes from each set $\mathcal{S}_p$ are shown. 56
2.14 *Discrete Time:* Path of the agents in the discrete time case. The leaders are denoted by the star points $\mathrm{p_L}$ and the non-adversarial agents are denoted by $\mathrm{p_N}$. 56
2.15 *Discrete Time:* Norm $\|\boldsymbol{u}_i[k]\|$ of a subset of the normal agents in the discrete time case. Again, the magnitude of each agents' control input never exceeds the bound $u_M = 1$ and goes to zero as the agents converge to formation. 57
2.16 Simulation of a network of 15 agents appling the FTRC-P. The dotted red lines represent the adversarial agents. 74

3.1 Depiction of all 12 possible (S_1, S_2) elements in $\mathcal{T}$ for a complete graph $\mathcal{D}$ of 3 agents. Each graph represents a different possible way of dividing $\mathcal{D}$ into sets S_1 and S_2. In each individual graph, yellow agents are in S_1, blue agents are in S_2, and white agents are in neither S_1 nor S_2. 81
3.2 An example of the elements of Θ for a digraph $\mathcal{D}_1$. Since $|\mathcal{V}| = 4$, the possible values of r and s for which the digraph is (r, s)-robust fall within the range $0 \leq r \leq 2$, $1 \leq s \leq 4$. One possible pair of subsets S_1 and S_2 is depicted, which satisfies $|\mathcal{X}^2_{S_1}| \neq |S_1|$, $|\mathcal{X}^2_{S_2}| \neq |S_2|$, $|\mathcal{X}^2_{S_1}| = 0$ and $|\mathcal{X}^2_{S_2}| = 1$. By Definition 3.4, $\mathcal{D}_1$ therefore cannot be $(2, 2)$-robust, $(2, 3)$-robust, or $(2, 4)$-robust. 83
3.3 The general structure of a circulant matrix. By defining the first row, the rest of the matrix is determined. Circulant digraphs have circulant adjacency matrices. 84
3.4 A 3-circulant digraph on 7 nodes, denoted $C_7\{1, 2, 3\}$. Nodes are arranged in a circle for visual clarity; in general the name "circulant" has nothing to do with the physical arrangement of the nodes or agents. 84
3.5 Example of a directed graph whose underlying graph is p-connected, but which is not $\lfloor \frac{p}{2} \rfloor$-robust. The graph shown has an underlying graph with vertex connectivity equal to 4. If the nodes of the graph are divided into the two nonempty, disjoint sets denoted by the green and blue colors, each node clearly only has one in-neighbor outside its set. This implies that the graph can be no more than 1-robust. Note that the two arrangements are the exact same graph; the second configuration is rearranged for clarity. 85
3.6 Counterexample showing that there exist digraphs with an arbitrarily large minimum vertex disconnecting set which are at most 1-robust. The class of digraphs in this figure are composed of two cliques with p directed edges going from clique 1 to clique 2 as shown, and p more directed edges going from clique 2 to clique 1. The size of a minimum vertex disconnecting set is therefore p. However, by Definition 3.2 the digraph can be at most 1-robust since no agent in either of the cliques has more than 1 in-neighbor outside its own clique. 86
3.7 Visualization of the sets $\mathcal{V}_i$ and $\mathcal{V}_{i+b}$, and the values $\alpha_1,\ \alpha_2,\ \beta_1,\ \beta_2$. Here, $i \in S_1$ with S_1 represented by the color blue. From the proof, there exists a node $i + b \in S_2$, with S_2 represented by the color yellow. Nodes $i - k$ through $i - 1$ are either in S_1 or S_2, while nodes $i + 1$ through $i + b - 1$ are not in S_2, i.e. either in S_1 or neither in S_1 nor S_2. 88

3.8 The network topology of digraphs D_1 and D_2. For sake of clarity, only the edges extending from one node are shown; in the actual graph, each node has the same pattern of edges extending from it. The first graph simulated is a $C_{15}\{1,\dots,6\}$ circulant digraph. The second is a $C_{15}\{1,\dots,9\}$ circulant digraph. In the first graph, nodes 1 and 7 are misbehaving. In the second, nodes 1, 7, and 13 are misbehaving. The nodes are visualized in a circular manner for ease of understanding rather than representing any kind of physical arrangement. 92
3.9 Simulation on the graph $\mathcal{D}_1 = C_{15}(1,2,\dots,6)$. The dotted red lines represent the state trajectories of the misbehaving agents. 93
3.10 Simulation on the graph $D_2 = C_{15}(1,2,\dots,9)$. 94
3.11 Illustration of how the (r^*,s^*)-robustness of a graph is found by the *DetermineRobustness* algorithm and the MILP method. Consider a digraph $\mathcal{D}$ of $n=6$ nodes which satisfies $(r^*,s^*)=(2,3)$. *DetermineRobustness* begins with the maximum possible r and s values ($r=\lceil n/2\rceil$ and $s=n$), then iterates in a lexicographically decreasing manner. The MILP formulation first determines $r_{\max}(\mathcal{D})$, then $\bar{s}_{\min}(r_{\max}(\mathcal{D}))$, then finally infers $s_{\max}(r_{\max}(D))$ (abbreviated to $\bar{s}_{\min}(r)$ and $s_{\max}(r)$ for clarity). 104
3.12 Comparison of $DetermineRobustness$ to (r,s)-*Rob. MILP* (Algorithm 3.3). The interpolating lines and circles represents the average computation time in seconds over 100 digraphs for each value of n, the upper and lower lines represent the maximum and minimum computation times, respectively, over the 100 trials for each n. Note that (r,s)-*Rob. MILP* actually solves *two* MILPs sequentially: one to find $r_{\max}(\mathcal{D})$, and one to find $s_{\max}(r_{\max}(\mathcal{D}))$. 118
3.13 Comparison of the *Mod. Det. Rob.* algorithm (Algorithm 3.4) which determines $r_{\max}(\mathcal{D})$ to three MILP formulations. The first MILP formulation labeled *r-Rob. MILP* is an implementation of Theorem 3.4 and calculates $r_{\max}(\mathcal{D})$ exactly. The MILP formulation labeled *r-Rob. Lower Bnd* is an implementation of Theorem 3.6 and calculates a lower bound on $r_{\max}(\mathcal{D})$. The MILP formulation labeled *r-Rob. Upper Bnd* is an implementation of Theorem 3.7 and calculates an upper bound on $r_{\max}(\mathcal{D})$. The interpolating lines and circles represents the average computation time over 100 digraphs for each value of n, the upper and lower lines represent the maximum and minimum computation times, respectively, over the 100 trials for each n. 120
3.14 (Left) Example of a digraph which has $\delta^{\text{in}}(\mathcal{D})=0$ but which is 1-robust. The graph is depicted on the far left, and all possible (S_1,S_2) pairs in $\mathcal{T}$ are depicted on the close left. (Right) Fig A.2. A rooted out-branching, where the in-degree of the root node (far left) is zero. All digraphs containing a rooted outbranching are at least $(1,1)$-robust [2]. 123

4.1 An overview of the trajectory propagation method using parameter vectors. Each leader broadcasts a vector of static parameters representing a Bezier-curve-based trajectory. Followers receive messages from both normally-behaving robots and misbehaving robots. From its received information each follower accepts a parameter vector and uses it to reconstruct a trajectory locally. 130

4.2 Example of the effects of Bezier control point perturbation. The control points for each Bezier curve are represented by the squares, with dotted connecting lines for visual clarity. The actual Bezier trajectories are the solid blue and yellow lines. The magenta lines represent various pointwise differences between points on the curves with corresponding $s \in [0, 1]$ value. 133
4.3 An example of an RDAG with parameter $r = 3$. 144
4.4 An illustration of how the MSRPA algorithm operates in a sychronous setting. The graph depicted is an RDAG with parameter 3 under an F-local adversarial model with $F = 1$. The set of leaders is indicated by the circles in the box on the left, adversarial agents are indicated by the color red, and agents possessing the reference vector of parameters are indicated by the color blue. Leaders begin by broadcasting the reference vector to their out-neighbors. At each time step, any normal follower which receives the same vector message from at least $F + 1$ in-neighbors accepts the vector message and begins rebroadcasting it to its out-neighbors at the next time step. . 145
4.5 Depiction of the method used to specify formational offsets in the simulations. The x-axis of the formation frame $\mathcal{F}_f$ is defined to be colinear with the tangent vector to the Bezier curve at the time-varying reference point $p_f^r(t)$. 156
4.6 The nominal Bezier path for Simulation 1, shown as a solid blue line. The Bezier control points are shown as squares, with dotted lines connecting the control points for clarity of visualization. The exact trajectory is not known to any of the leaders or followers; leaders each have parameter vectors perturbed from the nominal parameters for this trajectory. 158
4.7 Still frames from the video of Simulation 1. The dotted lines represent the reconstructed trajectories for normal robots. The diamonds on the dotted line trajectories represent each robot's reconstructed estimate of the formation reference point. The small x marks represent each normal robots' time-varying desired position $p_i^r(t)$. Two adversarial robots (red) move off towards infinity while simultaneously propagating misinformation through the network. 159
4.8 Plot of the maximum pointwise error between all pairs of normal robot reconstructed target trajectories for Simulation 1, along with the theoretical upper bound. The theoretical upper bound derived in this chapter is quite conservative for the given problem data; future work will investigate ways to tighten this bound. 160
4.9 Plot of the minimum inter-robot distances in Simulation 1. The red dotted line represents the minimum inter-robot distance required for safety to be maintained. 161
4.10 The nominal Bezier path for Simulation 2, shown as a solid blue line. The Bezier control points are depicted as squares, with dotted lines connection the control points for clarity of visualization. As in Simulation 1, this exact trajectory is not known to either leaders or followers. Leaders each have parameter vectors perturbed from the nominal parameters representing this trajectory. 162

4.11 Still frames from the video of Simulation 2. The dotted lines represent the reconstructed trajectories for normal robots. The diamonds on the dotted line trajectories represent each robot's reconstructed estimate of the formation reference point. The small x marks represent each normal robots' time-varying desired position $p_i^r(t)$. Both adversarial robots propagate misinformation throughout the network. One adversarial robots (red) moves off towards infinity while the other remains in place for the entire simulation. 164
4.12 Plot of the maximum pointwise error between all pairs of normal robot reconstructed target trajectories for Simulation 2, along with the theoretical upper bound. The plot begins at time $t = 1.33$ seconds when all normal robots have accepted a parameter vector and reconstructed a trajectory. Again, the theoretical upper bound derived in this chapter is quite conservative for the given problem data; future work will investigate ways to tighten this bound. 165
4.13 Plot of the minimum interrobot distances in Simulation 2. The red dotted line represents the minimum inter-robot distance required for safety to be maintained. 166
4.14 Minimum value of $h_{jo_i}(z_j(t), p_{o_i})$, as defined in (4.52), over all agents $j \in \mathcal{V}$ and obstacles o_1, o_2, o_3 as a function of time in Simulation 2. A log scale is used in the x-axis for greater clarity. This value never decreases below zero, which indicates there were no agent-obstacle collisions for all agents and obstacles. 167
4.15 Depiction of the network structure for the hardware experiments. Agents 1 through 3 are leaders, and agents 4 through 6 are followers. Agent 2 is a misbehaving leader and propagates misinformation to its out-neighbors. 168

5.1 An example of the sets S, ∂S_{ϵ^*}, and $\partial S_{2\epsilon^*}$ for a given $\epsilon^* > 0$. Note that each of the three ellipses is a separate view of the same set S. The dotted blue line in the rightmost ellipse is the inner boundary of ∂S_{ϵ^*}, highlighting the fact that $\partial S_{\epsilon^*} \subset \partial S_{2\epsilon^*}$. 189
5.2 Two examples of initial system states where it is impossible to guarantee forward nonemptiness of the normal agent's feasible controls set $K_i(\cdot)$. Agents have single integrator dynamics; the normal agent is depicted in blue, and adversarial agents are depicted in red. The line in the right image denotes an obstacle. Determining initial conditions for which nonemptiness of the feasible sets is guaranteed for all forward time is intractable in general when considering nonlinear control-affine systems. . . . 198
5.3 Still frames from the video of Simulation 1. Normal agents are represented by blue circles and adversarial agents are represented by red circles. The dotted red lines around the blue circles represent normal agents' safety radii. The time-varying formation trajectory is represented by the dotted magenta line; the magenta diamond represents the center of formation. Black crosses represent agents' nominal local time-varying formational points. 200
5.4 The value of the composed function h_{tot} representing the safe set S. Non-positive values represent safety of the normal agents. 200
5.5 The value of the composed function h_{tot} representing the safe set S when $\eta(\Gamma) = 0$ for all normal agents; i.e., sampling times and disturbances are not accounted for in the control input calculations. The safety bound for the normal agents is violated. . . . 201

5.6 Input values for (normal) agent 2. The blue solid line represents linear input value and the green solid line represents angular input value. Dotted lines represent input bounds. Times at which the worst-case LP is used are marked with red X's on both the linear and angular input lines. 202
5.7 Still frames from the video of Simulation 2. Normal agents are represented by blue circles and adversarial agents are represented by red circles. For clarity, the safety radii of the normal agents has been omitted. The time-varying formation trajectory is represented by the dotted magenta line; the magenta diamond represents the center of formation. Black crosses represent individual agents' nominal local time-varying formational points. Black spheres represent randomly placed obstacles. 205
5.8 The value of the composed function h_{tot} representing the safe set S for all normal agents in the second simulation. Non-positive values represent safety of the normal agents. For the entire duration of this simulation, the value of h_{tot} remains strictly negative, indicating that safety is maintained for all normal agents. 206
5.9 Infinity norm of control input for (normal) agent 2. The control norm bound is plotted in red, and the norm of agent 2's control input is plotted in blue. Times when the backup LP is used are marked with red X's. 207

Abstract

The capabilities of and demand for complex autonomous multi-agent systems, such as networks of unmanned aerial vehicles and robots, are rapidly increasing in both research and industry settings. As the size and complexity of these systems increase, dealing with faults and failures becomes a crucial element that must be accounted for when performing control design. In addition, the last decade has witnessed an ever-accelerating proliferation of adversarial attacks on cyberphysical systems across the globe. Unlike typical disturbances and noise considered in traditional control design, these adversarial attacks may not exhibit a predictable structure and are usually specifically designed to exploit and manipulate vulnerable elements in the targeted system.

In response to these challenges, recent years have seen an increased focus on *resilience* of multi-agent systems to faults and adversarial attacks. Broadly speaking, resilience refers to the ability of a system to accomplish control or performance objectives despite the presence of faults or attacks. Ensuring the resilience of cyberphysical systems is an interdisciplinary endeavor that can be tackled using a variety of methodologies. This dissertation approaches the resilience of such systems from a control-theoretic viewpoint and presents several advancements in resilient control methodologies. More specifically, this dissertation contains the following contributions and developments.

First, several novel advancements are given for resilient consensus in multi-agent systems using computationally inexpensive, distributed algorithms. Both leaderless and leader-follower consensus algorithms are fundamental methods for achieving cooperation and agreement on information among distributed agents in a broad variety of practical settings, including information

fusion in distributed sensor networks, rendezvous and formation tracking in networks of mobile robots, clock synchronization between distributed computers, cooperative surveillance with multiple UAVs, synchronization of oscillator networks, transaction commit protocols, opinion dynamics in social networks, and more. The consensus problem becomes more challenging however when a subset of the agents do not behave according to the nominally specified consensus control law due to faults or adversarial attacks. The resilient consensus problem studies how to guarantee consensus of normally-behaving agents despite such faults and attacks. We present conditions under which resilient leader-follower asymptotic consensus in time-varying graphs can be achieved in the presence of adversarially-behaving agents using a control law and information filtering from the class of Mean-Subsequence-Reduced family of algorithms. In addition, we present a novel algorithm for achieving finite-time leader-follower consensus in a continuous-time system setting. Finally, we present a general method for achieving finite-time leaderless consensus in a continuous-time setting using a class of nonlinear controllers. The analysis utilizes discontinuous systems theory and considers a more general model of adversarial distribution and behavior than prior work.

Second, novel graph theoretic tools for constructing and analyzing resilient networks are presented. Prior results on resilient control methods often require strict conditions on the underlying communication topology of the network of agents to guarantee the achievement of system objectives. Determining to what extent given networks satisfy these conditions is often a computationally challenging problem, and constructing graphs satisfying these conditions by design can also be difficult. This dissertation presents methods to ameliorate both of these challenges. To address the issue of construction, a class of circulant graphs is proposed whose resilience properties depend solely on a design parameter independent of the size of the graph. Such graphs can be scaled to an arbitrarily large number of nodes while preserving desired resilience properties. In addition, a mixed-integer linear programming (MILP) optimization framework is presented that can be used to determine the exact resilience properties of arbitrary directed and undirected graphs. These MILP

methods allow for the iterative approximation of resilience parameters and demonstrate reduced computation time in practice as opposed to prior techniques.

Third, an algorithm is proposed for resiliently broadcasting vector-valued information from a set of leaders to a set of followers in the presence of adversarial misinformation. This algorithm can tolerate both misbehaving leaders and followers, and can operate under a dynamic graph model for the network topology. The algorithm can operate even when the vector values of the normally-behaving leaders do not exactly agree due to noise or perturbations, and it is proven that bounded perturbations or noise results in bounded maximum error between normally-behaving leaders' and followers' accepted values. In addition, we present a novel application of resiliently propagating time-varying trajectory information in the form of Bezier curve parameters from a set of leaders to followers.

Finally, we present a novel framework for resilient safety maintenance of multi-agent, distributed, sampled-data systems in the presence of adversaries using Control Barrier Functions (CBFs). CBFs combined with quadratic programming (QP) methods have arisen in the last decade as a powerful method for computing control inputs that guarantee safety and set forward invariance for nonlinear control affine systems. Prior work has typically assumed that the control inputs resulting from the associated quadratic programs are continuous in time; however the behavior of practical systems is more accurately modeled by sampled-data dynamics. In addition, none of the prior literature on multi-agent CBF set invariance considers the effects of adversarially-behaving agents seeking to violate system safety constraints. The framework presented in this thesis is the first to present a method for normally-behaving agents in a multi-agent system to compute safety-preserving control inputs despite the actions of adversarial agents. This framework considers a general class of nonlinear, control-affine, sampled-data dynamics with disturbances and asychronous communication between agents and input bounds. In addition, it considers CBFs having high relative degree with respect to agents' dynamics.

CHAPTER 1

Introduction

1.1 Motivation

The first twenty years of the 21st century have witnessed an explosion in both technological advances and distribution of cyberphysical systems. These systems have become ubiquitous in the settings of academia, industry, and the everyday lives of billions of people across the globe. Within the technological transformation that has taken place since the dawn of the new millenium, two prominent trends stand out.

First, a wide variety of distributed control systems have rapidly proliferated in both academic and industrial settings. The rise of the Internet of Things has led to the incorporation of sensors, software, and control algorithms into a vast variety of physical objects, with the number of such devices projected to reach 43 billion by 2023 [3]. Many of these devices operate with limited power and computational resources. A similar rise in multi-agent systems has occurred in academia and industry, including multi-UAV systems for surveillance [4], mapping [5], atmospheric sensing [6], outdoor light shows [7], and military applications [8]; large-scale deployment of autonomous robots in places such as Amazon warehouses [9] and British supermarket company Ocado's Andover warehouse [10]; autonomous vehicle systems in mining [11], vehicle platooning [12], and commercial autonomous passenger vehicle research [13]; and many more settings.

Second, there has been an exponential rise in the number of attacks made on cyberphysical systems. The global cost of cybercrime for 2021 has been projected to reach USD $6 trillion annually, with this estimate growing to USD $10.5 trillion annually by 2025 [14, 15]. An estimated 9.9 billion cyberattacks worldwide took place in 2019 alone. Since the vast majority of control systems are implemented via computers or embedded systems, these systems have become vulnerable targets for attackers. Notable examples of attacks on cyber-physical control systems include Stuxnet in 2009 [16], a cyberattack attack on an RQ-170 military UAV in 2011 [17], multiple attacks conducted on consumer automobiles [18–20], and the hacking of multiple UAVs by the Russian military during an insurgent attack on a Syrian base [21].

Two conclusions can be drawn from this trend: first, the increase in number of cyberphysical systems will be accompanied by a rise in faults and failures of such systems. Systems composed of large numbers of agents, sensors, or processors will inevitably see failures in functionality and collaboration between those entities. Second, ensuring systems can withstand adversarial manipulation and attacks must become a fundamental consideration when designing or choosing the control algorithms responsible for enabling the system achieve its objectives. The necessity of overcoming faults and adversarial attacks has driven an increase in focus on the topic of *resilience* in control systems.

The word *resilience* has an extremely wide variety of meanings across multiple disciplines and contexts. Giving a satisfactory definition general enough to capture all of these meanings is beyond the scope of this work. However to describe how the term resilience is used in this dissertation, we consider a problem setting to be composed of a *system model*, a *nominal objective*, and an *adversarial model*. Informally speaking, a system is resilient if it can guarantee achievement of the nominal objective despite all possible adversarial actions described by the adversarial model. The study of resilience can be approached from an infinite number of perspectives and using tools from countless fields and specialties including control theory, game theory, computer science, artificial intelligence, biology, mathematics, statistics, and more. This dissertation does not attempt to approach resilience from all of these angles. Rather, this dissertation will present several contributions towards desigining resilient systems from a control theoretic perspective.

1.2 Literature Review

We first give an overview of several approaches in prior literature towards creating resilient systems. This overview is far from exhaustive; rather, we chiefly cover areas that are related to the contributions of this dissertation.

1.2.1 Origins of the Resilient Agreement Problem

Achieving agreement on common information by a set of agents has been a fundamental problem in both computer science and control theory for decades [22–25]. These problems, which are known by different names such as *agreement problems* or *consensus problems*, arise in many practical scenarios. In distributed computing the agreement problem underlies commit protocols [26], file copy storage [27], clock synchronization [28], among others. In the controls community, consensus is fundamental to distributed sensing [29], distributed observers [30, 31], rendezvous of mobile agents [32–34], formation control [35], flocking [36], leader-follower control of mobile agents [37], synchronization of biological and engineering oscillator networks [38], among others.

Consensus has also been addressed in cases when various elements of the system are subject to stochasticity and noise. Several works have studied a graph theoretic model where edge weights behave stochastically over time to model uncertainty in communication between agents [39, 40]. Other works have modeled signals between agents as having stochastic additive noise [41, 42].

Agreement and consensus protocols in the presence of faults or attacks have been long studied in computer science [22, 43]. A seminal work that sparked interest in agreement algorithms resilient to faults and adversaries is [44], which introduces the *Byzantine Generals Problem*. This problem considers several computer processors seeking to achieve agreement on a "correct" common value broadcasted by a leader processor when a subset of the processors, called "Byzantine", either stop working or communicate deliberate misinformation to their neighbors within the network. The recursive algorithms presented in this work require either signed messages to verify the identity of the sender, or they require each agent to possess global information about the structure of the communication graph. Many later papers built upon this idea of agreement in the presence of faults and adversarial behavior including [45] which extended the results to more general graph conditions; [46] which considers the weaker problem of agents simply coming to agreement on a value that may not be the "correct" one; [47] which uses a randomized lottery procedure and asymmetric cryptographic message verification; [48] which considers clock synchronization in the presence of Byzantine faults, among many others. The proposed solution in [44] to the Byzantine Generals problem has the advantage of guaranteeing exact agreement of normally-behaving agents in a finite number of time steps. However, one key shortcoming of this solution is that all normal agents must be able to reliably communicate information to all other normal agents in a particular sense. This requires the communication topology to be either a complete network, or agents are required to have nonlocal knowledge about the communication topology structure. More specifically, in the noncomplete case each agent is required to have knowledge of redundant, separate paths from itself to any other agent in the network.

One widely cited result with regards to consensus of computer processes in the presence of faults (and possibly Byzantine processors) is [49]. This work shows that it is impossible for asynchronous computer processes under a general model to achieve agreement on a common value. It is important however to understand the assumptions made upon the system model and the authors' definition of consensus; in particular, consensus for a system of message-passing processes is defined as reaching a decision on a particular value in finite time. Consensus in other fields (e.g. control theory) typically assumes a different state update model and/or *asymptotic* consensus, where agents' states converge to a common value as time tends to infinity.

The computer science community has presented many other algorithms for consensus of multiple "agents" or processors in the presence of various fault, asynchrony, and adversarial models. The works [50–53] outline algorithms for multi-agent vector consensus of asynchronous systems in the

presence of Byzantine adversaries. The Paxos algorithm [54, 55] presents a method for a network of agents to come to common agreement on values via a *propose, learn, accept* model in which a set of one or more *proposers* processes propose values to be agreed upon, a set of one or more *acceptors* processes collaboratively choose a proposed value, and a set of one or more *learners* processes learn the value that has been chosen. The algorithm considers asynchrony, temporary or permanent agent failure, and message delays / losses. The main disadvantage of the original Paxos algorithm, however, is that it cannot tolerate adversarial misinformation or corrupted messages. Some later studies have built upon the Paxos algorithm including [56] which presents a variant of Paxos guaranteeing learning of the consensus value within two message delays; [57] which considers the presence of Byzantine acceptors (but not proposers or learners) which can communicate false messages within the Paxos algorithm; [58] which presents a specialized version of Paxos for systems with $(2F + 1)$ processors which can tolerate up to F faults; among others. Even with the modifications for Byzantine resilience proposed in [57], there are two disadvantages for using the Paxos algorithm in certain settings. First, the Paxos algorithm requires proposed consensus values to be uniquely numbered with a total order on the numbering values, with higher ballot numbers receiving precedence over lower ballot numbers. With Byzantine adversaries, there are difficulties with guaranteeing that Byzantine leaders do not execute higher-numbered ballots to disrupt operation of the algorithm [57]. In addition, the Paxos algorithm chooses only one proposed value, typically one of the earliest proposed values, and discards all other proposed values. The value selected depends on the timing for which possible candidate values are proposed to the acceptor agents, with the algorithm biased towards selecting a value proposed earlier than other values. In some contexts such as sensor fusion however, allowing multiple sources to influence the final selected value may give a better estimate of the actual data being observed (e.g., the mean of multiple proposed values) than simply selecting the earliest proposed value by one of the sensors. Blockchain methods have also been used for decentralized Byzantine-resilient consensus [59–62]. The practical efficacy of these methods for implementing applications such as decentralized cryptocurrencies [63, 64], distributed ledgers [65, 66], and smart contracts [67] has been widely demonstrated. However, these methods possess their own vulnerabilities including the notorious *51% attack* and *selfish miners attack* [68]. In addition, blockchains relying upon proof-of-work methods for verification inherently require the expenditure of fairly significant time, memory, and computational effort to implement. This time and computational effort can be at odds with systems having limited energy and computational resources, and systems seeking to converge to consensus in a rapid manner.

Building upon the Byzantine Generals Problem, the paper [69] considers the problem of a leader agent securely broadcasting a value in a network with agents behaving in a Byzantine manner. This work proposes the *Certified Propagation Algorithm* (CPA), which operates in a

synchronous manner and guarantees that the original message will eventually reach all normal followers in the network if adversaries are able to corrupt no more than t of their neighbors, where t is a nonnegative integer. This result is extended in [70] to be able to handle message collisions and adversarial address spoofing. The work [71] demonstrates that the graph-theoretic condition of r-robustness on the network topology is a sufficient condition for the CPA algorithm to succeed for any arbitrarily chosen leader node in the network. Considering a similar problem scenario, [72] studies tighter bounds on the number of total corrupted agents t that the CPA can tolerate for particular graph structures, and characterizes the relationship of graph parameters for these structures with the number of corrupted agents t. In [73] the adversarial model is extended to the t-*local* model where the number of Byzantine agents in any normal agent's neighborhood is no more than t. The paper also studies more general resilient broadcast algorithms that consider *safety* of agents' accepted values; i.e., algorithms that guarantee that no agent will ever accept an incorrect or false value, even if agents do not come to consensus.

1.2.2 MSR Algorithms for Resilient Consensus

The Mean-Subsequence-Reduced (MSR) family of algorithms is introduced in [74] to address several disadvantages of algorithms solving the Byzantine Generals problem. Specifically, the authors seek to reduce the number of messages required and the complexity of the contents of each message. The salient characteristic of the MSR family of algorithms is that each agent updates its current information based on a *trimmed mean* of a subset of the information it receives from its neighbors. Building upon the concept of MSR algorithms, the seminal work [75] introduces both the *Weighted Mean-Subsequence-Reduced* (W-MSR) algorithm and the graph-theoretic notions of r- and (r, s)-robustness. These notions provide both necessary and sufficient conditions on when the W-MSR algorithm guarantees asymptotic convergence of a set of agents to a consensus value in the presence of a variety of adversarial models. The conditions in this work are closely related to those in [50], which also considers iterative approximate consensus in the presence of Byzantine agents. The introduction of the W-MSR algorithm sparked the development of a wide array of MSR algorithms specialized for various problem settings. Variants include the DP-MSR algorithm for zero-order-hold (ZOH) discretized continuous-time systems [76, 77], the QW-MSR algorithm for systems with quantized states [78], the SW-MSR algorithm for systems with dynamic graphs [79], the E-MSR and QE-MSR algorithms for event-based consensus [80, 81], and the MP-MSR algorithm which incorporates resilience into a model predictive control framework [82]. Resilient MSR-type consensus for continuous-time models include the ARC-P algorithm [83], the RAC algorithm for resilient synchronization of linear systems [84], the CT-MSR algorithm [85], a hybrid systems version of the W-MSR algorithm [86], and a discontinuous variant based on the

sign function called FTRC-P [87], among others [88]. MSR-type algorithms have been studied in a variety of settings including output synchronization [84], simultaneous arrival of interceptors [89], distributed optimization [90], clock synchronization [91], randomized quantized consensus [92], event-triggered consensus [80, 93], and differentially private consensus [94, 95].

The rise of interest in MSR-type algorithms was accompanied by an increased focus on the graph-theoretic properties of r-robustness and (r, s)-robustness, which are commonly part of the sufficient conditions for MSR-type algorithms to guarantee resilient consensus. Determining the values of r and s for which a graph is robust is NP-hard in general [96]; determining whether a given graph is r robust for a particular r is coNP-complete [97]. Subsequent work has focused heavily on circumventing this difficulty through various methods. The first algorithmic analysis of determining the values of r and s for arbitrary digraphs was given in [96]. The algorithms proposed in [96] employ an exhaustive search to determine the maximum values of r and s for a given digraph, and have exponential complexity w.r.t. the number of nodes in the network. Other methods to determine the robustness of graphs include graph construction methods that increase the graph size while preserving given values of r and s [2, 98]; lower bounding r with the isoperimetric constant and algebraic connectivity of undirected graphs [99]; and demonstrating the behavior of r as a function of certain graph properties [97, 100–104]. In particular, it has been shown that the r-robustness of some specific classes of graphs can be exactly determined in polynomial time from certain graph parameters. Examples include k-circulant graphs [102] and lattice-based formations [103, 104]. Another approach has used machine learning to correlate characteristics of certain graphs to the values of r and s [105]. Out of all of these prior efforts, none have used an optimization framework to determine or approximate the robustness properties of graphs, and none have even suggested that this might be possible. In addition, no methods exist in prior literature for lower bounding the robustness parameters of arbitrary directed graphs. Finding more efficient methods to determine the robustness of arbitrary graphs, especially directed graphs, remains an important open problem.

1.2.3 Resilient Safety and Control of Dynamical Systems

Within the controls community, several approaches to mitigating the effects of faults and adversarial attacks have been proposed. One of the earliest and most exhaustively studied topics studied in control theory that has relevance to designing resilient systems is the topic of robust control [106–109]. The essence of robust control is to guarantee system performance for a wide class of system model parameters and disturbances due to system uncertainty. Informally speaking, the chief distinguishing element between robust control and resilient control is the difference between the set of parameters / disturbances (in robust control) and the adversarial model (in resilient con-

trol). Typically, robust control considers uncertainties in the model parameters and disturbances that are inherently bounded in nature. Disturbances to the system are generally, roughly speaking, oblivious and indifferent to both the system objectives and the control actions and intents of the agents that compose the system. In contrast, resilient control considers an adversarial model in which adversaries have the express intent to prevent or degrade the accomplishment of system objectives. Adversarial actions or misinformation may not be bounded in nature and typically exploit weaknesses or vulnerabilities in the system. Despite these differences however, there is a large overlap between the domains of resilient and robust control. In [110], H_∞ control protocols are used to mitigate the effects of attacks on sensors and actuators of a dynamical system. In [111, 112] the robust minimal controllability problem is considered, where the objective is to find the minimum number of actuators for a linear system under which the system is controllable.

Motivated by the vulnerability of power grids and other cyber-physical systems to cyber attacks, much work has focused on the detection and mitigation of several types of attacks on networked control systems. System models considered in this literature tend to be linear; however centralized, decentralized, and distributed types of networked systems are all considered. The work [113] brought attention to why control theorists should be interested in the security of control systems, and formulated the problems of detecting and surviving attacks. Several control theoretic models of the effects of cyber attacks on linear control systems are given in [114,115]. In [116,117] a comprehensive theory of the detectability and identifiability of cyberattacks in networked cyber-physical systems is given, along with an analysis of the complexity of attack identification for these systems. Attack detection for networked cyber-physical systems has been well-studied by additional works as well [117–121]. Another interesting direction being pursued is the use of ℓ_0 norm decoding methods for both identification of attacks and error correcting of the corrupted state measurements. This problem was introduced in [122], which used results from the error correcting literature [123] to create a method for reconstructing the state and adversarial attack pattern for a linear dynamical system. This work prompted later variants of resilient state estimators including estimators incorporating noise and stochasticity [124], linear and nonlinear resilient Kalman filters [125], and approaches using satisfiability modulo theory [126]. Game theoretic approaches to mitigating attacks on cyber-physical systems have also been studied [127, 128].

Safety in dynamical systems is often modeled in the literature as dividing the state space into disjoint "safe" and "unsafe" sets of states. The question of safety then becomes whether or not it is possible for the system state to reach an unsafe set of states from a given initial configuration. A foundational result for determining the forward invariance of subsets of the state space for dynamical systems is Nagumo's Theorem [129]. The setting of differential games was another of the first approaches to address the question of safety in continuous systems [130, 131]. Other approaches for verifying safety involve calculating forward reachable sets using various methods including el-

lipsoidal calculus [132], flow pipe approximations [133,134], Hamilton-Jacobi analysis [135–138], and zonotope approximation algorithms [139–142]. These methods can be computationally expensive to implement in practice. To circumvent this difficulty, barrier certificates were proposed as a means for verifying safety without requiring computation of forward reachable sets [143–147]. The crux of the method is to determine a differentiable function of the system state that 1) takes on nonpositive values for states within the safe set, 2) takes on strictly positive values for states in the unsafe set, and 3) has a nonpositive time derivative under the flow of the dynamical system. Such a function satisfying these three conditions serves as a *certificate* that the state remains within the safe set for all forward time, and hence is called a *barrier certificate*. Lyapunov-like barrier functions [148–150] and potential function methods [151] have also been studied for guaranteeing forward invariance of safety sets. With regards to studying safety in the presence of adversarial actions, extensive study in the game theoretic domain has been performed on conditions under which a subset of agents ("evaders") in a system can evade collisions with or capture by another subset of agents ("pursuers") in *pursuit-evasion games*. Such a setting can be used to model "normal" agents avoiding physical attacks from "adversarial" agents, depending on the context. Some examples include the homicidal chauffeur problem [130, 152], multi-agent homicidal chauffeur variants [153], the suicidal pedestrian problem [154], the lion and man problem [155], among others. The simplest of these games admit closed-form solutions for agents' optimal control inputs. Methods to compute optimal control inputs in more complex games include solving the Hamilton-Jacobi-Isaacs equations [156] and computing dominance regions [157–159].

A major development in safety and set invariance methods within the last decade has been the creation of *control barrier functions* (CBFs) [160, 161]. First proposed in [162] and inspired by the concept of control Lyapunov functions [163], CBFs are a class of functions similar to barrier certificates, except that forward invariance can be established by considering the infimum of the time derivative of the CBF over an entire set of feasible agent inputs. When paired with parametric convex quadratic programming methods to compute feasible inputs satisfying the required conditions for forward invariance [161, 164, 165], CBFs have proven to be a powerful method for computing inputs online that guarantee forward invariance of a safe set in a computationally efficient and tractable manner. CBFs can be used in tandem with control Lyapunov functions to compute control inputs that safely take the system state to a goal set [161, 166, 167]. CBFs have been used in many applications including coordinating the motion of multi-agent systems [168–170], accomplishing spatiotemporal tasks [171–174], calculating control inputs for legged robots [165, 174, 175], and more. There are a few limitations to prior work on CBFs however. First, the majority of prior work assumes that control inputs are continuous functions of time. In particular, these works assume that the parametric convex quadratic program (QP) is solved infinitely often and that the resulting control input from the $\arg\min$ operation is continuous. A more accurate model is to treat the system

dynamics in a sampled-data manner, e.g. with zero-order-hold inputs (ZOH). A few works have explicitly considered sampled-data dynamics for control barrier functions [176, 177]. However, these works do not consider systems where the CBF function has high relative degree with respect to the given system dynamics; i.e., the control input does not appear in the first time derivative of the CBF function. In addition, these works do not consider multi-agent systems where control inputs must be calculated in a distributed manner and communication may be asynchronous. Finally, prior work considering multi-agent systems (with continuous controllers) do not consider the presence of agents behaving in an adversarial manner. Typically any disturbances to the system are assumed to be bounded in nature and do not have adversarial intent [178, 179].

1.3 Contributions

The objective of this dissertation is to present several contributions towards enabling multi-agent systems to operate despite a subset of the agents within the system behaving in a faulty or adversarial manner. Specifically, this dissertation presents several novel results pertaining to the problems of resilient consensus and resilient safety maintenance in multi-agent systems. The specific contributions of this dissertation are as follows:

1. Several novel advancements are given for resilient consensus in multi-agent systems using computationally inexpensive, distributed algorithms. We present conditions under which resilient leader-follower asymptotic consensus in time-varying graphs can be achieved using an MSR algorithm in the presence of adversarially-behaving agents. Unlike other approaches in prior literature, this method does not require any global information about the graph theoretic structure of the network, is computationally simple to perform, and is resilient to a subset of leaders as well as followers behaving in a faulty or adversarial manner. In addition, we present a novel method for achieving finite-time leader-follower consensus in a continuous-time system setting. Unlike prior literature, we present a novel norm-based filtering mechanism that explicitly guarantees a dwell time for the switching dynamics induced by agents performing filtering actions, and we consider bounds on the agents' control inputs. Finally, we present a general method for achieving finite-time leaderless consensus in a continuous-time setting using a class of nonlinear controllers. The analysis utilizes discontinuous systems theory and considers a more general model of adversarial distribution and behavior than prior work. In particular, any Lebesgue-measurable adversarial signals for an F-local class of Byzantine adversaries can be considered.

2. Novel methods for constructing and analyzing the resilience properties of communication graphs are presented. In particular, we present results showing that a class of circulant di-

rected graphs parameterized by an integer k have r and s robustness parameters as a function of k. This class of graphs greatly simplifies the creation of scalable networks of arbitrary size with known robustness. We also present a novel method for determining the r- and (r, s)-robustness for arbitrary digraphs and undirected graphs using mixed integer linear programming. This is the first work to demonstrate that the robustness determination problem can be solved using optimization methods, and it is the first work to give a method for calculating an approximate lower bound on the robustness values of general digraphs. Techniques from the mixed integer programming literature can be used to either determine exactly or approximate the robustness of digraphs, and simulation results demonstrate that the optimization framework outperforms prior algorithms for robustness determination in practice.

3. Inspired by the Certified Propagation Algorithm (CPA), an algorithm is proposed for resiliently broadcasting vector-valued information from a set of leaders to a set of followers in the presence of adversarial misinformation. Prior work on resilient broadcast using the CPA typically assumes that there exists only one leader that is invulnerable to adversarial attacks. In contrast, the proposed algorithm uses a multi-leader approach that can tolerate both misbehaving leaders and followers, and can operate under a time-varying graph model for the network topology. Unlike prior works, the algorithm can operate even when normally-behaving leaders' vector messages do not exactly agree due to noise or perturbations, and it is proven that bounded perturbations or noise results in bounded maximum error between normally-behaving leaders' and followers' accepted values. We apply this algorithm to the problem of resiliently propagating full knowledge of time-varying trajectories in the form of Bezier curve and timing law parameters from a set of leaders to followers. More specifically, we consider a system of mobile agents (e.g., robots) connected via a wireless communication network. Leaders use the resilient broadcast method presented in this section to propagate to the followers the parameters for a center of formation moving along a time-varying trajectory described by a Bezier curve and timing law. Followers are then able to use the received parameters to reconstruct their local time-varying formational positions such that normally-behaving leaders and followers in the system move in formation along the time-varying trajectory.

4. We present a novel framework for resilient safety of multi-agent, distributed, sampled-data systems in the presence of adversaries using methods from the literature on Control Barrier Functions (CBF). Prior work has typically assumed that control inputs are continuous in time and that all agents in a multi-agent CBF setting cooperate to preserve system safety. In contrast, we present conditions under which normally-behaving agents in a sampled-data system are able to maintain safety despite the actions of an adversarial set of agents seeking

to violate safety conditions. Our method considers nonlinear control-affine sampled-data dynamics with disturbances, and presents a computationally tractable method for normally-behaving agents to compute safety-preserving, bounded control inputs in a distributed manner. In addition, we consider safe sets having high relative degree with respect to agents' dynamics. This the first work to consider the presence of adversarial agents in a CBF setting and the first work to consider multi-agent safe sets having high relative degree with respect to system dynamics in a sampled-data setting. We demonstrate through simulations that normally-behaving agents applying our method can maintain safety despite the actions of adversarial agents within the system.

These contributions have several advantages compared to prior approaches, as well as a few disadvantages. We discuss these advantages and disadvantages here to give some perspective on why these approaches and contributions were pursued.

First, one key advantage of MSR-type algorithms over prior proposed solutions for resilient consensus is that they require purely local information to operate. Agents are not required to have any knowledge of the network topology; rather, the MSR algorithm is able to operate using only the local information received from each agents' immediate in-neighbors. In addition, the asymptotic convergence behavior of MSR-type algorithms can be leveraged towards achieving resilient consensus in both asychronous computer systems and physical control systems. When comparing MSR-type algorithms specifically to Paxos-type algorithms, there are two main advantages of MSR-type algorithms. MSR-type algorithms do not require individual messages to be uniquely numbered with a total order; this enables MSR-type algorithms to tolerate Byzantine leaders more easily than Byzantized Paxos variations. Second, in the leaderless case MSR-type algorithms do not exhibit the same bias towards simply selecting earlier-proposed values as Paxos-type algorithms do. Since MSR-type algorithms use a trimmed-mean approach for consensus, they are able to combine information from multiple agents into a final approximate consensus result while still exhibiting resilience to Byzantine misinformation. Disadvantages of MSR-type algorithms include the fact that MSR algorithms with asymptotic or exponential convergence rates can only guarantee *approximate* resilient consensus in finite time. In addition, MSR-type algorithms require specific conditions on the network structure which are computationally difficult to determine for the leaderless case. For communication networks defined in terms of spatio-temporal conditions, such as proximity-based communication models, ensuring that such conditions are satisfied using only local information can also be challenging.

With regards to constructing and analyzing the resilience properties of networks, the main advantage of the proposed circulant graphs is the simplicity of determining their robustness and their ability to be scaled to any arbitrary network size. The chief disadvantage of this approach is simply the specific form that is required for this class of graphs. For robustness determinination,

there are several advantages of the mixed-integer linear programming framework for determining r- and (r, s)-robustness as compared to prior literature. As mentioned previously, this framework is the first to apply optimization methods to the robustness determination problem. In particular, this framework allows the robustness problem to benefit from ongoing innovations in the highly active research area of integer programming. It also allows highly optimized commercial solvers to be applied towards determining graph robustness. In addition, this results enables for the first time the approximation of lower bounds on the robustness of arbitrary directed graphs, which was not possible with prior results. The main disadvantage of the mixed integer linear programming approach is the worst-case complexity of solving such problems (NP-hard). However since robustness determination is also NP-hard, this approach does not increase the worst-case complexity of robustness determination.

With respect to resilient broadcast, we chose to develop a variant of the CPA algorithm since it only requires local information, and it is optimal among resilient broadcast algorithms using only local information [73]. The advantages of using the CPA algorithm are its simplicity, its ability to operate using only local information, and its ability to operate even with asynchronous communication between agents. In addition, our proposed method for resiliently propagating entire knowledge of a time-varying trajectory is the first of its kind; prior work typically only considers the communication of static information. The main drawback to this approach is the specific conditions on the communication network required for CPA to achieve resilient broadcast, which are discussed in Chapter 4.

Finally, with respect to resilient safety of multi-agent, distributed systems using Control Barrier Function (CBF) methods, the main advantage of CBF methods is their balance of rigorous theoretical guarantees on safety and computational efficiency when applied in practice. Unlike prior literature, CBF methods do not require the calculation of forward reachable sets, and do not require the computation of game-theoretic optimal control inputs or dominance regions, which can be computationally expensive or prohibitive in practice. In contrast, CBF methods allow for control inputs that provably guarantee forward invariance of a set to be computed using computationally efficient convex Quadratic Programming (QP) techniques, under appropriate assumptions that the safety preservation problem is feasible. With respect to modeling agents as having sampled-data dynamics, there are two advantages as compared to prior literature: first, sampled-data dynamics more closely match the dynamics of practical control systems, which are most commonly implemented using computer systems. These dynamics also reflect the inherent limitation that the convex optimization problems computing control inputs require a nonzero amount of time to run. Prior work typically ignores this limitation and assumes that the optimization problems are solved infinitely often. Second, sampled-data dynamics ensure the existence and uniqueness of solutions to the ordinary differential equations modeling the evolution of system states. This is particularly

useful when control inputs are computed via convex parametric QPs, since it can be difficult in general to prove that the optimal point of a constrained, parametric QP is locally Lipschitz continuous with respect to the QP parameter. Disadvantages to the CBF approach include the myopic nature of the algorithm, which considers optimality of the control input only for the given instant and does not consider a time horizon, and the open question of guaranteeing forward nonemptiness of the set of safety-preserving control inputs.

1.4 Outline

This dissertation is organized as follows: Chapter 2 presents our results on resilient consensus in multi-agent systems. It is based upon the works [87, 180–182]. Chapter 4 presents our method for resiliently broadcasting vector-valued information from a set of leaders to a set of followers in the presence of adversarial misinformation. It is based on the work [183]. Chapter 3 presents our method for constructing resilient graphs and analyzing the resilience properties of communication graphs using mixed integer linear programming. It is based upon the works [102, 184–186]. Chapter 5 presents our framework for resilient safety maintenance of multi-agent, distributed, sampled-data systems in the presence of adversaries using Control Barrier Functions. It is based on the work [187, 188].

1.5 Notation

The following notation will be used throughout this dissertation. The sets of real numbers, integers, and natural numbers are denoted $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$ respectively. The sets of nonnegative real numbers and integers are denoted $\mathbb{R}_{\geq 0}$ and $\mathbb{Z}_{\geq 0}$, respectively. $\mathbb{R}^n$ denotes an n-dimensional vector space over the field $\mathbb{R}$, $\mathbb{Z}^n$ represents the set of n dimensional vectors with integer entries, and $\{0,1\}^n$ represents a binary vector of dimension n. Scalars are denoted in normal text (e.g. $x \in \mathbb{R}$) while vectors are denoted in bold (e.g. $\boldsymbol{x} \in \mathbb{R}^n$). The notation x_i denotes the ith entry of vector $\boldsymbol{x}$. The inequality symbol $\preceq$ denotes a componentwise inequality between vectors; i.e. for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{x} \preceq \boldsymbol{y}$ implies $x_i \leq y_i \; \forall i \in \{1, \ldots, n\}$. A vector of ones is denoted $\mathbf{1}$, and a vector of zeros is denoted $\mathbf{0}$, where the length of each vector will be implied by the context. The notations $B(x, \epsilon)$, $\bar{B}(x, \epsilon)$ denote the open and closed balls of radius $\epsilon > 0$ at $x \in \mathbb{R}^d$, respectively. The ith column of the identity matrix I is denoted e^i, with $I = \left[e^1 \; e^2 \; \ldots \; e^n \right]$.

The union, intersection, and set complement operations are denoted by $\cup$, $\cap$, and $\setminus$, respectively. The cardinality of a set S is denoted as $|S|$, and the empty set is denoted $\emptyset$. The infinity norm of a vector is denoted $\|\cdot\|_\infty$. The notations $C(n,k) = \binom{n}{k} = n!/(k!(n-k)!)$ are both used in this paper to denote the binomial coefficient with $n, k \in \mathbb{Z}_+$. Given a set S, the power set of S

is denoted $\mathcal{P}(S) = \{A : A \subseteq S\}$. The convex hull of a set S is denoted $\text{co}\{S\}$, and the convex closure of a set S is denoted $\overline{\text{co}}\{S\}$.

Given a function $f : D \to R$, the image of a set $A \subseteq D$ under f is denoted $f(A)$. Similarly, the preimage of $B \subseteq R$ under f is denoted $f^{-1}(B)$. The logical OR operator, AND operator, and NOT operator are denoted by $\vee, \wedge, \neg$, respectively. The lexicographic cone is defined as $K_{lex} = \{0\} \cup \{\boldsymbol{x} \in \mathbb{R}^n : x_1 = \ldots = x_k = 0,\ x_{k+1} > 0\}$ for some $0 \leq k < n$. The lexicographic ordering on $\mathbb{R}^n$ is defined as $\boldsymbol{x} \leq_{lex} \boldsymbol{y}$ if and only if $\boldsymbol{y} - \boldsymbol{x} \in K_{lex}$, with $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ [189, Ch. 2]. The sign function ($\text{sign} : \mathbb{R} \to \mathbb{R}$) is defined as follows:

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0\ , \\ -1 & \text{if } x < 0 \end{cases} \quad x \in \mathbb{R} \tag{1.1}$$

A directed graph (digraph) is denoted as $\mathcal{D} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of indexed vertices and $\mathcal{E}$ is the edge set. This dissertation will use the terms vertices, agents, and nodes interchangeably. A directed edge is denoted (i, j), with $i, j \in \mathcal{V}$, meaning that agent j can receive information from agent i. Agent i is called an *in-neighbor* of j and agent j is called an *out-neighbor* of i. The set of in-neighbors for an agent j is denoted $\mathcal{V}_j = \{i \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. The set of inclusive in-neighbors is defined as $\mathcal{J}_i = \mathcal{V}_i \cup \{i\}$; i.e. $\mathcal{J}_i$ contains both the agent i and its in-neighbors. The set of out-neighbors of each agent i is denoted $\mathcal{V}_i^{out} = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. The minimum in-degree of a digraph $\mathcal{D}$ is denoted $\delta^{in}(\mathcal{D}) = \min_{j \in \mathcal{V}} |\mathcal{V}_j|$. Occasionally, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ will be used to denote an undirected graph, i.e. a digraph in which $(i, j) \in \mathcal{E} \iff (j, i) \in \mathcal{E}$. The graph Laplacian L for a digraph (or undirected graph) is defined as follows, with $L_{i,j}$ denoting the entry in the ith row and jth column:

$$L_{i,j} = \begin{cases} |\mathcal{N}_i| & \text{if } i = j, \\ -1 & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{if } j \notin \mathcal{N}_i. \end{cases} \tag{1.2}$$

CHAPTER 2

Resilient Consensus Algorithms

2.1 Introduction

Guaranteeing resilience to adversarial misbehavior and misinformation is critically needed in modern autonomous systems. An ever-growing amount of cyber attacks has led to increasing attention on algorithms that guarantee safety and security despite the influence of faults and malicious behavior. Controllers that protect against adversarial actions are especially critical in distributed systems where agents may have limited power, computational capabilities, and knowledge of the system as a whole. In response to this need, the *resilient consensus problem* has been treated in the literature for several decades. In this problem, normally-behaving agents in a multi-agent network seek to come to agreement on one or more state variables in the presence of adversarially-behaving agents whose identity is unknown. Multiple resilient consensus algorithms based on a filtered-mean or median based approach have gained traction recently in the literature as a means to counteract the influence of faulty or adversarial agents. These algorithms include the W-MSR [75], ARC-P [190], SW-MSR [79], DP-MSR [76], QW-MSR [92], LFRE [191], and MCA [192] algorithms, which have all been used for resilient consensus. Several of these algorithms are based upon the *Mean-Subsequence-Reduced* (MSR) family of algorithms [74]. These algorithms guarantee consensus of the normally-behaving agents when the number of adversaries is bounded and the network communication structure satisfies certain robustness properties. The final consensus value of the normal agents is within the convex hull of the normal agents' initial states. However, the exact final value within this convex hull may depend in part upon the behavior of the adversarial agents.

There are several gaps in this body of prior work. First, the vast majority of these papers considers only the problem of resilient *leaderless* consensus. An interesting direction of research is extending the property of resilience to leader-follower consensus scenarios where follower agents track the leader agents' reference state, which may or may not lie within the convex hull of initial agent states, while rejecting the influence of adversarial agents whose identity is unknown.

This task becomes more challenging when the identities of the leader agents are not known to the follower agents, and when leader agents themselves are vulnerable to attacks and may behave adversarially. Only a few recent papers have considered problems related to the resilient leader-follower consensus problem [181, 191, 193, 194]. In [193], the problem of resilient distributed estimation is considered where certain "reliable agents" drive the errors of the remaining normal agents to the static reference value of zero in the presence of misbehaving agents; however this work assumes that reliable nodes are completely immune to adversarial attacks. In [191, 195], the problem of distributed, resilient estimation in the presence of misbehaving nodes is treated. The authors show conditions under which information about the decoupled modes of the system is resiliently transmitted from a group of source nodes to other nodes that cannot observe those modes. This work assumes however that the transmitted information is limited to trajectories of the form $x(k+1) = Ax(k)$ where the matrix A is known to all agents.

The second limitation to prior work is that the vast majority of papers considers only discrete-time systems and algorithms with asymptotic or exponential convergence guarantees. Less attention has been devoted to studying counterparts of these MSR algorithms designed for continuous-time systems that are not discretized [83–86, 88], and few papers consider finite-time consensus in the presence of adversarial agents. One of the difficulties in studying resilient consensus in the continuous-time domain with arbitrarily misbehaving adversaries is the issue of existence and uniqueness of system solutions that describe normal agents' state trajectories. For example, guaranteeing existence and uniqueness of system solutions can become difficult when adversarial signals are discontinuous without a minimum dwell time between discontinuities. In the seminal work [83] the *Adversarial Robust Consensus Protocol* (ARC-P) was presented, where continuous-time single-integrator agents apply a trimmed-mean approach to achieve resilient consensus. These results were extended in [84] to more general LTI agents achieving state synchronization. A limiting assumption made in [83, 84] is that all signals sent from adversarial agents to normal agents are continuous in time. The authors of [83] give reasonable justifications for this assumption, but their results have not yet been extended to more general adversarial signals that may exhibit discontinuities. A few prior works have made the assumption of minimum dwell time between instances where the system dynamics change due to filtering [86, 88]. Nevertheless for many prior control algorithms it is possible to construct adversarial signals that cause infinite switching of system dynamics in a finite amount of time (which is demonstrated in Section 2.5.2 of this chapter). The works [85, 86, 88] do not discuss the possibility of discontinuous adversarial signals or the existence and uniqueness of system solutions. Finite-time convergence for continuous systems with faulty or adversarial is considered in [196, 197]. However, in [196] it is assumed that only the *initial conditions* of certain agents are faulty, with all agents applying the nominally specified control protocol. In contrast, Byzantine adversaries may apply an arbitrary control protocol at any

instant subsequent to the initial time. In addition, [197] considers only undirected graphs, assumes that there exists a safe set of trusted agents that never misbehave, and assumes that all misbehaving agents are only connected to trusted agents.

A third limitation is that, to the best of our knowledge, none of this prior work considers input bounds. Many systems with agents coming to consensus on physical states are subject to bounds on their control inputs.

This chapter presents methods addressing each of these limitations. First, we present conditions under which resilient leader-follower consensus to arbitrary reference values can be achieved in the presence of adversarial agents. The results apply to both static and time-varying graphs, and consider the presence of adversarial leaders in addition to adversarial followers. In addition, this result is shown to have important implications for adversarial attacks which violate the typical assumptions made by prior literature. More specifically, when the F-locality of the adversarial set is violated, we demonstrate sufficient conditions for adversarial agents to drive a consensus network to arbitrary values despite normal followers' application of MSR-type resilient algorithms.

Second, we present a method for achieving finite-time resilient formation control in continuous-time multi-agent systems with bounded inputs using resilient leader-follower consensus methods. More specifically, we introduce a novel continuous finite-time controller that allows agents to achieve formations specified by a set of leaders in the presence of adversarial agents. The controller employs a novel filtering mechanism based on the norm of the difference between agents' states. In addition, we prove that this controller guarantees convergence of the normally-behaving agents to their respective formational positions when control inputs are bounded. We also define novel conditions for the filtering timing and input weights that ensure that agents can remain in formation even with a dwell time in the filtering mechanism. Finally, we also show that the norm-based filtering and bounded input elements of our continuous-time controller can be used in a similar resilient discrete-time system. Normally-behaving agents achieve exponential convergence to their respective formational positions under this discrete-time controller.

Third, we use discontinuous systems theory [198] to relax many of the assumptions of prior literature and consider finite-time resilient *leaderless* consensus. More specifically, we introduce a novel class of controllers that guarantee finite-time consensus for a class of nonlinear systems in the presence of adversarial attacks and faults. We also demonstrate using discontinuous systems theory that our analysis holds even for discontinuous adversarial signals with no minimum dwell time between discontinuities. Finally, we show that our analysis holds for the general F-local adversarial model on digraphs without assuming the presence of any trusted agents.

This chapter is organized as follows: Section 2.2 gives a brief overview of the resilient consensus problem in an MSR algorithm context. Our results on conditions for resilient leader-follower consensus to arbitrary reference values are given in Section 2.3. A method for finite-time resilient

formation control using bounded inputs is given in Section 2.4. Our results for finite-time leaderless consensus in continuous-time systems using discontinuous systems theory are given in Section 2.5. Finally, we give a brief conclusion in Section 2.7.

2.2 Preliminaries on Resilient Consensus and MSR Algorithms

This section will give a brief overview of the resilient leaderless consensus problem and how MSR-type algorithms operate in this context. Consider a system of $n \in \mathbb{Z}_{>0}$ agents. Connections between agents are described by a directed graph (digraph) $\mathcal{D} = (\mathcal{V}, \mathcal{E})$, which for now is assumed to be static in time. Time-varying graphs will be considered in Section 2.3. Each agent $i \in \mathcal{V}$ has a scalar state $x_i \in \mathbb{R}$. Agents update their states according to the dynamics $x_i[t+1] = u_i[t]$. Agents are also able to send information to their out-neighbors $j \in \mathcal{V}_i^{out}$ and receive information from their in-neighbors $j \in \mathcal{V}_i$. We will denote the inclusive set of in-neighbors as $\mathcal{J}_i = \mathcal{V}_i \cup \{i\}$. More specifically, at each timestep t each agent is able to receive the states $x_j^i[t]$, $j \in \mathcal{V}_i$ where $x_j^i[t]$ denotes the state received by agent i from agent j at time t.

For purposes of analysis, agents are classified into two types: *normal* agents (also called *normally-behaving* agents) and *adversarial* agents (also called *adversarially-behaving* agents). Given a nominally specified control law $f[t]$, the set of normal agents includes all agents $i \in \mathcal{V}$ that apply the nominal control law $u_i[t] = f[t]$ for all times $t \geq t_0$. The set of all normal agents is denoted $\mathcal{N} \subset \mathcal{V}$.

On the other hand, the set of adversarial agents includes all agents $j \in \mathcal{V}$ for which there exists a $t' \geq t_0$ such that at least one of the following conditions holds:

1. Agent j does not apply the nominal control law to update its state; i.e., $u_j[t'] \neq f[t']$,
2. Agent j sends a value which is not equal to its actual state to at least one out-neighbor; i.e. there exists $i \in \mathcal{V}_j^{out}$ such that $x_j^i[t'] \neq x_j[t']$,
3. Agent j sends different information to different out-neighbors; i.e. there exist agents $i_1, i_2 \in \mathcal{V}_j^{out}$ such that $x_j^{i_1}[t'] \neq x_j^{i_2}[t']$.

The set of all adversarial agents is denoted $\mathcal{A} \subset \mathcal{V}$. In general, **normal agents are *not* aware of which other agents are normal, and which are adversarial.** On the contrary, adversarial agents may have full knowledge of which other agents are normal and adversarial. Adversarial agents may also collude together and send state values of arbitrary size to their out-neighbors. Prior literature often further classifies adversarial behavior into *malicious* behavior, where adversaries send the same misinformation to all outneighbors, and *Byzantine* behavior, where agents may send different misinformation to different out-neighbors (i.e. condition 3 above). Faulty behavior such as crash

faults can also be modeled under malicious or Byzantine behavior, although faults typically do not exhibit underlying adversarial intent. The cardinality and distribution of the adversarial set is quantified by the notions of F-total and F-local sets:

Definition 2.1 ([2]). *Let $F \in \mathbb{Z}_{\geq 0}$. A set $S \subset \mathcal{V}$ is* F-total *if it contains at most F nodes; i.e. $|S| \leq F$.*

Definition 2.2 ([2]). *Let $F \in \mathbb{Z}_{\geq 0}$. A set $S \subset \mathcal{V}$ is* F-local *with respect to (w.r.t.) a given $t_0 \in \mathbb{Z}$ if $|S \cap \mathcal{V}_i[t]| \leq F \; \forall i \in \mathcal{V} \backslash S, \forall t \geq t_0$.*

The objective of the resilient leaderless consensus problem is to determine a nominal control law that guarantees that the states of normally-behaving agents converge to a common value in the presence of either an F-local or F-total adversarial set $\mathcal{A}$. More precisely, here we consider *asymptotic* consensus rather than exact consensus, where the objective is for all of the following conditions to be satisfied:

1. *Agreement:* There exists $L \in \mathbb{R}$ such that $\lim_{t \to \infty} x_i[t] = L$ for all $i \in \mathcal{N}$, and
2. *Safety:* Each normal state $i \in \mathcal{N}$ satisfies $x_i[t] \in [m[0], M[0]]$ for all $t \geq t_0$ where $m[t] = \min_{i \in \mathcal{N}} x_i[t]$, $M[t] = \max_{i \in \mathcal{N}}[t]$.

When combined together, the conditions of agreement and safety form the condition of *Validity*, where the final consensus value lies within the convex hull of initial normal agents' states [75].

Prior literature is replete with examples of consensus laws achieving asymptotic consensus in the absence of adversarial agents. A representative example is the control law

$$u_i[t] = \sum_{j \in \mathcal{V}} w_{ij}[t] x_j^i[t], \tag{2.1}$$

where the (possibly time-varying) weights $w_{ij} : \mathbb{R} \to \mathbb{R}$ satisfy all of the following conditions:

- $w_{ij}[t] = 0$ if $j \notin \mathcal{J}_i$,
- $w_i j[t] \geq \alpha > 0 \; \forall t \geq t_0$ for some $\alpha \in \mathbb{R}_{>0}$,
- $\sum_{j \in \mathcal{V}} w_{ij}[t] = 1$.

However, it is well-known that the presence of even one adversarial agent can result in agents applying (2.1) to be led off to arbitrary values. The presence of multiple adversarial agents can prevent consensus entirely.

MSR algorithms were proposed as a method for achieving resilient asymptotic consensus in a computationally lightweight, distributed manner. A seminal example of an MSR algorithm is

Algorithm 2.1 W-MSR ALGORITHM [75]:

1. At each time step t, each agent i forms a sorted list $\Omega_i[t]$ of the values received from its in-neighbors as follows:

$$\Omega_i[t] = \{x_j^i[t] : j \in \mathcal{J}_i\}, \tag{2.2}$$

2. If there are less than F values strictly greater than $x_i[t]$ in $\Omega_i[t]$, then agent i removes all values strictly greater than $x_i[t]$ from $\Omega_i[t]$. Otherwise, agent i removes the F largest values from $\Omega_i[t]$.

3. *In addition*, if there are less than F values strictly less than $x_i[t]$ in $\Omega_i[t]$, then agent i removes all values strictly less than $x_i[t]$ from $\Omega_i[t]$. Otherwise, agent i removes the F smallest values from $\Omega_i[t]$.

4. Let $\mathcal{R}_i[t]$ denote the set of all agent indices whose state values were removed from $\Omega_i[t]$ in steps 2) and 3). Each normal agent i applies the update

$$x_i[t+1] = u_i[t] \tag{2.3}$$

$$u_i[t] = \sum_{j \in \mathcal{J}_i^{T'}[t] \setminus \mathcal{R}_i[t]} w_{ij}[t] x_j^i[t] \tag{2.4}$$

where $\forall t \geq t_0$ and $\forall i \in S_f$ the weights satisfy $w_{ij}[t] \geq \alpha > 0 \; \forall j \in \mathcal{J}_i[t]$, and $\sum_{j \in \mathcal{J}_i[t] \setminus \mathcal{R}_i[t]} w_{ij}[t] = 1$.

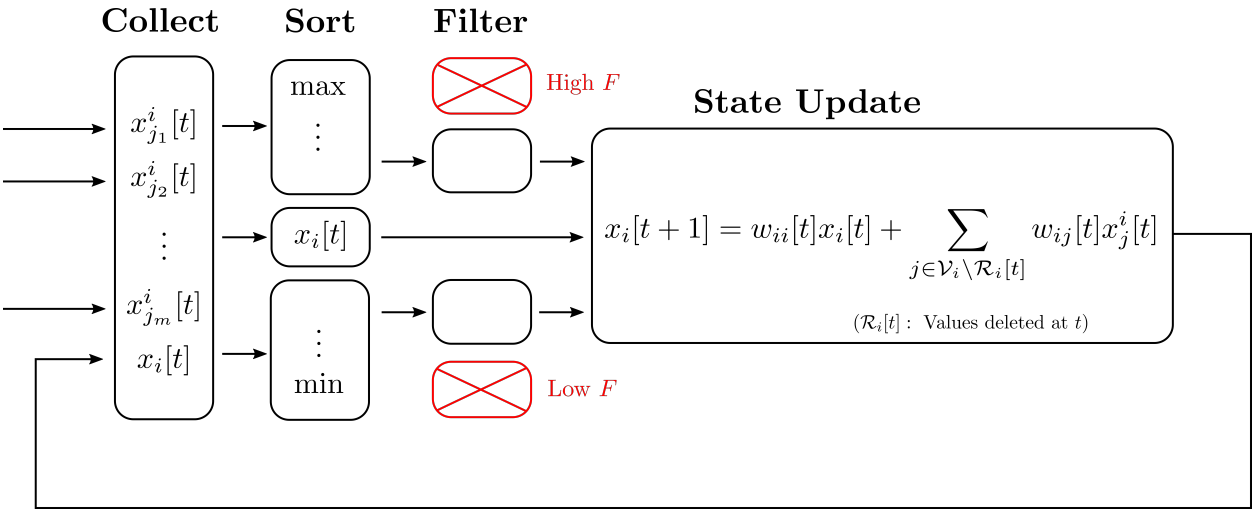


Figure 2.1: A pictorial representation of how the W-MSR algorithm operates (Algorithm 2.1).

the Weighted Mean-Subsequence-Reduced (W-MSR) algorithm which defines the normal agent update law as per Algorithm 2.1.

In short, rather than taking a convex combination of *all* received values, each agent ignores the F highest values above its own state and the F lowest values below its own state, and then takes the convex combination of the remaining values. Here, F is an integer parameter that all normal agents are assumed to have knowledge of. Note that when $F = 0$ no filtering takes place and the original consensus algorithm 2.1 is recovered.

It is straightforward to verify that when the adversarial set is either F-local or F-total, the safety condition $x_i[t] \in [m[0], M[0]]$ for all $t \geq 0$, $i \in \mathcal{N}$ is satisfied. To give a rough overview, this is because there will be at most F adversarial values $x_j^i[t]$, $j \in \mathcal{A}$ either above or below the value $x_i[t]$. These values will therefore be filtered out as per Algorithm 2.1. Any unfiltered values will be from normally-behaving agents, and therefore will lie within the convex hull of normal agents' initial states.

However, the conditions under which all normal agents converge to an agreement value $L \in [m[0], M[0]]$ are not as straightforward. The conditions under which the W-MSR algorithm guarantees convergence to agreement are based on the graph-theoretic notion known as r-robustness [2]:

Definition 2.3. *Let $r \in \mathbb{Z}_{\geq 0}$ and $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be a digraph. A nonempty subset $S \subset \mathcal{V}$ is r-reachable if $\exists i \in S$ such that $|\mathcal{V}_i \backslash S| \geq r$.*

Definition 2.4. *Let $r \in \mathbb{Z}_{\geq 0}$. A nonempty, nontrivial digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ on n nodes $(n \geq 2)$ is r-robust if for every pair of nonempty, disjoint subsets of $\mathcal{V}$, at least one of the subsets is r-reachable. By convention, the empty graph $(n = 0)$ is 0-robust and the trivial graph $(n = 1)$ is 1-robust.*

The problem of determining whether a given graph is r-robust for a particular integer r will be discussed in Chapter 3 of this dissertation. To give a brief example, it is an established fact that all complete graphs with $n \in \mathbb{Z}_{>0}$ agents are $\lceil n/2 \rceil$-robust.

If the digraph $\mathcal{D}$ is $(2F+1)$-robust and normally-behaving agents apply the W-MSR algorithm, resilient asymptotic consensus is guaranteed (i.e. consensus, safety, and validity are achieved). Full details and proofs of this fact can be found in [2]. A plot of normally-behaving agents achieving consensus using the W-MSR algorithm is shown in Figure 2.2. This simulation shows 20 agents running the normal W-MSR protocol in a digraph that is 8-robust under a 3-total adversarial model, with each agent having the parameter $F = 3$ in the W-MSR algorithm. The three agents behave in a malicious manner by sending the same misinformation to all their out-neighbors; the evolution of each adversarial agent's misinformation is indicated by the dotted red lines. The normal agents come to consensus to a value within the convex hull of their initial states despite having no knowledge of whether their in-neighbors are adversarial or not.

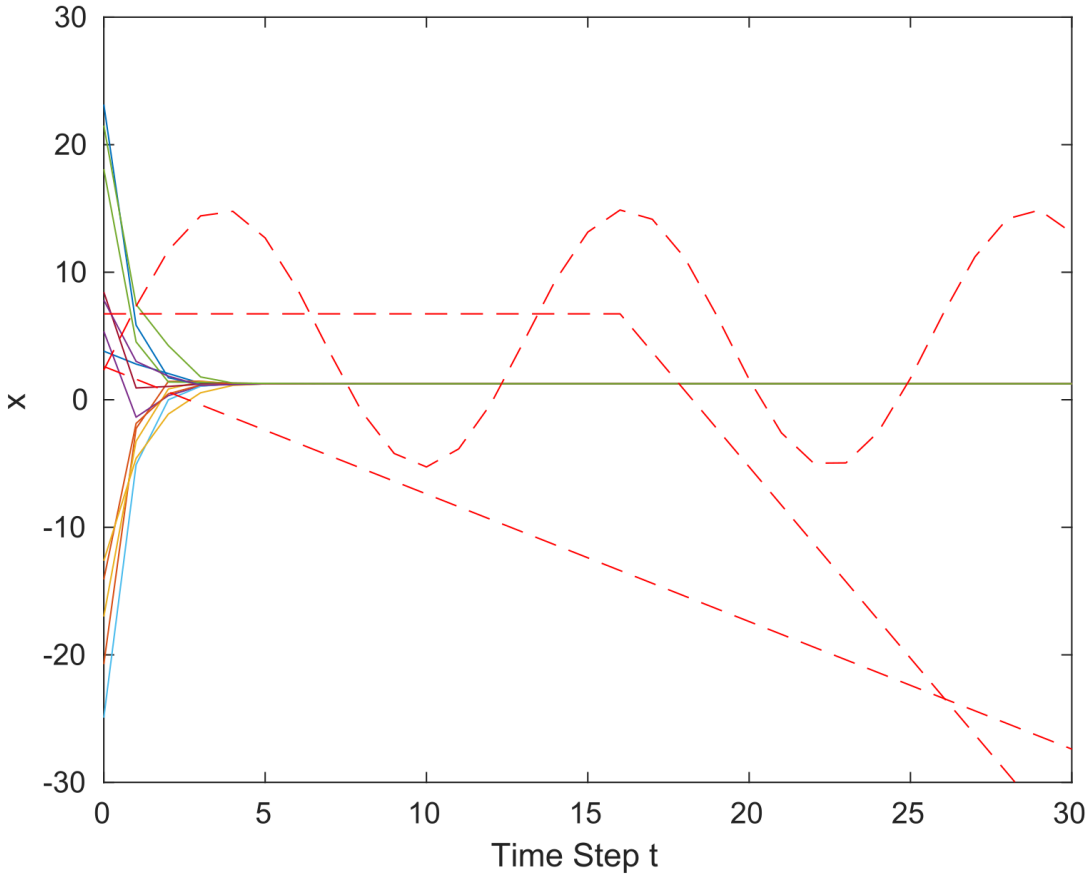


Figure 2.2: A network of agents running the normal W-MSR algorithm with $n = 20,\ k = 15$. The dotted red lines represent adversarial agents, while the solid lines represent normally behaving agents.

In addition to r-robustness, other graph-theoretic notions of robustness have been considered in prior literature. One that will be used in this chapter is *strong r-robustness w.r.t. a set $S \subset \mathcal{V}$*:

Definition 2.5 (Strong r-robustness w.r.t. S [191])**.** *Let $r \in \mathbb{Z}_{\geq 0}$, $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be a digraph, and $S \subset \mathcal{V}$ be a nonempty subset. $\mathcal{D}$ is strongly r-robust w.r.t. S if for any nonempty subset $C \subseteq \mathcal{V} \backslash S$, C is r-reachable.*

Whereas r-robustness quantifies the level of redundancy of information connections between any two nonempty, disjoint subsets of agents in a graph, strong r-robustness w.r.t. a set S quantifies the redundancy of information connections flowing from the set S down to other nonempty subsets of the remaining agents in the graph. As will be shown in the next section, this will have implications for the resilient leader-follower consensus problem when S is taken to be the set of leader agents. Unlike r-robustness, given a particular subset $S \subset \mathcal{V}$ it can be verified in polynomial time whether $\mathcal{D}$ is strongly robust w.r.t. S [194].

As a final note, the reader may perhaps wonder why a trimmed mean is used for the state update step in MSR algorithms rather than the median, which is more resistant to the effects of outliers than the mean. In complete graphs where agents have access to *global* information, it would indeed be more appropriate to use the median of received values rather than a trimmed mean. However in non-complete graphs where agents only have access to *local* information received from their in-neighbors, each agent's local median will in general not be equal to other agents' local medians or the global median. Ensuring that these local medians converge to a common value requires much stronger conditions on the graph structure and the number of edges within the network for a given adversarial distribution, as outlined in [192].[1] Using a trimmed mean approach allows for more relaxed graph-theoretic requirements on the network while still exhibiting resilience to adversarial misinformation.

2.3 Resilient Leader-Follower Consensus to Arbitrary Reference Values in Time-Varying Graphs

The previously mentioned results from prior literature do not address the resilient *leader-follower* consensus problem, which will be described more precisely below. Leader-follower consensus is employed in several practical scenarios including multi-agent rendezvous, trajectory tracking, and others. The chief difference between the leaderless and leader-follower consensus problems is that the objective of leader-follower consensus is for agents to converge to a particular *reference value* propagated by the leader or leaders. This reference value may change and take on values lying

[1] The graph-theoretic requirement for a median-based approach is called *r-excess robustness* and is treated in [192].

outside the convex hull of initial agents' states. In addition, the possibility of leaders themselves becoming adversarial must be considered. Agents must be able to converge to the desired reference value without necessarily having knowledge of which other leaders or followers are normal or adversarial.

There is a second important reason for studying the resilient leader-follower consensus problem. Given a network of agents applying an MSR-type algorithm with parameter $F \in \mathbb{Z}_{\geq 0}$, no prior literature asks the question *What happens if the F-local condition is violated?* For example, what happens when agents apply the W-MSR algorithm with parameter F, but the adversary model is $F+1$-total or -local? No prior work attempts to address whether or not the W-MSR algorithm exhibits graceful degradation when the F-local assumption is violated. Our results in this section present a sufficient condition for a set of adversarial agents to drive an entire network of agents applying an MSR-type algorithm to arbitrary values. This result serves as a caution for those seeking to design networks applying MSR-type algorithms for resilience.

2.3.1 Problem Formulation

Consider a digraph of n agents with time-varying edges, denoted $\mathcal{D}[t] = (\mathcal{V}, \mathcal{E}[t])$. Each agent $i \in \mathcal{V}$ has a state $x_i[t] \in \mathbb{R}$. Two types of agents are considered: leader agents (also called "source" agents) and follower agents. The set of leader agents consists of agents that propagate a desired reference signal to the set of follower agents.

Definition 2.6. *The set of leader agents is denoted $\mathcal{L} \subset \mathcal{V}$. The set of follower agents is denoted $S_f = \mathcal{V} \backslash \mathcal{L}$.*

Assumption 2.1. *The sets $\mathcal{L}$ and S_f are static and satisfy $\mathcal{L} \cup S_f = \mathcal{V}$ and $\mathcal{L} \cap S_f = \emptyset$.*

Each normally-behaving leader agent l updates its state according to a reference function $f_r : \mathbb{R} \to \mathbb{R}$ as follows:

$$x_l[t+1] = f_r[t]. \tag{2.5}$$

The precise definition of *normally-behaving* will be given in Definition 2.9.

The purpose of this section is to determine conditions under which normally-behaving follower agents resiliently achieve consensus with a static reference state of the set of normally-behaving leader agents in the presence of a possibly nonempty set of adversarial agents, where the precise definition of adversarial agents will be given in Definition 2.15.

Problem 2.1. *Given a digraph $\mathcal{D}[t] = (\mathcal{V}, \mathcal{E}[t])$ with a time-varying edge set satisfying Assumption 2.1, determine conditions under which $\lim_{t\to\infty} \max_{i,\, l} |x_i[t] - x_l[t]| = 0$ for all normally-behaving*

follower agents i and for all normally-behaving leaders l in the presence of a possibly nonempty adversarial subset of agents $\mathcal{A} \subset \mathcal{V}$.

Each normally-behaving leader agent is able to send its state value to its out-neighbors at each time t. In addition, each normally-behaving follower agent $i \in S_f$ can receive state values from its in-neighbors at each time t, and can also send its own state value to its out-neighbors at each time t.

Definition 2.7. *The value received by agent i from agent j at time t is denoted $x_j^i[t]$.*

Since the set of edges $\mathcal{E}[t]$ is time-varying, agents use a sliding-window approach over a time period $T \in \mathbb{Z}_{\geq 0}$ when taking into account information received from their in-neighbors. Let $T' = \min(T, t - t_0),\ t \geq t_0$. At each time $t \geq t_0$, each normally-behaving follower agent i considers information received from the set

$$\mathcal{J}_i^T[t] = \bigcup_{\tau=0}^{T'} \mathcal{J}_i[t - \tau], \tag{2.6}$$

i.e. the union of i's in-neighbor sets over the time interval $[t - T, t]$ if $t \geq t_0 + T$, or $[t_0, t]$ if $t < t_0 + T$.

Each normally-behaving follower agent i updates its state according to the *Sliding Weighted Mean-Subsequence-Reduced* (SW-MSR) algorithm [79], which is outlined in Algorithm 2.2. A pictorial description of the SW-MSR Algorithm is given in Figure 2.3. In essence, the SW-MSR algorithm is a generalization of the W-MSR algorithm where normally-behaving follower agents update their state based on the most recently received information from each in-neighbor in $\mathcal{J}_i^T[t]$ over a sliding time window of length T. If $T = 0$, the SW-MSR algorithm essentially reduces to the *Weighted Mean-Subsequence-Reduced* (W-MSR) algorithm [2].

In context of the leader-follower consensus problem considered in this section, the definition of adversarial agents is as follows:

Definition 2.8. *An agent $j \in \mathcal{V}$ is* adversarial *if at least one of the following conditions hold:*

1. *There exists a time t where agent j does not update its state according to either the leader update law (2.5) or the follower update law (2.8).*

2. *There exists a time t where j does not communicate its true state value $x_j(t)$ to at least one of its out-neighbors; i.e. $\exists t \geq t_0$ and $\exists k \in V_j^{out}[t]$ s.t. $x_j[t] \neq x_j^k[t]$.*

[2]Observe that by the definition of $\mathcal{J}_i[t]$, $x_i^i[\tau_{ii}[t]] \in \Omega_i[t]$ for all $t \geq t_0$. This implies that the set $\Omega_i[t]$ is never empty at any time, even for $t_0 \leq t < t + T$.

Algorithm 2.2 SW-MSR ALGORITHM [79]:

1. At each time step t, each agent i forms a sorted list $\Omega_i[t]$ of the most recently received values from its in-neighbors as follows:

$$\begin{aligned} \tau_{ij}[t] &= \max(\{\tau \in [t-T',t] : j \in \mathcal{J}_i[\tau]\}), \forall j \in \mathcal{J}_i^{T'}[t] \\ \Omega_i[t] &= \{x_j^i[\tau_{ij}[t]] : j \in \mathcal{J}_i^{T'}[t]\}, \end{aligned} \tag{2.7}$$

   with $T' = \min(T, t - t_0)$ and $\mathcal{J}_i^T[t]$ defined in (2.6).[2]

2. If there are less than F values strictly greater than $x_i[t]$ in $\Omega_i[t]$, then agent i removes all values strictly greater than $x_i[t]$ from $\Omega_i[t]$. Otherwise, agent i removes the F largest values from $\Omega_i[t]$.

3. *In addition*, if there are less than F values strictly less than $x_i[t]$ in $\Omega_i[t]$, then agent i removes all values strictly less than $x_i[t]$ from $\Omega_i[t]$. Otherwise, agent i removes the F smallest values from $\Omega_i[t]$.

4. Let $\mathcal{R}_i[t]$ denote the set of all agent indices whose state values were removed from $\Omega_i[t]$ in steps 2) and 3). Each normal agent i applies the update

$$x_i[t+1] = u_i[t] \tag{2.8}$$

$$u_i[t] = \sum_{j \in \mathcal{J}_i^{T'}[t] \setminus \mathcal{R}_i[t]} w_{ij}[t] x_j^i[\tau_{ij}[t]] \tag{2.9}$$

   where $\forall t$ and $\forall i \in S_f$ the weights satisfy $w_{ij}[t] \geq \alpha > 0$ $\forall j \in \mathcal{J}_i^{T'}[t]$, and $\sum_{j \in \mathcal{J}_i^{T'}[t] \setminus \mathcal{R}_i[t]} w_{ij}[t] = 1$.

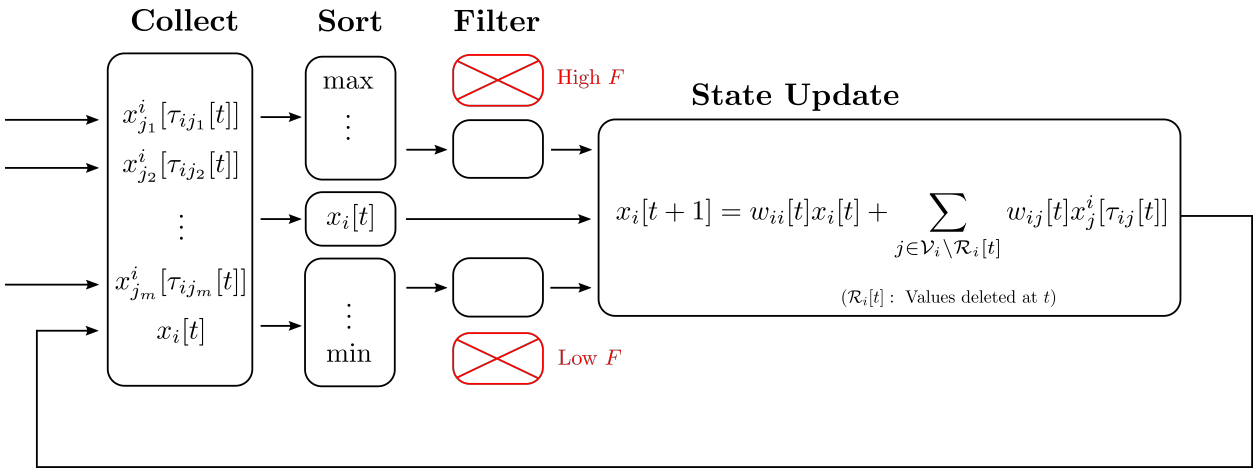


Figure 2.3: A pictorial representation of the Sliding Window Mean-Subsequence-Reduced (SW-MSR) algorithm (Algorithm 2.2). The chief difference between the SW-MSR and W-MSR algorithms are that the SW-MSR considers the most recently received information over a sliding time window to mitigate the effects of the network graph being time-varying.

3. *There exists a time t where j communicates different values to different out-neighbors; i.e. $\exists t \geq t_0$ and $\exists k_1, k_2 \in V_j^{out}[t]$ s.t. $x_j^{k_1}[t] \neq x_j^{k_2}[t]$.*

The set of adversarial agents is denoted $\mathcal{A} \subset \mathcal{V}$.

Definition 2.9. *The set of agents that are* not *adversarial are denoted $\mathcal{N} = \mathcal{V} \backslash \mathcal{A}$. Agents in $\mathcal{N}$ are referred to as* normally-behaving agents.

Again, adversarial agents are agents that update their states arbitrarily or communicate false information to their out-neighbors. By Definition 2.8, the set of adversarial agents $\mathcal{A}$ includes both malicious agents and Byzantine agents [2].

This section considers scenarios where both followers *and leaders* are vulnerable to adversarial attacks and faults, and therefore the set $\mathcal{A} \cap \mathcal{L}$ may possibly be nonempty, and the set $\mathcal{A} \cap S_f$ may possibly be nonempty. Observe that the sets $\mathcal{L}$ and S_f are disjoint, and the sets $\mathcal{N}$ and $\mathcal{A}$ are disjoint, but the intersections $\mathcal{L} \cap \mathcal{N}$ and $S_f \cap \mathcal{N}$ represent the normally-behaving leaders and followers, respectively, and the intersections $\mathcal{L} \cap \mathcal{A}$ and $S_f \cap \mathcal{A}$ represent the adversarially-behaving leaders and followers, respectively. The following notation will be used:

Definition 2.10 (Adversarially-behaving agent notation)**.** *The set of adversarial leaders is denoted as $\mathcal{L}^{\mathcal{A}} = \mathcal{L} \cap \mathcal{A}$. The set of adversarial followers is denoted as $S_f^{\mathcal{A}} = S_f \cap \mathcal{A}$.*

Definition 2.11 (Normally-behaving agent notation)**.** *The set of normally-behaving leaders is denoted $\mathcal{L}^{\mathcal{N}} = \mathcal{L} \backslash \mathcal{A}$. The set of normally-behaving followers is denoted $S_f^{\mathcal{N}} = S_f \backslash \mathcal{A}$.*

To achieve resilient leader-follower consensus, the graph-theoretic structure of the network will be required to satisfy certain conditions. For example, under an F-local or F-total adversary model it must be ensured that adversarial agents do not cut off the flow of information from the normally-behaving leaders to the normally-behaving followers. Towards this end we introduce the concept of *strong* (T, t_0, r)*-robustness*. Recall that the definition of strong r-robustness is given in Definition 2.5.

Definition 2.12. *Let $T, r \in \mathbb{Z}_{\geq 0}$ and let $t_0 \in \mathbb{Z}$. Let $\mathcal{D}[t] = (\mathcal{V}, \mathcal{E}[t])$ be a digraph with a time-varying edge set, and define $\mathcal{D}^T[t] = \bigcup_{\tau=0}^{T} \mathcal{D}[t-\tau]$. Then $\mathcal{D}[t]$ is strongly (T, t_0, r)-robust with respect to a subset $S \subset \mathcal{V}$ if $\mathcal{D}^T[t]$ is strongly r-robust with respect to $S \subset \mathcal{V}$ for all $t \geq t_0 + T$.*

Strong (T, t_0, r)-robustness generalizes the notion of strong r-robustness to digraphs with a time-varying edge set. Note that the property of strong r-robustness in Definition 3.10 is a particular case of strong (T, t_0, r)-robustness with $T = 0$. In many time-varying networks it may be difficult to ensure that a digraph $\mathcal{D}[t]$ is strongly r-robust w.r.t. S at every time step t. The time window T relaxes this requirement by only requiring the union of $\mathcal{D}[t]$ over the last T timesteps to be strongly r-robust w.r.t. S. Increasing T allows for edges to be "active" less often while still preserving the (T, t_0, r)-robustness of $\mathcal{D}[t]$.

Notice that for $r = 2F + 1$, a digraph which is strongly (T, t_0, r)-robust w.r.t. the leader set $\mathcal{L}$ must have at least $2F+1$ leaders. This implies that at least $F+1$ normally-behaving leaders will be present at all times under either an F-total or F-local adversarial model. The proof of Theorem 2.1 will demonstrate that the structure of strong $(T, t_0, 2F + 1)$-robust graphs w.r.t. $\mathcal{L}$ will guarantee that information from the normally-behaving leaders will reach all normally-behaving followers.[3]

2.3.2 Resilient Leader-Follower Consensus in Time-Varying Graphs

For our analysis of time-varying graphs, the following functions are defined for purposes of showing convergence of normally-behaving followers to normally-behaving leaders' reference value. Recall that $T' = \min(T, t - t_0)$ as per Algorithm 2.2.

$$\begin{aligned} \overline{M}[t] &= \max_{i \in S_f^{\mathcal{N}}, l \in \mathcal{L}^{\mathcal{N}}, \tau \in [0, T']} (x_i[t-\tau], x_l[t-\tau]) \\ \overline{m}[t] &= \min_{i \in S_f^{\mathcal{N}}, l \in \mathcal{L}^{\mathcal{N}}, \tau \in [0, T']} (x_i[t-\tau], x_l[t-\tau]) \\ V[t] &= \overline{M}[t] - \overline{m}[t] \end{aligned} \tag{2.10}$$

[3] To clarify, not all normally-behaving followers will necessarily have leaders as direct in-neighbors. But, roughly speaking, there will be enough redundant information paths from the normal leader set to each normal follower such that all normal followers will converge to the desired reference value.

The following Lemma establishes that $\overline{M}[t]$ and $\overline{m}[t]$ are nonincreasing and nondecreasing functions, respectively, on any time interval where $f_r[t]$ is constant.

Lemma 2.1. *Let $\mathcal{D}[t] = (\mathcal{V}, \mathcal{E}[t])$ be a nonempty, nontrivial, simple digraph with S_f nonempty. Let $F, \tau \in \mathbb{Z}_{\geq 0}$, $t_0, t_1, t_2 \in \mathbb{Z}$ with $t_2 > t_1 \geq t_0$. Suppose that $\mathcal{A}$ is an F-local set with respect to t_0, and suppose that all normally-behaving agents $i \in S_f^{\mathcal{N}}$ apply the SW-MSR algorithm with parameter F. If $f_r[t]$ is constant $\forall t \in [t_1, t_2)$, then all of the following statements hold $\forall t \in [t_1, t_2)$:*

- $x_i[t] \in [\overline{m}[t_1], \overline{M}[t_1]] \; \forall i \in S_f^{\mathcal{N}}$
- *$\overline{M}[t]$ and $\overline{m}[t]$ are nonincreasing and nondecreasing, respectively.*

Proof. First, observe that $x_l[t] = f_r[t] \; \forall l \in \mathcal{L}^{\mathcal{N}}, \forall t \geq t_0$ by (2.5). Since $f_r[t]$ is constant $\forall t \in [t_1, t_2)$, by (2.10) we have $x_l[t] \in [\overline{m}[t_1], \overline{M}[t_1]] \; \forall l \in \mathcal{L}^{\mathcal{N}}, \forall t \in [t_1, t_2)$. Next, consider any $i \in S_f^{\mathcal{N}}$. By definition of $\overline{M}[t]$ and $\overline{m}[t]$, $\forall j \in \mathcal{J}_i^T[t_1] \backslash \mathcal{A}$, $x_j^i[\tau_{ij}[t_1]] \in [\overline{m}[t_1], \overline{M}[t_1]]$ where $\tau_{ij}[t]$ is defined by (2.7). Now consider any agent $k \in \mathcal{A}$. If we have $x_k^i[\tau_{ik}[t_1]] > \overline{M}[t_1] \geq x_j(\tau_{ij}[t_1]) \; \forall j \in \mathcal{V} \backslash \mathcal{A}$, the fact that $|\mathcal{A}| \leq F$ implies any value $x_k^i[\tau_{ik}[t_1]]$ satisfying this condition is one of the F highest values in $\Omega_i[t_1]$ and will be filtered out as per the SW-MSR Algorithm (Algorithm 2.2). Similarly, if $x_k^i[\tau_{ik}[t_1]] < \overline{m}[t_1] \leq x_j(\tau_{ij}[t_1]) \; \forall j \in \mathcal{V} \backslash \mathcal{A}$, then $x_k^i[\tau_{ik}[t_1]]$ is one of the F lowest values in $\Omega_i[t_1]$ and will be filtered out. Therefore all state values in $\mathcal{J}_i^T[t_1] \backslash \mathcal{R}_i[t_1]$ fall in the interval $[\overline{m}[t_1], \overline{M}[t_1]] \; \forall i \in S_f^{\mathcal{N}}$. Since the values of $w_{ij}[t_1]$ imply a convex combination of values in the set $\mathcal{J}_i[^T[t_1] \backslash \mathcal{R}_i[t_1]$, $x_i[t_1 + 1] \in [\overline{m}[t_1], \overline{M}[t_1]]$. Further, since by definition of $\overline{m}$ and $\overline{M}$ we have $x_i[t - \tau] \in [\overline{m}[t_1], \overline{M}[t_1]] \; \forall i \in S_f^{\mathcal{N}}, \; \forall l \in \mathcal{L}^{\mathcal{N}}, \forall \tau \in [0, T']$ where $T' = \min(T, t - t_0)$, it holds that $x_i[t_1 + 1 - \tau] \in [\overline{m}[t_1], \overline{M}[t_1]] \; \forall \tau \in [0, T'], \; \forall i \in S_f^{\mathcal{N}}, \; \forall l \in \mathcal{L}$. These arguments imply $\overline{M}[t_1 + 1] \leq \overline{M}[t_1]$. Similar arguments can be used to show $\overline{m}[t_1 + 1] \geq \overline{m}[t_1]$.

Now by induction assume $\overline{M}[t_1 + p] \leq \overline{M}[t_1 + p - 1]$ and $\overline{m}[t_1 + p] \geq \overline{m}[t_1 + p - 1]$, for all $p \in \mathbb{Z}_{\geq 0}$ such that $p \geq 1$, $t_1 + p < t_2 - 1$. By (2.10), $x_i[t] \in [\overline{m}[t_1 + p], \overline{M}[t_1 + p]] \; \forall i \in S_f^{\mathcal{N}}, \forall t \in [t_1 + p - T, t_1 + p]$. In addition, $f_r[t]$ being constant on $[t_1, t_2)$ implies $x_l[t] \in [\overline{m}[t_1 + p], \overline{M}[t_1 + p]] \; \forall l \in \mathcal{L}^{\mathcal{N}}$. Therefore $x_j^i[\tau_{ij}[t_1 + p]] \in [\overline{m}[t_1], \overline{M}[t_1]] \; \forall j \in \mathcal{J}_i^T[t_1 + p] \backslash \mathcal{A}$. Since $|\mathcal{A}| \leq F$, it can be shown by prior arguments that $x_j^i[\tau_{ij}[t_1 + p]]$ for all $j \in \mathcal{J}_i^T[t_1 + p] \backslash \mathcal{R}_i[t_1 + p]$ will lie in the interval $[\overline{m}[t_1 + p], \overline{M}[t_1 + p]] \; \forall i \in S_f^{\mathcal{N}}$. Therefore all $i \in S_f^{\mathcal{N}}$ will update their states with a convex combination of values in $[\overline{m}[t_1 + p], \overline{M}[t_1 + p]]$, implying $\overline{m}[t_1 + p + 1] \geq \overline{m}[t_1 + p]$ and $\overline{M}[t_1 + p + 1] \leq \overline{M}[t_1 + p]$. □

The next theorem presents the first main result of this section. It demonstrates that the error between the normal agents and normally-behaving leaders decreases exponentially on any time interval $t \in [t_1, t_2)$ where $f_r[t]$ is constant and $t_2 - t_1$ is sufficiently large.

Theorem 2.1. *Let $\mathcal{D}[t] = (\mathcal{V}, \mathcal{E}[t])$ be a nonempty, nontrivial, simple digraph. Let $\mathcal{L}, S_f, S_f^{\mathcal{N}}, \mathcal{A}$ be defined as per Definitions 2.6 and 2.8. Let $F \in \mathbb{Z}_{\geq 0}$, $t_0, t_1, t_2 \in \mathbb{Z}$ with $t_2 > t_1 \geq t_0 + T$, and let $V[t]$ be defined as in (2.10). Suppose that S_f is nonempty, $\mathcal{A}$ is an F-local set with respect to t_0, $\mathcal{D}[t]$ is strongly $(T, t_0, 2F+1)$-robust w.r.t. the set $\mathcal{L}$, and all normally-behaving agents $i \in S_f^{\mathcal{N}}$ apply the SW-MSR algorithm with parameter F. If $f_r[t]$ is constant $\forall t \in [t_1 - T, t_2)$ and $t_2 > t_1 + (|S_f^{\mathcal{N}}| + 1)\sigma T$ for some $\sigma \in \mathbb{Z}_{\geq 0}$, then*

$$V[t_1 + (|S_f^{\mathcal{N}}| + 1)\sigma T] \leq (1 - \alpha^{(|S_f^{\mathcal{N}}|+1)T})^{\sigma} V[t_1 + T],$$

where $0 < \alpha < 1$ is defined in Algorithm 2.2. Furthermore, if $t_2 = \infty$,

$$\lim_{t\to\infty} V[t] = \lim_{t\to\infty} \max_{i \in S_f^{\mathcal{N}},\, l \in \mathcal{L}^{\mathcal{N}}} |x_i[t] - x_l[t]| = 0.$$

Proof. Consider the case where $f_r[t]$ is constant for $t \in [t_1 - T, t_2)$ and $t_2 < \infty$. This implies $x_l[t] = f_r[t]$ is constant $\forall l \in \mathcal{L}^{\mathcal{N}}$, $\forall t \in [t_1 - T, t_2)$. We define

$$\begin{aligned}
X_m(t, t', \underline{\epsilon}) &= \{i \in \mathcal{N} : x_i[t' - \tau] < \overline{m}[t] + \underline{\epsilon} \text{ for some } 0 \leq \tau \leq T,\ \tau \in \mathbb{Z}\},\\
X_M(t, t', \overline{\epsilon}) &= \{i \in \mathcal{N} : x_i[t' - \tau] > \overline{M}[t] - \overline{\epsilon} \text{ for some } 0 \leq \tau \leq T,\ \tau \in \mathbb{Z}\},\\
S_X(t, t', \underline{\epsilon}, \overline{\epsilon}) &= X_m(t, t', \underline{\epsilon}) \cup X_M(t, t', \overline{\epsilon}),\\
\overline{S}_X(t, t', \underline{\epsilon}, \overline{\epsilon}) &= \mathcal{V} \backslash S_X(t, t', \underline{\epsilon}, \overline{\epsilon}).
\end{aligned}$$

We prove the result by first showing that $|S_X(t, t', \underline{\epsilon}, \overline{\epsilon})|$ decreases over an appropriate sequence of t' and with an appropriate choice of $\underline{\epsilon}, \overline{\epsilon}$. Let $\underline{\epsilon} = f_r[t_1] - \overline{m}[t_1]$ and $\overline{\epsilon} = \overline{M}[t_1] - f_r[t_1]$. $\mathcal{D}[t]$ is strongly $(T, t_0, 2F+1)$-robust w.r.t $\mathcal{L}$, implying $\mathcal{D}^T[t]$ is strongly $(2F+1)$-robust w.r.t $\mathcal{L}$ $\forall t \geq t_0 + T$. $\mathcal{D}^T[t]$ being strongly $(2F+1)$-robust with respect to $\mathcal{L}$ implies there exists a nonempty $S_1 \subseteq S_f^{\mathcal{N}} \subset \mathcal{V} \backslash \mathcal{L}$ such that $\forall i_1 \in S_1$, $|\mathcal{J}_{i_1}^T[t_1] \backslash S_f^{\mathcal{N}}| \geq 2F + 1$. Since $\mathcal{A}$ is F-local and $\mathcal{V} \backslash S_f^{\mathcal{N}} = \mathcal{L} \cup \mathcal{A}$, this implies $|\mathcal{J}_{i_1}^T[t_1] \cap \mathcal{L}^{\mathcal{N}}| \geq F + 1$. This implies by the SW-MSR Algorithm, $\mathcal{J}_{i_1}^T[t_1] \backslash \mathcal{R}_{i_1}[t_1]$ contains at least one normally-behaving leader $l \in \mathcal{L}^{\mathcal{N}}$ with $x_l^i[\tau_{i_1 l}[t_1]] = f_r[t_1]$. This can be seen by noting that $x_l[t] = f_r[t]$ $\forall l \in \mathcal{L}^{\mathcal{N}}$, $\forall t \in [t_1 - T, t_2)$. Using this fact, lower bounds on $x_{i_1}[t]$ for all $i_1 \in S_1$ and $t \in [t_1 + 1, t_1 + T]$ can be established as follows: recall that the weights w_{ij} are lower bounded by $\alpha > 0$. By Lemma 2.1, $x_j^{i_1}[t] \in [\overline{m}[t_1], \overline{M}[t_1]]$ $\forall j \in \mathcal{J}_{i_1}^T[t] \backslash \mathcal{R}_{i_1}[t]$, $\forall t \in [t_1 - T, t_2)$. Observe that

$$\begin{aligned}
x_{i_1}[t_1 + 1] &= \sum_{j \in \mathcal{J}_{i_1}^T[t] \backslash \mathcal{R}_{i_1}[t]} w_{i_1 j}[t] x_j^{i_1}[\tau_{i_1 j}[t]], && (2.11)\\
&\geq \alpha f_r[t_1] + (1 - \alpha)\overline{m}[t_1]. && (2.12)
\end{aligned}$$

Since there exists at least one normally-behaving leader in $\mathcal{J}_{i_1}^T[t_1] \backslash \mathcal{R}_{i_1}[t_1]$, (2.12) represents the minimum possible value for $x_{i_1}[t_1+1]$. Extending these bounds to time t_1+T yields

$$\begin{aligned}
x_{i_1}[t_1+2] &\geq \alpha x_{i_1}[t_1+1] + (1-\alpha)\overline{m}[t_1], \\
&\geq \alpha^2 f_r[t_1] + (1+\alpha)(1-\alpha)\overline{m}[t_1], \\
x_{i_1}[t_1+3] &\geq \alpha^3 f_r[t_1] + (1+\alpha+\alpha^2)(1-\alpha)\overline{m}[t_1], \\
&\vdots \qquad\qquad\qquad \vdots \\
x_{i_1}[t_1+k] &\geq \alpha^k f_r[t_1] + \left(\sum_{j=0}^{k-1} \alpha^j\right)(1-\alpha)\overline{m}[t_1], \\
&\geq \alpha^k f_r[t_1] + (1-\alpha^k)\overline{m}[t_1], \\
&\geq \overline{m}[t_1] + \alpha^k \underline{\epsilon}.
\end{aligned} \tag{2.13}$$

This holds for $0 < k \leq T$. Using similar arguments, an upper bound on $x_{i_1}[t_1+k]$ can be established as follows:

$$\begin{aligned}
x_{i_1}[t_1+1] &\leq \alpha f_r[t_1] + (1-\alpha)\overline{M}[t_1], \\
&\vdots \qquad\qquad\qquad \vdots \\
x_{i_1}[t_1+k] &\leq \alpha^k f_r[t_1] + (1-\alpha^k)\overline{M}[t_1], \\
&\leq \overline{M}[t_1] - \alpha^k \overline{\epsilon},
\end{aligned} \tag{2.14}$$

for $0 < k \leq T$. Therefore $x_{i_1}[t_1+T] \in [\overline{m}[t_1] + \alpha^T\underline{\epsilon}, \overline{M}[t_1] - \alpha^T\overline{\epsilon}]$ for all $i_1 \in S_1$.

We show next that $|S_X(t_1, t_1+2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})| < |S_f^{\mathcal{N}}|$. Define C_2 as the set of all $i_2 \in S_f^{\mathcal{N}}$ such that $x_{i_2}[t_1+T] \in [\overline{m}[t_1] + \alpha^T\underline{\epsilon}, \overline{M}[t_1] - \alpha^T\overline{\epsilon}]$. Since $S_1 \subseteq C_2$ by (2.13) and (2.14), C_2 is therefore nonempty. Since each agent in $S_f^{\mathcal{N}}$ always uses its own state in (2.8) as per the SW-MSR algorithm, lower bounds on the state of each $i_2 \in C_2$ can be established as:

$$\begin{aligned}
x_{i_2}[t_1+T+1] &\geq \alpha x_{i_2}[t_1+T] + (1-\alpha)\overline{m}[t_1], \\
&\geq \alpha(\overline{m}[t_1] + \alpha^T\underline{\epsilon}) + (1-\alpha)\overline{m}[t_1], \\
&\geq \alpha^{T+1}\underline{\epsilon} + \overline{m}[t_1] \\
x_{i_2}[t_1+T+2] &\geq \overline{m}[t_1] + \alpha^{T+2}\underline{\epsilon}, \\
&\vdots \qquad\qquad\qquad \vdots \\
x_{i_2}[t_1+T+k] &\geq \overline{m}[t_1] + \alpha^{T+k}\underline{\epsilon},
\end{aligned} \tag{2.15}$$

which holds for $0 < k \leq T$. Similarly, the following upper bounds can be established:

$$\begin{aligned}
x_{i_2}[t_1+T+1] &\leq \alpha x_{i_2}[t_1+T] + (1-\alpha)\overline{M}[t_1], \\
&\leq \alpha(\overline{M}[t_1] - \alpha^T\overline{\epsilon}) + (1-\alpha)\overline{M}[t_1], \\
&\leq \overline{M}[t_1] + \alpha^{T+1}\overline{\epsilon}, \\
x_{i_2}[t_1+T+2] &\leq \overline{M}[t_1] + \alpha^{T+2}\overline{\epsilon}, \\
\vdots &\qquad\qquad \vdots \\
x_{i_2}[t_1+T+k] &\leq \overline{M}[t_1] + \alpha^{T+k}\overline{\epsilon},
\end{aligned} \tag{2.16}$$

which holds for $0 < k \leq T$. These arguments imply that for all $i_2 \in C_2$, $i_2 \notin S_X(t_1, t_1 + 2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})$. Therefore $|S_X(t_1, t_1+2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})| < |S_f^{\mathcal{N}}|$.

We next show that $|S_X(t_1, t_1+3T, \alpha^{3T}\underline{\epsilon}, \alpha^{3T}\overline{\epsilon})| < |S_X(t_1, t_1+2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})|$. Since $\mathcal{D}^T[t]$ is strongly $(T, t_0, 2F+1)$-robust, there exists a nonempty $S_3 \subseteq S_X(t_1, t_1+2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})$ such that for all $i_3 \in S_3$, $|\mathcal{J}_{i_3}^T[t_1+2T] \cap \overline{S}_X(t_1, t_1+2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})| \geq 2F+1$. Since $\mathcal{A}$ is an F-local set, $\mathcal{J}_{i_3}^T[t_1+2T] \cap \overline{S}_X(t_1, t_1+2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})$ includes at least $F+1$ normally-behaving agents from $\mathcal{N}$ $\forall i_3 \in S_3$. Observe that by the definition of $S_X(t_1, t_1+2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})$ the state of each $i_3 \in S_3$ satisfies either $x_{i_3}[t_1+2T] < x_j^{i_3}[\tau_{i_3 j}[t_1+2T]]$ or $x_{i_3}[t_1+2T] > x_j^{i_3}[\tau_{i_3 j}[t_1+2T]]$ for all $j \in \mathcal{N} \cap \overline{S}_X(t_1, t_1+2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})$. Therefore i_3 will incorporate at least one in-neighbor's state from the interval $[\overline{m}[t_1] + \alpha^{2T}\underline{\epsilon}, \overline{M}[t_1] - \alpha^{2T}\overline{\epsilon}]$ in its state update, yielding the following bounds for all $i_3 \in S_3$:

$$\begin{aligned}
x_{i_3}[t_1+2T+1] &\geq \alpha(\overline{m}[t_1] + \alpha^{2T}\underline{\epsilon}) + (1-\alpha)\overline{m}[t_1] \\
&\geq \overline{m}[t_1] + \alpha^{2T+1}\underline{\epsilon} \\
x_{i_3}[t_1+2T+2] &\geq \overline{m}[t_1] + \alpha^{2T+2}\underline{\epsilon} \\
\vdots &\qquad\qquad \vdots \\
x_{i_3}[t_1+2T+k] &\geq \overline{m}[t_1] + \alpha^{2T+k}\underline{\epsilon},
\end{aligned} \tag{2.17}$$

for all $0 < k \leq T$. Similarly,

$$\begin{aligned}
x_{i_3}[t_1+2T+1] &\leq \alpha(\overline{M}[t_1] + \alpha^{2T}\overline{\epsilon}) + (1-\alpha)\overline{M}[t_1] \\
&\leq \overline{M}[t_1] + \alpha^{2T+1}\underline{\epsilon} \\
x_{i_3}[t_1+2T+2] &\leq \overline{M}[t_1] + \alpha^{2T+2}\overline{\epsilon} \\
\vdots &\qquad\qquad \vdots \\
x_{i_3}[t_1+2T+k] &\leq \overline{M}[t_1] + \alpha^{2T+k}\overline{\epsilon}
\end{aligned} \tag{2.18}$$

for all $0 < k \leq T$. This implies that $i_3 \notin S_X(t_1, t_1 + 3T, \alpha^{3T}\underline{\epsilon}, \alpha^{3T}\overline{\epsilon})\ \forall i_3 \in S_3$. Furthermore, we define C_3 as the set of all $j_3 \in S_f^{\mathcal{N}}$ such that $x_{j_3}[t_1 + 2T] \in [\overline{m}[t_1] + \alpha^{2T}\underline{\epsilon}, \overline{M}[t_1] - \alpha^{2T}\overline{\epsilon}]$. By this definition, $C_2 \subseteq C_3$. Note that the bounds in equations (2.17) and (2.18) also apply to all agents $j_3 \in C_3$ since $x_{j_3}[t_1 + 2T] \in [\overline{m}[t_1] + \alpha^{2T}\underline{\epsilon}, \overline{M}[t_1] - \alpha^{2T}\overline{\epsilon}]\ \forall j_3 \in C_3$, and each j_3 does not filter out its own state. Therefore $j_3 \notin S_X(t_1, t_1 + 3T, \alpha^{3T}\underline{\epsilon}, \alpha^{3T}\overline{\epsilon})\ \forall j_3 \in C_3$, and therefore $|S_X(t_1, t_1 + 3T, \alpha^{3T}\underline{\epsilon}, \alpha^{3T}\overline{\epsilon})| < |S_X(t_1, t_1 + 2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})|$.

This logic can be continued iteratively to show that $|S_X(t_1, t_1+pT, \alpha^{pT}\underline{\epsilon}, \alpha^{pT}\overline{\epsilon})| < |S_X(t_1, t_1+(p-1)T, \alpha^{(p-1)T}\underline{\epsilon}, \alpha^{(p-1)T}\overline{\epsilon})|$ for all $p \geq 2$, $p \in \mathbb{Z}$ such that $t_1 + pT < t_2$. This can be done by defining

$$C_p = \{i_p \in S_f^{\mathcal{N}} : x_{i_p}[t_1+(p-1)T] \in [\overline{m}[t_1]+\alpha^{(p-1)T}\underline{\epsilon}, \overline{M}[t_1]+\alpha^{(p-1)T}\overline{\epsilon}]\},$$

which satisfies $C_{p-1} \subseteq C_p$, and considering each $S_X(t_1, t_1 + (p-1)T, \alpha^{(p-1)T}\underline{\epsilon}, \alpha^{(p-1)T}\overline{\epsilon})$ for $p \geq 3$. Since $\mathcal{D}^T[t]$ is $(T, t_0, 2F+1)$-robust, if $S_X(t_1, t_1+(p-1)T, \alpha^{(p-1)T}\underline{\epsilon}, \alpha^{(p-1)T}\overline{\epsilon})$ is nonempty at time $t_1+(p-1)T$ then there exists a nonempty $S_p \subseteq S_X(t_1, t_1+(p-1)T, \alpha^{(p-1)T}\underline{\epsilon}, \alpha^{(p-1)T}\overline{\epsilon})$ such that $\forall i_p \in S_p$, $|\mathcal{J}_{i_p}^T[t_1 + (p-1)] \cap \overline{S}_X(t_1, t_1 + (p-1)T, \alpha^{(p-1)T}\underline{\epsilon}, \alpha^{(p-1)T}\overline{\epsilon})| \geq 2F + 1$. Using prior arguments, it can then be shown that $x_{i_p}[t_1+pT] \in [\overline{m}[t_1]+\alpha^p\underline{\epsilon}, \overline{M}[t_1]-\alpha^p\overline{\epsilon}]$. This implies that $i_p \notin S_X(t_1, t_1+pT, \alpha^{pT}\underline{\epsilon}, \alpha^{pT}\overline{\epsilon})\ \forall i_p \in S_p$. Similarly, by using prior arguments it also holds that $x_{j_p}[t_1 + pT] \in [\overline{m}[t_1]+\alpha^p\underline{\epsilon}, \overline{M}[t_1]-\alpha^p\overline{\epsilon}]\ \forall j_p \in C_p$, and therefore $j_p \notin S_X(t_1, t_1+pT, \alpha^{pT}\underline{\epsilon}, \alpha^{pT}\overline{\epsilon})\ \forall j_p \in C_p$. This implies that $|S_X(t_1, t_1 + pT, \alpha^{pT}\underline{\epsilon}, \alpha^{pT}\overline{\epsilon})| < |S_X(t_1, t_1 + (p-1)T, \alpha^{(p-1)T}\underline{\epsilon}, \alpha^{(p-1)T}\overline{\epsilon})|$ for all $p \geq 2$, $p \in \mathbb{Z}$ such that $t_1 + pT < t_2$.

Since $S_f^{\mathcal{N}} \subset \mathcal{V}$ is finite, there exists a $p' > 1$, $p' \in \mathbb{Z}_{\geq 0}$ such that

$$S_X(t_1, t_1 + (p'+1)T, \alpha^{(p'+1)T}\underline{\epsilon}, \alpha^{(p'+1)T}\overline{\epsilon}) = \emptyset.$$

This implies that for all $i \in S_f^{\mathcal{N}}$,

$$\begin{aligned} x_i[t_1 + (p'+1)T] &\geq \overline{m}[t_1] + \alpha^{(p'+1)T}\underline{\epsilon} \\ x_i[t_1 + (p'+1)T] &\leq \overline{M}[t_1] + \alpha^{(p'+1)T}\overline{\epsilon} \end{aligned} \tag{2.19}$$

Considering $V[t_1 + (p'+1)T]$, we have

$$\begin{aligned} V[t_1 + (p'+1)T] &= \overline{M}[t_1 + (p'+1)T] - \overline{m}[t_1 + (p'+1)T] \\ &\leq \overline{M}[t_1] - \alpha^{(p'+1)T}\overline{\epsilon} - (\overline{m}[t_1] + \alpha^{(p'+1)T}\underline{\epsilon}) \\ &\leq V[t_1] - \alpha^{(p'+1)T}(\underline{\epsilon} + \overline{\epsilon}) \end{aligned} \tag{2.20}$$

Recall that $\underline{\epsilon} = f_r[t_1] - \overline{m}[t_1]$ and $\overline{\epsilon} = \overline{M}[t_1] - f_r[t_1]$. This implies that $\underline{\epsilon} + \overline{\epsilon} = \overline{M}[t_1] - \overline{m}[t_1] =$

$V[t_1]$, implying

$$V[t_1 + (p'+1)T] \leq V[t_1] - \alpha^{(p'+1)T} V[t_1] = (1 - \alpha^{(p'+1)T}) V[t_1] \tag{2.21}$$

Recalling that $|S_X(t_1, t_1 + 2T, \alpha^{2T}\underline{\epsilon}, \alpha^{2T}\overline{\epsilon})| < |S_f^{\mathcal{N}}|$ at time $t_1 + 2T$, and that

$$|S_X(t_1, t_1 + pT, \alpha^{pT}\underline{\epsilon}, \alpha^{pT}\overline{\epsilon})| < |S_X(t_1, t_1 + (p-1)T, \alpha^{(p-1)T}\underline{\epsilon}, \alpha^{(p-1)T}\overline{\epsilon})|$$

for all $p \geq 3$, it follows that $p' \leq |S_f^{\mathcal{N}}|$ since $S_X(t_1, t_1+(p'+1)T, \alpha^{(p'+1)T}\underline{\epsilon}, \alpha^{(p'+1)T}\overline{\epsilon}) = \emptyset$ after no more than $(|S_f^{\mathcal{N}}|+1)T$ time steps. Therefore we have $V[t_1+(|S_f^{\mathcal{N}}|+1)T] \leq (1-\alpha^{(|S_f^{\mathcal{N}}|+1)T})V[t_1]$ by substituting $p' = |S_f^{\mathcal{N}}|$ into (2.21). The above analysis can be repeated to show

$$V[t_1+(|S_f^{\mathcal{N}}|+1)\sigma T] \leq (1-\alpha^{(|S_f^{\mathcal{N}}|+1)T})V[t_1+(\sigma-1)T]$$

for $\sigma \geq 1$, $\sigma \in \mathbb{Z}$ such that $t_1 + (|S_f^{\mathcal{N}}| + 1)\sigma T < t_2$. This yields the result $V[t_1 + T + (|S_f^{\mathcal{N}}| + 1)\sigma T] \leq (1 - \alpha^{(|S_f^{\mathcal{N}}|+1)T})^{\sigma} V[t_1 + T]$ when $t_2 < \infty$.

If $t_2 = \infty$, then $\lim_{t\to\infty} V[t] = \lim_{\sigma\to\infty} V[t_1+T+(|S_f^{\mathcal{N}}|+1)\sigma T] \leq (1-\alpha^{(|S_f^{\mathcal{N}}|+1)T})^{\sigma} V[t_1+T]$. Note that $\alpha < 1$ implies $(1 - \alpha^{(|S_f^{\mathcal{N}}|+1)T}) < 1$, and therefore the limit converges to zero. By (2.10), $\lim_{t\to\infty} V[t] = 0$ implies $\lim_{t\to\infty} \max_{i\in S_f^{\mathcal{N}},\, l\in\mathcal{L}^{\mathcal{N}}} |x_i[t] - x_l[t]| = 0$. □

Remark 2.1. *Although the proof of Theorem 2.1 follows a similar line of reasoning as the results in [79], Theorem 2.1 contains two significant theoretical differences. First, Theorem 2.1 considers the more general* Byzantine *adversarial model [2], whereas the results in [79] consider only malicious adversaries.*[4] *Second, Theorem 2.1 considers consensus of the followers to a specific reference value propagated by the set of normally-behaving leader agents that may lie outside the convex hull of initial agents' states. The analysis in [79] considers leaderless consensus to some unknown value in the convex hull of the initial normal agents' states.*

2.3.3 Adversarial Implications

We next discuss the adversarial implications of Theorem 2.1. In most leaderless resilient consensus settings considered in prior work, the networks consist only of normally-behaving agents seeking a common consensus value, and adversarial agents behaving arbitrarily. Often, these results guarantee resilient consensus if the adversary model is *at most* F-local. However, these results for leaderless resilient consensus raise the following critical question: *What happens if the adversary*

[4] In essence, malicious adversaries may update their state arbitrarily, but will send the same state information to all out-neighbors. Byzantine adversaries may update their state arbitrarily and send different information to different out-neighbors.

model is NOT F-local? To the authors' best knowledge, little (if any) analysis has focused on the precise effects of the F-local assumption being violated in these scenarios. From a practical standpoint it is difficult to provide absolute guarantees that $\mathcal{A}$ will always be strictly F-local in any real-world application of resilient algorithms. It is therefore critical to understand the consequences which will occur if the F-local assumption does not hold.

Theorem 2.1 can be used to show one possible catastrophic outcome if the F-local assumption is violated in a *leaderless* network. More specifically, Theorem 1 can be used to demonstrate that for a leaderless network applying the SW-MSR algorithm, if there exists a colluding set of adversarial agents $\mathcal{A}$ and if the network is strongly $(T, t_0, 2F+1)$-robust with respect to $\mathcal{A}$, then the adversarial agents can drive the states of *all* normal agents *to any arbitrary value*. This result is presented more precisely in the following corollary:

Corollary 2.1. *Let $\mathcal{D}[t] = (\mathcal{V}, \mathcal{E}[t])$ be a nonempty, nontrivial, simple digraph with $\mathcal{L} = \emptyset$. Let $F \in \mathbb{Z}_{\geq 0}$, $t_0, t_1, t_2 \in \mathbb{Z}$ with $t_2 > t_1 \geq t_0 + T$. Suppose that $\mathcal{D}[t]$ is strongly $(T, t_0, 2F+1)$-robust w.r.t. a set of adversarial agents $\mathcal{A}$ and all normally-behaving agents $i \in \mathcal{V}\backslash\mathcal{A}$ apply the SW-MSR algorithm with parameter F. If all agents $j \in \mathcal{A}$ send a constant, common value $x_j^i[t]$ to all of their respective out-neighbors $i \in \mathcal{V}_j^{out}$ for all $t \in [t_1 - T, t_2)$, and if $t_2 > t_1 + (|\mathcal{V}\backslash\mathcal{A}| + 1)\sigma T$ for some $\sigma \in \mathbb{Z}_{\geq 0}$, then the error between the normally-behaving agents' states and the adversaries' common state $x_j^i[t]$ is exponentially decreasing for $t \in [t_1 - T, t_2)$. Furthermore, if $t_2 = \infty$ then $\lim_{t\to\infty} \max_{i\in\mathcal{V}\backslash\mathcal{A}} |x_i[t] - x_j^i[t]| = 0$.*

Proof. The proof follows from Theorem 2.1 by treating $\mathcal{A}$ as the set $\mathcal{L}$, $\mathcal{V}\backslash\mathcal{A}$ as the set $S_f^{\mathcal{N}}$, and $x_j^i[t]$ as the signal $f_r[t]$. Note that by Definition 2.8, $x_j^i[t]$ need not be equal to any of the actual states $x_j[t]$ of $j \in \mathcal{A}$. □

In short, if the digraph $\mathcal{D}$ for a leaderless consensus network is strongly $(T, t_0, 2F+1)$-robust w.r.t. the adversary set $\mathcal{A}$ and the adversaries collude to send a common constant to their out-neighbors on sufficiently long time intervals, the error between the normal agents and the adversarial signal will decrease exponentially. These conditions imply that the adversaries have the ability to drive the entire network to arbitrary state values. When working with a given digraph $\mathcal{D}[t]$, this result demonstrates the need for awareness of the agent subsets S such that $\mathcal{D}[t]$ is strongly $(T, t_0, 2F+1)$-robust w.r.t. S. Adversaries seeking to obtain control of the network will succeed if such subsets are successfully compromised.

2.3.4 Simulations

The resilient leader-follower consensus framework in the first part of this chapter can be applied to a wide range of problems where a network of agents need to be driven to a desired reference

value by a set of leaders. Some examples of such reference values include a reference altitude for unmanned aerial vehicles, a reference rendezvous time for multiple unmanned ground vehicles, and a reference radius for a circular patrolling path [79], to name only a few.

The simulations consider agents connected via time-varying k-circulant digraphs. The Appendix contains the definition of k-circulant digraphs and details about the conditions under which k-circulant digraphs are strongly r-robust w.r.t. a subset. For each simulation the network topology switches between the three graphs depicted in Figure 2.4. The union of the three graphs forms a 7-circulant digraph. The simulations consider the presence of malicious adversaries, which may send the *same* misinformation to their respective out-neighbors [2]. In all simulations, agents have no knowledge as to whether their in-neighbors are normal, malicious, or behaving as leaders. In addition, $t_0 = 0$ and the agents' initial states are random values on the interval $[-25, 25]$ for all agents in $(\mathcal{V} \backslash \mathcal{L})$. The results of the first simulation are shown in figure 2.5. In this simulation, the number of agents is 15, with $\mathcal{L} = \{4, 5, ..., 8\}$ (5 leaders). The time window is $T = 12$ steps, and the network switches graphs every 4 seconds $(\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_1 \ldots)$, where the graphs are depicted in Figure 2.4. By the results of the Appendix, the digraph is strongly $(12, 0, 5)$-robust w.r.t. $\mathcal{L}$. For all normal follower agents, parameter $F = 2$. Two of the agents in the network behave maliciously. The function $f_r[t]$ is simply the constant $f_r[t] = 30$. The error between the normally-behaving agents' states (denoted by colored lines) and the normally-behaving leaders' states (the solid black line) decreases exponentially in the presence of two adversarial agents (the dotted red lines). The second simulation, depicted in Figure 2.6, considers a scenario where $f_r[t]$ takes on different values over time. In this simulation, the network size is 30 agents, with $\mathcal{L} = \{1, 2, \ldots, 7\}$ (7 leaders). The time window for each agent is $T = 30$, and the network switches between graphs every 10 seconds $(\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_1 \ldots)$. By the results of the Appendix, the digraph is strongly $(30, 0, 7)$-robust w.r.t. $\mathcal{L}$. For all normal follower agents, parameter $F = 3$. Three of the agents in the network behave maliciously. The error between the normally-behaving agents and the normally-behaving leaders decreases exponentially on the time intervals where $f_r[t]$ is constant as per the conditions of Theorem 2.1.

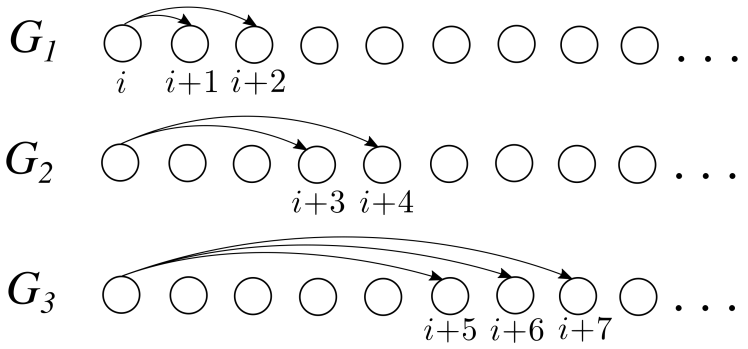


Figure 2.4: Time-varying graphs used in the last two simulations. In each graph $\mathcal{G}_j$, $\forall i \in \mathcal{V}$ each agent i sends its state information to the agents depicted. The terms $i + p$ for $p \in \{1, \ldots, 7\}$ are shorthand for $(i + p) \mod n$, where n is the total number of agents.

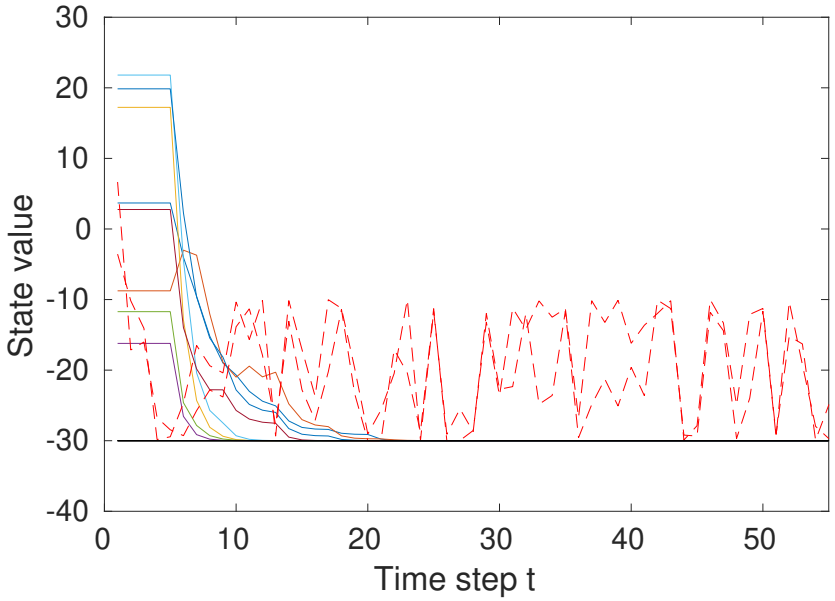


Figure 2.5: Leader-follower simulation using the SW-MSR algorithm with a constant reference value in the presence of 2 malicious agents.

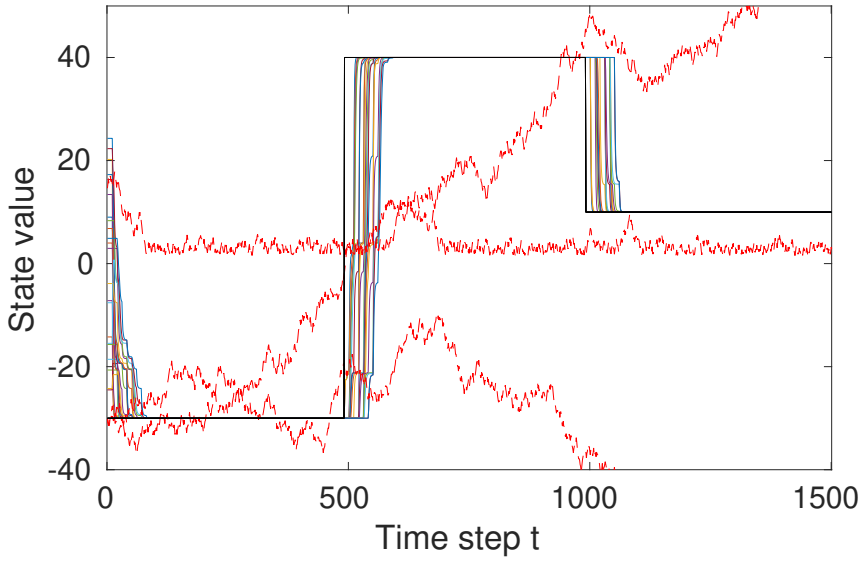


Figure 2.6: Leader-follower simulation using the SW-MSR algorithm with a time-varying reference value in the presence of 3 malicious agents. Note that the normal agents track the reference signal even when the behavior of the malicious agents may be unbounded.

2.4 Finite-Time Leader-Follower Consensus: Formation Control

As discussed in the Introduction to this chapter, prior work on MSR-type algorithms has focused on demonstrating asymptotic or exponential convergence for systems with discrete-time dynamics. Few works have considered MSR-type consensus algorithms in the continuous-time domain, and no prior work has considered finite-time convergence of MSR-type algorithms under a general F-total or F-local adversarial model.

In addition, the majority of MSR-type algorithms consider agents with scalar states. The chief difficulty with applying MSR-type algorithms to vector-valued states is determining a suitable total order on vectors. The filtering step in MSR algorithms requires each agent to sort the information it receives from maximum to minimum value, which requires a total order to exist for the information. One proposed extension to vector-valued states in $\mathbb{R}^n$ has focused on applying n separate MSR algorithms to each dimension and achieving resilient consensus elementwise [84]. However, more general methods for applying MSR techniques to systems with vector-valued states remains an open question.

As discussed in the Introduction, this section presents several contributions addressing these limitations to prior literature. This section presents a method for agents with continuous-time dynamics and states in $\mathbb{R}^n$ to achieve leader-follower consensus in finite time to a formation in $\mathbb{R}^n$.

To achieve this, we present a novel MSR-type filtering algorithm which uses a norm to induce a total order on the vector-valued information and allows for the filtering out of adversarial misinformation. This is the first MSR-type algorithm to use a norm-based filtering approach in this manner. Finally, the method in this section considers input bounds for each of the agents' dynamics, which is an aspect neglected by prior MSR algorithm literature. The section concludes by applying the norm-based filtering approach to a discrete-time system which is shown to converge in exponential time.

2.4.1 Notation and Problem Definition

Similar to the prior section, three subsets of $\mathcal{V}$ are considered in this paper: leader agents $\mathcal{L} \subset \mathcal{V}$, adversarial agents $\mathcal{A} \subset \mathcal{V}$, and normal follower agents denoted $\mathcal{N} \subset \mathcal{V}$.[5] These subsets will be described in more detail later in this section. We denote $\mathcal{A}_i = \mathcal{V}_i \cap \mathcal{A}$, i.e. the set of adversarial agents in the in-neighbour set of agent i. The upper right Dini derivative of a function $g : [a, b) \to \mathbb{R}$ is defined as follows [199]:

$$D^+ g(t) = \limsup_{h \to 0^+} \frac{1}{h}[g(t + h) - g(t)],\ t \in [a, b).$$

Finally, $\|\cdot\|$ in this section denotes any p-norm defined on $\mathbb{R}^n$.

The MSR-type control protocol presented in this section will involve a sorting and filtering method similar to the prior MSR algorithms presented in this chapter. For a given $i \in \mathcal{N}$ the in-neighbors which are *not* filtered out are denoted $\mathcal{K}_i \subseteq \mathcal{V}_i$; i.e. letting $\mathcal{R}_i(t)$ denote the set of agents which are filtered out at time t, we define $\mathcal{K}_i \triangleq \mathcal{V}_i \backslash \mathcal{R}_i$. For brevity of notation, we will denote $\mathcal{K}_i^{\mathcal{N}} = \mathcal{K}_i \setminus (\mathcal{A}_i \cap \mathcal{K}_i)$ and $\mathcal{K}_i^{\mathcal{A}} = \mathcal{A}_i \cap \mathcal{K}_i$, which implies $\mathcal{K}_i = \mathcal{K}_i^{\mathcal{N}} \cup \mathcal{K}_i^{\mathcal{A}}$. We also denote $K_i \triangleq |\mathcal{K}_i(t)|$.

Consider a time-invariant digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ of n agents with states $\boldsymbol{p}_i \in \mathbb{R}^n$. Each agent $i \in \mathcal{V}$ has the system model

$$\delta \boldsymbol{p}_i(t) = \boldsymbol{u}_i(t), \tag{2.22}$$

where $\delta \boldsymbol{p}_i(t)$ denotes the time derivative $\dot{\boldsymbol{p}}_i$ for the case of continuous-time system and the time-difference $\boldsymbol{p}_i[t + 1] - \boldsymbol{p}_i[t]$ for the case of discrete-time systems. The variable $\boldsymbol{u}_i$ is the input to agent i, which will be explained in sections 2.4.2 and 2.4.3 respectively for continuous- and discrete-time system.

There is much prior literature on formation control problems involving a set of leaders to which

[5] Unlike the previous section, $\mathcal{N}$ will be used to denote normally behaving agents which are *not* leaders; i.e. $\mathcal{N} \cap \mathcal{L} = \emptyset$.

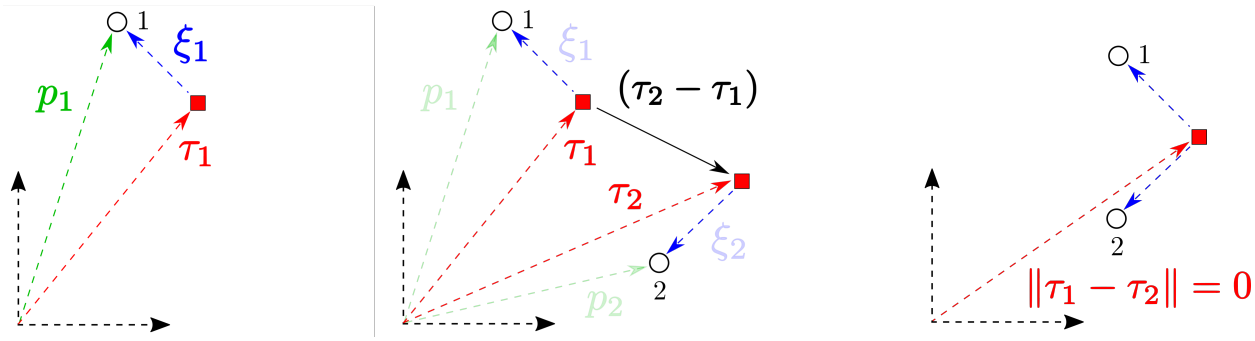


Figure 2.7: Visual depiction of the vectors $\boldsymbol{p}_i$, $\boldsymbol{\xi}_i$, and $\boldsymbol{\tau}_i$ used in this section. The formation is achieved when agents' $\boldsymbol{\tau}$ vectors come to consensus.

the rest of the network converges. We assume that a subset of the agents $\mathcal{L} \subset \mathcal{V}$ are designated to behave as leaders. However, these leaders are not invulnerable to attacks, implying $(\mathcal{L} \cap \mathcal{A})$ may possibly be nonempty. Any nodes which are neither leaders nor adversarial are designated as normal nodes $\mathcal{N} \subset \mathcal{V}$. In all, $\mathcal{N} \cup \mathcal{L} \cup \mathcal{A} = \mathcal{V}$.

We assume that prescribed constant formation vectors $\boldsymbol{\xi}_i \in \mathbb{R}^n$ have been specified for these agents. Each $\boldsymbol{\xi}_i \in \mathbb{R}^n$ represents agent i's desired formational offset from a group reference point. The formation offsets of the entire network is written as $\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}_1^T & \dots & \boldsymbol{\xi}_n^T \end{bmatrix}^T$. As outlined in [24, Chapter 6], we define the variable $\boldsymbol{\tau}_i(t) = \boldsymbol{p}_i(t) - \boldsymbol{\xi}_i$. If non-adversarial agents come to formation on their values of $\boldsymbol{\tau}_i(t)$, i.e. $\|\boldsymbol{\tau}_i(t) - \boldsymbol{\tau}_j(t)\| \to 0 \ \forall i, j \in (\mathcal{L} \cup \mathcal{N}) \backslash \mathcal{A}$ then they have achieved formation. In essence, the agents will have achieved consensus on the center of formation. The behaving leaders are assumed to be maintaining their $\boldsymbol{\tau}$ values at some arbitrary point $\boldsymbol{\tau}_L$. The goal of this section is to design a resilient control protocol such that all the normal behaving agents can come to formation at $\boldsymbol{\tau}_L$. We assume that each agent i is able to obtain the time-varying relative vectors $\boldsymbol{\tau}_j(t) - \boldsymbol{\tau}_i(t)$ for all $j \in \mathcal{V}_i$. Two ways in which this might be accomplished include each agent i measuring this vector via on-board sensors or calculating it by receiving transmitted messages from each $j \in \mathcal{V}_i$. In the former case, it is required that agents share a common reference orientation, and in the latter both a reference orientation and a common origin point.

We assume that there exists a subset of the agents $\mathcal{A} \subset \mathcal{V}$ that is adversarial, and that $\mathcal{A}$ is an *F-total* set; i.e. for any $i \in (\mathcal{V} \backslash \mathcal{A})$, $|\mathcal{V}_i \cap \mathcal{A}| \leq F$ ([75]). Any adversarial agent $k \in \mathcal{A}$ may attempt to prevent its normal out-neighbors from coming to formation by manipulating the value of $\boldsymbol{\tau}_k(t)$ received by its out-neighbors. Two ways in which this may occur include physical or communication misbehavior. Explicitly, an agent misbehaves if at any time $t \geq t_0$ it applies a different control law than the nominal one in the former case, or by sending false information in the latter. In either case, this misbehavior is modeled as normal agent i obtaining the time-varying relative vector $\boldsymbol{\tau}_k(t) - \boldsymbol{\tau}_i(t)$ where the adversarial dynamics of the value of $\boldsymbol{\tau}_k(t)$ received by any

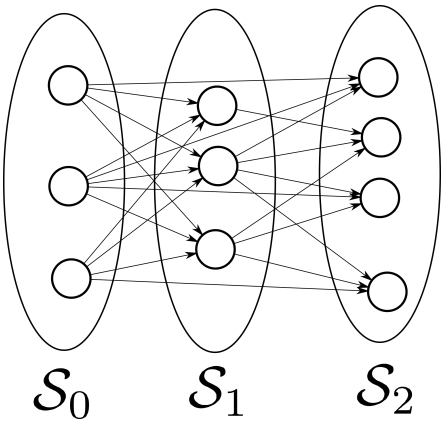


Figure 2.8: An example of a Resilient Directed Acyclic Graph (RDAG) with parameter $r = 3$.

normal agent i are

$$\delta\boldsymbol{\tau}_k(t) = \boldsymbol{f}_{k,m}(t), \tag{2.23}$$

where $\boldsymbol{f}_{k,m}(t)$ is the adversarial input. This adversarial behavior is malicious ([75]) in the sense that each out-neighbor of k receives the same misbehavior. As outlined in [190], since in the continuous time case each normal agent will have continuous state trajectories, any discontinuity in an adversarial agent's transmitted signal could expose its misbehavior to the network. Hence we assume that in the continuous time case, the time-varying relative vector $\boldsymbol{\tau}_k(t) - \boldsymbol{\tau}_i(t)$ obtained by any normal agent $i \in \mathcal{N}$ from any adversary $k \in \mathcal{A}$ is continuous. The assumption of continuity of $\boldsymbol{\tau}_k(t)$ is also made for the case where agents make on-board measurements.

The method for finite-time resilient formational consensus in this section employs a graph-theoretical structure which we call a Resilient Directed Acyclic Graph (RDAG). This structure is a special case of a class of graphs called Mode Estimated Directed Acyclic Graphs (MEDAGs) [200], and is defined as follows:

Definition 2.13. *A digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ is a* Resilient Directed Acyclic Graph *(RDAG) with parameter r if it satisfies the following properties for an integer $r \geq 0$:*

1. *There exists a partitioning of $\mathcal{V}$ into sets $\mathcal{S}_0, \ldots, \mathcal{S}_m \subset \mathcal{V},\ m \in \mathbb{Z}$ such that $|\mathcal{S}_j| \geq r$ for all $0 \leq j \leq m$.*

2. *For each $i \in \mathcal{S}_j,\ 1 \leq j \leq m$, $\mathcal{V}_i \subseteq \bigcup_{k=0}^{j-1} S_k$*

An example of an RDAG is depicted in Figure 2.8. Intuitively, an RDAG is a graph defined by successive subsets of agents $\mathcal{S}_j$. Agents in each subset only have in-neighbors from preceding subsets. The purpose of an RDAG is to introduce enough edge redundancy to ensure the existence of an unfiltered directed path of behaving nodes from the leaders to each normal agent. This can be

achieved by designating all agents in the set $\mathcal{S}_0$ to behave as leaders, i.e. $\mathcal{S}_0 = \mathcal{L}$. In our analysis, we consider RDAGs with parameter $r \geq 3F + 1$. As we will show, an RDAG of this form is a sufficient condition implying that normal agents applying our filtering methods and controllers will converge to the leaders. A method exists by which RDAGs can be constructed from existing graph topologies even in the presence of adversaries ([191]). This method involves agents successively receiving in a resilient manner and rebroadcasting a communication signal initiated by the set of leaders, identifying their own set and the agents in the preceding set, and then restricting their in-neighbor set to only agents in the preceding set. In particular, an RDAG can be constructed from an initial graph that is strongly robust with respect to a subset $\mathcal{S} \subset \mathcal{V}$. An example of such a graph is a k-circulant graph [102, 181]. The existence of an RDAG graph structure does not guarantee that normal agents are able to identify adversarial agents. Rather, the edge redundancy guarantees that each normal agent has enough behaving in-neighbors to still achieve formation under the proposed controllers.

2.4.2 Continuous-time System

2.4.2.1 Filtering Algorithm and Control Law

We first consider the continuous-time setting. In this setting, each agent applies Algorithm 2.3 at every time instance $t = m\epsilon_d$, where $\epsilon_d > 0$ is defined later in this section.

Algorithm 2.3 Continuous-Time Filtering

procedure UPDATEFILTEREDLIST
  Calculate $\tau_{ij} = \|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\| \ \forall j \in \mathcal{V}_i$
  if $t = m\epsilon_d,\ m \in \mathbb{Z}_{\geq 0},\ \epsilon_d > 0$ **then**
    Sort τ_{ij} values such that $\tau_{ij_1} \geq \ldots \geq \tau_{ij_{|\mathcal{V}_i|}}$
    $\mathcal{K}_i(t) \leftarrow \{j : \tau_{ij} \in \{\tau_{ij_{F+1}}, \ldots, \tau_{ij_{|\mathcal{V}_i|}}\}\}$
  end if
end procedure

The dynamics of continuous time $\boldsymbol{\tau}(t)$ are given as:

$$\dot{\boldsymbol{\tau}}_i(t) = \dot{\boldsymbol{p}}_i(t) - \dot{\boldsymbol{\xi}}_i = \boldsymbol{u}_i(t). \tag{2.24}$$

We will sometimes omit the argument t for the sake of brevity when the dependence on t is clear from the context. We assume that the speed of each agent i is bounded above by u_M, i.e. $\|\boldsymbol{u}_i(t)\| \leq$

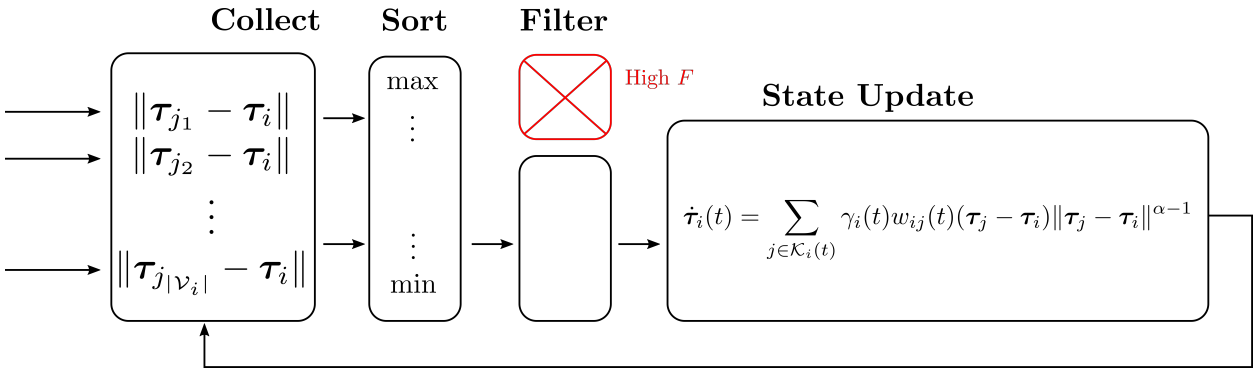


Figure 2.9: Diagram of the filtering and state update control law used for the continuous time system in this section. Note that as per Algorithm 2.3, the filtered set $\mathcal{K}_i(t)$ is updated only at time instances $t = m\epsilon_d$ where $\epsilon_d > 0$, $m \in \mathbb{Z}_{\geq 0}$. The reasons for this behavior are discussed below.

u_M for all $t \geq 0$. Under this constraint, the saturation function is defined as

$$\sigma_i(t) = \min\{\|\boldsymbol{u}_i^p(t)\|, u_M\}, \tag{2.25}$$

$$\boldsymbol{u}_i^p(t) = \sum_{j \in \mathcal{K}_i(t)} w_{ij}(t)(\boldsymbol{\tau}_j(t) - \boldsymbol{\tau}_i(t))\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1}, \tag{2.26}$$

where $0 < \alpha < 1$. To simplify the notation, define the term $\gamma_i(t) = \frac{\sigma_i(t)}{\|\boldsymbol{u}_i^p\|}$. With this saturation function[6], inspired from the control law used in [201] and using results from [202], we define the continuous time control law as:

$$\boldsymbol{u}_i(t) = \sum_{j \in \mathcal{K}_i(t)} \gamma_i(t) w_{ij}(t)(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1} \tag{2.27}$$

where $0 < \alpha < 1$. It can be verified from (2.27) that $\|\boldsymbol{u}_i(t)\| \leq u_M$ for all $t \geq 0$ and that the control input goes to zero as agent i goes to its equilibrium.[7] Note that for $\alpha = 1$, the control law (2.27) is same as the traditional formation control law (see [203] for example), while for $\alpha = 0$, we obtain a control law similar to the one introduced in [204]. We make use of this type of controller to not only ensure that $\boldsymbol{\tau}_i$ converges to $\boldsymbol{\tau}_L$, but does so in finite time.

As opposed to [84], this protocol is designed such that agents do not update their filtered list $\mathcal{K}_i(t)$ at every time instance t, but instead only at time instances $t_1, t_2, t_3, ...$ while keeping it constant during the interval (t_l, t_{l+1}). Each of these intervals have constant length, i.e. $t_{l+1} - t_l = \epsilon_d$ for all $l \in \{1, 2, 3, ...\}$ where $\epsilon_d > 0$ is a user-defined small, positive constant. The weights $w_{ij}(t)$ for all $i \in \mathcal{N}$ are designed such that malicious agents are not able to exploit this behavior of $\mathcal{K}_i(t)$.

[6] For all $t \geq 0$, $0 \leq \gamma_i(t) \leq 1$. Note that if the distances of agent from its in-neighbours $j \in \mathcal{K}_i$ are finite, then $\gamma_i(t)$ is strictly positive.

[7] As $\boldsymbol{\tau}_j \to \boldsymbol{\tau}_i$, term $(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1} \to 0$ for $\alpha > 0$

Let $\Xi_i(t)$ be the set of in-neighbour agents whose $\boldsymbol{\tau}$ vectors are NOT equal to that of agent i, i.e.

$$\Xi_i(t) = \{j \in \mathcal{V}_i : \|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\| > 0\}.$$

Then for all $i \in \mathcal{N}$, we define the control weights $w_{ij}(t)$ for all $j \in \mathcal{K}_i(t)$ as

$$w_{ij}(t) = \begin{cases} 0, & |\Xi_i(t)| \leq F, \\ \frac{1}{K_i}, & |\Xi_i(t)| > F. \end{cases} \tag{2.28}$$

To the authors' best knowledge, this choice of control weights have never been introduced in the prior literature. Intuitively, this implies that each normal agent i will have a velocity of zero if its $\boldsymbol{\tau}$ is co-located with the $\boldsymbol{\tau}$ of all but at most F of its in-neighbors. We impose this constraint to ensure that when all normal agents' $\boldsymbol{\tau}$ values have converged to $\boldsymbol{\tau}_L$, the malicious agents cannot perturb them away from $\boldsymbol{\tau}_L$ during the dwell time. This could happen, for example, if for some $i \in \mathcal{N}$, $\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_k\| = 0$ at all $t = m\epsilon_d$ and $\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_k\| > 0$ for time $t \in (m\epsilon_d, (m+1)\epsilon_d)$, where $k \in \mathcal{A}_i$, $m \in \mathbb{Z}_{\geq 0}$. Since $\mathcal{K}_i(t)$ is constant for each $t \in [m\epsilon_d, (m+1)\epsilon_d)$, the malicious agents would not be filtered out by agent i. The properties we impose on the weights prevent the malicious agents from steering the normal agents away during such period.

Theorem 2.2. *For each agent $i \in \mathcal{N}$, $|\Xi_i(t)| \leq F$ for all $t \geq t_i$ if and only if $\boldsymbol{\tau}_i(t) = \boldsymbol{\tau}_L$ for all $t \geq t_i$, for some time t_i.*

Proof. Sufficiency: Assume that there exists some time instant t_i such that for all future times $t \geq t_i$, $\|\boldsymbol{\tau}_i(t) - \boldsymbol{\tau}_L\| \equiv 0$. This can only happen if all the filtered in-neighbors of the agent i (i.e. $j \in \mathcal{K}_i$) are at $\boldsymbol{\tau}_L$. To see why this is true, assume that there exists a filtered in-neighbour of agent i which is not at $\boldsymbol{\tau}_L$. Then, by the virtue of the control law (2.27), agent i would have a non-zero control input $\boldsymbol{u}_i(t)$, which is a contradiction to the assumption that agent stays at the point $\boldsymbol{\tau}_L$. Hence, all its filtered in-neighbours are at the point $\boldsymbol{\tau}_L$. Since we assume that there are at most F agents in the filtered set $\mathcal{V}_i \backslash \mathcal{K}_i$, we have that at most these F agents may not be at τ_L, i.e. $|\Xi_i(t)| \leq F$ and $w_{ij}(t) = 0 \; \forall j \in \mathcal{K}_i$.

Necessity: We prove this by contradiction. Let us assume that there exist $\boldsymbol{\tau}^* \neq \boldsymbol{\tau}_L$ and a time t_i such that $\boldsymbol{\tau}_i(t) = \boldsymbol{\tau}^*$ and in addition we have that $|\Xi_i(t)| \leq F$ for all $t \geq t_i$. Let us assume that $i \in \mathcal{S}_p$. Since $|\mathcal{V}_i| \geq 3F + 1$ and $|\Xi_i(t)| \leq F$, there are at least $2F + 1$ in-neighbors which are also staying at $\boldsymbol{\tau}^*$. This implies that there is at least one normal behaving agent in the in-neighbour set of agent i in the set $\bigcup_{l=0}^{p-1} \mathcal{S}_l$, which stays at $\boldsymbol{\tau}^*$. This in turn means that one of its normal behaving in-neighbors in the set $\bigcup_{l=0}^{p-2} \mathcal{S}_l$ stays identically at $\boldsymbol{\tau}^*$. Using this argument recursively, we have that there exists a normal in-neighbor in the set $\mathcal{S}_0$, which stays identically at the location $\boldsymbol{\tau}^*$.

Since all the normal behaving in-neighbors $\mathcal{S}_0$ stay at $\boldsymbol{\tau}_L$, this contradicts the assumption $\boldsymbol{\tau}^* \neq \boldsymbol{\tau}_L$. Hence, we obtain $\boldsymbol{\tau}_i^* = \boldsymbol{\tau}_L$, and that $|\Xi(t)| \leq F$ for all $t \geq t_i$ only if $\boldsymbol{\tau}_i(t) = \boldsymbol{\tau}_L$ for all $t \geq t_i$. $\square$

Intuitively speaking, Theorem 2.2 states that if there exists a time t_i such that the set $|\Xi_i(t)| \leq F$ for all future time after t_i, this implies that $\boldsymbol{\tau}_i$ has converged to the normal leader reference value $\boldsymbol{\tau}_L$ and will remain there for all future time after t_i. Since all weights $w_{ij}(t) = 0$ when $|\Xi_i(t)| \leq F$, this fact establishes that there will never exist a situation when all weights $w_{ij}(t) = 0$ become zero for i for all future time with $\boldsymbol{\tau}_i$ not equal to the reference value $\boldsymbol{\tau}_L$.

2.4.2.2 Convergence Analysis for Continuous-Time System

We now prove that under the control law (2.27), filtering Algorithm 2.3, and the definition of control weights w_{ij} in (2.28), all the normal behaving agents achieve formation in finite time, despite the presence of adversarial agents. First, we show that for each normal agent $i \in \mathcal{S}_1$, $\|\boldsymbol{\tau}_i(t) - \boldsymbol{\tau}_L\|$ converges to zero in finite time:

Lemma 2.2. *Consider a digraph $\mathcal{D}$ which is an RDAG with parameter $3F+1$, where $\mathcal{S}_0 = \mathcal{L}$ and $\mathcal{A}$ is an F-total set. For each normal agent $i \in \mathcal{S}_1$, $\boldsymbol{\tau}_L$ is a globally finite-time stable equilibrium for the closed-loop dynamics* (2.24)-(2.28).

Proof. Choose the candidate Lyapunov function $V(\boldsymbol{\tau}_i) = \frac{1}{2}\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|^2$. Since $\dot{\boldsymbol{\tau}}_i$ is piece-wise continuous in each interval (t_l, t_{l+1}), the trajectory $\boldsymbol{\tau}_i(t)$ is piecewise differentiable in each such interval. Let $\dot{\boldsymbol{\tau}}_i(t_{l+1}^-)$ and $\dot{\boldsymbol{\tau}}_i(t_{l+1}^+)$ denote the value of the vector $\dot{\boldsymbol{\tau}}_i$ just before and after the filtering at time instant t_{l+1}, respectively. Because the right hand side of (2.27) is bounded at the beginning of each interval, the upper right Dini derivative is defined for $\boldsymbol{\tau}_i(t)$ everywhere, and takes values as

$$D^+(V(\boldsymbol{\tau}_i))(t) = \begin{cases} \nabla V(\boldsymbol{\tau}_i)\dot{\boldsymbol{\tau}}_i(t), & t_l \leq t < t_{l+1}, \\ \nabla V(\boldsymbol{\tau}_i)\dot{\boldsymbol{\tau}}_i(t_{l+1}^+), & t = t_{l+1}. \end{cases},$$

For the worst case, assume that there are F adversarial agents and $K_i - F$ leaders in the filtered list $\mathcal{K}_i$. This requires that the adversarial agent should satisfy $\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_j\| \leq \|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|$ for all $j \in \mathcal{A}_i$ and for all $t \geq 0$, otherwise agent j would be filtered out as per Section 2.4.2.1. Using this and taking the upper right Dini-derivative of the candidate Lyapunov function along the closed loop

trajectories of (2.24), we have:

$$\begin{aligned}
D^+(V(\boldsymbol{\tau}_i)) &= (\boldsymbol{\tau}_i - \boldsymbol{\tau}_L)^T \sum_{j\in\mathcal{K}_i^{\mathcal{N}}} \gamma_i w_{ij}(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1} \\
&\quad + (\boldsymbol{\tau}_i - \boldsymbol{\tau}_L)^T \sum_{j\in\mathcal{K}_i^{\mathcal{A}}} \gamma_i w_{ij}(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1} \\
&= \gamma_i \frac{K_i - F}{K_i}(\boldsymbol{\tau}_i - \boldsymbol{\tau}_L)^T(\boldsymbol{\tau}_L - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_L - \boldsymbol{\tau}_i\|^{\alpha-1} \\
&\quad + (\boldsymbol{\tau}_i - \boldsymbol{\tau}_L)^T \sum_{j\in\mathcal{K}_i^{\mathcal{A}}} \gamma_i w_{ij}(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1}
\end{aligned}$$

Since $\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_j\| \leq \|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|$ for all $j \in \mathcal{K}_i^{\mathcal{A}}$, we have:

$$\begin{aligned}
D^+(V(\boldsymbol{\tau}_i)) &\leq -\gamma_i \frac{K_i - F}{K_i}\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|^{1+\alpha} + \gamma_i \sum_{j\in\mathcal{K}_i^{\mathcal{A}}} w_{ij}\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1} \\
&\leq -\gamma_i \frac{K_i - F}{K_i}\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|^{1+\alpha} + \gamma_i \frac{F}{K_i - F}\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|\|\boldsymbol{\tau}_L - \boldsymbol{\tau}_i\|\|\boldsymbol{\tau}_L - \boldsymbol{\tau}_i\|^{\alpha-1} \\
\Rightarrow D^+(V(\boldsymbol{\tau}_i)) &\leq -cV(\boldsymbol{\tau}_i)^\beta,
\end{aligned}$$

where $\beta = \frac{1+\alpha}{2} < 1$. Note that $D^+(V(\boldsymbol{\tau}_i)) \leq 0$ which means that the Lyapunov candidate $V(\boldsymbol{\tau}_i(t))$ is bounded by $V(\boldsymbol{\tau}_i(0))$. This implies that the agent i remains at a bounded distance from the leaders. Also, if any adversarial agent's state moves further away, by the filtering algorithm, they would be filtered out. Hence, each term in $\boldsymbol{u}_i^p$ remains bounded, which in turn means that $\gamma_i(t) > 0$. Define $\gamma_i^* = \min_t \gamma_i(t)$. Hence, we have that $c \triangleq \gamma_i^* \frac{K_i - 2F}{K_i} > 0$. From the results in [199], since Dini derivative satisfies $D^+(V(\boldsymbol{\tau}_i)) \leq -cV(\boldsymbol{\tau}_i)^\beta$ for all $\boldsymbol{\tau}_i \in \mathbb{R}^2$, we obtain that $\boldsymbol{\tau}_L$ is finite-time stable, with the bound on the finite time of convergence given as $T_{1i} \leq \frac{V(\boldsymbol{\tau}_i(0))^{1-\beta}}{c(1-\beta)} = \frac{\|\boldsymbol{\tau}_i(0) - \boldsymbol{\tau}_L\|^{2(1-\beta)}}{2^{1-\beta}c(1-\beta)}$. Now, at $t = T_{1i}$, agent i has its $\boldsymbol{\tau}_i$ co-located with all the normal leaders' $\boldsymbol{\tau}$. This means that there can be at max F agents (i.e. the adversarial leaders) which are not co-located with the agent's $\boldsymbol{\tau}_i$. Hence, we obtain that $|\Xi_i(t)| \leq F$ for all $t \geq T_{1i}$. Therefore, by Theorem 2.2 agent i will stay at $\boldsymbol{\tau}_L$ for all future times. □

Next we take the case of normal agents $i \in \mathcal{S}_2$:

Lemma 2.3. *Consider a digraph $\mathcal{D}$ which is an RDAG with parameter $3F+1$, where $\mathcal{S}_0 = \mathcal{L}$ and $\mathcal{A}$ is an F-total set. Under the closed loop dynamics (2.24)-(2.28), the value $\boldsymbol{\tau}_i(t)$ for each normal agent $i \in \mathcal{S}_2$ converges to $\boldsymbol{\tau}_L$ in finite time T_{2i}.*

Proof. For the worst case analysis, assume that all the agents in $\mathcal{K}_i(0)$ are from $\mathcal{S}_1$ and are located such that $(\boldsymbol{\tau}_j(0) - \boldsymbol{\tau}_i(0))^T(\boldsymbol{\tau}_L - \boldsymbol{\tau}_i(0)) < 0$ for each $j \in \mathcal{K}_i(0)$. This simply means that the

agents in $\mathcal{K}_i$ at time $t = 0$ are located on one side of the agent while the leaders are on the other side. This is the worst case because this arrangement of in-neighbors would make agent i move away from the leaders, initially. Also, assume that $|\mathcal{K}_i^{\mathcal{A}}| = F$ and $|\mathcal{K}_i^{\mathcal{N}}| = K_i - F$, so that agent i has maximum number of adversarial in-neighbours. Consider the candidate Lyapunov function $V(\boldsymbol{\tau}_i(t)) = \frac{1}{2}\|\boldsymbol{\tau}_i(t) - \boldsymbol{\tau}_L\|^2$. Taking its upper right Dini derivative along the closed-loop trajectories of agent i, we have $D^+(V(\boldsymbol{\tau}_i)) = (\boldsymbol{\tau}_i - \boldsymbol{\tau}_L)^T \sum_{j\in\mathcal{K}_i} \gamma_i w_{ij}(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1}$. Now, from the assumption on the initial locations of agents in $\mathcal{K}_i(t)$, we have that $D^+(V(\boldsymbol{\tau}_i(0))) = \gamma_i(0)\sum_{j\in\mathcal{K}_i} w_{ij}(\boldsymbol{\tau}_i - \boldsymbol{\tau}_L)^T(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1} > 0$. Also, define $T_1 \triangleq \max_{l\in\mathcal{S}_1\cap\mathcal{N}} T_{1l}$, i.e. T_1 is the maximum time after which each normal agent in $\mathcal{S}_1$ would achieve formation and have $\boldsymbol{\tau}_i = \boldsymbol{\tau}_L$. Hence, at time $t = T_1$, we have that:

$$\begin{aligned}
D^+(V(\boldsymbol{\tau}_i)) &= \sum_{j\in\mathcal{K}_i^{\mathcal{N}}} \gamma_i w_{ij}(\boldsymbol{\tau}_i - \boldsymbol{\tau}_L)^T(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1} \\
&+ \sum_{j\in\mathcal{K}_i^{\mathcal{A}}} \gamma_i w_{ij}(\boldsymbol{\tau}_i - \boldsymbol{\tau}_L)^T(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1} \\
&= \gamma_i \frac{K_i - F}{K_i}(\boldsymbol{\tau}_i - \boldsymbol{\tau}_L)^T(\boldsymbol{\tau}_L - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_L - \boldsymbol{\tau}_i\|^{\alpha-1} \\
&+ \sum_{j\in\mathcal{K}_i^{\mathcal{A}}} \gamma_i w_{ij}(\boldsymbol{\tau}_i - \boldsymbol{\tau}_L)^T(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha-1} \\
&\leq -\gamma_i \frac{K_i - F}{K_i}\|\boldsymbol{\tau}_L - \boldsymbol{\tau}_i\|^{1+\alpha} + \gamma_i \sum_{j\in\mathcal{K}_i^{\mathcal{A}}} w_{ij}\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\|^{\alpha}
\end{aligned}$$

Now, for all $j \in \mathcal{K}_i^{\mathcal{A}}$, the norm $\|\boldsymbol{\tau}_j(T_1) - \boldsymbol{\tau}_i(T_1)\| \leq \|\boldsymbol{\tau}_k(T_1) - \boldsymbol{\tau}_i(T_1)\|$ for some $k \in \mathcal{K}_i^{\mathcal{N}}$ otherwise, these adversarial agents would be filtered out. Using this and the fact that $\boldsymbol{\tau}_k(T_1) = \boldsymbol{\tau}_L$, we have that for all $t \geq T_1$:

$$\begin{aligned}
D^+(V(\boldsymbol{\tau}_i(t))) &\leq -\gamma_i \frac{K_i - F}{K_i}\|\boldsymbol{\tau}_L - \boldsymbol{\tau}_i\|^{1+\alpha} + \sum_{j\in\mathcal{K}_i^{\mathcal{A}}} \gamma_i w_{ij}\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|\|\boldsymbol{\tau}_L - \boldsymbol{\tau}_i\|^{\alpha} \\
&= -\gamma_i \frac{K_i - 2F}{K_i}\|\boldsymbol{\tau}_L - \boldsymbol{\tau}_i\|^{1+\alpha} < 0.
\end{aligned}$$

Since $D^+(V(\boldsymbol{\tau}_i))(0) > 0$ while $D^+(V(\boldsymbol{\tau}_i))(T_1) < 0$, and it is bounded above in the interval $(0, T_1)$, the increment in the value of $V(\boldsymbol{\tau}_i)$ is bounded in the interval. Hence, agent i would be at a finite distance away from the leaders at time T_1. This also implies that $\boldsymbol{u}_i^p(t)$ is bounded and hence $\gamma_i^* = \min_t \gamma_i(t) > 0$. Hence, we obtain that $D^+(V(\boldsymbol{\tau}_i)) \leq -cV(\boldsymbol{\tau}_i)^\beta$ where $c = \gamma_i^* \frac{K_i - 2F}{K_i} > 0$ and $\beta = \frac{1+\alpha}{2} < 1$. Hence, we have that $\boldsymbol{\tau}_i \to \boldsymbol{\tau}_L$ in finite time. Let $\boldsymbol{\tau}_i(T_1)$ be the position of agent at time instant T_1. Using the bound on finite time of convergence, we obtain that for $t \geq T_{2i}$,

$\boldsymbol{\tau}_i(t) = \tau_L$ where

$$T_{2i} \leq T_1 + \frac{V(\boldsymbol{\tau}_i(T_1))^{1-\alpha}}{c(1-\alpha)} = T_1 + \frac{\|\boldsymbol{\tau}_i(T_1) - \boldsymbol{\tau}_L\|^{2(1-\beta)}}{2^{1-\beta}c(1-\beta)}$$

Since both T_1 and $\|\boldsymbol{\tau}_i(T_1) - \boldsymbol{\tau}_L\|$ are finite, $\alpha < 1$ and $c > 0$ we obtain that T_{2i} is also finite. Again, after time instant T_{2i}, agent i has its $\boldsymbol{\tau}_i$ co-located with all the normal in-neighbors' $\boldsymbol{\tau}$. This means that there can be at max F agents (i.e. the adversarial agents) which are not co-located with the agent's $\boldsymbol{\tau}_i$. Hence, we have that $|\Xi_i(t)| \leq F$ for all $t \geq T_{2i}$. Therefore, Theorem 2.2 implies that agent i will stay at $\boldsymbol{\tau}_L$ for all $t \geq T_{1i}$. □

We have shown that each normal agent $i \in \mathcal{S}_2$ will achieve the formation in finite time. Now we present the general case:

Theorem 2.3. *Consider a digraph $\mathcal{D}$ which is an RDAG with parameter $3F + 1$, where $\mathcal{S}_0 = \mathcal{L}$ and $\mathcal{A}$ is an F-total set. Under the closed loop dynamics (2.24)-(2.28), $\boldsymbol{\tau}_i$ will converge to $\boldsymbol{\tau}_L$ in finite time for all normal agents $i \in \mathcal{N}$.*

Proof. We have already shown that all the agents in $\mathcal{S}_1$ and $\mathcal{S}_2$ will achieve formation in finite time. Consider any agent $i \in S_3$. Since all the in-neighbors of agents in $\mathcal{S}_3$ are from $\bigcup_{i=0}^{2} \mathcal{S}_i$, after a finite time period all the agents in $\mathcal{V}_i \cap \mathcal{N}$ will satisfy $\boldsymbol{\tau}_i = \boldsymbol{\tau}_L$. Define $T_2 \triangleq \max_k T_{2k}$, where k belongs to the set of normal agents in $\mathcal{S}_1$. After the time instant $t = T_2$, the Lyapunov candidate $V(\boldsymbol{\tau}_i) = \frac{1}{2}\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|^2$ and its Dini derivative will satisfy the conditions similar to Lemma 2.3. Hence, we have that all the normal agents in $\mathcal{S}_3$ will achieve formation in finite time. This time can be bounded as $T_{3i} \leq T_2 + \frac{\|\boldsymbol{\tau}_i(T_2) - \boldsymbol{\tau}_L\|^{1-\alpha}}{c(1-\alpha)}$ for each $i \in \mathcal{S}_3$. This argument can be used recursively to show that each normal agent in $\bigcup_{l=1}^{p} \mathcal{S}_l$ will achieve formation in finite time. Defining T_l as the maximum time by which all the normal agents in set $\mathcal{S}_l$ will achieve the formation, one can establish the following relation for $l \geq 1$:

$$T_{l+1} \leq T_l + \max_{i \in \mathcal{S}_{l+1}} \frac{\|\boldsymbol{\tau}_i(T_l) - \boldsymbol{\tau}_L\|^{2(1-\beta)}}{2^{1-\beta}c(1-\beta)},$$

where $T_1 \leq \max_{i \in \mathcal{S}_1} \frac{\|\boldsymbol{\tau}_i(0) - \boldsymbol{\tau}_L\|^{2(1-\beta)}}{2^{1-\beta}c(1-\beta)}$. Since T_l and $\|\boldsymbol{\tau}_i(T_l) - \boldsymbol{\tau}_L\|$ both are finite $\forall\ l \geq 1$, we have $T_{l+1} < \infty$. □

Hence, we have shown that under the effect of our protocol, each normal agent i would achieve formation in finite time, despite adversarial agents. In the next section, we show that our filtering mechanism can be used for the case of discrete time systems as well.

2.4.3 Discrete-time System

In this subsection we demonstrate that the norm-based sorting and filtering law defined previously can be applied in the discrete-time domain as well. More specifically, we demonstrate that normal follower agents with bounded inputs and discrete-time dynamics can achieve formational consensus in exponential time to a formation defined by normally-behaving leaders in the presence of adversarial agents.

2.4.3.1 Filtering Algorithm and Control Law

At each time step t, each agent $i \in \mathcal{N}$ applies the following algorithm:

Algorithm 2.4 Discrete-Time Filtering

procedure UPDATEFILTEREDLIST
  Calculate $\tau_{ij} = \|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\| \ \forall j \in \mathcal{V}_i$
  Sort τ_{ij} values such that $\tau_{ij_1} \geq \ldots \geq \tau_{ij_{|\mathcal{V}_i|}}$
  $\mathcal{K}_i[t] \leftarrow \{j : \tau_{ij} \in \{\tau_{ij_{F+1}}, \ldots, \tau_{ij_{|\mathcal{V}_i|}}\}\}$
end procedure

The discrete time system dynamics are given as

$$\begin{aligned}\boldsymbol{\tau}_i[t+1] &= \boldsymbol{p}_i[t+1] - \boldsymbol{\xi}_i = \boldsymbol{p}_i[t] + \boldsymbol{u}_i[t] - \boldsymbol{\xi}_i \\ &= \boldsymbol{\tau}_i[t] + \boldsymbol{u}_i[t] \end{aligned} \tag{2.29}$$

The input of each agent i is bounded above by $u_M > 0$, i.e. $\|\boldsymbol{u}_i[t]\| \leq u_M$ for all $t \geq 0$. Under this constraint, the saturation function is given as

$$\sigma_i[t] = \min\{\|\boldsymbol{u}_i^p[t]\|, u_M\}, \tag{2.30}$$

$$\boldsymbol{u}_i^p[t] = \sum_{j \in \mathcal{K}_i[t]} w_{ij}[t](\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_i[t]). \tag{2.31}$$

To simplify the notation, define $\gamma_i[t] = \frac{\sigma_i[t]}{\|\boldsymbol{u}_i^p[t]\|}$. We define the control law $\boldsymbol{u}_i[t]$ as

$$\boldsymbol{u}_i[t] = \gamma_i[t] \sum_{j \in \mathcal{K}_i[t]} w_{ij}[t](\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_i[t]), \tag{2.32}$$

where for all time steps t and for all $i \in \mathcal{N}$, $w_{ij}[t] > 0$ and $\sum_{j \in \mathcal{K}_i[t]} w_{ij}[t] = 1$. For simplicity, we choose $w_{ij}[t] = \frac{1}{K_i}$. We point out that $0 < \gamma_i[t] \leq 1$. In the following subsection, we prove that under the effect of the control law (2.32) and Algorithm 2.4, normal behaving agents in the

discrete time setting are also guaranteed to achieve formation despite the presence of adversarial agents.

2.4.3.2 Convergence Analysis for Discrete-Time System

For our analysis, we need the following result:

Lemma 2.4. *Let $b[k] = kc^k b[0],\ k \in \mathbb{Z}_{\geq 0}$ be a series where $b[0] > 0$ and $0 < c < 1$. Then there exist positive constants α, β with $c < \beta < 1$ such that $\forall k \in \mathbb{Z}_{\geq 0}$,*

$$b[k] = kc^k b[0] \leq \alpha \beta^k. \tag{2.33}$$

Proof. It can be readily verified that for any $c < \beta < 1$ and $\alpha \geq \frac{b[0]}{e \log \frac{\beta}{c}}$, the inequality (2.33) holds for all $k \geq 0$. □

First, consider the normal agents in the set $\mathcal{S}_1$:

Lemma 2.5. *Consider a digraph $\mathcal{D}$ which is an RDAG with parameter $3F+1$, where $\mathcal{S}_0 = \mathcal{L}$ and $\mathcal{A}$ is an F-total set. For every normal agent $i \in \mathcal{S}_1$, $\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|$ converges to zero exponentially.*

Proof. For the worst case, assume there are F adversarial agents. Consider any normal agent $i \in \mathcal{S}_1$. Since all of its in-neighbours are from $\mathcal{S}_0$, we have that $\mathcal{V}_i \subset \mathcal{L}$ and for all $k \in \mathcal{V}_i \cap \mathcal{N}$, $\boldsymbol{\tau}_k = \boldsymbol{\tau}_L$. By definition of an RDAG, $|\mathcal{V}_i| \geq 3F+1$ which implies $K_i \geq 2F+1$ and $|\mathcal{K}_i^{\mathcal{N}}| \geq F+1$. For the worst case, suppose that $\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_j[t]\| \leq \|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|\ \forall j \in \mathcal{A}_i$ so that none of the adversarial agents are filtered out. This implies that $|\mathcal{K}_i^{\mathcal{A}}| = F$ and $|\mathcal{K}_i^{\mathcal{N}}| = K_i - F$. From the closed loop dynamics, we obtain:

$$\boldsymbol{\tau}_i[t+1] - \boldsymbol{\tau}_L = \boldsymbol{\tau}_i[t] + \sum_{j \in \mathcal{K}_i} \gamma_i w_{ij} (\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_i[t]) - \boldsymbol{\tau}_L.$$

Noting that $\mathcal{K}_i \subset \mathcal{L}$, after some manipulation we obtain:

$$\boldsymbol{\tau}_i[t+1] - \boldsymbol{\tau}_L = (1 - \gamma_i \frac{K_i - F}{K_i})(\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L) + \sum_{j \in \mathcal{K}_i^{\mathcal{A}}} \gamma_i w_{ij} (\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_i[t]). \tag{2.34}$$

Since $\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_j[t]\| \leq \|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L[t]\|$ for all $j \in \mathcal{K}_i^{\mathcal{A}}$, we have $\|\sum_{j \in \mathcal{K}_i^{\mathcal{A}}} w_{ij}(\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_i[t])\| \leq \frac{F}{|\mathcal{K}_i|} \|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|$. Hence, we obtain the bound on $\|\boldsymbol{\tau}_i[t+1] - \boldsymbol{\tau}_L\|$ as:

$$\|\boldsymbol{\tau}_i[t+1] - \boldsymbol{\tau}_L\| \leq (1 - \gamma_i \frac{K_i - 2F}{K_i}) \|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|. \tag{2.35}$$

Let $\gamma_i^* = \min_k \gamma_i[k] > 0$. Since $1 - \gamma_i \frac{K_i - 2F}{K_i} \leq 1 - \gamma_i^* \frac{K_i - 2F}{K_i} < 1$, define $c = 1 - \gamma_i^* \frac{K_i - 2F}{K_i}$, so that we have $\|\boldsymbol{\tau}_i[t+1] - \boldsymbol{\tau}_L\| \leq c\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|$, i.e. $\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|$ is an exponentially converging sequence. □

For $i \in \mathcal{S}_p$ where $p \geq 2$, we know that there are at most F adversarial agents in $\mathcal{K}_i$. Note that by definition of the network RDAG, all agents in $\mathcal{K}_i$ are from $\bigcup_{j=0}^{p-1} \mathcal{S}_j$. For the worst-case analysis, we assume there are F adversarial agents in $\mathcal{K}_i$ and all the normal agents in $\mathcal{K}_i$ are from $\mathcal{S}_{p-1}$. From the closed-loop dynamics of the agent i, we have:

$$\boldsymbol{\tau}_i[t+1] - \boldsymbol{\tau}_L = \boldsymbol{\tau}_i[t] + \sum_{j \in \mathcal{K}_i} \gamma_i w_{ij} (\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_i[t]) - \boldsymbol{\tau}_L,$$

which after some manipulation gives:

$$\begin{aligned} \boldsymbol{\tau}_i[t+1] - \boldsymbol{\tau}_L &= (1 - \gamma_i \frac{K_i - F}{K_i})(\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L) \\ &+ \sum_{j \in \mathcal{K}_i^{\mathcal{N}}} \gamma_i w_{ij} (\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_L) + \sum_{j \in \mathcal{K}_i^{\mathcal{A}}} \gamma_i w_{ij} (\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_i[t]). \end{aligned} \tag{2.36}$$

Using the same logic as in Lemma 2.5, we assume for the worst case that $\forall j \in \mathcal{A}_i$, $\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_j[t]\| \leq \|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_k[t]\|$ for some $k \in \mathcal{K}_i^{\mathcal{N}}$. Using this, the fact that $|\mathcal{K}_i^{\mathcal{A}}| = F$, we obtain:

$$\begin{aligned} \| \sum_{j \in \mathcal{K}_i^{\mathcal{N}}} w_{ij} (\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_L) \| &\leq \frac{|\mathcal{K}_i| - F}{|\mathcal{K}_i|} \|\boldsymbol{\tau}_k[t] - \boldsymbol{\tau}_L\|, \\ \| \sum_{j \in \mathcal{K}_i^{\mathcal{A}}} w_{ij} (\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_i[t]) \| &\leq \frac{F}{|\mathcal{K}_i|} \|\boldsymbol{\tau}_k[t] - \boldsymbol{\tau}_i\|. \end{aligned}$$

We can bound $\|\boldsymbol{\tau}_k - \boldsymbol{\tau}_i\| \leq \|\boldsymbol{\tau}_k - \boldsymbol{\tau}_L\| + \|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|$ to obtain:

$$\|\boldsymbol{\tau}_i[t+1] - \boldsymbol{\tau}_L\| \leq c\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\| + \|\boldsymbol{\tau}_k - \boldsymbol{\tau}_L\|, \tag{2.37}$$

where $c = 1 - \gamma_i^* \frac{K_i - 2F}{K_i} < 1$ where γ_i^* is defined as in Lemma 2.5. Inequality (2.37) is true for every normal agent in $\mathcal{S}_p$ with $p \geq 2$. Using this observation, we next consider the case of agents in set $\mathcal{S}_2$:

Lemma 2.6. *Consider a digraph $\mathcal{D}$ which is an RDAG with parameter $3F+1$, where $\mathcal{S}_0 = \mathcal{L}$ and $\mathcal{A}$ is an F-total set. For every normal agent $i \in \mathcal{S}_2$, $\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|$ converges to zero exponentially.*

Proof. Define $a[t] = \|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|$, $b_k[t] = \|\boldsymbol{\tau}_k[t] - \boldsymbol{\tau}_L\|$ so that (2.37) can be written as $a[t+1] \leq$

$ca[t] + b_k[t]$:

$$a[t+1] \leq c^{t+1}a[0] + \sum_{i=0}^{t} c^{t-i} b_k[i]. \tag{2.38}$$

Now, $b_k[i]$ represents the norm $\|\boldsymbol{\tau}_k[i] - \boldsymbol{\tau}_L\|$ of a normal agent $k \in \mathcal{S}_1$, which can be bounded as $b_k[i] \leq c_k^i b_k[0]$ as per (2.35) where $c_k = 1 - \gamma_k^* \frac{R_k - 2F}{R_k} < 1$. For the sake of brevity, let $a_0 = a[0]$, $b_{k0} = b_k[0]$. Using this, we obtain:

$$a[t+1] \leq c^{t+1}a_0 + \sum_{i=0}^{t} c^{t-i} c_k^i b_{k0}$$

Define $b_0^* = \max_{k \in \mathcal{K}_i^{\mathcal{N}}} b_{k0}$, $c^* = \max_{k \in \mathcal{K}_i^{\mathcal{N}}} c_k$, and $\tilde{c} = \max\{c, c^*\}$, so that

$$a[t+1] \leq \tilde{c}^{t+1}a_0 + \sum_{i=0}^{t} \tilde{c}^t b_0^* = \tilde{c}^{t+1}a_0 + (t+1)\tilde{c}^t b_0^*.$$

Using this and Lemma 2.4, i.e., $k\tilde{c}^t b_0^* \leq \alpha\beta^t$, we have that:

$$a[t+1] \leq \tilde{c}^t(ca_0 + b_0^*) + t\tilde{c}^t b_0^* \leq \tilde{c}^t(ca_0 + b_0^*) + \alpha\beta^t,$$

where $\alpha > 0$ and $\tilde{c} < \beta < 1$. Now, since $\tilde{c} < \beta$, we have:

$$a[t+1] \leq \tilde{c}^t(ca_0 + b[0]) + \alpha\beta^t \leq \beta^t(ca_0 + b[0] + \alpha).$$

As $\beta < 1$, a_t converges to zero exponentially, i.e., for a normal agent $i \in \mathcal{S}_2$, $\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|$ converges to zero exponentially. □

Note that this result can be interpreted as follows: $\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|$ for $i \in \mathcal{N}$ converges to zero exponentially if $\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_L\|$ converges to zero exponentially for all its normal in-neighbours $j \in \mathcal{K}_i \cap \mathcal{N}$. Using this, we can state the following result for all normal behaving agents:

Theorem 2.4. *Consider a digraph $\mathcal{D}$ which is an RDAG with parameter $3F+1$, where $\mathcal{S}_0 = \mathcal{L}$ and $\mathcal{A}$ is an F-total set. Under the closed loop dynamics (2.29)-(2.32), $\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|$ converges to zero exponentially for all agents $i \in \mathcal{N}$.*

Proof. We have proven this result for agents in $\mathcal{S}_1$ and $\mathcal{S}_2$ in Lemmas 2.5 and 2.6. We now consider any node $i \in \mathcal{S}_p$ for arbitrary p. Observe that every agent $i \in \mathcal{S}_p$ satisfies the equation (2.38), where $a[t]$ represents the norm $\|\boldsymbol{\tau}_i[t] - \boldsymbol{\tau}_L\|$ and $b_j[t] = \|\boldsymbol{\tau}_j[t] - \boldsymbol{\tau}_L\|$, where $j \in \bigcup_{l=0}^{p-1} \mathcal{S}_l$. From Lemma

2.6, we have that $\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|$ for normal agents in $\mathcal{S}_2$ converges exponentially to zero. Hence, it follows that for each normal agent in $\mathcal{S}_3$, $\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|$ converges to zero exponentially since all of its normal behaving agents are from the set $\bigcup_{l=0}^{2} \mathcal{S}_l$. Repeating this logic shows that for each normal agent $i \in \mathcal{S}_p$, $\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_L\|$ converges exponentially to zero for each $p \geq 1$. □

2.4.4 Simulations

To demonstrate the methods presented in this section, we consider an RDAG of 80 agents with parameter $r = 16$ and $F = 5$. There are 5 sub-levels, $\mathcal{S}_l$ with $|\mathcal{S}_l| = 16$ for $l \in \{0, 1, 2, 3, 4\}$. The set $\mathcal{S}_0$ is composed entirely of agents designated to behave as leaders. In the simulation, $\mathcal{A}$ is a 5-local model with 5 agents in each of the levels $\mathcal{S}_l$ becoming adversarial (including in $\mathcal{S}_0$). The simulation treats a worst-case scenario in the sense that each agent $i \in \mathcal{S}_l,\ l \geq 1$, has in-neighbours only in $\mathcal{S}_{l-1}$ and no leader in-neighbors. The agents have states in $\mathbb{R}^2$. The vector $\boldsymbol{\xi}$ specifies the formation as points on circle of radius 10 m centered at $\begin{bmatrix} 0 & 10 \end{bmatrix}^T$. The vectors $\boldsymbol{p}_i(0),\ i \in \mathcal{L}$ are chosen such that $\boldsymbol{\tau}_i(0)$ is at the origin for all $i \in \mathcal{L}$. The vectors $\boldsymbol{p}_j(0)$ for all other agents $j \in (\mathcal{V}_i \backslash \mathcal{L})$ are initialized such that their $\boldsymbol{\tau}_j(0)$ values are randomly initialized values. We choose the maximum allowed speed of the agents as $u_M = 1$. These conditions are used for both the continuous and discrete time simulations. For the continuous time case, the control parameter α is chosen as $\alpha = 0.8$.

Figure 2.10 shows a plot of $\|\boldsymbol{\tau}_i(t) - \boldsymbol{\tau}_L\|$ versus time for a subset of the normal agents. It is clear that all the normal agents converge to the point where their $\boldsymbol{\tau}$ values are same as those of leader in finite time. Figure 2.11 shows the path $\boldsymbol{p}_i(t) = \begin{bmatrix} x_i(t) & y_i(t) \end{bmatrix}^T$ of all the agents and a subset of the adversarial agents. The paths of the adversarial agents are chosen to as random walks and are not shown in the figure. In Figure 2.11 and Figure 2.14, it can be noted that while some normal agents (belonging to set $\mathcal{S}_1$ move directly towards their desired locations, other normal agents first move away from their desired locations. This is in agreement with our analysis; malicious agents are able to exert a bounded influence on normal agents in $\mathcal{S}_l$, $l \geq 2$ which do not have any leaders as in-neighbors, while convergence is still guaranteed in a finite time period.

For the case of the discrete system, Figure 2.13 shows the variation of $\|\boldsymbol{\tau}_i[k] - \boldsymbol{\tau}_L\|$ with number of steps. Figure 2.14 shows the path $\boldsymbol{p}_i[k] = \begin{bmatrix} x_i[k] & y_i[k] \end{bmatrix}^T$ of the agents. From both the figures, it is clear that despite the 5-local adversarial model, each normal agent achieves the desired formation.

From Figures 2.10 and 2.13, it can be seen that agents in $\mathcal{S}_l$ converge before agents in $\mathcal{S}_{l+1}$, which is consistent with our analysis for this particular worst-case scenario.

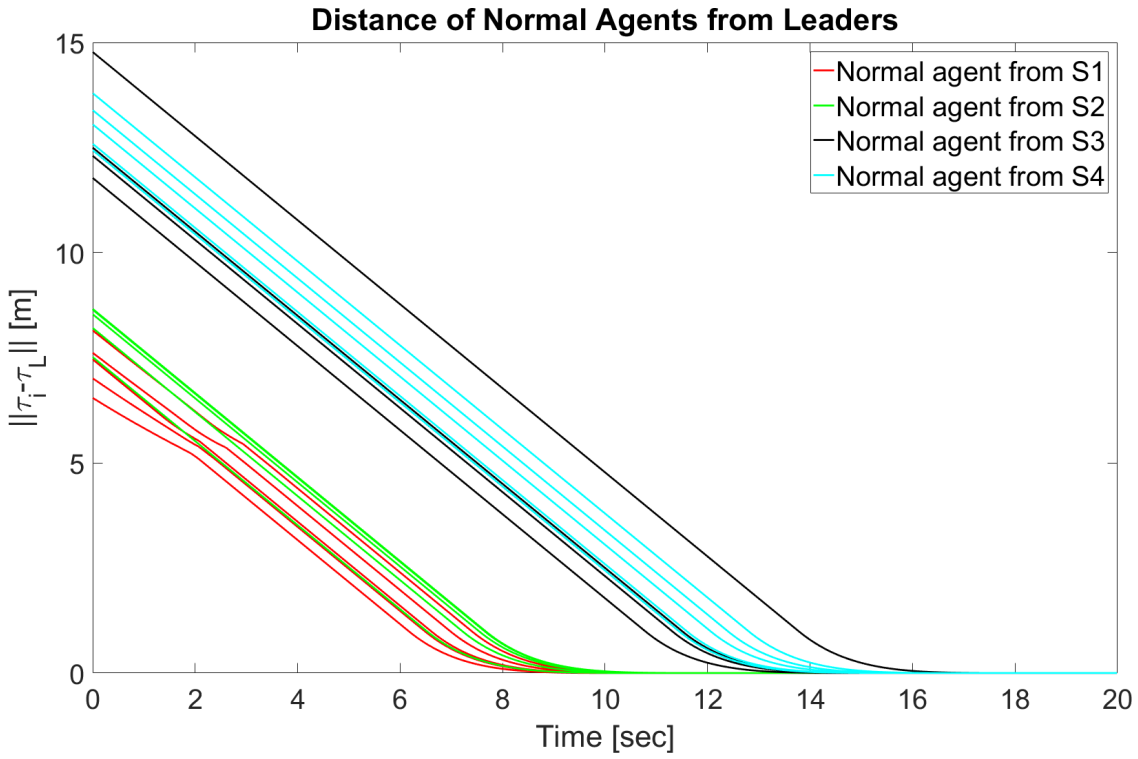


Figure 2.10: Norm $\|\boldsymbol{\tau}_i(t) - \boldsymbol{\tau}_L\|$ of a subset of the normal agents in the continuous time case. For sake of clarity, only a few normal nodes from each set $\mathcal{S}_p$ are shown.

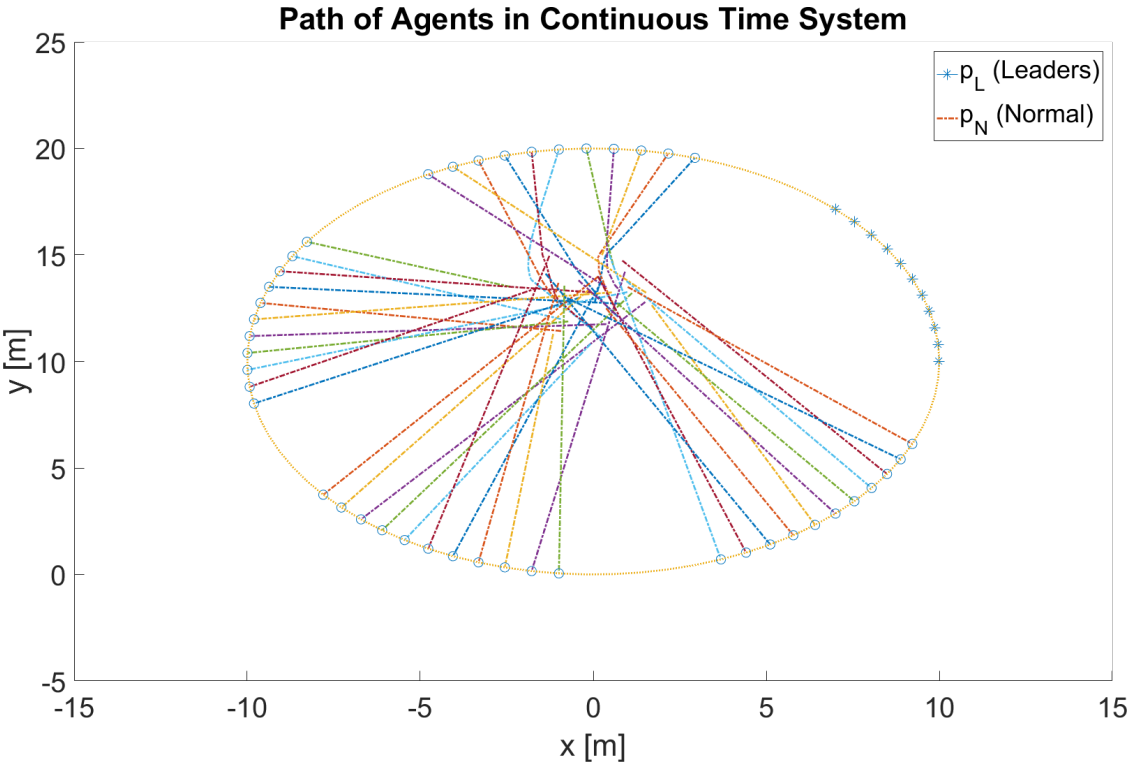


Figure 2.11: Path of the agents in the continuous time case. All normal and adversarial agents start from the centre of the circle marked by red dots. The leaders are denoted by the star points p_L and the non-adversarial agents are denoted by p_N.

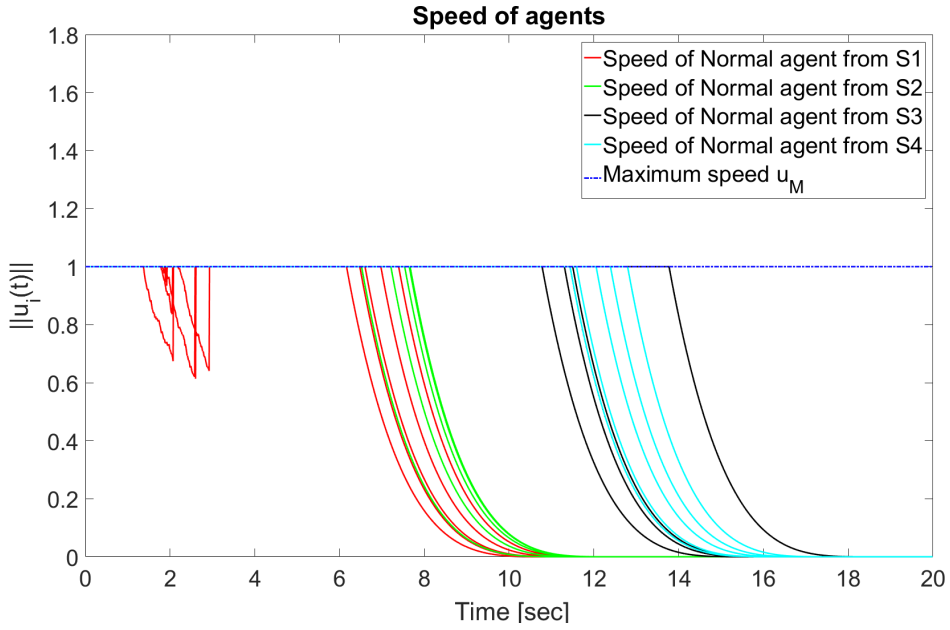


Figure 2.12: Norm $\|\boldsymbol{u}_i(t)\|$ of a subset of the normal agents in the continuous-time case, demonstrating that their input magnitudes never exceed the bound $u_M = 1$. The rest of the network is not shown for sake of clarity.

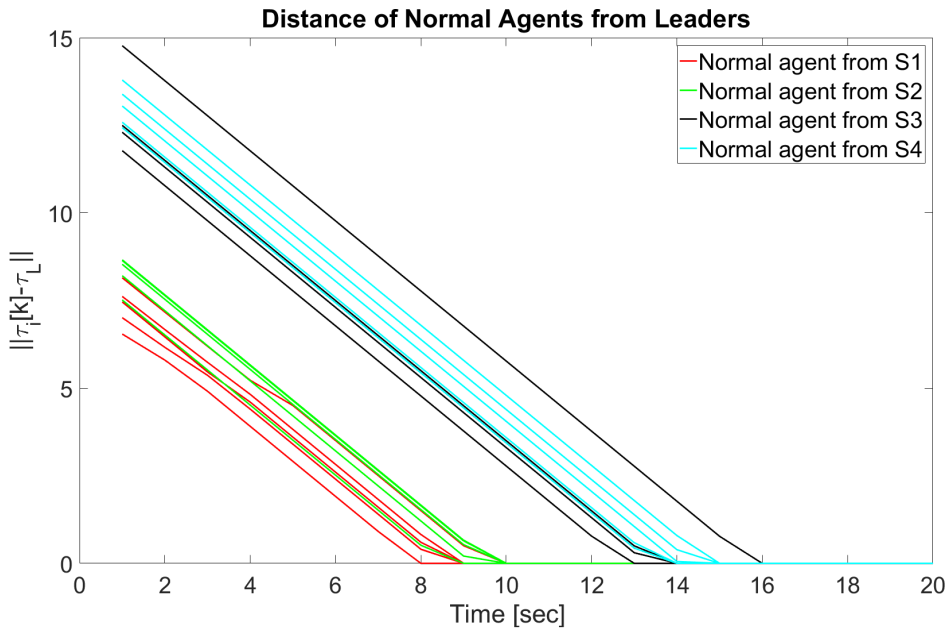


Figure 2.13: *Discrete Time*: Norm of formational position differences $\|\boldsymbol{\tau}_i[k] - \boldsymbol{\tau}_L\|$ of a subset of the normal agents in the discrete time case. For sake of clarity, only a few normal nodes from each set $\mathcal{S}_p$ are shown.

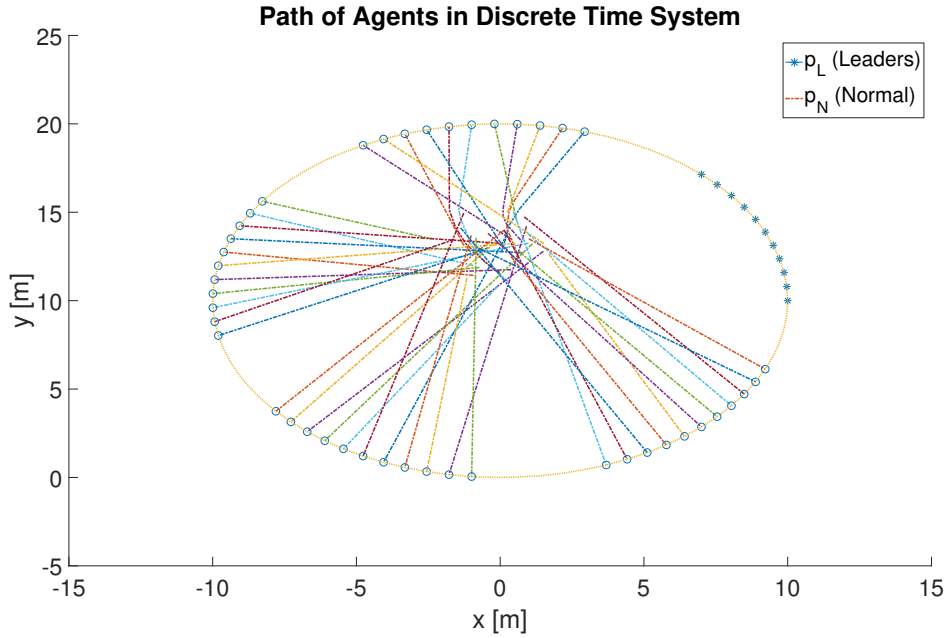


Figure 2.14: *Discrete Time:* Path of the agents in the discrete time case. The leaders are denoted by the star points p_L and the non-adversarial agents are denoted by p_N.

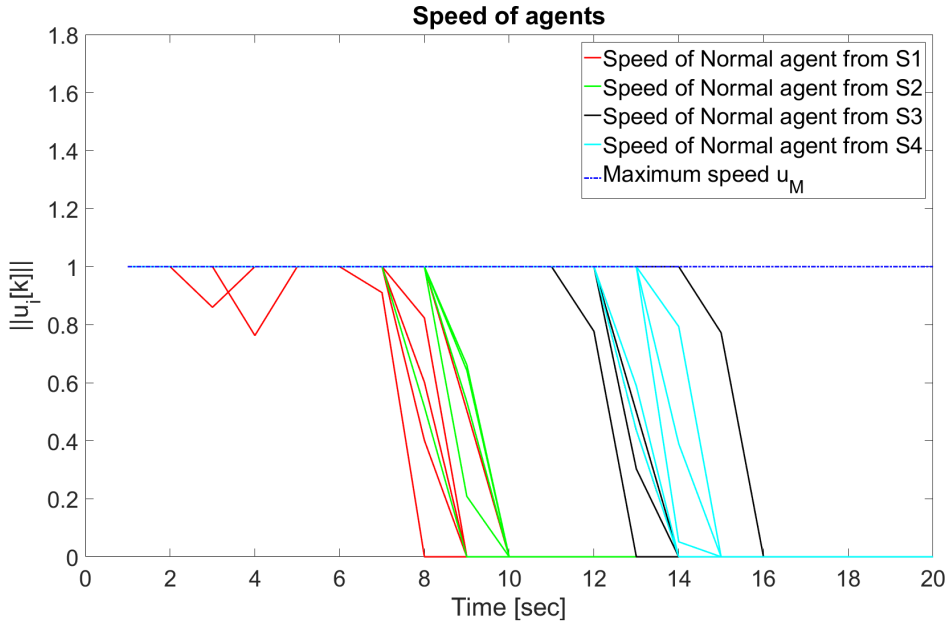


Figure 2.15: *Discrete Time:* Norm $\|\boldsymbol{u}_i[k]\|$ of a subset of the normal agents in the discrete time case. Again, the magnitude of each agents' control input never exceeds the bound $u_M = 1$ and goes to zero as the agents converge to formation.

2.5 Resilient Finite-Time Consensus: A Discontinuous Systems Perspective

As discussed in the Introduction to this chapter, there are few works in prior literature that consider resilient leaderless consensus in the continuous-time domain. One of the challenges of considering the continuous-time domain is proving existence of solutions to the resulting differential equations describing the system dynamics. The earlier studies that do consider the continuous-time domain make the assumption that adversarial signals sent to other agents are continuous. Some works model the system as having switched system dynamics and make the assumption that there exists a minimum dwell time between switching instances. This section relaxes these prior assumptions and presents a novel class of controllers that guarantee resilient leaderless consensus in finite time under an F-local or F-total adversarial model. No minimum dwell time to system dynamics is assumed, and adversarial signals are only assumed to be Lebesgue measurable rather than continuous. Discontinuous systems theory is used to rigorously demonstrate the finite-time convergence of normally-behaving agents to a common consensus value in the convex hull of initial normal agents' states.

2.5.1 Problem Formulation

Consider a network of n agents with $n \geq 2$ whose communication structure is modeled by the digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$. Without loss of generality we assume an initial time of $t_0 = 0$. Each agent i has a scalar state $x_i : \mathbb{R} \to \mathbb{R}$ and continuous-time first-order dynamics

$$\dot{x}_i(t) = u_i(t) \tag{2.39}$$

where the form of $u_i(t)$ will be given in Algorithm 2.5. At all times $t \geq 0$ each agent i is able to send a signal to its out-neighbors containing a function of its state $g(x_i(t))$, where $g : \mathbb{R} \to \mathbb{R}$ is a strictly increasing function with domain equal to $\mathbb{R}$. The function $g(\cdot)$ is the same for all agents and is not required to be continuous.

Definition 2.14. *The notation $g(x_j^i(t))$, $x_j^i : \mathbb{R} \to \mathbb{R}$, denotes the signal received by agent i from agent j at time t.*

A *normally-behaving* agent is defined as an agent i that sends the function of its true state value $g(x_i(t))$ to all of its out-neighbors and updates its state according to the *Finite-Time Resilient Consensus Protocol* (FTRC-P) defined in Algorithm 2.5. The set of all normal agents is denoted $\mathcal{N} \subset \mathcal{V}$.

We consider the presence of adversarial adversaries in this problem setting, which are defined as follows:

Definition 2.15. *An agent $k \in \mathcal{V}$ is called* adversarial *if at least one of the following conditions holds:*

- *There exists $t \geq t_0$ such that $u_k(t)$ is not equal to the input (2.40) defined by the FTRC Protocol in Algorithm 2.5.*
- *There exists $i \in \mathcal{V}_k^{out}$ and $t \geq t_0$ such that $g(x_k^i(t)) \neq g(x_k(t))$; i.e. agent k sends an out-neighbor a different value than its actual state value.*
- *There exists $i_1, i_2 \in \mathcal{V}_k^{out}$ and $t \geq t_0$ such that $g(x_k^{i_1}(t)) \neq g(x_k^{i_2}(t))$; i.e. agent k sends different values to different out-neighbors.*

The set of adversarial agents is denoted $\mathcal{A} \subset \mathcal{V}$.

Note that the definition of adversarial agents encompasses both *Byzantine* adversaries [83] and faulty agents. All nodes in $\mathcal{V}$ are either normal or adversarial; i.e. $\mathcal{A} \cap \mathcal{N} = \emptyset$ and $\mathcal{A} \cup \mathcal{N} = \mathcal{V}$. The only assumption made on the signals $g(x_k^i(\cdot))$ originating from the adversaries is the following condition:

Algorithm 2.5 FTRC PROTOCOL (FTRC-P):

1. At time t, each normal agent i receives values $g(x_j^i(t))$ from its in-neighbors $j \in \mathcal{V}_i(t)$ and forms a sorted list.

2. If there are less than F values strictly larger than i's own value $g(x_i(t))$, then i removes all values that are strictly larger than its own. Otherwise i removes precisely the largest F values in the sorted list.

3. In addition, if there are less than F values strictly smaller than i's own value $g(x_i(t))$, then i removes all values that are strictly smaller than its own. Otherwise i removes precisely the smallest F values in the sorted list.

4. Let $\mathcal{R}_i(t)$ denote the set of agents whose values are removed by agent i in steps 2) and 3) at time t. Agent i applies the following update:

$$u_i(t) = \alpha \operatorname{sign}\left(\sum_{\mathcal{J}_i \setminus \mathcal{R}_i[t]} g(x_j^i(t)) - g(x_i(t)) \right) \tag{2.40}$$

where $\alpha > 0$ and $g : \mathbb{R} \to \mathbb{R}$ is defined in Section 2.5.1. Note that since $i \in \mathcal{J}_i$ by definition and agent i never filters out the function of its own state $g(x_i(t))$, (2.40) is always well-defined.

Assumption 2.2. *For any $k \in \mathcal{A}$ and $i \in \mathcal{N}$, the function $g \circ x_k^i$ is Lebesgue measurable.*

Remark 2.2. *Assumption 2.2 widens the class of adversarial signals that can be considered as compared to prior work. Prior work typically assumes that adversarial signals are continuous [83, 84] or have a finite number of discontinuities in any compact interval [86, 88]. Under Assumption 2.2 however, the techniques in this section consider adversarial signals that may be discontinuous and have possibly infinite discontinuities in a finite interval.*

Naturally, Assumption 2.2 raises the question of what happens if one or more of the adversarial signals are not Lebesgue measurable. The answer to this question hinges upon whether there exist subsets of $\mathbb{R}$ that are not Lebesgue measurable, which in itself depends on which core axioms of mathematics are assumed to hold (e.g. the axiom of choice). Further discussion on this point is given in Section 2.5.5.

To quantify the number and distribution of adversarial agents in the network, we will use the F-local model described in Definition 2.2. Recall that under the F-local model no agents are assumed to be invulnerable to attacks or faults.

The objective of the normal agents is to achieve consensus in their state values despite the presence of an F-local adversarial set $\mathcal{A}$. We ultimately are not concerned with the trajectories of the adversarial agents' states–we are only concerned with ensuring that the actions of the adversarial agents do not prevent the consensus of the normal agents. In this light, we define the vector of normal agents' states as follows:

$$x_{\mathcal{N}}(t) = \begin{bmatrix} x_{\mathcal{N}_1}(t) \\ x_{\mathcal{N}_2}(t) \\ \vdots \\ x_{\mathcal{N}_{|\mathcal{N}|}}(t) \end{bmatrix}, \; x_{\mathcal{N}}(t) \in \mathbb{R}^{|\mathcal{N}|}, \tag{2.41}$$

where $\mathcal{N}_j$ is the index of the jth agent in $\mathcal{N}$ according to any arbitrary fixed ordering of $\mathcal{N}$, with $\{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_{|\mathcal{N}|}\} = \mathcal{N}$. To give a brief example, in a network of $n = 5$ agents with the normal agents being $\{2, 4, 5\}$, we have $\mathcal{N}_1 = 2$, $\mathcal{N}_2 = 4$, and $\mathcal{N}_3 = 5$ with $x_{\mathcal{N}}(t) = \begin{bmatrix} x_2(t) & x_4(t) & x_5(t) \end{bmatrix}^T$. Consensus of the normal agents is achieved when $x_{\mathcal{N}}(t) \in \text{span}(\mathbf{1})$. However, note by the form of (2.40) that each $u_i(\cdot)$ is a function of both signals from normal agents *and* signals from any adversarial agents that are in-neighbors of i. For all $i \in \mathcal{N}$, the vector of adversarial signals sent to i at time t is denoted $x_{\mathcal{A}}^i \in \mathbb{R}^{|\mathcal{V}_i \cap \mathcal{A}|}$. The dynamics of the normal agents are therefore written as

follows:

$$\begin{aligned}\dot{x}_{\mathcal{N}}(t) &= \begin{bmatrix} u_{\mathcal{N}_1}(x_{\mathcal{N}}(t), x_{\mathcal{A}}^{\mathcal{N}_1}(t))) \\ u_{\mathcal{N}_2}(x_{\mathcal{N}}(t), x_{\mathcal{A}}^{\mathcal{N}_2}(t)) \\ \vdots \\ u_{\mathcal{N}_{|\mathcal{N}|}}(x_{\mathcal{N}}(t), x_{\mathcal{A}}^{\mathcal{N}_{|\mathcal{N}|}}(t)) \end{bmatrix}, \\ &= f_{\mathcal{N}}(x_{\mathcal{N}}(t), x_{\mathcal{A}}^{\mathcal{N}}(t)), \end{aligned} \tag{2.42}$$

where $\{\mathcal{N}_1, \ldots, \mathcal{N}_{|\mathcal{N}|}\} = \mathcal{N}$ and

$$x_{\mathcal{A}}^{\mathcal{N}}(t) = \left[(x_{\mathcal{A}}^{\mathcal{N}_1}(t))^T \ \cdots \ (x_{\mathcal{A}}^{\mathcal{N}_{|\mathcal{N}|}}(t))^T\right]^T \in \mathbb{R}^{\sum_{\mathcal{N}_j \in \mathcal{N}} |\mathcal{V}_{\mathcal{N}_j} \cap \mathcal{A}|} \tag{2.43}$$

is the vector of all adversarial signals at time t. By definition, the adversarial signals are arbitrary functions of time and in general will not be functions of the normal agent state vector $x_{\mathcal{N}}(t)$. The adversarial signals in each vector $x_{\mathcal{A}}^{\mathcal{N}_i}$ can therefore be viewed as arbitrary, possibly discontinuous inputs to the system of normal agents.

The objective of the normally-behaving agents is to achieve *Finite-Time Resilient Consensus* (FTRC). To define FTRC, we first introduce the following functions:

$$\begin{aligned} M(x_{\mathcal{N}}) &= \max_{i \in \mathcal{N}} x_i = \max_{j \in \{1,\ldots,|\mathcal{N}|\}} (e^j)^T x_{\mathcal{N}} \\ m(x_{\mathcal{N}}) &= \min_{i \in \mathcal{N}} x_i = \min_{j \in \{1,\ldots,|\mathcal{N}|\}} (e^j)^T x_{\mathcal{N}} \\ V(x_{\mathcal{N}}) &= M(x_{\mathcal{N}}) - m(x_{\mathcal{N}}) \end{aligned} \tag{2.44}$$

We also define the following sets to describe the agents with state values equal to $M(x_{\mathcal{N}})$ or $m(x_{\mathcal{N}})$:

$$\begin{aligned} S_M &= \{i \in \mathcal{N} : x_i = M(x_{\mathcal{N}})\} \\ S_m &= \{i \in \mathcal{N} : x_i = m(x_{\mathcal{N}})\} \end{aligned} \tag{2.45}$$

Definition 2.16. *The normal agents $i \in \mathcal{N}$ achieve Finite-Time Resilient Consensus (FTRC) if all of the following conditions hold:*

(i) $x_i(t) \in [m(x_{\mathcal{N}}(0)), M(x_{\mathcal{N}}(0))]$ *for all* $t \geq 0$ *and for all* $i \in \mathcal{N}$.

(ii) $\exists T : \mathbb{R}^{|\mathcal{N}|} \to \mathbb{R}_+$ *such that* $V(x_{\mathcal{N}}(t)) = 0$ *for all* $t \geq T(x_{\mathcal{N}}(0))$. *Equivalently,* $x_{\mathcal{N}}(t) \in$ *span*$(\mathbf{1})$ *for all* $t \geq T(x_{\mathcal{N}}(0))$.

Remark 2.3. *The notion of FTRC is based on the notion of* Continuous-Time Resilient Asymptotic Consensus *(CTRAC) in [83], but imposes the stricter requirement that $V(x_{\mathcal{N}}(t))$ converges* exactly *to zero in a finite amount of time and remains there for all future time.*

Problem 2.2. *Determine conditions under which FTRC is achieved by the normal agents $i \in \mathcal{N}$ in the presence of a adversarial subset of agents $\mathcal{A} \subset \mathcal{V}$.*

2.5.2 Justification for Discontinuous Systems Approach

This section uses discontinuous systems theory and nonsmooth analysis to prove that a network of agents applying the FTRC-P achieves FTRC. There are two reasons for such an approach. First, the form of $u_i(\cdot)$ in (2.40) implies that the right hand side (RHS) of (2.42) is discontinuous. Note that we cannot simply assume a minimum "dwell time" and treat the system as a switching system, since cleverly designed adversarial signals may induce an arbitrary number of discontinuities in any given time interval. To give a pathological example, suppose an agent $i \in \mathcal{N}$ receives an adversarial signal $x_k^i(t)$ from $k \in \mathcal{A}$ defined as follows:

$$x_k^i(t) = \begin{cases} a \in \mathbb{R} & \text{if } t \in \mathbb{I}, \\ b \in \mathbb{R},\ b \neq a & \text{if } t \in \mathbb{Q} \end{cases} \tag{2.46}$$

where a and b are chosen appropriately, and $\mathbb{I}$ and $\mathbb{Q}$ represent the sets of irrational and rational numbers in $\mathbb{R}$, respectively. Both $\mathbb{I}$ and $\mathbb{Q}$ are dense in $\mathbb{R}$, implying that no positive minimum dwell time can be assumed for the system. The second reason for a discontinuous systems approach is that the Lyapunov-like candidate $V(x_{\mathcal{N}}(t))$ from (2.44) which will be used for convergence analysis is nonsmooth in general. Discontinuous systems theory allows for nonsmoothness and discontinuities to be addressed in a mathematically precise manner while solving Problem 2.2.

2.5.3 Review of Discontinuous Systems Theory

This subsection gives a brief overview of several fundamental concepts from discontinuous systems theory that are relevant to this paper. The reader is referred to [198, 205–207] for more detailed information.

A differential inclusion is a system with dynamics

$$\dot{x}(t) \in \mathcal{F}(t, x(t)), \tag{2.47}$$

where $x : \mathbb{R} \to \mathbb{R}^d$ and $\mathcal{F} : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$, where $\mathcal{P}(\mathbb{R}^d)$ denotes the power set of $\mathbb{R}^d$ as defined in Section 1.5. The set-valued map $\mathcal{F}$ indicates that at every time t there can be multiple possible

evolutions of the system state rather than just one. A Caratheodory solution of (2.47) defined on $[t_0, t_1] \subset [0, \infty)$ is an absolutely continuous function $x : [t_0, t_1] \to \mathbb{R}^d$ such that $\dot{x}(t) \in \mathcal{F}(t, x(t)$ for almost all $t \in [t_0, t_1]$ in the sense of Lebesgue measure. Existence of Caratheodory solutions to (2.47) is guaranteed by the following proposition:

Proposition 2.1 ([198]). *Suppose the set-valued map $\mathcal{F} : [0, \infty) \times \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ is locally bounded and takes nonempty, compact and convex values. Assume that, for each $t \in \mathbb{R}$, the set-valued map $x \mapsto \mathcal{F}(t, x)$ is upper semicontinuous, and for each $x \in \mathbb{R}^d$, the set-valued map $t \mapsto \mathcal{F}(t, x)$ is measurable. Then, for all $(t_0, x_0) \in [0, \infty) \times \mathbb{R}^d$ there exists a Caratheodory solution of (2.47) with initial condition $x(t_0) = x_0$.*

For convenience, the definitions of locally bounded, upper semicontinuity, and local Lipschitzness are given below.

Definition 2.17 (Locally bounded [198]). *The set-valued map $\mathcal{F} : [t_0, \infty) \times \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ is locally bounded at $(t, x) \in [t_0, \infty) \times \mathbb{R}^d$ if there exist $\epsilon, \delta > 0$ and an integrable function $m : [t, t + \delta] \to (0, \infty)$ such that $\|z\|_2 \leq m(s)$ for all $z \in \mathcal{F}(s, y)$, all $s \in [t, t + \delta]$, and all $y \in B(x, \epsilon)$ where $B(x, \epsilon)$ is the unit ball of radius ϵ centered at x.*

Definition 2.18 (Upper semicontinuity [198]). *The time-invariant set-valued map $\mathcal{F} : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ is upper semicontinuous at $x \in \mathbb{R}^d$ if for all $\epsilon > 0$ there exists $\delta > 0$ such that $\mathcal{F}(y) \subseteq \mathcal{F}(x) + B(0, \epsilon)$ for all $y \in B(x, \delta)$.*

Definition 2.19 ([198]). *The set-valued map $\mathcal{F} : [t_0, \infty) \times \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ is locally Lipschitz at $x \in \mathbb{R}^d$ if there exists $L(x), \epsilon > 0$ such that $\mathcal{F}(y) \subset \mathcal{F}(z) + L(x) \|y - z\|_2 \bar{B}(0, 1)$ for all $y, z \in B(x, \epsilon)$. Note that a set-valued map being locally Lipschitz implies that it is also upper semi-continuous [198].*

Existence intervals for Caratheodory solutions to (2.47) can be extended forward in time using the following result.

Theorem 2.5 ([208] Ch. 2 §7 Thm 2). *Let $\mathcal{F} : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ satisfy the hypotheses of Proposition 2.1 in a compact domain $D \subset \mathbb{R} \times \mathbb{R}^d$, and be upper semicontinuous in t and x on D. Then each solution of (2.47) with $\left[\begin{smallmatrix} t_0 \\ x(t_0) \end{smallmatrix}\right] \in D$ can be continued in time until $\left[\begin{smallmatrix} t \\ x(t) \end{smallmatrix}\right]$ reaches the boundary of D.*

Although there are multiple ways to define set-valued maps, the following method will be used in this paper.

Definition 2.20 ([198]). *Let $f : \mathbb{R}^d \times \mathcal{U} \to \mathbb{R}$, where $\mathcal{U} \subset \mathbb{R}^m$ is the set of allowable control inputs, and let $u : \mathbb{R} \to \mathcal{U}$ be a control signal. Consider the function $\dot{x}(t) = f(x(t), u(t)),\ u(t) \in \mathcal{U}$. The set-valued map $G[f] : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ is defined as*

$$G[f](x) \triangleq \{f(x, u) : u \in \mathcal{U}\}. \tag{2.48}$$

The notion of generalized gradient extends the notion of gradient to locally Lipschitz functions that may not be continuously differentiable everywhere.

Definition 2.21 (Generalized Gradient [205,206]). *Let $V : \mathbb{R}^d \to \mathbb{R}$ be a locally Lipschitz function [209, Sec. 3.1], and let $\Omega_V \subset \mathbb{R}^d$ denote the set of points where V fails to be differentiable,*[8] *and let $S \subset \mathbb{R}^d$ denote any other set of measure zero. The* generalized gradient $\partial V : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ *of V is defined as*

$$\partial V(x) = co\left\{\lim_{i\to\infty} \nabla V(x^i) : x^i \to x,\ x^i \notin \Omega_V \cup S\right\} \tag{2.49}$$

Computing generalized gradients can be difficult in general. However several useful results exist in the literature that facilitate this calculation, including the following one.

Proposition 2.2 ([198]). *For $k \in \{1, \dots, m\}$, let $g_k : \mathbb{R}^d \to \mathbb{R}$ be locally Lipschitz at $x \in \mathbb{R}^d$, and define the functions $g_{\max} : \mathbb{R}^d \to \mathbb{R}$ and $g_{\min} : \mathbb{R}^d \to \mathbb{R}$ as*

$$g_{\max}(y) \triangleq \max\{g_k(y) : k \in \{1, \dots, m\}\} \tag{2.50}$$
$$g_{\min}(y) \triangleq \min\{g_k(y) : k \in \{1, \dots, m\}\} \tag{2.51}$$

Then all of the following statements hold:

1. *$f_{\max}$ and $f_{\min}$ are locally Lipschitz at x*
2. *Let $I_{\max}(x)$ denote the set of indices k for which $g_k(x) = g_{\max}(x)$. Then the function $g_{\max}$ is locally Lipschitz at x, and*

$$\partial g_{\max} \subseteq co \bigcup \{\partial g_i(x) : i \in I_{\max}(x)\}. \tag{2.52}$$

Furthermore, if g_i is regular[9] *at x for all $i \in I_{\max}(x)$, then equality holds in (2.52) and $g_{\max}$ is regular at x.*

[8] Note that by Rademacher's Theorem, a locally Lipschitz function is differentiable almost everywhere in the sense of Lebesgue measure [205, Sec. 1.2].

[9] The precise definition of *regular functions* can be found in [205, Defn. 2.3.4] and [198]. Notably, all convex functions are regular [205, Prop. 2.3.6].

3. *Let $I_{\min}(x)$ denote the set of indices k for which $g_k(x) = g_{\min}(x)$. Then the function $g_{\min}$ is locally Lipschitz at x, and*

$$\partial g_{\min} \subseteq co \bigcup \{\partial g_i(x) : i \in I_{\min}(x)\}. \tag{2.53}$$

Furthermore, if $-g_i$ is regular at x for all $i \in I_{\min}(x)$, then equality holds in (2.52) and $-g_{\min}$ is regular at x.

The *set-valued Lie derivative* is used to analyze the stability of differential inclusions:

Definition 2.22 ([197, 198]). *Given a locally Lipschitz function $V : \mathbb{R}^d \to \mathbb{R}$ and a set-valued map $\mathcal{F} : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$, the* set-valued Lie derivative $\widetilde{\mathcal{L}}_{\mathcal{F}} V : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ *of V with respect to (w.r.t.) $\mathcal{F}$ at x is defined as*

$$\widetilde{\mathcal{L}}_{\mathcal{F}} V(x) = \{a \in \mathbb{R} : \exists v \in \mathcal{F}(x) \text{ such that } \zeta^T v = a \text{ for all } \zeta \in \partial V(x)\} \tag{2.54}$$

Given a locally Lipschitz and regular function f and a Caratheodory solution $x(t)$ of (2.47), the following result describes properties of the time derivative of the composition $f(x(t))$.

Proposition 2.3 ([197, 198]). *Let $x : [0, t_1] \to \mathbb{R}^d$ be a solution of the differential inclusion (2.47) with $\mathcal{F}(\cdot)$ satisfying the hypotheses of Proposition 2.1, and let $h : \mathbb{R}^d \to \mathbb{R}$ be locally Lipschitz and regular. Then the composition $t \mapsto h(x(t))$ is differentiable at almost all $t \in [t_0, t_1]$, and the derivative of $t \mapsto h(x(t))$ satisfies*

$$\frac{d}{dt}(h(x(t))) \in \widetilde{\mathcal{L}}_{\mathcal{F}} h(x(t)) \tag{2.55}$$

for almost every $t \in [0, t_1]$.

Lastly, the following result will be used to demonstrate finite-time convergence.

Theorem 2.6 ([197]). *Let $\mathcal{M} = span(\mathbf{1})$. Consider a scalar function $V(x) : \mathbb{R}^d \to \mathbb{R}$ with $V(x) = 0$ for all $x \in \mathcal{M}$ and $V(x) > 0$ for all $x \in \mathbb{R}^d \backslash \mathcal{M}$. Let $x : \mathbb{R} \to \mathbb{R}^d$ and $V(x(t))$ be absolutely continuous on $[t_0, \infty)$ with $d/dt(V(x(t))) \leq -\epsilon < 0$ almost everywhere on $\{t : x(t) \notin \mathcal{M}\}$. Then $V(x(t))$ converges to 0 in finite time, implying that $x(t)$ reaches the subspace $\mathcal{M}$ in finite time.*

2.5.4 Main Results

The first result of this section describes a differential inclusion for the total system in (2.42) under the controller (2.40) and demonstrates that it satisfies all the conditions of Proposition 2.1. This will guarantee existence of solutions despite the discontinuous nature of (2.40) and the possibly discontinuous nature of the adversarial signals.

Lemma 2.7. *Consider the system* (2.42) *where all normally-behaving agents apply the FTRC Protocol (Algorithm 2.5). Then the dynamics of the system* (2.42) *satisfy the differential inclusion*

$$\dot{x}_{\mathcal{N}}(t) \in G[f_{\mathcal{N}}](x_{\mathcal{N}}(t)), \tag{2.56}$$

where

$$G[f_{\mathcal{N}}](x_{\mathcal{N}}) = \overline{co}\left\{-\alpha\mathbf{1}, \alpha\mathbf{1}\right\}. \tag{2.57}$$

Furthermore, $G[f_{\mathcal{N}}](x_{\mathcal{N}})$ satisfies all the hypotheses of Proposition 2.1 and is locally Lipschitz for all $x_{\mathcal{N}} \in \mathbb{R}^{|\mathcal{N}|}$ and for all $t \geq 0$.

Proof. By the definition of the sign$(\cdot)$ function, observe that for all $i \in \mathcal{N}$ it holds that $u_i \in \{-\alpha, 0, \alpha\}$. Note that this holds for all possible adversarial signals $x_{\mathcal{A}}^{\mathcal{N}}$ defined in (2.43). Therefore $\dot{x}_i(t) \in [-\alpha, \alpha]$ for all $i \in \mathcal{N}$, implying that $\dot{x}_{\mathcal{N}}(t) \in \overline{\text{co}}\left\{-\alpha\mathbf{1}, \alpha\mathbf{1}\right\} = G[f_{\mathcal{N}}](x_{\mathcal{N}})$ for all $x_{\mathcal{N}} \in \mathbb{R}^{|\mathcal{N}|}$.

Next, we show that $G[f_{\mathcal{N}}](x_{\mathcal{N}})$ satisfies all the hypotheses of Proposition 2.1. Note that $G[f_{\mathcal{N}}](x_{\mathcal{N}})$ takes nonempty, compact, and convex values. Since $G[f_{\mathcal{N}}](x_{\mathcal{N}})$ is time-invariant and equal to the Cartesian product of intervals $[-\alpha, \alpha] \times \ldots \times [-\alpha, \alpha]$, it is measurable for all $x \in \mathbb{R}^{|\mathcal{N}|}$ and for all $t \geq 0$. To show local boundedness note that for all $x_{\mathcal{N}} \in \mathbb{R}^{|\mathcal{N}|}$, for all $t \geq 0$, and for all $v \in \mathcal{G}[f](x_{\mathcal{N}})$ we have $\|v\|_2 = \left(\sum_{i=1}^{|\mathcal{N}|} |v_i|^2\right)^{(1/2)} \leq \left(\sum_{i=1}^{|\mathcal{N}|} |\alpha|^2\right)^{(1/2)} = \sqrt{|\mathcal{N}|}\alpha$. Letting $\gamma(t) = \sqrt{|\mathcal{N}|}\alpha$, it follows that for all $(t, x) \in [0, \infty) \times \mathbb{R}^{|\mathcal{N}|}$ and for all $\epsilon, \delta > 0$ we have $\|v\|_2 \leq \gamma(s)$ for all $v \in \mathcal{G}[f](x_{\mathcal{N}})$, for all $s \in [t, t + \delta]$, and for all $y \in B(x, \epsilon)$.

Finally, $G[f_{\mathcal{N}}](x_{\mathcal{N}})$ can be shown to be locally Lipschitz by noting that since $G[f_{\mathcal{N}}](x_{\mathcal{N}})$ is constant for all $x \in \mathbb{R}^{|\mathcal{N}|}$, it holds that for all $x_{|\mathcal{N}|} \in \mathbb{R}^{\mathcal{N}}$ there exists $L > 0$, $\epsilon > 0$ such that $G[f_{\mathcal{N}}](y) = \overline{\text{co}}\left\{-\alpha\mathbf{1}, \alpha\mathbf{1}\right\} \subseteq \overline{\text{co}}\left\{-\alpha\mathbf{1}, \alpha\mathbf{1}\right\} + L\|y - z\|\bar{B}(0, 1) = G[f_{\mathcal{N}}](z) + L\|y - z\|\bar{B}(0, 1)$ for all $y, z \in B(x_{\mathcal{N}}, \epsilon)$. Since local Lipschitzness of $G[f_{\mathcal{N}}](x_{\mathcal{N}})$ implies upper semicontinuity of $G[f_{\mathcal{N}}](x_{\mathcal{N}})$ [198], $G[f_{\mathcal{N}}](x_{\mathcal{N}})$ therefore satisfies all the hypotheses of Proposition 2.1. □

We will next characterize the functions $M(\cdot)$, $m(\cdot)$, and $V(\cdot)$. These results will be necessary to demonstrate that FTRC is achieved by the system of normal agents.

Lemma 2.8. *Let the functions $M : \mathbb{R}^{|\mathcal{N}|} \to \mathbb{R}$, $m : \mathbb{R}^{|\mathcal{N}|} \to \mathbb{R}$, and $V : \mathbb{R}^{|\mathcal{N}|} \to \mathbb{R}$ be defined as in (2.44). Then $M(\cdot)$, $(-m(\cdot))$, and $V(\cdot)$ are all regular, locally Lipschitz, and absolutely continuous on $\mathbb{R}^{|\mathcal{N}|}$.*

Proof. Recall that e^i is the ith column of the $|\mathcal{N}| \times |\mathcal{N}|$ identity matrix. Observe that $M(x_\mathcal{N})$ is the pointwise maximum over the functions $(e^i)^T x_\mathcal{N}$ for $i \in \mathcal{N}$, which are all locally Lipschitz on $\mathbb{R}^{|\mathcal{N}|}$. By Proposition 2.2, $M(x_\mathcal{N})$ is therefore locally Lipschitz on $\mathbb{R}^{|\mathcal{N}|}$. In addition, each function $(e^i)^T x_\mathcal{N}$ is affine, and therefore convex and regular on $\mathbb{R}^{|\mathcal{N}|}$. Since for all possible indices $i \in \{1, \dots, \mathcal{N}\}$ the functions $(e^i)^T x_\mathcal{N}$ are regular on $\mathbb{R}^{|\mathcal{N}|}$, by Proposition 2.2 $M(x_\mathcal{N})$ is regular on $\mathbb{R}^{|\mathcal{N}|}$.

Similarly, $m(x_\mathcal{N})$ is the pointwise minimum over the functions $(e^i)^T x_\mathcal{N}$ for $i \in \{1, \dots, \mathcal{N}\}$, which are all locally Lipschitz on $\mathbb{R}^{|\mathcal{N}|}$. By Proposition 2.2, $m(x_\mathcal{N})$ is therefore locally Lipschitz on $\mathbb{R}^{|\mathcal{N}|}$. Since each $(e^i)^T x_\mathcal{N}$ is affine, each function $-(e^i)^T x_\mathcal{N}$ is also affine and therefore convex and regular on $\mathbb{R}^{|\mathcal{N}|}$. Therefore by Proposition 2.2 the function $(-m(x_\mathcal{N}))$ is regular on $\mathbb{R}^{|\mathcal{N}|}$.

Since $V(x_\mathcal{N})$ is equal to the sum of two locally Lipschitz and regular functions, it holds that $V(x_\mathcal{N})$ is also locally Lipschitz and regular [198]. Finally, every locally Lipschitz function on $\mathbb{R}^{|\mathcal{N}|}$ is absolutely continuous on $\mathbb{R}^{|\mathcal{N}|}$ [198], which implies that $M(x_\mathcal{N})$, $(-m(x_\mathcal{N}))$, and $V(x_\mathcal{N})$ are all absolutely continuous. □

Remark 2.4. *We primarily consider the function $(-m(\cdot))$ rather than $m(\cdot)$ because Proposition 2.2 only allows us to prove the regularity of $(-m(\cdot))$. The property of regularity is required by Proposition 2.3 to prove that the time derivative of $(-m(x_\mathcal{N}(t))$ exists for almost all $t \in [0, t_1)$, which will be shown later in Theorem 2.7.*

We next derive the Clarke generalized gradients for $M(\cdot)$ and $m(\cdot)$, which are defined in (2.44).

Lemma 2.9. *Let $M : \mathbb{R}^{|\mathcal{N}|} \to \mathbb{R}$ and $m : \mathbb{R}^{|\mathcal{N}|} \to \mathbb{R}$ be defined as in (2.44). Let $\{\mathcal{N}_1, \dots, \mathcal{N}_{|\mathcal{N}|}\}$ be the indices of the normal agents, with $\mathcal{N}_i$ being the index of the ith agent in $\mathcal{N}$. The Clarke generalized gradients ∂M and ∂m are*

$$\partial M(x_\mathcal{N}) = co \bigcup \left\{ e^i : \mathcal{N}_i \in S_M \right\}, \tag{2.58}$$

$$\partial m(x_\mathcal{N}) = co \bigcup \left\{ e^i : \mathcal{N}_i \in S_m \right\}. \tag{2.59}$$

Proof. By Lemma 2.8, $M(x_\mathcal{N})$ is the pointwise maximum over the functions $(e^i)^T x_\mathcal{N}$ for $i \in \{1, \dots, |\mathcal{N}|\}$, which are all locally Lipschitz and regular on $\mathbb{R}^{|\mathcal{N}|}$. Furthermore, each function

$(e^i)^T x_{\mathcal{N}}$ is continuously differentiable at all $x_{\mathcal{N}} \in \mathbb{R}^{|\mathcal{N}|}$, implying that the following holds [198]:

$$\partial((e^i)^T x_{\mathcal{N}}) = \nabla((e^i)^T x_{\mathcal{N}}) = e^i.$$

By Proposition 2.2, we therefore have

$$\partial M(x_{\mathcal{N}}) = \text{co} \bigcup \left\{ e^i : i \in I_{\max}(x_{\mathcal{N}}) \right\}, \tag{2.60}$$

where $I_{\max}(x_{\mathcal{N}})$ denotes the indices j such that $(e^j)^T x_{\mathcal{N}} = x_{\mathcal{N}_j} = M(x_{\mathcal{N}})$ (recall from (2.41) that $(e^j)^T x_{\mathcal{N}} = x_{\mathcal{N}_j}$, where $\mathcal{N}_j$ is the index of the jth normal agent in $\mathcal{N}$). By equation (2.45), the set of indices $\mathcal{N}_j$ such that $x_{\mathcal{N}_j} = M(x_{\mathcal{N}})$ is precisely $S_M(x_{\mathcal{N}})$, which by substitution into (2.60) yields (2.58).

Similar arguments can be used to derive $\partial m(x_{\mathcal{N}})$. The function $m(x_{\mathcal{N}})$ is the pointwise minimum over the functions $(e^i)^T x_{\mathcal{N}}$ for $i \in \{1, \dots, |\mathcal{N}|\}$ which are all locally Lipschitz, regular, and continuously differentiable on $\mathbb{R}^{|\mathcal{N}|}$. Observe that the functions $-(e^i)^T x_{\mathcal{N}}$ are also locally Lipschitz, regular, and continuously differentiable on $\mathbb{R}^{|\mathcal{N}|}$. By Proposition 2.2, we therefore have

$$\partial m(x_{\mathcal{N}}) = \text{co} \bigcup \left\{ e^i : i \in I_{\min}(x_{\mathcal{N}}) \right\}, \tag{2.61}$$

where $I_{\min}(x_{\mathcal{N}})$ denotes the indices j such that $(e^j)^T x_{\mathcal{N}} = x_{\mathcal{N}_j} = m(x_{\mathcal{N}})$. By Definition 2.45, the set of indices $\mathcal{N}_j$ such that $x_{\mathcal{N}_j} = m(x_{\mathcal{N}})$ is precisely $S_m(x_{\mathcal{N}})$, which by substitution into (2.61) yields (2.59). As a final note, observe that since $m(\cdot)$ is locally Lipschitz by Lemma 2.8, by the Dilation Rule [198] we can derive $\partial(-m(x_{\mathcal{N}})) = \partial((-1)m(x_{\mathcal{N}})) = -(\partial m(x_{\mathcal{N}}))$. □

Before presenting our next main result, we will first need the following Lemma.

Lemma 2.10. *Let $q \in \mathbb{R}^m$ and let $\Theta = \{\theta \in \mathbb{R}^m : \theta \succeq \mathbf{0},\ \mathbf{1}^T\theta = 1\}$. Let $a \in \mathbb{R}$. Then $\theta^T q = a$ for all $\theta \in \Theta$ if and only if $q = a\mathbf{1}$.*

Proof. Necessity: If $q = a\mathbf{1}$, then for all $\theta \in \Theta$ we have $\theta^T q = q^T\theta = a(\mathbf{1}^T\theta) = a$.

Sufficiency: We prove the contrapositive, i.e. $q \neq a\mathbf{1}$ implies there exists $\theta^* \in \Theta$ such that $(\theta^*)^T q \neq a$. If $q \neq a\mathbf{1}$ then there exists $j \in \{1, \dots, m\}$ such that $q_j \neq a$. Choose $\theta^* = e_j$, where e_j is the jth column of the identity matrix. Clearly, we then have $\theta_* \succeq \mathbf{0}$ and $\mathbf{1}^T\theta^* = 1$, implying $\theta^* \in \Theta$. Then $(\theta^*)^T q = e_j^T q = q_j \neq a$. □

The next theorem proves that $m(x_{\mathcal{N}}(t))$ is nondecreasing on the interval $t \in [0, t_1)$ and that $M(x_{\mathcal{N}}(t))$ is nonincreasing on the interval $t \in [0, t_1)$, where $[0, t_1)$ is the interval on which $x_{\mathcal{N}}(t)$ is a solution to (2.56). This will imply that the states of all agents remain within the invariant set $[m(x_{\mathcal{N}}(0)), M(x_{\mathcal{N}}(0))]$ for all $t \geq 0$.

Theorem 2.7. *Consider a digraph $\mathcal{D} = \{\mathcal{V}, \mathcal{E}\}$ with the system dynamics (2.56) under the FTRC Protocol in Algorithm 2.5. Suppose that $\mathcal{A}$ is an F-local model and that $\mathcal{D}$ is $(2F+1)$-robust. Let $m(x_{\mathcal{N}}(t))$ and $M(x_{\mathcal{N}}(t))$ be defined as in (2.44). Then the derivatives $\frac{d}{dt}(M(x_{\mathcal{N}}(t)))$ and $\frac{d}{dt}(m(x_{\mathcal{N}}(t)))$ exist at almost all $t \in [0, t_1)$ and satisfy*

$$\frac{d}{dt}(M(x_{\mathcal{N}}(t))) \in [-\alpha, 0], \tag{2.62}$$

$$\frac{d}{dt}(m(x_{\mathcal{N}}(t))) \in [0, \alpha], \tag{2.63}$$

at almost all $t \in [0, t_1)$.

Proof. Where possible, we abbreviate $x_{\mathcal{N}}(t)$ to $x_{\mathcal{N}}$ for brevity. By Lemma 2.7, solutions $x_{\mathcal{N}}(t)$ to the differential inclusion (2.56) are guaranteed. By Lemma 2.8, the functions $M(\cdot)$ and $(-m(\cdot))$ are both locally Lipschitz and regular on $\mathbb{R}^{|\mathcal{N}|}$. Therefore by Proposition 2.3, the compositions $M(x_{\mathcal{N}}(t))$ and $(-m(x_{\mathcal{N}}(t)))$ are differentiable at almost all $t \in [0, t_1)$. In addition, by Proposition 2.3 we have $\frac{d}{dt}M(x_{\mathcal{N}}) \in \widetilde{\mathcal{L}}_G M(x_{\mathcal{N}})$ and $\frac{d}{dt}(-m(x_{\mathcal{N}})) \in \widetilde{\mathcal{L}}_G(-m(x_{\mathcal{N}}))$ at almost all $t \in [0, t_1)$, where $\widetilde{\mathcal{L}}_G M(x_{\mathcal{N}})$ and $\widetilde{\mathcal{L}}_G(-m(x_{\mathcal{N}}))$ represents the set-valued Lie derivatives of $M(x_{\mathcal{N}})$ and $(-m(x_{\mathcal{N}}))$, respectively. The next part of the proof focuses on characterizing $\widetilde{\mathcal{L}}_G M(x_{\mathcal{N}})$ and $\widetilde{\mathcal{L}}_G(-m(x_{\mathcal{N}}))$, from which we derive the range of possible values for $\frac{d}{dt}M(x_{\mathcal{N}})$ and $\frac{d}{dt}m(x_{\mathcal{N}})$.

We first consider $\widetilde{\mathcal{L}}_G M(x_{\mathcal{N}})$. By definition,

$$\begin{aligned}\widetilde{\mathcal{L}}_G M(x_{\mathcal{N}}) = &\{a \in \mathbb{R} : \exists v \in G[f_{\mathcal{N}}](x_{\mathcal{N}}) \text{ such that} \\ &z^T v = a \; \forall z \in \partial M(x_{\mathcal{N}})\}\end{aligned} \tag{2.64}$$

Define E_M as a matrix with columns e^i such that $\mathcal{N}_i \in S_M$.[10] By the definition of $\partial M(x_{\mathcal{N}})$ from Lemma 2.9, each $z \in \partial M(x_{\mathcal{N}})$ can be written as the convex combination $z = E\theta$, where $\theta \in \mathbb{R}^{|S_M|}$, $\theta \succeq \mathbf{0}$ and $\mathbf{1}^T\theta = 1$. It therefore holds that $a \in \widetilde{\mathcal{L}}_G M(x_{\mathcal{N}})$ if and only if there exists a $v \in G[f_{\mathcal{N}}](x_{\mathcal{N}})$ such that $z^T v = (\theta^T E_M^T)v = \theta^T(E_M^T v) = a$ for all $\theta \succeq \mathbf{0}$, $\mathbf{1}^T\theta = 1$. Lemma 2.10 proves that this holds if and only if $E_M^T v = a\mathbf{1}$. Recall that E_M is composed of the columns e^i such that $\mathcal{N}_i \in S_M$. By the form of $G[f_{\mathcal{N}}](x_{\mathcal{N}})$, choosing any $v \in G[f_{\mathcal{N}}](x_{\mathcal{N}})$ with $v_i = a \in [-\alpha, \alpha]$ for all i such that $\mathcal{N}_i \in S_M$ yields $E_M^T v = a\mathbf{1}$, and therefore $\frac{d}{dt}M(x_{\mathcal{N}}) \in \widetilde{\mathcal{L}}_G M(x_{\mathcal{N}}) = [-\alpha, \alpha]$.

We can further restrict the range of values for $\frac{d}{dt}M(x_{\mathcal{N}})$ to the range $[-\alpha, 0]$ by considering the form of (2.40). We prove this by contradiction. Suppose there exists a $t \geq 0$ such that $\frac{d}{dt}M(x_{\mathcal{N}}(t)) > 0$. This implies that there exists $t \geq 0$ and $\mathcal{N}_{i'} \in S_M(t)$ such that $u_{\mathcal{N}_{i'}}(t) = \alpha \operatorname{sign}\left(\sum_{\mathcal{J}_{\mathcal{N}_{i'}} \setminus \mathcal{R}_{\mathcal{N}_{i'}}[t]} g(x_j^{\mathcal{N}_{i'}}(t)) - g(x_{\mathcal{N}_{i'}}(t))\right) > 0$. However, for all $\mathcal{N}_i \in S_M$ all normal in-neighbors $j \in \mathcal{V}_i(t)$ have state values less than or equal to $x_{\mathcal{N}_i}(t)$ by the definition of S_M. Since $g(\cdot)$

[10] Recall that $\mathcal{N}_i$ is defined immediately after Eq. (2.41).

is strictly increasing, we have $g(x_j^{\mathcal{N}_i}) - g(x_{\mathcal{N}_i}) \leq 0$ for all normal in-neighbors $j \in \mathcal{V}_{\mathcal{N}_i} \cap \mathcal{N}$. In addition, since $\mathcal{A}$ is F-local, any adversarial signals satisfying $g(x_k^{\mathcal{N}_i}(t)) > g(x_{\mathcal{N}_i}(t))$ for $k \in (\mathcal{V}_{\mathcal{N}_i} \cap \mathcal{A})$ are filtered out by Algorithm 2.5. Therefore we must have $\sum_{\mathcal{J}_{\mathcal{N}_i} \setminus \mathcal{R}_{\mathcal{N}_i}[t]} g(x_j^{\mathcal{N}_i}(t)) - g(x_{\mathcal{N}_i}(t)) \leq 0$ for all $\mathcal{N}_i \in S_M$, which implies that $u_{\mathcal{N}_i}(t) = (\alpha)\text{sign}\left(\sum_{\mathcal{J}_{\mathcal{N}_i} \setminus \mathcal{R}_{\mathcal{N}_i}[t]} g(x_j^{\mathcal{N}_i}(t)) - g(x_{\mathcal{N}_i}(t))\right) \leq 0$ for all $\mathcal{N}_i \in S_M$. This contradicts the assumption that there exists an $\mathcal{N}_{i'} \in S_M$ with $u_{\mathcal{N}_{i'}}(t) > 0$. Therefore $\frac{d}{dt}M(x_{\mathcal{N}}) \leq 0$ wherever it exists, which yields $\frac{d}{dt}M(x_{\mathcal{N}}) \in [-\alpha, 0]$.

The preceding logic can be repeated to demonstrate that $\frac{d}{dt}(-m(x_{\mathcal{N}})) \in [-\alpha, 0]$ wherever this derivative exists, from which we can conclude that $\frac{d}{dt}m(x_{\mathcal{N}}) \in [0, \alpha]$. □

The preceding result demonstrates that $M(x_{\mathcal{N}}(t))$ is nonincreasing and $m(x_{\mathcal{N}}(t))$ is nondecreasing for all $t \geq 0$, and therefore all agents' states remain within $[m(x_{\mathcal{N}}(0)), M(x_{\mathcal{N}}(0))]$ for all $t \geq 0$. This implies that the hyperrectangle $P(0) \subset \mathbb{R}^{|\mathcal{N}|}$ defined as

$$P(0) = \begin{bmatrix} [m(x_{\mathcal{N}}(0)), M(x_{\mathcal{N}}(0))] \\ \vdots \\ [m(x_{\mathcal{N}}(0)), M(x_{\mathcal{N}}(0))] \end{bmatrix} \tag{2.65}$$

is invariant for all $t \geq 0$, which is precisely condition (i) of Finite-Time Resilient Consensus (Definition 2.16).

Our next result will require the following Lemma.

Lemma 2.11. *Under the conditions of Theorem 2.7, if $\frac{d}{dt}M(x_{\mathcal{N}}(t))$ exists at $t \geq 0$ then $u_{i_1}(t) = u_{i_2}(t)$ for all $i_1, i_2 \in S_M$. Similarly, if $\frac{d}{dt}m(x_{\mathcal{N}}(t))$ exists at $t \geq 0$, then $u_{j_1}(t) = u_{j_2}(t)$ for all $j_1, j_2 \in S_m$.*

Proof. We prove the contrapositive. If at some $t \geq 0$ there exists $i_1, i_2 \in S_M$ such that $u_{i_1}(t) \neq u_{i_2}(t)$, then by (2.39) $\dot{x}_{i_1}(t) \neq \dot{x}_{i_2}(t)$. Since $M(x_{\mathcal{N}}(t))$ is the pointwise maximum $\max_{i \in \mathcal{N}} x_i(t)$ and $x_{i_1}(t) = x_{j_1}(t) = M(x_{\mathcal{N}}(t))$ by definition of S_M, the derivative $\frac{d}{dt}M(x_{\mathcal{N}}(t))$ is therefore undefined at t. Similar arguments demonstrate the same result for $\frac{d}{dt}m(x_{\mathcal{N}}(t))$. □

The next result demonstrates that the time derivative of the composition $V(x_{\mathcal{N}}(t))$, wherever it exists, is upper bounded by $-\alpha$ when $x_{\mathcal{N}}(t)$ is not in span$(\mathbf{1})$.

Theorem 2.8. *Let $V(\cdot)$ be defined as in (2.44). Under the conditions of Theorem 2.7, the derivative $\frac{d}{dt}V(x_{\mathcal{N}}(t))$ exists at almost all $t \in [0, t_1)$. Furthermore, for all $x_{\mathcal{N}}(t) \notin$ span$(\mathbf{1})$, the derivative of $V(x_{\mathcal{N}}(t))$ satisfies*

$$\frac{d}{dt}(V(x_{\mathcal{N}}(t))) \leq -\alpha < 0 \tag{2.66}$$

at almost all $t \in [0, t_1)$.

Proof. Where possible, we abbreviate $x_{\mathcal{N}}(t)$ to $x_{\mathcal{N}}$ for brevity. Recall from the definition of the function $V(x_{\mathcal{N}}(t))$ that $V(x_{\mathcal{N}}(t)) = M(x_{\mathcal{N}}(t)) - m(x_{\mathcal{N}}(t))$, which implies $\frac{d}{dt}V(x_{\mathcal{N}}(t)) = \frac{d}{dt}M(x_{\mathcal{N}}(t)) - \frac{d}{dt}m(x_{\mathcal{N}}(t))$. Since $\frac{d}{dt}M(x_{\mathcal{N}}(t))$ and $\frac{d}{dt}m(x_{\mathcal{N}}(t))$ exist at almost all $t \in [0, t_1)$, $\frac{d}{dt}V(x_{\mathcal{N}}(t))$ exists at almost all $t \in [0, t_1)$.

Next, we show that for all $x_{\mathcal{N}} \notin \text{span}(\mathbf{1})$, there exists an agent $i \in (S_M \cup S_m)$ such that either $u_i(t) = -\alpha$ or $u_i(t) = \alpha$. Observe that $x_{\mathcal{N}} \notin \text{span}(\mathbf{1})$ implies that S_M and S_m are nonempty and disjoint. By the definition of $(2F+1)$-robustness (Definition 2.4), at least one of the sets S_M, S_m is $(2F+1)$-reachable. Without loss of generality, suppose S_M is $(2F+1)$-reachable. Then there exists $i \in S_M$ with $|\mathcal{V}_i \backslash S_M| \geq 2F+1$. By the FTRC-P, agent i will filter out at most $2F$ values. Since $i \in S_M$, any normal values received by i will be less than or equal to $g(x_i(t))$. Since $\mathcal{A}$ is F-local, any adversarial values greater than $g(x_i(t))$ will be filtered out as per the FTRC-P. This implies that agent i will *not* filter out at least one value $g(x_j^i(t)) < g(x_i(t))$, and that $\sum_{\mathcal{J}_i \backslash \mathcal{R}_i[t]} g(x_j^i(t)) - g(x_i(t)) < 0$. Therefore $u_i(t) = -\alpha$. Similar arguments can be used to show that if S_m is $(2F+1)$-reachable, there exists $i \in S_m$ with $u_i(t) = \alpha$.

Consider any $t \geq t_0$ such that $x_{\mathcal{N}}(t) \notin \text{span}(\mathbf{1})$ and $\frac{d}{dt}V(x_{\mathcal{N}})$ exists. The existence of $\frac{d}{dt}V(x_{\mathcal{N}}(t))$ implies that both $\frac{d}{dt}M(x_{\mathcal{N}})$ and $\frac{d}{dt}m(x_{\mathcal{N}})$ exist. Since $x_{\mathcal{N}}(t) \notin \text{span}(\mathbf{1})$, by prior arguments there either exists a $i_M \in S_M$ with $u_{i_M}(t) = -\alpha$ or an $i_m \in S_m$ with $u_{i_m}(t) = \alpha$. We consider each case separately.

Case 1: Suppose there there exists an i_M with $u_{i_M}(t) = -\alpha$. Recall that we are considering any $t \geq 0$ such that $x_{\mathcal{N}}(t) \notin \text{span}(\mathbf{1})$ and $\frac{d}{dt}V(x_{\mathcal{N}})$ exists, implying that $\frac{d}{dt}M(x_{\mathcal{N}})$ exists. Since $\frac{d}{dt}M(x_{\mathcal{N}})$ exists at t, then by Lemma 2.11 we have $u_j(t) = -\alpha$ at t for all $j \in S_M$. This implies that $\frac{d}{dt}M(x_{\mathcal{N}}) = -\alpha$. Since $\frac{d}{dt}m(x_{\mathcal{N}})$ also exists at our chosen t and $m(x_{\mathcal{N}}) \in [0, \alpha]$, we have $\frac{d}{dt}V(x_{\mathcal{N}}) \leq -\alpha < 0$.

Case 2: Suppose there there exists an i_m with $u_{i_m}(t) = \alpha$ Since $\frac{d}{dt}m(x_{\mathcal{N}})$ exists at our choice of t, then by Lemma 2.11 we have $u_j(t) = \alpha$ at t for all $j \in S_M$. This implies that $\frac{d}{dt}m(x_{\mathcal{N}}) = \alpha$. Since $\frac{d}{dt}M(x_{\mathcal{N}})$ also exists at our chosen t and $M(x_{\mathcal{N}}) \in [-\alpha, 0]$, we have $\frac{d}{dt}V(x_{\mathcal{N}}) \leq -\alpha < 0$.

Since in each case we have $\frac{d}{dt}V(x_{\mathcal{N}}) \leq -\alpha < 0$, for all $x_{\mathcal{N}}(t) \notin \text{span}(\mathbf{1})$ the equation (2.66) holds at almost all $t \in [0, t_1)$. □

Our final theorem completes this section by showing that FTRC is achieved by the system of normal agents. In particular, this theorem demonstrates that solutions to the trajectories of the normal agents exist on the time interval $t \in [0, \infty)$, and that there exists a time $T \geq 0$ such that $x_{\mathcal{N}}(t) \in \text{span}(\mathbf{1})$ for all $t \geq T$.

Theorem 2.9. *Consider a digraph $\mathcal{D} = \{\mathcal{V}, \mathcal{E}\}$ with the system dynamics (2.56) under the FTRC Protocol in Algorithm 2.5. Suppose that $\mathcal{A}$ is an F-local model and that $\mathcal{D}$ is $(2F+1)$-robust. Then the normal agents achieve FTRC as described in Definition 2.16.*

Proof. By Theorem 2.7, all normal agents remain within the invariant set $P(0)$ defined in (2.65), satisfying condition (i) of FTRC. By Theorem 2.8, condition (ii) of FTRC is satisfied. To show that condition (iii) of FTRC is satisfied, observe that by Lemma 2.8 $V(\cdot)$ is locally Lipschitz on $\mathbb{R}^{|\mathcal{N}|}$. Since Caratheodory solutions $x_{\mathcal{N}}(t)$ of (2.56) are absolutely continuous, the composition $V(x_{\mathcal{N}}(t))$ is therefore absolutely continuous [197, Appendix B]. By Lemma 2.7 $G[f](x_{\mathcal{N}})$ satisfies the hypotheses of Proposition 2.1 for all $x_{\mathcal{N}} \in \mathbb{R}^{|\mathcal{N}|}$ and for all $t \geq 0$, implying that these hypotheses are satisfied for the compact set $Q = \overline{\text{co}}(P(0) + B(0, \epsilon))$ for some $\epsilon > 0$ (where addition is in terms of the Minkowski sum). Since $P(0)$ is an invariant set and $P(0)$ does not intersect the boundary of Q, no solution $x_{\mathcal{N}}(t)$ will reach the boundary of Q for all $t \geq 0$. Consider any domain $D(t_1') = [-\delta, t_1'] \times Q$ for $\delta, t_1' > 0$. Each domain $D(t_1')$ is therefore compact. By Theorem 2.5 this implies that all solutions $x_{\mathcal{N}}(t)$ to (2.56) exist on $t \in [0, t_1')$ for any $t_1' > 0$, which implies that all solutions $x_{\mathcal{N}}(t)$ to (2.56) exist on $t \in [0, \infty)$. By Theorems 2.6 and 2.8 $V(x_{\mathcal{N}}(t))$ converges to span$(\mathbf{1})$ in finite time, implying that $x_{\mathcal{N}}(t)$ reaches consensus in finite time and condition (iii) of FTRC is satisfied. Since by Theorem 2.8 we have $\frac{d}{dt}(V(x_{\mathcal{N}}(t))) \leq -\alpha$ at almost all $t \in [0, \infty)$, the time of convergence satisfies $T(x_{\mathcal{N}}(0)) = \frac{1}{\alpha} V(x_{\mathcal{N}}(0))$. □

2.5.5 Discussion

We now discuss two aspects of the finite time resilient consensus approach presented in this section. First, we briefly discuss the tradeoffs in choosing the α parameter in Algorithm 2.5. Second, we discuss further the implications of Assumption 2.2; specifically, the possibility of the adversaries sending signals that are not Lebesgue measurable.

The parameter α in Algorithm 2.5 controls the speed at which agents' states move. The overall effect of applying Algorithm 2.5 is that, intuitively speaking, the gap between the minimum and maximum normal nodes will shrink at a rate of at least α. As per the proof of Theorem 2.9, the time of convergence $T(x_{\mathcal{N}}(0))$ satisfies $T(x_{\mathcal{N}}(0)) = \frac{1}{\alpha} V(x_{\mathcal{N}}(0))$. Therefore, the larger the value of α the faster the normal agents will converge to consensus. When implemented in systems where agents states represent physical systems, the maximum value of α will be limited by the control input bounds for the agents. In addition, applying signum-type controllers in physical systems can lead to chattering as agents approach the final consensus value [210–212]. The larger the value of α, the more pronounced the chattering effects will be. Investigating the application of standard

chattering suppression methods [212] to the method presented in this section is left as an interesting direction for future work.

We now discuss further the implications of Assumption 2.2 and the possibility of the adversaries sending signals that are not Lebesgue measurable. To give a simple example of non-Lebesgue-measurable function, the indicator function $\mathbf{1}_S : \mathbb{R} \to \mathbb{R}$ defined as

$$\mathbf{1}_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases}$$

is not Lebesgue measurable if the subset $S \subset \mathbb{R}$ is not Lebesgue measurable. Note that by definition of measurability, the existence of a non-Lebesgue-measurable mapping from $\mathbb{R}$ to $\mathbb{R}$ implies the existence of a subset of $\mathbb{R}$ which is not Lebesgue measurable. Contrapositively, the nonexistence of non-Lebesgue-measurable subsets of $\mathbb{R}$ implies that all functions mapping $\mathbb{R}$ to $\mathbb{R}$ are Lebesgue measurable.

There are at least two schools of thought on this point. If one assumes that the axiom of choice holds, then the axiom of choice can be used to demonstrate the existence of subsets of $\mathbb{R}$ that are not Lebesgue measurable (e.g. Vitali sets [213]). However, the Solovay model [214] demonstrated that the existence of a non-Lebesgue-measurable subset of $\mathbb{R}$ cannot be proven without using the axiom of choice. Under the Solovay model, which does not assume the axiom of choice but instead assumes the existence of an inaccessible cardinal, *all* subsets of $\mathbb{R}$ are Lebesgue measurable.

The question of whether the adversaries can send non-Lebesgue-measurable signals therefore hinges upon which assumptions are made about the axiom of choice and the existence of an inaccessible cardinal. A full discussion of the merits of each approach is completely beyond the scope of this dissertation, and so we conclude by simply asserting that the results of this section hold under Assumption 2.2, i.e. when all adversarial signals are Lebesgue measurable.

2.5.6 Simulations

Finally, we present simulation results demonstrating our method for finite-time leaderless consensus in continuous-time systems using possibly discontinuous control inputs. Our simulations are for a system of $n = 15$ agents. The underlying communication graph is a k-circulant digraph with $k = 11$, which can be shown to be at least 6-robust using results from [102]. The highest (integer) value of F for which we can infer the graph is $(2F + 1)$-robust is therefore $F = 2$. Each agent's initial state $x_i(0) \in \mathbb{R}$, $i \in \mathcal{V}$ is a random value on the interval $[0, 50]$. Two agents are chosen at random to be adversaries, resulting in $\mathcal{A} = \{2, 13\}$. We emphasize that *the normally-behaving agents have no knowledge as to whether their in-neighbors are adversarial or normal*. The adversarial agents are *malicious* [2], meaning each adversary updates its state according to some arbitrary

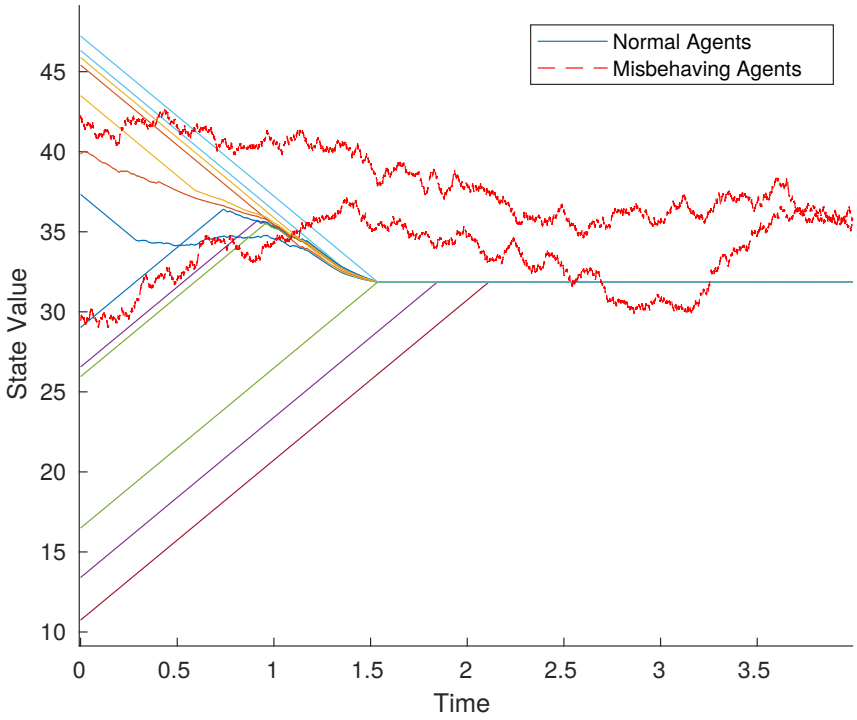


Figure 2.16: Simulation of a network of 15 agents appling the FTRC-P. The dotted red lines represent the adversarial agents.

function of time but sends the same state value to all of its out-neighbors. All other agents are normal and apply the FTRC-P from Algorithm 2.5 with $\alpha = 10$. The function $g : \mathbb{R} \to \mathbb{R}$ in (2.40) is chosen to be $g(x) = (1/10)x^3 + (1/1000)x^5 + (1/10000)x^7$, which can be verified to be a strictly increasing function. Figure 2.16 shows the results of this first simulation, with malicious agents represented by red dotted lines and normally-behaving agents represented by solid colored lines. The normal agents achieve consensus in a finite amount of time despite the influence of the adversarial agents.

2.6 Discussion

The results in this section for resilient consensus based on MSR-type algorithms hinge upon the F-totality or F-locality of the adversarial model, and the structure of the network communication graph (i.e. r-robustness, strong r-robustness w.r.t. a set S, RDAG with parameter F). Some discussion is warranted as to how realistic these conditions/assumptions are, and how they compare to scenarios found in the real world.

Some readers may question the perceived assumption that the adversarial model will always

satisfy the condition of being F-total or F-local for a fixed $F \in \mathbb{Z}_{\geq 0}$. There are two things to keep in mind in this regards. First, any consensus algorithm that is *not* designed to be resilient to adversarial attacks makes a similar assumption; in particular, such an algorithm assumes that the adversarial model is 0-total and 0-local ($F = 0$). Considering the proliferation of adversarial attacks discussed in the Introduction to this dissertation, it is far more unrealistic to assume that there will be exactly zero adversaries in a network than it is to assume that the adversaries form an F-local or F-total set for some positive F. Second, instead of being viewed as an *assumption*, the condition of F-totality or F-locality can instead be viewed as a *quantification of resilience*. In other words, the parameter F precisely quantifies the number and distribution of adversaries that a given network running an MSR-type algorithm can tolerate. This allows networks to be designed and modified to be able to handle an appropriate value of F for which engineers and stakeholders believe will mitigate the most likely threat model for the network.

The approach of modeling the adversary set as F-total or F-local is most appropriate in settings where it is sufficiently difficult (although perhaps not impossible) for adversaries to attack and compromise agents. In the particular case of mobile agents with proximity-based communication, it is typically more appropriate to consider an F-total adversarial set since the set of adversaries is able to collude by simultaneously entering the in-neighborhood of a common normal agent or set of normal agents. The approach in this chapter does assume that agents are able to verify that received information originates from actual agents in the network and not "spoofed" agents. In this regards the methods presented in this chapter are not designed to be resilient against Sybil attacks. However, alternate methods for extending MSR algorithms to mitigate such attacks have been proposed in the literature [215].

We acknowledge that more work needs to be performed to understand the effects of violating the F-total / F-local adversarial set assumptions on more general network configurations. The results in Section 2.3.3 outline one particular scenario where violation of these assumptions leads to the adversaries gaining full control over a leaderless network; however, studying more general network topologies and even scenarios with normally-behaving leaders could also be investigated more thoroughly.

We also point out that there is nothing preventing MSR-type algorithms from being combined with other adversary mitigation techniques in a network to provide additional protection against adversarial attacks. For example, MSR-type algorithms could be used in tandem with adversarial detection techniques to allow the network to be able to withstand a particular attack without negative effects until the adversaries' identities could be identified.

Finally, with regards to the network notions of r-robustness, strong r-robustness w.r.t. a set S, and RDAGs, prior work has studied the emergence of robustness properties in certain classes of random graphs [97, 99]. More work remains to be done in studying the emergence of robustness in

other classes of random graphs exhibited in real-world scenarios. However, when it is possible to design or control the communication structure between agents in the network, there exist methods to build or modify a network to satisfy these robustness conditions to a desired degree of network resilience. Such methods are discussed in Chapter 3.

2.7 Conclusion

This chapter presented several novel contributions to the area of resilient consensus techniques. We introduced conditions for agents with discrete-time dynamics to resiliently track a reference signal propagated by a set of leader agents despite a bounded number of the leaders and followers behaving adversarially. We also presented a novel continuous-time resilient controller that guarantees that normally-behaving agents can converge to a formation with respect to a set of leaders in the presence of adversarial agents. This controller guarantees convergence in finite-time even with bounded control inputs, and the resilient filtering method was also applied to a discrete-time algorithm that guarantees exponential convergence of agents to formation in the presence of adversaries under bounded inputs. Finally, we presented conditions under which finite-time convergence of normally-behaving agents in the presence of discontinuous, nonlinear adversarial signals is guaranteed. These final results introduce a novel and general class of MSR-type algorithms with the ability to mitigate adversarial attacks that are Lebesgue measurable in time.

CHAPTER 3

Determining r- and (r, s)-Robustness for Design and Analysis of Resilient Networks

3.1 Introduction

As discussed in Chapters 1 and 2 of this dissertation, many recent results in resilient consensus in prior literature have been based on MSR-type algorithms. These results typically employ the graph theoretical notions known as r-robustness and (r, s)-robustness [2, 71]. For example, r-robustness and (r, s)-robustness are key notions included in the sufficient conditions for convergence of the W-MSR [2], ARC-P [190], SW-MSR [79], DP-MSR [76] algorithms, and others. Given an upper bound on the global or local number of adversaries in the network, the aforementioned resilient algorithms guarantee convergence of normally behaving agents' states to a value within the convex hull of initial states if the integers r and s are sufficiently large. Determining the values of r and s for which a graph is r- or (r, s)-robust is critical for knowing the maximum adversary model that a network can tolerate while applying an MSR-type algorithm. Unfortunately, determining the r- and (r, s)-robustness of arbitrary digraphs is an NP-hard problem in general [96]. The decision problem of determining if an arbitrary graph is r-robust for a given integer r is coNP-complete [97].

The first algorithmic analysis of determining the values of r and s for arbitrary digraphs was given in [96]. The algorithms proposed in [96] employ an exhaustive search to determine the maximum values of r and s for a given digraph, and have exponential complexity w.r.t. the number of nodes in the network. Subsequent work has focused on methods to circumvent the complexity of robustness determination including graph construction methods which increase the graph size while preserving given values of r and s [2, 98]; lower bounding r with the isoperimetric constant and algebraic connectivity of undirected graphs [99]; and demonstrating the behavior of r as a function of certain graph properties [97, 100–104]. In particular, it has been shown that the r-robustness of some specific classes of graphs can be exactly determined in polynomial time from

certain graph parameters. Examples include k-circulant graphs [102] and lattice-based formations [103, 104]. Another recent approach has used machine learning to correlate characteristics of certain graphs to the values of r and s [105], but these correlations do not provide explicit guarantees. Despite the impressive results of prior literature, methods to either approximate or determine exactly the r- and (r, s)-robustness of arbitrary *digraphs* remain relatively rare. Methods to determine the *exact* r- or (r, s)-robustness of arbitrary *undirected* graphs are also uncommon. Finding more efficient or practical ways of determining the robustness of arbitrary graphs in general, and digraphs in particular, remains an open problem.

In response to this open problem, this chapter first introduces a novel class of directed graphs called k-circulant digraphs for which lower bounds on the maximum r-robustness and (r, s)-robustness can be easily determined. The structure of these graphs is completely determined by the number of agents n and the integer parameter k. We demonstrate that the r- and (r, s)-robustness of k-circulant digraphs is a function of k regardless of the graph size n. This property makes it possible to scale such graphs to an arbitrary value of agents while guaranteeing that the graph remains r-robust for a particular r. In addition, we demonstrate that these graphs can also satisfy the conditions of *strong r-robustness* with respect to a given subset, which has applications to resilient leader-follower consensus scenarios. The results of this portion of the chapter are based on the published works [102, 180].

Next, this chapter introduces novel methods for determining the r- and (r, s)-robustness of digraphs and undirected graphs using mixed integer linear programming (MILP). These methods only require knowledge of the graph Laplacian matrix and are zero-one MILPs, i.e. with all integer variables being binary. To the best of our knowledge, this is the first time the robustness determination problem has been formulated using an optimization framework. These results connect the problem of graph robustness determination to the extensive and well-established literature on integer programming and linear programming. More specifically, this second part of the chapter presents a method to determine the maximum integer for which a nonempty, nontrivial, simple digraph is r-robust using mixed integer linear programming. Next, we present a method which determines the (r, s)-robustness of a digraph using linear programming. Here, the (r, s)-robustness of a digraph refers to the maximal (r, s) integer pair according to a lexicographical order for which a given digraph is (r, s)-robust, as first described in [96]. Furthermore, we show that our method can also determine the maximum integer F for which a digraph is $(F + 1, F + 1)$-robust, which is not considered in [96]. Finally, we present two mixed integer linear programs whose optimal values provide lower and upper bounds on the maximum r for which a nonempty, nontrivial, simple digraph is r-robust. These two formulations exhibit a lower complexity than the method in the first contribution described above. Formulating the r- and (r, s)-robustness determination problem in an optimization setting provides several advantages. First, expressing the robustness determination

problem in MILP form allows for approximate *lower* bounds on a given digraph's r-robustness to be iteratively tightened using algorithms such as branch-and-bound. Lower bounds on the maximum value of s for which a given digraph is (r, s) robust (for a given nonnegative integer r) can also be iteratively tightened using the approach in this chapter. Prior algorithms are only able to tighten the upper bound on the maximum robustness for a given digraph or undirected graph. Second, this formulation enables commercially available solvers such as Gurobi or MATLAB's *intlinprog* to be used to find the maximum robustness of any digraph. Finally, experimental results using this new formulation suggest a reduction in computation time as compared to the centralized algorithm proposed in [96]. The results of this portion of the chapter are based on the published works [184, 186].

The chapter is organized as follows: the formulation for the r- and (r, s)-robustness determination problem is given in Section 3.2. Next, a description of k-circulant digraphs and a proof that their robustness is a function of k is given in Section 3.3. Then, a framework for determining r- and (r, s)-robustness of digraphs using mixed integer linear programming is given in Section 3.4. Finally, we present simulation results demonstrating the results of this chapter in Section 3.4.5 and give a brief conclusion in section 3.5. The first simulation demonstrates the efficacy of using k-circulant digraphs in a resilient leaderless consensus scenario. The final simulations present a numerical analysis of the performance of the proposed mixed integer robustness determination method as compared to prior state-of-the-art algorithms.

3.2 Problem Formulation

The notions of r- and (r, s)-robustness are graph-theoretical properties used to describe the interconnections (e.g., communication topologies) in multi-agent networks. Examples of such networks include stations in a power grid, satellites in formation, or a group of mobile robots. In these networks, edges model the ability for one agent i to transmit information to another agent j. Prior literature commonly considers *simple* digraphs, which have no repeated edges or self edges [2, 50, 216–218]. More specifically, simple digraphs satisfy $(i, i) \notin \mathcal{E}\ \forall i \in \mathcal{V}$, and if the directed edge $(i, j) \in \mathcal{E}$, then it is the only directed edge from i to j. Prior work also commonly considers nonempty and nontrivial graphs, where $|V| > 1$.

Assumption 3.1. *This section considers nonempty, nontrivial, simple digraphs.*

The property of r-robustness is based upon the notion of r-reachability. The definitions of r-reachability and r-robustness are as follows:

Definition 3.1 ([2]). *Let $r \in \mathbb{Z}_{\geq 0}$ and $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be a digraph. A nonempty subset $S \subset \mathcal{V}$ is r-reachable if $\exists i \in S$ such that $|\mathcal{V}_i \backslash S| \geq r$.*

Definition 3.2 ([2]). *Let $r \in \mathbb{Z}_{\geq 0}$. A nonempty, nontrivial digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ on n nodes $(n \geq 2)$ is r-robust if for every pair of nonempty, disjoint subsets of $\mathcal{V}$, at least one of the subsets is r-reachable. By convention, the empty graph $(n = 0)$ is 0-robust and the trivial graph $(n = 1)$ is 1-robust.*

The property of (r, s)-robustness is based upon the notion of (r, s)-reachability. The definitions of (r, s)-reachability and (r, s)-robustness are as follows:

Definition 3.3 ([2]). *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be a nonempty, nontrivial, simple digraph on $n \geq 2$ nodes. Let $r, s \in \mathbb{Z}_{\geq 0}$, $0 \leq s \leq n$. Let S be a nonempty subset of $\mathcal{V}$, and define the set $\mathcal{X}_S^r = \{j \in S : |\mathcal{V}_j \backslash S| \geq r\}$. We say that S is an (r, s)-reachable set if there exist s nodes in S, each of which has at least r in-neighbors outside of S. More explicitly, S is (r, s)-reachable if $|\mathcal{X}_S^r| \geq s$.*

Definition 3.4 ([2]). *Let $r, s \in \mathbb{Z}_{\geq 0}$, $0 \leq s \leq n$. Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be a nonempty, nontrivial, simple digraph on $n \geq 2$ nodes. Define $\mathcal{X}_S^r = \{j \in S : |\mathcal{V}_j \backslash S| \geq r\}$, $S \subset \mathcal{V}$. The digraph $\mathcal{D}$ is (r, s)-robust if for every pair of nonempty, disjoint subsets $S_1, S_2 \subset \mathcal{V}$, at least one of the following conditions holds:*

A) $|\mathcal{X}_{S_1}^r| = |S_1|$,

B) $|\mathcal{X}_{S_2}^r| = |S_2|$,

C) $|\mathcal{X}_{S_1}^r| + |\mathcal{X}_{S_2}^r| \geq s$.

The properties of r- and (r, s)-robustness are used to quantify the ability of several resilient consensus algorithms to guarantee convergence of normally behaving agents in the presence of Byzantine and malicious adversaries, collectively referred to in this section as *misbehaving* agents [2, 76, 79, 84, 190]. Larger values of r and s generally imply the ability of networks applying these resilient algorithms to tolerate a greater number of misbehaving agents in the network. For a more detailed explanation of the properties of r-robustness and (r, s)-robustness, the reader is referred to [2, 97, 216].

It should be clear from Definitions 3.2 and 3.4 that determining r and (r, s)-robustness for a digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ by using an exhaustive search method is a combinatorial problem, which involves checking the reachabilities of all nonempty, disjoint subsets of $\mathcal{V}$. For notational purposes, we will denote the set of all possible pairs of nonempty, disjoint subsets of $\mathcal{V}$ as $\mathcal{T} \subset \mathcal{P}(\mathcal{V}) \times \mathcal{P}(\mathcal{V})$. More explicitly, $\mathcal{T}$ is defined as

$$\mathcal{T} = \big\{(S_1, S_2) \in \mathcal{P}(\mathcal{V}) \times \mathcal{P}(\mathcal{V}) : |S_1| > 0,\ |S_2| > 0,\ |S_1 \cap S_2| = 0\big\}. \tag{3.1}$$

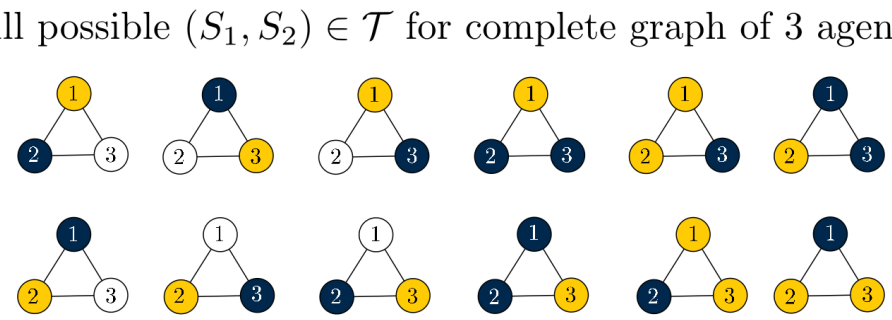


Figure 3.1: Depiction of all 12 possible (S_1, S_2) elements in $\mathcal{T}$ for a complete graph $\mathcal{D}$ of 3 agents. Each graph represents a different possible way of dividing $\mathcal{D}$ into sets S_1 and S_2. In each individual graph, yellow agents are in S_1, blue agents are in S_2, and white agents are in neither S_1 nor S_2.

It was shown in [96] that $|\mathcal{T}| = \sum_{p=2}^{n} \binom{n}{p}(2^p - 2)$.[1] As a simple example, Figure 3.1 depicts all elements of $\mathcal{T}$ for a graph of 3 agents, i.e. all possible ways to choose two nonempty, disjoint subsets from the graph.

When considering a particular digraph $\mathcal{D}$, there may be multiple values of r for which $\mathcal{D}$ is r-robust. Similarly, there may be multiple values of r and s for which $\mathcal{D}$ is (r, s)-robust. The following properties of robust graphs demonstrate this characteristic. Note that r-robustness is equivalent to $(r, 1)$-robustness; i.e. $\mathcal{D}$ is r-robust if and only if it is $(r, 1)$-robust [96, Property 5.21] [2, Section VII-B].

Property 3.1 ([219], Prop. 5.13)**.** *Let $\mathcal{D}$ be an arbitrary, simple digraph on n nodes. Suppose $\mathcal{D}$ is (r, s)-robust with $r \in \mathbb{N}$ and $s \in \{0, \dots, n\}$. Then $\mathcal{D}$ is also (r', s')-robust $\forall r' : \ 0 \leq r' \leq r$ and $\forall s' : \ 1 \leq s' \leq s$.*

Property 3.2 ([219], Prop. 5.20)**.** *Let $\mathcal{D}$ be an arbitrary, simple digraph on n nodes. Suppose $\mathcal{D}$ is (r, s)-robust with $r \in \mathbb{N}$ and $s \in \{1, \dots, n\}$. Then $\mathcal{D}$ is $(r - 1, s + 1)$-robust.*

As an example, if a digraph $\mathcal{D}_1$ is 4-robust, it is $(4, 1)$-robust, and therefore by Property 3.1 it is also simultaneously 3-robust, 2-robust, and 1-robust. In addition, if a digraph $\mathcal{D}_2$ is $(5, 4)$-robust, then it is simultaneously (r', s')-robust for all integers $0 \leq r' \leq 5$ and $0 \leq s' \leq 4$. Moreover, by Property 3.2, $\mathcal{D}_2$ is also $(4, 5)$-robust, $(3, 6)$-robust, $(2, 7)$-robust, and $(1, 8)$-robust. For notational purposes, we denote the set of all values for which a digraph $\mathcal{D}$ is (r, s)-robust as Θ, where $\Theta \subset \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0}$. By Definition 3.4, Θ is explicitly defined as

$$\Theta = \{(r, s) \in \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0} : \forall (S_1, S_2) \in \mathcal{T}, (|\mathcal{X}^r_{S_1}| = |S_1|) \text{ or } (|\mathcal{X}^r_{S_2}| = |S_2|) \text{ or } (|\mathcal{X}^r_{S_1}| + |\mathcal{X}^r_{S_2}| \geq s)\}. \tag{3.2}$$

[1] Since $(S_1, S_2) \in \mathcal{T} \implies (S_2, S_1) \in \mathcal{T}$, the total number of *unique* nonempty, disjoint subsets is $(1/2)|\mathcal{T}|$, denoted as $R(n)$ in [96].

Note that the conditions of (3.2) are simply an alternate way of expressing the conditions of Definition 3.4.

To characterize the resilience of graphs however, prior literature has generally been concerned with only a few particular values of r and s for which a given digraph is r- or (r, s)-robust. For r-robustness, the value of interest is the maximum integer r for which the given digraph is r-robust.

Definition 3.5. *We denote the maximum integer r for which a given digraph $\mathcal{D}$ is r-robust as $r_{\max}(\mathcal{D}) \in \mathbb{Z}_{\geq 0}$.*

Several resilient algorithms guarantee convergence of the normal agents when the adversary model is F-total or F-local in scope,[2] and the digraph is $(2F + 1)$-robust. The value of $r_{\max}(\mathcal{D})$ therefore determines the maximum adversary model under which these algorithms can operate successfully. Furthermore, all other values of r for which a digraph $\mathcal{D}$ is r-robust can be determined from $r_{\max}(\mathcal{D})$ by using Property 3.1.

For (r, s)-robustness, there are two (r, s) pairs of interest. The authors of [96] order the elements of Θ using a lexicographical total order, where elements are ranked by r value first and s value second. More specifically, $(r_1, s_1) \leq_{lex} (r_2, s_2)$ if and only if $\left[\begin{smallmatrix} r_2 - r_1 \\ s_2 - s_1 \end{smallmatrix}\right] \in \mathcal{V}_{lex}$, where K_{lex} is the lexicographic cone defined in Section 1.5. The algorithm $DetermineRobustness$ from [96] finds the maximum element of Θ with respect to this order. For notational clarity, we denote this maximum element as $(r^*, s^*) \in \Theta$.

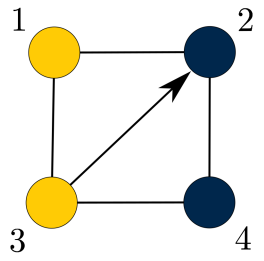
Definition 3.6. *Let Θ be defined as in (3.2). The element (r^*, s^*) is defined as the maximum element of Θ under the lexicographical order on $\mathbb{R}^2$.*

The other (r, s) pair of interest is $(F_{\max} + 1, F_{\max} + 1)$, where $F_{\max} = \max(\{F \in \mathbb{Z}_{\geq 0} : (F+1, F+1) \in \Theta\})$. Several resilient algorithms guarantee convergence of the normally behaving agents when the (malicious [2]) adversary model is F-total in scope and the digraph is $(F+1, F+1)$-robust. The value $F_{\max}$ determines the maximum malicious adversary model under which these algorithms can operate successfully. The value of $(F_{\max} + 1, F_{\max} + 1)$ does not always coincide with the (r^*, s^*)-robustness of the digraph. A simple counterexample is given in Figure 3.2, where the (r^*, s^*)-robustness of the graph is $(2, 1)$ but the value of $(F_{\max} + 1, F_{\max} + 1)$ is equal to $(1, 1)$.

Finally, the values of r for which a digraph can be r-robust lie within the interval $0 \leq r \leq \lceil n/2 \rceil$ [219, Property 5.19]. Since r-robustness is equivalent to $(r, 1)$-robustness, the values of r for which a graph can be (r, s)-robust fall within the same interval. The values of s for which a digraph can be (r, s)-robust lie within the interval $1 \leq s \leq n$.[3] However, we will use an abuse of notation by denoting a graph as $(r, 0)$-robust for a given $r \in \mathbb{Z}_{\geq 0}$ if the graph is not $(r, 1)$-robust.

[2] An F-total adversary model implies that there are at most $F \in \mathbb{Z}_{\geq 0}$ misbehaving agents in the entire network. An F-local adversary model implies that each normal agent has at most F misbehaving agents in its in-neighbor set.

[3] Footnote 8 in [2] offers an excellent explanation for restricting s to this range by convention.



$$S_1 = \{1, 3\}$$
$$S_2 = \{2, 4\}$$
$$\mathcal{X}^2_{S_1} = \{\emptyset\}$$
$$\mathcal{X}^2_{S_2} = \{2\}$$

Elements in Θ: $(1,4)$ $(2,4)$ $(1,3)$ $(2,3)$ $(1,2)$ $(2,2)$ $(1,1)$ $(2,1)$

Elements *not* in Θ

Figure 3.2: An example of the elements of Θ for a digraph $\mathcal{D}_1$. Since $|\mathcal{V}| = 4$, the possible values of r and s for which the digraph is (r, s)-robust fall within the range $0 \leq r \leq 2$, $1 \leq s \leq 4$. One possible pair of subsets S_1 and S_2 is depicted, which satisfies $|\mathcal{X}^2_{S_1}| \neq |S_1|$, $|\mathcal{X}^2_{S_2}| \neq |S_2|$, $|\mathcal{X}^2_{S_1}| = 0$ and $|\mathcal{X}^2_{S_2}| = 1$. By Definition 3.4, $\mathcal{D}_1$ therefore cannot be $(2, 2)$-robust, $(2, 3)$-robust, or $(2, 4)$-robust.

3.3 Robustness of k-Circulant Digraphs

As described previously, determining the r- or (r, s)-robustness of an arbitrary digraph (or undirected graph) is NP-hard in general [96]. This makes it difficult in general to determine how many adversaries a given network can tolerate under an MSR-type algorithm. It can also be difficult to determine the robustness of a given graph after adding or removing nodes or edges, unless the changes in the nodes or edges are limited to specific operations which allow robustness to be preserved from the initial graph to the final, altered graph [220]; however these operations assume that the robustness of the initial graph is known. This section introduces a class of graphs called *k-circulant digraphs*: digraphs whose structure and robustness can be calculated directly from two integer parameters n and k. First, we give the definition of circulant undirected graphs:

Definition 3.7. *An undirected graph of n nodes is called circulant if there exists a set*

$$\{a_1, a_2, \ldots, a_l \in \mathbb{Z}_{\geq 0} : a_1 < a_2 < \ldots < a_l < n\},$$

such that $(i, [i \pm a_1] \, mod \, n) \in \mathcal{E}_g, \ldots, (i, [i \pm a_l] \, mod \, n) \in \mathcal{E}_g$ [221]. We call such a graph an undirected circulant graph. *It should be noted that these graphs are constructed over the additive group of integers modulo n (the nodes $n + a$ and a are congruent modulo n).*

A similar concept for directed graphs is given as follows:

Definition 3.8. *A digraph of n nodes is called* circulant *if there exists a set*

$$\{a_1, a_2, \ldots, a_m : 0 < a_1 < a_2 < \ldots < a_m < n\}, \; m \in \mathbb{Z}_{\geq 0}$$

such that $(i, [i + a_1] \, mod \, n) \in \mathcal{E}_d, \ldots, (i, [i + a_m] \, mod \, n) \in \mathcal{E}_d$. *We denote such a graph as* $C_n(a_1, a_2, \ldots, a_m) = (\mathcal{V}, \mathcal{E}_d)$ *and call it a* directed circulant graph *or* circulant digraph.

The name *circulant* arises from the fact that the adjacency matrix for such a graph is a circulant matrix; i.e. a matrix where each row is defined by cyclically shifting every entry of the previous row one entry to the right. The matrix can therefore be defined by the entries of its first row [221, 222]. In general the name "circulant" has nothing to do with a physical circle or the physical orientation of its agents.

$$\begin{bmatrix} b_0 & b_1 & b_2 & \ddots & b_n \\ b_n & b_0 & b_1 & \ddots & b_{n-1} \\ b_{n-1} & b_n & b_0 & \ddots & b_{n-2} \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ b_1 & b_2 & b_3 & \ddots & b_0 \end{bmatrix}$$

Figure 3.3: The general structure of a circulant matrix. By defining the first row, the rest of the matrix is determined. Circulant digraphs have circulant adjacency matrices.

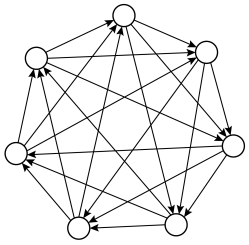


Figure 3.4: A 3-circulant digraph on 7 nodes, denoted $C_7\{1, 2, 3\}$. Nodes are arranged in a circle for visual clarity; in general the name "circulant" has nothing to do with the physical arrangement of the nodes or agents.

In this chapter, we analyze the robustness properties of a specific class of circulant digraphs which we call *k-circulant digraphs*:

Definition 3.9. *Let* $n \in \mathbb{Z}, \ n \geq 2$ *and let* $k \in \mathbb{Z} : 1 \leq k \leq n - 1$. *A* k*-circulant digraph is any circulant digraph of the form* $C_n(1, 2, 3, \ldots, k) = (\mathcal{V}, \mathcal{E}_d)$.

This type of graph is fully determined by the number of nodes n and by the parameter k, which determines the in- and out-neighbors of each node. In a graph without self-loops and without more than one edge between any two nodes, $1 \leq k \leq n - 1$. When $k = n - 1$, the graph becomes a complete graph.

Prior work has demonstrated that if an *undirected* line or ring graph is $2p$-connected, then it is at least $\left\lfloor \frac{p}{2} \right\rfloor$-robust [97, Thm 4]. However, this theorem does not apply to directed graphs because some ambiguity arises with the definition of vertex connectivity for digraphs. One possibility is to define the connectivity of a digraph as the minimum number of vertices whose removal results in the digraph's underlying graph being either disconnected or a trivial single vertex [217].[4] Under

[4]The underlying graph of a digraph is the undirected graph obtained by replacing all directed edges of the original digraph with undirected edges.

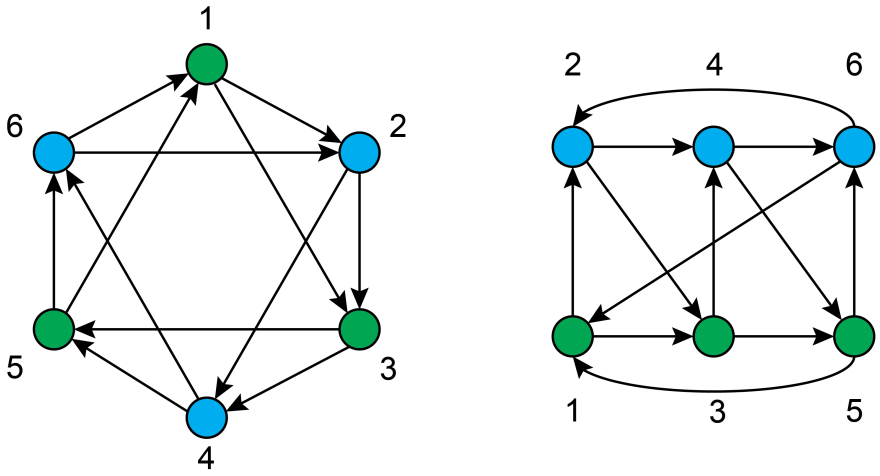


Figure 3.5: Example of a directed graph whose underlying graph is p-connected, but which is not $\lfloor \frac{p}{2} \rfloor$-robust. The graph shown has an underlying graph with vertex connectivity equal to 4. If the nodes of the graph are divided into the two nonempty, disjoint sets denoted by the green and blue colors, each node clearly only has one in-neighbor outside its set. This implies that the graph can be no more than 1-robust. Note that the two arrangements are the exact same graph; the second configuration is rearranged for clarity.

this definition, however, it can be shown that there exist digraphs which are not $\lfloor \frac{p}{2} \rfloor$-robust, but whose underlying graphs are p-connected. For example, under the definition just described the digraph in Figure 3.5 is 4-connected but is only 1-robust. Another measure of connectivity generalized to digraphs exists which involves the cardinality of a minimum vertex disconnecting set (see [223] and the equivalent definition of cutset in [224]). It can be shown however that graphs with arbitrarily large minimum vertex disconnecting sets are at most 1-robust, and therefore this metric cannot be used to determine robustness. A specific example is shown in Figure 3.6. In summary, neither of the aforementioned definitions allow [97, Thm 4] to be applied to digraphs; a different proof is therefore required.

The following Theorem demonstrates that the r-robustness of k-circulant digraphs is a function of the parameter k, regardless of the value of n.

Theorem 3.1. *The circulant digraph $C_n(1, \ldots, k) = (\mathcal{V}, \mathcal{E}_d)$ is $\lceil \frac{k}{2} \rceil$-robust, where $k < n$. Moreover, if $k = n - 1$ the graph is $\lceil \frac{n}{2} \rceil$-robust.*

Proof. From the definition of r-robustness it follows that if a graph is *not* σ-robust for some value $\sigma \in \mathbb{Z}$, then $\exists S_1, S_2$ such that $\forall i \in S_1,\ |\mathcal{V}_i \backslash S_1| < \sigma$ and $\forall j \in S_2,\ |\mathcal{V}_j \backslash S_2| < \sigma$. No graph can be less than 0-robust, hence $\sigma \geq 0$.

Suppose that a k-circulant graph is not σ-robust. Without loss of generality, this implies that there are S_1 and S_2 such that for any node $i \in S_1$ there exists a $b \in \mathbb{Z},\ 0 < b < \sigma$ such that node

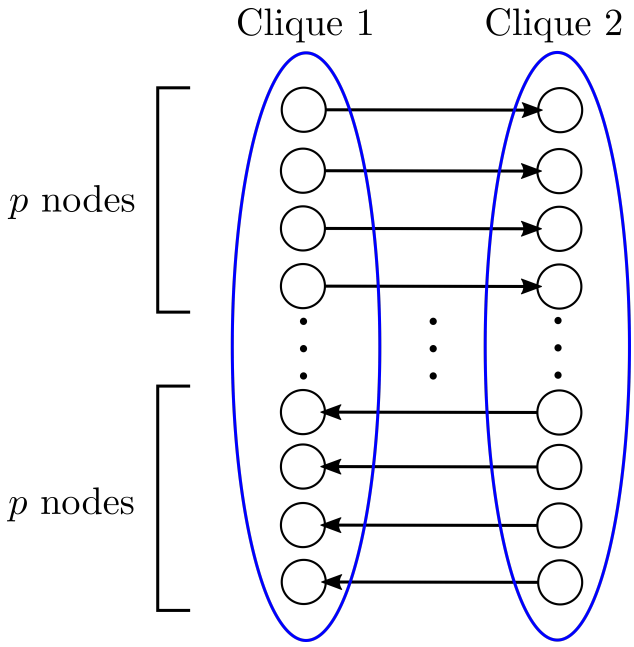


Figure 3.6: Counterexample showing that there exist digraphs with an arbitrarily large minimum vertex disconnecting set which are at most 1-robust. The class of digraphs in this figure are composed of two cliques with p directed edges going from clique 1 to clique 2 as shown, and p more directed edges going from clique 2 to clique 1. The size of a minimum vertex disconnecting set is therefore p. However, by Definition 3.2 the digraph can be at most 1-robust since no agent in either of the cliques has more than 1 in-neighbor outside its own clique.

$i+b \in S_2$ and nodes $\{i+1,\dots,i+b-1\} \notin S_2$. This can be seen by noting that S_2 is nonempty, and if $b \geq \sigma$ then $|\mathcal{V}_{i+b} \backslash S_2| \geq \sigma$, contradicting our initial assumption.

We next establish a lower bound on the value σ. Note that the in-neighbor set of i is $\mathcal{V}_i = \{i-k,\dots,i-1\}$ and the in-neighbor set of $i+b$ is $K_{i+b} = \{i+b-k,\dots,i+b-1\}$. The intersection of these two in-neighbors sets is $\mathcal{V}_i \cap \mathcal{V}_{i+b} = \{i+b-k,\dots,i-1\}$ Denote the number of S_1 nodes and the number of S_2 nodes in the set $\{i-k,\dots,i+b-k-1\}$ as α_1 and β_1 respectively, $\alpha_1, \beta_1 \in \mathbb{Z}_{\geq 0}$. Let $\gamma_1 \in \mathbb{Z}_{\geq 0}$ denote the nodes in $\{i-k,\dots,i+b-k-1\}$ as α_1 which are neither in S_1 nor S_2. Similarly, denote the number of S_1 nodes and S_2 nodes in the set $\{i+b-k,\dots,i-1\}$ as α_2 and β_2 respectively, $\alpha_2, \beta_2 \in \mathbb{Z}_{\geq 0}$. Let $\gamma_2 \in \mathbb{Z}_{\geq 0}$ denote the nodes in $\{i+b-k,\dots,i-1\}$ which are neither in S_1 nor S_2. Figure 3.7 presents a graphic depicting these sets and variables.

By our assumption that the graph is *not* σ-robust, observe that we must have

$$|\mathcal{V}_i \backslash S_1| = \beta_1 + \beta_2 + \gamma_1 + \gamma_2 \leq \sigma - 1, \tag{3.3}$$

$$|\mathcal{V}_{i+b} \backslash S_2| = \alpha_2 + \gamma_2 + b \leq \sigma - 1 \tag{3.4}$$

The constant b appears in (3.4) since nodes since nodes $\{i,\dots,i+b-1\} \notin S_2$ as previously discussed. Next, by definition of $\alpha_2, \beta_2, \gamma_2$ it holds that $\alpha_2 + \beta_2 + \gamma_2 = k - b$. Adding β_2 to both sides of (3.4) yields

$$\begin{aligned} \alpha_2 + \gamma_2 + b + \beta_2 &\leq \sigma - 1 + \beta_2, \\ k &\leq \sigma - 1 + \beta_2, \\ k - \sigma + 1 &\leq \beta_2. \end{aligned} \tag{3.5}$$

Substituting (3.5) into (3.3) yields

$$\begin{aligned} \beta_1 + \gamma_1 + \gamma_2 + k - \sigma + 1 &\leq \sigma - 1, \\ \beta_1 + \gamma_1 + \gamma_2 + k &\leq 2\sigma - 2. \end{aligned}$$

Using the fact that $\beta_1, \gamma_1, \gamma_2 \geq 0$ yields

$$\begin{aligned} k &\leq 2\sigma - 2, \\ \implies \frac{k}{2} + 1 &\leq \sigma. \end{aligned}$$

Since $\sigma \in \mathbb{Z}$, this implies that the smallest value of σ for which a k-circulant graph is *not* σ-robust is $\frac{k}{2}+1$ for even k and $\left\lceil \frac{k}{2} + 1 \right\rceil$ for odd k. Therefore a k-circulant graph must be $\left\lceil \frac{k}{2} \right\rceil$-robust.

Lastly, the case when $k = n-1$ implies a complete graph. From [75] it can be shown that such

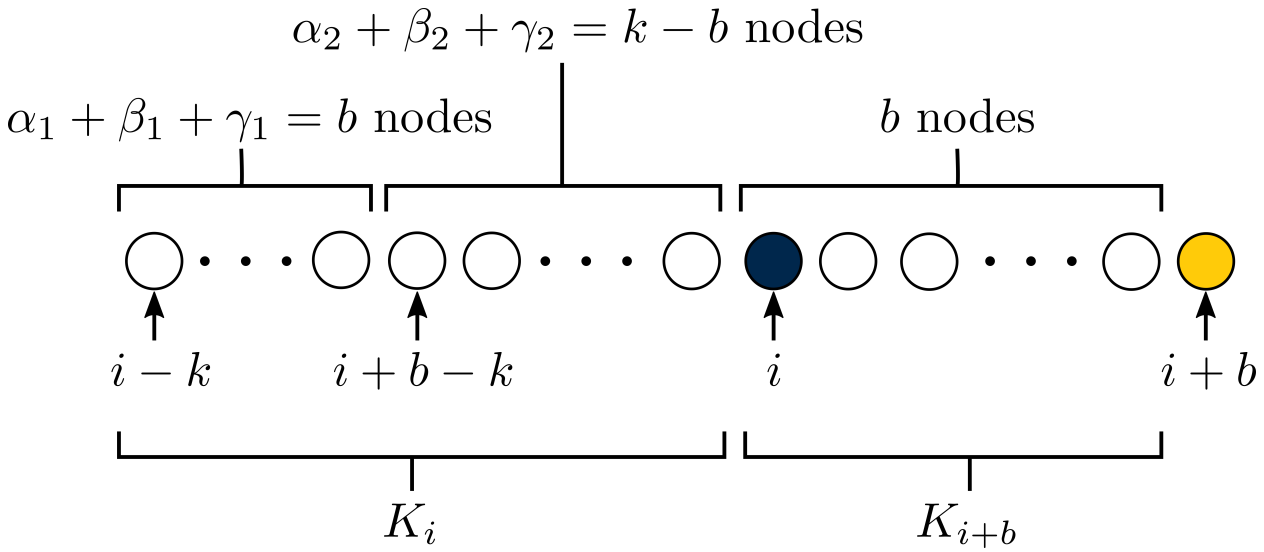


Figure 3.7: Visualization of the sets $\mathcal{V}_i$ and $\mathcal{V}_{i+b}$, and the values $\alpha_1,\ \alpha_2,\ \beta_1,\ \beta_2$. Here, $i \in S_1$ with S_1 represented by the color blue. From the proof, there exists a node $i + b \in S_2$, with S_2 represented by the color yellow. Nodes $i - k$ through $i - 1$ are either in S_1 or S_2, while nodes $i + 1$ through $i + b - 1$ are not in S_2, i.e. either in S_1 or neither in S_1 nor S_2.

graphs are $\lceil \frac{n}{2} \rceil$-robust. □

Since the robustness is a function of k only and not of n, it is worth noting that the r-robustness of circulant digraphs can be determined regardless of the size of the network. As a result these graphs can easily be scaled to any number of nodes while maintaining a desired robustness level. The main limitation is that $k \leq n - 1$, implying that a graph with a desired robustness will require a minimum number of nodes.

Next, we establish lower bounds on the values of r, s for which k-circulant digraphs are (r, s)-robust. In [220] a connection between r-robustness and (r, s)-robustness was given:

Lemma 3.1. *If $\mathcal{D} = (\mathcal{V}, \mathcal{E}_d)$ is $(r + s - 1)$-robust with $r \in \mathbb{Z}_{\geq 0}$, $s \in \mathbb{N}$, and $1 \leq r + s - 1 \leq \lceil \frac{n}{2} \rceil$, then $\mathcal{D}$ is (r, s)-robust.*

The proof is outlined in [220]. It should be noted that this is a sufficient condition only, and so the graph may actually have a higher (r, s)-robustness (e.g. consider a complete graph). We use this lemma to demonstrate the relationship between a lower bound of (r, s)-robustness of $C_n(1, \ldots, k)$-circulant digraphs and the parameter k:

Corollary 3.1. *The circulant digraph $C_n(1, \ldots, k) = (\mathcal{V}, \mathcal{E}_d)$ is at least $(\lfloor \frac{k+2}{4} \rfloor, \lfloor \frac{k+2}{4} \rfloor)$-robust if k is even and at least $(\lfloor \frac{k+3}{4} \rfloor, \lfloor \frac{k+3}{4} \rfloor)$-robust if k is odd.*

Proof. If k is even, then $C_n(1,\dots,k)$ is at least $\frac{k}{2}$-robust by Theorem 3.1. Since we are interested in establishing an upper bound F on the number of adversaries in the network, we seek to find the maximum value of F for which the network is $(F+1, F+1)$-robust. This implies $r = s$ for the network's (r,s)-robustness. Hence by Lemma 3.1, and letting $r = s$,

$$\begin{aligned} r + s - 1 &= \frac{k}{2} \\ r + s &= \frac{k+2}{2} \\ r = s &= \frac{k+2}{4} \\ &\geq \lfloor \frac{k+2}{4} \rfloor \end{aligned}$$

If k is odd, then $C_n(1,\dots,k)$ is at least $\lceil \frac{k}{2} \rceil$-robust by Theorem 3.1. Hence

$$\begin{aligned} r + s - 1 &= \lceil \frac{k}{2} \rceil = \frac{k+1}{2} \\ r + s &= \frac{k+3}{2} \\ r = s &= \frac{k+3}{4} \\ &\geq \lfloor \frac{k+3}{4} \rfloor \end{aligned}$$

□

3.3.1 Strong r-Robustness of k-Circulant Graphs

In addition to exhibiting r- and (r,s)-robustness properties, k-circulant digraphs also may exhibit the property of *strong r-robustness*. This additional characteristic is used, for example, by the method in Chapter 2, Section 2.3 for resilient leader-follower consensus scenarios. The definition of strong r-robustness was given in Chapter 2, Definition 2.5, but is repeated here for convenience:

Definition 3.10 (Strong r-robustness w.r.t. S [191])**.** *Let $r \in \mathbb{Z}_{\geq 0}$, $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be a digraph, and $S \subset \mathcal{V}$ be a nonempty subset. $\mathcal{D}$ is strongly r-robust w.r.t. S if for any nonempty subset $C \subseteq \mathcal{V} \backslash S$, C is r-reachable.*

Intuitively speaking, strong r-robustness w.r.t S considers the reachability of all possible subsets which are disjoint from the *fixed* subset $S \subset \mathcal{V}$. This differs from r-robustness which considers all possible nonempty, disjoint subsets of the node set $\mathcal{V}$. Another notable difference between strong r-robustness w.r.t. $S \subset \mathcal{V}$ and r-robustness is that given the subset S, it can be verified

in polynomial time whether a digraph is strongly robust w.r.t. S [194]. As used in Chapter 2, Section 2.3, the set S can represent a set of leaders propagating information to the remainder of the network.

When the subset S is properly selected, k-circulant graphs are strongly r-robust w.r.t. S. We now provide formal conditions on the set S under which these graphs are strongly r-robust. As per the definition of circulant graphs, we assume all agents are indexed $1, \ldots, n$. In our next proof we refer to sets of consecutive agents by index. An example is $P_S = \{2, 3, 4, 5, \ldots, 9\}$ in a network of $n = 15$ agents. Since the index numbers are defined on the set of integers modulo n, the set $P_S = \{14, 15, 1, 2\}$ would also be a set of consecutive agents in a network of $n = 15$ agents.

Theorem 3.2. *Let $k, r, n \in \mathbb{Z}_{\geq 0}$, $k > 0$. A k-circulant digraph $\mathcal{D} = C_n\{1, 2, \ldots, k\}$ is strongly r-robust with respect to $S \subset \mathcal{V}$ if $\mathcal{D}$ contains a set of consecutive agents by index $P_S \subset \mathcal{V}$ such that $|P_S| \leq k$ and $|P_S \cap S| \geq r$.*

Proof. Suppose $k \geq |P_S|$ and $|P_S \cap S| \geq r$. Without loss of generality, let the first agent in P_S be labeled as agent $(n - |P_S| + 1)$ and the last agent in P_S as agent n. We must show that all nonempty $C \subseteq \mathcal{V} \backslash S$ are r-reachable. If agent $1 \in C$ then C is r-reachable since $\{(n - |P_S| + 1), \ldots, n\} \subseteq \mathcal{V}_1$ which implies $|\mathcal{V}_1 \cap (\mathcal{V} \backslash C)| \geq r$. Next, suppose that agent $1 \notin C$ and $2 \in C$. Since $\{(n - |P_S| + 2), \ldots, 1\} \subseteq \mathcal{V}_2$, this implies that $|\mathcal{V}_2 \cap (\mathcal{V} \backslash C)| \geq |\mathcal{V}_1 \cap (\mathcal{V} \backslash C)| \geq r$ and therefore C is r-reachable. This reasoning can be continued inductively by assuming $\{1, \ldots p-1\} \notin C$, $p \in C$, and observing that $|\mathcal{V}_p \cap (\mathcal{V} \backslash C)| \geq |\mathcal{V}_{p-1} \cap (\mathcal{V} \backslash C)|$. Analyzing the remaining subsets of this form in the graph yields the result. Note that if p is ever the number of an agent in S, then we need not consider it ever being in C and the analysis can be continued with the next agent not in S. □

Similar results also hold for undirected k-circulant graphs:

Theorem 3.3. *Let $k, r, n \in \mathbb{Z}_{\geq 0}$, $k > 0$. An undirected k-circulant graph $\mathcal{G} = C_n\{\pm 1, \pm 2, \ldots, \pm k\}$ is strongly r-robust with respect to $S \subset \mathcal{V}$ if $\mathcal{G}$ contains a set of consecutive agents $P_S \subset \mathcal{V}$ such that $|P_S| \leq k$ and $|P_S \cap S| \geq r$.*

Proof. The same method as in Theorem 3.2 can be applied to prove the result. □

3.3.2 Implementation of W-MSR Algorithm on k-Circulant Digraphs

To demonstrate the robustness of k-circulant digraphs, we present simulations of agents in a k-circulant network running the W-MSR protocol. The network size is $n = 15$ nodes. Each agent in the graph has state $x[t] \in \mathbb{R}$, and each normal agent follows the W-MSR algorithm to update its

own state at each time step. The initial state value for each agent is a random value on the interval $[-50, 50]$.

Several models exist to describe the number and distribution of misbehaving nodes in a network, including the F-total, F-local, and f-fraction local models (see [190, 225]). For our simulations we consider an F-local model, meaning that any normal agent has at most F misbehaving agents in the set of its in-neighbors. Theorem 2 and Corollary 4 of [75] establish that $(2F + 1)$-robustness is a sufficient condition for a network using the W-MSR algorithm to achieve consensus among its normal nodes under an F-local model of misbehaving agents.

We consider two graphs on 15 nodes, each with different values of k. The first graph $\mathcal{D}_1$ has $k = 6$, implying $\mathcal{D}_1 = C_{15}(1, 2, \ldots, 6)$. Figure 3.8 shows the communication topology of $\mathcal{D}_1$. By Theorem 3.1 and Corollary 3.1, $\mathcal{D}_1$ is 3-robust, implying that consensus is guaranteed under a F-local malicious adversary model with $F = 1$. Figure 3.9 shows our simulation with $F = 1$ and with nodes 1 and 7 misbehaving. Note that any normal agent i has at most 1 misbehaving agent in $\mathcal{V}_i$.The red dotted lines represent the state values of misbehaving nodes, while the solid colored lines represent the state values of normal nodes. The normal nodes are clearly able to achieve consensus in the presence of the misbehaving nodes.

The second graph $\mathcal{D}_2$ has $k = 9$, and therefore is 5-robust which guarantees consensus under an F-local malicious adversary model with $F = 2$. Agents 1, 7, and 13 are misbehaving, which implies that any normal agent i has at most 2 misbehaving agents in its in-neighbor set K_i. Figure 3.10 shows the simulation results for the second graph with $F = 2$. Again, the normal agents are clearly able to achieve consensus in the presence of the misbehaving nodes. This second simulation also demonstrates the simplicity of changing the robustness of k-circulant digraphs. By varying k, the network's robustness can be increased or decreased to a desired level.

In summary, k-circulant digraphs allow for the creation of networks of arbitrary size whose r-robustness and strong r-robustness w.r.t. a set S can easily be determined. This makes them particularly useful in both resilient leaderless and leader-follower consensus scenarios such as those outlined in Chapter 2 and the works [87, 94, 95, 180]. The reader is referred to the simulations in Chapter 2 for additional simulations which use k-circulant digraphs for resilient consensus.

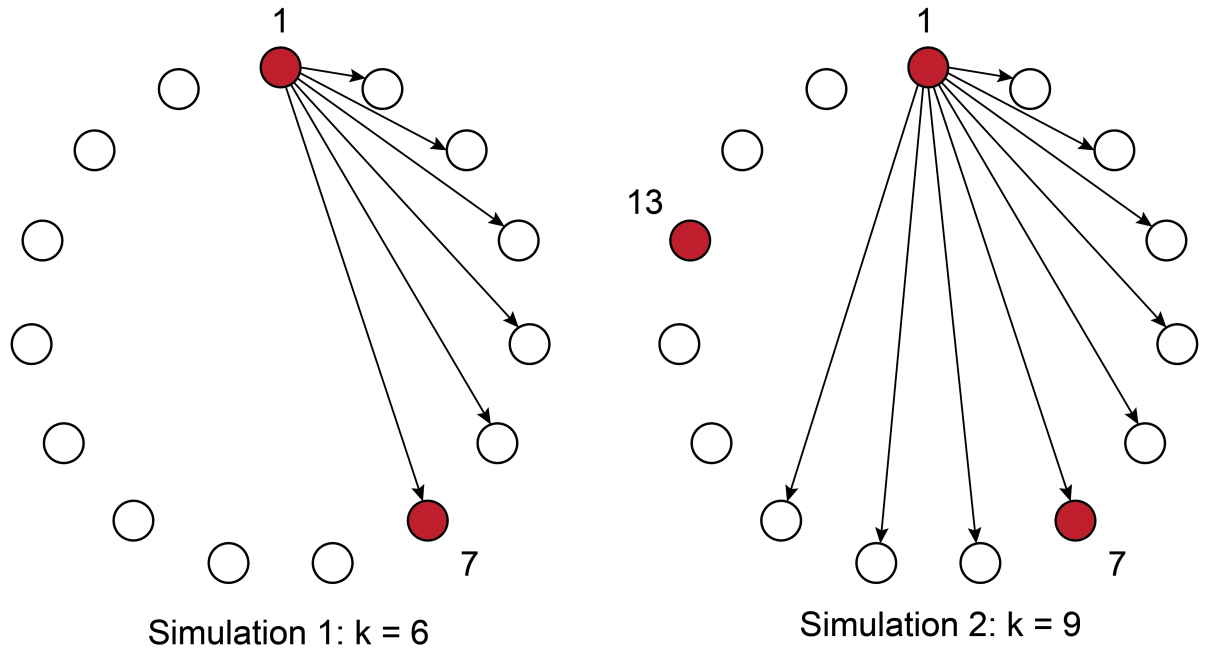


Figure 3.8: The network topology of digraphs D_1 and D_2. For sake of clarity, only the edges extending from one node are shown; in the actual graph, each node has the same pattern of edges extending from it. The first graph simulated is a $C_{15}\{1, \ldots, 6\}$ circulant digraph. The second is a $C_{15}\{1, \ldots, 9\}$ circulant digraph. In the first graph, nodes 1 and 7 are misbehaving. In the second, nodes 1, 7, and 13 are misbehaving. The nodes are visualized in a circular manner for ease of understanding rather than representing any kind of physical arrangement.

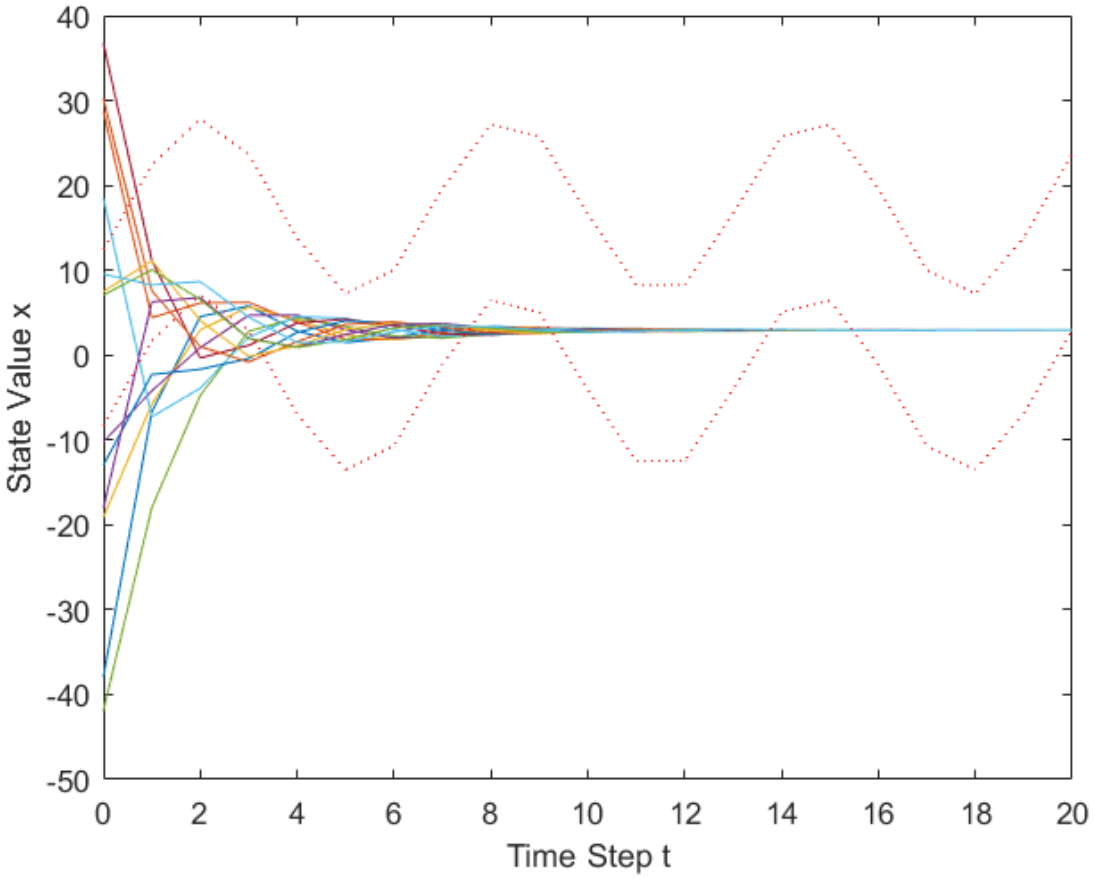


Figure 3.9: Simulation on the graph $\mathcal{D}_1 = C_{15}(1, 2, \ldots, 6)$. The dotted red lines represent the state trajectories of the misbehaving agents.

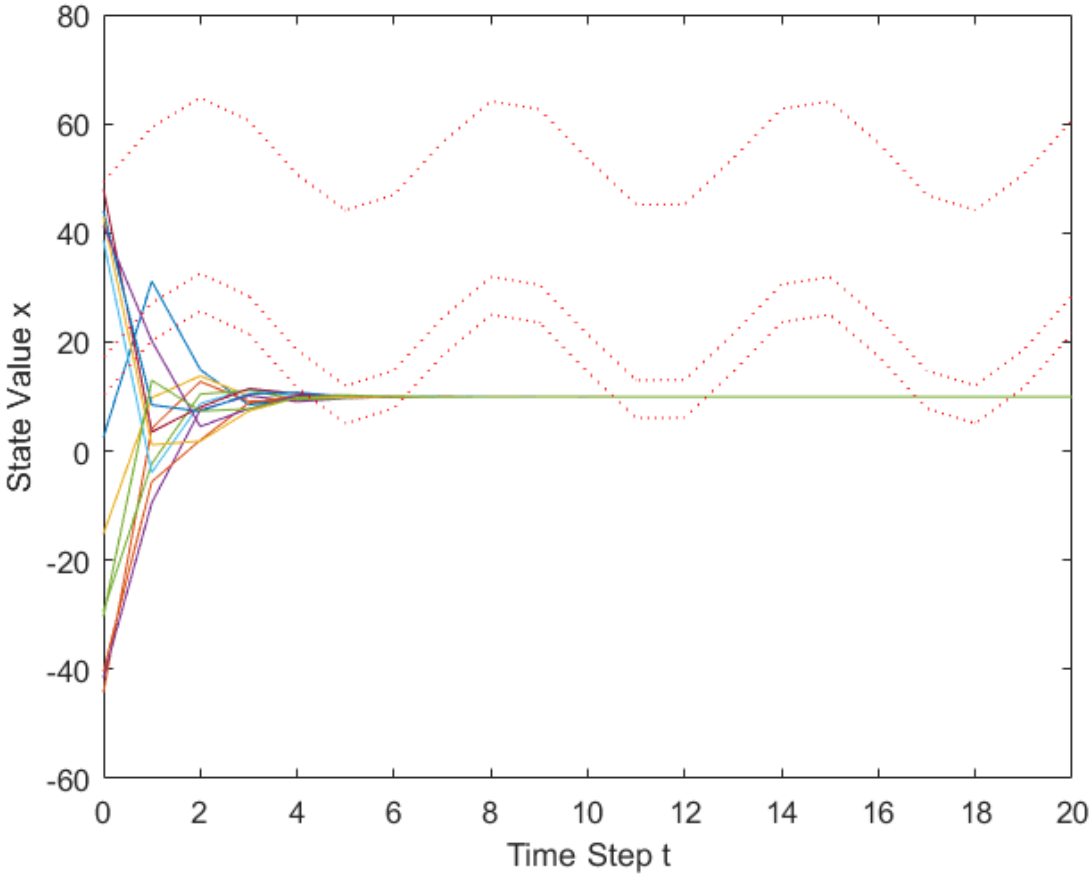


Figure 3.10: Simulation on the graph $D_2 = C_{15}(1, 2, \ldots, 9)$.

3.4 Determining r- and (r, s)-Robustness of Digraphs using Mixed Integer Programming

Despite all of the prior work done on graph construction methods and finding specific classes of graphs whose robustness is a function of graph parameters, there still exist a large number of directed and undirected graphs whose robustness cannot be determined by these prior methods. The algorithms in [96] were presented as a method to calculate the exact values of $r_{\max}$, r^*, s^*, and $F_{\max}$ for arbitrary directed or undirected graphs by exhaustively checking the appropriate robustness conditions for all nonempty, disjoint subset pairs $(S_1, S_2) \in \mathcal{T}$. Although these algorithms are guaranteed to find the exact values of these robustness properties, the lower bound on their runtime is exponential in the number of nodes n.[5] In addition, these algorithms are incapable of calculating approximate lower bounds on $r_{\max}$, r^*, s^*, and $F_{\max}$; the algorithms are only able to tighten upper bounds on these values as all pairs in $\mathcal{T}$ are searched through.

The purpose of this section is therefore to present methods using mixed integer linear programming to determine $r_{\max}(\mathcal{D})$, the (r^*, s^*)-robustness of $\mathcal{D}$, and the $(F_{\max}+1, F_{\max}+1)$-robustness of $\mathcal{D}$ for any nonempty, nontrivial, simple digraph $\mathcal{D}$.

Problem 3.1. *Given an arbitrary nonempty, nontrivial, simple digraph $\mathcal{D}$, determine the value of $r_{\max}(\mathcal{D})$.*

Problem 3.2. *Given an arbitrary nonempty, nontrivial, simple digraph $\mathcal{D}$, determine the (r^*, s^*)-robustness of $\mathcal{D}$.*

Problem 3.3. *Given an arbitrary nonempty, nontrivial, simple digraph $\mathcal{D}$, determine the $(F_{\max}+1, F_{\max}+1)$-robustness of $\mathcal{D}$.*

3.4.1 Determining r-Robustness using Mixed Integer Linear Programming

In this section we will demonstrate a method for solving Problem 3.1 using a mixed integer linear program (MILP) formulation. An MILP will be presented whose optimal value is equal to $r_{\max}(\mathcal{D})$ for any given nonempty, nontrivial, simple digraph $\mathcal{D}$.

First, an equivalent way of expressing the maximum robustness $r_{\max}(\mathcal{D})$ of a digraph $\mathcal{D}$ is derived. This equivalent expression will clarify how $r_{\max}(\mathcal{D})$ can be determined by means of an

[5] More precisely, for any graph with $r_{\max} > 0$ or $r^* > 0$ the algorithms in [96] will check all possible nonempty, disjoint subsets before terminating. If a subset pair $(S_1, S_2) \in \mathcal{T}$ is found which demonstrates that $r_{\max} \leq 0$ or $r^* \leq 0$, the search can be terminated immediately since $r_{\max}$ and r^* are nonnegative by definition.

optimization problem. Given an arbitrary, simple digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ and a subset $S \subset V$, the reachability function $\mathcal{R} : \mathcal{P}(\mathcal{V}) \to \mathbb{Z}_{\geq 0}$ is defined as follows:

$$\mathcal{R}(S) = \begin{cases} \max_{i \in S} |\mathcal{V}_i \backslash S|, & \text{if } S \neq \{\emptyset\}, \\ 0, & \text{if } S = \{\emptyset\}. \end{cases} \tag{3.6}$$

In other words, the function $\mathcal{R}(S)$ returns the maximum r for which S is r-reachable. Using this function, the following Lemma presents an optimization formulation which yields $r_{\max}(\mathcal{D})$:

Lemma 3.2. *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be an arbitrary nonempty, nontrivial, simple digraph with $|\mathcal{V}| = n$. Let $r_{\max}(\mathcal{D})$ be defined as in Definition 3.5. The following holds:*

$$\begin{aligned} r_{\max}(\mathcal{D}) = \min_{S_1, S_2 \in \mathcal{P}(\mathcal{V})} \quad & \max\left(\mathcal{R}(S_1), \mathcal{R}(S_2)\right) \\ \text{subject to} \quad & |S_1| > 0,\ |S_2| > 0,\ |S_1 \cap S_2| = 0. \end{aligned} \tag{3.7}$$

Proof. Note that $S_1, S_2 \in \mathcal{P}(\mathcal{V})$ and the three constraints of the RHS of (3.7) imply that the feasible set consists of all subsets S_1, S_2 such that $(S_1, S_2) \in \mathcal{T}$, as per (3.1). In addition, $\max\left(\mathcal{R}(S_1), \mathcal{R}(S_2)\right) = m$ implies $\mathcal{R}(S_1) = m$ or $\mathcal{R}(S_2) = m$. Let (S_1^*, S_2^*) be a minimizer of (3.7). Then $\max\left(\mathcal{R}(S_1^*), \mathcal{R}(S_2^*)\right) \leq \max\left(\mathcal{R}(S_1), \mathcal{R}(S_2)\right)\ \forall (S_1, S_2) \in \mathcal{T}$. Therefore $\forall (S_1, S_2) \in \mathcal{T}$, either $\mathcal{R}(S_1) \geq \max\left(\mathcal{R}(S_1^*), \mathcal{R}(S_2^*)\right)$ or $\mathcal{R}(S_2) \geq \max\left(\mathcal{R}(S_1^*), \mathcal{R}(S_2^*)\right)$. This satisfies the definition of r-robustness as per Definition 3.2, therefore $\mathcal{D}$ is at least $\max\left(\mathcal{R}(S_1^*), \mathcal{R}(S_2^*)\right)$-robust. This implies $r_{\max}(\mathcal{D}) \geq \max\left(\mathcal{R}(S_1^*), \mathcal{R}(S_2^*)\right)$.

We next show that $r_{\max}(\mathcal{D}) = \max\left(\mathcal{R}(S_1^*), \mathcal{R}(S_2^*)\right)$. We prove by contradiction. Recall from Definition 3.5 that $r_{\max}(\mathcal{D})$ is the maximum integer r for which $\mathcal{D}$ is r-robust, which means $\mathcal{D}$ is $r_{\max}(\mathcal{D})$-robust by definition. Suppose $r_{\max}(\mathcal{D}) > \max\left(\mathcal{R}(S_1^*), \mathcal{R}(S_2^*)\right)$. This implies $\mathcal{R}(S_1^*) < r_{\max}(\mathcal{D})$ and $\mathcal{R}(S_2^*) < r_{\max}(\mathcal{D})$. Since the nonempty, disjoint subsets $(S_1^*, S_2^*) \in \mathcal{T}$ satisfy $\mathcal{R}(S_1^*) < r_{\max}(\mathcal{D})$ and $\mathcal{R}(S_2^*) < r_{\max}(\mathcal{D})$, by the negation of Definition 3.2 this implies that $\mathcal{D}$ is *not* $r_{\max}(\mathcal{D})$-robust. However, this contradicts the definition of $r_{\max}(\mathcal{D})$ being the largest integer for which $\mathcal{D}$ is r-robust (Definition 3.5). This provides the desired contradiction; therefore $r_{\max}(\mathcal{D}) = \max\left(\mathcal{R}(S_1^*), \mathcal{R}(S_2^*)\right)$. □

Remark 3.1. *Using the definition of $\mathcal{T}$ in (3.1), the constraints on the RHS of (3.7) can be made implicit [189, section 4.1.3] as follows:*

$$r_{\max}(\mathcal{D}) = \min_{(S_1, S_2) \in \mathcal{T}} \quad \max\left(\mathcal{R}(S_1), \mathcal{R}(S_2)\right). \tag{3.8}$$

We demonstrate next that the objective function of (3.7) can be expressed as a function of the network Laplacian matrix. Recall that $n = |\mathcal{V}|$ and that $\{0,1\}^n$ represents a binary vector of dimension n. The indicator vector $\boldsymbol{\sigma}(\cdot) : \mathcal{P}(\mathcal{V}) \to \{0,1\}^n$ is defined as follows: for all $S \in \mathcal{P}(\mathcal{V})$,

$$\sigma_j(S) = \begin{cases} 1 & \text{if } j \in S \\ 0 & \text{if } j \notin S \end{cases}, \; j = \{1, \ldots, n\}. \tag{3.9}$$

In other words the jth entry of $\boldsymbol{\sigma}(S)$ is 1 if the node with index j is a member of the set $S \in \mathcal{P}(\mathcal{V})$, and zero otherwise. It is straightforward to verify that $\boldsymbol{\sigma} : \mathcal{P}(\mathcal{V}) \to \{0,1\}^n$ is a bijection. Therefore given $\boldsymbol{x} \in \{0,1\}^n$, the set $\boldsymbol{\sigma}^{-1}(\boldsymbol{x}) \in \mathcal{P}(\mathcal{V})$ is defined by

$$\boldsymbol{\sigma}^{-1}(\boldsymbol{x}) = \{j \in \mathcal{V} : x_j = 1\}. \tag{3.10}$$

Finally, observe that for any $S \in \mathcal{P}(\mathcal{V})$, $|S| = \mathbf{1}^T \boldsymbol{\sigma}(S)$. The following Lemma demonstrates that for any $S \in \mathcal{P}(\mathcal{V})$, the function $\mathcal{R}(S)$ can be determined as an affine function of the network Laplacian matrix and the indicator vector of S:

Lemma 3.3. *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be an arbitrary nonempty, nontrivial, simple digraph, let L be the Laplacian matrix of $\mathcal{D}$, and let $S \in \mathcal{P}(\mathcal{V})$. Then the following holds for all $j \in \{1, \ldots, n\}$:*

$$L_j \boldsymbol{\sigma}(S) = \begin{cases} |\mathcal{V}_j \backslash S|, & \text{if } j \in S, \\ -|\mathcal{V}_j \cap S|, & \text{if } j \notin S, \end{cases} \tag{3.11}$$

where L_j is the jth row of L. Furthermore,

$$\mathcal{R}(S) = \max_j L_j \boldsymbol{\sigma}(S), \; j \in \{1, \ldots, n\}. \tag{3.12}$$

Proof. The term $\boldsymbol{\sigma}(S)$ is shortened to $\boldsymbol{\sigma}$ for brevity. Recall that the entry in the jth row and ith column of L is denoted $L_{j,i}$. The definition of L from (1.2) implies

$$\begin{aligned} L_j \boldsymbol{\sigma} &= (L_{j,j})\sigma_j + \sum_{q \in \{1,\ldots,n\} \backslash j} (L_{j,q})\sigma_q \\ &= |\mathcal{V}_j|\sigma_j - \sum_{q \in \mathcal{V}_j \cap S} \sigma_q - \sum_{q \in \mathcal{V}_j \backslash S} \sigma_q. \end{aligned} \tag{3.13}$$

Since by (3.9), $q \in S$ implies $\sigma_q = 1$, the term $\sum_{q \in \mathcal{V}_j \cap S} \sigma_q = |\mathcal{V}_j \cap S|$. In addition, since $q \notin S$ implies $\sigma_q = 0$, the term $\sum_{q \in \mathcal{V}_j \backslash S} \sigma_q = 0$. By this, equation (3.13) simplifies to $L_j \boldsymbol{\sigma} = |\mathcal{V}_j|\sigma_j - |\mathcal{V}_j \cap S|$.

The value of the term $|\mathcal{V}_j|\sigma_j$ depends on whether $j \in S$ or $j \notin S$. If $j \in S$, then $\sigma_j = 1$, implying $L_j\boldsymbol{\sigma} = |\mathcal{V}_j| - |\mathcal{V}_j \cap S| = (|\mathcal{V}_j \cap S| + |\mathcal{V}_j \backslash S|) - |\mathcal{V}_j \cap S| = |\mathcal{V}_j \backslash S|$. If $j \notin S$, then $\sigma_j = 0$ implying $L_j\boldsymbol{\sigma} = -|\mathcal{V}_j \cap S|$. This proves the result for equation (3.11).

To prove (3.12), we first consider nonempty sets $S \in \mathcal{P}(\mathcal{V})\backslash\{\emptyset\}$. By the results above and (3.6), the maximum reachability of any $S \in \mathcal{P}(\mathcal{V})\backslash\{\emptyset\}$ is

$$\mathcal{R}(S) = \max_{j \in S} |\mathcal{V}_j \backslash S| = \max_{j \in S}(L_j\boldsymbol{\sigma}(S)). \tag{3.14}$$

By its definition, $\mathcal{R}(S) \geq 0$. Observe that if $j \in S$ then $L_j\boldsymbol{\sigma}(S) = |\mathcal{V}_j \backslash S| \geq 0$, implying $\max_{j \in S} L_j\boldsymbol{\sigma}(S) \geq 0$. Conversely, if an agent j is *not* in the set S, then the function $L_j\boldsymbol{\sigma}(S)$ takes the nonpositive value $-|\mathcal{V}_j \cap S|$. This implies $\max_{j \notin S} L_j\boldsymbol{\sigma}(S) \leq 0$. By these arguments, we therefore have $\max_{j \notin S} L_j\boldsymbol{\sigma}(S) \leq 0 \leq \max_{j \in S} L_j\boldsymbol{\sigma}(S)$, which implies

$$\begin{aligned} \max_{j \in \{1,\ldots,n\}} L_j\boldsymbol{\sigma}(S) &= \max\left((\max_{j \in S} L_j\boldsymbol{\sigma}(S)), (\max_{j \notin S} L_j\boldsymbol{\sigma}(S))\right) \\ &= \max_{j \in S} L_j\boldsymbol{\sigma}(S). \end{aligned} \tag{3.15}$$

Therefore by equations (3.15) and (3.14), the maximum reachability of S is found by the expression

$$\mathcal{R}(S) = \max_j(L_j\boldsymbol{\sigma}(S)),\ j \in \{1, \ldots, n\}. \tag{3.16}$$

Lastly, if $S = \emptyset$, then by (3.6) we have $\mathcal{R}(S) = 0$. In addition, $\boldsymbol{\sigma}(S) = 0$, implying that $\max_j L_j\boldsymbol{\sigma}(S) = 0 = \mathcal{R}(S),\ j \in \{1, \ldots, n\}$. □

Using Lemma 3.3, it will next be shown that the objective function of (3.7) can be rewritten as the maximum over a set of affine functions:

Lemma 3.4. *Consider an arbitrary, nonempty, nontrivial, simple digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$. Let L be the Laplacian matrix of $\mathcal{D}$, and let L_i be the ith row of L. Let $\mathcal{T}$ be defined as in (3.1). Then for all $(S_1, S_2) \in \mathcal{T}$ the following holds:*

$$\max\left(\mathcal{R}(S_1), \mathcal{R}(S_2)\right) = \max\left(\max_{i \in \{1,\ldots,n\}} \left(L_i\boldsymbol{\sigma}(S_1)\right), \max_{j \in \{1,\ldots,n\}} \left(L_j\boldsymbol{\sigma}(S_2)\right)\right)$$

Proof. By Lemma 3.3, $\mathcal{R}(S_1) = \max_i L_i\boldsymbol{\sigma}(S_1)$ and $\mathcal{R}(S_2) = \max_j L_j\boldsymbol{\sigma}(S_2)$ for $i, j \in \{1, \ldots, n\}$. The result follows. □

From Lemma 3.2, Lemma 3.4, and Remark 3.1, we can immediately conclude that $r_{\max}(\mathcal{D})$

satisfies

$$r_{\max}(\mathcal{D}) = \min_{(S_1,S_2)\in\mathcal{T}} \max\left(\max_i \left(L_i\boldsymbol{\sigma}(S_1)\right), \max_j \left(L_j\boldsymbol{\sigma}(S_2)\right)\right). \tag{3.17}$$

Note that the terms $\boldsymbol{\sigma}(S_1)$ and $\boldsymbol{\sigma}(S_2)$ are each n-dimensional binary vectors. Letting $\boldsymbol{b}^1 = \boldsymbol{\sigma}(S_1)$ and $\boldsymbol{b}^2 = \boldsymbol{\sigma}(S_2)$, the objective function of (3.17) can be written as

$$\max\left(\max_i \left(L_i\boldsymbol{b}^1\right), \max_j \left(L_j\boldsymbol{b}^2\right)\right).$$

Every pair $(S_1, S_2) \in \mathcal{T}$ can be mapped into a pair of binary vectors $(\boldsymbol{b}^1, \boldsymbol{b}^2)$ by the function $\Sigma : \mathcal{T} \to \{0,1\}^n \times \{0,1\}^n$, where $\Sigma(S_1, S_2) = (\boldsymbol{\sigma}(S_1), \boldsymbol{\sigma}(S_2)) = (\boldsymbol{b}^1, \boldsymbol{b}^2)$. By determining the image of $\mathcal{T}$ under $\Sigma(\cdot,\cdot)$, the optimal value of (3.17) can be found by minimizing over pairs of binary vectors $(\boldsymbol{b}_1, \boldsymbol{b}_2) \in \Sigma(\mathcal{T})$ directly. Using binary vector variables instead of set variables (S_1, S_2) will allow (3.17) to be written directly in an MILP form. Towards this end, the following Lemma defines the set $\Sigma(\mathcal{T})$:

Lemma 3.5. *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be an arbitrary nonempty, nontrivial, simple digraph, and let $\mathcal{T}$ be defined as in (3.1). Define the function $\Sigma : \mathcal{T} \to \{0,1\}^n \times \{0,1\}^n$ as*

$$\Sigma(S_1, S_2) = (\boldsymbol{\sigma}(S_1), \boldsymbol{\sigma}(S_2)),\ (S_1, S_2) \in \mathcal{T}. \tag{3.18}$$

Define the set $\mathcal{B} \subset \{0,1\}^n \times \{0,1\}^n$ as

$$\mathcal{B} = \left\{(\boldsymbol{b}^1, \boldsymbol{b}^2) \in \{0,1\}^n \times \{0,1\}^n : 1 \le \mathbf{1}^T\boldsymbol{b}^1 \le (n-1),\ 1 \le \mathbf{1}^T\boldsymbol{b}^2 \le (n-1),\ \boldsymbol{b}^1 + \boldsymbol{b}^2 \preceq \mathbf{1}\right\}. \tag{3.19}$$

Then both of the following statements hold:

1. *The image of $\mathcal{T}$ under Σ is equal to $\mathcal{B}$, i.e. $\Sigma(\mathcal{T}) = \mathcal{B}$*
2. *The mapping $\Sigma : \mathcal{T} \to \mathcal{B}$ is a bijection.*

Proof. We prove *1)* by showing first that $\Sigma(\mathcal{T}) \subseteq \mathcal{B}$, and then $\mathcal{B} \subseteq \Sigma(\mathcal{T})$. Any $(S_1, S_2) \in \mathcal{T}$ satisfies $|S_1| > 0$, $|S_2| > 0$, $|S_1 \cap S_2| = 0$ as per (3.1). Observe that

$$|S_1| > 0 \implies \mathbf{1}^T\boldsymbol{\sigma}(S_1) \ge 1,$$
$$|S_2| > 0 \implies \mathbf{1}^T\boldsymbol{\sigma}(S_2) \ge 1.$$

Because $S_1, S_2 \subset \mathcal{V}$ and $|S_1 \cap S_2| = 0$, then $|S_1| < n$. Otherwise if $|S_1| = n$ then either $|S_2| = 0$ or $|S_1 \cap S_2| \neq 0$, which both contradict the definition of $\mathcal{T}$. Therefore $|S_1| < n$, and by similar arguments $|S_2| < n$. Observe that

$$\begin{aligned} |S_1| < n &\implies \mathbf{1}^T \boldsymbol{\sigma}(S_1) \leq n-1, \\ |S_2| < n &\implies \mathbf{1}^T \boldsymbol{\sigma}(S_2) \leq n-1. \end{aligned}$$

Finally, $|S_1 \cap S_2| = 0$ implies that $j \in S_1 \implies j \notin S_2$ and $j \in S_2 \implies j \notin S_1 \; \forall j \in \{1, \ldots, n\}$. Therefore $\sigma_j(S_1) = 1 \implies \sigma_j(S_2) = 0$ and $\sigma_j(S_2) = 1 \implies \sigma_j(S_1) = 0$. This implies that

$$|S_1 \cap S_2| = 0 \implies \boldsymbol{\sigma}(S_1) + \boldsymbol{\sigma}(S_2) \preceq \mathbf{1}.$$

Therefore for all $(S_1, S_2) \in \mathcal{T}$, $(\boldsymbol{\sigma}(S_1), \boldsymbol{\sigma}(S_2)) = \Sigma(S_1, S_2)$ satisfies the constraints of the set on the RHS of (3.19). This implies that $\Sigma(\mathcal{T}) \subseteq \mathcal{B}$.

Next, we show $\mathcal{B} \subseteq \Sigma(\mathcal{T})$ by showing that for all $(\boldsymbol{b}^1, \boldsymbol{b}^2) \in \mathcal{B}$, there exists an $(S_1, S_2) \in \mathcal{T}$ such that $(\boldsymbol{b}^1, \boldsymbol{b}^2) = \Sigma(S_1, S_2)$. Choose any $(\boldsymbol{b}^1, \boldsymbol{b}^2) \in \mathcal{B}$ and define sets (S_1, S_2) as follows:

$$\begin{aligned} b_j^1 = 1 &\implies j \in S_1, \; j \in \{1, \ldots, n\}, \\ b_j^1 = 0 &\implies j \notin S_1, \\ b_j^2 = 1 &\implies j \in S_2, \\ b_j^2 = 0 &\implies j \notin S_2. \end{aligned} \tag{3.20}$$

For the considered sets (S_1, S_2), $1 \leq \mathbf{1}^T \boldsymbol{b}^1$ implies $|S_1| > 0$ and $1 \leq \mathbf{1}^T \boldsymbol{b}^2$ implies $|S_2| > 0$. In addition since $\boldsymbol{b}^1 + \boldsymbol{b}^2 \preceq \mathbf{1}$, we have $b_j^1 = 1 \implies b_j^2 = 0$ and $b_j^2 = 1 \implies b_j^1 = 0$. By our choice of S_1 and S_2, we have $b_j^1 = 1 \implies j \in S_1$, and from previous arguments $b_j^1 = 1 \implies b_j^2 = 0 \implies j \notin S_2$. Similar reasoning can be used to show that $b_j^2 = 1 \implies j \notin S_1$. These arguments imply that $|S_1 \cap S_2| = 0$. Consequently, (S_1, S_2) satisfies all the constraints of $\mathcal{T}$ and is therefore an element of $\mathcal{T}$. Clearly, by (3.20) we have $\Sigma(S_1, S_2) = (\boldsymbol{b}^1, \boldsymbol{b}^2)$, which shows that there exists an $(S_1, S_2) \in \mathcal{T}$ such that $(\boldsymbol{b}^1, \boldsymbol{b}^2) = \Sigma(S_1, S_2)$. Since this holds for all $(\boldsymbol{b}^1, \boldsymbol{b}^2) \in \mathcal{B}$, this implies $\mathcal{B} \subseteq \Sigma(\mathcal{T})$. Therefore $\Sigma(\mathcal{T}) = \mathcal{B}$.

We next prove *2)*. Since $\Sigma(\mathcal{T}) = \mathcal{B}$, the function $\Sigma : \mathcal{T} \to \mathcal{B}$ is surjective. To show that it is injective, consider any $\Sigma(S_1, S_2) \in \mathcal{B}$ and $\Sigma(\bar{S}_1, \bar{S}_2) \in \mathcal{B}$ such that $\Sigma(S_1, S_2) = \Sigma(\bar{S}_1, \bar{S}_2)$. This implies $(\boldsymbol{\sigma}(S_1), \boldsymbol{\sigma}(S_2)) = (\boldsymbol{\sigma}(\bar{S}_1), \boldsymbol{\sigma}(\bar{S}_2))$. Note that $(\boldsymbol{\sigma}(S_1), \boldsymbol{\sigma}(S_2)) = (\boldsymbol{\sigma}(\bar{S}_1), \boldsymbol{\sigma}(\bar{S}_2))$ if and only if $\boldsymbol{\sigma}(S_1) = \boldsymbol{\sigma}(\bar{S}_1)$ and $\boldsymbol{\sigma}(S_2) = \boldsymbol{\sigma}(\bar{S}_2)$. Since the indicator function $\boldsymbol{\sigma} : \mathcal{P}(\mathcal{V}) \to \{0, 1\}^n$ is itself injective, this implies $S_1 = \bar{S}_1$ and $S_2 = \bar{S}_2$, which implies $(S_1, S_2) = (\bar{S}_1, \bar{S}_2)$. Therefore $\Sigma : \mathcal{T} \to \mathcal{B}$ is injective. □

Using this result, we now present the following mixed integer program which solves Problem 3.1:

Theorem 3.4. *Let $\mathcal{D}$ be an arbitrary nonempty, nontrivial, simple digraph and let L be the Laplacian matrix of $\mathcal{D}$. The maximum r-robustness of $\mathcal{D}$, denoted $r_{\max}(\mathcal{D})$, is obtained by solving the following minimization problem:*

$$\begin{aligned} r_{\max}(\mathcal{D}) = \min_{\boldsymbol{b}^1,\boldsymbol{b}^2} \quad & \max\left(\max_i\left(L_i\boldsymbol{b}^1\right), \max_j\left(L_j\boldsymbol{b}^2\right)\right) \\ \text{subject to} \quad & \boldsymbol{b}^1 + \boldsymbol{b}^2 \preceq \mathbf{1} \\ & 1 \leq \mathbf{1}^T\boldsymbol{b}^1 \leq (n-1) \\ & 1 \leq \mathbf{1}^T\boldsymbol{b}^2 \leq (n-1) \\ & \boldsymbol{b}^1, \boldsymbol{b}^2 \in \{0,1\}^n. \end{aligned} \tag{3.21}$$

Furthermore, (3.21) is equivalent to the following mixed integer linear program:

$$\begin{aligned} r_{\max}(\mathcal{D}) = \min_{t,\boldsymbol{b}} \quad & t \\ \text{subject to} \quad & 0 \leq t,\ t \in \mathbb{R},\ \boldsymbol{b} \in \mathbb{Z}^{2n} \\ & \begin{bmatrix} L & \mathbf{0} \\ \mathbf{0} & L \end{bmatrix} \boldsymbol{b} \preceq t \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \\ & \mathbf{0} \preceq \boldsymbol{b} \preceq \mathbf{1} \\ & \begin{bmatrix} I_{n\times n} & I_{n\times n} \end{bmatrix} \boldsymbol{b} \preceq \mathbf{1} \\ & 1 \leq \begin{bmatrix} \mathbf{1}^T & \mathbf{0} \end{bmatrix} \boldsymbol{b} \leq n-1 \\ & 1 \leq \begin{bmatrix} \mathbf{0} & \mathbf{1}^T \end{bmatrix} \boldsymbol{b} \leq n-1 \end{aligned} \tag{3.22}$$

Proof. From Lemmas 3.2 and 3.4 we have

$$\begin{aligned} r_{\max}(\mathcal{D}) = \min_{S_1,S_2\in\mathcal{P}(\mathcal{V})} \quad & \max\left(\max_i\left(L_i\boldsymbol{\sigma}(S_1)\right), \max_j\left(L_j\boldsymbol{\sigma}(S_2)\right)\right) \\ \text{subject to} \quad & |S_1| > 0,\ |S_2| > 0,\ |S_1 \cap S_2| = 0, \end{aligned}$$

for $i, j \in \{1, \ldots, n\}$. As per Remark 3.1, the definition of $\mathcal{T}$ can be used to make the constraints implicit:

$$r_{\max}(\mathcal{D}) = \min_{(S_1,S_2)\in\mathcal{T}} \max\left(\max_i\left(L_i\boldsymbol{\sigma}(S_1)\right), \max_j\left(L_j\boldsymbol{\sigma}(S_2)\right)\right), \tag{3.23}$$

for $i, j \in \{1, \dots, n\}$. Since $\Sigma : \mathcal{T} \to \mathcal{B}$ is a bijection by Lemma 3.5, (3.23) is equivalent to

$$r_{\max}(\mathcal{D}) = \min_{(\boldsymbol{b}^1, \boldsymbol{b}^2) \in \mathcal{B}} \quad \max\left(\max_i \left(L_i \boldsymbol{b}^1\right), \max_j \left(L_j \boldsymbol{b}^2\right)\right), i, j \in \{1, \dots, n\}. \tag{3.24}$$

Making the constraints of (3.24) explicit yields (3.21).

Next, we prove that (3.22) is equivalent to (3.21). The variables $\boldsymbol{b}^1$ and $\boldsymbol{b}^2$ from (3.21) are combined into the variable $\boldsymbol{b} \in \mathbb{Z}^{2n}$ in (3.22); i.e. $\boldsymbol{b} = \left[\, (\boldsymbol{b}^1)^T \ (\boldsymbol{b}^2)^T \,\right]^T$. The first and third constraints of (3.22) restrict $\boldsymbol{b} \in \{0, 1\}^{2n}$. Next, it can be demonstrated [189, Chapter 4] that the formulation $\min_{\boldsymbol{x}} \max_i(x_i)$ is equivalent to $\min_{t,x} t$ subject to $0 \leq t$, $\boldsymbol{x} \preceq t\mathbf{1}$.

Reformulating the objective of the RHS of (3.21) in this way yields the objective and first two constraints of (3.22):

$$\begin{aligned} \min_{t, \boldsymbol{b}} \quad & t \\ \text{subject to} \quad & 0 \leq t, \\ & \begin{bmatrix} L & \mathbf{0} \\ \mathbf{0} & L \end{bmatrix} \boldsymbol{b} \preceq t \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}. \end{aligned} \tag{3.25}$$

The fourth, fifth, and sixth constraints of (3.22) restrict $(\boldsymbol{b}^1, \boldsymbol{b}^2) \in \mathcal{B}$ and are simply a reformulation of the first three constraints in (3.21). □

3.4.2 Determining (r, s)-Robustness using Mixed Integer Linear Programming

In this section we address Problems 3.2 and 3.3, which involve determining the (r^*, s^*)-robustness and $(F_{\max} + 1, F_{\max} + 1)$-robustness of a given digraph. To determine these values, we will use the following notation:

Definition 3.11. *For a digraph $\mathcal{D}$ and a given $r \in \mathbb{Z}_{\geq 0}$, the maximum integer s for which $\mathcal{D}$ is (r, s)-robust is denoted as $s_{\max}(r) \in \mathbb{Z}_{\geq 0}$. If $\mathcal{D}$ is not (r, s)-robust for any $1 \leq s \leq n$, we will denote $s_{\max}(r) = 0$.*

Using this notation, the (r^*, s^*)-robustness of a nonempty, nontrivial simple digraph satisfies $r^* = r_{\max}(\mathcal{D})$ and $s^* = s_{\max}(r_{\max}(\mathcal{D}))$. This can be verified by recalling that r-robustness is equivalent to $(r, 1)$-robustness [96, Property 5.21], and that (r^*, s^*) is the maximum element of Θ according to the lexicographic ordering defined in Section 1.5. Since a method for determining $r_{\max}(\mathcal{D})$ has already been presented, this section will introduce a method for determining $s_{\max}(r)$ for any given $r \in \mathbb{Z}_{\geq 0}$. This can then be used to find $s_{\max}(r_{\max}(\mathcal{D}))$ after $r_{\max}(\mathcal{D})$ is determined.

Recall that $\vee$ indicates logical OR. An equivalent definition of $s_{\max}(r)$ can be given using the following notation:

Definition 3.12. *Let Θ be the set of all (r,s) values for which a given digraph $\mathcal{D}$ is (r,s)-robust, as per (3.2). Let $r \in \mathbb{Z}_{\geq 0}$, and let $\mathcal{X}_S^r$ be defined as in Definition 3.4. The set $\Theta_r \subset \Theta$ is defined as follows:*

$$\Theta_r = \{s \in \mathbb{Z}_{\geq 0} : \forall (S_1, S_2) \in \mathcal{T},\ \left(|\mathcal{X}_{S_1}^r| = |S_1|\right)\left(|\mathcal{X}_{S_2}^r| = |S_2|\right) \vee \left(|\mathcal{X}_{S_1}^r| + |\mathcal{X}_{S_2}^r| \geq s\right)\}, \quad (3.26)$$

In words, Θ_r is the set of all integers s for which the given digraph $\mathcal{D}$ is (r,s)-robust for a given $r \in \mathbb{Z}_{\geq 0}$. By this definition, $s_{\max}(r) = \max \Theta_r$, i.e. $s_{\max}(r)$ is simply the maximum element of Θ_r.

As per (3.26), checking directly if an integer $s \in \Theta_r$ involves testing a logical disjunction for *all* possible $(S_1, S_2) \in \mathcal{T}$. This quickly becomes impractical for large n since $|\mathcal{T}|$ grows exponentially with n. This difficulty can be circumvented, however, by defining the set

$$\begin{aligned} \bar{\Theta}_r &= \mathbb{Z}_{\geq 0} \backslash \Theta_r \qquad (3.27)\\ &= \{\bar{s} \in \mathbb{Z}_{\geq 0} : \bar{s} \notin \Theta_r\} \\ &= \{\bar{s} \in \mathbb{Z}_{\geq 0} : \exists (S_1, S_2) \in \mathcal{T} \text{ s.t. } | \left(\mathcal{X}_{S_1}^r| < |S_1|\right) \wedge \left(|\mathcal{X}_{S_2}^r| < |S_1|\right) \wedge \left(|\mathcal{X}_{S_1}^r| + |\mathcal{X}_{S_2}^r| < \bar{s}\right)\}, \end{aligned}$$

where $\wedge$ denotes logical AND. The set $\bar{\Theta}_r$ contains all integers $\bar{s}$ for which the given digraph is *not* $(r,\bar{s})$-robust for the given value of r.

Definition 3.13. *For a digraph $\mathcal{D}$ and a given $r \in \mathbb{Z}_{\geq 0}$, the minimum integer $\bar{s}$ for which $\mathcal{D}$ is* not *$(r,\bar{s})$-robust is denoted as $\bar{s}_{\min}(r) \in \mathbb{Z}_{\geq 0}$.*

As a simple example, consider a digraph $\mathcal{D}$ of 7 nodes where $s_{\max}(3) = 2$. This implies that $\Theta_3 = \{1, 2\}$, i.e. the digraph is $(3,1)$- and $(3,2)$-robust. In this case, the set $\bar{\Theta}_3 = \{3, 4, 5, \ldots\}$ since $\mathcal{D}$ is not $(3,3)$-robust, $(3,4)$-robust, etc. Here we have $\bar{s}_{\min}(3) = 3$. In general, observe that by definitions 3.11 and 3.13 we have

$$s_{\max}(r) = \bar{s}_{\min}(r) - 1. \quad (3.28)$$

It is therefore sufficient to find $\bar{s}_{\min}(r)$ in order to determine $s_{\max}(r)$. An illustration is given in Figure 3.11. The methods in this section will solve for $\bar{s}_{\min}(r)$ using a mixed integer linear program. Note that since possible values of $s_{\max}(r)$ are limited to $0 \leq s_{\max}(r) \leq n$ (Definition 3.11), possible values of $\bar{s}_{\min}(r)$ are limited to $1 \leq \bar{s}_{\min}(r) \leq n+1$. We point out that it is easier to test if an integer $\bar{s} \in \bar{\Theta}_r$ than to test if an integer $s \in \Theta_r$, in the sense that only one element

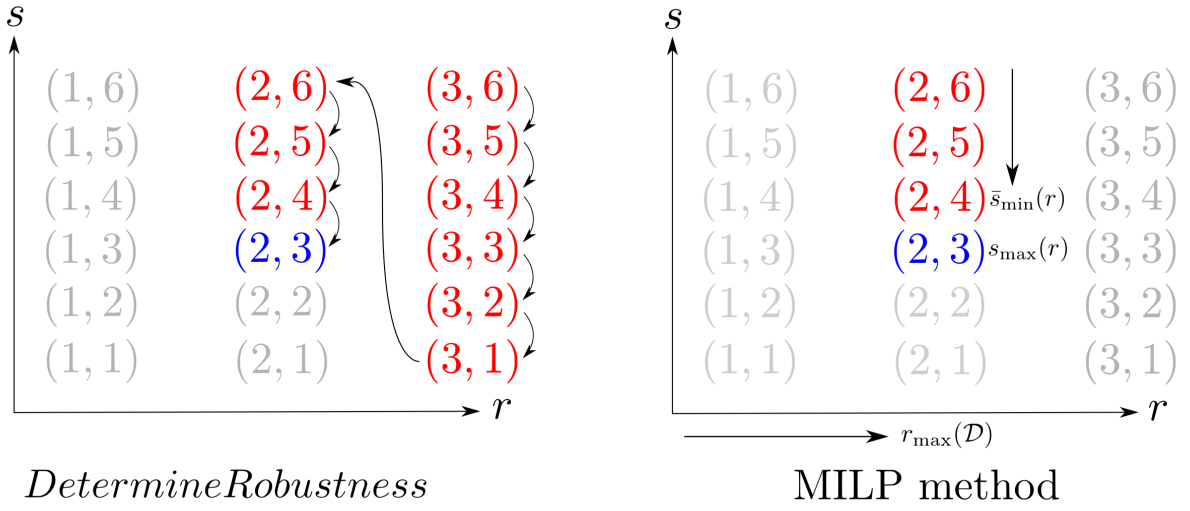


Figure 3.11: Illustration of how the (r^*, s^*)-robustness of a graph is found by the *DetermineRobustness* algorithm and the MILP method. Consider a digraph $\mathcal{D}$ of $n = 6$ nodes which satisfies $(r^*, s^*) = (2, 3)$. *DetermineRobustness* begins with the maximum possible r and s values ($r = \lceil n/2 \rceil$ and $s = n$), then iterates in a lexicographically decreasing manner. The MILP formulation first determines $r_{\max}(\mathcal{D})$, then $\bar{s}_{\min}(r_{\max}(\mathcal{D}))$, then finally infers $s_{\max}(r_{\max}(D))$ (abbreviated to $\bar{s}_{\min}(r)$ and $s_{\max}(r)$ for clarity).

$(S_1, S_2) \in \mathcal{T}$ is required to verify that $\bar{s} \in \bar{\Theta}_r$ (as per (3.27)) whereas *all* $(S_1, S_2) \in \mathcal{T}$ must be checked to verify that $s \in \Theta_r$ (as per (3.26)).

The following Lemma is needed for our main result. It shows that given any $r \in \mathbb{Z}_{\geq 0}$ and $S \subset \mathcal{V}$, the indicator vector of the set $\mathcal{X}_S^r$, denoted $\boldsymbol{\sigma}(\mathcal{X}_S^r)$, can be expressed using an MILP. Recall that $\mathcal{X}_S^r$ is the set of agents in S which have r in-neighbors outside of S, implying $\sigma_j(\mathcal{X}_S^r) = 1$ if $L_j \boldsymbol{\sigma}(S) = |\mathcal{V}_j \backslash S| \geq r$.

Lemma 3.6. *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be an arbitrary nonempty, nontrivial, simple digraph. Let L be the Laplacian matrix of $\mathcal{D}$, and let $r \in \mathbb{N}$. Consider any subset $S \subset \mathcal{V}$, $|S| > 0$ and let $\mathcal{X}_S^r$ be defined as in Definition 3.4. Then the following holds:*

$$\begin{aligned} \boldsymbol{\sigma}(\mathcal{X}_S^r) = \arg\min_{\boldsymbol{y}} \quad & \mathbf{1}^T \boldsymbol{y} \\ \text{subject to} \quad & L\boldsymbol{\sigma}(S) - (n)\boldsymbol{y} \preceq (r-1)\mathbf{1} \\ & \boldsymbol{y} \in \{0,1\}^n. \end{aligned} \tag{3.29}$$

Proof. Recall that the entries of the indicator vector $\boldsymbol{\sigma}(\mathcal{X}_S^r)$ are defined as

$$\sigma_j(\mathcal{X}_S^r) = \begin{cases} 1, & \text{if } j \in \mathcal{X}_S^r, \\ 0, & \text{otherwise.} \end{cases} \tag{3.30}$$

Let $\boldsymbol{y}^*$ be an optimal point of the RHS of (3.29). To prove that $\boldsymbol{y}^* = \boldsymbol{\sigma}(\mathcal{X}_S^r)$, we demonstrate that $\forall j \in \{1, \dots, n\}$, $\sigma_j(\mathcal{X}_S^r) = 1 \iff y_j^* = 1$. Observe that this is equivalent to demonstrating $\sigma_j(\mathcal{X}_S^r) \neq 1 \iff y_j^* \neq 1 \; \forall j$, which is equivalent to demonstrating $\sigma_j(\mathcal{X}_S^r) = 0 \iff y_j^* = 0 \; \forall j$. This can be seen by noting $\boldsymbol{\sigma}(\mathcal{X}_S^r) \in \{0,1\}^n$ which implies $\sigma_j(\mathcal{X}_S^r) \neq 1 \iff \sigma_j(\mathcal{X}_S^r) = 0$, and $\boldsymbol{y}^* \in \{0,1\}^n$ which implies $y_j^* \neq 1 \iff y_j^* = 0$. Since proving $\sigma_j(\mathcal{X}_S^r) = 1 \iff y_j^* = 1$ for all $j \in \{1, \dots, n\}$ is equivalent to proving $\sigma_j(\mathcal{X}_S^r) = 0 \iff y_j^* = 0$ for all $j \in \{1, \dots, n\}$, and both $\boldsymbol{y}^* \in \{0,1\}^n$ and $\boldsymbol{\sigma}(\mathcal{X}_S^r) \in \{0,1\}^n$, we therefore have $(\sigma_j(\mathcal{X}_S^r) = 1 \iff y_j^* = 1 \; \forall j)$ if and only if $(\boldsymbol{y}^* = \boldsymbol{\sigma}(\mathcal{X}_S^r))$.

Sufficiency: Consider any $j \in \{1, \dots, n\}$ such that $\sigma_j(\mathcal{X}_S^r) = 1$. This implies $j \in \mathcal{X}_S^r$ and therefore $|\mathcal{V}_j \backslash S| \geq r$ by Definition 3.4. By Lemma 3.3, $|\mathcal{V}_j \backslash S| = L_j \boldsymbol{\sigma}(S)$, and therefore $L_j \boldsymbol{\sigma}(S) > (r-1)$. Since $\boldsymbol{y}^*$ is an optimal point, it is therefore a feasible point. If $y_j^* = 0$, the jth row of the first constraint on the RHS of (3.29) is be violated since $L_j \boldsymbol{\sigma}(S) - (n) y_j^* = L_j \boldsymbol{\sigma}(S) \not\leq (r-1)$. Therefore we must have $y_j^* = 1$. Note that $|\mathcal{V}_j \backslash S| \leq n \; \forall j \in S$ for any $S \subset \mathcal{V}$.

Necessity: We prove by contradiction. Suppose $y_j^* = 1$ and $\sigma_j(\mathcal{X}_S^r) = 0$. This implies that $L_j \boldsymbol{\sigma}(S) = |\mathcal{V}_j \backslash S| < r$. Consider the vector $\tilde{\boldsymbol{y}}$ where $\tilde{y}_j = 0$ and $\tilde{y}_i = y_i^* \; \forall i \neq j, \; i \in \{1, \dots, n\}$. Since $L_j \boldsymbol{\sigma}(S) = |\mathcal{V}_j \backslash S| < r$, then $\tilde{\boldsymbol{y}}$ is therefore also a feasible point, and $\mathbf{1}^T \tilde{\boldsymbol{y}} < \mathbf{1}^T \boldsymbol{y}^*$. This contradicts $\boldsymbol{y}^*$ being an optimal point to (3.29); therefore we must have $\sigma_j(\mathcal{X}_S^r) = 1$. □

The next Theorem presents a mixed integer linear program which determines $\bar{s}_{\min}(r)$ for any fixed $r \in \mathbb{Z}_{\geq 0}$.

Theorem 3.5. *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be an arbitrary nonempty, nontrivial, simple digraph. Let L be the Laplacian matrix of $\mathcal{D}$. Let $r \in \mathbb{Z}_{\geq 0}$, and let $\bar{s}_{\min}(r)$ be the minimum value of s for which $\mathcal{D}$ is* not *(r,s)-robust. Then if $\bar{s}_{\min}(r) < n+1$, the following holds:*

$$
\begin{aligned}
\bar{s}_{\min}(r) = \min_{\bar{s},\boldsymbol{b}^1,\boldsymbol{b}^2,\boldsymbol{y}^1,\boldsymbol{y}^2} \quad & \bar{s} \\
\text{subject to} \quad & 1 \leq \bar{s} \leq n+1,\ \bar{s} \in \mathbb{Z} \\
& \mathbf{1}^T\boldsymbol{y}^1 \leq \mathbf{1}^T\boldsymbol{b}^1 - 1 \\
& \mathbf{1}^T\boldsymbol{y}^2 \leq \mathbf{1}^T\boldsymbol{b}^2 - 1 \\
& \mathbf{1}^T\boldsymbol{y}^1 + \mathbf{1}^T\boldsymbol{y}^2 \leq (\bar{s}-1) \\
& \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} \boldsymbol{b}^1 \\ \boldsymbol{b}^2 \end{bmatrix} - (n) \begin{bmatrix} \boldsymbol{y}^1 \\ \boldsymbol{y}^2 \end{bmatrix} \preceq (r-1) \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \\
& \boldsymbol{b}^1 + \boldsymbol{b}^2 \preceq \mathbf{1} \\
& 1 \leq \mathbf{1}^T\boldsymbol{b}^1 \leq (n-1) \\
& 1 \leq \mathbf{1}^T\boldsymbol{b}^2 \leq (n-1) \\
& \boldsymbol{b}^1, \boldsymbol{b}^2, \boldsymbol{y}^1, \boldsymbol{y}^2 \in \{0,1\}^n.
\end{aligned}
\tag{3.31}
$$

Furthermore, for any $r > 0$, $\bar{s}_{\min}(r) = n+1$ if and only if the integer program in (3.31) is infeasible.

Proof. Note that the theorem statement assumes r is a fixed integer. First, consider the case where $\bar{s}_{\min}(r) < (n+1)$. The value of $\bar{s}_{\min}$ can be found by solving the problem $\bar{s}_{\min}(r) = \min_{\bar{s} \in \bar{\Theta}_r} \bar{s}$. Making the constraints explicit yields

$$
\begin{aligned}
\bar{s}_{\min}(r, \mathcal{D}) = \min_{\bar{s},(S_1,S_2)\in\mathcal{T}} \quad & \bar{s} \\
\text{subject to} \quad & |\mathcal{X}^r_{S_1}| \leq |S_1| - 1 \\
& |\mathcal{X}^r_{S_2}| \leq |S_2| - 1 \\
& |\mathcal{X}^r_{S_1}| + |\mathcal{X}^r_{S_2}| \leq \bar{s} - 1.
\end{aligned}
\tag{3.32}
$$

To put this problem in an MILP form, we show that the terms $|\mathcal{X}^r_{S_1}|$, $|\mathcal{X}^r_{S_2}|$, $|S_1|$, and $|S_2|$ can be represented by functions of binary vectors. This can be done by first observing that for any $S \subseteq \mathcal{V}$, $|S| = \mathbf{1}^T\boldsymbol{\sigma}(S)$. Therefore the following relationships hold:

$$
\begin{aligned}
|S_1| &= \mathbf{1}^T\boldsymbol{\sigma}(S_1), & |\mathcal{X}^r_{S_1}| &= \mathbf{1}^T\boldsymbol{\sigma}(\mathcal{X}^r_{S_1}), \\
|S_2| &= \mathbf{1}^T\boldsymbol{\sigma}(S_2), & |\mathcal{X}^r_{S_2}| &= \mathbf{1}^T\boldsymbol{\sigma}(\mathcal{X}^r_{S_2}).
\end{aligned}
$$

Equation (3.32) can therefore be rewritten as

$$\begin{aligned} \bar{s}_{\min}(r, \mathcal{D}) = \min_{\bar{s}, (S_1, S_2) \in \mathcal{T}} \quad & \bar{s} \\ \text{subject to} \quad & \mathbf{1}^T \boldsymbol{\sigma}(\mathcal{X}^r_{S_1}) \leq \mathbf{1}^T \boldsymbol{\sigma}(S_1) - 1 \\ & \mathbf{1}^T \boldsymbol{\sigma}(\mathcal{X}^r_{S_2}) \leq \mathbf{1}^T \boldsymbol{\sigma}(S_2) - 1 \\ & \mathbf{1}^T \boldsymbol{\sigma}(\mathcal{X}^r_{S_1}) + \mathbf{1}^T \boldsymbol{\sigma}(\mathcal{X}^r_{S_2}) \leq \bar{s} - 1. \end{aligned} \tag{3.33}$$

By Lemma 3.5, the terms $\boldsymbol{\sigma}(S_1),\ \boldsymbol{\sigma}(S_2)$ for $(S_1, S_2) \in \mathcal{T}$ can be represented by vectors $(\boldsymbol{b}^1, \boldsymbol{b}^2) \in \mathcal{B}$. This yields

$$\begin{aligned} \bar{s}_{\min}(r, \mathcal{D}) = \min_{\bar{s}, \boldsymbol{b}^1, \boldsymbol{b}^2} \quad & \bar{s} \\ \text{subject to} \quad & \mathbf{1}^T \boldsymbol{\sigma}(\mathcal{X}^r_{S_1}) \leq \mathbf{1}^T \boldsymbol{b}^1 - 1 \\ & \mathbf{1}^T \boldsymbol{\sigma}(\mathcal{X}^r_{S_2}) \leq \mathbf{1}^T \boldsymbol{b}^2 - 1 \\ & \mathbf{1}^T \boldsymbol{\sigma}(\mathcal{X}^r_{S_1}) + \mathbf{1}^T \boldsymbol{\sigma}(\mathcal{X}^r_{S_2}) \leq \bar{s} - 1 \\ & (\boldsymbol{b}^1, \boldsymbol{b}^2) \in \mathcal{B}. \end{aligned} \tag{3.34}$$

Expanding the last constraint using the definition of $\mathcal{B}$ in (3.19) yields the sixth, seventh, and eighth constraints in (3.31) as well as the constraint that $\boldsymbol{b}^1, \boldsymbol{b}^2 \in \{0, 1\}^n$. In addition, the first constraint of the RHS of (3.31) limits the search for feasible value of $\bar{s}$ to the range of possible values for $\bar{s}_{\min}(r)$.

The vectors $\boldsymbol{y}^1, \boldsymbol{y}^2$ are constrained to satisfy $\boldsymbol{y}^1 = \boldsymbol{\sigma}(\mathcal{X}^r_{S_1})$ and $\boldsymbol{y}^2 = \boldsymbol{\sigma}(\mathcal{X}^r_{S_2})$ as follows: by Lemma 3.5, $\boldsymbol{b}^1 = \boldsymbol{\sigma}(S_1)$ and $\boldsymbol{b}^2 = \boldsymbol{\sigma}(S_2)$ for $(S_1, S_2) \in \mathcal{T}$ as per the sixth through ninth constraints. Therefore by Lemma 3.6,

$$\begin{aligned} \boldsymbol{\sigma}(\mathcal{X}^r_{S_1}) = \arg\min_{\boldsymbol{y}^1} \quad & \mathbf{1}^T \boldsymbol{y}^1 \\ \text{subject to} \quad & L\boldsymbol{b}^1 - (n)\boldsymbol{y}^1 \preceq (r-1)\mathbf{1} \\ & \boldsymbol{y}^1 \in \{0, 1\}^n, \end{aligned} \tag{3.35}$$

$$\begin{aligned} \boldsymbol{\sigma}(\mathcal{X}^r_{S_2}) = \arg\min_{\boldsymbol{y}^2} \quad & \mathbf{1}^T \boldsymbol{y}^2 \\ \text{subject to} \quad & L\boldsymbol{b}^2 - (n)\boldsymbol{y}^2 \preceq (r-1)\mathbf{1} \\ & \boldsymbol{y}^2 \in \{0, 1\}^n. \end{aligned} \tag{3.36}$$

The constraints of (3.35) and (3.36) are contained in the fifth and last constraints of (3.31). Since the fourth constraint of (3.31), $\mathbf{1}^T \boldsymbol{y}^1 + \mathbf{1}^T \boldsymbol{y}^2 \leq (\bar{s} - 1)$, simultaneously minimizes $\mathbf{1}^T \boldsymbol{y}^1$ and $\mathbf{1}^T \boldsymbol{y}^2$, the fourth, fifth, and last constraints of (3.31) ensure that $\boldsymbol{y}^1 = \boldsymbol{\sigma}(\mathcal{X}^r_{S_1})$ and $\boldsymbol{y}^2 = \boldsymbol{\sigma}(\mathcal{X}^r_{S_2})$.

Therefore $\mathbf{1}^T\boldsymbol{y}^1 = |\mathcal{X}^r_{S_1}|$ and $\mathbf{1}^T\boldsymbol{y}^2 = |\mathcal{X}^r_{S_2}|$.

Now, by the above arguments, solving the RHS of (3.31) yields $\bar{s}_{\min}(r)$ when $\bar{s}_{\min}(r) < (n+1)$. We now prove that for $r > 0$, $\bar{s}_{\min}(r) = n+1$ if and only if the RHS of (3.31) is infeasible. Note that if $r = 0$, then it trivially holds by Definition 3.4 that $s_{\max}(0) = n$ and therefore $\bar{s}_{\min}(0) = n+1$.

Sufficiency: $\bar{s}_{\min}(r) = n+1$ implies that $s_{\max}(r) = n$. Recall that $\mathcal{D}$ is $(r, s_{\max}(r))$-robust by Definition 3.11, since $s_{\max}(r)$ is the largest integer s for which $\mathcal{D}$ is (r,s)-robust. By Definition 3.4, this implies that for all $(S_1, S_2) \in \mathcal{T}$, at least one of the following three conditions holds: $|\mathcal{X}^r_{S_1}| = |S_1|$, or $|\mathcal{X}^r_{S_2}| = |S_2|$, or $|\mathcal{X}^r_{S_1}| + |\mathcal{X}^r_{S_2}| \geq s_{\max}(r) = n$. Given any $(S_1, S_2) \in \mathcal{T}$, we consider each condition separately and show that at least one constraint of (3.31) is violated if the condition holds true:

- $|\mathcal{X}^r_{S_1}| = |S_1|$ being true implies that the second constraint of (3.31) is violated. This can be shown using earlier arguments from this proof. Specifically, we have $\mathbf{1}^T\boldsymbol{y}^1 = |\mathcal{X}^r_{S_1}| = |S_1| = \mathbf{1}^T\boldsymbol{b}^1 > \mathbf{1}^T\boldsymbol{b}^1 - 1$. Therefore no feasible point can be constructed from the given set pair (S_1, S_2) if $|\mathcal{X}^r_{S_1}| = |S_1|$.

- $|\mathcal{X}^r_{S_2}| = |S_2|$ being true implies that the third constraint of (3.31) is violated. Specifically, we have $\mathbf{1}^T\boldsymbol{b}^2 = |S_2|$ and $\mathbf{1}^T\boldsymbol{y}^2 = |\mathcal{X}^r_{S_2}|$. This implies that $\mathbf{1}^T\boldsymbol{y}^2 = |\mathcal{X}^r_{S_2}| = |S_2| = \mathbf{1}^T\boldsymbol{b}^2 > \mathbf{1}^T\boldsymbol{b}^2 - 1$. Therefore no feasible point can be constructed from the given set pair (S_1, S_2) if $|\mathcal{X}^r_{S_2}| = |S_2|$.

- $|\mathcal{X}^r_{S_1}| + |\mathcal{X}^r_{S_2}| \geq n$ being true implies that both $|\mathcal{X}^r_{S_1}| = |S_1|$ and $\mathcal{X}^r_{S_2} = |S_2|$. This follows by observing that $\mathcal{X}^r_{S_1} \subseteq S_1$ and $\mathcal{X}^r_{S_2} \subseteq S_2$, $S_1 \cap S_2 = \{\emptyset\}$ by definition of $\mathcal{T}$ in (3.1), and therefore $\mathcal{X}^r_{S_1} \cap \mathcal{X}^r_{S_2} = \{\emptyset\}$. Since $S_1, S_2 \subset \mathcal{V}$ and $|\mathcal{V}| = n$, we have $n \leq |\mathcal{X}^r_{S_1}| + |\mathcal{X}^r_{S_2}| \leq |S_1| + |S_2| \leq |\mathcal{V}| = n$. We must therefore have $|\mathcal{X}^r_{S_1}| = |S_1|$ and $|\mathcal{X}^r_{S_2}| = |S_2|$, which from prior arguments both imply that a constraint of (3.31) is violated. Therefore no feasible point can be constructed from the given set pair (S_1, S_2) if $|\mathcal{X}^r_{S_1}| + |\mathcal{X}^r_{S_2}| \geq n$

Since for all $(S_1, S_2) \in \mathcal{T}$ at least one of these three conditions holds, for all $(S_1, S_2) \in \mathcal{T}$ at least one constraint of (3.31) is violated when $s_{\max}(r) = n$, which is equivalent to $\bar{s}_{\min}(r) = n+1$. Therefore $\bar{s}_{\min}(r) = n+1$ implies that (3.31) is infeasible.

Necessity: We prove the contrapositive, i.e. we prove that $\bar{s}_{\min} \neq n+1$ implies that there exists a feasible point to the RHS of (3.31). First, no digraph on n nodes is $(r, n+1)$-robust [2, Definition 13], and the contrapositive of Property 3.1 implies that if a graph is *not* $(\bar{r}, \bar{s})$-robust, then it is also not $(\bar{r}', \bar{s}')$-robust for all $\bar{r}' \geq \bar{r}$, and for all $\bar{s}' \geq \bar{s}$. Therefore $\bar{s}_{\min} \neq n+1$ implies $\bar{s}_{\min} \leq n$. Next, $\bar{s}_{\min} \leq n$ implies $n \in \bar{\Theta}_r$, which implies that there exists $(S_1, S_2) \in \mathcal{T}$ such that $|\mathcal{X}^r_{S_1}| \leq |S_1| - 1$ and $|\mathcal{X}^r_{S_2}| \leq |S_2| - 1$ and $|\mathcal{X}^r_{S_1}| + |\mathcal{X}^r_{S_2}| \leq n - 1$, as per (3.27). Letting $\bar{s} = n$, $\boldsymbol{b}^1 = \boldsymbol{\sigma}(S_1)$, $\boldsymbol{b}^2 = \boldsymbol{\sigma}(S_2)$, $\boldsymbol{y}^1 = \boldsymbol{\sigma}(\mathcal{X}^r_{S_1})$, and $\boldsymbol{y}^2 = \boldsymbol{\sigma}(\mathcal{X}^r_{S_2})$ yields a feasible point to (3.31). $\square$

The MILPs in Theorems 3.4 and 3.5 can be used to determine the (r^*, s^*)-robustness of any digraph satisfying Assumption 3.1, thereby solving Problem 3.2. Recall from the beginning of Section 3.4.2 that $r^* = r_{\max}(\mathcal{D})$ and $s^* = s_{\max}(r_{\max}(\mathcal{D}))$. Theorem 3.4 can first be used to determine the value of $r_{\max}(\mathcal{D}) = r^*$. Using $r_{\max}(\mathcal{D})$, Theorem 3.5 can then be used to find the value of $s_{\max}(r_{\max}(\mathcal{D})) = s^*$.

More generally however, the MILP formulation in Theorem 3.5 allows for $s_{\max}(r)$ to be determined for *any* $r \in \mathbb{Z}_{\geq 0}$. Since $(r, 1)$-robustness is equivalent to r-robustness, the MILP in Theorem 3.5 can also be used to determine whether a digraph $\mathcal{D}$ is r robust for a given $r \in \mathbb{Z}_{\geq 0}$. If $\bar{s}_{\min}(r) \geq 2$, then $s_{\max}(r) \geq 1$ which implies that $\mathcal{D}$ is $(r, 1)$-robust. On the other hand, $\bar{s}_{\min}(r) = 1$ implies that $1 \in \bar{\Theta}_r$ and therefore $\mathcal{D}$ is *not* $(r, 1)$-robust (and *not* r-robust).

Finally, to solve Problem 3.3 Theorem 3.5 can be used to determine the $(F_{\max} + 1, F_{\max} + 1)$-robustness of a nonempty, nontrivial, simple digraph. Recall that $F_{\max} = \max(\{F \in \mathbb{Z}_{\geq 0} : (F + 1, F + 1) \in \Theta\})$. The value of $F_{\max}$ is determined by Algorithm 3.1, presented below. In

Algorithm 3.1 DETERMINEFMAX

$r' \leftarrow r_{\max}(\mathcal{D})$ from MILP in Theorem 3.4
while $r' > 0$ **do**
    $s' \leftarrow s_{\max}(r')$ from MILP in Theorem 3.5
    if $s' \geq r'$ **then** ▷ *Prop. 3.1 implies graph is* (r', s)*-rob.* $\forall s \leq s'$*, therefore* $\mathcal{D}$ *is* (r', r')*-robust*
        $F_{\max} \leftarrow (r' - 1)$
        return $F_{\max}$
    else
        $r' \leftarrow (r' - 1)$
    end if
end while
$F_{\max} \leftarrow 0$
return $F_{\max}$

essence, Algorithm 3.1 finds the largest values of r' and s' such that $r' = s'$ and $(r', s') \in \Theta$. It begins by setting $r' \leftarrow r_{\max}(\mathcal{D})$, and finding $s_{\max}(r')$ using Theorem 3.5. If $s_{\max}(r') \geq r'$, then by Proposition 3.1 the digraph $\mathcal{D}$ is (r', s)-robust for $s = r'$ and therefore (r', r')-robust. This implies $r' = F_{\max} + 1$. However, if $s_{\max}(r') < r'$ then r' is decremented, $s_{\max}(r')$ recalculated, and the process is repeated until the algorithm terminates with the highest integer r' such that $\mathcal{D}$ is (r', r')-robust, yielding $F_{\max} = (r' - 1)$.

3.4.3 Approximate Bounds on $\mathbf{r}_{\max}(\mathcal{D})$

When solving a MILP with zero-one integer variables using a branch-and-bound technique, the maximum number of subproblems to be solved is equal to 2^n, where n is the dimension of the zero-

one integer vector variable. In Section 3.4.1, the MILP in Theorem 1 which solves for $r_{\max}(\mathcal{D})$ has a binary vector variable with dimension $2n$. In this section, we present two MILPs whose optimal values provide upper and lower bounds on the value of $r_{\max}(\mathcal{D})$. Each MILP has a binary vector variable with dimension of only n, which implies a lower complexity in terms of maximum number of subproblems as compared to the MILP in Theorem 3.4.

3.4.3.1 A Lower Bound on Maximum r-Robustness

In [98], a technique is presented for lower bounding $r_{\max}(\mathcal{D})$ of undirected graphs by searching for the minimum reachability of subsets $S \subset \mathcal{V}$ such that $|S| \leq \lfloor n/2 \rfloor$. We extend this result to digraphs in the next Lemma.

Lemma 3.7. *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be an arbitrary nonempty, nontrivial, simple digraph. Let $\Psi = \{S \subset \mathcal{V} : 1 \leq |S| \leq \lfloor n/2 \rfloor\}$. Then the following holds:*

$$r_{\max}(\mathcal{D}) \geq \min_{S \in \Psi} \mathcal{R}(S). \tag{3.37}$$

Proof. By Lemma 3.2 and Remark 3.1, proving (3.37) is equivalent to proving

$$\min_{S \in \Psi} \mathcal{R}(S) \leq \min_{(S_1, S_2) \in \mathcal{T}} \max\left(\mathcal{R}(S_1), \mathcal{R}(S_2)\right). \tag{3.38}$$

Denote $S^* = \arg\min_{S \in \Psi} \mathcal{R}(S)$ and $(S_1^*, S_2^*) = \arg\min_{(S_1, S_2) \in \mathcal{T}} \max(\mathcal{R}(S_1), \mathcal{R}(S_2))$. We prove by contradiction. Suppose $\mathcal{R}(S^*) > \max(\mathcal{R}(S_1^*), \mathcal{R}(S_2^*))$. Since S_1^* and S_2^* are nonempty, $|S_1^*| \geq 1$ and $|S_2^*| \geq 1$. Since they are disjoint, we must have either $|S_1^*| \leq \lfloor n/2 \rfloor$, or $|S_2^*| \leq \lfloor n/2 \rfloor$, or both $|S_1^*|$ and $|S_2^*|$ less than or equal to $\lfloor n/2 \rfloor$. Therefore either $S_1^* \in \Psi$ or $S_2^* \in \Psi$. This implies that either $\mathcal{R}(S_1^*) \geq \mathcal{R}(S^*)$ (if $S_1^* \in \Psi$) or $\mathcal{R}(S_2^*) \geq \mathcal{R}(S^*)$ (if $S_2^* \in \Psi$), since S^* is an optimal point. But this contradicts the assumption that $\mathcal{R}(S^*) > \max(\mathcal{R}(S_1^*), \mathcal{R}(S_2^*))$. Therefore we must have

$$\min_{S \in \Psi} \mathcal{R}(S) \leq \min_{(S_1, S_2) \in \mathcal{T}} \max\left(\mathcal{R}(S_1), \mathcal{R}(S_2)\right) = r_{\max}(\mathcal{D}), \tag{3.39}$$

which concludes the proof. □

Using this result, a lower bound on $r_{\max}(\mathcal{D})$ can be obtained by the following optimization problem:

Theorem 3.6. *Let $\mathcal{D}$ be an arbitrary nonempty, nontrivial, simple digraph and let L be the Laplacian matrix of $\mathcal{D}$. A lower bound on the maximum integer for which $\mathcal{D}$ is r-robust, denoted $r_{\max}(\mathcal{D})$,*

is found as follows:

$$\begin{aligned} r_{\max}(\mathcal{D}) \geq \min_{\boldsymbol{b}} \quad & \max_i \left(L_i \boldsymbol{b}\right) \\ \text{subject to} \quad & 1 \leq \mathbf{1}^T \boldsymbol{b} \leq \lfloor n/2 \rfloor \\ & \boldsymbol{b} \in \{0,1\}^n. \end{aligned} \tag{3.40}$$

Furthermore, (3.40) *is equivalent to the following mixed integer linear program:*

$$\begin{aligned} r_{\max}(\mathcal{D}) \geq \min_{t,\boldsymbol{b}} \quad & t \\ \text{subject to} \quad & t \geq 0,\ \boldsymbol{b} \in \mathbb{Z}^n \\ & L\boldsymbol{b} \preceq t\mathbf{1} \\ & \mathbf{0} \preceq \boldsymbol{b} \preceq \mathbf{1} \\ & 1 \leq \mathbf{1}^T \boldsymbol{b} \leq \lfloor n/2 \rfloor \end{aligned} \tag{3.41}$$

Proof. To prove the result we show that the RHS of (3.40) is equivalent to the RHS of (3.37). By Lemma 3.3, $\mathcal{R}(S) = \max_i L_i \boldsymbol{\sigma}(S)$. Therefore (3.37) is equivalent to

$$r_{\max}(\mathcal{D}) \geq \min_{S \in \Psi} \max_i L_i \boldsymbol{\sigma}(S). \tag{3.42}$$

Next, we demonstrate that the set

$$\mathcal{B}_\Psi = \{\boldsymbol{b} \in \{0,1\}^n : 1 \leq \mathbf{1}^T \boldsymbol{b} \leq \lfloor n/2 \rfloor\} \tag{3.43}$$

satisfies $\mathcal{B}_\Psi = \boldsymbol{\sigma}(\Psi)$, where $\boldsymbol{\sigma}(\Psi)$ is the image of Ψ under $\boldsymbol{\sigma} : \mathcal{P}(\mathcal{V}) \to \{0,1\}^n$. Since $1 \leq |S| \leq \lfloor n/2 \rfloor$ for all $S \in \Psi$, then by (3.9) we have $1 \leq \mathbf{1}^T \boldsymbol{\sigma}(S) \leq \lfloor n/2 \rfloor$ for all $S \in \Psi$. Also, $\boldsymbol{\sigma}(S) \in \{0,1\}^n$, and therefore $\boldsymbol{\sigma}(S) \in \mathcal{B}_\Psi\ \forall S \in \Psi$, implying that $\boldsymbol{\sigma}(\Psi) \subseteq \mathcal{B}$. Next, for any $\boldsymbol{b} \in \mathcal{B}_\Psi$, choose the set $S = \boldsymbol{\sigma}^{-1}(\boldsymbol{b})$ (recall from (3.10) that $\boldsymbol{\sigma}^{-1} : \{0,1\}^n \to \mathcal{P}(\mathcal{V})$). Then clearly $\boldsymbol{\sigma}(S) = \boldsymbol{\sigma}(\boldsymbol{\sigma}^{-1}(\boldsymbol{b})) = \boldsymbol{b}$, and therefore $\mathcal{B}_\Psi \subseteq \boldsymbol{\sigma}(\Psi)$. Therefore $\mathcal{B}_\Psi = \boldsymbol{\sigma}(\Psi)$.

The function $\boldsymbol{\sigma} : \Psi \to \mathcal{B}_\Psi$ is therefore surjective. Since $\boldsymbol{\sigma} : \mathcal{P}(\mathcal{V}) \to \{0,1\}^n$ is injective, $\Psi \subset \mathcal{P}(\mathcal{V})$, and $\mathcal{B}_\Psi \subset \{0,1\}^n$, then $\boldsymbol{\sigma} : \Psi \to \mathcal{B}_\Psi$ is also injective and therefore a bijection. This implies that (3.42) is equivalent to

$$r_{\max}(\mathcal{D}) \geq \min_{\boldsymbol{b} \in \mathcal{B}_\Psi} \max_i L_i \boldsymbol{b}. \tag{3.44}$$

Making the constraints of (3.44) explicit yields (3.40). More specifically, since $\mathcal{B}_\Psi = \{\boldsymbol{b} \in$

$\{0,1\}^n : 1 \leq \mathbf{1}^T\boldsymbol{b} \leq \lfloor n/2 \rfloor\}$ by (3.43), equation (3.44) can be rewritten with explicit constraints on $\boldsymbol{b}$ as follows:

$$\begin{aligned} r_{\max}(\mathcal{D}) \geq \min_{\boldsymbol{b}} \quad & \max_i (L_i\boldsymbol{b}) \\ \text{subject to} \quad & 1 \leq \mathbf{1}^T\boldsymbol{b} \leq \lfloor n/2 \rfloor \\ & \boldsymbol{b} \in \{0,1\}^n. \end{aligned} \tag{3.45}$$

Equation (3.45) is the same as (3.40).

We next prove that (3.41) is equivalent to (3.40). As per the proof of Theorem 3.4, the objective and first two constraints of (3.41) are a reformulation of the objective of the RHS of (3.40). The first and third constraint ensure b to be in $\{0,1\}^n$, and the fourth constraint ensures $\boldsymbol{b} \in \mathcal{B}_\Psi$. □

3.4.3.2 An Upper Bound on Maximum r-Robustness

This section will present an MILP whose solution provides an *upper* bound on the value of $r_{\max}(\mathcal{D})$, and whose binary vector variable has a dimension of n. This will be accomplished by searching a subset $\mathcal{T}' \subset \mathcal{T}$ which is defined as

$$\mathcal{T}' = \{(S_1, S_2) \in \mathcal{T} : S_1 \cup S_2 = \mathcal{V}\}. \tag{3.46}$$

In other words, $\mathcal{T}'$ is the set of all possible partitionings of $\mathcal{V}$ into S_1 and S_2. Considering only elements of $\mathcal{T}'$ yields certain properties that allow us to calculate an upper bound on $r_{\max}(\mathcal{D})$ using an MILP with only an n-dimensional binary vector.

Observe that $|\mathcal{T}'| = 2^n - 2$, since neither $\mathcal{T}$ nor $\mathcal{T}'$ include the cases where $S_1 = \{\emptyset\}$ or $S_2 = \{\emptyset\}$. Similar to the methods discussed earlier, the partitioning of $\mathcal{V}$ into S_1 and S_2 can be represented by the indicator vectors $\boldsymbol{\sigma}(S_1)$ and $\boldsymbol{\sigma}(S_2)$, respectively. Note that since $S_1 \cup S_2 = \mathcal{V}$ for all $(S_1, S_2) \in \mathcal{T}'$, it can be shown that $\boldsymbol{\sigma}(S_1) + \boldsymbol{\sigma}(S_2) = \mathbf{1}\ \forall (S_1, S_2) \in \mathcal{T}'$. These properties allow the following Lemma to be proven:

Lemma 3.8. *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be an arbitrary nonempty, nontrivial, simple digraph. Let L be the Laplacian matrix of $\mathcal{D}$ and let L_j be the jth row of L. Let $\mathcal{T}'$ be defined as in (3.46). Then for all*

$(S_1, S_2) \in \mathcal{T}'$, *the following holds:*

$$L_j\boldsymbol{\sigma}(S_1) = \begin{cases} |\mathcal{V}_j \backslash S_1|, & \text{if } j \in S_1, \\ -|\mathcal{V}_j \backslash S_2|, & \text{if } j \in S_2. \end{cases}$$
$$L_j\boldsymbol{\sigma}(S_2) = \begin{cases} |\mathcal{V}_j \backslash S_2|, & \text{if } j \in S_2, \\ -|\mathcal{V}_j \backslash S_1|, & \text{if } j \in S_1. \end{cases} \tag{3.47}$$

Proof. Lemma 3.3 implies that $L_j\boldsymbol{\sigma}(S_1) = |\mathcal{V}_j \backslash S_1|$ if $j \in S_1$, and $L_j\boldsymbol{\sigma}(S_2) = |\mathcal{V}_j \backslash S_2|$ if $j \in S_2$. Since $(S_1, S_2) \in \mathcal{T}' \implies \boldsymbol{\sigma}(S_1) + \boldsymbol{\sigma}(S_2) = \mathbf{1}$, we have

$$L_j\boldsymbol{\sigma}(S_1) = L_j(\mathbf{1} - \sigma(S_2)) = -L_j\boldsymbol{\sigma}(S_2). \tag{3.48}$$

This relation holds because, by the definition of L, $\mathbf{1}$ is always in the null space of L. Therefore for $j \in S_2$ we have $L_j\boldsymbol{\sigma}(S_1) = -L_j\boldsymbol{\sigma}(S_2) = -|\mathcal{V}_j \backslash S_2|$, and for $j \in S_1$ we have $L_j\boldsymbol{\sigma}(S_2) = -L_j\boldsymbol{\sigma}(S_1) = -|\mathcal{V}_j \backslash S_1|$. □

An interesting result of Lemma 3.8 is that for any subsets $(S_1, S_2) \in \mathcal{T}'$, the maximum reachability of the two subsets can be recovered using the infinity norm.

Lemma 3.9. *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be an arbitrary nonempty, nontrivial, simple digraph and let L be the Laplacian matrix of $\mathcal{D}$. For all $(S_1, S_2) \in \mathcal{T}'$, the following holds:*

$$\|L\boldsymbol{\sigma}(S_1)\|_\infty = \|L\boldsymbol{\sigma}(S_2)\|_\infty = \max(\mathcal{R}(S_1), \mathcal{R}(S_2)). \tag{3.49}$$

Proof. Denote the nodes in S_1 as $\{i_1, \ldots, i_p\}$ and the nodes in S_2 as $\{j_1, \ldots, j_{(n-p)}\}$ with $p \in \mathbb{Z},\ 1 \leq p \leq (n-1)$. Note that since $S_1 \cup S_2 = \mathcal{V}$, we have $\{i_1, \ldots, i_p\} \cup \{j_1, \ldots, j_{n-p}\} = \{1, \ldots, n\}$.

The right hand side of equation (3.49) can be expressed as

$$\max(\mathcal{R}(S_1), \mathcal{R}(S_2)) = \max(|N_{i_1} \backslash S_1|, \ldots, |N_{i_p} \backslash S_1|, |N_{j_1} \backslash S_2|, \ldots, |N_{j_{(n-p)}} \backslash S_2|). \tag{3.50}$$

Similarly, using Lemma 3.8 yields

$$\begin{aligned} \|L\boldsymbol{\sigma}(S_1)\|_\infty &= \max\left(|L_1\boldsymbol{\sigma}(S_1)|, \ldots, |L_n\boldsymbol{\sigma}(S_1)|\right) \\ &= \max\left(|N_{i_1} \backslash S_1|, \ldots, |N_{i_p} \backslash S_1|, |N_{j_1} \backslash S_2|, \ldots, |N_{j_{(n-p)}} \backslash S_2|\right) \\ &= \max(\mathcal{R}(S_1), \mathcal{R}(S_2)). \end{aligned} \tag{3.51}$$

Finally, observe that

$$\|L\boldsymbol{\sigma}(S_2)\|_\infty = \|L(\mathbf{1} - \boldsymbol{\sigma}(S_1))\|_\infty = \|L\boldsymbol{\sigma}(S_1)\|_\infty , \tag{3.52}$$

which completes the proof. □

A mixed integer linear program yielding an upper bound on the value of $r_{\max}(\mathcal{D})$ is therefore given by the following Theorem:

Theorem 3.7. *Let $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ be an arbitrary nonempty, nontrivial, simple digraph. Let L be the Laplacian matrix of $\mathcal{D}$. The maximum integer for which $\mathcal{D}$ is r-robust, denoted $r_{\max}(\mathcal{D})$, is upper bounded as follows:*

$$\begin{aligned} r_{\max}(\mathcal{D}) \leq \min_{\boldsymbol{b}} \quad & \|L\boldsymbol{b}\|_\infty \\ \text{subject to} \quad & 1 \leq \mathbf{1}^T\boldsymbol{b} \leq (n-1) \\ & \boldsymbol{b} \in \{0,1\}^n. \end{aligned} \tag{3.53}$$

Furthermore, (3.53) is equivalent to the following mixed integer linear program:

$$\begin{aligned} r_{\max}(\mathcal{D}) \leq \min_{t,\boldsymbol{b}} \quad & t \\ \text{subject to} \quad & 0 \leq t,\ \boldsymbol{b} \in \mathbb{Z}^n \\ & -t\mathbf{1} \preceq L\boldsymbol{b} \preceq \mathbf{1}t \\ & \mathbf{0} \preceq \boldsymbol{b} \preceq \mathbf{1} \\ & 1 \leq \mathbf{1}^T\boldsymbol{b} \leq (n-1) \end{aligned} \tag{3.54}$$

Proof. Consider the optimization problem

$$\min_{(S_1,S_2)\in\mathcal{T}'} \quad \max\left(\mathcal{R}(S_1), \mathcal{R}(S_2)\right). \tag{3.55}$$

Since $\mathcal{T}' \subset \mathcal{T}$, the optimal value of (3.55) is a valid upper bound on the value of $r_{\max}(\mathcal{D})$ as per Remark 3.1. From (3.55) and Lemma 3.9 we obtain

$$\begin{aligned} r_{\max}(\mathcal{D}) &\leq \min_{(S_1,S_2)\in\mathcal{T}'} \|L\boldsymbol{\sigma}(S_1)\|_\infty \\ &= \min_{(S_1,S_2)\in\mathcal{T}'} \|L\boldsymbol{\sigma}(S_2)\|_\infty . \end{aligned} \tag{3.56}$$

Since S_1, S_2 are nonempty and $S_1 \cup S_2 = \mathcal{V}$ for all $(S_1, S_2) \in \mathcal{T}'$, the set of all possible S_1 subsets within elements of $\mathcal{T}'$ is $(\mathcal{P}(\mathcal{V})\backslash\{\emptyset, \mathcal{V}\})$. Similarly, the set of all possible S_2 subsets within elements of $\mathcal{T}'$ is also $(\mathcal{P}(\mathcal{V})\backslash\{\emptyset, \mathcal{V}\})$. For brevity, denote $\mathcal{P}_{\emptyset,\mathcal{V}} = \mathcal{P}(\mathcal{V})\backslash\{\emptyset, \mathcal{V}\}$.

Next, we demonstrate that the set $\mathcal{B}' = \{\boldsymbol{b} \in \{0,1\}^n : 1 \leq \mathbf{1}^T\boldsymbol{b} \leq (n-1)\}$ satisfies $\boldsymbol{\sigma}(\mathcal{P}_{\emptyset,\mathcal{V}}) = \mathcal{B}'$. Since $1 \leq |S| \leq (n-1)$ for all $S \in \mathcal{P}_{\emptyset,\mathcal{V}}$, then by (3.9) we have $1 \leq \mathbf{1}^T\boldsymbol{\sigma}(S) \leq (n-1)$ for all $S \in \mathcal{P}_{\emptyset,\mathcal{V}}$. Also, $\boldsymbol{\sigma}(S) \in \{0,1\}^n$, and therefore $\boldsymbol{\sigma}(S) \in \mathcal{B}'\ \forall S \in \mathcal{P}_{\emptyset,\mathcal{V}}$, implying that $\boldsymbol{\sigma}(\mathcal{P}_{\emptyset,\mathcal{V}}) \subseteq \mathcal{B}'$. Next, for any $\boldsymbol{b} \in \mathcal{B}'$, choose the set $S = \boldsymbol{\sigma}^{-1}(\boldsymbol{b})$. Then clearly $\boldsymbol{\sigma}(S) = \boldsymbol{\sigma}(\boldsymbol{\sigma}^{-1}(\boldsymbol{b})) = \boldsymbol{b}$, and therefore $\mathcal{B}' \subseteq \boldsymbol{\sigma}(\mathcal{P}_{\emptyset,\mathcal{V}})$. Therefore $\mathcal{B}' = \boldsymbol{\sigma}(\mathcal{P}_{\emptyset,\mathcal{V}})$.

The function $\boldsymbol{\sigma} : \mathcal{P}_{\emptyset,\mathcal{V}} \to \mathcal{B}'$ is therefore surjective. Since $\boldsymbol{\sigma} : \mathcal{P}(\mathcal{V}) \to \{0,1\}^n$ is injective, $\mathcal{P}_{\emptyset,\mathcal{V}} \subset \mathcal{P}(\mathcal{V})$, and $\mathcal{B}' \subset \{0,1\}^n$, then $\boldsymbol{\sigma} : \mathcal{P}_{\emptyset,\mathcal{V}} \to \mathcal{B}'$ is also injective. Therefore $\boldsymbol{\sigma} : \mathcal{P}_{\emptyset,\mathcal{V}} \to \mathcal{B}'$ is a bijection, implying that (3.56) is equivalent to

$$r_{\max}(\mathcal{D}) \leq \min_{\boldsymbol{b} \in \mathcal{B}'} \|L\boldsymbol{b}\|_\infty . \tag{3.57}$$

Making the constraints of (3.57) explicit yields (3.53). We next prove that (3.54) is equivalent to (3.53). It can be shown [189, Chapter 4] that $\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_\infty$ is equivalent to

$$\begin{aligned} \min_{t,\boldsymbol{x}} \quad & t \\ \text{subject to} \quad & -t\mathbf{1} \preceq \boldsymbol{x} \preceq t\mathbf{1}. \end{aligned}$$

Likewise, the objective and first two constraints of the RHS of (3.54) are a reformulation of the objective of the RHS of (3.53). The first and third constraint restrict $\boldsymbol{b} \in \{0,1\}^n$, and the fourth constraint restricts $\boldsymbol{b}$ to be an element of $\mathcal{B}'$. These arguments imply that the RHS of (3.54) is equivalent to the RHS of (3.53). □

3.4.4 Discussion

MILP problems are NP-hard problems to solve in general. As such, the formulations presented in this chapter do not reduce the theoretical complexity of the robustness determination problem. However, it has been pointed out that algorithmic advances and improvement in computer hardware have led to a speedup factor of 800 billion for mixed integer optimization problems during the last 25 years [226]. The results of this section allow for the robustness determination problem to benefit from ongoing and future improvements in the active areas of optimization and integer programming.

In addition, one crucial advantage of the MILP formulations is the ability to iteratively tighten a global lower bound on the optimal value over time by using a branch-and-bound algorithm. The

reader is referred to [227] for a concise overview of how such a lower bound can be calculated. In context of robustness determination, lower bounds on $r_{\max}(\mathcal{D})$ and $s_{\max}(r)$ are generally more useful than upper bounds since they can be used to calculate lower bounds on the maximum adversary model that the network can tolerate. The ability to use branch-and-bound algorithms for solving the robustness determination problem offers the flexibility of terminating the search for $r_{\max}(\mathcal{D})$ and/or $s_{\max}(r)$ when sufficiently high lower bounds have been determined. In this manner, approximations of these values can be found when it is too computationally expensive to solve for them exactly. In addition, unlike prior exhaustive search algorithms the MILP formulation also allows for portions of the search space to be pruned via branch-and-bound and cutting plane methods without needing to search them explicitly. This is currently the only method that can prune the search space in the r-and (r, s)-robustness determination problem for general undirected and directed graphs. The investigation of additional methods to efficiently find and approximate solutions to the MILP formulations in this section is left for future work.

3.4.5 Comparison of MILP Robustness Determination with Prior Methods

This section presents simulations which demonstrate the performance of the MILP formulations as compared to a robustness determination algorithm from prior literature called $DetermineRobustness$ [96]. Computations for these simulations are performed in MATLAB 2018b on a desktop computer with 8 Intel core i7-7820X CPUs (3.60 GHz) capable of handling 16 total threads. All MILP problems are solved using MATLAB's *intlinprog* function.

Four types of random graphs are considered in the simulations: Erdős-Rényi random graphs, random digraphs, k-out random graphs [228], and k-in random graphs. The Erdős-Rényi random graphs in these simulations consist of n agents, with each possible undirected edge present independently with probability $p \in [0, 1]$ and absent with probability $1 - p$. Three values of p are considered: 0.3, 0.5, and 0.8. The random digraphs considered consist of n nodes with each possible *directed* edge present independently with probability p and absent with probability $1 - p$. The values of p considered are again 0.3, 0.5, and 0.8. The k-out random graphs consist of n nodes. For each vertex $i \in \mathcal{V}$, $k \in \mathbb{N}$ other nodes are chosen with all having equal probability and all choices being independent. For each node j of the k chosen nodes, a directed edge (i, j) is formed. k-in random graphs are formed in the same manner as k-out random graphs with the exception that the direction of the directed edges are reversed; i.e. edges (j, i) are formed. The values of k considered are $\{3, 4, 5\}$.

Two sets of simulations are considered. The first set compares two algorithms which determine the pair (r^*, s^*) for a digraph: the $DetermineRobustness$ algorithm from [96] and Algorithm 3.3, (r, s)-*Rob. MILP*, which is an MILP formulation using results from Theorems 3.4 and 3.5. Details

about the implementation of these algorithms can be found in the Appendix, section 3.6. The algorithms are tested on the four types of graphs described above with values of n ranging from 7 to 15. In addition, the MILP formulation is tested on digraphs with values of n ranging from 17 to 25. The $DetermineRobustness$ algorithm is not tested on values of n above 15 since the convergence rate trend is clear from the existing data, and the projected convergence times are prohibitive for large n. 100 graphs per graph type and combination of n and p (or n and k depending on the respective graph type), are randomly generated, and the algorithms are run on each graph. Overall, 10,800 total graphs are analyzed with $DetermineRobustness$ and 16,800 total graphs are analyzed with Algorithm 3.3. The time for each algorithm to determine the pair (r^*, s^*) is averaged for each combination of n and p (for Erdős-Rényi random graphs and random digraphs), and for each combination of n and k (for k-out and k-in random graphs). The interpolated circles represent the average convergence time in seconds over 100 trials for each value of n, while the vertical lines represent the spread between maximum convergence time and minimum convergence time over trials for the respective value of n. Note the logarithmic scale of the y-axis.

To facilitate the large number of graphs being tested, a time limit of 10^3 seconds (roughly 17 minutes) is imposed on Algorithm 3.3 (the MILP formulation). However, out of the 16,800 graphs tested by Algorithm 3.3, there are only 62 instances where the algorithm did not converge to optimality before this time limit. Instances where the time limit was violated are given the maximum time of 10^3 seconds and included in the data. The graphs where optimality was not reached by the time limit all have between 21 and 25 nodes, are either Erdős-Rényi random graphs or random digraphs, and have edge formation probabilities of $p = 0.8$.

Several patterns in the data warrant discussion. It is clear that in some cases, the minimum time of $DetermineRobustness$ is less than that of (r, s)-*Rob. MILP*. $DetermineRobustness$ terminates if two subsets S_1 and S_2 which are both 0-reachable are encountered, since this implies that the graph is at most $(0, n)$-robust. This can result in fast termination if such subsets are encountered early in the search. Second, there are instances where the maximum time for the (r, s)-*Rob. MILP* is much higher than that of the $DetermineRobustness$ algorithm (e.g. for k-out random digraphs with $k = 4$). It is not immediately clear why this is the case; future work will investigate graph characteristics which affect the convergence time of the MILP formulations. Finally, for small values of n (e.g. $n \in \{7, 8\}$) the average time for $DetermineRobustness$ is lower than the average time for the (r, s)-*Rob. MILP*. This likely reflects that it may be quicker to simply test all unique nonempty, disjoint subsets in these cases (966 for $n = 7$, 3025 for $n = 8$) than to incur computational overhead associated with solving the MILP formulations. We point out that, with a few exceptions, the difference in this case is small: the average convergence time for both algorithms is generally under 10^{-1} seconds for $n \in \{7, 8\}$.

The second simulation set compares the performance of four algorithms which determine

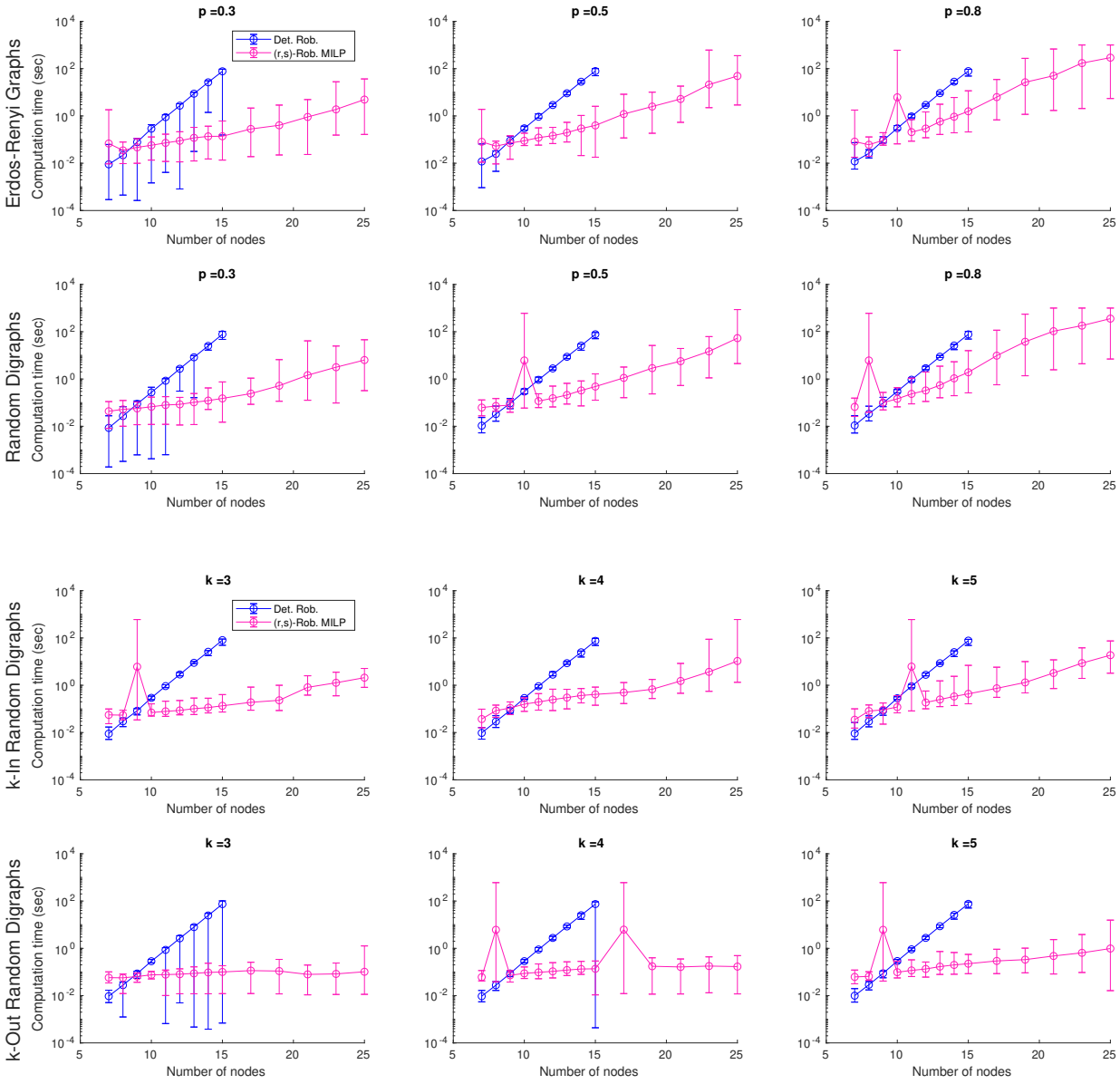


Figure 3.12: Comparison of $DetermineRobustness$ to (r, s)-*Rob. MILP* (Algorithm 3.3). The interpolating lines and circles represents the average computation time in seconds over 100 digraphs for each value of n, the upper and lower lines represent the maximum and minimum computation times, respectively, over the 100 trials for each n. Note that (r, s)-*Rob. MILP* actually solves *two* MILPs sequentially: one to find $r_{\max}(\mathcal{D})$, and one to find $s_{\max}(r_{\max}(\mathcal{D}))$.

only the value of $r_{\max}(\mathcal{D})$ for digraphs. These include Algorithm 3.4, a modi fied version of $DetermineRobustness$ which determines $r_{\max}(\mathcal{D})$, the MILP formulation from Theorem 3.4 (denoted *r-Rob. MILP*), the lower bound MILP formulation from Theorem 3.6 (denoted *r-Rob. Lower Bnd*), and the upper bound MILP formulation from Theorem 3.7 (denoted *r-Rob. Upper Bnd*). These algorithms are tested on the four types of graphs described above with values of n ranging from 7 to 15. Additionally, the MILP formulations are tested on digraphs with values of n ranging from 17 to 25. Again, 100 graphs per graph type and combination of n and p (or n and k, depending on the respective graph type) are randomly generated, and the algorithms are run on each graph. Overall, 10,800 graphs are analyzed with Algorithm 3.4 and 16,800 graphs are analyzed by each of the three MILP formulations. The time for each algorithm to determine $r_{\max}(\mathcal{D})$ is averaged for each combination of n, p or k, and graph type. The average, minimum, and maximum times per combination are plotted in Figure 3.13. A time limit of 10^3 seconds is again imposed on all three of the MILP formulations, but out of the 16,800 graphs tested there are no instances where this time limit was violated.

Some of the same patterns as in the first set of simulations (with the $DetermineRobustness$ and (r,s)-*Rob. MILP* algorithms) are evident in the second set of simulations. The *Mod. Det. Rob.* algorithm also terminates if a pair of subsets S_1 and S_2 are found which are both 0-reachable, which is likely the reason for the small minimum computation time of this algorithm for several of the graphs. *Mod. Det. Rob.* generally has a lower average computational time for $n \in \{7, 8\}$, again likely due to the speed of checking the relatively low number of unique nonempty, disjoint subset pairs as compared to solving the MILPs. It is not clear why the r-*Rob. Upper Bnd* MILP exhibits high average and maximum computational times for the k-out random digraphs. Future work will further analyze graph characteristics which negatively affect the convergence time of the MILP formulations.

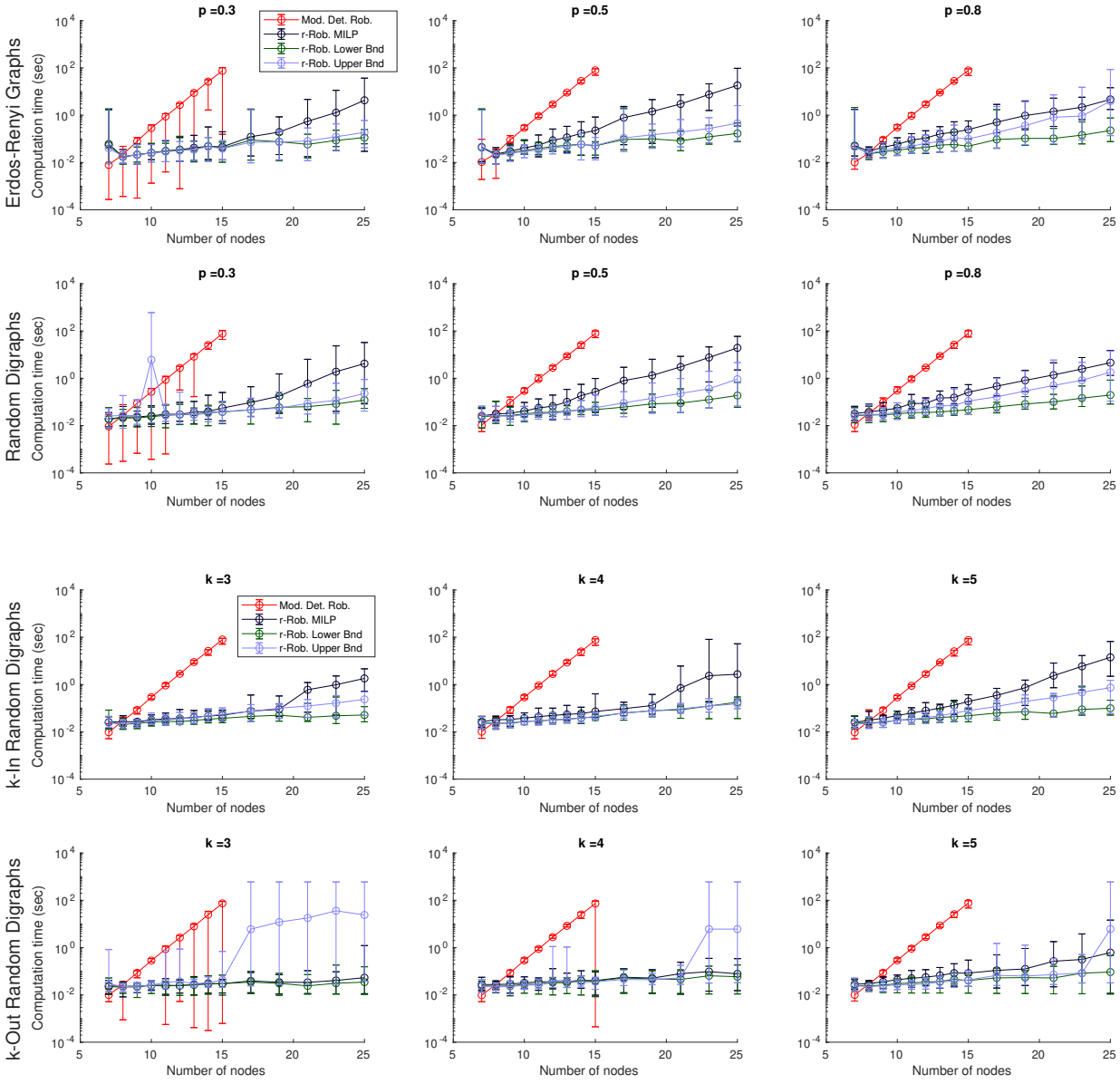


Figure 3.13: Comparison of the *Mod. Det. Rob.* algorithm (Algorithm 3.4) which determines $r_{\max}(\mathcal{D})$ to three MILP formulations. The first MILP formulation labeled *r-Rob. MILP* is an implementation of Theorem 3.4 and calculates $r_{\max}(\mathcal{D})$ exactly. The MILP formulation labeled *r-Rob. Lower Bnd* is an implementation of Theorem 3.6 and calculates a lower bound on $r_{\max}(\mathcal{D})$. The MILP formulation labeled *r-Rob. Upper Bnd* is an implementation of Theorem 3.7 and calculates an upper bound on $r_{\max}(\mathcal{D})$. The interpolating lines and circles represents the average computation time over 100 digraphs for each value of n, the upper and lower lines represent the maximum and minimum computation times, respectively, over the 100 trials for each n.

3.5 Conclusion

This chapter presents novel approaches for the problems of constructing robust graphs and determining the robustness of digraphs. It was demonstrated that the class of k-circulant digraphs has robustness properties as a function of the parameter k, making it conducive to constructing scalable networks with known levels of robustness. In addition, a method was presented for determining the r- and (r, s)-robustness of digraphs using mixed integer linear programming (MILP). The advantages of the MILP formulations and branch-and-bound algorithms over prior algorithms are discussed, and the performance of the MILP methods to the $DetermineRobustness$ algorithm is compared.

Much work remains to be done in the area of robustness determination. The results in this chapter merely open the door for the extensive literature on mixed integer programming to be applied to the robustness determination problem. Future work will focus on applying more advanced integer programming techniques to the formulations in this chapter to yield faster solution times. In particular, the Laplacian matrix exhibits a high degree of structure and plays a central role in the MILP formulations presented in this chapter. A promising direction for investigation is to explore how to leverage this structure to determine the robustness of digraphs more efficiently.

3.6 Appendix: Description of Algorithm Implementations

This section gives additional details about the implementations of the algorithms tested in the Simulation. Algorithm 3.2 provides the details about the implementation of the $DetermineRobustness$ algorithm implemented in the simulations. One modification was made to the $DetermineRobustness$ algorithm to ensure accuracy of results. In the first line of the original $DetermineRobustness$ algorithm, r is initialized with $\min\left(\delta^{\text{in}}(\mathcal{D}), \lceil n/2 \rceil\right)$. This yields incorrect results however for directed spanning trees where the in-degree of the root node is zero. Consider the digraph depicted in Figure 3.14. Here, $\delta^{\text{in}}(\mathcal{D}) = 0$, since the left agent has no in-neighbors. This implies that the original $DetermineRobustness$ algorithm would initialize $r \leftarrow 0$ and return $(0, n)$ as the values of $(r_{\max}(\mathcal{D}), s_{\max}(r_{\max}(\mathcal{D})))$. However, Figure 3.14 shows that for all nonempty, disjoint subsets S_1 and S_2, at least one is 1-reachable. The depicted graph is therefore $(1, 1)$-robust with $r_{\max}(\mathcal{D}) = 1$ and $s_{\max}(r_{\max}(\mathcal{D})) = 1$. In fact, initializing $r \leftarrow \delta^{\text{in}}(\mathcal{D})$ will always yield this error for any directed spanning tree where the in-degree of the root node is 0. This happens because any digraph is 1-robust if and only if it contains a rooted out-branching [2, Lemma 7], yet $DetermineRobustness$ initializes $r \leftarrow \delta^{\text{in}}(\mathcal{D}) = 0$ which results in termination at line 23. In Algorithm 3.2, r is instead initialized with $\min\left(\max\left(\delta^{\text{in}}(\mathcal{D}), 1\right), \left\lceil \frac{n}{2} \right\rceil\right)$. This initializes $r \leftarrow 1$ if $\delta^{\text{in}}(\mathcal{D}) = 0$, since it is still possible for the digraph to be 1-robust.

Algorithm 3.2 [96] DETERMINEROBUSTNESS($A(\mathcal{D})$)

comment: $A(\mathcal{D})$ is the adjacency matrix of the graph.
$r \leftarrow \min\left(\max\left(\delta^{\text{in}}(\mathcal{D}), 1\right), \left\lceil \frac{n}{2} \right\rceil\right)$
$s \leftarrow n$
comment: $\delta^{\text{in}}(\mathcal{D})$ is the min. in-degree of nodes in $\mathcal{D}$
for each $k \leftarrow 2$ to n **do**
    for each $K_i \in \mathcal{K}_k$ $(i = 1, 2, \ldots, \binom{n}{k})$ **do**
    comment: $\mathcal{K}_k$ is the set of $\binom{n}{k}$ unique subsets of $\mathcal{V}$
        for each $P_j \in \mathcal{P}_{K_i}$ $(j = 1, 2, \ldots, 2^{k-1} - 1)$ **do**
        comment: $\mathcal{P}_{K_i}$ is set of partitions of K_i with exactly two nonempty parts
            comment: $\mathcal{P}_j = \{S_1, S_2\}$
            $isRSRobust \leftarrow$ ROBUSTHOLDS$(A(\mathcal{D}), S_1, S_2, r, s)$
            if ($isRSRobust ==$ **false**) **and** $s > 0$ **then**
                $s \leftarrow s - 1$
            end if
            while $isRRobust ==$ **false and** $(r > 0)$ **do**
                while $isRSRobust ==$ **false and** $(s > 0)$ **do**
                    $isRRobust \leftarrow$ ROBUSTHOLDS$(A(\mathcal{D}), S_1, S_2, r, s)$
                    if not $isRSRobust$ **then**
                        $s \leftarrow s - 1$
                    end if
                end while
                if $isRSRobust ==$ **false then**
                    $r \leftarrow r - 1$
                    $s \leftarrow n$
                end if
            end while
            if $r == 0$ **then**
                return (r, s) ▷ *Implies* $r_{\max}(\mathcal{D}) = 0$
            end if
        end for
    end for
end for
return (r, s) ▷ *Returned values are* $(r_{\max}(\mathcal{D}), s_{\max}(r_{\max}(\mathcal{D})))$

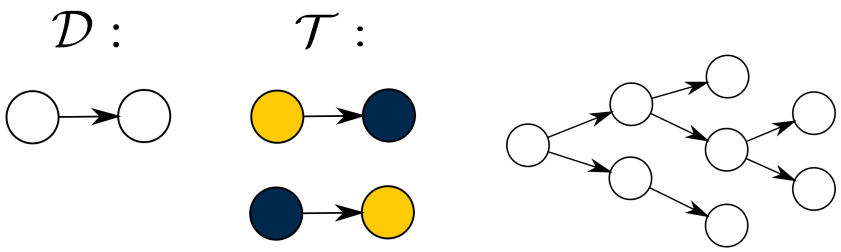


Figure 3.14: (Left) Example of a digraph which has $\delta^{\text{in}}(\mathcal{D}) = 0$ but which is 1-robust. The graph is depicted on the far left, and all possible (S_1, S_2) pairs in $\mathcal{T}$ are depicted on the close left. (Right) Fig A.2. A rooted out-branching, where the in-degree of the root node (far left) is zero. All digraphs containing a rooted outbranching are at least $(1, 1)$-robust [2].

To compare the MILP methodologies of this chapter with the $DetermineRobustness$ algorithm, Algorithm 3.3 was used which determines the values of $(r_{\max}(\mathcal{D}), s_{\max}(r_{\max}(\mathcal{D})))$. The first

Algorithm 3.3 (r, s)-ROB. MILP

1: $r \leftarrow r_{\max}(\mathcal{D})$ from MILP in Theorem 3.4
2: **if** r == 0 **then**
3: $\quad s \leftarrow n$
4: **else**
5: $\quad$ **if** $\delta^{\text{in}}(\mathcal{D}) \geq \lfloor n/2 \rfloor + r - 1$ **then** ▷ *See Property 5.23 in [219]*
6: $\quad\quad s \leftarrow n$
7: $\quad$ **else**
8: $\quad\quad \bar{s}_{\min}(r) \leftarrow$ from MILP in Theorem 3.5
9: $\quad\quad s \leftarrow \bar{s}_{\min}(r) - 1$
10: $\quad$ **end if**
11: **end if**
12: **return** (r, s)

part of Algorithm 3.3 uses the formulation in Theorem 3.4 to determine $r_{\max}(\mathcal{D})$. If $r_{\max}(\mathcal{D}) = 0$, then $s \leftarrow n$ and the algorithm returns (r, s). If $r_{\max}(\mathcal{D}) > 0$, then the algorithm determines in line 5 whether the minimum in-degree of the graph $\delta^{\text{in}}(\mathcal{D}) \geq \lfloor n/2 \rfloor + r - 1$. By Property 5.23 in [219], if this is satisfied then the digraph is (r, s)-robust for all $1 \leq s \leq n$. Since for $r > 0$ the MILP in Theorem 3.5 is infeasible if and only if $s = n$, this test attempts to help the MILP solver avoid a fruitless search for a feasible solution if s is indeed equal to n. This test works for graphs with large minimum in-degrees (e.g. complete graphs), but since it is a sufficient condition only it may not always detect when $s_{\max}(r) = n$. Determining a more rigorous test to determine when $s_{\max}(r) = n$ is left for future work. Finally, if the test in line 5 fails then the MILP formulation in Theorem 3.5 is performed to determine $\bar{s}_{\min}$. Since $s_{\max}(r) = \bar{s}_{\min}(r) - 1$, the value of $s_{\max}(r)$ is stored in s and the algorithm returns (r, s).

The MILP algorithms in Section 3.4.1 consider r-robustness, which is equivalent to $(r, 1)$-robustness [96, Property 5.21], [2, Section VII-B]. Since they effectively do not consider values of s greater than 1, it is unfair to compare them directly with the $DetermineRobustness$ algorithm. Algorithm 3.4 is a modified version of $DetermineRobustness$ which only considers $(r, 1)$-robustness. This is accomplished by initializing $s \leftarrow 1$ in lines 2 and 20 instead of $s \leftarrow n$. Algorithm 3.4 is labeled *Mod. Det. Rob.* in the simulation legends.

Algorithm 3.4 Modified version of DETERMINEROBUSTNESS

comment: $A(\mathcal{D})$ is the adjacency matrix of the graph
$r \leftarrow \min\left(\max\left(\delta^{\text{in}}(\mathcal{D}), 1\right), \left\lceil \frac{n}{2} \right\rceil\right)$
$s \leftarrow 1$ ▷ *(Different than Alg. 3.2 in [96])*
comment: $\delta^{\text{in}}(\mathcal{D})$ is the min. in-degree of nodes in $\mathcal{D}$
for each $k \leftarrow 2$ to n **do**
comment: $\mathcal{K}_k$ is the set of $\binom{n}{k}$ unique subsets of $\mathcal{V}$
    for each $K_i \in \mathcal{K}_k$ $(i = 1, 2, \ldots, \binom{n}{k})$ **do**
        for each $P_j \in \mathcal{P}_{K_i}$ $(j = 1, 2, \ldots, 2^{k-1} - 1)$ **do**
        comment: $\mathcal{P}_{K_i}$ is set of partitions of K_i into S_1 and S_2
            $isRSRobust \leftarrow$ ROBUSTHOLDS$(A(\mathcal{D}), S_1, S_2, r, s)$
            if ($isRSRobust ==$ **false**) **and** $s > 0$ **then**
                $s \leftarrow s - 1$
            end if
            while $isRRobust ==$ **false and** $(r > 0)$ **do**
                while $isRSRobust ==$ **false and** $(s > 0)$ **do**
                    $isRRobust$ $\leftarrow$ ROBUSTHOLDS(A, S_1, S_2, r, s)
                    if not $isRSRobust$ **then**
                        $s \leftarrow s - 1$
                    end if
                end while
                if $isRSRobust ==$ **false then**
                    $r \leftarrow r - 1$
                    $s \leftarrow 1$ ▷ *(Diff. than Alg. 3.2 in [96])*
                end if
            end while
            if $r == 0$ **then**
                return r
            end if
        end for
    end for
end for
return r

CHAPTER 4

Resilient Broadcast

4.1 Introduction

As discussed in Chapter 1 of this dissertation, prior literature has studied the problem of resiliently broadcasting information from a central source throughout a network [229–231]. This formulation involves a leader (sometimes called a "dealer") robot that seeks to propagate a static message to all normally-behaving nodes in a network, despite the presence of faulty or adversarial nodes that broadcast misinformation. Prior literature presents several algorithms and graph-theoretic conditions under which the reference value is accepted by all normally-behaving agents in the network in finite time.

A limitation of this approach is that the message being broadcast is inherently only able to contain static information. This complicates the application to scenarios where the reference value is inherently time-varying. An interesting question is whether the prior approaches to resilient broadcast could be used to transmit information that is time-varying in nature, e.g., the information about a time-varying center of formation moving along a trajectory. Another limitation to prior work is that resilient broadcast algorithms often assume that there is a single leader who is *trusted*, i.e. invulnerable to adversarial attacks or faults.

This chapter presents a method by which leaders can resiliently propagate complete knowledge of entire time-varying trajectories to all followers in finite time, despite the presence of faulty or adversarial agents. Specifically, we consider a network of mobile robots with the objective to track a time-varying trajectory point in a formation. Initially only a set of leader robots have knowledge of the trajectory. The structure of the trajectory, including the Bezier path parameters and the timing law information, is encoded via static parameters that are propagated from the set of normally-behaving leaders to the followers via the proposed algorithm. In addition to being resilient against adversarial misinformation, we demonstrate that our method is robust to several additional sources of errors including bounded clock synchronization errors and perturbations of

the trajectory parameters. Follower robots are then able to uniquely reconstruct their desired trajectories, and track them using nominal control inputs that are computed onboard each individual follower in a distributed fashion.

The contributions of this chapter are as follows: first, a novel method is presented for propagating vector-valued messages within a finite time from a set of leaders to normally-behaving followers. Under proper graph-theoretic conditions, the method is proven to be resilient to faults and adversarial attacks, and can operate under asynchronous communication. Second, theoretical bounds on the allowable error between leaders' vector-valued messages are derived, which guarantee that the maximum error between normal leaders' and followers' reconstructed formational trajectories are also bounded. The robustness of the proposed trajectory propagation method to parameter perturbations and clock synchronization errors is analyzed.

This chapter is organized as follows: in Section 4.2 we give an overview of relevant concepts from prior literature on resilient broadcasting. In Section 4.3 we present the notation and problem formulation for the paper. In Section 4.4 we analyze the sensitivity of the proposed method to perturbations in the trajectory data and clock synchronization errors. In Section 4.5 we present a method for resiliently propagating vector-valued messages from normally-behaving leaders to all normally-behaving followers in a multi-robot network. In Section 4.6 we present simulations of our results in a scenario involving a leader-follower network subject to clock perturbations and robots subject to adversarial attacks. In Section 4.7 we give a brief conclusion and directions for future work.

4.2 Preliminaries on Resilient Broadcasting

This section gives a brief overview of the resilient broadcast problem and the Certified Propagation Algorithm (CPA). We consider a network of N agents with a communication structure described by a digraph $\mathcal{D} \in (\mathcal{V}, \mathcal{E})$. One agent $i_L \in \mathcal{V}$ is designated as the *leader*[1] and possesses a message m. For now we assume that the network is synchronous with time steps $t \in \mathbb{Z}_{\geq 0}$, $t_0 = 0$. At each time step t each agent $i \in \mathcal{V}$ is able to receive messages from its in-neighbors $j \in \mathcal{V}_i$ and send messages to its out-neighbors $j \in \mathcal{V}_i^{\text{out}}$. Agents may also choose to send no message at all at a given time step.

Within the network, a subset of agents may behave adversarially, which is defined as follows:

Definition 4.1. *An agent $k \in \mathcal{V}$ is defined as* adversarial *if there exists $t' \geq 0$ such that at least one of the following conditions holds:*

1. *Agent k sends a value $m_k \neq m$ to at least one of its out-neighbors,*

[1]The leader is sometimes called "dealer" in the literature

2. *Agent k sends different values to different out-neighbors; i.e. there exist $i_1, i_2 \in \mathcal{V}_k^{out}$ such that agent k sends message m_1 to i_1, message m_2 to i_2, and $m_1 \neq m_2$.*

The set of adversarial agents is denoted $\mathcal{A} \subset \mathcal{V}$. All agents that are not adversarial are called normal agents, with the set of normal agents denoted $\mathcal{N} \subset \mathcal{V}$.

The objective of the normal agents is to ensure that the message m is propagated from the leader i_L to all normal agents in the network despite the behavior of the adversarial agents. The Certified Propagation Algorithm was proposed towards this end, and is summarized as follows:

1. At $t = 0$, the leader sends m to all of its out-neighbors.

2. Any agent $i \in \mathcal{N}$ that is an out-neighbor of the leader, i.e. $i \in \mathcal{V}_{i_L}^{\text{out}}$, accepts the message m, sends the message to all its out-neighbors, and terminates.

3. Any agent that is not an out-neighbor of the dealer and that receives $F + 1$ copies of a message m' from its in-neighbors accepts the message m', sends m' to its out-neighbors, and terminates.

Under an F-total or F-local[2] set $\mathcal{A}$, the CPA prevents any normally-behaving agent $i \in \mathcal{V}$ from accepting adversarial information since there will be no more than F adversaries in agent i's in-neighborhood. However, the condition that all normally-behaving agents accept the correct message m from the leader depends upon the graph-theoretic structure of the communication links in the network. A sufficient graph theoretic condition for *undirected* graphs that guarantees that CPA is able to achieve resilient broadcast of m to all normally-behaving agent is given in [73]. More specifically, if no F-partial-local-pair (F-plp) cut[3] exists in the graph then CPA achieves resilient broadcast from the leader to all normal followers. For general graphs determing if an F-plp cut with respect to a chosen leader exists is NP-hard [73]. However, a specific class of directed graphs called *Mode Estimation Acyclic Directed Graphs* (MEDAGs) introduced in [194] can be shown to satisfy the required sufficient conditions for CPA to achieve resilient broadcast in the presence of an F-local adversary set.

4.3 Notation and Problem Formulation

A summary of notation and the problem formulation are given in this section. The set of real numbers and integers are denoted $\mathbb{R}$ and $\mathbb{Z}$, respectively. The set of nonnegative reals and integers

[2]The concepts of F-total and F-local are outlined in Definitions 2.1 and 2.2.

[3]The terminology used in [73] is "t-plp" cut rather than "F-plp" cut. We use the variable name F to more closely match terminology in prior chapters of this dissertation.

are denoted $\mathbb{R}_+$ and $\mathbb{Z}_+$, respectively. The cardinality of a set S is denoted $|S|$. The set union, intersection, and set difference operations of two sets S_1 and S_2 are denoted by $S_1 \cup S_2$, $S_1 \cap S_2$, and $S_1 \backslash S_2$ respectively. We denote $\bigcup_{i=1}^{n} S_i = S_1 \cup S_2 \cup \ldots \cup S_n$. We denote the ball of radius $r \in \mathbb{R}$ centered at $x \in \mathbb{R}$ as $B(x,r) = \{z \in \mathbb{R} : |x - z| \leq r\}$. A digraph is denoted as $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices or robots, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges. An edge from i to j, $i, j \in \mathcal{V}$, denoted as $(i,j) \in \mathcal{E}$, represents the ability of i to send information to j. Note that for digraphs $(i,j) \in \mathcal{E} \implies (j,i) \in \mathcal{E}$. The set of in-neighbors of robot i is denoted $\mathcal{V}_i = \{j \in \mathcal{V} : (j,i) \in \mathcal{E})\}$. The Lie derivative of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ along a vector field $f : \mathbb{R}^n \to \mathbb{R}^n$ is denoted $L_f h(x) \triangleq \frac{\partial h}{\partial x} f(x)$. The set of out-neighbors of each robot i is denoted $\mathcal{V}_i^{out} = \{k \in \mathcal{V} : (i,k) \in \mathcal{E}\}$. The jth derivative of a function $f : \mathbb{R} \to \mathbb{R}^n$ with respect to its variable is denoted $f^{(j)}(\cdot)$.

Given a digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ and a nonnegative integer $F \in \mathbb{Z}_+$, a subset $S \subset \mathcal{V}$ is *F-total* if $|S| \leq F$. A subset $S \subset \mathcal{V}$ is *F-local* if for all $i \in \mathcal{V} \backslash S$ it holds that $|\mathcal{V}_i \cap S| \leq F$. It follows that any F-total model is also simultaneously F-local.

4.3.1 Problem Formulation

We consider a network of N mobile robots. The robots are represented by the set of indexed nodes $\mathcal{V} = \{1, \ldots, N\}$, and communication links between robots are represented by a directed graph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$. The set of robots $\mathcal{V}$ is partitioned into a set of *leaders*, denoted $\mathcal{L} \subset \mathcal{V}$, and a set of *followers*, denoted $\mathcal{S}_f \subset \mathcal{V}$, respectively, such that $\mathcal{L} \cup \mathcal{S}_f = \mathcal{V}$ and $\mathcal{L} \cap \mathcal{S}_f = \emptyset$.

The objective of the follower robots in the network is to track a reference trajectory in $\mathbb{R}^n$ that is initially known only by the leaders. A common approach used in prior literature to accomplish this objective is via a leader-follower asymptotic consensus approach [232, 233], where leaders transmit their current states to the network and the followers achieve consensus to these states. However, under this approach any communication delays or asynchrony between the leaders and followers may have a negative effect on trajectory tracking. This chapter considers an alternative approach in which leaders propagate *full knowledge* of the entire trajectory to followers as follows. Each trajectory consists of a *path* and a *timing law* for traversing the path [234]. The trajectory begins at some time $t_i \in \mathbb{R}$, $t_i \geq 0$ and ends at time $t_f \in \mathbb{R}$, $t_f > t_i$. We assume that each path is expressed as a ηth order Bezier curve $P : \mathbb{R} \to \mathbb{R}^n$ with $\eta \in \mathbb{Z}_+$. More specifically, $P(\cdot)$ has the form:

$$P(s) = \sum_{k=0}^{\eta} \alpha_k b_{k,\eta}(s), \tag{4.1}$$

where the points $\alpha_k \in \mathbb{R}^n$, $k \in \{0, \ldots, \eta\}$ are the *control points* of the Bezier curve, and the

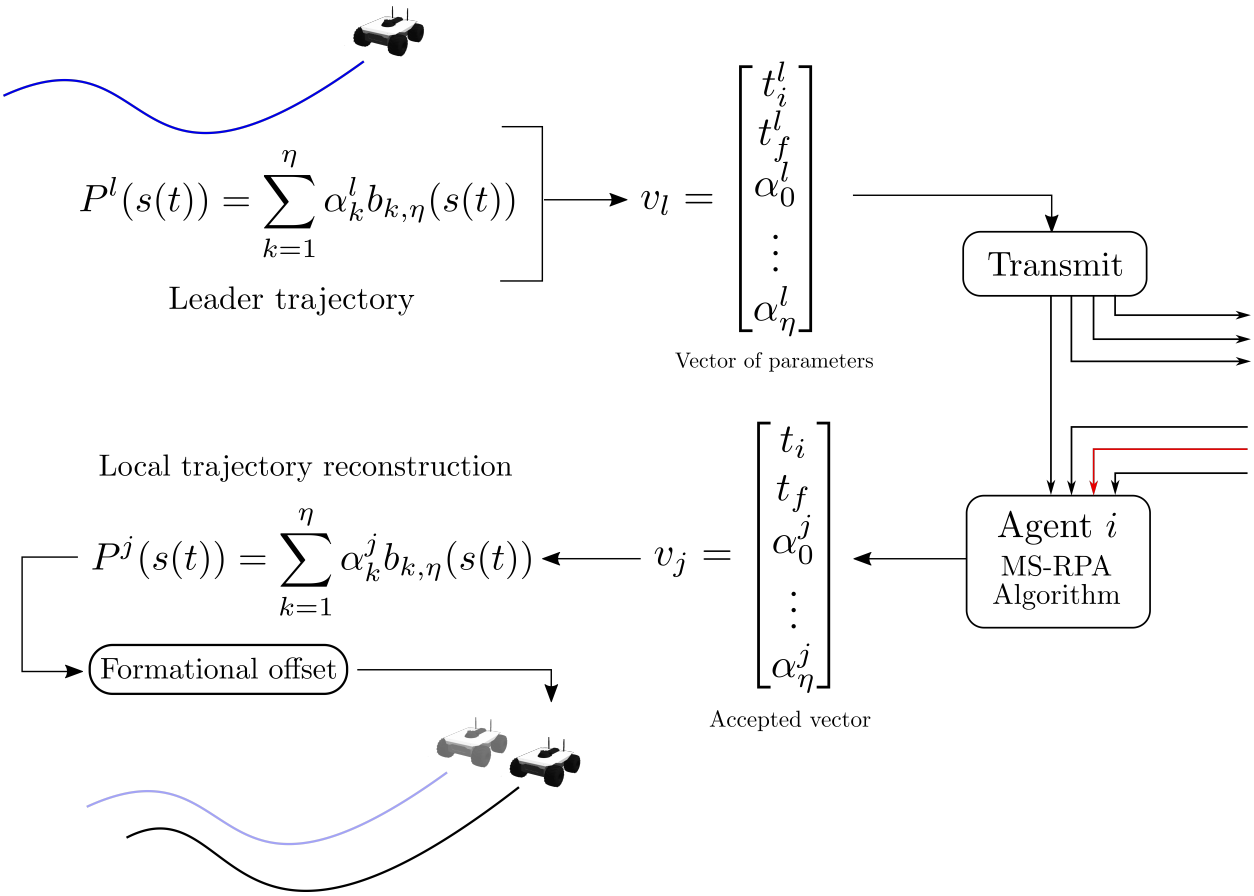


Figure 4.1: An overview of the trajectory propagation method using parameter vectors. Each leader broadcasts a vector of static parameters representing a Bezier-curve-based trajectory. Followers receive messages from both normally-behaving robots and misbehaving robots. From its received information each follower accepts a parameter vector and uses it to reconstruct a trajectory locally.

functions $b_{k,\eta} : \mathbb{R} \to \mathbb{R}$ are the Bernstein basis polynomials of degree η. More information about Bezier curves and their properties can be found in the Appendix Section 4.8.1. Given initial and final times $t_i, t_f \in \mathbb{R}$, a general timing law $s : \mathbb{R} \to \mathbb{R}$ is assumed to have the form

$$s(t) = \begin{cases} s_i & \text{if } t < t_i, \\ f_s(t, t_i, t_f) & \text{if } t_i \leq t \leq t_f, \\ s_f & \text{if } t > t_f, \end{cases} \tag{4.2}$$

with $f_s : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to [s_i, s_f]$ being non-decreasing in t. The timing law $s(t)$ defines a unique time-varying point $x(s(t))$, which traverses the path $x(s)$. Different definitions of $s(t)$ result in different velocity profiles for this point. For purposes of presentation we will let $s_i = 0$ and $s_f = 1$ unless otherwise noted. It is assumed that all leaders and followers share a common reference

frame and are able to locally compute the Bernstein basis polynomials $b_{k,\eta}(t)$. For simplicity, we will initially assume that the timing law function $f_s(\cdot,\cdot,\cdot)$ is known to all robots in the network, but a relaxation of this assumption will be discussed later in the paper.

Under these assumptions, the reference trajectory can be uniquely represented by the following vector of *static* parameter values:

$$v_r = \begin{bmatrix} t_i & t_f & \alpha_1^T & \cdots & \alpha_\eta^T \end{bmatrix}^T. \tag{4.3}$$

Given a vector of parameters, each follower is able to locally reconstruct the corresponding Bezier curve and timing law $s(t)$, yielding a corresponding trajectory. The trajectory reconstructed by robot $i \in \mathcal{V}$ is denoted $P^i(t)$. Each robot i maintains an internal vector v_i containing the parameters from which it reconstructs $P^i(t)$. If $v_i = v_r$, the locally reconstructed polynomial satisfies $P^i(t) = P(t)$, $\forall t_i \leq t \leq t_f$, and the robot then has access to $P(t)$ and its time derivatives. We define the pointwise trajectory error as:

$$e_{ij}(t) = P^i(t) - P^j(t),\ i,j \in \mathcal{V}. \tag{4.4}$$

To propagate the parameters for the desired trajectory to the remainder of the network, leaders send vector messages v_l, $l \in \mathcal{L}$ to their out-neighbors. Followers are able to send vector messages to their out-neighbors, and receive vector messages from their in-neighbors. These communications need not be synchronous. The notation $v_j^i(t)$ represents the vector message received by robot i from robot j at time t. An overview of this process is depicted in Figure 4.1.

Several sources of error are considered in this setting however. First, robots in the network may be subject to misbehavior due to faults and adversarial attacks.

Definition 4.2. *An robot $k \in \mathcal{V}$ is called* misbehaving *if at least one of the following conditions holds:*

1. *Robot k sends arbitrary, unbounded messages v_k to at least one of its out-neighbors*
2. *There exists a time t such that robot k sends different messages to different out-neighbors; i.e. there exist $i_1, i_2 \in \mathcal{V}$ such that $v_k^{i_1}(t) \neq v_k^{i_2}(t)$.*

Note that both followers and leaders may be adversarial. The set of misbehaving robots is denoted $\mathcal{A} \subset \mathcal{V}$. The set of *normally-behaving robots* is defined as $\mathcal{N} = \mathcal{V} \backslash \mathcal{A}$. The sets of normal and misbehaving leaders are denoted as $\mathcal{L}^{\mathcal{N}} = \mathcal{L} \cap \mathcal{N}$ and $\mathcal{L}^{\mathcal{A}} = \mathcal{L} \cap \mathcal{A}$, respectively. The sets of normal and misbehaving followers are denoted as $S_f^{\mathcal{N}} = S_f \cap \mathcal{N}$ and $S_f^{\mathcal{A}} = S_f \cap \mathcal{A}$, respectively.

Second, the vector messages v_l, $l \in \mathcal{L}$, of the normally-behaving leader robots may not precisely agree due to noise or numerical errors. More specifically, the maximum normed error between the parameter vectors of the normal leaders is defined as:

$$\epsilon_{lp} = \max_{l_1,\, l_2 \in \mathcal{L}^{\mathcal{N}}} \|v_{l_1} - v_{l_2}\|_\infty \,. \tag{4.5}$$

When $\epsilon_{lp} = 0$, all normal leaders possess exactly the same parameter vector; i.e., $v_{l_1} = v_{l_2}$, $\forall l_1, l_2 \in \mathcal{L}^{\mathcal{N}}$. This may not hold true in general however. Follower robots therefore need a method to select a parameter vector that is "close enough" in some sense to the vectors of normal leaders in $\mathcal{L}^{\mathcal{N}}$. This becomes particularly difficult considering the presence of misbehaving robots as defined above, which may propagate arbitrary, unbounded vector messages. The effect of differences between robots' Bezier control points is demonstrated in Figure 4.2.

Finally, due to jitter and clock synchronization errors, each robot $i \in \mathcal{V}$ may have a different estimate of time $t_i(t)$. Differences in time estimates will result in a different estimate of the time-varying reference point $P(t)$. The supremum error between normal robots' clocks is defined as follows:

$$\epsilon_{t,\sup} = \sup_{t \geq t_0} \max_{i,j \in \mathcal{N}} |t_i(t) - t_j(t)|. \tag{4.6}$$

Given the aforementioned problem setting and sources of errors and attacks, this chapter addresses the problem of guaranteeing that the maximum pointwise error between all normal leaders' and followers' reconstructed trajectories is bounded under given upper bounds on the perturbation and time errors ϵ_{lp} and $\epsilon_{t,\max}$, and in the presence of an F-local adversarial model.

Problem 4.1. *Let $\delta_{lp}, \delta_t > 0$. Given ϵ_{lp}, $\epsilon_{t,\sup}$, and an F-local adversarial set $\mathcal{A} \subset \mathcal{V}$, ensure that the maximum pointwise error between any two normal robots' reconstructed trajectory $e_{\max}(t) = \max_{i,j \in \mathcal{N}} P_i(t) - P_j(t)$ satisfies*

$$\|e_{\max}(t)\| \leq \alpha \left(\left\| \begin{bmatrix} \epsilon_{lp} \\ \epsilon_{t,\sup} \end{bmatrix} \right\| \right), \tag{4.7}$$

where $\alpha(\cdot)$ is a class-$\mathcal{K}$ function.

Remark 4.1. *The ability to upper bound the pointwise trajectory error has applications in scenarios where it is critical for agents to maintain a specified formation within a given tolerance. Examples of such scenarios include multi-agent cooperative payload transportation [235–237] and multi-spacecraft interferometry missions [238]. Establishing a relationship between clock*

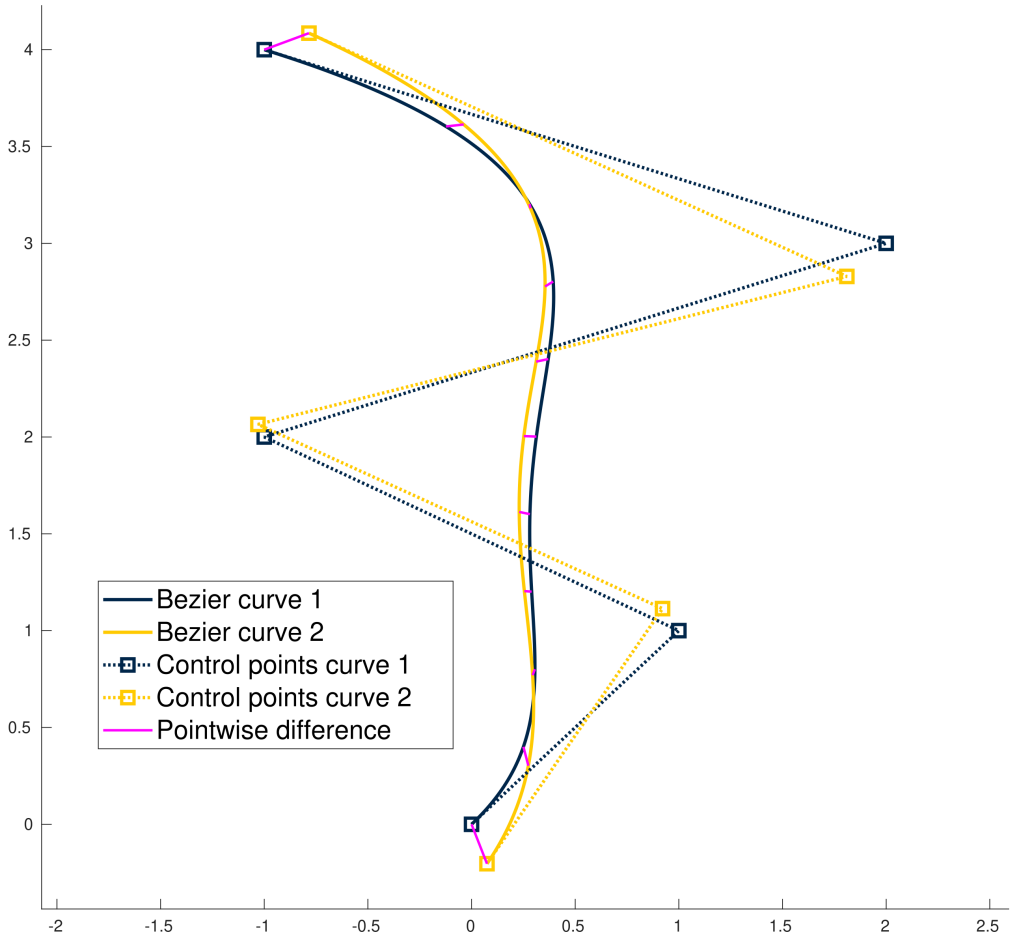


Figure 4.2: Example of the effects of Bezier control point perturbation. The control points for each Bezier curve are represented by the squares, with dotted connecting lines for visual clarity. The actual Bezier trajectories are the solid blue and yellow lines. The magenta lines represent various pointwise differences between points on the curves with corresponding $s \in [0, 1]$ value.

errors, parameter errors, and the resulting reconstructed trajectory errors in the form of (4.7) *ensures that the maximum resulting pointwise error between agents' reconstructed trajectories can be bounded by the magnitudes of clock and parameter errors.*

4.4 Sensitivity Analysis

If the Bezier curve or timing parameters between any two agents differ, each agent's reconstructed trajectory for the center of formation will also differ. Since the leaders' vector messages containing the Bezier parameters for the desired trajectory do not precisely agree due to noise or numerical error, there will therefore exist differences between the reconstructed trajectories of leaders and followers. The key question to be answered is how the error in parameters affects the error in the reconstructed trajectories.

Consider two Bezier-curve-based trajectories $P_1(s(t))$ and $P_2(s(t))$, with the error function $e_{12}(t) = P_1(s(t)) - P_2(s(t))$. This section provides the relation of the normed error $e_{12}(t)$ between two reconstructed trajectories with respect to the two sources of bounded errors considered in Problem 4.1:

1. Clock error $\epsilon_t(t) = t_1(t) - t_2(t)$
2. Parameter perturbation error $\epsilon_p(t) = \|v_1 - v_2\|_\infty$

The results in this section will be used later in the chapter to demonstrate that bounded error in the initial leader Bezier trajectory parameters will result in a bounded error between the leaders' and followers' reconstructed trajectories.

For simplicity of presentation and without loss of generality, this section will assume a normalized timing law $s(\cdot)$ satisfying $s_0 = 0$, $s_f = 1$, $s(t) = s_0 \ \forall t \leq t_i$, $s(t) = s_f \ \forall t \geq t_f$ for given $t_i, t_f \in \mathbb{R}$, $t_i < t_f$.

4.4.1 Sensitivity to Clock Synchronization Errors

Clock synchronization errors are first considered. Consider two robots j_1, j_2 with time measurements $t^{j_1}(t)$, $t^{j_2}(t)$ related by the equation:

$$t^{j_1}(t) - t^{j_2}(t) = \epsilon(t). \tag{4.8}$$

The variable t represents the nominal "actual time", which is not known to either robot and is used solely for error analysis. For notational brevity, the dependence of t^{j_1} and t^{j_2} on the actual time will be omitted.

The following result presents a bound on the normed error between a Bezier curve under two different time laws. Before presenting this result we define the following quantities recursively:

$$\Delta^0 \alpha_i = \alpha_i$$
$$\Delta^r \alpha_i = \Delta^{r-1}\alpha_{i+1} - \Delta^{r-1}\alpha_i$$

These expressions arise when discussing the derivatives of Bezier curves. More information on this point is given in the Appendix, Section 4.8.1.

Lemma 4.1. *Let $P(s(t))$ be an ηth order Bezier curve under the timing law $s(\cdot)$. Let t^{j_1} and t^{j_2} be two time sequences related by (4.8). The normed error $\|e(t)\| = \|P(s(t^{j_1})) - P(s(t^{j_2}))\|$ is bounded as*

$$\|e(t)\| \leq \sum_{j=1}^{\eta} (\eta - j + 1) \max_k \left\|\Delta^{j-1}\alpha_{k+1} - \Delta^{j-1}\alpha_k\right\| \frac{(\delta_{\max}(\epsilon_{\max}))^j}{j!} \tag{4.9}$$

where $\epsilon(t) \leq \epsilon_{\max}\ \forall t \in [t_i, t_f]$ and the term $\delta_{\max}(\epsilon_{\max})$ is defined as

$$\delta_{\max}(\epsilon_{\max}) = \max_{t\in[t_i,t_f]} \left(\max_{|h|\leq\epsilon_{\max}} |s(t+h) - s(t)| \right). \tag{4.10}$$

Proof. The error for a trajectory represented as an ηth order Bezier curve is

$$P(s(t^{j_1})) - P(s(t^{j_2})) = P(s(t^{j_2} + \epsilon(t))) - P(s(t^{j_2})) \tag{4.11}$$

The influence of $\epsilon(t)$ can be made explicit by considering the Taylor series for the Bezier curve:

$$P(s(t^{j_2} + \epsilon(t))) = P(s(t^{j_2})) + \sum_{j=1}^{\eta} \left(\frac{d^j}{(dt)^j} P(s(t^{j_2})) \right) \left(\frac{(\epsilon(t))^j}{j!} \right)$$

Using an alternate form of the Taylor series [239, Eq 3.2], (4.12) can be equivalently written as

$$P(s(t^{j_2} + \epsilon(t))) = P(s(t^{j_2})) + \sum_{j=1}^{\eta} \frac{P^{(j)}(s(t^{j_2}))}{j!} \left(s(t^{j_2} + \epsilon(t)) - s(t^{j_2})\right)^j \tag{4.12}$$

Substituting into (4.11) we obtain

$$P(s(t^{j_2} + \epsilon(t))) - P(s(t^{j_2})) = \sum_{j=1}^{\eta} \frac{P^{(j)}(s(t^{j_2}))}{j!} \left(s(t^{j_2} + \epsilon(t)) - s(t^{j_2})\right)^j \tag{4.13}$$

For brevity of notation, define

$$\delta_s(t, \epsilon) = s(t + \epsilon) - s(t). \tag{4.14}$$

This yields

$$P(s(t^{j_2} + \epsilon(t))) - P(s(t^{j_2})) = \sum_{j=1}^{\eta} \frac{P^{(j)}(s(t^{j_2}))}{j!} (\delta_s(t^{j_2}, \epsilon(t)))^j \tag{4.15}$$

Recall that $s(t) \in [0, 1]$ for $t \in [t_0, t_f]$. As outlined in the Appendix Section 4.8.1, this implies that

$$\left\| P^{(r)}(s) \right\| \leq (n - r + 1) \max_k \left\| \Delta^{r-1} \alpha_{k+1} - \Delta^{r-1} \alpha_k \right\|.$$

The magnitude of the right hand side of (4.15) can therefore be bounded as follows:

$$\begin{aligned} \left\| \sum_{j=1}^{\eta} \frac{P^{(j)}(s(t^{j_2}))}{j!} (\delta_s(t^{j_2}, \epsilon(t)))^j \right\| &\leq \sum_{j=1}^{\eta} \left\| P^{(j)}(s(t^{j_2})) \right\| \frac{\|\delta_s(t^{j_2}, \epsilon(t))\|^j}{j!} \\ &\leq \sum_{j=1}^{\eta} (\eta - j + 1) \max_k \left\| \Delta^{j-1} \alpha_{k+1} - \Delta^{j-1} \alpha_k \right\| \frac{\|\delta_s(t^{j_2}, \epsilon(t))\|^j}{j!} \end{aligned} \tag{4.16}$$

From equation (4.15) it follows that equation (4.16) is therefore an upper bound on the value of $\|P(t^{j_1}) - P(t^{j_2})\|$. Given the upper bound $\epsilon_{\max}$ on the magnitude of the error $\epsilon(t)$ in the interval $t \in [t_0, t_f]$, i.e. $\max_{t \in [t_0, t_f]} |\epsilon(t)| \leq \epsilon_{\max}$, the maximum value of $\delta_s(\cdot, \cdot)$ can be calculated as

$$\begin{aligned} \delta_{\max}(\epsilon_{\max}) &= \max_{t \in [t_0, t_f]} \left(\max_{|h| \leq \epsilon_{\max}} |\delta_s(t, h)| \right) \\ &= \max_{t \in [t_0, t_f]} \left(\max_{|h| \leq \epsilon_{\max}} |s(t + h) - s(t)| \right). \end{aligned} \tag{4.17}$$

This can be substituted into (4.16) to yield the following upper bound:

$$\left\| P(t^{j_1}) - P(t^{j_2}) \right\| \leq \sum_{j=1}^{\eta} (\eta - j + 1) \max_k \left\| \Delta^{j-1} \alpha_{k+1} - \Delta^{j-1} \alpha_k \right\| \frac{(\delta_{\max}(\epsilon_{\max}))^j}{j!},$$

which concludes the proof. □ □

Note that the bound in Lemma 4.1 is independent of time. In the particular case when $s(\cdot)$ is

linear, e.g., $s = \frac{t-t_i}{t_f-t_i}$, it is straightforward to calculate the value of $\delta_{\max}$ from the time derivative $\frac{d}{dt}s(t)$ as $\delta_{\max} = \frac{\epsilon_{\max}}{t_f-t_i}$.

The next two results will be useful for solving Problem 4.1. Corollary 4.1 demonstrates that the upper bound (4.9) on the clock synchronization error is a class-$\mathcal{K}$ function of the clock error $\epsilon_t(t)$.

Corollary 4.1. *The right hand side of* (4.9) *is a class-$\mathcal{K}$ function with respect to the variable* $\epsilon_{\max}$.

Proof. From (4.9) we define the following notation for brevity:

$$g_j(\delta) = (\eta - j + 1) \max_k \left\| \Delta^{j-1}\alpha_{k+1} - \Delta^{j-1}\alpha_k \right\| \frac{(\delta)^j}{j!},$$
$$j \in \{1, \ldots, \eta\}.$$

It is straightforward to show that $\delta_{\max}(\cdot)$ from (4.10) is a class-$\mathcal{K}$ function on the domain $[0, \infty)$ (observe that $\delta_{\max}(0) = 0$, $\epsilon_1 < \epsilon_2$ implies that $\delta_{\max}(\epsilon_1) < \delta_{\max}(\epsilon_2)$ for $\epsilon_1, \epsilon_2 > 0$, and $\delta_{\max}(\cdot)$ is continuous in its argument). Next, consider each $g_j(\cdot)$. Observe that $g_j(0) = 0$. Furthermore, observe that $(1/j!)(\eta - j + 1) \left\| \Delta^{j-1}\alpha_{k+1} - \Delta^{j-1}\alpha_k \right\| > 0$ for all $1 \leq j \leq \eta$. Since for all $j \geq 1$ we have $\delta_1 < \delta_2$ implies $(\delta_1)^j < (\delta_2)^j$ for $\delta_1, \delta_2 \in [0, \infty)$, each $g_j(\cdot)$ is therefore a class-$\mathcal{K}$ function. The result follows by observing that the right hand side of (4.9) is equal to $\sum_{j=1}^{\eta} g_j(\delta(\epsilon_{\max}))$ and recalling that sums and compositions of class-$\mathcal{K}$ functions yield class-$\mathcal{K}$ functions [209, Sec 4.4]. □

Next, given two Bezier curves $P^1(t), P^2(t)$ whose corresponding control points satisfy $\max_{k\in\{1,\ldots,\eta\}} \|\alpha_k^1 - \alpha_k^2\| \leq \epsilon_\alpha$, the next question is whether it is possible to bound the timing error $\|P^2(t^{j_1}) - P^2(t^{j_2})\|$ for trajectory 2 as a function of the timing error $\|P^1(t^{j_1}) - P^1(t^{j_2})\|$ for trajectory 1. The following lemma proves the affirmative to this question.

Lemma 4.2. *Consider two ηth order Bezier trajectories $P^1(\cdot)$ and $P^2(\cdot)$ with timing law $s(\cdot)$ defined as in* (4.2). *Let t^{j_1} and t^{j_2} be two time sequences related by* (4.8). *Suppose the control points for $P^1(\cdot), P^2(\cdot)$ satisfy $\max_{k\in\{1,\ldots,\eta\}} \|\alpha_k^1 - \alpha_k^2\| \leq \epsilon_\alpha$, $\epsilon_\alpha \in \mathbb{R}_+$. Then the following holds:*

$$\left| \left\| P^1(s(t^{j_1})) - P^1(s(t^{j_2})) \right\| - \left\| P^2(s(t^{j_1})) - P^2(s(t^{j_2})) \right\| \right| \leq 2\epsilon_\alpha.$$

Proof. By the reverse triangle inequality,

$$\begin{aligned}
&\left| \left\|P^1(s(t^{j_1})) - P^1(s(t^{j_2}))\right\| - \left\|P^2(s(t^{j_1})) - P^2(s(t^{j_2}))\right\| \right| \leq \\
&\qquad \left\|(P^1(s(t^{j_1})) - P^1(s(t^{j_2}))) - (P^2(s(t^{j_1})) - P^2(s(t^{j_2})))\right\|, \\
&\leq \left\|P^1(s(t^{j_1})) - P^2(s(t^{j_1}))\right\| + \left\|P^1(s(t^{j_2})) - P^2(s(t^{j_2}))\right\|, \\
&\leq \left\|\sum_{k=1}^{\eta}(\alpha_k^1 - \alpha_k^2)b_{k,\eta}(s(t^{j_1}))\right\| + \left\|\sum_{k=1}^{\eta}(\alpha_k^1 - \alpha_k^2)b_{k,\eta}(s(t^{j_2}))\right\|, \\
&\leq \sum_{k=1}^{\eta}\left\|\alpha_k^1 - \alpha_k^2)\right\| |b_{k,\eta}(s(t^{j_1}))| + \sum_{k=1}^{\eta}\left\|\alpha_k^1 - \alpha_k^2\right\| |b_{k,\eta}(s(t^{j_2}))|.
\end{aligned}$$

Since $\max_{k\in\{1,\ldots,\eta\}} \|\alpha_k^1 - \alpha_k^2\| \leq \epsilon_\alpha$, we obtain

$$\begin{aligned}
&\sum_{k=1}^{\eta}\left\|\alpha_k^1 - \alpha_k^2)\right\| |b_{k,\eta}(s(t^{j_1}))| + \sum_{k=1}^{\eta}\left\|\alpha_k^1 - \alpha_k^2\right\| |b_{k,\eta}(s(t^{j_2}))| \leq \\
&\qquad \epsilon_\alpha\left(\sum_{k=1}^{\eta} b_{k,\eta}(s(t^{j_1}))\right) + \epsilon_\alpha\left(\sum_{k=1}^{\eta} b_{k,\eta}(s(t^{j_2}))\right), \\
&= 2\epsilon_\alpha,
\end{aligned}$$

where the last line follows from the fact that the Bernstein basis polynomials $b_{k,\eta}(s)$ are nonnegative and form a partition of unity on the interval $s \in [0,1]$. □

Lemma 2 confirms the intuition that two Bezier trajectories that have similar control points will have similar timing errors for the same clock synchronization error model $\epsilon(t)$.

4.4.2 Sensitivity to Differences in Parameters

As per (4.3) there are two types of parameters that can be perturbed: time parameters t_i, t_f and control points $\alpha_0, \ldots, \alpha_n$. We consider perturbations to each type separately.

First we consider perturbations to t_i and t_f. By (4.2), any changes to these parameters results in a different $s(\cdot)$ function. We show in the next result that such perturbations can be modeled in the same manner as a clock synchronization error. Consider two Bezier-curve-based trajectories $P_1(s(t)), P_2(s(t))$ with the same control points $\alpha_1, \ldots, \alpha_\eta$ and timing law $s(t)$, but different time parameters t_0^1, t_f^1 and t_0^2, t_f^2. Denote the resulting $s(\cdot)$ functions for each curve as $s_1 : \mathbb{R} \to [0,1]$ and $s_2 : \mathbb{R} \to [0,1]$. Furthermore we will use the following notation for the inverse image of the

time instant t under $s(\cdot)$:

$$s_i^{-1}(t) \triangleq \{\tau \in \mathbb{R} : s_i(\tau) = t\},\ \tau \in \mathbb{R}. \tag{4.18}$$

Lemma 4.3. *The error between $P_1(t)$ and $P_2(t)$ with identical Bezier control points but different time parameters t_0^1, t_f^1 and t_0^2, t_f^2 can be expressed as a timing error $t + \epsilon_{1,2}(t)$, where*

$$\epsilon_{1,2}(t) \in s_1^{-1}(s_2(t)) - t. \tag{4.19}$$

The right hand side is understood in terms of Minkowski subtraction. Furthermore, when s_1 is strictly increasing on $[t_0, t_f]$ and $t \in [t_0, t_f]$, (4.19) holds with equality.

Proof. Since $s_1 : \mathbb{R} \to [0, 1]$ and $s_2 : \mathbb{R} \to [0, 1]$ and are both continuous, for all $t \in \mathbb{R}$ there exists an $\epsilon(t) \in \mathbb{R}$ such that $s_1(t + \epsilon(t)) = s_2(t)$. Rearranging, we have

$$t + \epsilon(t) \in s_1^{-1}(s_2(t)) \Rightarrow \epsilon(t) \in s_1^{-1}(s_2(t)) - t. \tag{4.20}$$

By definition, both $s_1 : [t_i, t_f] \to [0, 1]$ and $s_2 : [t_i, t_f] \to [0, 1]$ are surjections. If $s_1(\cdot)$ is strictly increasing then the mapping $s(t) : [t_0, t_f] \to [0, 1]$ is a bijection, implying that $s_1^{-1}(\cdot)$ is a single-valued function. □

This result implies that the error caused by perturbations to the parameters t_0, t_f can be analyzed using the same techniques presented previously for clock synchronization errors. For the specific case when $s(t)$ is linear, i.e. $f_s(t, t_i, t_f) = \frac{t - t_i}{t_f - t_i}$, the error $\epsilon_{1,2}(t)$ can be obtained explicitly as a function of the variables $t_i^1, t_f^1, t_i^2, t_f^2$, which is shown in the following corollary.

Corollary 4.2. *Consider the trajectories $P_1(t)$ with time parameters t_i^1, t_f^1 and $P_2(t)$ with time parameters t_i^2, t_f^2. If the function f_s from the definition of $s(t)$ in (4.2) is defined as $f_s(t, t_i, t_f) = \frac{t - t_i}{t_f - t_i}$, then the timing error $\epsilon_{1,2}(t)$ satisfies:*

$$|\epsilon_{1,2}(t)| \leq \max(|t_i^1 - t_i^2|, |t_f^1 - t_f^2|). \tag{4.21}$$

Proof. Given the definition of f_s and the parameters $t_i^1, t_f^1, t_i^2, t_f^2$, the timing laws for P_1 and P_2 satisfy

$$\begin{aligned} s_1(t) &= \frac{t - t_i^1}{t_f^1 - t_i^1},\ t \in [t_i^1, t_f^1], \\ s_2(t) &= \frac{t - t_i^2}{t_f^2 - t_i^2},\ t \in [t_i^2, t_f^2]. \end{aligned} \tag{4.22}$$

Consider the time interval $[\max(t_i^1, t_i^2), \max(t_f^1, t_f^2)]$. Choose two time instances t^1, t^2 such that $s_1(t^1) = s_2(t^2) = s^* \in [0, 1]$. Using (4.22) we obtain the following:

$$s_1(t) = s^* = \frac{t^1 - t_i^1}{t_f^1 - t_i^1} \implies t^1 = s^*(t_f^1 - t_i^1) + t_i^1. \tag{4.23}$$

Similar arguments yield:

$$t^2 = s^*(t_f^2 - t_i^2) + t_i^2. \tag{4.24}$$

Note that $\epsilon_{1,2}(t) = t^1 - t^2$. To find the maximum value of $|\epsilon_{1,2}(t)|$, we first calculate:

$$\begin{aligned} t^1 - t^2 &= s^* \left((t_f^1 - t_i^1) - (t_f^2 - t_i^2) \right) + (t_i^1 - t_i^2), \\ & s^* \in [0, 1]. \end{aligned} \tag{4.25}$$

Observe that the difference $t^1 - t^2$ is a function of s^*. Taking the derivative with respect to s^* yields:

$$\frac{d}{ds^*}(t^1 - t^2) = \left((t_f^1 - t_i^1) - (t_f^2 - t_i^2) \right). \tag{4.26}$$

Since this derivative is constant, by the extreme value theorem the maximum value of $t^1 - t^2$ must occur at one of the endpoints $s^* = 0$ or $s^* = 1$. This also holds for the minimum value of $t^1 - t^2$. Substituting these values of s^* into (4.25) and taking the absolute value yields:

$$|\epsilon_{1,2}(t)| \leq \max(|t_i^1 - t_i^2|, |t_f^1 - t_f^2|),$$

which completes the proof. □

Next, we consider perturbations to the control points $\alpha_1, \ldots, \alpha_\eta$. The following result shows that the normed error between two Bezier trajectories $P^1(t)$ and $P^2(t)$ with different control points can be upper bounded by the maximum normed difference between corresponding control points.

Lemma 4.4. *Consider two trajectories represented as ηth order Bezier curves defined as $P^1(s) = \sum_{j=0}^{\eta} \alpha_j^1 b_{j,\eta}(t)$ and $P^2(s) = \sum_{j=0}^{\eta} \alpha_j^2 b_{j,\eta}(t)$ under the timing law $s(\cdot)$, $s \in [0, 1]$. The maximum normed pointwise error $\|e(t)\| = \|P_1(t) - P_2(t)\|$ between the curves is bounded by*

$$\|e(t)\| \leq \max_{j \in \{1, \ldots, \eta\}} \left\| \alpha_j^1 - \alpha_j^2 \right\|. \tag{4.27}$$

Proof. The error between $P^1(s(t))$ and $P^2(s(t))$ at any $t \in [t_0, t_f]$ satisfies:

$$\begin{aligned} e(t) &= \sum_{j=1}^{\eta} (\alpha_j^1 - \alpha_j^2) b_{j,\eta}(s(t)), \\ \|e(t)\| &= \left\| \sum_{j=1}^{\eta} (\alpha_j^1 - \alpha_j^2) b_{j,\eta}(s(t)) \right\|, \\ &\leq \sum_{j=1}^{\eta} \left\| \alpha_j^1 - \alpha_j^2 \right\| |b_{j,m}(s(t))|, \\ &\leq \max_{j \in \{1,\ldots,\eta\}} \left\| \alpha_j^1 - \alpha_j^2 \right\| \sum_{j=1}^{\eta} |b_{j,\eta}(s(t))|, \\ &\leq \max_{j \in \{1,\ldots,\eta\}} \left\| \alpha_j^1 - \alpha_j^2 \right\|. \end{aligned} \tag{4.28}$$

The last inequality follows from all Bernstein basis polynomials of degree $\eta \geq 1$ being nonnegative on $s \in [0, 1]$ and forming a partition of unity; i.e. $\sum_{j=1}^{\eta} b_{j,\eta}(s) = 1$ for all $s \in [0, 1]$. □

Notably, this error bound is independent of t. Since the coefficients of a Bernstein polynomial can be represented as an $n\eta \times 1$ vector, e.g. $\begin{bmatrix} \alpha_1^T & \cdots & \alpha_\eta^T \end{bmatrix}^T$, this upper bound on the error between $P^1(t)$ and $P^2(t)$ can also be equivalently expressed as

$$\|e(t)\| \leq \left\| \begin{bmatrix} \|\alpha_1^1 - \alpha_1^2\| \\ \vdots \\ \left\|\alpha_\eta^1 - \alpha_\eta^2\right\| \end{bmatrix} \right\|_\infty.$$

4.4.3 Combined Clock and Parameter Perturbation Errors

Finally, the question remains of what error bounds can be guaranteed when both clock errors *and* parameter perturbations are present. The following lemma demonstrates that an upper bound on the norm of this error can be calculated by separately considering clock errors and parameter errors, then summing the independent error bounds.

Lemma 4.5. *Consider two trajectories represented as ηth order Bezier curves defined as $P^1(s) = \sum_{j=0}^{\eta} \alpha_j^1 b_{j,\eta}(t)$ and $P^2(s) = \sum_{j=0}^{\eta} \alpha_j^2 b_{j,\eta}(t)$ under the timing law $s(\cdot)$, $s \in [0, 1]$. Let t^{j_1} and t^{j_2} be two time sequences related by (4.8). The maximum normed pointwise error $\|e(t)\| = \|P_1(t^{j_1}) - P_2(t^{j_2})\|$ between the curves is bounded by*

$$\|e(t)\| \leq E_t(t) + E_p(t), \tag{4.29}$$

where $E_t(t), E_p(t)$ are defined as

$$E_t(t) = \sum_{j=1}^{\eta}(n-j+1)\max_k \left\|\Delta^{j-1}\alpha_{k+1} - \Delta^{j-1}\alpha_k\right\| \frac{(\delta_{\max})^j}{j!}$$
$$E_p(t) = \max_{j\in\{1,\ldots,\eta\}} \left\|\alpha_j^1 - \alpha_j^2\right\|. \tag{4.30}$$

Proof. By using the elementary properties of norms, we obtain

$$\begin{aligned}\left\|P^1(t^{j_1}) - P^2(t^{j_2})\right\| &= \left\|P^1(t^{j_1}) - P^1(t^{j_2}) + P^1(t^{j_2}) - P^2(t^{j_2})\right\| \\ &\leq \left\|P^1(t^{j_1}) - P^1(t^{j_2})\right\| + \left\|P^1(t^{j_2}) - P^2(t^{j_2})\right\|. \end{aligned} \tag{4.31}$$

Observe that the first quantity on the right hand side of (4.31), $\|P^1(t^{j_1}) - P^1(t^{j_2})\|$, corresponds to a clock synchronization error. It can therefore be upper bounded by using the results from Lemma 4.1, equation (4.9). Also observe that the second quantity on the right hand side of (4.31), $\|P^1(t^{j_2}) - P^2(t^{j_2})\|$, corresponds to Bezier parameter perturbations between the two trajectories. It can therefore be upper bounded by using the results from Lemma 4.4, equation (4.27). The result follows. □

4.5 Resilient Parameter Propagation

With expressions for the effects of clock errors and parameter perturbations on reconstructed trajectories explicitly derived, we are now prepared to discuss a method to resiliently propagate trajectory parameter vectors from the normally-behaving leaders to the normally-behaving followers under clock errors, leader parameter perturbations, and misbehaving robots.

To clearly explain the key ideas underlying the proposed method, the following three scenarios are considered in this section: synchronous propagation without parameter perturbations, asynchronous propagation without parameter perturbations, and finally asynchronous propagation with parameter perturbations.

4.5.1 Synchronous propagation without parameter perturbations

We first consider synchronous communication without parameter perturbations. It is assumed that robots have access to a synchronized, discretized clock with time steps $k \in \mathbb{Z}_+$. To propagate the parameter vector from leaders to followers, all normally-behaving robots apply the Synchronized

MS-RPA algorithm as described in Algorithm 4.1. A graphic depicting how Algorithm 4.1 operates is given in Figure 4.4.

Algorithm 4.1 SYNCHRONOUS MS-RPA WITH PARAMETER F:

1. At each time step $k \in Z_+$, each leader robot $l \in \mathcal{L}$ broadcasts v_r to its out-neighbors.
2. At each time step k, each normal robot $i \in S_f^{\mathcal{N}}$ stores the most recently received message $v_j^i(k)$ from each of its in-neighbors $j \in \mathcal{V}_i$.
3. If an robot i receives the same vector v^* from at least $F+1$ of its in-neighbors at a time k, robot i sets $\hat{v}_i = v^*$. Let k_i^0 be the first time k at which i receives the same vector from at least $F+1$ in-neighbors.
4. Each robot broadcasts $\hat{v}_i$ to its out-neighbors $j \in \mathcal{V}_i^{out}$ for all timesteps $k \geq k_i^0$.

Under an F-local adversarial model, the structure of the communication network must ensure that a subset of the faulty or adversarial robots cannot cut off the flow of information from the normally-behaving leaders to any subset of the normally-behaving followers. The notion of resilient directed acyclic graphs (RDAGs) will serve as a sufficient condition to ensure that such information bottlenecks cannot occur. Recall from Section 4.3 that $\mathcal{V}_i$ is the in-neighbor set of robot i.

Definition 4.3. *A digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ is a* resilient directed acyclic graph *(RDAG) with parameter $r \in \mathbb{Z}_+$ if there exists a partitioning of $\mathcal{V}$ into subsets $\mathcal{S}_0, \ldots, \mathcal{S}_\eta \subset \mathcal{V}$, $\eta \in \mathbb{Z}_+$ where each subset satisfies the following properties:*

- *For each $i \in \mathcal{S}_0$, $\mathcal{V}_i = \emptyset$;*
- *For each $i \in \mathcal{S}_j$, $1 \leq j \leq \eta$, $\mathcal{V}_i \subseteq \bigcup_{k=0}^{j-1} \mathcal{S}_k$;*
- *For each $i \in \mathcal{S}_1, \ldots \mathcal{S}_\eta$, $|\mathcal{V}_i| \geq r$.*

Given a digraph $\mathcal{D}$ which is an RDAG with parameter r having subsets $\mathcal{S}_j$, $0 \leq j \leq j^*$, the integer j is called the *subset index* and the integer j^* is called the *maximum subset index*.

RDAGs are a specific case of mode estimation directed acyclic graphs (MEDAGs) from [240]. The salient property of an RDAG with parameter $r = (2F+1)$ is that under the removal of any F-local set, there exists at least $F+1$ distinct directed paths from the set $\mathcal{S}_0$ to any remaining node in the graph. An example of an RDAG is depicted in Figure 4.3.

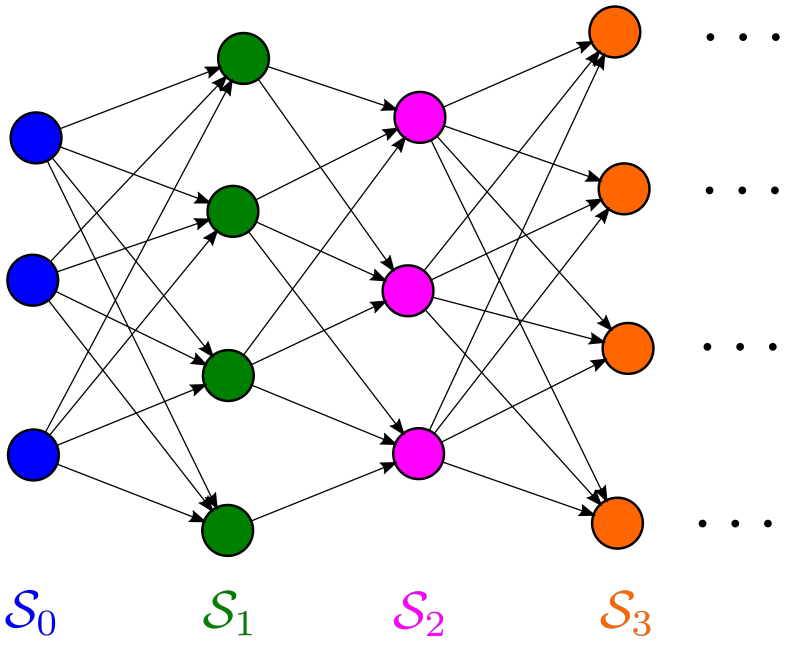


Figure 4.3: An example of an RDAG with parameter $r = 3$.

Our first theorem considers the case when all normally-behaving leaders propagate the exact same vector with synchronous communication. Although this theorem will not be the most practically applicable result in this paper, the proof will help introduce the main ideas and intuition behind how the later algorithms operate.

Theorem 4.1. *Consider a digraph $\mathcal{D}$ which is an RDAG with parameter $2F+1$ with $\mathcal{S}_0 = \mathcal{L}$ under an F-local misbehavior model. Suppose that all normally-behaving robots in the digraph apply Algorithm 4.1 with parameter F. Then under synchronous communication, all robots in $S_f^{\mathcal{N}}$ accept the vector v_r in at most ρ time steps, where $\rho \in \mathbb{Z}_+$ is the maximum subset index in the RDAG.*

Proof. Let k_0 be the first time at which all normally-behaving leaders $l \in \mathcal{L}^{\mathcal{N}}$ broadcast the vector v_r. By the definition of an RDAG, each robot i_1 in the subset $\mathcal{S}_1$ satisfies $\mathcal{V}_{i_1} \subset \mathcal{L}$ and $|\mathcal{V}_{i_1}| \geq 2F+1$. Since the misbehaving set is F-local, this implies that each normally-behaving robot $i_1 \in S_f^{\mathcal{N}} \cap \mathcal{S}_1$ has at least $F+1$ normally-behaving leaders as in-neighbors; i.e., $|\mathcal{V}_{i_1} \cap \mathcal{L}^{\mathcal{N}}| \geq F+1$. Each normal robot $i_1 \in S_f^{\mathcal{N}} \cap \mathcal{S}$ therefore accepts the vector v_r and sets $\hat{v}_{i_1} = v_r$ at timestep k_0 as per Algorithm 4.1.

At timestep k_0+1, each robot $i_1 \in \mathcal{S}_1 \cap \mathcal{N}$ broadcasts the value $\hat{v}_{i_1} = v_r$, and each normally-behaving leader $l \in \mathcal{L}^{\mathcal{N}}$ broadcasts the value v_r to its out-neighbors. The definition of an RDAG implies that each robot $i_2 \in \mathcal{S}_2$ satisfies $\mathcal{V}_{i_2} \subset \mathcal{L} \cup \mathcal{S}_1$, and $|\mathcal{V}_{i_2}| \geq 2F+1$. Since the misbehaving set is F-local, each i_2 will receive the value v_r from at least $F+1$ normally-behaving in-neighbors

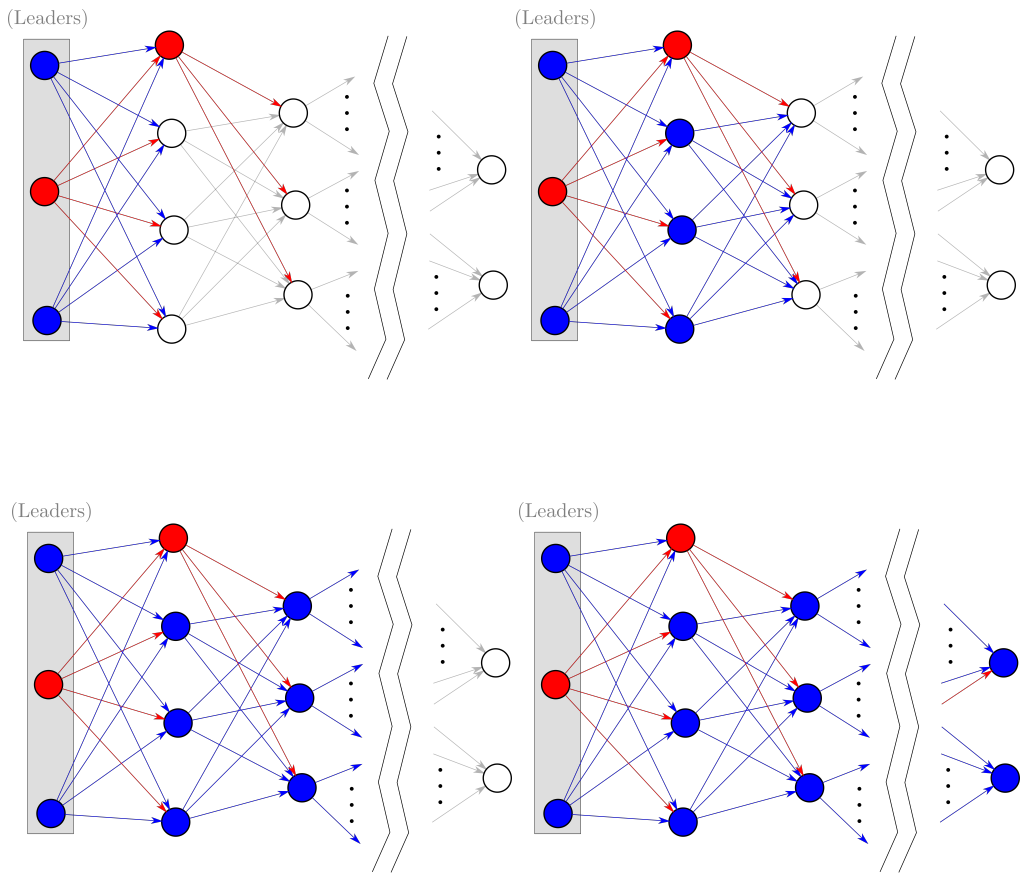


Figure 4.4: An illustration of how the MSRPA algorithm operates in a sychronous setting. The graph depicted is an RDAG with parameter 3 under an F-local adversarial model with $F = 1$. The set of leaders is indicated by the circles in the box on the left, adversarial agents are indicated by the color red, and agents possessing the reference vector of parameters are indicated by the color blue. Leaders begin by broadcasting the reference vector to their out-neighbors. At each time step, any normal follower which receives the same vector message from at least $F + 1$ in-neighbors accepts the vector message and begins rebroadcasting it to its out-neighbors at the next time step.

at time k_0+1. By the synchronous MS-RPA algorithm, each normally-behaving robot $i_2 \in \mathcal{S}_2 \cap S_f^{\mathcal{N}}$ will accept the vector v_r and set $\hat{v}_{i_2} = v_r$ at time step $k_0 + 1$.

Continuing on in this manner it can be shown that given any $p \leq \rho$ (where ρ is the maximum subset index in the RDAG), at timestep $k_0 + (p-1)$ all normal leaders $\mathcal{L}^{\mathcal{N}}$ broadcast v_r and all normal robots j in the set $(\bigcup_{q=1}^{p-1} \mathcal{S}_q) \cap \mathcal{N}$ broadcast $\hat{v}_j = v_r$ to their out-neighbors. In addition, all robots $i_p \in \mathcal{S}_p$ satisfy $\mathcal{V}_{i_p} \subset \mathcal{L} \cup \bigcup_{q=1}^{p-1} \mathcal{S}_q$ and $|\mathcal{V}_{i_p}| \geq 2F + 1$. Since the misbehaving set is F-local, each i_p will receive the value v_r from at least $F + 1$ normally-behaving in-neighbors at time $k_0 + (p-1)$. By the synchronous MS-RPA algorithm, each normally-behaving robot $i_p \in \mathcal{S}_p \cap S_f^{\mathcal{N}}$ will accept the vector v_r and set $\hat{v}_{i_p} = v_r$ at time step $k_0 + (p-1)$.

Since the RDAG is comprised of subsets $\mathcal{S}_0, \ldots, \mathcal{S}_\rho$, all normally-behaving robots will therefore have accepted the vector v_r by timestep $k_0 + (\rho - 1)$, which concludes the proof. □

4.5.2 Propagation with Time-Varying Graphs

In practical conditions, communication between robots is commonly asynchronous; i.e., robot transmission of information does not occur in sync with a common discretized clock. In addition, the graph-theoretic structure of the underlying communication network may vary over time. We model the effects of a time-varying graph and asynchronous communications as a time-varying edge set for the network communication graph. The notation $\mathcal{D}^T(t) = \bigcup_{\tau \in [t-T,t]} \mathcal{D}[\tau]$ denotes the union of the digraph across the time interval $[t - T, t]$ for $T \in \mathbb{R}_+$. We likewise define the following quantities:

$$\mathcal{V}^T(t) = \bigcup_{\tau \in [t-T,t]} \mathcal{V}(\tau)$$
$$\mathcal{V}_i^T(t) = \bigcup_{\tau \in [t-T,t]} \mathcal{V}_i(\tau)$$
$$\mathcal{E}^T(t) = \bigcup_{\tau \in [t-T,t]} \mathcal{E}(\tau)$$

This method of considering the graph structure over a sliding time window will enable the analysis of how quickly the normal leaders' message will propagate to all normal followers in the network.

Algorithm 4.2 extends Algorithm 4.1 to handle asynchronous and time-varying communication within the network. Algorithm 4.2 uses the following definition to denote the most recent time at which an robot i receives a vector message from another robot j.

Definition 4.4. *The time $\tau_j^i(t)$, $t \in \mathbb{R}$, is defined as follows:*

$$\tau_j^i(t) = \underset{t' \in \{t_0, \ldots, t\}}{\arg\max} \{t' : j \in \mathcal{V}_i(t')\} \tag{4.32}$$

Algorithm 4.2 ASYNCHRONOUS MS-RPA WITH PARAMETER F:

1. At each time instance $t \geq t_0$, each leader robot $l \in \mathcal{L}$ broadcasts v_r to its out-neighbors.
2. Each normal robot $i \in S_f^{\mathcal{N}}$ stores the most recently received message $v_j^i(\tau_j^i(t))$ from each of its in-neighbors $j \in \mathcal{V}_i^T(t)$.
3. If at or before time t an robot i has received the same vector v^* from at least $F+1$ of its in-neighbors, robot i sets $\hat{v}_i = v^*$. Let t_i^0 be the first time t at which i has received the same vector from at least $F+1$ in-neighbors.
4. Each robot broadcasts $\hat{v}_i$ to its out-neighbors $j \in \mathcal{V}_i^{out}$ for all time instances $t \geq t_i^0$.

To summarize Algorithm 4.2 in words, each robot stores the most recently received vector values from its in-neighbors over the time window T. If an robot i has received the same vector from at least $F+1$ in-neighbors, it accepts this vector and rebroadcasts it to its out-neighbors. Intuitively speaking, under an F-local model this will ensure that robot i cannot accept any vector other than the vector propagated by the normally-behaving leaders.

We now give a formal proof of conditions under which normally-behaving leaders are able to propagate a vector message to followers using Algorithm 4.2.

Theorem 4.2. *Consider a digraph $\mathcal{D}$ with $\mathcal{S}_0 = \mathcal{L}$ under an F-local misbehavior model. Let $\mathcal{D}^*$ be an RDAG with parameter $2F+1$. Suppose that all normally-behaving robots in the digraph apply Algorithm 4.2 with parameter F. If there exists a $T > 0$ such that $\mathcal{D}^T(t) = \mathcal{D}^*$ with $\mathcal{S}_0 = \mathcal{L}$ for all $t \geq t_0 + T$, then all robots in $S_f^{\mathcal{N}}$ will accept the vector v_r by time $t = t_0 + \rho T$, where ρ is the maximum subset index in the RDAG.*

Proof. Consider time $t = t_0 + T$. Since $\mathcal{D}^T(t) = \mathcal{D}^*$ is an RDAG with parameter $2F+1$ for all $t \geq t_0 + T$, it follows that for all $i_1 \in \mathcal{S}_1$, we have $\mathcal{V}_{i_1}^T \subseteq \mathcal{S}_0 = \mathcal{L}$ and $|\mathcal{V}_{i_1}^T| \geq 2F+1$. Since the misbehaving set is F-local, this implies that $|\mathcal{V}_{i_1}^T(t) \cap \mathcal{L}^{\mathcal{N}}| \geq F+1$; i.e., each i_1 received the vector v_r from at least $F+1$ normally-behaving leaders in the interval $[t_0, t_0+T]$. As per Algorithm 4.2, each normal follower $i_1 \in S_f^{\mathcal{N}} \cap \mathcal{S}_1$ therefore accepts the vector v_r, sets $\hat{v}_{i_1} = v_r$, and begins broadcasting $\hat{v}_{i_1}$ to its out-neighbors no later than time $t = t_0 + T$.

Next, consider time $t = t_0 + 2T$. Since $\mathcal{D}^T(t) = \mathcal{D}^*$ is an RDAG with parameter $2F+1$ each robot i_2 satisfies $\mathcal{V}_{i_2}(t) \subseteq \mathcal{L} \cup \mathcal{S}_1$ and $|\mathcal{V}_{i_2}(t)| \geq 2F+1$. Since the misbehaving set $\mathcal{A}$ is F-local,

this implies that $|\mathcal{V}_{i_2}^T(t) \cap \mathcal{N}| \geq F+1$. robot i_2 therefore receives the vector v_r from at least $F+1$ robots, sets $\hat{v}_{i_2} = v_r$, and broadcasts $\hat{v}_{i_2}$ to its out-neighbors no later than time $t = t_0 + 2F$.

To continue inductively, assume that at any time $t = t_0 + (p-1)T$, $2 \leq p \leq \rho$, each robot $i_{p-1} \in \mathcal{L} \cup (\bigcup_{k=1}^{p-1} \mathcal{S}_k^{\mathcal{N}})$ satisfies $\hat{v}_{i_{p-1}} = v_r$ and is broadcasting $\hat{v}_{i_{p-1}}$ to all of its out-neighbors. Using prior arguments, at time $t = t_0 + pT$, $2 \leq p \leq \rho + 1$ (where ρ is the maximum subset index in the RDAG), each robot $i_p \in \mathcal{S}_p$ satisfies $\mathcal{V}_{i_p}^T \subseteq (\mathcal{L} \cup \bigcup_{j=1}^{p-1} \mathcal{S}_j)$ and $|\mathcal{V}_{i_{p_1}}| \geq 2F+1$. Since the misbehaving set is F-local, this implies that $|\mathcal{V}_{i_{p-1}} \cap \mathcal{N}| \geq F+1$, and therefore each i_p receives the vector v_r from at least $F+1$ normally-behaving robots in the interval $[t_0 + (p-1)T, t_0 + pT]$. Each i_p therefore receives the vector v_r from at least $F+1$ robots, sets $\hat{v}_{i_p} = v_r$, and broadcasts $\hat{v}_{i_p}$ to its out-neighbors no later than time $t = t_0 + pF$.

Since the RDAG consists of sublevel sets $\mathcal{S}_0, \ldots, \mathcal{S}_\rho$, it therefore holds that by time $t = t_0 + \rho T$, all normal robots within the network have received the vector message v_r, which concludes the proof. □

4.5.3 Incorporating Parameter Perturbations

To reiterate from Section 4.3, due to the nature of real-world conditions, the vector messages from the leaders v_l, $l \in \mathcal{L}$ may not be precisely equal due to, for instance, noise or numerical errors. Given ϵ_{lp}, i.e., the maximum normed error between leaders' parameter vectors defined in (4.5), we seek to ensure that the maximum normed error between *any* two normal robots' states, both leaders and followers, is upper bounded by ϵ_{lp}. In other words, our objective is for

$$\max_{i,j \in \mathcal{N}} \|v_i - v_j\|_\infty \leq \epsilon_{lp}. \tag{4.33}$$

To accomplish this, we consider the hyperrectangle formed by the *elementwise convex hull* of the entries of a set of vectors. Let $v_{j,k}$ denote the kth entry of vector v_j. The elementwise convex hull is defined as follows:

$$\begin{aligned} &\text{co}_\text{e}\{v_1, \ldots, v_q\} \triangleq \begin{bmatrix} \text{co}\{\min_j v_{j,1}, \max_j v_{j,1}\} \\ \vdots \\ \text{co}\{\min_j v_{j,m}, \max_j v_{j,m}\} \end{bmatrix}, \\ &v_1, \ldots, v_q \in \mathbb{R}^m. \end{aligned} \tag{4.34}$$

From (4.5) it is straightforward to verify that any two vectors $v_i, v_j \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}|}}\}$, $v_{l_1}, \ldots, v_{l_{|\mathcal{L}|}} \in \mathcal{L}$ satisfy $\|v_i - v_j\|_\infty \leq \epsilon_{lp}$. Therefore our objective will be to ensure that each normal robot i selects a parameter vector v_i such that $v_i \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}|}}\}$.

To accomplish this objective, robots will augment an elementwise median approach to the techniques presented previously. Given vectors $v_1, \ldots, v_q \in \mathbb{R}^m$, we denote the elementwise median function as

$$\text{MEDIAN}\{v_1, \ldots, v_q\} \triangleq \begin{bmatrix} \text{median}\{v_{1,1}, \ldots, v_{q,1}\} \\ \vdots \\ \text{median}\{v_{1,m}, \ldots, v_{q,m}\} \end{bmatrix} \tag{4.35}$$

This elementwise median function is used by Algorithm 4.3 for the asynchronous case with parameter perturbations.

Algorithm 4.3 ASYNCHRONOUS PERTURBED MS-RPA WITH PARAMETER F:

- At each time instance $t \geq t_0$, each leader robot $l \in \mathcal{L}$ broadcasts v_r to its out-neighbors.
- Each normal robot $i \in S_f^{\mathcal{N}}$ stores the most recently received message $v_j^i(\tau_j^i(t))$ from each of its in-neighbors $j \in \mathcal{V}_i^T(t)$ in the set $\Gamma_i(t) = \{v_j^i(\tau_{j_1}^i(t)), \ldots\}$.
- Let t_{Γ_i} be the first time such that $|\Gamma_i(t)| \geq 2F+1$. At time t_{Γ_i} robot i sets $\hat{v}_i = \text{MEDIAN}\{\Gamma_i(t)\}$ and broadcasts $\hat{v}_i$ to all of its out-neighbors.
- At each subsequent time $\tau_j^i(t)$ such that $t \geq t_{\Gamma_i}$, $j \in \mathcal{V}_i$, robot i updates $\hat{v}_i = \text{MEDIAN}\{\Gamma_i(t)\}$ and broadcasts $\hat{v}_i$ to all of its out-neighbors.

In words, similar to Algorithm 4.2 each robot i stores the most recent vector message it has received from each of its in-neighbors. The set $\Gamma_i(t)$ is this set of stored vectors. Unlike Algorithm 4.2 however, each robot i does nothing until it has received at least $2F+1$ messages from its in-neighbors; i.e. $|\Gamma_i(t)| \geq 2F+1$. Once this has occurred, robot i takes the elementwise median of the vectors in $\Gamma_i(t)$ and accepts the resulting vector as $\hat{v}_i$. This vector is rebroadcasted to its out-neighbors. If additional vectors are received, robot i stores them in the set $\Gamma_i(t)$, recalculates the elementwise median, updates $\hat{v}_i$, and broadcasts this new value.

The reason that each robot waits until $|\Gamma_i(t)| \geq 2F+1$ to take the elementwise median and accept a vector value is demonstrated by the following Lemma:

Lemma 4.6. *Let* $S_1 = \{v_1^1, \ldots, v_{q_1}^1\}$, $|S_1| = q_1$, *and* $S_2 = \{v_1^2, \ldots, v_{q_2}^2\}$, $|S_2| = q_2$, *with* $v_i^1, v_j^2 \in \mathbb{R}^m \; \forall i, j$. *If* $q_1 > q_2$, *then the following holds:*

$$\text{MEDIAN}(S_1 \cup S_2) \in co_e\{S_1\} \tag{4.36}$$

Proof. Follows from the definition of MEDIAN$(\cdot)$ and $\text{co}_\text{e}\{\cdot\}$. □

More explicitly, when taking the elementwise median of two sets of vectors S_1, S_2 with $|S_1| > |S_2|$, the resulting vector will lie within the elementwise convex hull of S_1. Given an F-total or F-local adversarial model, each robot i waiting until it has received at least $2F+1$ vectors ensures that $|\mathcal{V}_i(t) \cap \mathcal{N}| > |\mathcal{V}_i(t) \cap \mathcal{A}|$; i.e. the set $\Gamma_i(t)$ contains more vectors from normal in-neighbors than adversarial in-neighbors.

We now present conditions under which Algorithm 4.3 guarantees that all normally-behaving follower robots are able to accept a vector within the elementwise convex hull of the normally-behaving leader robots' vector messages in the presence of perturbations and an F-local adversarial model.

Theorem 4.3. *Consider a digraph $\mathcal{D}$ with $\mathcal{S}_0 = \mathcal{L}$ under an F-local misbehavior model. Let $\mathcal{D}^*$ be an RDAG with parameter $2F+1$. Suppose all normally-behaving robots in the digraph apply Algorithm 4.3 with parameter F. Suppose further that the misbehaving set $\mathcal{A}$ is F-local, and that each normally-behaving leader $l_j \in \mathcal{L}^{\mathcal{N}}$, $j \in 1, \ldots, |\mathcal{L}^{\mathcal{N}}|$ broadcasts the vector $v_{l_j} \in \mathbb{R}^m$. If there exists a $T > 0$ such that $\mathcal{D}^T(t) = \mathcal{D}^*$ with $\mathcal{S}_0 = \mathcal{L}$ for all $t \geq t_0 + T$, all robots in $S_f^{\mathcal{N}}$ will accept a vector $\hat{v}_i \in co_e\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$ by time $t = t_0 + (\rho+1)T$, where ρ is the maximum subset index in the RDAG. Furthermore, for all $i \in S_f^{\mathcal{N}}$ it holds that $\hat{v}_i \in co_e\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$ for all $t \geq t_{\Gamma_i}$.*[4]

Proof. Consider time $t = t_0 + T$. Since $\mathcal{D}^T(t) = \mathcal{D}^*$ is an RDAG with parameter $2F+1$ for all $t \geq t_0 + T$, it follows that for all $i_1 \in \mathcal{S}_1$ we have $\mathcal{V}_{i_1}^T(t) \subseteq \mathcal{S}_0 = \mathcal{L}$ and $|\mathcal{V}_{i_1}^T(t)| \geq 2F+1$. This implies that $t_{\Gamma_{i_1}} \leq t_0 + T$; i.e., each i_1 will have received at least $2F+1$ messages from its in-neighbors at or before $t = t_0 + T$. Denote the following:

$$\Gamma_i^{\mathcal{N}}(t) = \{j \in \mathcal{N} : v_j^i(\tau_j^i(t)) \in \Gamma_i(t)\}, \tag{4.37}$$

$$\Gamma_i^{\mathcal{A}}(t) = \{j \in \mathcal{A} : v_j^i(\tau_j^i(t)) \in \Gamma_i(t)\}. \tag{4.38}$$

The misbehaving set $\mathcal{A}$ being F-local implies that $|\Gamma_{i_1}^{\mathcal{A}}(t)| \leq F$ for all $t \geq 0$. Since $|\mathcal{V}_{i_1}^T(t)| \geq 2F+1$ by definition of an RDAG, we have $|\Gamma_{i_1}^{\mathcal{N}}(t)| \geq F+1$. By Algorithm 4.3, each robot i_1 sets $v_{i_1} = \text{MEDIAN}(\Gamma_i(t))$ at time $t_{\Gamma_{i_1}}$. Observe that $\Gamma_{i_1}(t) = \Gamma_{i_1}^{\mathcal{N}}(t) \cup \Gamma_{i_1}^{\mathcal{A}}(t)$. Since at $t_{\Gamma_{i_1}}$ we have $|\Gamma_{i_1}(t_{\Gamma_{i_1}})| \geq 2F+1$ and $|\mathcal{A}| \leq F$, it holds that $|\Gamma_{i_1}^{\mathcal{N}}(t_{\Gamma_{i_1}})| \geq |\Gamma_{i_1}^{\mathcal{A}}(t_{\Gamma_{i_1}})|$. It follows from Lemma 4.6 that $\text{MEDIAN}(\Gamma_{i_1}(t_{\Gamma_{i_1}})) \in \text{co}_\text{e}\{\Gamma_{i_1}^{\mathcal{N}}(t_{\Gamma_{i_1}})\}$ for $t = t_{\Gamma_i}$. But since $\mathcal{V}_i^T(t) \subseteq \mathcal{L}$, this implies that $\text{co}_\text{e}\{\Gamma_{i_1}^{\mathcal{N}}(t_{\Gamma_{i_1}})\} \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$, which implies that $v_{i_1}(t_{\Gamma_{i_1}}) \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$. By Algorithm 4.3, each $i_1 \in \mathcal{S} \cap \mathcal{N}$ accepts this v_{i_1} and begins broadcasting v_{i_1} to its out-neighbors no later than time $t_{\Gamma_{i_1}} \leq t_0 + T$. Observe that since $\mathcal{A}$ is F-local and $\mathcal{D}^T(t) = \mathcal{D}^*$ is an RDAG with parameter $2F+1$, it will always hold that $|\Gamma_{i_1}^{\mathcal{N}}(t)| > |\Gamma_{i_1}^{\mathcal{A}}(t)|$ for all $t \geq t_{\Gamma_{i_1}}$. Since $t_{\Gamma_{i_1}} \leq t_0 + T$, by Lemma 4.6 and the preceding logic $v_{i_1}(t) \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$ for all $t \geq t_0 + T \; \forall i_1 \in \mathcal{S}_1$.

[4] Recall that t_{Γ_i} is defined in Algorithm 4.3.

To continue inductively, assume that at any time $t_0 + (q-1)T$, $2 \leq q \leq \rho$, each robot $i_{q-1} \in \mathcal{L} \cup (\bigcup_{k=1}^{q-1} \mathcal{S}_k^{\mathcal{N}})$ satisfies $v_{i_{q-1}} \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$ and is broadcasting $v_{i_{q-1}}$ to all of its out-neighbors. Using prior arguments, at time $t = t_0 + qT$ consider any robot $i_q \in \mathcal{S}_q$. Since $\mathcal{D}^T(t) = \mathcal{D}^*$ is an RDAG with parameter $2F+1$, $\mathcal{V}_{i_q}(t) \subseteq \mathcal{L} \cup (\bigcup_{k=1}^{q-1} \mathcal{S}_k)$ and $|\mathcal{V}_{i_q}^T(t)| \geq 2F+1$. This implies that $t_{\Gamma_{i_q}} \leq t_0 + qT$. Since the misbehaving set $\mathcal{A}$ is F-local, $|\Gamma_{i_q}^{\mathcal{A}}(t)| \leq F$ for all $t \geq t_0$ which further implies that $|\Gamma_{i_q}^{\mathcal{N}}(t_{\Gamma_{i_q}})| \geq F+1$ and therefore $|\Gamma_{i_q}^{\mathcal{N}}(t_{\Gamma_{i_q}})| > |\Gamma_{i_q}^{\mathcal{A}}(t_{\Gamma_{i_q}})|$. By Algorithm 4.3, at time $t_{\Gamma_{i_q}}$ robot i_q sets $v_{i_q} = \text{MEDIAN}(\Gamma_i(t_{\Gamma_{i_q}}))$. Since $|\Gamma_{i_q}^{\mathcal{N}}(t_{\Gamma_{i_q}})| > |\Gamma_{i_q}^{\mathcal{A}}(t_{\Gamma_{i_q}})|$ it therefore follows from Lemma 4.6 that $\text{MEDIAN}(\Gamma_{i_q}(t_{\Gamma_{i_q}})) \in \text{co}_\text{e}\{\Gamma_{i_q}^{\mathcal{N}}(t_{\Gamma_{i_q}})\}$. Since $\mathcal{V}_i^T(t) \subseteq \mathcal{L} \cup (\bigcup_{k=1}^{q-1} \mathcal{S}_k$ and each normal robot $j \in \mathcal{L} \cup (\bigcup_{k=1}^{q-1}$ will be broadcasting a a vector $v_j(t) \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$ for all $t \geq t_0 + (q-1)T$, it follows that $\text{co}_\text{e}\{\Gamma_{i_q}^{\mathcal{N}}(t_{\Gamma_{i_q}})\} \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$ and therefore $\text{MEDIAN}(\Gamma_{i_q}(t_{\Gamma_{i_q}})) \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$ for all $i_q \in \mathcal{S}_q$. Furthermore, since $\mathcal{A}$ is F-local, for all $t \geq t_{\Gamma_{i_q}}$ we have that $|\Gamma_{i_q}^{\mathcal{A}}(t)| < |\Gamma_{i_q}^{\mathcal{N}}(t)|$, implying that $v_{i_q}(t) \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$ for all $t \geq t_{\Gamma_{i_q}}$ $\forall i_q \in \mathcal{S}_q$. Since $t_{\Gamma_{i_q}} \leq t_0 + qT$ we therefore have $v_{i_q}(t) \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$ for all $t \geq t_0 + qT$ $\forall i_q \in \mathcal{S}_q$. The result follows by noting that $q = \rho$ is the maximum subset index in the RDAG, implying that for all $t \geq t_0 + \rho T$ we therefore have $v_i(t) \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$ for all $i \in \mathcal{N}$. $\square$

Theorem 4.3 guarantees that within finite time each normal follower i will accept a vector v_i within the elementwise convex hull of the leader vector messages; i.e. $v_i \in \text{co}_\text{e}\{v_{l_1}, \cdots, v_{l_{|\mathcal{L}^{\mathcal{N}}|}}\}$. As per the statement of Problem 4.1, the final result of this chapter proves that under Algorithm 4.3 the maximum pointwise error between any normal robots' trajectories is bounded by the sum of class-$\mathcal{K}$ functions of the maximum clock error $\epsilon_{t,\max}(t)$ and the maximum parameter perturbation error ϵ_{lp}.

Theorem 4.4. *Let $\epsilon_{t,\text{sup}}$ be defined as in (4.6), let ϵ_{lp} be defined as in (4.5), and let $e_{\max}(t) = \max_{i,j \in \mathcal{N}} P_i(t) - P_j(t)$ be the maximum normed pointwise trajectory error between any two normal robots. Then under the conditions of Theorem 4.3, the following holds:*

$$\|e_{\max}(t)\| \leq \alpha \left(\left\| \begin{bmatrix} \epsilon_{lp} \\ \epsilon_{t,\text{sup}} \end{bmatrix} \right\| \right), \tag{4.39}$$

where α is a class-$\mathcal{K}$ function.

Proof. Consider any two arbitrary normal robots $i, j \in \mathcal{N}$ and recall the definition of the error $e_{ij}(t)$ from (4.4). By Lemma 4.5, the normed error $\|e_{ij}(t)\|$ can be upper bounded by the summation of the parameter error and the clock synchronization error.

We first consider the parameter error. By Theorem 4.3 and Lemma 4.6, each normal follower robot $i \in S_f^{\mathcal{N}}$ will accept a vector message within the elementwise convex hull of the leader vectors, i.e. $v_i \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^{\mathcal{N}}}}\}$. By the definition of ϵ_{lp} in (4.5), this implies that

$\max_{i,j\in\mathcal{N}} \|v_i - v_j\|_\infty \leq \epsilon_{lp}$. By Lemma 4.4 we therefore have $\|e_{ij}(t)\| \leq \epsilon_{lp}$ for all $i, j \in \mathcal{N}$, where ϵ_{lp} is clearly a class-$\mathcal{K}$ function in ϵ_{lp}.

The clock synchronization error is considered next. The nominal supremum time error is $\epsilon_{t,\sup}$ as per (4.6). However, as per Lemma 4.3 and 4.2 the effects of the differences between any two agents' t_i and t_f variables is also treated as an additional source of clock synchronization error satisfying $|\epsilon_{j_1,j_2}(t)| \leq \max(|t_i^{j_1} - t_i^{j_2}|, |t_f^{j_1} - t_f^{j_2}|)$, $j_1, j_2 \in \mathcal{N}$. Since $v_{j_1}, v_{j_2} \in \text{co}_\text{e}\{v_{l_1}, \ldots, v_{l_{|\mathcal{L}^\mathcal{N}}}$ for all $j_1, j_2 \in \mathcal{N}$, we therefore have $\max(|t_i^{j_1} - t_i^{j_2}|, |t_f^{j_1} - t_f^{j_2}|) \leq \epsilon_{lp}$ which implies that $|\epsilon_{j_1,j_2}(t)| \leq \epsilon_{lp}$. The total clock synchronization error is therefore upper bounded by $\epsilon_{t,\sup} + \epsilon_{lp}$. Observe that since $\epsilon_{t,sup} > 0$ and $\epsilon_{lp} > 0$ by definition, it therefore holds that $\epsilon_{lp} + \epsilon_{t,\sup} = |\epsilon_{lp}| + |\epsilon_{t,\sup}| = \left\| \begin{bmatrix} \epsilon_{lp} & \epsilon_{t,sup} \end{bmatrix}^T \right\|_1$. This total clock synchronization error holds for all normal agents.

By Corollary 4.1, given any specific robot $j \in \mathcal{N}$ the upper bound on the timing error for the reconstructed trajectory $P^j(t)$ under the total clock synchronization error is

$$\sum_{j=1}^{\eta} (\eta - j + 1) \max_k \left\| \Delta^{j-1}\alpha_{k+1}^j - \Delta^{j-1}\alpha_k^j \right\| \frac{\left(\delta_{\max} \left(\left\| \begin{bmatrix} \epsilon_{lp} \\ \epsilon_{t,\sup} \end{bmatrix} \right\|_1 \right) \right)^j}{j!}. \tag{4.40}$$

By Corollary 4.1 this is a class-$\mathcal{K}$ function in $\left\| \begin{bmatrix} \epsilon_{lp} & \epsilon_{lp} \end{bmatrix}^T \right\|_1$. For brevity we denote the quantity in (4.40) as $E_t^j \left(\left\| \begin{bmatrix} \epsilon_{lp} & \epsilon_{lp} \end{bmatrix}^T \right\|_1 \right)$. To obtain an upper bound which holds for *all* normal robots $j \in \mathcal{N}$, by Lemma 4.2 we have

$$E_t^i \left(\left\| \begin{bmatrix} \epsilon_{t,sup} \\ \epsilon_{lp} \end{bmatrix} \right\|_1 \right) \leq E_t^j \left(\left\| \begin{bmatrix} \epsilon_{t,sup} \\ \epsilon_{lp} \end{bmatrix} \right\|_1 \right) + 2\epsilon_{lp}, \quad \forall i, j \in \mathcal{N}.$$

Combining this with the parameter perturbation error upper bound ϵ_{lp}, we obtain

$$\|e_{ij}(t)\| \leq 3\epsilon_{lp} + E_t^j \left(\left\| \begin{bmatrix} \epsilon_{lp} \\ \epsilon_{t,sup} \end{bmatrix} \right\|_1 \right) \leq 3 \left\| \begin{bmatrix} \epsilon_{lp} \\ \epsilon_{t,\sup} \end{bmatrix} \right\|_1 + E_t^j \left(\left\| \begin{bmatrix} \epsilon_{lp} \\ \epsilon_{t,sup} \end{bmatrix} \right\|_1 \right), \quad \forall i, j \in \mathcal{N}. \tag{4.41}$$

The result follows by recalling the equivalence of norms and noting that the sum of any class-$\mathcal{K}$ functions in $\left\| \begin{bmatrix} \epsilon_{lp} & \epsilon_{t,sup} \end{bmatrix} \right\|_1^T$ is also a class-$\mathcal{K}$ function in $\left\| \begin{bmatrix} \epsilon_{lp} & \epsilon_{t,sup} \end{bmatrix} \right\|_1^T$. □

4.6 Simulations

The simulations consider a network of $N = 12$ unicycle robots in the plane. Each robot $i \in \mathcal{V}$ has the state $p_i = \begin{bmatrix} x_i & y_i & \theta_i \end{bmatrix}$ with dynamics defined as

$$\dot{p}_i = \begin{bmatrix} \cos(\theta_i) & 0 \\ \sin(\theta_i) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}. \tag{4.42}$$

Each robot's control inputs are bounded as follows:

$$\begin{aligned} |v_i| &\leq v_{\max} = 1 \\ |\omega_i| &\leq \omega_{\max} = 2 \end{aligned} \tag{4.43}$$

robots are controlled using the following input-output linearization method [234, 241]: Given a scalar $b > 0$, the outputs are defined as

$$z_{i,1} = x_i + b\cos(\theta_i), \tag{4.44a}$$
$$z_{i,2} = y_i + b\sin(\theta_i). \tag{4.44b}$$

The point $\begin{bmatrix} z_{i,1} & z_{i,2} \end{bmatrix}^T \in \mathbb{R}^2$ defines a point at a distance of b along the forward axis of the robot from the robot's position $\begin{bmatrix} x_i & y_i \end{bmatrix}^T$. Treating the dynamics of $z_{i,1}$ and $z_{i,2}$ as single integrators yields the following dynamics:

$$\dot{z}_{i,1} = u_{i,1}, \tag{4.45a}$$
$$\dot{z}_{i,2} = u_{i,2}, \tag{4.45b}$$
$$\dot{\theta} = \frac{u_{i,2}\cos(\theta) - u_{i,1}\sin(\theta)}{b}. \tag{4.45c}$$

Using these dynamics, the transformation between the linearized control inputs $\begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$ and the original control inputs $\begin{bmatrix} v_i & \omega_i \end{bmatrix}^T$ can be shown to be as follows:

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i/b) & \cos(\theta_i/b) \end{bmatrix} \begin{bmatrix} u_{i,1} \\ u_{i,2} \end{bmatrix}. \tag{4.46}$$

Control of each robot is accomplished by designing the linearized control inputs $u_{i,1}, u_{i,2}$. These resulting control commands are then transformed into the actual system commands $\begin{bmatrix} v & \omega \end{bmatrix}^T$ via

(4.46). More specifically, given a desired time-varying reference point $p_i^r(t) \in \mathbb{R}^2$, the following control law is employed for robot i [234, Ch. 11]:

$$\begin{aligned} u_{i,1} &= \dot{p}_{i,1}^r + k_1(p_{i,1}^r - p_{i,1}), \\ u_{i,2} &= \dot{p}_{i,2}^r + k_2(p_{i,2}^r - p_{i,2}). \end{aligned} \tag{4.47}$$

To incorporate collision avoidance between robots and other external obstacles, the nominal control law (4.47) is minimally modified using control barrier function quadratic programming techniques. More details on control barrier function techniques can be found in Chapter 5 of this dissertation and [160]. The convex optimization solver used in these simulations is the Operator Splitting Quadratic Program solver [242]. Given $\hat{u}(\cdot) = \begin{bmatrix} u_{i,1} & u_{i,2} \end{bmatrix}^T$, quadratic programming (QP) methods can be used to compute a new control input $u(\cdot) \in \mathbb{R}^2$ which minimally modifies $\hat{u}(\cdot)$ (in the sense of a desired norm $\|\cdot\|$) while ensuring forward invariance of a safe set S. When the control input is constrained to lie within a convex polytope $A_u u \leq b_u$, this minimally modified controller can be computed by the following QP [160]:

$$\begin{aligned} \begin{bmatrix} u_{i,1} \\ u_{i,2} \end{bmatrix} = \underset{u \in \mathbb{R}^2}{\arg\min} \quad & \|u - \hat{u}(x)\|_2^2 \\ \text{subject to} \quad & \frac{\partial h_j(x)}{\partial x} u \geq -\alpha(h_j(x)) \\ & A_u u \leq b_u, \end{aligned} \tag{4.48}$$

where $A_u \in \mathbb{R}^{q \times m}$, $b_u \in \mathbb{R}^q$. Note that the formulation in the first constraint of (4.48) is with respect to the dynamics of the linearized outputs $z_{i,1}, z_{i_2}$ and not the original system. Each agent solves a separate QP in the form (4.48) locally to compute its own minimally modified control inputs.

The control input constraints in (4.43) are with respect to v_i, ω_i. The equivalent constraints for the linearized inputs $\begin{bmatrix} u_{i,1} & u_{i,2} \end{bmatrix}$ can be derived as follows: (4.43) can be expressed in matrix form as

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \preceq \begin{bmatrix} v_{\max} \\ v_{\max} \\ \omega_{\max} \\ \omega_{\max} \end{bmatrix}. \tag{4.49}$$

However, equivalent input constraints on $u_{i,1}, u_{i,2}$ can be derived by substituting in the right

hand side of (4.46) into (4.49):

$$\begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\cos(\theta_i) & -\sin(\theta_i) \\ -\sin(\theta_i)/b & \cos(\theta_i)/b \\ \sin(\theta_i)/b & -\cos(\theta_i)/b \end{bmatrix} \begin{bmatrix} u_{i,1} \\ u_{i,2} \end{bmatrix} \preceq \begin{bmatrix} v_{\max} \\ v_{\max} \\ \omega_{\max} \\ \omega_{\max} \end{bmatrix},$$

$$A_u \begin{bmatrix} u_{i,1} \\ u_{i,2} \end{bmatrix} \preceq b_u.$$

To incorporate collision avoidance between the robots, each robot is given the safety radius $R > 0$. The functions $h_{ij}(\cdot)$ are defined as follows:

$$h_{ij}(z_i, z_j) = \left\| \begin{bmatrix} z_{i,1} \\ z_{i,2} \end{bmatrix} - \begin{bmatrix} z_{j,1} \\ z_{j,2} \end{bmatrix} \right\|_2^2 - (2R + b)^2, \tag{4.50}$$

where z_i, z_j are the outputs of the robots. Intuitively, $h(p_i, p_j) \geq 0$ implies that $\left\| \left[\begin{smallmatrix} x_i \\ y_i \end{smallmatrix} \right] - \left[\begin{smallmatrix} x_j \\ y_j \end{smallmatrix} \right] \right\| \geq 2(R + b)$, which implies that the robots are at least twice the safety radius from each other. The term $2R + b$ reflects that collision avoidance for each robot i is being performed with respect to the outputs $\begin{bmatrix} z_{i,1} & z_{i,2} \end{bmatrix}$ which is offset from the actual state $\begin{bmatrix} x_i & y_i \end{bmatrix}$ by the distance b.

4.6.1 Incorporating Formational Offsets

Formations are specified as follows: the formation is defined in terms of a time-varying formation reference point $p_f^r : \mathbb{R} \to \mathbb{R}^2$, a set of offset vectors $\xi_i \in \mathbb{R}^2$, $i \in \mathcal{V}$, and a formation frame of reference $\mathcal{F}_f$. Each robot i's time-varying reference point is defined as $p_i^r(t) = p_f^r(t) + \xi_i$ with respect to the formation frame.

The formation reference point $p_f^r(t)$ is expressed as a C^1 Bezier-curve-based trajectory with parameter vector v and linear timing law $f_s(t, t_i, t_f) = \frac{t - t_i}{t_f - t_i}$. The orientation of the formation frame $\mathcal{F}_f$ is defined as having its x-axis parallel to the tangent vector of $p_f^r(t)$.[5] Each leader propagates its (possibly perturbed) vector v_l, $l \in \mathcal{L}$ as described in this paper. Each robot i is assumed to know its unique time-invariant formational offset vector $\xi_i \in \mathbb{R}^2$ specifying its desired offset from $p_f^r(t)$ with respect to the formation frame. Once it accepts a parameter vector v_i, each robot reconstructs the corresponding trajectory, determines the orientation of $\mathcal{F}_f$ from the tangent vector of the trajectory, and uses $\mathcal{F}_f$ and ξ to determine its local desired reference point $p_i^r(t)$.

[5] Note that for a C^1 curve in $\mathbb{R}^2$ this orientation is unique.

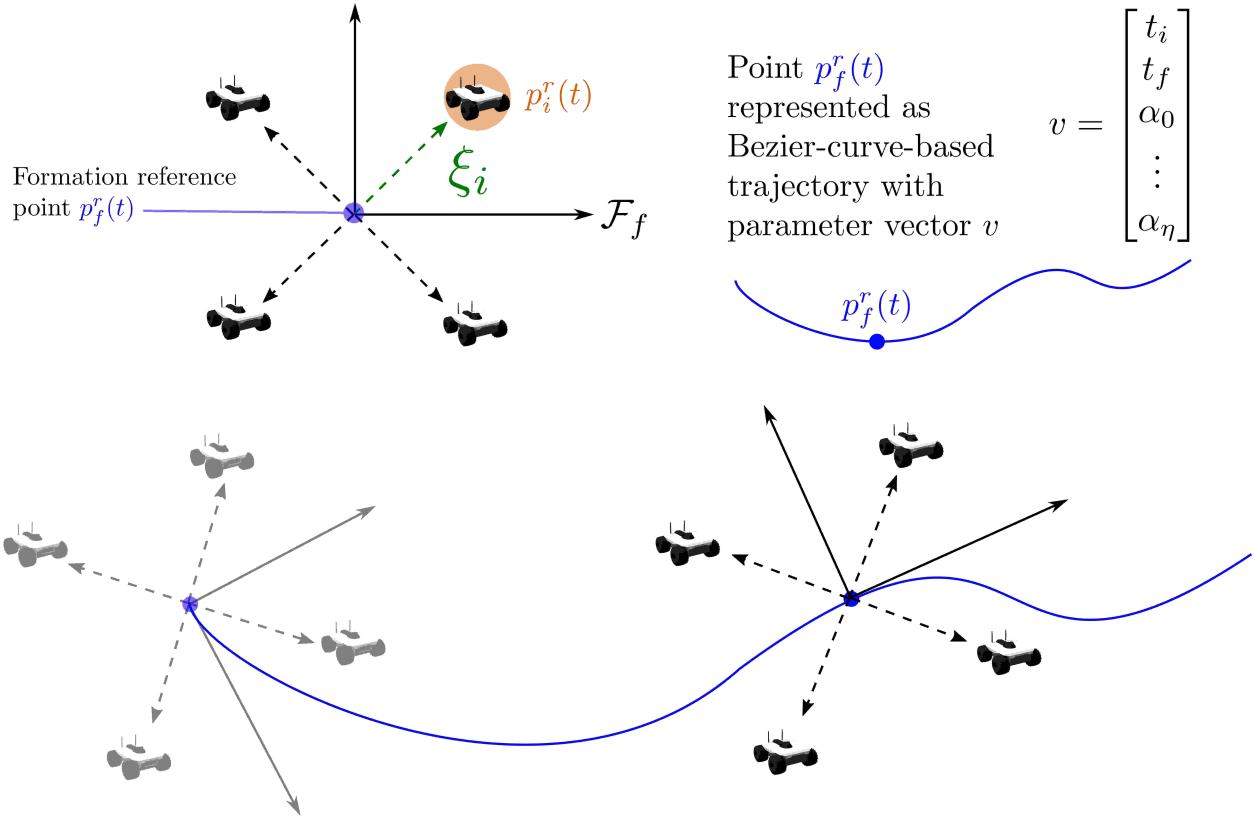


Figure 4.5: Depiction of the method used to specify formational offsets in the simulations. The x-axis of the formation frame $\mathcal{F}_f$ is defined to be colinear with the tangent vector to the Bezier curve at the time-varying reference point $p_f^r(t)$.

4.6.2 Simulation 1

In all simulations, the value of b is set as $b = 0.1$, and the value of the safety distance R as $R = 1$. All applicable units are SI.

In the first simulation, the set of leader robots is $\mathcal{L} = \{1, 2, 3, 4, 5\}$. The set of misbehaving robots is $\mathcal{A} = \{2, 7\}$, which is an F-total model with $F = 2$. Note that the misbehaving set $\mathcal{A}$ includes one leader and one follower. The misbehaving robots both propagate incorrect, arbitrary vector messages v_k, $k \in \mathcal{A}$ to their out-neighbors and physically misbehave by moving off to infinity in arbitrary directions. Any deliberate attempts to collide with other robots would clearly identify an robot as adversarial; therefore to avoid detection, the misbehaving robots apply the nominal CBF modification in (4.48) to avoid collisions with other robots in this simulation.

The nominal graph structure is a 5-circulant digraph [180, 243]. Under the given set of leaders, it can be verified that this forms an RDAG with parameter $r = 2F + 1 = 5$. Message broadcasting to out-neighbors is performed asynchronously with each robot i broadcasting its vector v_i at time instances $t = q\gamma_i$, $q \in \mathbb{Z}_+$, with γ_i being chosen randomly from the interval $[0.2, 0.4]$ using the uniform distribution. To simulate clock perturbations in each reconstructed trajectory $P^i(t)$, each robot's time estimate model is $t_i(t) = t + w_i^t(t)$ where each $w_i^t(t)$, $i \in \mathcal{V}$ is a Gaussian noise vector with zero mean and variance $\sigma^2 = 1 \times 10^{-6}$ sec^2.

The nominally specified trajectory is visualized in Figure 4.6. The nominal parameter vector v has the structure

$$v = \begin{bmatrix} t_i & t_f & \alpha_0 & \cdots & \alpha_6 \end{bmatrix}, \tag{4.51}$$

with the parameters being

$$\begin{aligned} t_i &= 0 & \alpha_1 &= \begin{bmatrix} 20 & 35 \end{bmatrix}^T & \alpha_4 &= \begin{bmatrix} -20 & 65 \end{bmatrix}^T \\ t_f &= 250 & \alpha_2 &= \begin{bmatrix} 20 & -65 \end{bmatrix}^T & \alpha_5 &= \begin{bmatrix} -20 & -35 \end{bmatrix}^T \\ \alpha_0 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T & \alpha_3 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T & \alpha_6 &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T. \end{aligned}$$

Each normal leader's parameter vector v_l, $l \in \mathcal{L}^{\mathcal{N}}$ satisfies $v_l = v + w_l^v$ where each w_l^v, $l \in \mathcal{L}$ is a Gaussian noise vector with zero mean and variance $\sigma^2 = 0.1$. The maximum error ϵ_{lp} for the leaders satisfies $\epsilon_{lp} = 0.3467$.

Still frames from Simulation 1 are shown in Figure 4.7. Each robot is represented by a solid circle with a dotted circle representing the collision avoidance radius. Each robot's reconstructed Bezier curve trajectory P^i is represented with a dotted line representing the full path and a moving diamond representing the robot's current reconstructed $P^i(t)$ value at time t.

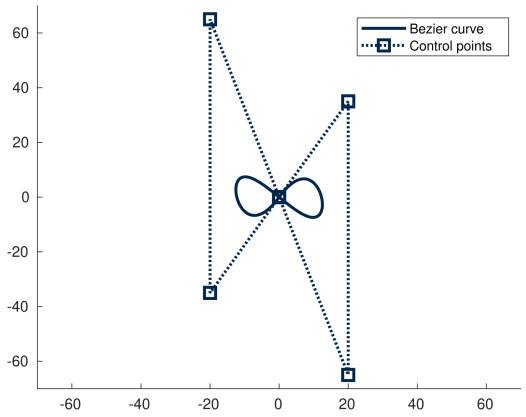


Figure 4.6: The nominal Bezier path for Simulation 1, shown as a solid blue line. The Bezier control points are shown as squares, with dotted lines connecting the control points for clarity of visualization. The exact trajectory is not known to any of the leaders or followers; leaders each have parameter vectors perturbed from the nominal parameters for this trajectory.

Figure 4.8 depicts the maximum pointwise reconstructed trajectory error between any two normal robots in the network. Note that 4.8 plots the data beginning at the first time instance where all normal robots have accepted a parameter vector ($t = 0.39$ seconds). A plot of the minimum interrobot distances is depicted in Figure 4.9, showing that collision avoidance was maintained for the network. Figure 4.9 shows the minimum inter-robot distances over time and demonstrates that collision avoidance was maintained between all robots.

4.6.3 Simulation 2

The second simulation runs under the same conditions as the first simulation with the following exceptions. The set of malicious robots is $\{4, 9\}$. A different nominal Bezier trajectory is used, which is shown in Figure 4.10. The parameters for this Bezier trajectory are

$$
\begin{aligned}
t_i &= 0 & \alpha_4 &= \begin{bmatrix} -12.5, 0 \end{bmatrix} \\
t_f &= 250 & \alpha_5 &= \begin{bmatrix} -12.5 & 12.5 \end{bmatrix} \\
\alpha_0 &= \begin{bmatrix} 12.5 & 0 \end{bmatrix} & \alpha_6 &= \begin{bmatrix} 0 & -12.5 \end{bmatrix} \\
\alpha_1 &= \begin{bmatrix} 12.5 & 12.5 \end{bmatrix} & \alpha_7 &= \begin{bmatrix} 12.5, -12.5 \end{bmatrix} \\
\alpha_2 &= \begin{bmatrix} 0 & 12.5 \end{bmatrix} & \alpha_8 &= \begin{bmatrix} 12.5, 0 \end{bmatrix} \\
\alpha_3 &= \begin{bmatrix} -12.5 & 12.5 \end{bmatrix}.
\end{aligned}
$$

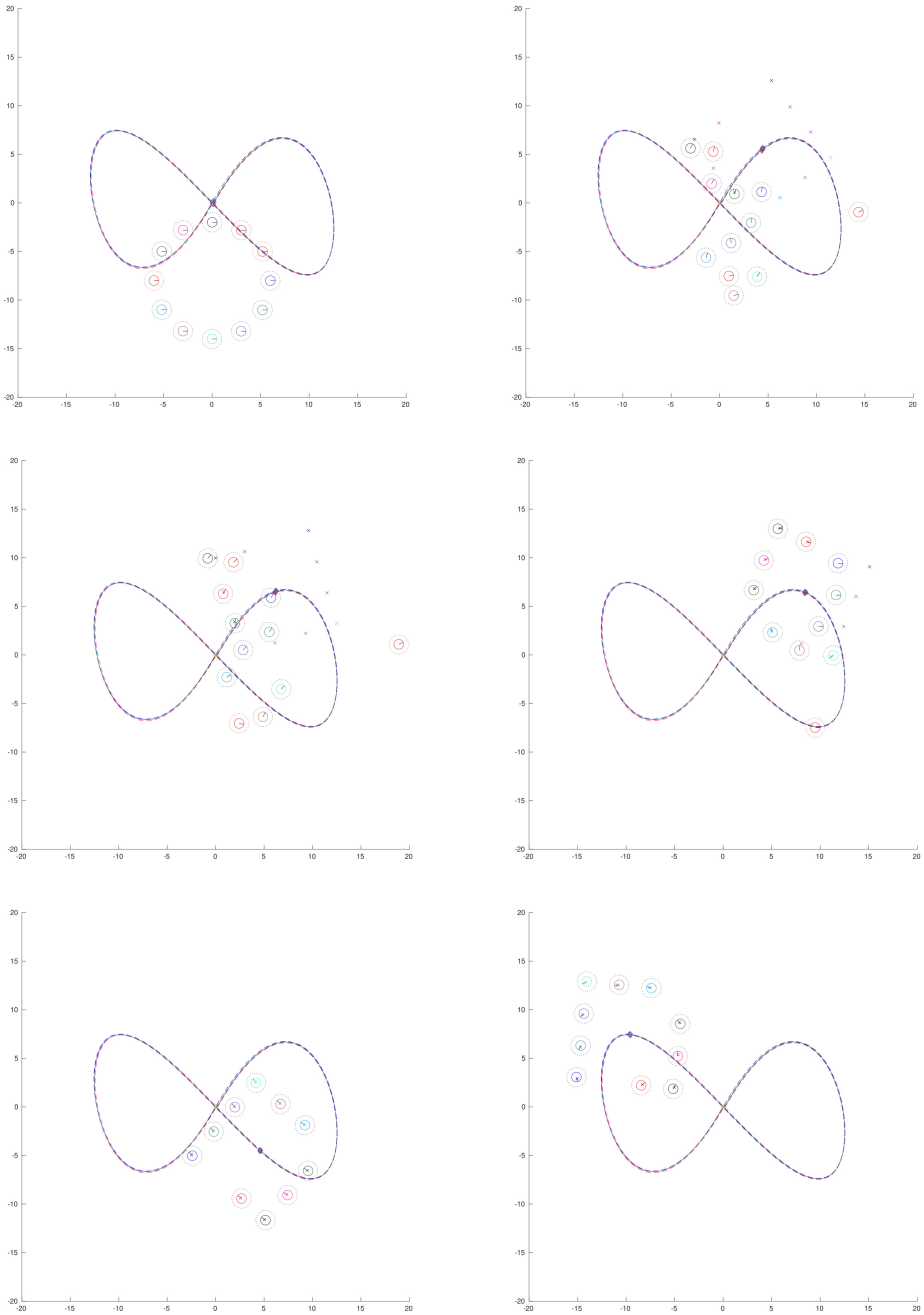


Figure 4.7: Still frames from the video of Simulation 1. The dotted lines represent the reconstructed trajectories for normal robots. The diamonds on the dotted line trajectories represent each robot's reconstructed estimate of the formation reference point. The small x marks represent each normal robots' time-varying desired position $p_i^r(t)$. Two adversarial robots (red) move off towards infinity while simultaneously propagating misinformation through the network.

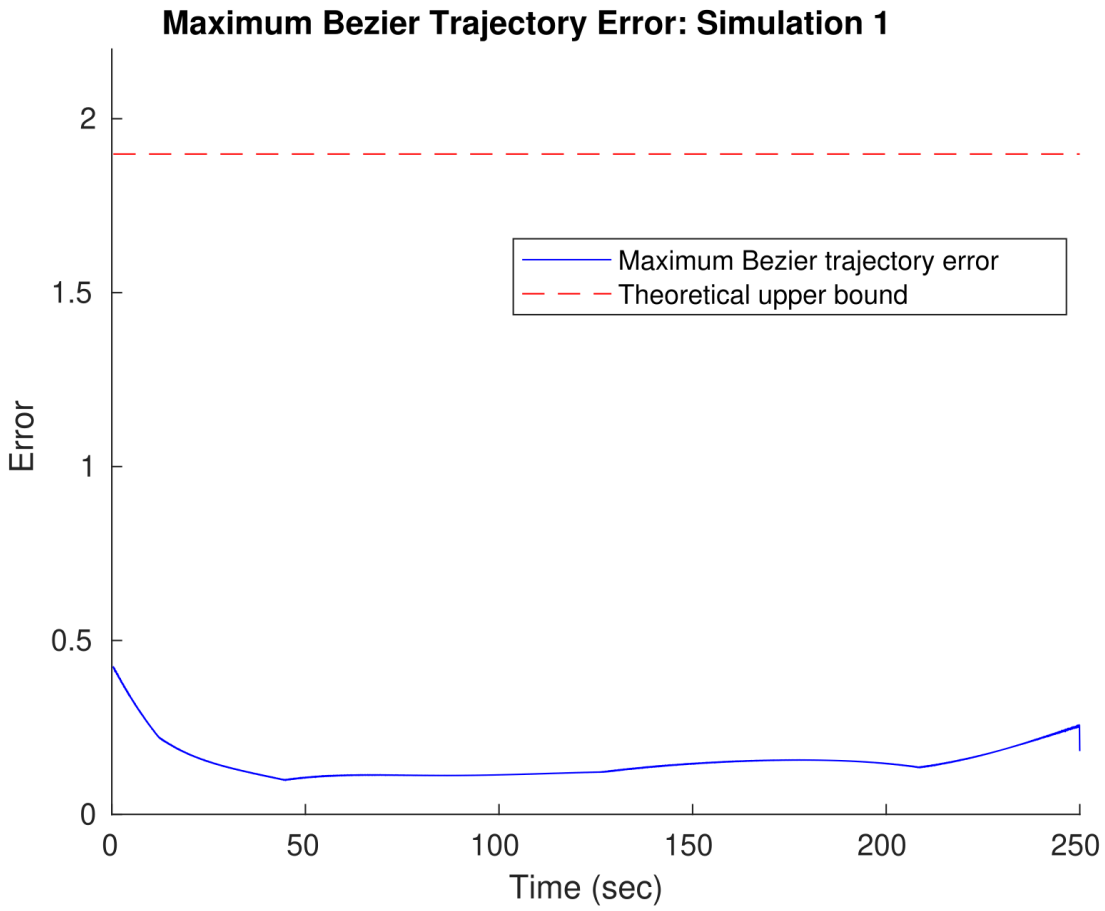


Figure 4.8: Plot of the maximum pointwise error between all pairs of normal robot reconstructed target trajectories for Simulation 1, along with the theoretical upper bound. The theoretical upper bound derived in this chapter is quite conservative for the given problem data; future work will investigate ways to tighten this bound.

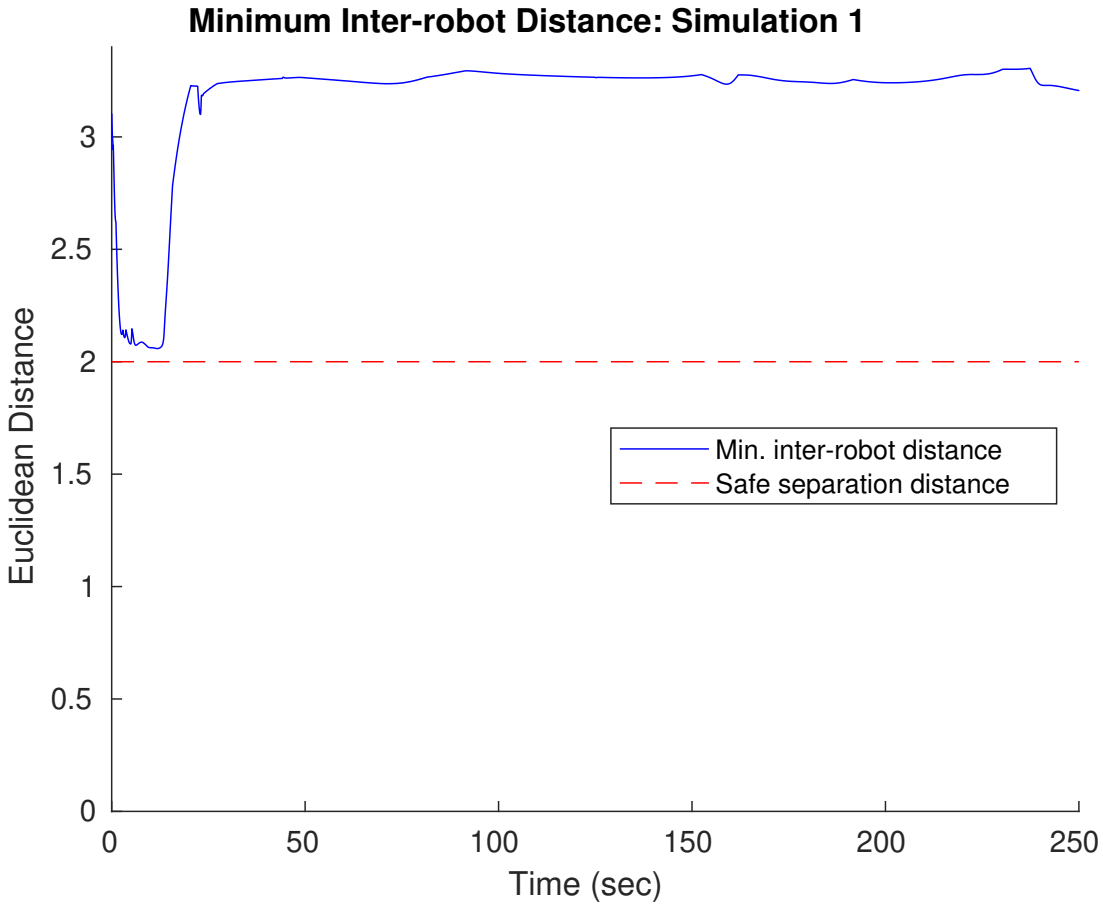


Figure 4.9: Plot of the minimum inter-robot distances in Simulation 1. The red dotted line represents the minimum inter-robot distance required for safety to be maintained.

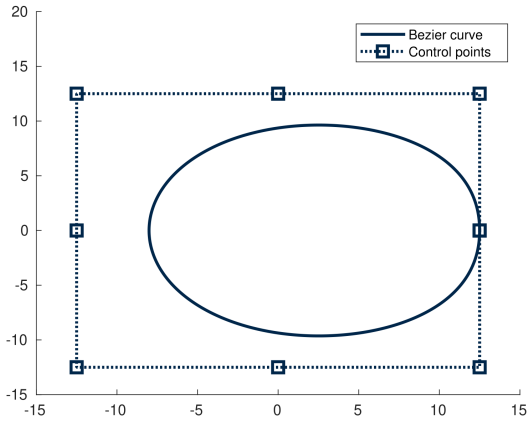


Figure 4.10: The nominal Bezier path for Simulation 2, shown as a solid blue line. The Bezier control points are depicted as squares, with dotted lines connection the control points for clarity of visualization. As in Simulation 1, this exact trajectory is not known to either leaders or followers. Leaders each have parameter vectors perturbed from the nominal parameters representing this trajectory.

Three large circular obstacles are included in this simulation, which are represented as black solid circles in the still frames. The first has radius $R_{o_1} = 20$ and is centered at $p_{o_1} = [0, 32.5]$; the second has radius $R_{o_2} = 5$ and is centered at $p_{o_2} = [2.5, 0]$; and the third has radius $R_{o_3} = 10$ and is centered at $p_{o_3} = [-15, -15]$. Collision avoidance between each agent j and the obstacles is incorporated by defining the functions

$$h_{jo_i}(z_j, p_{o_i}) = \|z_j - p_{o_i}\|_2^2 - (R + b + R_{o_i})^2, \tag{4.52}$$

and incorporating these functions into the QP constraints for each agent as per (4.48).

The leader robots for this simulation are the same as in Simulation 1. Similar to Simulation 1, each normal leader's parameter vector v_l, $l \in \mathcal{L}^{\mathcal{N}}$ satisfies $v_l = v + w_l^v$ where each w_l^v, $l \in \mathcal{L}$ is a Gaussian noise vector with zero mean and variance $\sigma^2 = 0.1$. The maximum error ϵ_{lp} for the leaders for Simulation 2 satisfies $\epsilon_{lp} = 0.2894$.

The nominal network formation is a circle of radius 8 with the formational reference point as the center and each robot spaced equidistantly around the circle's edge.

Each time a vector message is broadcasted, it is received successfully by the recipient with probability $p = 0.5$, which models the effect of packet drops in the network [244].

Still frames from Simulation 2 are shown in Figure 4.11. A plot of the minimum interrobot distances is depicted in Figure 4.13, showing that collision avoidance was maintained for the network. Figure 4.12 depicts the maximum pointwise reconstructed trajectory error between any two nor-

mal robots in the network. Note that Figure 4.12 plots the data beginning at the first time instance where all normal robots have accepted a parameter vector ($t = 2.88$ seconds).

4.6.4 Hardware Experiments

In addition to the simulations presented previously, a preliminary version of the methods in this chapter was implemented on a 6-robot hardware platform. The robots used were the AION R1 rover [] which used the NVidia Jetson TX2 as onboard computers and the ROS framework for communication middleware. Experiments were run in the M-Air facility at the University of Michigan.

The network of 6 robots consisted of 3 leaders and 3 followers in an RDAG with parameter 3, as depicted in Figure 4.15. The nominal formation for the system was a circle with a diameter of 3 meters with agents distributed equidistantly about the edge. Each agent was given a local offset that, when added to the time-varying center of formation described by the Bezier trajectory, resulted in that agent's local formational target point. The nominal trajectory for the center of formation was a time-varying point moving along the edge of a circle with diameter 8 meters. The hardware experiments implemented the equivalent of Algorithm 4.1. The adversarial set was 1-local, with one of the leader agents misbehaving by sending arbitrary misinformation to its outneighbors. Collision avoidance was accomplished by blending each agent's nominal trajectory tracking controller with a set of Lyapunov-like barrier functions [148, 150, 245]. A video of the demonstration can be accessed at `https://youtu.be/PoMOhzbg3PI`. All normally-behaving followers were able to successfully reconstruct and track the formational trajectory broadcasted by the normally-behaving leaders despite the misinformation being broadcast by the adversarial leader.

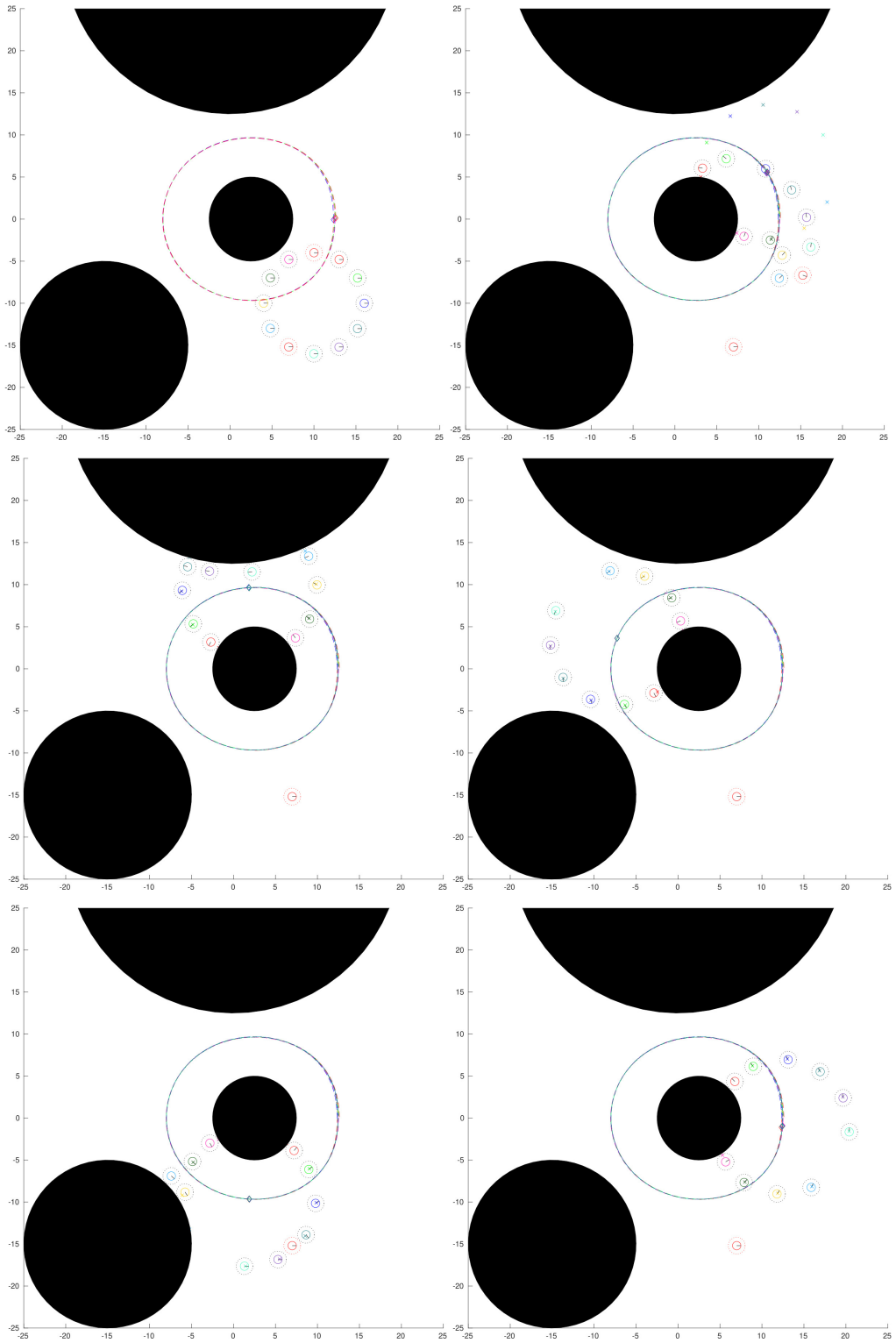


Figure 4.11: Still frames from the video of Simulation 2. The dotted lines represent the reconstructed trajectories for normal robots. The diamonds on the dotted line trajectories represent each robot's reconstructed estimate of the formation reference point. The small x marks represent each normal robots' time-varying desired position $p_i^r(t)$. Both adversarial robots propagate misinformation throughout the network. One adversarial robots (red) moves off towards infinity while the other remains in place for the entire simulation.

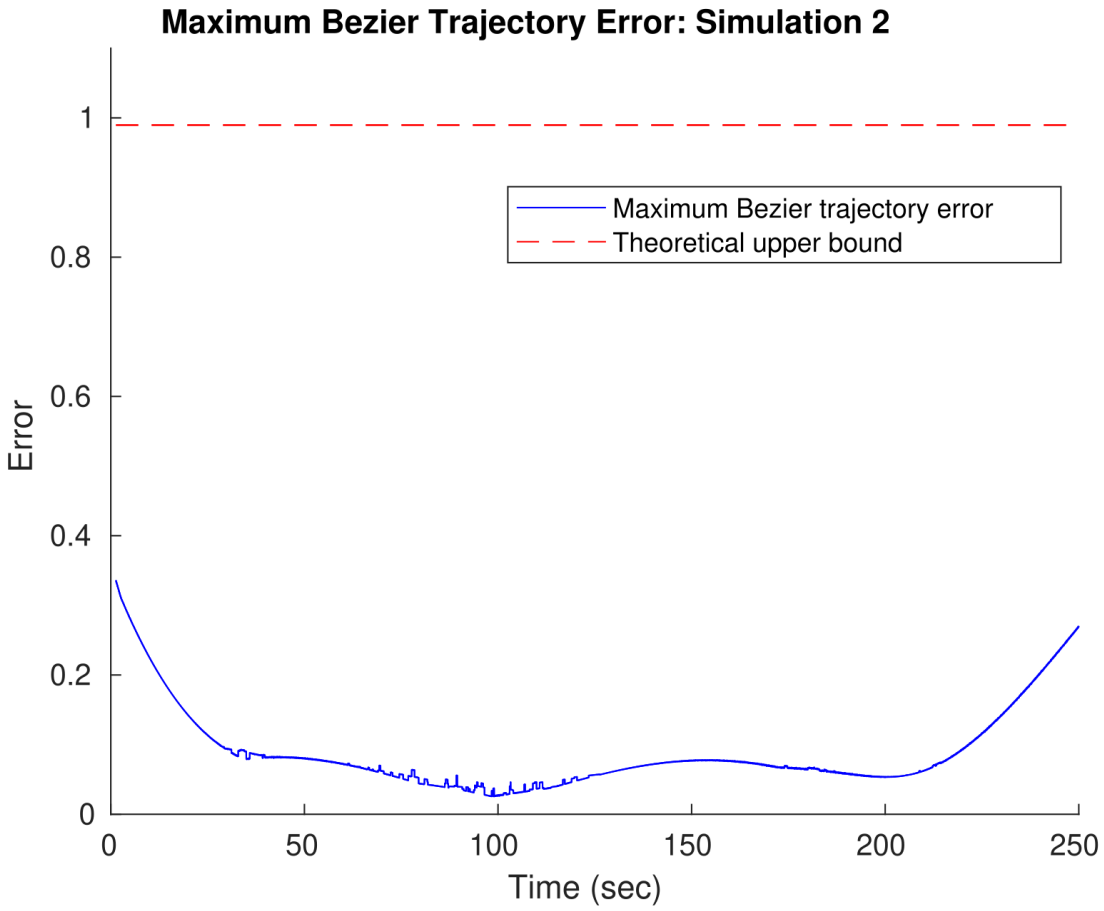


Figure 4.12: Plot of the maximum pointwise error between all pairs of normal robot reconstructed target trajectories for Simulation 2, along with the theoretical upper bound. The plot begins at time $t = 1.33$ seconds when all normal robots have accepted a parameter vector and reconstructed a trajectory. Again, the theoretical upper bound derived in this chapter is quite conservative for the given problem data; future work will investigate ways to tighten this bound.

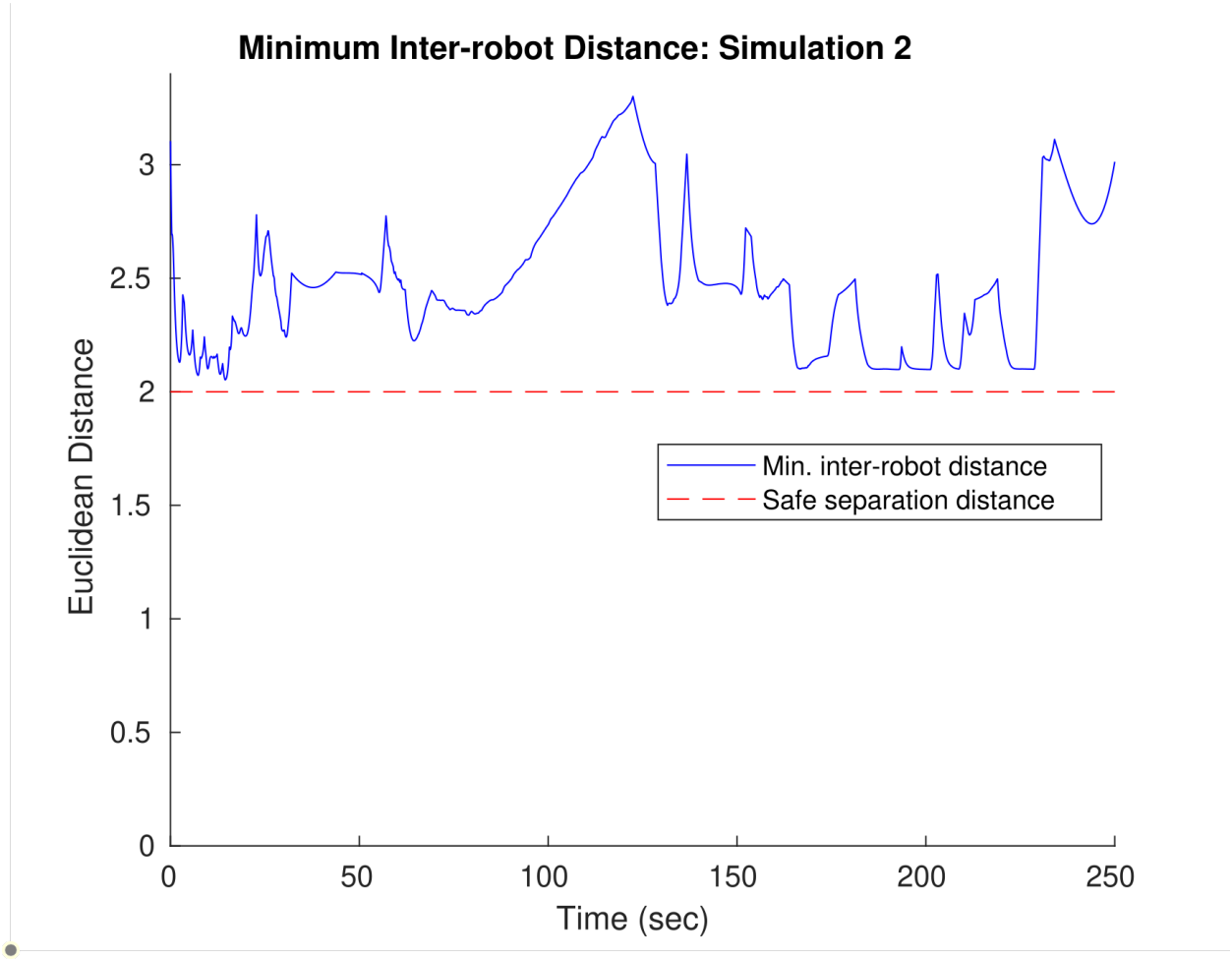


Figure 4.13: Plot of the minimum interrobot distances in Simulation 2. The red dotted line represents the minimum inter-robot distance required for safety to be maintained.

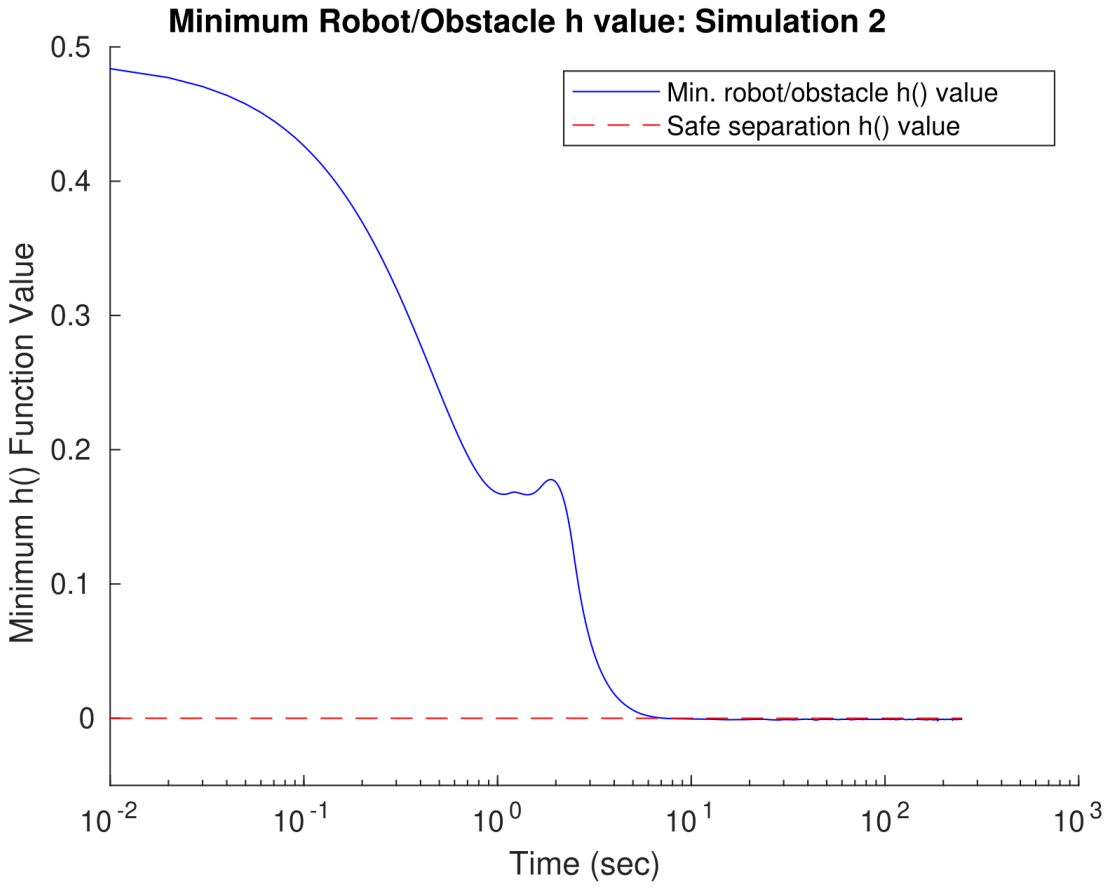


Figure 4.14: Minimum value of $h_{jo_i}(z_j(t), p_{o_i})$, as defined in (4.52), over all agents $j \in \mathcal{V}$ and obstacles o_1, o_2, o_3 as a function of time in Simulation 2. A log scale is used in the x-axis for greater clarity. This value never decreases below zero, which indicates there were no agent-obstacle collisions for all agents and obstacles.

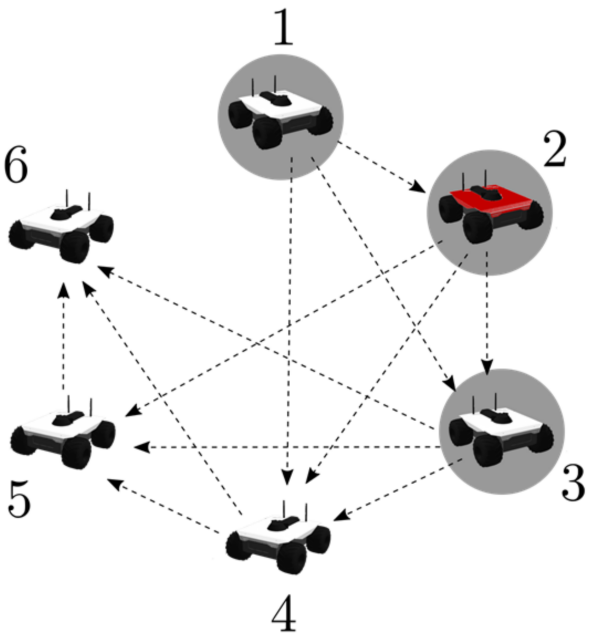


Figure 4.15: Depiction of the network structure for the hardware experiments. Agents 1 through 3 are leaders, and agents 4 through 6 are followers. Agent 2 is a misbehaving leader and propagates misinformation to its out-neighbors.

4.7 Conclusion

In this chapter we presented a method for a multi-robot system to resiliently propagate a vector of parameters from a set of normally-behaving leaders to all normally-behaving followers in the presence of faulty or adversarial robots. The method is able to operate even with asynchronous communication and perturbations to the data which represent a time-varying trajectory via static parameters. An analysis of the effects of clock synchronization errors and parameter perturbations on the trajectories reconstructed from the vector parameters was performed. Simulations were presented using Control Barrier Function quadratic programming techniques to achieve time-varying formation tracking in a multi-robot system while preserving the safety of the robots. Future work will incorporate distributed path planning methods into the framework presented in this chapter, and will explore the resilient propagation of time-varying evolutions of formational offsets and formational frame orientations from leaders to followers.

4.8 Appendix

4.8.1 Bernstein Polynomials and Bezier Curves

The Bernstein basis polynomials of degree n are defined as follows:

$$b_{j,n}(t) = \binom{n}{j} t^j (1-t)^{n-j},\ j = 0, \ldots, n \tag{4.53}$$

Bernstein basis polynomials of degree n form a basis for the vector space of all polynomials having real coefficients and degree of at most n. Notably, these basis polynomials form a partition of unity:

$$\sum_{j=0}^{n} b_{j,n}(t) = 1 \tag{4.54}$$

A Bezier curve is a linear combination of Bernstein basis polynomials. Given control points $\alpha_0, \ldots, \alpha_n \in \mathbb{R}^m$, the corresponding Bezier curve $P(t)$ is

$$P(t) = \sum_{j=0}^{n} \alpha_j b_{j,n}(t) \tag{4.55}$$

The rth derivative of a Bezier curve $P(t) = \sum_{j=0}^{n} \alpha_j b_{j,n}(t)$ with $t \in [0,1]$ is itself a Bezier curve, and can be expressed as follows: define

$$\Delta^0 \alpha_i = \alpha_i \tag{4.56}$$

$$\Delta^r \alpha_i = \Delta^{r-1} \alpha_{i+1} - \Delta^{r-1} \alpha_i \tag{4.57}$$

The rth derivative of $P(t)$ is then given by the following equation [246, Sec 2.4], [247]:

$$P^{(r)}(t) = \frac{n!}{(n-r)!} \sum_{j=0}^{n-r} \Delta^r \alpha_j b_{j,n-1}(t) \tag{4.58}$$

Several upper bounds on the value of $\left\| \frac{d}{dt} P(t) \right\|$ have been established in prior literature [248]. For a Bezier curve $P(t) = \sum_{j=0}^{n} \alpha_j b_{j,n}(t)$ with $t \in [0,1]$, a simple (although non-tight) upper bound given in [249] is

$$\left\| \frac{d}{dt} P(t) \right\| \leq n \max_i \|\alpha_{i+1} - \alpha_i\| \ \ \forall t \in [0,1]. \tag{4.59}$$

Since higher-order derivatives of $P(t)$ are themselves Bezier curves, it is straightforward to derive

the following expression from (4.59):

$$\left\|P^{(r)}(t)\right\| \leq (n-r+1)\max_i \left\|\Delta^{r-1}\alpha_{i+1} - \Delta^{r-1}\alpha_i\right\|,$$
$$\forall t \in [0,1].$$

This can be seen by noting that $P^{(r-1)}(t)$ is an $(n-r+1)$th order Bezier curve with control points $\Delta^{r-1}\alpha_i$.

CHAPTER 5

Adversarial Resilience for Sampled-Data Systems under High-Relative-Degree Safety Constraints

5.1 Introduction

Guaranteeing the safety of autonomous systems is a critical challenge in modern control theory. Safety is frequently modeled by defining a safe subset of the state space for a given system and generating control inputs that render this subset forward invariant. Control barrier function (CBF) methods [160, 168, 171, 250, 251] that leverage quadratic programming (QP) techniques have risen as a powerful framework for establishing forward invariance of a safe set. Both single-agent [161, 165, 176, 252] and multi-agent systems [251, 253–256] have been considered, where agents have control-affine dynamics. Multi-agent CBF techniques have been applied to a variety of settings including collision avoidance for quadrotors [257] and mobile robots [258], accomplishing spatiotemporal tasks [174, 250], forming or maintaining network communication topologies between mobile agents [256, 259], and more.

Prior work on multi-agent CBF methods typically assumes that all agents apply the nominally specified control law. This assumption does not encompass faulty or adversarial behavior of agents within the system. Adversarial agents may apply control laws specifically crafted in an attempt to violate set invariance conditions within given control constraints. Much recent work has considered the accomplishment of control objectives in the presence of faulty or adversarial agents [180–182, 260, 261]. However, these prior works either do not consider safety bounds on the system or do not provide explicit safety guarantees. CBFs are used in [256] to construct resilient network communication topologies in finite time; however, all agents are assumed to apply the nominal CBF-based controller without any adversarial misbehavior with respect to control actions.

In addition, the majority of prior work involving CBF methods considers a continuous-time system with continuous inputs. Practical systems are often more appropriately modeled using sampled-data dynamics, where state measurements and control inputs remain constant between

sampling times. Notable studies that have explicitly considered the effects of sampling in CBF methods include [176, 177]. However these papers do not consider multi-agent systems and do not consider the presence of faulty or adversarial agents. Many systems also consider a CBF having high relative degree with respect to agents' dynamics, where the control input of the agents does not appear in the expression for the first derivative of the function whose sublevel or superlevel sets describe the safe set (e.g., systems with double-integrator dynamics). Methods to apply CBF set-invariance methods to such systems have been presented in prior literature [252, 262]; however these methods do not consider sampled-data dynamics and do not consider the presence of adversarial agents.

In this chapter, we present a framework for guaranteeing set invariance in sampled-data multi-agent systems in the presence of adversarial agents. This framework considers a class of functions describing the safe set that have high relative degree with respect to system dynamics. Unlike [176], this chapter explicitly considers multi-agent systems, asynchronous sampling times with clock disturbances, the presence of worst-case adversarially behaving agents, and functions describing safe sets that have high relative degree with respect to (w.r.t.) the system dynamics. Our specific contributions are as follows: First, we present a method under which a set of normally-behaving agents in a system with sampled-data dynamics can collaboratively render a safe set forward invariant despite the actions of adversarial agents. Our analysis considers asychronous sampling times and distributed calculation of agents' control inputs. Second, we present a method under which a system of normally-behaving agents with sampled-data dynamics can render a safe set forward invariant in the presence of adversarial agents when the safe set is described by a class of functions with high relative degree with respect to agents' dynamics. The class of functions will be described in more detail in Section 5.5.

The organization of this chapter is as follows: Section 5.2 gives a brief overview of control barrier function methods. Section 5.3.1 outlines the notation and problem formulation. Section 5.4 presents the main results for safe set functions having a relative degree of 1 with respect to system dynamics. Section 5.5 presents the main results for a class of safe set functions having a relative degree strictly greater than 1 with respect to system dynamics. Section 5.6 outlines simulations demonstrating this chapter's results. Section 5.7 gives a conclusion to this chapter.

5.2 Overview of Control Barrier Function Methods

This section briefly reviews control barrier function (CBF) methods for guaranteeing forward invariance of a safe set. There is an extremely large literature devoted to safety techniques using CBFs; here we review only concepts related to the contributions of this chapter. For a more thorough overview of CBFs, we refer the reader to [160].

We first consider a system consisting of a single agent with state $x \in \mathbb{R}^n$ having the following nonlinear control affine dynamics:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t). \tag{5.1}$$

The functions f, g are assumed to be locally Lipschitz on $\mathbb{R}^n$. Without loss of generality we assume that $t_0 = 0$. Control inputs $u(t)$ are assumed to fall within a feasible control set $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$. In many practical applications the feasible control set $\mathcal{U}$ is represented or approximated by a polytope of the form $\{u \in \mathbb{R}^m : A_u u \leq b_u\}$, with $A_u \in \mathbb{R}^{m \times p}$, $b_u \in \mathbb{R}^p$.

A subset of the state space $S \subseteq \mathbb{R}^n$ is designated as a *safe set*. Under the assumption that $x(0) \in S$, one of the objectives of the system is to guarantee that the trajectory $x(t)$ of (5.1) remains in S for all forward time; i.e., $x(t) \in S \ \forall t \geq 0$. The safe set S is often modeled as the sublevel sets of a locally Lipschitz function $h \in \mathcal{C}^{1,1}_{loc}$ as follows:[1]

$$\begin{aligned} S &= \{x \in \mathbb{R}^n : h(x) \leq 0\}, \\ \partial S &= \{x \in \mathbb{R}^n : h(x) = 0\}, \\ \text{int}(S) &= \{x \in \mathbb{R}^n : h(x) < 0\}. \end{aligned} \tag{5.2}$$

Set invariance methods for more general set descriptions have been derived [251, 255, 263, 264], but the above definition of S is commonly used in prior literature [161, 265].

Necessary and sufficient conditions on set invariance under general nonlinear system dynamics were given by Nagumo in what is now called Nagumo's Theorem [129]. Roughly speaking, the theorem states that a set remains invariant under given system dynamics if and only if at every point on the boundary of the set the system velocity points "inside the set". The reader is referred to [263, Thm. 3.1] for a general formulation of Nagumo's Theorem. For purposes of this chapter we give a version tailored to the problem setting at hand. Note that for this result we use the Lie derivative notation $L_f h(x) \triangleq \frac{\partial h(x)}{\partial x} f(x)$.

Theorem 5.1. *Consider the system* (5.1) *and the set* S *defined by* (5.2)*. Assume that* $x(0) \in S$*, and for each initial* $x(0) \in S$ *the system* (5.1) *admits a globally unique solution. Then* $x(t) \in S$ *for all* $t \geq 0$ *if and only if*

$$\dot{h}(x(t)) = L_f h(x(t)) + L_g h(x(t))u(t) \leq 0, \quad \forall x(t) \in \partial S. \tag{5.3}$$

A more conservative form of Nagumo's theorem is commonly used where the right hand side (RHS) (5.3) is replaced by a condition involving an extended class-$\mathcal{K}_\infty$ function α:

[1] The set S can also be equivalently defined in terms of *superlevel* sets.

Corollary 5.1. *Under the conditions of Theorem 5.1, a sufficient condition for $x(t) \in S$ for all $t \geq 0$ is the following:*

$$\dot{h}(x(t)) = L_f h(x(t)) + L_g h(x(t))u(t) \leq -\alpha(h(x(t))), \quad \forall x(t) \in S. \tag{5.4}$$

Note that by the properties of extended class-$\mathcal{K}$ functions, $h(x) = 0$ for all $x \in \partial S$, which implies that the conditions (5.4) and (5.3) are equivalent on the boundary of S. One reason condition (5.4) is often used is as follows: if there exists a domain $D \subset \mathbb{R}^n$ with $S \subset D$ such that (5.4) holds for all $x \in D$, it can be shown that any trajectory with initial conditon $x(0) \in D \backslash S$ converges asymptotically to the boundary of the set S. This property is useful when considering scenarios where the initial condition of the state is outside of the set S or systems subject to disturbances that may briefly push $x(t)$ outside of S.

Control barrier functions were inspired by Control Lyapunov Functions [163], and are defined as follows:

Definition 5.1 ([160]). *Let $S \subset D \subset \mathbb{R}^n$ be the sublevel set of a function $h \in \mathcal{C}^{1,1}_{loc}$. The function h is a Control Barrier Function (CBF) if there exists an extended class-$\mathcal{K}_\infty$ function α such that for the system (5.1),*

$$\sup_{u \in \mathcal{U}} \left[L_f h(x) + L_g h(x) u \right] \leq -\alpha(h(x)), \quad \forall x \in D. \tag{5.5}$$

For compact $\mathcal{U}$, $\max$ may be used in (5.5) instead of $\sup$. The existence of a CBF for a set S serves as both a certificate that rendering a set forward invariant is possible, and as a means to calculate control inputs that restrict $x(t)$ to remain within S. If there exists a CBF for a given safe set S, then at any state $x \in D$ there exists at least one feasible control input satisfying the sufficient condition (5.4) for forward invariance of S. The set of all such feasible control inputs can be expressed as

$$K(x) \triangleq \left\{ u \in \mathcal{U} : L_f h(x) + L_g h(x) u \leq -\alpha(h(x)) \right\}.$$

When $u(\cdot)$ is assumed to be applied in a continuous manner, any Lipschitz continuous control input $u(t)$ that lies within $K(x(t))$ for all $t \geq 0$ will provably render the set S forward invariant. However, generalizations of this result have been made for control inputs $u(\cdot)$ that are only Lebesgue measurable and possibly discontinuous [255, 266].

Computing an input $u \in K(x)$ can be done efficiently by using convex optimization techniques. Suppose there exists a nominal control input $u_{\text{nom}}(t)$ computed for the purpose of accomplishing an objective such as converging to a goal location, tracking a trajectory, etc. We seek to minimally

modify $u_{\text{nom}}(t)$ in the sense of the Euclidean norm such that $x(t) \in S$ is guaranteed for forward time and the control input constraints $u \in \mathcal{U}$ are satisfied. More specifically, we seek to compute a control input $u \in K(x(t)) \cap \mathcal{U}$ such that the objective function $\|u - u_{\text{nom}}(t)\|_2$ is minimized. Note that if $u_{\text{nom}}(t) \in K(x(t)) \cap \mathcal{U}$, then we trivially have $u = u_{\text{nom}}(t)$ and we may simply apply the nominal controller. This problem may be expressed as a convex quadratic program (QP) as follows:

$$\begin{aligned} u(x) = \arg\min_{u \in \mathcal{U}} \quad & \|u - u_{\text{nom}}\|_2 \\ \text{s.t.} \quad & L_f h(x) + L_g h(x) u \leq -\alpha(h(x)) \\ & A_u u \leq b_u \end{aligned} \tag{5.6}$$

This QP is parametric in the state x, meaning that x is not an optimization variable but the entries of the constraint equations may change as x changes. The optimization variable is u, the objective is quadratic in u, and both constraints are affine in u. The first constraint ensures that the computed u lies within $K(x)$, and the second constraint ensures that the computed u lies within $\mathcal{U} = \{u \in \mathbb{R}^m : A_u u \leq b_u\}$. Prior literature typically assumes that the resulting $u(x)$ computed from (5.6) is locally Lipschitz continuous in x. In practice however it is impossible for the QP (5.6) to be solved infinitely often, and a more accurate model for the system dynamics is a sampled-data approach such as zero-order-hold (ZOH) dynamics. Section 5.3 of this chapter will introduce such an approach in more detail.

More general methods exist for computing safety-preserving controllers using CBFs and QP methods, including combining CBFs with Control Lyapunov Functions to compute controllers satisfying both safety and convergence to a goal set simultaneously. However these methods are beyond the scope of this chapter; we refer the interested reader to [160] for further reading.

5.3 Adversarial Resilience in Sampled-Data Systems Under Safety Constraints

As discussed in the Introduction to this chapter, no prior work on set invariance techniques using CBF techniques has considered the presence of adversarial agents with the specific intent to violate safety bounds. In addition, no prior work has considered multi-agent set invariance using CBF techniques when the agents have sampled-data dynamics, which more accurately models the state evolution in practice. We now present our framework for guaranteeing set invariance in sampled-data multi-agent systems in the presence of adversarial agents.

5.3.1 Notation and Problem Formulation

The nonnegative and strictly positive integers are denoted $\mathbb{Z}_{\geq 0}$ and $\mathbb{Z}_{>0}$, respectively. We use the notation $h \in \mathcal{C}^{1,1}_{loc}$ to denote a continuously differentiable function h whose gradient ∇h is locally Lipschitz continuous. Let $x_i \in \mathbb{R}^{n_i}$, $n_i \in \mathbb{Z}_{\geq 1}$ for $i = 1, \ldots, N$ be a set of vectors, and let $\bar{n} = \sum_{i=1}^{N} n_i$. We let $\vec{x} = \left[x_1^T, \ldots, x_N^T\right]^T$ denote the vector concatenating all x_i vectors. The partial Lie derivative of a function $f(\vec{x})$ with respect to x_i is denoted $L_f h^{x_i}(\vec{x}) = \frac{\partial h(\vec{x})}{\partial x_i} f(\vec{x})$. The n-ary Cartesian product of sets $S_1, \ldots, S_N$ is denoted $\bigtimes_{i=1}^{N} S_i = S_1 \times \ldots \times S_N$. The Minkowski sum of sets S_1, S_2 is denoted $S_1 \oplus S_2$. The open and closed norm balls of radius $\epsilon > 0$ centered at $\vec{x} \in \mathbb{R}^n$ are respectively denoted $B(\vec{x}, \epsilon)$, $\overline{B}(\vec{x}, \epsilon)$. The boundary and interior of a set $S \subset \mathbb{R}^n$ are denoted ∂S and $\text{int}(S)$, respectively.

5.3.2 Problem Formulation

Consider a group of $N \in \mathbb{Z}_{>0}$ agents, with the set of agents denoted by $\mathcal{V}$ and each agent indexed $\{1, \ldots, N\}$. Each agent $i \in \mathcal{V}$ has the state $x_i \in \mathbb{R}^{n_i}$, $n_i \in \mathbb{Z}_{>0}$ and input $u_i \in \mathbb{R}^{m_i}$, $m_i \in \mathbb{Z}_{>0}$. The system and input vectors $\vec{x}, \vec{u}$, respectively, denote the vectors that concatenate all agents' states and inputs, respectively, as $\vec{x} = \left[x_1^T, \ldots, x_N^T\right]^T$, $\vec{x} \in \mathbb{R}^{\bar{n}}$ and $\vec{u} = \left[u_1^T, \ldots, u_N^T\right]$, $\vec{u} \in \mathbb{R}^{\bar{m}}$, $\bar{n} = \sum_{i=1}^{N} n_i$, $\bar{m} = \sum_{i=1}^{N} m_i$. Agents receive knowledge of the system state $\vec{x}$ in a sampled-data fashion; i.e., each agent $i \in \mathcal{V}$ has knowledge of $\vec{x}(\cdot)$ only at times $\mathcal{T}_i = \{t_i^0, t_i^1, t_i^2, \ldots\}$, where t_i^k represents agent i's kth sampling time, with $t_i^{k+1} > t_i^k \; \forall k \in \mathbb{Z}_{\geq 0}$. In addition, at each $t_i^k \in \mathcal{T}_i$ the agent i applies a ZOH control input $u(t_i^k)$ that is constant on the time interval $t \in [t_i^k, t_i^{k+1})$. For brevity, we denote $x_i^{k_i} = x_i(t_i^k)$ and $u_i^{k_i} = u_i(t_i^k)$. The sampled-data dynamics of each agent $i \in \mathcal{V}$ under its ZOH controller on each interval $t \in [t_i^k, t_i^{k+1})$ is as follows:

$$\dot{x}_i(t) = f_i(x_i(t)) + g_i(x_i(t))u_i(t_i^k) + \phi_i(t). \tag{5.7}$$

The functions f_i, g_i may differ among agents, but are all locally Lipschitz on their respective domains $\mathbb{R}^{n_i}$. Note that under these definitions for any $i \in \mathcal{V}$ there exists a matrix $C_i \in \mathbb{R}^{n_i} \times \mathbb{R}^{\bar{n}}$ such that $x_i = C_i \vec{x}$. We abuse notation by sometimes writing the expression $f(x_i)$ as $f(\vec{x})$. The functions $\phi_i : \mathbb{R} \to \mathbb{R}^{n_i}$, $i \in \mathcal{V}$, are locally Lipschitz in t and model disturbances to the system (5.7). Each ϕ_i is bounded as per the following assumption:

Assumption 5.1. *For all $i \in \mathcal{V}$, the disturbances $\phi_i(t)$ satisfy $\|\phi_i(t)\| \leq \phi_i^{\max} \in \mathbb{R}_{\geq 0}, \; \forall t \geq 0$.*

Since each control input $u_i(\cdot)$ is piecewise constant, the existence and uniqueness of solutions to (5.7) are guaranteed by Carathéodory's theorem [267, Sec. 2.2].

Each agent $i \in \mathcal{V}$ has control input constraints that are represented by a nonempty, convex, compact polytope, i.e. $u_i \in \mathcal{U}_i(x_i) = \{u \in \mathbb{R}^{m_i} : A_i(x_i)u \leq b_i(x_i)\}$, where the functions $A_i : \mathbb{R}^{n_i} \to \mathbb{R}^{q_i \times m_i}$, $b_i : \mathbb{R}^{n_i} \to \mathbb{R}^{q_i}$ are locally Lipschitz on their respective domains. Representation of control input constraints as polytopes is common in prior literature [161,171,268]. Similar to prior work, it is assumed there exists a nominal control law $\vec{u}_{\text{nom}}(\cdot)$ that the system computes in order to accomplish some nominal objective [160]. Examples of such a $\vec{u}_{\text{nom}}$ might include a feedback control law to track a time-varying trajectory or to converge to a goal set. The nominal control law is designed without any safety consideration, and therefore it is desired to minimally modify $\vec{u}_{\text{nom}}$ in order to render a safe set $S \subset \mathbb{R}^{\bar{n}}$ forward invariant under the dynamics (5.7). The set S is defined as the sublevel sets of a function $h : \mathbb{R}^{\bar{n}} \to \mathbb{R}$, $h \in C^{1,1}_{loc}$ as follows:

$$\begin{aligned} S &= \{\vec{x} \in \mathbb{R}^{\bar{n}} : h(\vec{x}) \leq 0\}, \\ \partial S &= \{\vec{x} \in \mathbb{R}^{\bar{n}} : h(\vec{x}) = 0\}, \\ \text{int}(S) &= \{\vec{x} \in \mathbb{R}^{\bar{n}} : h(\vec{x}) < 0\}. \end{aligned} \tag{5.8}$$

Assumption 5.2. *The set S is compact.*

Assumption 5.3. *For all $i \in \mathcal{V}$ and $\forall \vec{x} \in S$, the interior of $\mathcal{U}_i(\vec{x})$ is nonempty and $\mathcal{U}_i(\vec{x})$ is uniformly compact near $\vec{x}$.*

Remark 5.1. *Note that the conditions for Assumption 5.3 are trivially satisfied when A_i, b_i are constant and the interior set $\{u \in \mathbb{R}^{m_i} : A_i u < b_i\}$ is nonempty. For a specific example satisfying Assumption 5.3 when $\mathcal{U}_i(\cdot)$ is not constant, see (5.52) in Section 5.6 of this chapter.*

We will refer to functions describing safe sets as simply "safe set functions" for brevity. For multi-agent systems that apply continuous controllers $u_i(t)$ to the dynamics (5.7), forward invariance can be collaboratively guaranteed by satisfying the sufficient condition $\dot{h}(\vec{x}(t)) \leq -\alpha(h(\vec{x}(t)))$ based on Nagumo's theorem [129], where $\alpha(\cdot)$ is an extended class-$\mathcal{K}$ function and locally Lipschitz on $\mathbb{R}$. The dependence of $\vec{x}(t)$ on t will be omitted for brevity. For the multi-agent system (5.7), expanding the term $\dot{h}(\vec{x})$ yields

$$\sum_{i \in \mathcal{V}} (L_{f_i} h^{x_i}(\vec{x}) + L_{g_i} h^{x_i}(\vec{x}) u_i + L_{\phi_i} h^{x_i}(\vec{x})) \leq -\alpha(h(\vec{x})), \tag{5.9}$$

where the partial Lie derivative notation $L_{f_i} h^{x_i}(\vec{x})$ is defined at the beginning of Section 5.3.1. When all agents behave normally, methods exist for agents to locally solve for appropriate local control inputs that together satisfy the condition in (5.9) (e.g. [250]).

In contrast to prior work, this chapter considers systems containing agents that exhibit adversarial behavior. More specifically, this chapter considers a subset of agents $\mathcal{A} \subset \mathcal{V}$ that apply the following control input for all sampling times t_j^k, $k \in \mathbb{Z}_{\geq 0}$, $j \in \mathcal{A}$:

$$u_j^{\max}(\vec{x}^{k_j}) = \arg\max_{u \in \mathcal{U}_j} \left[L_{f_j} h^{x_j}(\vec{x}^{k_j}) + L_{g_j} h^{x_j}(\vec{x}^{k_j}) u \right]. \tag{5.10}$$

The agents in $\mathcal{A}$ are called *adversarial*.

Remark 5.2. *The control input* (5.10) *models adversarial intent in the sense that* (5.10) *maximizes agent j's contribution to the left-hand side (LHS) of* (5.9)*. Violating the inequality in* (5.9) *removes the forward invariance guarantee for the safe set S, and therefore the control law* (5.10) *represents an adversarial agent's best instantaneous control effort towards violating system safety.*

Agents that are not adversarial are called *normal*. The set of normal agents is denoted $\mathcal{N} = \mathcal{V} \backslash \mathcal{A}$. Dividing the left-hand side (LHS) of (5.9) into normal and adversarial parts yields the following sufficient condition for set invariance in the presence of adversaries:

$$\begin{aligned} &\sum_{j \in \mathcal{A}} \left(L_{f_j} h^{x_j}(\vec{x}) + L_{g_j} h^{x_j}(\vec{x}) u_j^{\max} + L_{\phi_j} h^{x_j}(\vec{x}) \right) + \\ &\sum_{i \in \mathcal{N}} \left(L_{f_i} h^{x_i}(\vec{x}) + L_{g_i} h^{x_i}(\vec{x}) u_i + L_{\phi_i} h^{x_i}(\vec{x}) \right) \leq -\alpha(h(\vec{x})). \end{aligned} \tag{5.11}$$

Again, the equation (5.11) being satisfied for all $t \geq 0$ is equivalent to $\dot{h}(\vec{x}(t)) \leq \alpha(h(\vec{x}(t)))$ being satisfied for all $t \geq 0$ which implies forward invariance of the set S. The form of (5.11) reflects sampled-data adversarial agents seeking to violate the set invariance condition in (5.9) by maximizing their individual contributions to the LHS sum. The problem considered in this chapter is for the normal agents to compute control inputs that render the set S forward invariant using the sufficient condition in (5.11) despite the worst-case behavior of the adversarial agents in $\mathcal{A}$.

Problem 5.1. *Determine control inputs for the normal agents $i \in \mathcal{V}$ which render the set S forward invariant under the perturbed sampled-data dynamics* (5.7) *in the presence of a set of worst-case adversarial agents $\mathcal{A}$.*

Remark 5.3. *Since faulty or adversarial agents' states are generally modeled as being uncontrollable under the nominal system control law, the function $h(\vec{x})$ can be defined to consider only the safety of normal agents.*

Remark 5.4. *This chapter assumes the identities of the adversarial agents are known to the normal agents. Methods for identifying misbehavior are beyond the scope of this chapter.*

5.4 Safe Set Functions with Relative Degree 1

We first present results for safe set functions h where the control inputs u_i for all agents appear simultaneously in the expression for the first time derivative $\dot{h}(\vec{x}(t))$. Such functions are said to have relative degree 1 with respect to the system dynamics (5.7). Functions with relative degree strictly greater than 1 are considered in Section 5.5.

5.4.1 Preliminaries

The results of this subsection will be needed for our later analysis. The minimum and maximum value functions $\gamma_i^{\min}(\cdot)$, $\gamma_i^{\max}(\cdot)$ for $i \in \mathcal{V}$ are defined as follows:

$$\begin{aligned} \gamma_i^{\min}(\vec{x}) &= \min_{u_i \in \mathcal{U}_i} \left[L_{f_i} h^{x_i}(\vec{x}) + L_{g_i} h^{x_i}(\vec{x}) u_i \right], \\ \gamma_i^{\max}(\vec{x}) &= \max_{u_i \in \mathcal{U}_i} \left[L_{f_i} h^{x_i}(\vec{x}) + L_{g_i} h^{x_i}(\vec{x}) u_i \right]. \end{aligned} \tag{5.12}$$

Each $\gamma_i^{\min}(\vec{x})$ and $\gamma_i^{\max}(\vec{x})$ can be calculated by solving a parametric linear program

$$\min_{u_i \in \mathbb{R}^{m_i}} \quad c(\vec{x})^T u_i \quad \text{s.t.} \quad A_i(\vec{x}) u_i \leq b_i(\vec{x}), \tag{5.13}$$

where the vector $c(\vec{x})^T = L_{g_i} h^{x_i}(\vec{x})$ when calculating $\gamma_i^{\min}$ and $c(\vec{x})^T = -L_{g_i} h^{x_i}(\vec{x})$ when calculating $\gamma_i^{\max}$. Note that (5.13) is feasible for all $\vec{x} \in S$ under Assumption 5.3.

For an adversarial $j \in \mathcal{A}$, the function $\gamma_j^{\max}(\cdot)$ represents the bound on the worst-case contribution of j to the sum on the LHS of (5.11). Similarly, the function $\gamma_i^{\min}(\cdot)$ for a normal agent $i \in \mathcal{N}$ represents the bound on agent i's best control effort towards minimizing the LHS of (5.11).

Remark 5.5. *Note that for any $j \in \mathcal{A}$, for all $u_j \in \mathcal{U}_j$ it holds that*

$$L_{f_j} h^{x_j}(\vec{x}) + L_{g_j} h^{x_j}(\vec{x}) u_j \leq \gamma_j^{\max}(\vec{x}), \quad \forall \vec{x} \in \mathbb{R}^{\bar{n}}. \tag{5.14}$$

Due to this property, it will be demonstrated later in this paper that the results obtained by considering $\gamma_i^{\max}$ will hold for any $u_j \in \mathcal{U}_j$ for all $j \in \mathcal{A}$.

The following result presents a sufficient condition under which $\gamma_i^{\min}(\cdot)$ and $\gamma_i^{\max}(\cdot)$ are locally Lipschitz on the set S.

Lemma 5.1. *If the interior of $\mathcal{U}_i(\vec{x})$ is nonempty for all $\vec{x} \in S$ and $\mathcal{U}_i(\vec{x})$ is uniformly compact near $\vec{x}$ for all $\vec{x} \in S$, then the functions $\gamma_i^{\min}(\cdot)$ and $\gamma_i^{\max}(\cdot)$ defined by (5.12) are locally Lipschitz on S.*

Proof. The proofs for $\gamma_i^{\min}(\cdot)$ and $\gamma_i^{\max}(\cdot)$ are identical except for trivially changing the sign of the objective function; therefore only the proof for $\gamma_i^{\min}(\cdot)$ is given. Define the set of optimal points

$$P_i(\vec{x}) = \left\{ u_i^* : u_i^* = \arg\min_{u \in \mathcal{U}_i} L_{f_i} h^{x_i}(\vec{x}) + L_{g_i} h^{x_i}(\vec{x}) u \right\}.$$

The result in [269, Thm. 5.1] states that if $\mathcal{U}_i(\vec{x})$ is nonempty and uniformly compact near $\vec{x} \in \mathbb{R}^{\bar{n}}$ and if the Mangasarian-Fromovitz (M-F) conditions hold at each $u_i^* \in P_i(\vec{x})$, then $\gamma_i^{\min}(\cdot)$ is locally Lipschitz near $\vec{x}$ (see [269] for the definition of the M-F conditions). The first two conditions hold by assumption, and so we next prove that the M-F conditions hold at each $u_i^* \in P(\vec{x})$. Let $A_{i,j}(\cdot)$ denote the jth row of $A_i(\cdot)$ and $b_{i,j}(\cdot)$ denote the jth entry of $b_i(\cdot)$.

Consider any $\vec{x} \in S$ and $u_i^* \in P_i(\vec{x})$. Define the set

$$J_i(\vec{x}) = \{ j \in \{1, \dots, q_i\} : A_{i,j}(\vec{x}) u_i^* - b_{i,j}(\vec{x}) = 0 \}.$$

In words, $J_i(\vec{x})$ is the set of constraint indices where equality holds at u_i^*. Note that by definition of $J_i(\vec{x})$, for all $j' \notin J_i(\vec{x})$ it holds that $A_{i,j'}(\vec{x}) < 0$. The interior $\text{int}\,(\mathcal{U}_i(\vec{x}))$ being nonempty and convex implies there exists an $r \in \mathbb{R}^{m_i}$ such that for all $j \in J_i(\vec{x})$,

$$\begin{aligned} & A_{i,j}(\vec{x}) \left(u_i^* + r\right) - b_{i,j}(\vec{x}) < 0, \\ \implies & A_{i,j}(\vec{x}) r < b_{i,j}(\vec{x}) - A_{i,j}(\vec{x}) u_i^* = 0. \end{aligned} \tag{5.15}$$

This implies that there exists an r such that $A_i(\vec{x}) r < 0$. The point u_i^* is therefore M-F regular. Since this holds for any $u_i^* \in P_i(\vec{x})$ and $\forall \vec{x} \in S$, by [269, Thm. 5.1] it holds that $\gamma_i^{\min}(\cdot)$ is locally Lipschitz on S. □

We briefly emphasize the difference between the min / max ***value*** functions $\gamma_i^{\min}, \gamma_i^{\max}$ in (5.12) and the min / max ***point*** functions defined as

$$u_i^{\min}(\vec{x}) = \arg\min_{u_i \in \mathcal{U}_i} \left[L_{f_i} h^{x_i}(\vec{x}) + L_{g_i} h^{x_i}(\vec{x}) u_i \right], \tag{5.16}$$

$$u_i^{\max}(\vec{x}) = \arg\max_{u_i \in \mathcal{U}_i} \left[L_{f_i} h^{x_i}(\vec{x}) + L_{g_i} h^{x_i}(\vec{x}) u_i \right]. \tag{5.17}$$

In words, $u_i^{\min}$ and $u_i^{\max}$ represent the control actions such that, respectively, $\gamma_i^{\min}(\vec{x}) = L_{f_i} h^{x_i}(\vec{x}) + L_{g_i} h^{x_i}(\vec{x}) u_i^{\min}$ and $\gamma_i^{\max}(\vec{x}) = L_{f_i} h^{x_i}(\vec{x}) + L_{g_i} h^{x_i}(\vec{x}) u_i^{\max}$. Although the min / max ***value*** functions $\gamma_i^{\min}(\cdot), \gamma_i^{\max}(\cdot)$ are locally Lipschitz under the conditions of Lemma 5.1 and [269], the min / max ***point*** functions $u_i^{\min}$ and $u_i^{\max}$ may *not* be locally Lipschitz in general.[2]

[2]We re-emphasize however that when (5.17) is applied in a ZOH manner, existence and uniqueness of solutions to (5.7) is guaranteed by Carathéodory's theorem [267, Sec. 2.2].

The following Lemma is based on results in [176], [209, Thm. 3.4]. It establishes an upper bound on the difference between the sampled state $\vec{x}^{k_i}$ and the state $\vec{x}(t)$ on a time interval $t \in [t_i^k, t_i^k + \Gamma)$, $\Gamma \geq 0$.

Lemma 5.2. *For any $\Gamma \geq 0$, there exists a $\mu \geq 0$, $L' > 0$ such that the following holds:*

$$\left\|\vec{x}(t) - \vec{x}^{k_i}\right\| \leq \frac{\mu}{L'}\left(e^{L'\Gamma} - 1\right) \ \forall t \in [t_i^k, t_i^k + \Gamma).$$

Proof. Using the same method as [209, Thm. 3.4], define the functions

$$\mathfrak{f}(t, \vec{x}) = 0, \tag{5.18}$$

$$\mathfrak{g}(t, \vec{x}) = \begin{bmatrix} f_1(x_1) + g_1(x_1)u_1(t) + \phi_1(t) \\ \vdots \\ f_N(x_N) + g_N(x_N)u_N(t) + \phi_N(t) \end{bmatrix} \tag{5.19}$$

Next, observe that

$$\frac{d}{dt}\vec{x}^{k_i} = 0 = \mathfrak{f}(t, \vec{x}^{k_i}),$$
$$\frac{d}{dt}\vec{x}(t) = \mathfrak{f}(t, \vec{x}) + \mathfrak{g}(t, \vec{x}).$$

Observe that S is compact by Assumption 5.2, each f_i, g_i is locally Lipschitz, and each $\phi_i(t)$ is locally Lipschitz with $\|\phi_i(t)\| \leq \phi_i^{\max}$. In addition, by Assumption 5.3, there exists an upper bound $u_M \in \mathbb{R}$ such that $\|u_l\| \leq u_M$. Therefore there exists $\mu \in \mathbb{R} \geq 0$ such that

$$\sup_{\vec{x} \in S} \left\| \begin{bmatrix} f_1(x_1) + g_1(x_1)u_1 + \phi_1(t) \\ \vdots \\ f_N(x_N) + g_N(x_N)u_N + \phi_N(t) \end{bmatrix} \right\| \leq \mu. \tag{5.20}$$

Note that for $t = t_i^k$ we have $\left\|\vec{x}(t) - \vec{x}^{k_i}(t)\right\| = 0$. Therefore by [209, Thm. 3.4], it holds that

$$\left\|\vec{x} - \vec{x}^{k_i}\right\| \leq \frac{\mu}{L'}\left(e^{L'(t - t_i^k)} - 1\right), \quad \forall t \in [t_i^k, t_i^k + \Gamma), \tag{5.21}$$

where $L' \in \mathbb{R}_{>0}$ is any strictly positive constant.

□

For brevity, we define the function $\epsilon : \mathbb{R} \times \mathbb{R} \times \mathbb{R}_{>0} \to \mathbb{R}$ as

$$\epsilon(\Gamma, \mu, L') = \frac{\mu}{L'} \left(e^{L'\Gamma} - 1 \right). \tag{5.22}$$

For fixed μ, L', we abuse notation by writing $\epsilon(\Gamma)$ as a function of Γ only. It can be shown that for fixed μ, L', $\epsilon(\cdot)$ is a class-$\mathcal{K}$ function in Γ.

5.4.2 Synchronous Sampling Times

To facilitate the presentation of the main results, we first consider the case where all agents in the system have synchronous sampling times with a period of Γ, i.e., $\mathcal{T}_i = \{k\Gamma : k \in Z_{\geq 0}\}$ $\forall i \in \mathcal{N}$. This assumption is later relaxed to consider agents with asynchronous, nonidentical sampling times. The Cartesian product of the admissible controls for all normal agents is denoted $\mathcal{U}_{\mathcal{N}} = \bigtimes_{i \in \mathcal{N}} \mathcal{U}_i$. Under Assumption 5.3, each $\mathcal{U}_i(\vec{x})$ being uniformly compact near all $\vec{x} \in S$ implies that $\mathcal{U}_{\mathcal{N}}$ is also uniformly compact near all $\vec{x} \in S$. We will denote $\vec{u}_{\mathcal{N}} \in \mathcal{U}_{\mathcal{N}}$ as the vector containing only normal agents' control inputs; i.e., $\vec{u}_{\mathcal{N}} = \begin{bmatrix} u_{i_1}^T & \dots & u_{i_{|\mathcal{N}|}}^T \end{bmatrix}^T$, $\{i_1, \dots, i_{|\mathcal{N}|}\} \in \mathcal{N}$.

Our ultimate aim is to demonstrate that for all $t \geq 0$,

$$\dot{h}(\vec{x}(t)) + \alpha(h(\vec{x}(t))) \leq 0. \tag{5.23}$$

The dependence of $\vec{x}(t)$ on t will be omitted for brevity. Prior results have typically focused on designing continuous $u(\cdot)$ functions that guarantee that (5.23) is satisfied. Satisfying (5.23) in sampled-data systems for all intermediate times $t \in [t_k, t_{k+1})$, $k \in \mathbb{Z}_{\geq 0}$ is more challenging since $u(\cdot)$ is constant on each interval $t \in [k\Gamma, (k+1)\Gamma)$. Inspired by [176], this challenge will be addressed as follows: given the sampled state $\vec{x}(t^k)$ and the state $\vec{x}(t)$, $t \in [t^k, t^{k+1})$, define the error term

$$e(t, t^k) = \left(\dot{h}(\vec{x}) - \dot{h}(\vec{x}^k) \right) + \left(\alpha(h(\vec{x})) - \alpha(h(\vec{x}^k)) \right).$$

From the LHS of (5.23) we obtain

$$\begin{aligned} \dot{h}(\vec{x}) + \alpha(h(\vec{x})) &= \dot{h}(\vec{x}^k) + \left(\dot{h}(\vec{x}) - \dot{h}(\vec{x}^k) \right) + \alpha(h(\vec{x}^k)) + \left(\alpha(h(\vec{x})) - \alpha(h(\vec{x}^k)) \right), \\ &= \dot{h}(\vec{x}^k) + \alpha(h(\vec{x}^k)) + e(t, t^k), \\ &\leq \dot{h}(\vec{x}^k) + \alpha(h(\vec{x}^k)) + \sup_{t \in [t^k, t^{k+1})} \left\| e(t, t^k) \right\|. \end{aligned}$$

By defining a function $\eta(\cdot)$ such that $\eta(\Gamma) \geq \sup_{t \in [t^k, t^{k+1})} \left\| e(t, t^k) \right\|$, the inequality condition in (5.23) is therefore satisfied for all times on the interval $t \in [t^k, t^{k+1})$ if for every $t^k \in \mathcal{T}$ the

following condition holds:

$$\dot{h}(\vec{x}^k) + \alpha(h(\vec{x}^k)) + \eta(\Gamma) \leq 0. \tag{5.24}$$

Satisfaction of (5.24) implies that $\dot{h}(\vec{x}) + \alpha(h(\vec{x})) \leq \dot{h}(\vec{x}^k) + \alpha(h(\vec{x}^k)) + \eta(\Gamma) \leq 0$ for all $t \in [t^k, t^{k+1})$. To define such a function $\eta(\cdot)$, the following Lemma will be used.

Lemma 5.3. *Consider the system (5.7). There exist constants $c_f, c_g, c_\alpha, c_\gamma, c_h \in \mathbb{R}$ such that for all $\vec{x}^1, \vec{x}^2 \in S$, all of the following inequalities hold:*

$$\sum_{i \in \mathcal{N}} \left\| L_{f_i} h^{x_i}(\vec{x}^1) - L_{f_i} h^{x_i}(\vec{x}^2) \right\| \leq c_f \left\| \vec{x}^1 - \vec{x}^2 \right\|, \tag{5.25}$$

$$\sum_{i \in \mathcal{N}} \left\| L_{g_i} h^{x_i}(\vec{x}^1) - L_{g_i} h^{x_i}(\vec{x}^2) \right\| \leq c_g \left\| \vec{x}^1 - \vec{x}^2 \right\|, \tag{5.26}$$

$$\left\| \alpha(h(\vec{x}^1)) - \alpha(h(\vec{x}^2)) \right\| \leq c_\alpha \left\| \vec{x}^1 - \vec{x}^2 \right\|, \tag{5.27}$$

$$\sum_{j \in \mathcal{A}} \left\| \gamma_j^{\max}(\vec{x}^1) - \gamma_j^{\max}(\vec{x}^2) \right\| \leq c_\gamma \left\| \vec{x}^1 - \vec{x}^2 \right\|, \tag{5.28}$$

$$\left\| \sum_{l \in \mathcal{V}} L_{\phi_l} h^{x_l}(\vec{x}^1) \right\| \leq c_h \sum_{l \in \mathcal{V}} \phi_l^{\max} \tag{5.29}$$

Proof. The inequalities (5.25)-(5.28) follow from the fact that each f_i, g_i, α, and $\gamma_j^{\max}$ are locally Lipschitz, $h \in C^{1,1}_{loc}$, and S is compact. To demonstrate that (5.29) holds, observe that S being compact and $h \in C^{1,1}_{loc}$ implies $\left\| \frac{\partial h(\vec{x})}{\partial x_l} \right\|$ is bounded on S for all $l \in \mathcal{V}$. Therefore, there exists a constant $c_h \in \mathbb{R}_{>0}$ such that for all $t \in [t^k, t^{k+1})$, $\forall \vec{x} \in S$,

$$\left\| \sum_{l \in \mathcal{V}} L_{\phi_l} h^{x_l}(\vec{x}^1) \right\| \leq \sum_{l \in \mathcal{V}} \left\| \frac{\partial h(\vec{x}^1)}{\partial x_l} \phi_l(t) \right\| \leq c_h \sum_{l \in \mathcal{V}} \phi_l^{\max}$$

□

In addition to the inequalities in Lemma 5.3, observe that each set $\mathcal{U}_i$ being uniformly compact implies that there exist a constant $u_{\max} \geq 0$ such that $\left\| u_i^k \right\| \leq u_{\max}$ for all $i \in \mathcal{N}$, $k \geq 0$. Using this definition of $u_{\max}$, the constants defined in Lemma 5.3, and the function $\epsilon\cdot$ in (5.22) we define the function $\eta : \mathbb{R}_{\geq 0} \to \mathbb{R}$ as follows:

$$\eta(\Gamma) = (c_f + c_g u_{\max} + c_\alpha + c_\gamma)\, \epsilon(\Gamma) + c_h \sum_{l \in \mathcal{V}} \phi_l^{\max}. \tag{5.30}$$

The proof that $\eta(\Gamma) \geq \sup_{t\in[t^k,t^{k+1})} \left\|e(t,t^k)\right\|$ will be given in Theorem 5.2. This definition of $\eta(\cdot)$ is used to define the following *safety-preserving controls set* for the normal agents in $\mathcal{N}$:

$$K(\vec{x}) = \left\{\vec{u}^k_{\mathcal{N}} \in \mathcal{U}_{\mathcal{N}} : \sum_{i\in\mathcal{N}} \left[L_{f_i} h^{x_i}(\vec{x}^k) + L_{g_i} h^{x_i}(\vec{x}^k) u^k_i\right] + \sum_{j\in\mathcal{A}} \gamma^{\max}_j(\vec{x}^k) + \alpha(h(\vec{x}^k)) + \eta(\Gamma) \leq 0\right\} \tag{5.31}$$

Using this definition of $K(\cdot)$, the following Theorem presents conditions under which the set S can be rendered forward invariant for the system (5.7) with synchronous sampling times despite the actions of the adversarial agents.

Theorem 5.2. *Consider the system (5.7) with synchronous sampling times. If $\vec{x}^k \in S$ for $k \geq 0$, then for any control input $\vec{u}^k \in K(\vec{x}^k)$ the trajectory $\vec{x}(t)$ satisfies $\vec{x}(t) \in S$ for all $t \in [k\Gamma, (k+1)\Gamma)$.*

Proof. First, denote $\dot{h}'(\vec{x}^k) = \dot{h}(\vec{x}^k) - \sum_{l\in\mathcal{V}} L_{\phi_l} h^{x_l}(\vec{x}^k)$. In words, $\dot{h}'(\vec{x}^k)$ is equal to $\dot{h}(\vec{x}^k)$ with all disturbance-related Lie derivatives subtracted out. Observe that

$$\begin{aligned}
\dot{h}(\vec{x}^k) + \left(\dot{h}(\vec{x}) - \dot{h}(\vec{x}^k)\right) &= \dot{h}'(\vec{x}^k) + \sum_{l\in\mathcal{V}} L_{\phi_l} h^{x_l}(\vec{x}^k) + \left(\dot{h}(\vec{x}) - \dot{h}'(\vec{x}^k) - \sum_{l\in\mathcal{V}} L_{\phi_l} h^{x_l}(\vec{x}^k)\right), \\
&= \dot{h}'(\vec{x}^k) + \left(\dot{h}(\vec{x}) - \dot{h}'(\vec{x}^k)\right), \\
&= \dot{h}'(\vec{x}^k) + \left(\dot{h}'(\vec{x}) - \dot{h}'(\vec{x}^k)\right) + \sum_{l\in\mathcal{V}} L_{\phi_l} h^{x_l}(\vec{x}).
\end{aligned}$$

From (5.7) and the definition of adversarial agents in (5.10), define the error term

$$\begin{aligned}
e'(t,t^k) &= \left(\dot{h}'(\vec{x}) - \dot{h}'(\vec{x}^k)\right) + \sum_{l\in\mathcal{V}} L_{\phi_l} h^{x_l}(\vec{x}) + \left(\alpha(h(\vec{x})) - \alpha(h(\vec{x}^k))\right), \\
&= \left(\sum_{i\in\mathcal{N}} L_{f_i} h^{x_i}(\vec{x}) - L_{f_i} h(\vec{x}^k)\right) + \sum_{l\in\mathcal{V}} L_{\phi_l} h^{x_l}(\vec{x}) + \left(\sum_{i\in\mathcal{N}} \left[L_{g_i} h^{x_i}(\vec{x}) - L_{g_i} h^{x_i}(\vec{x}^k)\right] u^k_i\right) + \\
&\quad \sum_{j\in\mathcal{A}} \left(\gamma^{\max}_j(\vec{x}) - \gamma^{\max}_j(\vec{x}^k)\right) + \left(\alpha(h(\vec{x})) - \alpha(h(\vec{x}^k))\right)
\end{aligned}$$

Since $t^{k+1} - t^k = (k+1)\Gamma - k\Gamma = \Gamma$ for all $k \geq 0$, by Lemma 5.2 we have $\left\|\vec{x} - \vec{x}^k\right\| \leq \epsilon(\Gamma)$ for all $t \in [t^k, t^{k+1})$. Using Lemma 5.3 and the definition of $\eta(\cdot)$ in (5.30) yields the following upper

bound on $\left\|e'(t,t^k)\right\|$:

$$\begin{aligned}&\sup_{t\in[t^k,t^{k+1})}\left\|e'(t,t^k)\right\| \leq (c_f + c_g u_{\max} + c_\alpha + c_\gamma)\,\epsilon(\Gamma) + c_h \sum_{l\in\mathcal{V}} \phi_l^{\max},\\ \implies &\sup_{t\in[t^k,t^{k+1})}\left\|e'(t,t^k)\right\| \leq \eta(\Gamma).\end{aligned}$$

Therefore for all $t \in [t^k, t^{k+1})$, it holds that

$$\begin{aligned}\dot{h}(\vec{x}) + \alpha(h(\vec{x})) &= \dot{h}'(\vec{x}^k) + \left(\dot{h}'(\vec{x}) - \dot{h}'(\vec{x}^k)\right) + \sum_{l\in\mathcal{V}} L_{\phi_l} h^{x_l}(\vec{x}) + \alpha(h(\vec{x}^k))\\ &\quad + \left(\alpha(h(\vec{x})) - \alpha(h(\vec{x}^k))\right),\\ &= \dot{h}'(\vec{x}^k) + \alpha(h(\vec{x}^k)) + e'(t,t^k),\\ &\leq \dot{h}'(\vec{x}^k) + \alpha(h(\vec{x}^k)) + \sup_{t\in[t^k,t^{k+1})}\left\|e'(t,t^k)\right\|\\ &\leq \sum_{i\in\mathcal{N}}\left[L_{f_i}h^{x_i}(\vec{x}^k) + L_{g_i}h^{x_i}(\vec{x}^k)u_i^k\right] + \sum_{j\in\mathcal{A}}\gamma_j^{\max}(\vec{x}^k) + \alpha(h(\vec{x}^k)) + \eta(\Gamma).\end{aligned}$$

Choosing any $\vec{u}_{\mathcal{N}}^k \in K(\vec{x})$, observe from (5.31) that

$$\sum_{i\in\mathcal{N}}\left[L_{f_i}h^{x_i}(\vec{x}^k) + L_{g_i}h^{x_i}(\vec{x}^k)u_i^k\right] + \sum_{j\in\mathcal{A}}\gamma_j^{\max}(\vec{x}^k) + \alpha(h(\vec{x}^k) + \eta(\Gamma) \leq 0, \tag{5.32}$$

$$\implies \dot{h}(\vec{x}) + \alpha(h(\vec{x})) \leq 0. \tag{5.33}$$

Therefore any $\vec{u}_{\mathcal{N}}^k \in K(\vec{x}^k)$ renders the set S forward invariant for all $t \in [t^k, t^{k+1})$. These arguments hold for all $k \in \mathbb{Z}_{\geq 0}$, which concludes the proof. □

Remark 5.6. *Using Remark 5.5 observe that given any $\vec{u}_{\mathcal{N}}^k \in K(\vec{x}^k)$, for all $u_j \in \mathcal{U}_j$, $j \in \mathcal{A}$ it holds that*

$$\begin{aligned}&\sum_{i\in\mathcal{N}}\left(L_{f_i}h^{x_i}(\vec{x}^k) + L_{g_i}h^{x_i}(\vec{x}^k)u_i\right) + \sum_{j\in\mathcal{A}}\left(L_{f_j}h^{x_j}(\vec{x}^k) + L_{g_j}h^{x_j}(\vec{x}^k)u_j\right) + \alpha(h(\vec{x})) + \eta(\Gamma) \leq\\ &\quad \sum_{i\in\mathcal{N}}\left(L_{f_i}h^{x_i}(\vec{x}^k) + L_{g_i}h^{x_i}(\vec{x}^k)u_i\right) + \sum_{j\in\mathcal{A}}\gamma_j^{\max}(\vec{x}^k) + \alpha(h(\vec{x})) + \eta(\Gamma).\end{aligned}$$

Therefore the results of Theorem 5.2 hold for any feasible control inputs $u_j \in \mathcal{U}_j$ of any agent $j \in \mathcal{A}$.

In other words, since the analysis of Theorem 5.2 uses the maximum upper bounds $\gamma_j^{\max}(\cdot)$ on the contributions of the adversarial agents $j \in \mathcal{A}$ to the LHS of the safety condition (5.9), the results of Theorem 5.2 hold for any feasible control inputs $u_j \in \mathcal{U}_j$ of any agent $j \in \mathcal{A}$. In this

sense the results of Theorem 5.2 can be applied to a broader definition of the set $\mathcal{A}$ than the one given in Section 5.3.2.

When $K(\vec{x})$ defined in (5.31) is nonempty, a feasible $\vec{u}^*_{\mathcal{N}} \in K(\vec{x})$ rendering S invariant while minimally modifying $\vec{u}_{\text{nom}}$ can be computed by solving the following QP:

$$\begin{aligned} \vec{u}^*_{\mathcal{N}}(\vec{x}^k) = \underset{\vec{u}_{\mathcal{N}} \in \mathcal{U}_{\mathcal{N}}}{\arg\min} \; & \|\vec{u}_{\mathcal{N}} - \vec{u}_{\text{nom}}\|_2^2 \\ \text{s.t.} \quad & \sum_{i \in \mathcal{N}} \left(L_{f_i} h^{x_i}(\vec{x}^k) + L_{g_i} h^{x_i}(\vec{x}^k) u_i \right) + \sum_{j \in \mathcal{A}} \gamma_j^{\max}(\vec{x}^k) + \alpha(h(\vec{x})) + \eta(\Gamma) \leq 0 \end{aligned} \tag{5.34}$$

Note that this QP requires the values of $\gamma_j^{\max}(\vec{x}^k)$, $j \in \mathcal{A}$, which can be solved for via a separate LP. Once $\vec{u}^*_{\mathcal{N}}(\vec{x}^k) \in K(\vec{x})$ has been obtained, each agent $i \in \mathcal{N}$ can then apply the local control input $u_i(\vec{x}^k)$. By Theorem 5.2, safety of the entire system is guaranteed under the adversarial behavior for all forward time. The case when $K(\vec{x})$ is empty is discussed in Section 5.4.4.

5.4.3 Asynchronous Sampling Times

The assumption of identical, synchronous sampling times typically does not hold in practice. In addition, a distributed system may not have access to a centralized entity to solve the QP in (5.34) to obtain $\vec{u}_{\mathcal{N}}$. This subsection will therefore consider asynchronous sampling times and a distributed method for computing local control inputs. Each agent $i \in \mathcal{V}$ is assumed to have a nominal sampling period $\Gamma_i \in \mathbb{R}_{>0}$ and the perturbed sequence of sampling times

$$\mathcal{T}_i = \{t_i^0, t_i^1, \ldots\} \text{ s.t. } t_i^{k+1} - t_i^k = \Gamma_i + \delta_i(k), \; \forall k \in \mathbb{Z}_{\geq 0}, \tag{5.35}$$

where $\delta_i(k)$ is a disturbance satisfying $\|\delta_i(k)\| \leq \delta_i^{\max}$. The function δ_i can be used to model time delays due to disturbances such as clock asynchrony or packet drops in the communication network. We denote $\Gamma^{\max} = \max_{i \in \mathcal{V}} \Gamma_i$ and $\delta^{\max} = \max_{i \in \mathcal{V}} \delta_i^{\max}$. Recall from Section 5.3.2 that we denote $\vec{x}^{k_i} = \vec{x}(t_i^k)$ and $u_i^{k_i} = u_i(t_i^k)$.

Each agent $i \in \mathcal{N}$ updates its control input $u_i^{k_i}$ at sampling times t_i^k and also broadcasts $u_i^{k_i}$ to all other agents in the network. Each agent i stores the values of the most recently received inputs from its normal in-neighbors $l \in \mathcal{N}$. The notation $\hat{u}_l^{k_i}$ denotes the most recently received input value by agent i from agent l at time t_i^k.

Using the definition of $\eta(\cdot)$ from (5.30) following feasible set is defined for each $i \in \mathcal{N}$:

$$K_i(\vec{x}^{k_i}) = \Big\{ u_i \in \mathcal{U}_i : L_{f_i} h^{x_i}(\vec{x}^{k_i}) + L_{g_i} h^{x_i}(\vec{x}^{k_i}) u_i + \sum_{l \in \mathcal{N} \setminus \{i\}} \left[L_{f_l} h^{x_l}(\vec{x}^{k_i}) + L_{g_l} h^{x_l}(\vec{x}^{k_i}) \hat{u}_l^{k_i} \right] + \sum_{j \in \mathcal{A}} \gamma_j^{\max}(\vec{x}^{k_i}) + \alpha(h(\vec{x}^{k_i})) + \eta(\Gamma_i + \delta^{\max}) \leq 0 \Big\}$$

Theorem 5.3 presents conditions under which forward invariance of the set S can be guaranteed for the distributed, asynchronous system described in this subsection.

Theorem 5.3. *Consider the system (5.7) with sampling times described by (5.35). If at sampling time t_i^k for $k \geq 0$, $i \in \mathcal{N}$ it holds that $\vec{x}^{k_i} \in S$, then for any $u_i^{k_i} \in K_i(\vec{x}^{k_i})$ the trajectory $\vec{x}(t)$ satisfies $\vec{x}(t) \in S$ for all $t \in [t_i^k, t_i^{k+1})$.*

Proof. Choose any $i \in \mathcal{N}$ and consider the time interval $t \in [t_i^k, t_i^{k+1})$. Recall that $t_i^{k+1} - t_i^k \leq \Gamma_i + \delta^{\max}$ $\forall k \in \mathbb{Z}_{\geq 0}$ by virtue of (5.35) and the definition of $\delta^{\max}$. In particular, this implies $\epsilon(\Gamma_i + \delta_i(k)) \leq \epsilon(\Gamma_i + \delta^{\max})$ for all $k \in \mathbb{Z}_{\geq 0}$ since $\epsilon(\cdot)$ is a class-$\mathcal{K}$ function in Γ. For each $i \in \mathcal{N}$ define the value $e_i'(t, t^k)$ in a similar manner as Theorem 5.2 and observe

$$\sup_{t \in [t_i^k, t_i^{k+1})} \left\| e_i'(t, t_i^k) \right\| \leq (c_f + c_g u_{\max} + c_\alpha + c_\gamma)\, \epsilon(\Gamma_i + \delta^{\max}) + c_h \sum_{l \in \mathcal{V}} \phi_l^{\max},$$
$$\implies \sup_{t \in [t_i^k, t_i^{k+1})} \left\| e_i'(t, t_i^k) \right\| \leq \eta(\Gamma_i + \delta^{\max})$$

The same logic as in Theorem 5.2 can then be used to demonstrate that $\dot{h}(\vec{x}) + \alpha(h(\vec{x})) \leq 0$ for all $t \in [t_i^k, t_i^{k+1})$. □

Under the communication protocol described previously, each agent can use the most recently received inputs $\hat{u}_l^{k_i}$ from other normal agents to calculate a control input $u_i^{k_i} \in K_i(\vec{x}^{k_i})$. Such a $u_i^{k_i}$ can be computed by solving the following QP:

$$u_i(\vec{x}^{k_i}) = \arg\min_{u_i \in \mathcal{U}_i} \left\| u_i - u_{i,\text{nom}}^{k_i} \right\|_2^2 \tag{5.36}$$
$$\text{s.t.} \quad \left(L_{f_i} h^{x_i}(\vec{x}^{k_i}) + L_{g_i} h^{x_i}(\vec{x}^{k_i}) u_i \right) + \sum_{l \in \mathcal{N} \setminus \{i\}} \left(L_{f_l} h^{x_l}(\vec{x}^{k_i}) + L_{g_l} h^{x_l}(\vec{x}^{k_i}) \hat{u}_l^{k_i} \right) + \sum_{j \in \mathcal{A}} \gamma_j^{\max}(\vec{x}^{k_i}) + \alpha(h(\vec{x}^{k_i})) + \eta(\Gamma_i + \delta^{\max}) \leq 0.$$

Like the previous formulations, the values of $\gamma_j^{\max}(\cdot)$ for $j \in \mathcal{A}$ can be calculated via solving a separate LP. By the results of Theorem 5.3, when each $K_i(\vec{x})$ is nonempty and each normal

agent applies the controller defined by (5.36) the multi-agent safe set is rendered forward invariant despite any collective worst-case behavior of the adversarial agents.

5.4.4 Maximum Safety-Preserving Control Action

One of the required conditions of the foregoing results is the nonemptiness of the feasible sets $K(\vec{x})$ and $K_i(\vec{x})$, which is also closely related to the feasibility of the QP (5.34). Conditions under which such feasible sets remain nonempty for general systems remains an open question. Guaranteeing both safety and the feasibility of the QP calculating the control input $u_i(\vec{x}^{k_i})$ has been a recent topic of study [268, 270], and can depend on the choice of extended class-$\mathcal{K}$ function $\alpha(\cdot)$.

In contrast, consider the sampled-data control law $u_i^{\min}(\cdot)$ defined in (5.16). Intuitively speaking, (5.16) represents the strongest control effort agent $i \in \mathcal{N}$ can apply towards minimizing the LHS of (5.9). This control input can be solved for by taking the $\arg\min$ of the minimizing LP in (5.13):

$$\begin{aligned} u_i^{\min}(\vec{x}^{k_i}) = \underset{u_i \in \mathbb{R}^{m_i}}{\arg\min} \quad & L_{g_i} h^{x_i}(\vec{x}^{k_i}) u_i \\ \text{s.t.} \quad & A_i(\vec{x}^{k_i}) u_i \leq b_i(\vec{x}^{k_i}) \end{aligned} \tag{5.37}$$

For any system satisfying Assumption 3, the set $\mathcal{U}_i(\vec{x}) = \{u : A_i(\vec{x})u \leq b_i(\vec{x})\}$ is nonempty for all $\vec{x} \in S$. This implies that (5.37) is always guaranteed to be feasible for $\vec{x} \in S$. However the question remains as to when the control action (5.16) can guarantee forward invariance of S. Towards this end, define the set

$$\partial S_\epsilon = \left\{ x \in S : \min_{z \in \partial S} \|x - z\| \leq \epsilon \right\}, \quad \epsilon > 0. \tag{5.38}$$

In words, ∂S_ϵ is an "inner boundary region" of S that includes all points in S within distance ϵ of ∂S with respect to a chosen norm. An example is given in Figure 5.1.

The following theorem presents a sufficient condition for when the control $u_i^{\min}(\cdot)$ for each normal agent renders the set S invariant in the presence of an adversarial set $\mathcal{A}$.

Theorem 5.4. *Let $\epsilon^* = \epsilon(\Gamma^{\max} + 2\delta^{\max})$ and define the sets $\partial S_{\epsilon^*}, \partial S_{2\epsilon^*}$ as per (5.38). Suppose that each normal agent $i \in \mathcal{N}$ applies the control input $u_i^{\min}(\vec{x}^{k_i})$ from (5.16) for all sampled states $\vec{x}^{k_i}$ satisfying $\vec{x}^{k_i} \in \partial S_{2\epsilon^*}$. Then S is forward invariant if $\vec{x}(0) \in S \backslash \partial S_{2\epsilon^*}$ and the following condition holds:*

$$\max_{\vec{x} \in \partial S_{2\epsilon^*}} \left[\sum_{i \in \mathcal{N}} \max_{\vec{x}^i \in B(\vec{x}, \epsilon^*)} \left[\gamma_i^{\min}(\vec{x}^i) \right] + \sum_{j \in \mathcal{A}} \gamma_j^{\max}(\vec{x}) + \alpha(h(\vec{x})) \right] \leq -\eta(\Gamma^{\max} + 2\delta^{\max}). \tag{5.39}$$

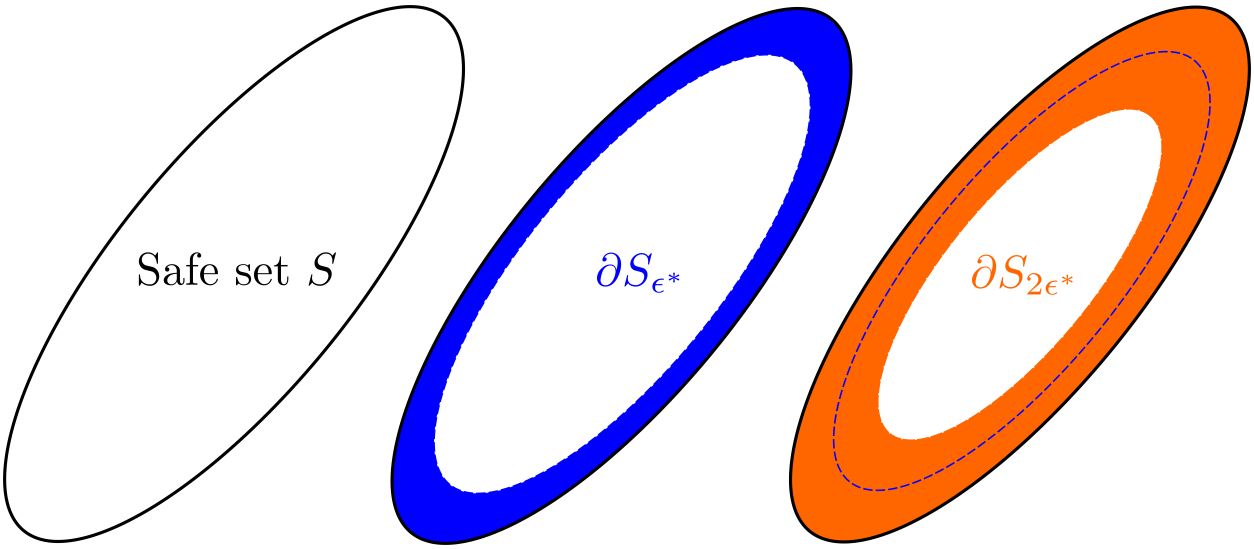


Figure 5.1: An example of the sets S, ∂S_{ϵ^*}, and $\partial S_{2\epsilon^*}$ for a given $\epsilon^* > 0$. Note that each of the three ellipses is a separate view of the same set S. The dotted blue line in the rightmost ellipse is the inner boundary of ∂S_{ϵ^*}, highlighting the fact that $\partial S_{\epsilon^*} \subset \partial S_{2\epsilon^*}$.

Proof. The proof first demonstrates that the most recently sampled states of all agents always lie within a closed ball of radius $\epsilon^* = \epsilon(\Gamma^{\max} + 2\delta^{\max})$. Next, it shows that $\vec{x}(0) \in S \backslash \partial S_{\epsilon^*}$ implies that $\vec{x}(t)$ cannot leave S without all agents sampling the state at least once within the region ∂S_{ϵ^*}. Finally, it is shown that this fact combined with (5.39) implies that S is forward invariant.

Choose any $i \in \mathcal{N}$ and any sampling time t_i^k for agent i. By the definition of $\Gamma^{\max}$ and $\delta^{\max}$, the next sampling time t_i^{k+1} satisfies $t_i^{k+1} - t_i^k \leq t_i^k + \Gamma^{\max} + 2\delta^{\max}$. Since this holds for all $i \in \mathcal{V}$, given any $i_1, i_2 \in \mathcal{N}$ and interval $[t_{i_1}^k, t_{i_1}^k + \Gamma^{\max} + \delta^{\max}]$, there exists a sampling time for i_2 satisfying $t_{i_2}^{k'} \in [t_{i_1}^k, t_{i_1}^k + \Gamma^{\max} + 2\delta^{\max}]$. Using Lemma 5.2, this implies that the maximum normed difference between any two most recently sampled states $\vec{x}(t_{i_1}^{k^*})$ and $\vec{x}(t_{i_2}^{k^*})$ satisfies $\left\|\vec{x}(t_{i_1}^{k^*}) - \vec{x}(t_{i_2}^{k^*})\right\| \leq \epsilon(\Gamma^{\max} + 2\delta^{\max}) = \epsilon^*$. Since this holds for all $i_1, i_2 \in \mathcal{V}$ at any $t_{i_1}^k$, the most recently sampled states of all agents therefore always lie within a ball of radius ϵ^*.

Next, consider any agent i with sampling time t_i^k such that $\vec{x}(t_i^k) \in S \backslash \partial S_{\epsilon^*}$ and $\vec{x}(t_i^{k+1}) \notin S \backslash \partial S_{\epsilon^*}$. Since $\left\|\vec{x}(t_i^{k+1}) - \vec{x}(t_i^k)\right\| \leq \epsilon^*$ by previous arguments, this implies that $\vec{x}(t_i^{k+1}) \in \partial S_{\epsilon^*}$. Therefore $\vec{x}(0) \in S \backslash \partial S_{\epsilon^*}$ implies that $\vec{x}$ cannot leave S without each agent $i \in \mathcal{N}$ having at least one sampling time t_i^k such that $\vec{x}^{k_i} \in \partial S_{\epsilon^*}$. Note that $\vec{x}(0) \in S \backslash \partial S_{2\epsilon^*}$ as per the Theorem statement implies that $\vec{x}(0) \in S \backslash \partial S_{\epsilon^*}$ since $\partial S_{\epsilon^*} \subset \partial S_{2\epsilon^*}$.

Define $e'(t, t_i^k)$ in a similar manner to Theorem 5.2. Observe that $t_i^{k+1} - t_i^k \leq \Gamma^{\max} + 2\delta^{\max}$ for all $i \in \mathcal{N}$. In addition, for any $i_1, i_2 \in \mathcal{N}$ with most recent sampling times $t_{i_1}^{k_{i_1}}$ and $t_{i_2}^{k_{i_2}}$, it can be shown that $|t_{i_1}^{k_{i_1}} - t_{i_2}^{k_{i_2}}| \leq \Gamma^{\max} + 2\delta^{\max}$. Therefore on any interval $t \in [t_{i_1}^{k_{i_1}}, t_{i_2}^{k_{i_2}})$, we have

$$\sup_{t\in[t_{i_1}^{k_{i_1}},t_{i_2}^{k_{i_2}})}\left\|e'(t,t_{i_1}^{k_{i_1}})\right\|\leq$$
$$(c_f+c_g u_{\max}+c_\alpha+c_\gamma)\,\epsilon^*+c_h\sum_{l\in\mathcal{V}}\phi_l^{\max},$$
$$\implies \sup_{t\in[t_{i_1}^{k_{i_1}},t_{i_2}^{k_{i_2}})}\left\|e'(t,t_{i_1}^{k_{i_1}})\right\|\leq\eta(\Gamma^{\max}+2\delta^{\max}).$$

Choose the first sampling time $t_{i_1}^{k_{i_1}}$ such that $t_{i_1}^{k_{i_1}}\geq\Gamma^{\max}+2\delta^{\max}$ and $\vec{x}^{k_{i_1}}\in\partial S_{\epsilon^*}\subset\partial S_{2\epsilon^*}$. Since $\vec{x}(0)\in S_{2\epsilon^*}$ by the Theorem statement, it can be shown using prior arguments that such a sampling time is guaranteed to exist. This choice of $t_{i_1}^{k_{i_1}}$ implies that all agents have sampled at least once at or before $t_{i_1}^{k_{i_1}}$. Let $t_{i_2}^{k_{i_2}}>t_{i_1}^{k_{i_1}}$ be the next normal agent sampling time strictly greater than $t_{i_1}^{k_{i_1}}$, with the associated agent denoted $i_2\in\mathcal{N}$. Let $\vec{x}^{k_{i_1}},\ldots,\vec{x}^{k_{i_{|\mathcal{N}|}}}$ denote the most recently sampled states of all normal agents. By prior arguments $\vec{x}^{k_{i_l}}\in\overline{B}(\vec{x}^{k_{i_1}},\epsilon^*)$ for all $l\in1,\ldots,|\mathcal{N}|$, and therefore by (5.39) at time $t_{i_1}^{k_{i_1}}$ we have

$$\sum_{p\in1,\ldots,|\mathcal{N}|}\gamma_i^{\min}(\vec{x}^{k_{i_p}})-\sum_{j\in\mathcal{A}}\gamma_j^{\max}(\vec{x}^{k_{i_1}})+\alpha(h(\vec{x}^{k_{i_1}}))+$$
$$\eta^*(\Gamma^{\max}+2\delta^{\max})\leq0.$$

From this it holds that for all $t\in[t_{i_1}^{k_{i_1}},t_{i_2}^{k_{i_2}})$ we have

$$\dot{h}(\vec{x})+\alpha(h(\vec{x}))\leq\sum_{p\in1,\ldots,N}\gamma_i^{\min}(\vec{x}^{k_{i_p}})-\sum_{j\in\mathcal{A}}\gamma_j^{\max}(\vec{x}^{k_{i_1}})+$$
$$\alpha(h(\vec{x}^{k_{i_1}}))+\eta^*(\Gamma^{\max}+2\delta^{\max})\leq0.$$

It follows that S is forward invariant on the interval $t\in[t_{i_1}^{k_{i_1}},t_{i_2}^{k_{i_2}})$. The preceding arguments can be repeated for any subsequent adjacent sampling times $t_{i_l}^{k_{i_l}},t_{i_p}^{k_{i_p}},t_{i_l}^{k_{i_l}}<t_{i_p}^{k_{i_p}}$ to show that S is forward invariant on $[t_{i_l}^{k_{i_l}},t_{i_p}^{k_{i_p}})$, which concludes the proof. □

5.5 Safe Set Functions with High Relative Degree

It has been demonstrated in prior literature that there exist safe set functions h where agents' control inputs do not appear in the expression for the time derivative $\dot{h}(\vec{x})$ [252, 262], i.e., $\frac{\partial h(\vec{x})}{\partial x_i}g_i(\vec{x})=\mathbf{0}$ for all $\vec{x}$. These functions are said to have *high relative degree with respect to the system dynamics*. In such cases, prior literature has considered methods for computing continuous-time controllers that provably maintain forward invariance of the safe set. These prior results do not consider sys-

tems with sampled-data dynamics however, nor do they consider the presence of agents behaving in an adversarial manner. In this section we extend our previous results to consider a class of safe set functions having high relative degree w.r.t. system dynamics.

In prior work, safety of systems without disturbances and having continuous control inputs using safe set functions having high relative degree w.r.t. system dynamics is typically considered as follows: a function $h : \mathbb{R}^{\bar{n}} \to \mathbb{R}$ describing the safe set is used to define a series of functions $\psi_j : \mathbb{R}^{\bar{n}} \to \mathbb{R}$, $j = 1, \ldots, q$ in the following manner:

$$\begin{aligned} \psi_0(\vec{x}) &\triangleq h(\vec{x}), \\ \psi_1(\vec{x}) &\triangleq \dot{\psi}_0(\vec{x}) + \alpha_1(\psi_0(\vec{x})), \\ &\vdots \\ \psi_q(\vec{x}) &\triangleq \dot{\psi}_{q-1} + \alpha_q(\psi_{q-1}(\vec{x})), \end{aligned} \tag{5.40}$$

where each $\alpha_j : \mathbb{R} \to \mathbb{R}$ is an extended class-$\mathcal{K}_\infty$ function and is locally Lipschitz continuous on $\mathbb{R}$. The integer $q \in \mathbb{Z}_{\geq 1}$ is chosen to be the smallest integer such that a control input u_i for some $i \in \mathcal{V}$ appears in the expression for $\psi_q(\vec{x})$. The integer q is called the *relative degree* of h w.r.t the system dynamics. The functions in (5.40) are associated with the following series of sets:

$$\begin{aligned} S_1 &\triangleq \{\vec{x} \in \mathbb{R}^{\bar{n}} : \psi_0(\vec{x}) \leq 0\}. \\ S_2 &\triangleq \{\vec{x} \in \mathbb{R}^{\bar{n}} : \psi_1(\vec{x}) \leq 0\}. \\ &\vdots \\ S_q &\triangleq \{\vec{x} \in \mathbb{R}^{\bar{n}} : \psi_{q-1}(\vec{x}) \leq 0\}. \end{aligned} \tag{5.41}$$

For brevity, we denote $S_I \triangleq \bigcap_{r=1}^{p} S_r$. The following result from prior literature applies to systems with *continuous* control inputs:

Theorem 5.5 ([252]). *Suppose $\vec{x}(t_0) \in \bigcap_{i=1}^{p} S_i$. Then the set $\bigcap_{i=1}^{q} S_i$ is rendered forward invariant under any Lipschitz continuous controller $\vec{u}(t)$ that ensures the condition $\psi_q(\vec{x}(t)) \leq 0$ for all $t \geq t_0$.*

However, this prior result considers continuous control inputs, does not account for the disturbances $\psi_i(t)$ in (5.7), and does not consider the presence of agents behaving in an adversarial manner.

This section will extend the results in the previous section to present a method for normally-behaving agents with the sampled-data dynamics (5.7) to maintain safety using a function h with high relative degree w.r.t. (5.7) in the presence of adversarial agents. First, to address the presence of the disturbances $\phi_i(t)$, $i \in \mathcal{V}$, recall from Lemma 5.3 that there exists a constant $c_h \geq 0$ such

that $\left\|\sum_{i\in\mathcal{V}} L_{\phi_i} h^{x_i}(\vec{x})\right\| \leq c_h \sum_{i\in\mathcal{V}} \phi_i^{\max}$. We define the constant

$$\xi = c_h \sum_{i\in\mathcal{V}} \phi_i^{\max}. \tag{5.42}$$

The function $h(\vec{x})$ and constant ξ are used to define a series of functions $\psi_j^d : \mathbb{R}^{\bar{n}} \to \mathbb{R}$, $j = 1, \ldots, q$ in the following manner:

$$\begin{aligned}
\psi_0^d(\vec{x}) &\triangleq h(\vec{x}),\\
\psi_1^d(\vec{x}) &\triangleq \sum_{i\in\mathcal{V}} L_{f_i} h^{x_i}(\vec{x}) + \xi + \alpha_1(\psi_0^d(\vec{x})),\\
\psi_2^d(\vec{x}) &\triangleq \dot{\psi}_1^d(\vec{x}) + \alpha_2(\psi_1^d(\vec{x})),\\
&\vdots\\
\psi_q^d(\vec{x}) &\triangleq \dot{\psi}_{q-1}^d + \alpha_q(\psi_{q-1}^d(\vec{x})),
\end{aligned} \tag{5.43}$$

where each $\alpha_j : \mathbb{R} \to \mathbb{R}$ is an extended class $\mathcal{K}_\infty$ function and is locally Lipschitz on $\mathbb{R}$. We make the following assumptions:

Assumption 5.4. *The agent inputs u_i for all $i \in \mathcal{V}$ appear simultaneously in $\psi_q^d(\vec{x})$, $q \in \mathbb{Z}_{\geq 1}$, and are all absent in all ψ_j^d, $0 \leq j < q$.*

Assumption 5.5. *The function ψ_{q-1}^d satisfies $\psi_{q-1}^d \in \mathcal{C}_{loc}^{1,1}$.*

In particular, this section considers cases where the relative degree $q > 1$, since cases where $q = 1$ can be treated by the results in the previous section. The sets $S_1^d, \ldots, S_q^d$ and S_I^d are defined as

$$\begin{aligned}
S_1^d &\triangleq \{\vec{x} \in \mathbb{R}^{\bar{n}} : \psi_0^d(\vec{x}) \leq 0\}.\\
S_2^d &\triangleq \{\vec{x} \in \mathbb{R}^{\bar{n}} : \psi_1^d(\vec{x}) \leq 0\}.\\
&\vdots\\
S_q^d &\triangleq \{\vec{x} \in \mathbb{R}^{\bar{n}} : \psi_{q-1}^d(\vec{x}) \leq 0\}\\
S_I^d &\triangleq \bigcap_{k=1}^{q} S_k^d.
\end{aligned} \tag{5.44}$$

The following Lemma will be needed for our main result. It allows the analysis to consider disturbances $\phi_i(t)$ that possibly cannot be differentiated q times with respect to time.

Lemma 5.4. *Let h have relative degree $q > 1$ with respect to (5.7). Then it holds that $\dot{\psi}_0^d(\vec{x}) + \alpha_1(\psi_0^d(\vec{x})) \leq \psi_1^d(\vec{x})$ for all $t \geq 0$.*

Proof. Since $q > 0$, by Assumption 5.4 the time derivative of $\psi_0^d(\vec{x})$ satisfies

$$\dot{\psi}_0^d(\vec{x}) = \sum_{i \in \mathcal{V}} L_{f_i} h^{x_i}(\vec{x}) + L_{\phi_i} h^{x_i}(\vec{x}).$$

From Lemma 5.3 and equation (5.42) we have $\left\| \sum_{i \in \mathcal{V}} L_{\phi_i} h^{x_i}(\vec{x}) \right\| \leq c_h \sum_{i \in \mathcal{V}} \phi_i^{\max} = \xi$. Using (5.43) it follows that

$$\begin{aligned} \dot{\psi}_0^d(\vec{x}) + \alpha_1(\psi_0^d(\vec{x})) &\leq \sum_{i \in \mathcal{V}} (L_{f_i} h^{x_i}(\vec{x})) + \xi + \alpha_1(\psi_0^d(\vec{x})), \\ &= \psi_1^d(\vec{x}), \end{aligned}$$

which concludes the proof. □

By upper bounding the term $L_{\phi_i} h^{x_i}(\vec{x})$ with the constant ξ, no time derivatives of $\phi_i(t)$ appear in the functions $\psi_2, ...\psi_q$.

Similar to Theorem 5.5, to achieve forward invariance of S_I^d under a ZOH control law the key condition is to show that $\psi_q^d(\vec{x}(t), \vec{u}(t)) \leq 0$ for all $t \geq t_0$. Using a similar method as the prior section, for a ZOH $\vec{u}^k$ we can define the error term

$$e_\psi(t, t^k) = \left(\psi_q^d(\vec{x}) - \psi_q^d(\vec{x}^k) \right). \tag{5.45}$$

For all $t \in [t_i^k, t_i^{k+1})$ it therefore holds that

$$\begin{aligned} \psi_q^d(\vec{x}(t)) &= \psi_q^d(\vec{x}^k) + e_\psi(t, t_i^k) \\ &\leq \psi_q^d(\vec{x}^k) + \sup_{t \in [t_i^k, t_i^{k+1})} \left\| e_\psi(t, t^k) \right\|. \end{aligned}$$

If it holds that $\psi_q^d(\vec{x}^k) + \sup_{t \in [t_i^k, t_i^{k+1})} \left\| e_\psi(t, t^k) \right\| \leq 0$, then for all $t \in [t_i^k, t_i^{k+1})$ we therefore have $\psi_q^d(\vec{x}) \leq 0$ for all $t \in [t_i^k, t_i^{k+1})$.

Consider the asynchronous system with perturbed sampling times from section 5.4.3 such that Assumption 5.4 is satisfied and the function h has relative degree q under (5.7). Using (5.7) and (5.43), the function $\psi_q^d(\vec{x})$ can be expanded into the expression

$$\begin{aligned} \psi_q^d(\vec{x}) =& \dot{\psi}_{q-1}^d(\vec{x}) + \alpha_q(\psi_{q-1}^d(\vec{x})), \\ =& \sum_{i \in \mathcal{N}} L_{f_i} (\psi_{q-1}^d)^{x_i}(\vec{x}) + L_{g_i} (\psi_{q-1}^d)^{x_i}(\vec{x}) u_i + \sum_{j \in \mathcal{A}} L_{f_j} (\psi_{q-1}^d)^{x_j}(\vec{x}) + L_{g_j} (\psi_{q-1}^d)^{x_j}(\vec{x}) u_j \\ &+ \alpha_q(\psi_{q-1}^d(\vec{x})) \end{aligned} \tag{5.46}$$

Observe that the RHS of (5.46) is affine in $\vec{u}$. This follows from (5.46) and the definition of the relative degree q from Assumption 5.4. Similar to equation (5.12), define the functions

$$\begin{aligned}\hat{\gamma}_i^{\min}(\vec{x}) &= \min_{u_i \in \mathcal{U}_i} \left[L_{f_i}(\psi_{q-1}^d)^{x_i}(\vec{x}) + L_{g_i}(\psi_{q-1}^d)^{x_i}(\vec{x}) u_i \right], \\ \hat{\gamma}_i^{\max}(\vec{x}) &= \max_{u_i \in \mathcal{U}_i} \left[L_{f_i}(\psi_{q-1}^d)^{x_i}(\vec{x}) + L_{g_i}(\psi_{q-1}^d)^{x_i}(\vec{x}) u_i \right].\end{aligned} \tag{5.47}$$

As in the previous section, the functions $\hat{\gamma}_i^{\min}$, $\hat{\gamma}_i^{\max}$ can be shown to be locally Lipschitz on the set S_I^d.

Lemma 5.5. *If the interior of $\mathcal{U}_i(\vec{x})$ is nonempty for all $\vec{x} \in S_I^d$ and $\mathcal{U}_i(\vec{x})$ is uniformly compact near $\vec{x}$ for all $\vec{x} \in S_I^d$, then the functions $\hat{\gamma}_i^{\min}(\cdot)$ and $\hat{\gamma}_i^{\max}(\cdot)$ defined by (5.47) are locally Lipschitz on S_I^d.*

Proof. The result follows from Assumption 5.5 and by using similar arguments as in Lemma 5.1. □

Similar to Section 5.4, the following result will be needed to define a function $\eta' : \mathbb{R}_{\geq 0} \to \mathbb{R}$ that will be used to upper bound the normed error term $\sup_{t \in [t_i^k, t_i^{k+1})} \left\| e_\psi(t, t^k) \right\|$:

Lemma 5.6. *Consider the system (5.7) and the function $\psi_{q-1}^d(\vec{x})$. There exist constants $c'_f, c'_g, c'_\alpha, c'_{\hat{\gamma}} \in \mathbb{R}$ such that for all $\vec{x}^1, \vec{x}^2 \in S_I^d$, all of the following inequalities hold:*

$$\begin{aligned}\sum_{i \in \mathcal{N}} \left\| L_{f_i}(\psi_{q-1}^d)^{x_i}(\vec{x}^1) - L_{f_i}(\psi_{q-1}^d)^{x_i}(\vec{x}^2) \right\| &\leq c'_f \left\| \vec{x}^1 - \vec{x}^2 \right\|, \\ \sum_{i \in \mathcal{N}} \left\| L_{g_i}(\psi_{q-1}^d)^{x_i}(\vec{x}^1) - L_{g_i}(\psi_{q-1}^d)^{x_i}(\vec{x}^2) \right\| &\leq c'_g \left\| \vec{x}^1 - \vec{x}^2 \right\|, \\ \left\| \alpha_q(\psi_{q-1}^d(\vec{x}^1)) - \alpha_q(\psi_{q-1}^d(\vec{x}^2)) \right\| &\leq c'_\alpha \left\| \vec{x}^1 - \vec{x}^2 \right\|, \\ \sum_{j \in \mathcal{A}} \left\| \hat{\gamma}_j^{\max}(\vec{x}^1) - \hat{\gamma}_j^{\max}(\vec{x}^2) \right\| &\leq c'_{\hat{\gamma}} \left\| \vec{x}^1 - \vec{x}^2 \right\|,\end{aligned}$$

Proof. Follows from $\psi_{q-1} \in \mathcal{C}_{loc}^{1,1}$ by Assumption 5.5, from α_q being locally Lipschitz on $\mathbb{R}$ by definition, and from $\hat{\gamma}_i^{\min}, \hat{\gamma}_i^{\max}$ being locally Lipschitz by Lemma 5.5. □

Using the constants defined in Lemma 5.6 and the function $\epsilon(\cdot)$ in (5.22), we define the function $\eta' : \mathbb{R}_{\geq 0} \to \mathbb{R}$ as follows:

$$\eta'(\Gamma) = \left(c'_f + c'_g u_{\max} + c'_\alpha + c'_{\hat{\gamma}} \right) \epsilon(\Gamma). \tag{5.48}$$

This definition of $\eta'(\cdot)$ is used to define the following feasible sets for $i \in \mathcal{V}$. Recall from Section 5.4.3 that $\hat{u}_l^{k_i}$ denotes the most recently received input value by agent $i \in \mathcal{N}$ from agent $l \in \mathcal{N}$ at time t_i^k.

$$
\begin{aligned}
K_i^{\psi}(\vec{x}^{k_i}) &= \left\{u_i \in \mathcal{U}_i : \psi_q^d(\vec{x}^{k_i}) \leq 0\right\}, \\
&= \Big\{u_i \in \mathcal{U}_i : L_{f_i}(\psi_{q-1}^d)^{x_i}(\vec{x}^{k_i}) + L_{g_i}(\psi_{q-1}^d)^{x_i}(\vec{x}^{k_i})u_i \\
&\quad + \sum_{l \in \mathcal{N}\setminus\{i\}} \left[L_{f_l}(\psi_{q-1}^d)^{x_l}(\vec{x}^{k_i}) + L_{g_l}(\psi_{q-1}^d)^{x_l}(\vec{x}^{k_i})\hat{u}_l^{k_i}\right] \\
&\quad + \sum_{j \in \mathcal{A}} \gamma_j^{\max}(\vec{x}^{k_i}) + \alpha_q(\psi_{q-1}^d(\vec{x}^{k_i})) + \eta'(\Gamma_i + \delta^{\max}) \leq 0\Big\}.
\end{aligned}
$$

The next Theorem demonstrates conditions under which the set S may be rendered forward invariant for trajectories of the system (5.7).

Theorem 5.6. *Consider the system (5.7) with sampling times described by (5.35). Let $\psi_1^d, \ldots, \psi_q^d$ be defined as in (5.43). If at sampling time t_i^k for $k \geq 0$, $i \in \mathcal{N}$ it holds that $\vec{x}^{k_i} \in S_I^d$, then for any $u_i^{k_i} \in K_i^{\psi}(\vec{x}^{k_i})$ the trajectory $\vec{x}(t)$ satisfies $\vec{x}(t) \in S_I^d$ for all $t \in [t_i^k, t_i^{k+1})$.*

Proof. From (5.45) and (5.43), we have

$$
\begin{aligned}
e_{\psi}(t, t^k) &= \left(\psi_q^d(\vec{x}) - \psi_q^d(\vec{x}^k)\right), \\
&= \sum_{i \in \mathcal{N}} \left(L_{f_i}(\psi_{q-1}^d)^{x_i}(\vec{x}) - L_{f_i}(\psi_{q-1}^d)^{x_i}(\vec{x}^k)\right) + \\
&\quad \sum_{i \in \mathcal{N}} \left(L_{g_i}(\psi_{q-1}^d)^{x_i}(\vec{x}) - L_{g_i}(\psi_{q-1}^d)^{x_i}(\vec{x}^k)\right) u_i^k + \\
&\quad \sum_{j \in \mathcal{A}} \left(\hat{\gamma}_j^{\max}(\vec{x}) - \hat{\gamma}_j^{\max}(\vec{x}^k)\right) + \\
&\quad \left(\alpha_q(\psi_{q-1}(\vec{x})) - \alpha_q(\psi_{q-1}(\vec{x}^k))\right)
\end{aligned} \tag{5.49}
$$

Choose any $i \in \mathcal{N}$ and consider the time interval $t \in [t_i^k, t_i^{k+1})$. Recall that $t_i^{k+1} - t_i^k \leq \Gamma_i + \delta^{\max}$ $\forall k \in \mathbb{Z}_{\geq 0}$ by virtue of (5.35) and the definition of $\delta^{\max}$. In particular, this implies $\epsilon(\Gamma_i + \delta_i(k)) \leq \epsilon(\Gamma_i + \delta^{\max})$ for all $k \in \mathbb{Z}_{\geq 0}$ since $\epsilon(\cdot)$ is a class-$\mathcal{K}$ function in Γ. Using equations (5.49), (5.48), Lemma 5.6, and Lemma (5.2) observe that

$$
\begin{aligned}
&\sup_{t \in [t_i^k, t_i^{k+1})} \left\|e_{\psi}(t, t_i^k)\right\| \leq \left(c_f' + c_g' u_{\max} + c_{\alpha}' + c_{\hat{\gamma}}'\right) \epsilon(\Gamma_i + \delta^{\max}), \\
\implies &\sup_{t \in [t_i^k, t_i^{k+1})} \left\|e_{\psi}(t, t_i^k)\right\| \leq \eta'(\Gamma_i + \delta^{\max})
\end{aligned}
$$

The same logic as in Theorem 5.2 can then be used to demonstrate that for any $u_i \in K_i^{\psi}(\vec{x}^{k_i})$ it holds that $\psi_q^d(\vec{x}(t)) \leq \psi_q^d(\vec{x}^k) + \eta'(\Gamma_i + \delta^{\max}) \leq 0$ for all $t \in [t_i^k, t_i^{k+1})$.

We next demonstrate that $\psi_q^d(\vec{x}) \leq 0$ for all $t \in [t_i^k, t_i^{k+1})$ implies that $\vec{x} \in S_I^d$ $\forall t \in [t_i^k, t_i^{k+1})$. For brevity, denote $I_i^k = [t_i^k, t_i^{k+1})$. Since $\psi_q^d(\vec{x}) \leq 0$ for all $t \in I_i^k$, from (5.43) this implies that $\dot{\psi}_{q-1}^d(\vec{x}) + \alpha_q(\psi_{q-1}^d(\vec{x})) \leq 0$ for all $t \in I_i^k$. By Nagumo's Theorem, this implies that $\psi_{q-1}^d(\vec{x}) \leq 0$ for all $t \in I_i^k$. Continuing inductively, observe that for all $2 \leq j \leq q$ it holds that $\psi_j^d(\vec{x}) \leq 0$ $\forall t \in I_i^k$, which implies $\dot{\psi}_{j-1}^d(\vec{x}) + \alpha_j(\psi_{j-1}^d(\vec{x})) \leq 0$ $\forall t \in I_i^k$, which further implies by Nagumo's Theorem that $\psi_{j-1}^d(\vec{x}) \leq 0$ $\forall t \in I_i^k$. By this logic we therefore have $\psi_q^d(\vec{x}) \leq 0 \implies \psi_{q-1}^d(\vec{x}) \leq 0 \implies \ldots \implies \psi_1^d(\vec{x}) \leq 0$ $\forall t \in I_i^k$. By Lemma 5.4, $\psi_1^d(\vec{x}) \geq \dot{\psi}_0^d(\vec{x}) + \alpha_1(\psi_0^d(\vec{x}))$ for all $t \geq 0$. Therefore $\psi_1^d(\vec{x}) \leq 0$ $\forall t \in I_i^k$ implies that $\dot{\psi}_0^d(\vec{x}) + \alpha_1(\psi_0^d(\vec{x})) \leq 0$ $\forall t \in I_i^k$, which implies that $\psi_0^d(\vec{x}) \leq 0$ $\forall t \in I_i^k$. Using the definitions in (5.44), it follows that the trajectory $\vec{x}(t)$ satisfies $\vec{x}(t) \in S_I^d = \bigcap_{j=1}^q S_j^d$ for all $t \in I_i^k$, which concludes the proof. □

Under the communication protocol described in Section 5.4.3, each normal agent $i \in \mathcal{N}$ can use the most recently received inputs $\hat{u}_l^{k_i}$ from other normal agents to calculate a control input $u_i^{k_i} \in K_i^{\psi}(\vec{x}^{k_i})$. Such a $u_i^{k_i}$ can be computed by solving the following QP:

$$\begin{aligned} u_i(\vec{x}^{k_i}) = \underset{u_i \in \mathcal{U}_i}{\arg\min} \quad & \left\| u_i - u_{i,\text{nom}}^{k_i} \right\|_2^2 \\ \text{s.t.} \quad & L_{f_i}(\psi_{q-1}^d)^{x_i}(\vec{x}^{k_i}) + L_{g_i}(\psi_{q-1}^d)^{x_i}(\vec{x}^{k_i}) u_i + \\ & \sum_{l \in \mathcal{N} \setminus \{i\}} \left(L_{f_l}(\psi_{q-1}^d)^{x_l}(\vec{x}^{k_i}) + L_{g_l}(\psi_{q-1}^d)^{x_l}(\vec{x}^{k_i}) \hat{u}_l^{k_i} \right) + \\ & \sum_{j \in \mathcal{A}} \hat{\gamma}_j^{\max}(\vec{x}^{k_i}) + \alpha_q(\psi_{q-1}^d(\vec{x}^{k_i})) + \eta'(\Gamma_i + \delta^{\max}) \leq 0. \end{aligned} \tag{5.50}$$

5.5.1 Discussion

This section has considered systems satisfying Assumption 5.4 where all agents' inputs appear simultaneously for the same relative degree q of h under (5.7). However, Assumption 5.4 may not be satisfied in general for systems composed of agents with heterogeneous control-affine dynamics. A simple example is a system composed of both single- and double-integrator agents with states in $\mathbb{R}^3$. Only control inputs for the single integrators appear in the function $\psi_1(\vec{x}, \vec{u})$ from (5.43), while the function $\psi_2(\vec{x}, \vec{u}, \dot{\vec{u}}) = \frac{d}{dt}(\psi_1(\vec{x}, \vec{u})) + \alpha_2(\psi_1(\vec{x}, \vec{u}))$ simultaneously contains single-integrator inputs, time-derivatives of single-integrator inputs, and double-integrator inputs.

The extension of this chapter's results to the general case does not immediately follow for two reasons. First, the time derivatives of inputs $\dot{\vec{u}}, \ddot{\vec{u}}, \ldots, \vec{u}^{(r)}$, $r \in \mathbb{Z}_{\geq 1}$ for ZOH controllers are undefined at sampling instances. This necessitates a careful and rigorous mathematical analysis of

the behavior of each $\psi_j(\vec{x}, \vec{u}, \dot{\vec{u}}, \ldots)$ to ensure that safety can indeed be guaranteed under a ZOH control law. Second, when considering multi-agent safe set functions $h(\cdot)$ the functions ψ_j for higher values of j are not guaranteed to be convex in $\vec{u}$ when Assumption 5.4 is not satisfied. This nonconvexity inhibits the ability to efficiently compute safety-preserving control inputs. We therefore leave the general case as an interesting direction for future investigation.

5.6 Simulations

Simulations were performed using a combination of MATLAB and the Julia programming language [271]. The simulations used the OSQP optimization package [242] and the ForwardDiff automatic differentiation package [272].

While forward invariance of the safe set is guaranteed for any control inputs in the feasible sets $K_i(\cdot)$, $K_i^{\psi}(\cdot)$, a key issue is guaranteeing that the feasible sets $K_i(\cdot)$, $K_i^{\psi}(\cdot)$ remain nonempty for all forward time. Due to the difficulty of calculating forward reachable sets for general nonlinear systems subject to disturbances [273, 274], prior literature typically does not provide guarantees on the forward nonemptiness of such feasible sets except in very specific cases (e.g. when control input constraints are not considered). Even in the absence of obstacles, it is trivial to find examples where forward invariance of the safe set is impossible. Two such examples are given in Figure 5.2 for single integrator agents in the plane $\mathbb{R}^2$, where adversaries surround a normal agent or pin a normal agent against an obstacle. Proving the forward nonemptiness of sets $K_i(\cdot)$ and $K_i^{\psi}(\cdot)$, however, is beyond the scope of this work.

5.6.1 Unicycle Agents in $\mathbb{R}^2$

The first simulation involves a network of $n = 5$ agents with unicycle dynamics in $\mathbb{R}^2$. Agents are nominally tasked with tracking time-varying trajectories defined by a Bezier curve, timing law, and local formational offsets. The agents must also avoid static obstacles. Two agents misbehave by each pursuing the respective closest normal agent. The state of each unicycle $i \in \mathcal{V}$ is denoted $x_i = \begin{bmatrix} x_{i,1} & x_{i,2} & x_{i,3} \end{bmatrix}^T$. Each unicycle is controlled via an input-output linearization method [234, Ch. 11] where each agent has the outputs

$$\begin{aligned} p_{i,1} &= x_{i,1} + b\cos(x_{i,3}), \\ p_{i,2} &= x_{i,2} + b\sin(x_{i,3}),\ b > 0. \end{aligned} \tag{5.51}$$

The output $p_i = \begin{bmatrix} p_{i,1} & p_{i,2} \end{bmatrix}^T$ is treated as having single integrator dynamics $\dot{p}_i = u_i = \begin{bmatrix} u_{i,1} & u_{i,2} \end{bmatrix}^T$. Each agent i is controlled by first computing the output control input

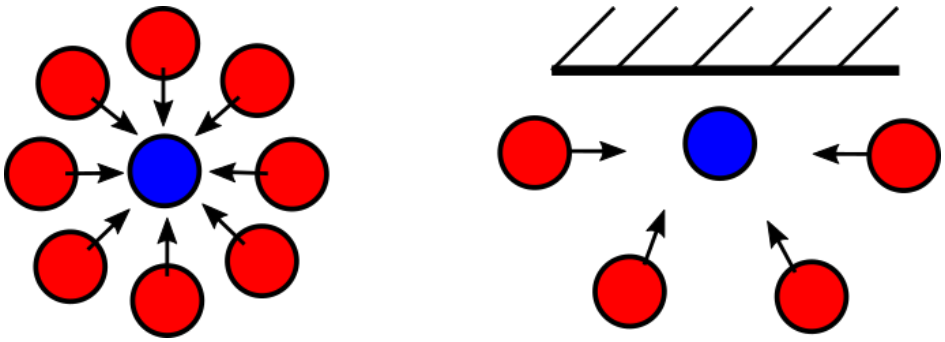


Figure 5.2: Two examples of initial system states where it is impossible to guarantee forward nonemptiness of the normal agent's feasible controls set $K_i(\cdot)$. Agents have single integrator dynamics; the normal agent is depicted in blue, and adversarial agents are depicted in red. The line in the right image denotes an obstacle. Determining initial conditions for which nonemptiness of the feasible sets is guaranteed for all forward time is intractable in general when considering nonlinear control-affine systems.

u_i and minimally modifying u_i via the CBF-based QP method described previously. The final unicycle control inputs $\begin{bmatrix} \nu_i & \omega_i \end{bmatrix}$ are then obtained via the transformation

$$\begin{bmatrix} \nu_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -sin(\theta_i)/b & \cos(\theta_i)/b \end{bmatrix} \begin{bmatrix} u_{i,1} \\ u_{i,2} \end{bmatrix}.$$

At any timestep where the QP is infeasible, each normal agent applies the best-effort safety preserving control (5.16) calculated via the LP (5.37). Infeasibility of the QP generating the control inputs does not necessarily imply that safety cannot be maintained. Reasons why the QP may go infeasible at particular time steps include the conservative nature of the form of $\eta(\cdot)$ and the choice of $\alpha(\cdot)$ function. The LP in (5.37) is applied whenever an agent's QP is infeasible to apply the agent's best control efforts towards maintaining safety. Given control bounds $|\nu_i| \leq \nu_i^{\max}$ and $|\omega_i| \leq \omega_i^{\max}$, it can be shown that the corresponding linear control bounds on $u_{i,1}, u_{i,2}$ are $A_i(x_i) \begin{bmatrix} u_{i,1} \\ u_{i,2} \end{bmatrix} \leq b_i$, with

$$A_i(x_i) = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\cos(\theta_i) & -\sin(\theta_i) \\ -\sin(\theta_i)/b & \cos(\theta_i)/b \\ \sin(\theta_i)/b & -\cos(\theta_i)/b \end{bmatrix}, \; b_i = \begin{bmatrix} \nu_i^{\max} \\ \nu_i^{\max} \\ \omega_i^{\max} \\ \omega_i^{\max} \end{bmatrix} \tag{5.52}$$

For strictly positive $\nu_i^{\max}$, $\omega_i^{\max}$, and b, the set $\mathcal{U}_i = \{u_i : A_i(x_i)u_i - b_i \leq 0\}$ satisfies the conditions of Assumption 5.3 for all $x_i \in \mathbb{R}^3$. In this simulation each normal agent has $\nu_i^{\max} = 4$, $\omega_i^{\max} = 2$,

$i \in \mathcal{N}$. For purposes of this simulation, each adversarial agent has lower maximum linear and angular velocities than the normal agents with $\nu_j^{\max} = 2$, $\omega_j^{\max} = 1$, $j \in \mathcal{A}$. The safe set S is defined using a boolean composition of pairwise collision-avoidance sets for normal-to-normal pairs, normal-to-adversarial pairs, and normal-to-obstacle pairs. More specifically, given $i, i' \in \mathcal{N}$ each safe set $h_{i,i'}(\vec{x})$ is defined with respect to the linearized outputs (5.51) as $h_{i,i'} = (R_c + 2b)^2 - \|p_i - p_{i'}\|_2^2$, with partial derivative $\frac{\partial h_{i,i'}}{\partial p_i} = -2(p_i - p_{i'})$. The normal-to-adversarial and normal-to-obstacle pairwise safe sets for $i \in \mathcal{N}$, $j \in \mathcal{A}$ are defined in a similar manner. The pairwise adversarial-to-adversarial and adversarial-to-obstacle safe sets are *not* considered (as per Remark 5.3), since the nominal control law by definition has no effect on adversarial agents. All pairwise safe sets are composed into a single CBF h_{tot} via boolean AND operations using the *log-sum-exp* smooth approximation to the $\max(\cdot)$ function:

$$h_{tot}(\vec{x}) = \text{LSE}(\Big[h_1, \ldots, h_p\Big]) = \sigma + \frac{1}{\rho} \ln \left(\sum_{i=1}^{p} e^{\rho(h_i - \sigma)} \right), \\ \rho \in \mathbb{R}_{>0}, \ \sigma \in \mathbb{R}. \tag{5.53}$$

The term σ is used to ensure numerical stability of (5.53). The term ρ controls how tightly $\text{LSE}(\cdot)$ approximates $\max(\cdot)$. The reader is referred to [172], [275, Eq (10)] for more details. Sampling times in this simulation are asynchronous; each agent has a nominal time period of $\Gamma = 0.01$ with a time-varying random disturbance satisfying $\delta_i^{\max} = .002$. For each agent $i \in \mathcal{V}$, the disturbance bound satisfies $\phi_i^{\max} = 1.73$, and the term η is set as $\eta(\Gamma) = 8.0566$. Several frames from the simulation are shown in Figure 5.3. A plot of h_{tot} is given in Figure 5.4. As shown by Figure 5.4, under the proposed resilient controller the safety bounds for normal agents are not violated for the duration of the simulation. This is achieved despite the actions of the adversarial agents.

For comparison, Figure 5.5 shows the result of a simulation run under the same parameters but with $\eta(\Gamma) = 0$ for all $t \geq 0$; i.e., sampling, disturbances, and time delays are not taken into account in the normal agents' control actions. In this case Figure 5.5 shows that the safety of the normal agents is not preserved—the value of h_{tot} is temporarily positive, indicating that one or more of the composed safe sets was not invariant for the duration of the simulation.

5.6.2 Double Integrators in $\mathbb{R}^3$

The second simulation involves a network of $n = 8$ double integrator agents in $\mathbb{R}^3$. Four of the agents behave normally and four are adversarial. Similar to the prior simulation, agents are nominally tasked with tracking positions in a time-varying formation defined by a Bezier curve, timing law, and local formational offsets. Each agent $i \in \mathcal{V}$ has the state

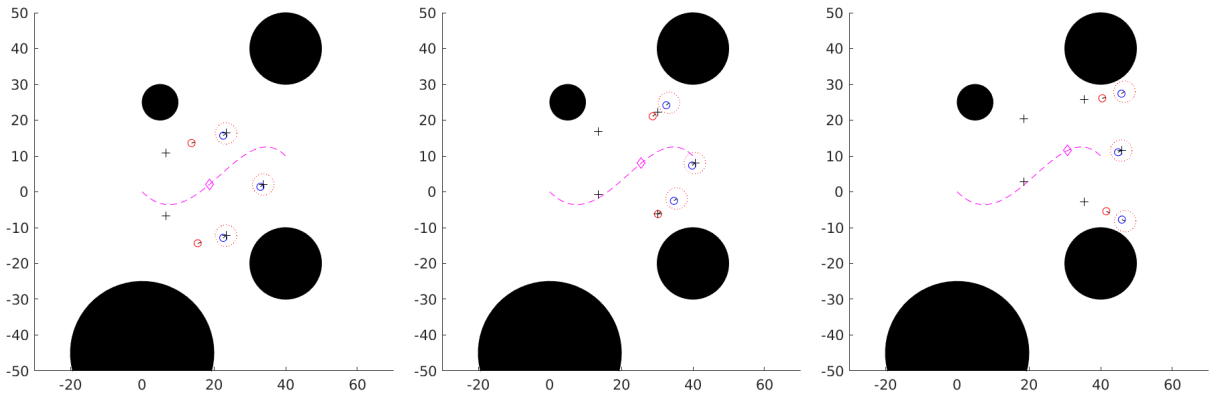


Figure 5.3: Still frames from the video of Simulation 1. Normal agents are represented by blue circles and adversarial agents are represented by red circles. The dotted red lines around the blue circles represent normal agents' safety radii. The time-varying formation trajectory is represented by the dotted magenta line; the magenta diamond represents the center of formation. Black crosses represent agents' nominal local time-varying formational points.

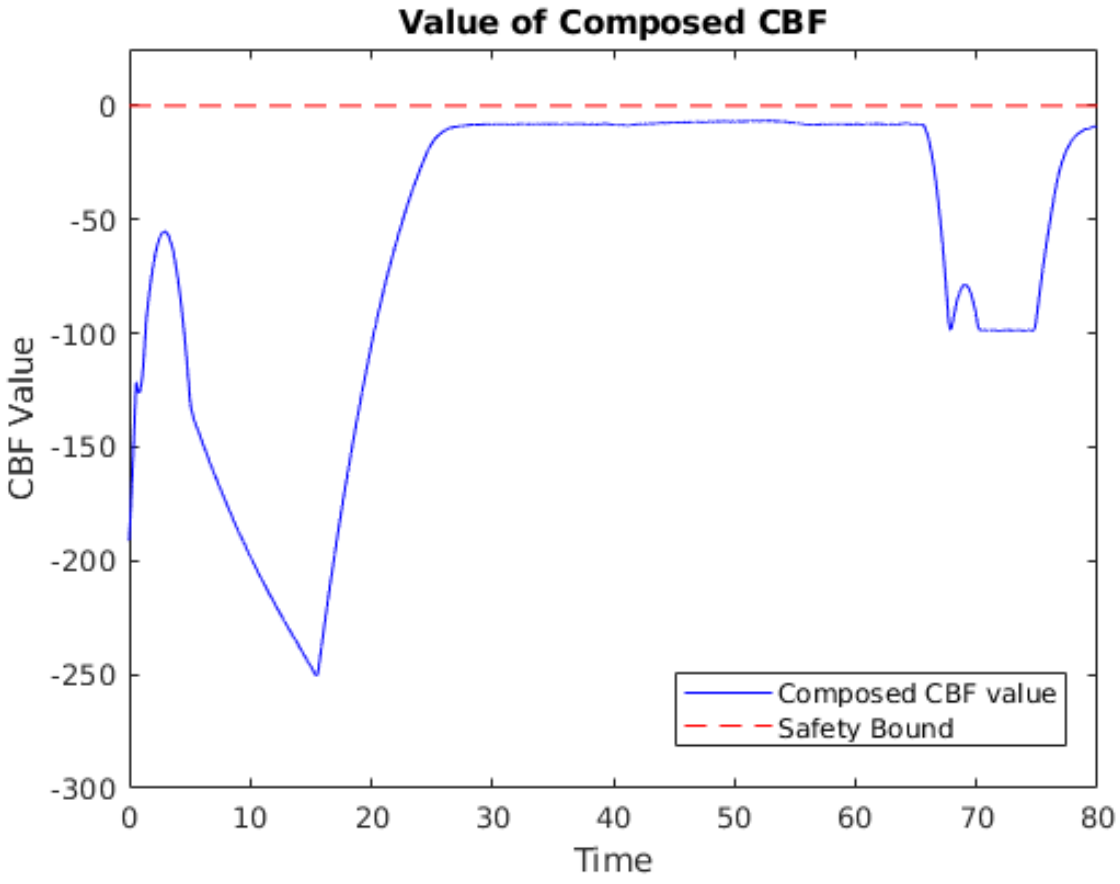


Figure 5.4: The value of the composed function h_{tot} representing the safe set S. Non-positive values represent safety of the normal agents.

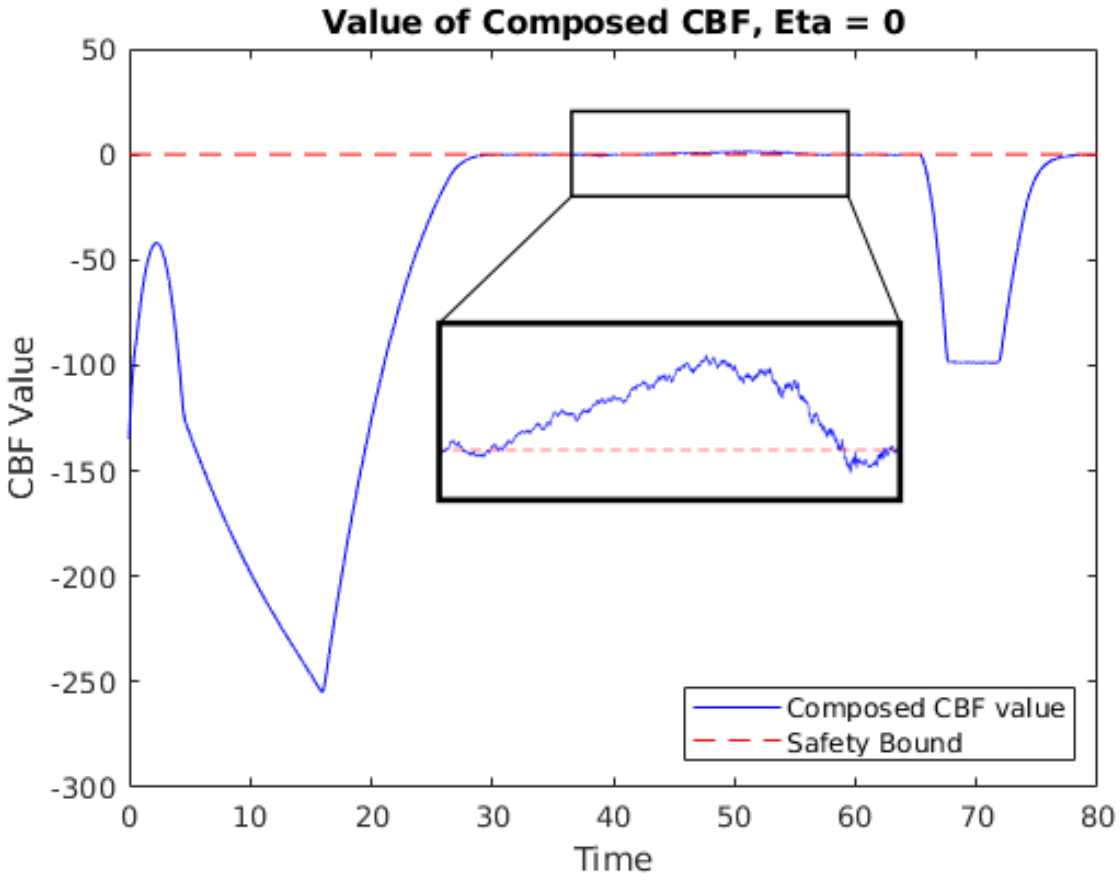


Figure 5.5: The value of the composed function h_{tot} representing the safe set S when $\eta(\Gamma) = 0$ for all normal agents; i.e., sampling times and disturbances are not accounted for in the control input calculations. The safety bound for the normal agents is violated.

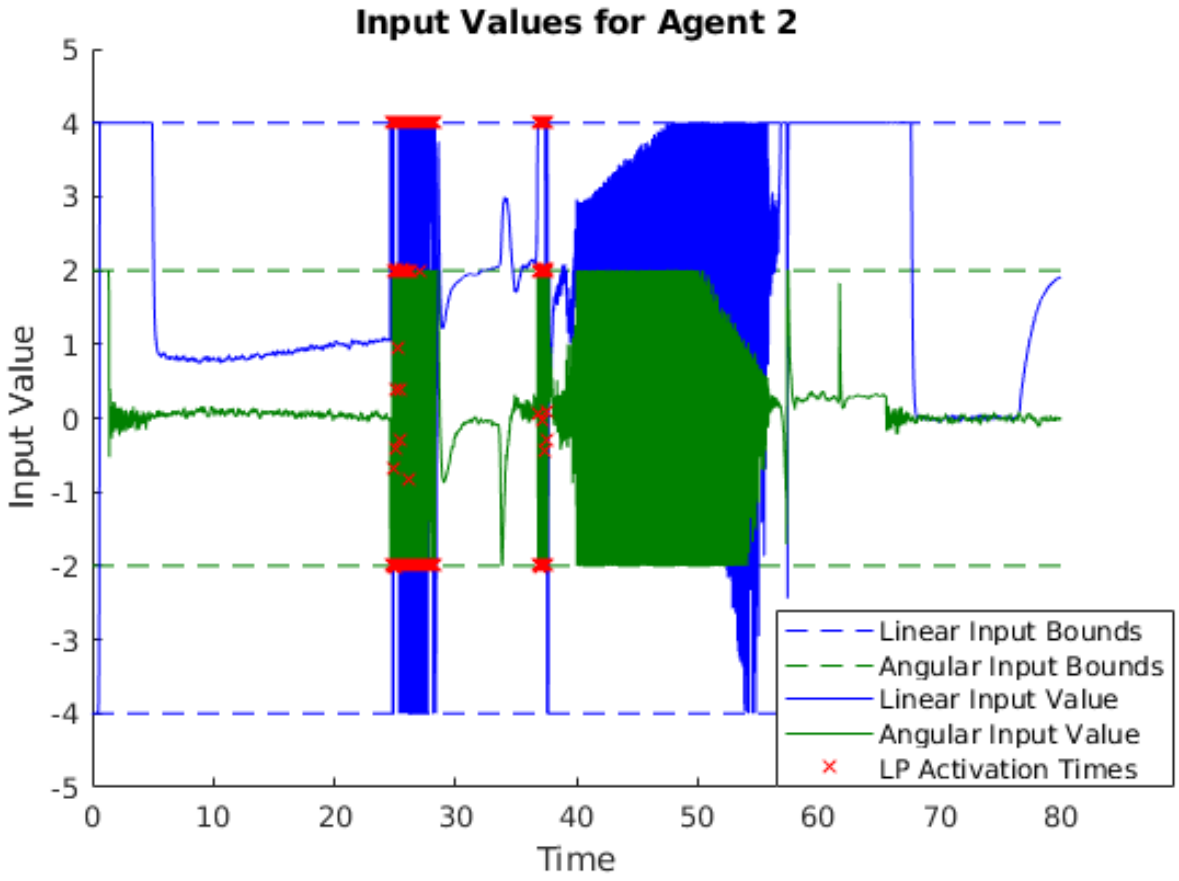


Figure 5.6: Input values for (normal) agent 2. The blue solid line represents linear input value and the green solid line represents angular input value. Dotted lines represent input bounds. Times at which the worst-case LP is used are marked with red X's on both the linear and angular input lines.

$\vec{x}_i = \begin{bmatrix} x_{i,1} & x_{i,2} & x_{i,3} & v_{i,1} & v_{i,2} & v_{i,3} \end{bmatrix}^T$ with the following dynamics:

$$\dot{\vec{x}}_i = \underbrace{\begin{bmatrix} \mathbf{0}_{3\times3} & I_{3\times3} \\ \mathbf{0}_{3\times3} & -\beta I_{3\times3} \end{bmatrix}}_{A} \vec{x}_i + \underbrace{\begin{bmatrix} \mathbf{0}_{3\times3} \\ I_{3\times3} \end{bmatrix}}_{B} \begin{bmatrix} u_{i,1} \\ u_{i,2} \\ u_{i,3} \end{bmatrix} + \phi_i(t). \tag{5.54}$$

Each agent $i \in \mathcal{V}$ has an identical input bound $\|u_i\|_\infty \leq u^{\max} \in \mathbb{R}_{>0}$, with $u^{\max} = 2$. The terms $\beta_i \in \mathbb{R}_{\geq 0}$ are chosen such that each agent has a velocity bound $v_i^{\max} \in \mathbb{R}_{>0}$, with $v_i^{\max} = 3\ \forall i \in \mathcal{V}$. Specifically, $\beta_i = \frac{v_i^{max}}{u^{\max}}$.

Each normal agent $i \in \mathcal{N}$ seeks to track a time-varying formational state $\vec{x}_i^d \in \mathbb{R}^3$. The nominal formation states for all agents are equidistantly distributed around the edge of a circle of radius 30 whose center translates along a time-varying trajectory described by a 3rd order Bezier curve $B(t) = \sum_{k=0}^{3} \vec{\beta}_i b_{i,3}(s(t))$ described by the timing law $s(t) = \frac{t_f - t}{t_f - t_0}$ for $t_f = 140$ and $t_0 = 0$, Bernstein basis polynomials $b_{i,3}(s)$ and the vector coefficients

$$\vec{\beta}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \vec{\beta}_1 = \begin{bmatrix} -25 \\ 25 \\ 30 \end{bmatrix} \quad \vec{\beta}_2 = \begin{bmatrix} 125 \\ 75 \\ -30 \end{bmatrix} \quad \vec{\beta}_3 = \begin{bmatrix} 100 \\ 100 \\ 0 \end{bmatrix}. \tag{5.55}$$

Letting the error $\vec{e}_i$ be defined as $\vec{e}_i = \vec{x}_i^d - \vec{x}_i$, each $i \in \mathcal{N}$ calculates the nominal control law $\vec{u}_{i,\text{nom}} = -K\vec{e}_i - \ddot{\vec{x}}_i^d$ with $K = \begin{bmatrix} k_1 I_{3\times3} & k_2 I_{3\times3} \end{bmatrix}$, where $\ddot{\vec{x}}_i^d$ is the acceleration of $\vec{x}_i^d$, $k_1 = 2$, and $k_2 = 2\sqrt{k_1}$. The nominal input $\vec{u}_{i,\text{nom}}$ is minimally modified via the higher-order CBF-based QP method described in 5.5. Similar to (5.37), at any timestep t_i^k where the QP is infeasible each normal agent $i \in \mathcal{N}$ applies the control action

$$u_i^{\min}(\vec{x}^{k_i}) = \arg\min_{u_i \in \mathcal{U}_i} \left[L_{f_i} \psi_{q-1}^{x_i}(\vec{x}^{k_i}) + L_{g_i} \psi_{q-1}^{x_i}(\vec{x}^{k_i}) u_i \right].$$

The environment contains 10 spherical obstacles with radius 2 randomly distributed across the volume containing the second half of the time-varying trajectory. Adversarial agents $j \in \mathcal{A}$ in this simulation are each assigned a target agent to pursue, with one of the normal agents having multiple pursuers. Each adversarial agent $j \in \mathcal{A}$ is assumed to have full knowledge of its target's current state, but does not have knowledge of its target's control inputs. Defining the error term $\vec{e}_{i,j} = \vec{x}_i - \vec{x}_j$, $i, j \in \mathcal{V}$, each adversary $j \in \mathcal{A}$ applies the control law $\vec{u}_j = -K\vec{e}_{i,j}$ with the matrix K defined as previously described but with $k_1 = 1$. This control input is minimally modified using a CBF QP method to avoid collisions with other adversaries and obstacles, but not with normal agents.

The safe set S in this simulation is defined using a similar boolean composition of pairwise collision avoidance sets as in the previous simulation. At each sampling instance, the normal agent i considers all other agents whose positions lie within a neighborhood of radius 35 from agent i's position $\begin{bmatrix} x_{i,1} & x_{i,2} & x_{i,3} \end{bmatrix}$. All normal-to-normal, normal-to-adversarial, and normal-to-obstacle pairwise safe sets are composed into a single CBF h_{tot} via boolean AND operations using the *log-sum-exp* function. Sampling times in this simulation are asynchronous for normal agents; each $i \in \mathcal{N}$ has a nominal time period of $\Gamma = 0.07$ with $\delta_i^{\max} = 0.03$ for each normal agent. The disturbance $\phi_i(t)$ for each agent $i \in \mathcal{V}$ (normal and adversarial) satisfies $\phi_i^{\max} = .4899$. For each normal agent $i \in \mathcal{N}$ the term η' satisfies $\eta'(\Gamma_i + \delta^{\max}) = 5$, and the term ξ satisfies $\xi = 39.19$. Still frames from the simulation are shown in Figure 5.7, and a plot of the value of h_{tot} vs time is given in Figure 5.8. As shown by Figure 5.8, the safety bounds for the normal agents are not violated for the duration of the simulation despite the actions of the adversaries.

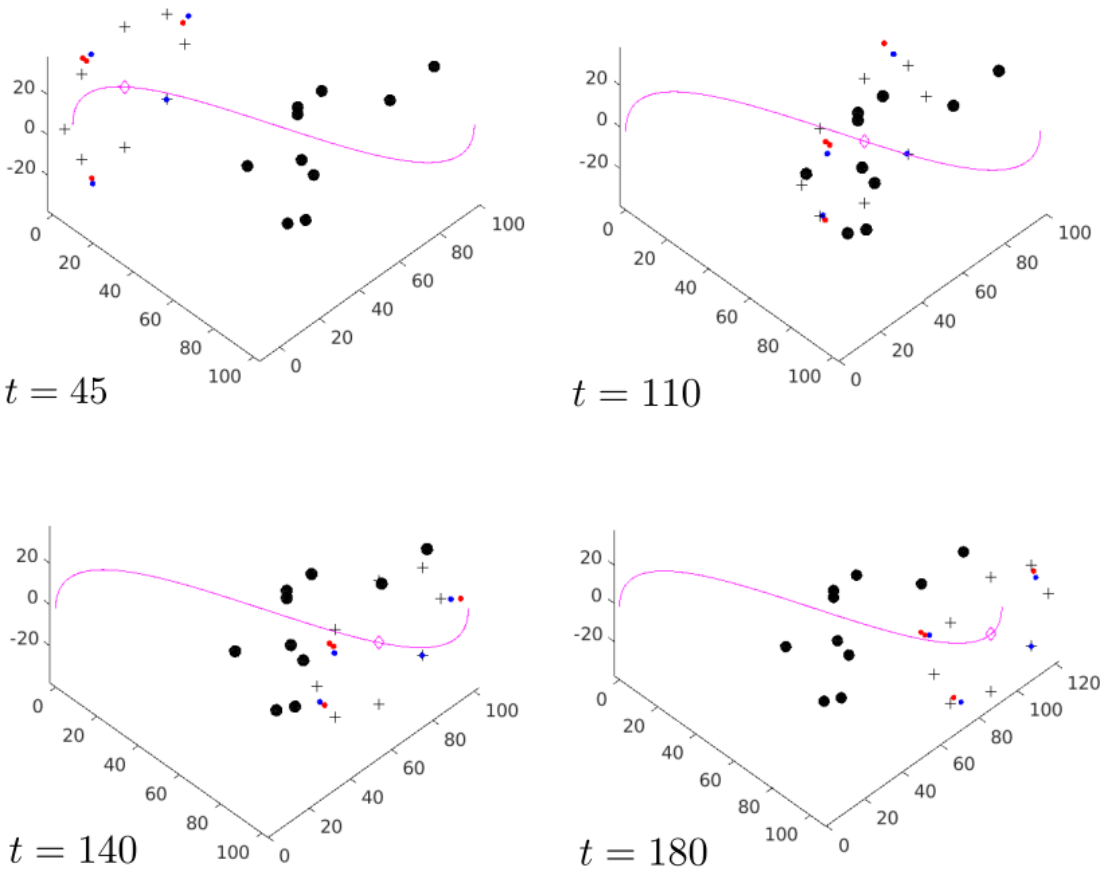


Figure 5.7: Still frames from the video of Simulation 2. Normal agents are represented by blue circles and adversarial agents are represented by red circles. For clarity, the safety radii of the normal agents has been omitted. The time-varying formation trajectory is represented by the dotted magenta line; the magenta diamond represents the center of formation. Black crosses represent individual agents' nominal local time-varying formational points. Black spheres represent randomly placed obstacles.

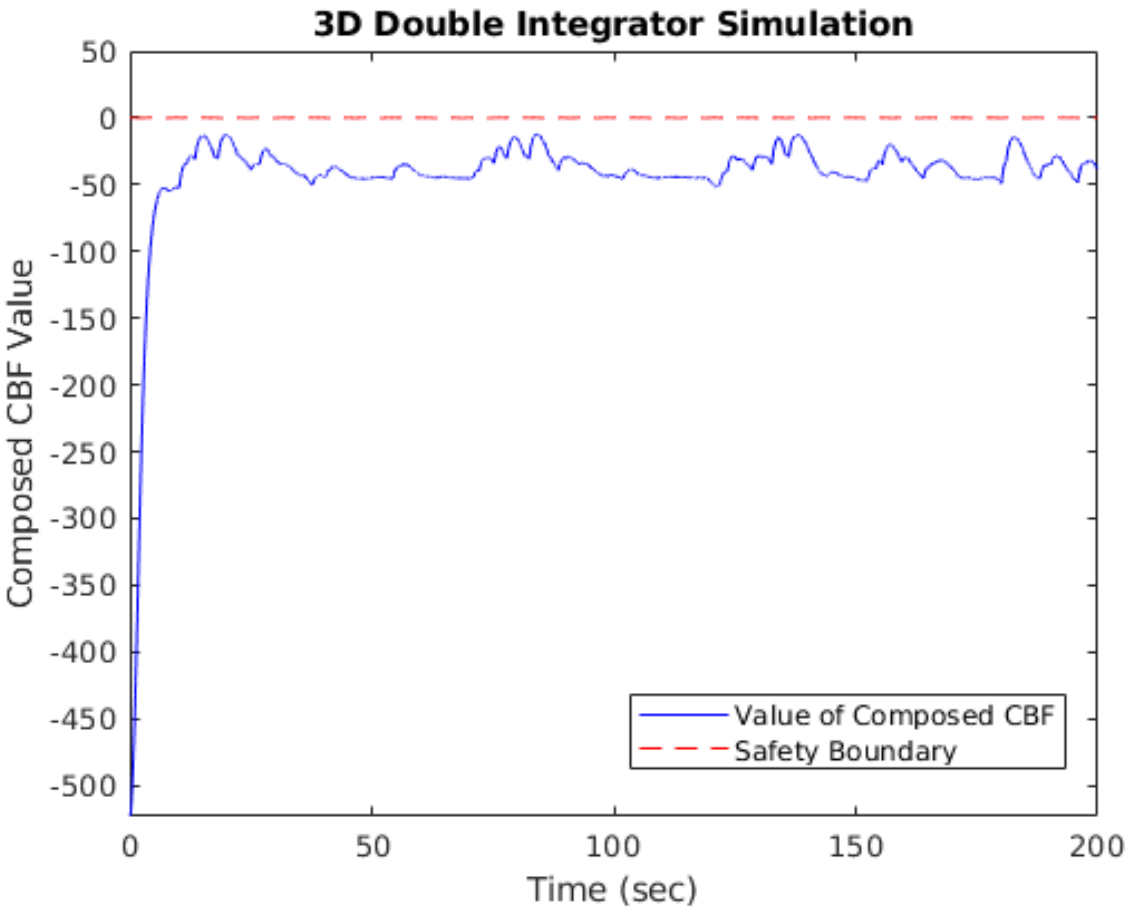


Figure 5.8: The value of the composed function h_{tot} representing the safe set S for all normal agents in the second simulation. Non-positive values represent safety of the normal agents. For the entire duration of this simulation, the value of h_{tot} remains strictly negative, indicating that safety is maintained for all normal agents.

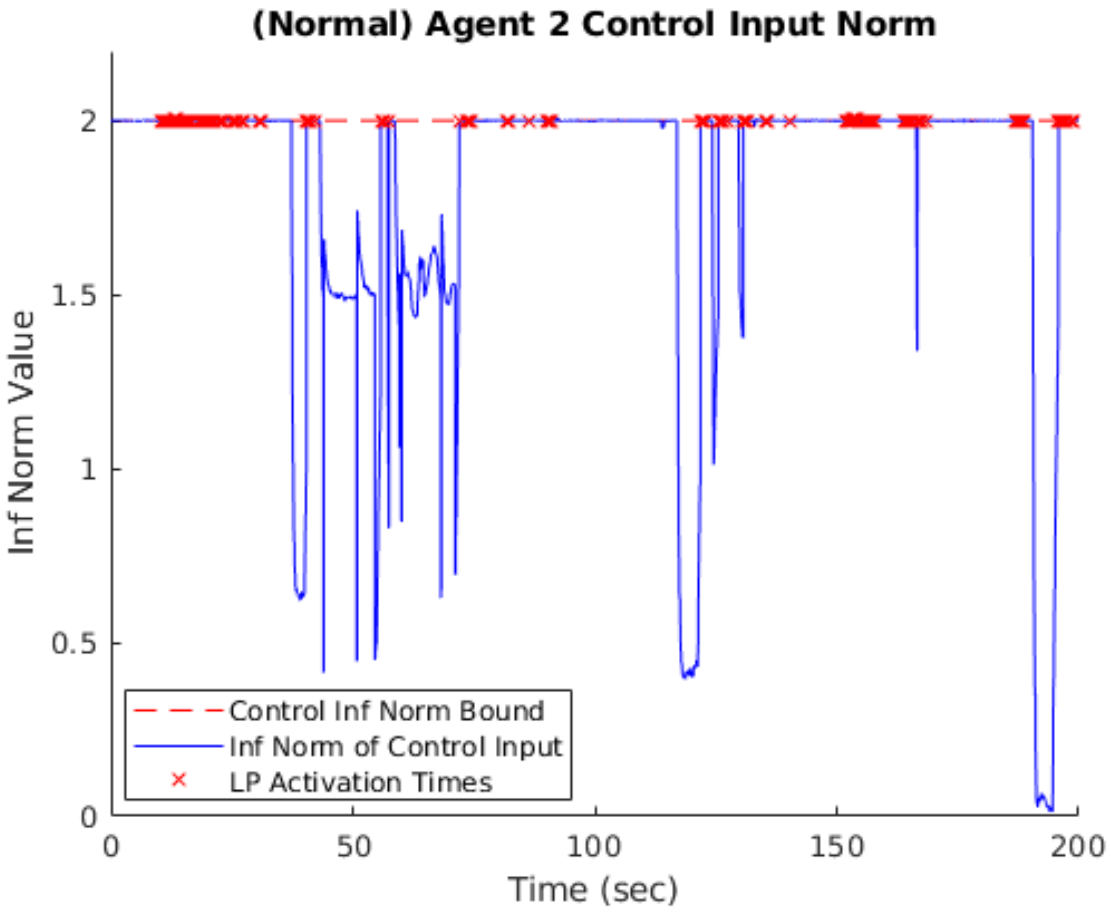


Figure 5.9: Infinity norm of control input for (normal) agent 2. The control norm bound is plotted in red, and the norm of agent 2's control input is plotted in blue. Times when the backup LP is used are marked with red X's.

5.7 Conclusion

In this chapter, we presented a framework for normally-behaving agents to render a safe set forward invariant in the presence of adversarial agents. The proposed method considers distributed sampled-data systems with heterogeneous, asynchronous control affine dynamics, and a class of functions with high relative degree that define the safe set. Directions for future work include investigating how estimates of the forward reachable sets for each agent can be included in the analysis, and cases where the control inputs of heterogeneous agents do not appear simultaneously in higher derivatives of the function h.

CHAPTER 6

Conclusions and Future Work

6.1 Conclusions

This dissertation has presented several novel results regarding resilient consensus and safety maintenance in multi-agent systems.

We summarize these results below, outline opportunities for future work, and then give some points of final discussion on the approach taken by this dissertation.

Chapter 2 presented novel results for several forms of resilient multi-agent consensus using Mean-Subsequence-Reduced (MSR) algorithms. We first presented conditions under which resilient leader-follower consensus in the presence of adversaries can be achieved under time-varying graphs in a discrete-time setting. This result was shown to also have an important adversarial implication in cases where the assumption of an F-local adversarial set is violated. Second, we presented a novel method for achieving resilient finite-time formational consensus in a continuous-time system setting. A novel norm-based filtering method was introduced that ensures a minimum dwell time to the system switching dynamics, and guarantees under proper conditions that normally-behaving follower agents converge to formational positions defined by the locations of a set of leaders. This method was shown to work with bounded control inputs, and a modified version for a discrete-time setting was also presented. Third, we presented a method for resilient finite-time leaderless consensus in a continuous-time setting under a novel nonlinear class of MSR-type controllers. Our analysis uses discontinuous systems theory to rigorously demonstrate convergence, and any Lebesgue-measurable adversarial signals for an F-local set of Byzantine adversarial agents can be considered.

Chapter 3 presented novel results for constructing and analyzing the resilience properties for communication graphs. These results focus on the graph theoretic notions of r-robustness and (r, s)-robustness that are fundamental to the operation of many MSR-type algorithms. To mitigate the NP-hardness of determining the r- and (r, s)-robustness of arbitrary graphs, we first presented results demonstrating a class of circulant digraphs whose structure, r-robustness, and

(r, s)-robustness is parameterized by an integer k. This class of digraphs can be scaled to abitrary size and also exhibits the property of strong r-robustness with respect to (w.r.t.) a set S for a properly chosen subset S of the nodes in the graph. We also presented novel methods for determining the r- and (r, s)-robustness of digraphs and undirected graphs using mixed integer linear programming (MILP). These are the first results to demonstrate that robustness can be determined using an optimization framework, and open the door for the extensive literature on integer programming to be applied to the robustness determination problem. These results are also the first to enable calculating an approximate lower bound on the robustness values of general directed graphs. Simulations were presented showing that in practice the MILP methods generally outperform prior algorithms for robustness determination.

Chapter 4 presented a novel algorithm inspired by the Certified Propagation Algorithm (CPA) for resilient broadcasting vector-valued information from a set of leaders to all followers within a network. Prior work on resilient broadcast using the CPA algorithm typically assumed that there exists a single leader that is immune to adversarial attacks to propagate a specified message to a network. In contrast, the algorithm proposed in Chapter 4 used a multi-leader approach that can tolerate both misbehaving leaders and followers, and that can operate under a time-varying graph model for the network topology. Unlike prior works, the algorithm can operate even when normally-behaving leaders' vector messages do not exactly agree due to noise or perturbations, and it was proven that bounded errors between the normal leaders' vector messages results in bounded maximum error between normally-behaving leaders' and followers' accepted values. It was demonstrated that this algorithm could be used to resiliently propagate full-knowledge of time-varying trajectories in the form of Bezier curve and timing law parameters from a set of leaders to all normally-behaving followers in a network.

Chapter 5 presented a novel framework for resilient safety of multi-agent, distributed systems having sampled data dynamics using Control Barrier Function (CBF) techniques. Unlike prior work which typically assumes continuous-time control inputs and the complete absence of adversarial behavior, we presented conditions under which normally-behaving agents applying zero-order-hold (ZOH) inputs are able to maintain safety despite the actions of an adversarial set of agents seeking to violate safety conditions. Our analysis considered nonlinear control-affine sampled-data dynamics with disturbances, and we presented a computationally tractable convex quadratic programming formulation for normally-behaving agents to compute safety-preserving, bounded control inputs in a distributed manner. The results presented in this chapter are the first to consider the presence of adversarial agents in a CBF setting, and the first to consider multi-agent CBFs having high relative degree with respect to system dynamics in a sampled-data setting.

6.2 Future Work

There are several opportunities for future work building upon the results in this dissertation. With respect to MILP techniques for determining r and (r, s)-robustness of digraphs, as mentioned previously one of the most promising directions is studying whether the structure of the Laplacian matrix can be exploited to provide provably tractable methods for approximating a lower bound on the values r and s. This could include using deep learning techniques to identify incumbent sets with as low as an objective value as possible in order to more rapidly prune the search space when applying branch-and-bound algorithms. Another possibility may be to explore whether there exist operations on the Laplacian matrix that predictably change the optimal value. If such operations exist, then given a Laplacian L_1 of known maximum r_1 and s_1 values and a target Laplacian of unknown r_2 and s_2 values, it might be possible to determing r_2, s_2 via the transformations required to transform the matrix L_1 into L_2. Finally, perhaps the most straightforward direction for future work would be to explore the vast existing literature on MILP solving methods to identify any methods that would be particularly effective on the robustness determination MILP formulation.

With respect to resilient safety preservation via Control Barrier Function (CBF) methods, the method proposed in this dissertation, like the majority of CBF methods proposed in prior literature, is myopic in the sense that only the current velocity is considered in the conditions for safety and the convex quadratic programs computing the control input. In particular, the method does not take into account any projections over a future time window and does not take into account any information (if available) about the forward reachable sets of the agents. An interesting question is whether methods from techniques such as Model Predictive Control could be used in an adversarial CBF setting to provide better guarantees on the forward nonemptiness of the feasible sets of safety-preserving control inputs. Another direction for further research is to study methods for identifying adversarial agents from observations. The proposed method assumes that the identities of adversarial agents are known, but does not discuss how the adversaries are identified in the first place. Given sufficiently accurate estimates of other agents' states and control inputs, it may be possible to use the Nagumo's Theorem inequality condition to identify agents that are not applying sufficient control effort towards minimizing the left-hand side term, and therefore can be classified as behaving adversarially.

6.2.1 Final Discussion

Designing control algorithms resilient to faults and adversarial attacks is quite often a difficult endeavor. Many of the problems associated with adversarial behavior are combinatorial or NP-hard in nature, due to the inherent uncertainty of which agents will become faulty or adversarial. One focus that would be beneficial for future work would be studying approximation algorithms

that could potentially find sufficiently satisfactory approximate solutions with lower complexity.

In addition, a pervading theme of this dissertation is the necessity of conservatism due to lack of information. When agents lack information on the network structure and the identity of adversaries and leaders, they must apply conservative measures to ensure that accomplishment of control objectives is ensured despite all possible adversarial attacks. An interesting question to consider is how the introduction of addition information to normal agents would allow this conservatism to be relaxed. As a simple example, methods to more intelligently identify adversarial agents in MSR-type scenarios would possibly relax the requirement for agents to filter out much of their information at each time step. Non-local knowledge of the network structure would also allow the graph theoretic requirements for MSR-type consensus algorithms to be relaxed, as demonstrated in prior computer science literature. In addition, the conservatism of the resilient safety methods in CBF scenarios is due to the assumed lack of knowledge of agents' forward reachable sets. The introduction of computationally efficient approximation for these reachable sets might allow for this conservatism to be relaxed, and for stronger guarantees on safety to be established.

Finally, due to time constraints this dissertation was not able to consider some aspects of practical systems such as stochasticity in communication topologies and in agents' signals, proximity-limited communication models, chattering in signum-based controllers, incorporation of state estimators rather than direct access to system states, and imperfect knowledge of the system dynamics models, among others. Such practical considerations are certainly a promising avenue for future research enabling more accurate implementations in real-world control systems.

To conclude, the study of resilience in multi-agent autonomous systems is a critically important area of research for the modern era. There are innumerable disciplines and angles from which this area can be approached, but advances on this front will serve to solve many of the challenges that multi-agent autonomous systems will face in the 21st century and beyond.

BIBLIOGRAPHY

[1] Berkun, S., *The myths of innovation*, O'Reilly Media, Inc., 2010.

[2] LeBlanc, H. J., Zhang, H., Koutsoukos, X., and Sundaram, S., "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, Vol. 31, No. 4, 2013, pp. 766–781.

[3] Dahlqvist, F., Rajko, A., and Shulman, J., "Growing Opportunities in the Internet of Things," July 2019, https://web.archive.org/web/20201218193429/https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/growing-opportunities-in-the-internet-of-things, Accessed 01-08-2020.

[4] Bethke, B., How, J., and Vian, J., "Multi-UAV persistent surveillance with communication constraints and health mangement," *AIAA Guidance, Navigation, and Control Conference*, 2009, p. 5654.

[5] Schmuck, P. and Chli, M., "Multi-UAV collaborative monocular SLAM," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3863–3870.

[6] Frew, E. W., Argrow, B., Lawrence, D., Elston, J., and Stachura, M., "Unmanned aircraft systems for communication and atmospheric sensing missions," *2013 American Control Conference*, 2013, pp. 1482–1487.

[7] Zhan, E., "3,051 Drones Create Spectacular Record-Breaking Light Show in China," October 2020, https://web.archive.org/web/20201217100112/https://www.guinnessworldrecords.com/news/commercial/2020/10/3051-drones-create-spectacular-record-breaking-light-show-in-china, Accessed 01-08-2020.

[8] Atherton, K., "LOCUST Launcher Fires a Swarm of Navy Drones," May 2016, https://web.archive.org/web/20201022160225/https://www.popsci.com/navys-locust-launcher-fires-swarm-drones/, Accessed 01-08-2020.

[9] Simon, M., "Inside the Amazon Warehouse Where Humans and Machines Become One," June 2019,

https://web.archive.org/web/20201225104944/https://www.wired.com/story/amazon-warehouse-robots/, Accessed 01-08-2020.

[10] Vincent, J., "Welcome to the Automated Warehouse of the Future," May 2018, https://web.archive.org/web/20201215073541/https://www.theverge.com/2018/5/8/17331250/automated-warehouses-jobs-ocado-andover-amazon, Accessed 01-08-2020.

[11] "Volvo Trucks Provides Autonomous Transport Solution to Brønnøy Kalk AS," November 2018, https://web.archive.org/web/20190604165140/https://www.volvogroup.com/en-en/news/2018/nov/news-3126261.html, Accessed 01-08-2020.

[12] "U.S. States Are Allowing Automated Follower Truck Platooning While The Swedes May Lead In Europe," https://web.archive.org/web/20210108225649/https://www.forbes.com/sites/richardbishop1/2020/05/02/us-states-are-allowing-automated-follower-truck-platooning-while-the-swedes-may-lead-in-europe/?sh=d65d64cd7e8d, Accessed 01-08-2020.

[13] "M City: Fast Facts," 2020, https://mcity.umich.edu/our-vision/fast-facts/, Accessed 01-08-2020.

[14] "Partnership Against Cybercrime," 2020, https://web.archive.org/web/20201231023006/https://www.weforum.org/projects/partnership-against-cybercime, Accessed 01-13-2020.

[15] "Cybercrime To Cost The World $10.5 Trillion Annually By 2025," November 2020, https://web.archive.org/web/20201224190034/https://cybersecurityventures.com/cybercrime-damage-costs-10-trillion-by-2025/, Accessed 01-13-2020.

[16] Langner, R., "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, Vol. 9, No. 3, 2011, pp. 49–51.

[17] "Drone Crash in Iran Reveals Secret U.S. Surveillance Effort," December 2011, http://web.archive.org/web/20201213124921/https://www.nytimes.com/2011/12/08/world/middleeast/drone-crash-in-iran-reveals-secret-us-surveillance-bid.html, Accessed 01-13-2020.

[18] Schellekens, M., "Car hacking: Navigating the regulatory landscape," *Computer law & security review*, Vol. 32, No. 2, 2016, pp. 307–315.

[19] Jafarnejad, S., Codeca, L., Bronzi, W., Frank, R., and Engel, T., "A Car Hacking Experiment: When Connectivity Meets Vulnerability," *2015 IEEE Globecom Workshops (GC Wkshps)*, 2015, pp. 1–6.

[20] Shahani, A., "Tesla Model S Can Be Hacked, And Fixed (Which Is The Real News)," Aug 2015, https://web.archive.org/web/20201031083336/https://www.npr.org/sections/alltechconsidered/2015/08/06/429907506/tesla-model-s-can-be-hacked-and-fixed-which-is-the-real-news/, Accessed 01-08-2020.

[21] "How Russia Says It Swatted Down a Drone Swarm in Syria," http://web.archive.org/web/20201109003836/https://www.vice.com/en/article/43qbbw/russia-says-it-swatted-down-drone-swarm-syria-isis, Accessed 01-13-2020.

[22] Turek, J. and Shasha, D., "The many faces of consensus in distributed systems," *Computer*, Vol. 25, No. 6, 1992, pp. 8–17.

[23] Ren, W. and Beard, R. W., *Distributed consensus in multi-vehicle cooperative control*, Vol. 27, Springer, 2008.

[24] Mesbahi, M. and Egerstedt, M., *Graph theoretic methods in multiagent networks*, Vol. 33, Princeton University Press, 2010.

[25] Olfati-Saber, R., Fax, J. A., and Murray, R. M., "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, Vol. 95, No. 1, 2007, pp. 215–233.

[26] Gray, J. and Lamport, L., "Consensus on transaction commit," *ACM Transactions on Database Systems (TODS)*, Vol. 31, No. 1, 2006, pp. 133–160.

[27] Popek, G., Walker, B., Chow, J., Edwards, D., Kline, C., Rudisin, G., and Thiel, G., "LOCUS a network transparent, high reliability distributed system," *ACM SIGOPS Operating Systems Review*, Vol. 15, No. 5, 1981, pp. 169–177.

[28] Cristian, F., "Probabilistic clock synchronization," *Distributed Computing*, Vol. 3, 2005, pp. 146–158.

[29] Olfati-Saber, R. and Shamma, J. S., "Consensus filters for sensor networks and distributed sensor fusion," *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE, 2005, pp. 6698–6703.

[30] Mitra, A. and Sundaram, S., "Distributed observers for LTI systems," *IEEE Transactions on Automatic Control*, Vol. 63, No. 11, 2018, pp. 3689–3704.

[31] Açıkmeşe, B., Mandić, M., and Speyer, J. L., "Decentralized observers with consensus filters for distributed discrete-time linear systems," *Automatica*, Vol. 50, No. 4, 2014, pp. 1037–1052.

[32] Hui, Q., "Finite-time rendezvous algorithms for mobile autonomous agents," *IEEE Transactions on Automatic Control*, Vol. 56, No. 1, 2010, pp. 207–211.

[33] Cortes, J., Martinez, S., and Bullo, F., "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, Vol. 51, No. 8, 2006, pp. 1289–1298.

[34] Dimarogonas, D. V. and Kyriakopoulos, K. J., "On the Rendezvous Problem for Multiple Nonholonomic Agents," *IEEE Transactions on Automatic Control*, Vol. 52, No. 5, 2007, pp. 916–922.

[35] Fax, J. A. and Murray, R. M., "Information flow and cooperative control of vehicle formations," *IEEE transactions on automatic control*, Vol. 49, No. 9, 2004, pp. 1465–1476.

[36] Olfati-Saber, R., "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on automatic control*, Vol. 51, No. 3, 2006, pp. 401–420.

[37] Hu, J. and Feng, G., "Distributed tracking control of leader–follower multi-agent systems under noisy measurement," *Automatica*, Vol. 46, No. 8, 2010, pp. 1382–1387.

[38] Dörfler, F. and Bullo, F., "Synchronization in complex networks of phase oscillators: A survey," *Automatica*, Vol. 50, No. 6, 2014, pp. 1539–1564.

[39] Hatano, Y. and Mesbahi, M., "Agreement over random networks," *IEEE Transactions on Automatic Control*, Vol. 50, No. 11, 2005, pp. 1867–1872.

[40] Tahbaz-Salehi, A. and Jadbabaie, A., "A necessary and sufficient condition for consensus over random networks," *IEEE Transactions on Automatic Control*, Vol. 53, No. 3, 2008, pp. 791–795.

[41] Xiao, L., Boyd, S., and Kim, S.-J., "Distributed average consensus with least-mean-square deviation," *Journal of parallel and distributed computing*, Vol. 67, No. 1, 2007, pp. 33–46.

[42] Huang, M. and Manton, J. H., "Stochastic consensus seeking with noisy and directed inter-agent communication: Fixed and randomly varying topologies," *IEEE Transactions on Automatic Control*, Vol. 55, No. 1, 2009, pp. 235–241.

[43] Fischer, M. J., "The consensus problem in unreliable distributed systems (a brief survey)," *International conference on fundamentals of computation theory*, Springer, 1983, pp. 127–140.

[44] Lamport, L., Shostak, R., and Pease, M., "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, Vol. 4, No. 3, 1982, pp. 382–401.

[45] Dolev, D. et al., "The Byzantine generals strike again," *J. Algorithms*, Vol. 3, No. 1, 1982, pp. 14–30.

[46] Lamport, L., "The weak Byzantine generals problem," *Journal of the ACM (JACM)*, Vol. 30, No. 3, 1983, pp. 668–676.

[47] Rabin, M. O., "Randomized byzantine generals," *24th Annual Symposium on Foundations of Computer Science (sfcs 1983)*, IEEE, 1983, pp. 403–409.

[48] Lamport, L. and Melliar-Smith, P. M., "Byzantine clock synchronization," *Proceedings of the third annual ACM symposium on Principles of distributed computing*, 1984, pp. 68–74.

[49] Fischer, M. J., Lynch, N. A., and Paterson, M. S., "Impossibility of distributed consensus with one faulty process," *Journal of the ACM (JACM)*, Vol. 32, No. 2, 1985, pp. 374–382.

[50] Vaidya, N. H., Tseng, L., and Liang, G., "Iterative approximate byzantine consensus in arbitrary directed graphs," *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, ACM, 2012, pp. 365–374.

[51] Vaidya, N. H. and Garg, V. K., "Byzantine vector consensus in complete graphs," *Proceedings of the 2013 ACM symposium on Principles of distributed computing*, ACM, 2013, pp. 65–73.

[52] Tseng, L. and Vaidya, N., "Iterative approximate byzantine consensus under a generalized fault model," *International Conference on Distributed Computing and Networking*, Springer, 2013, pp. 72–86.

[53] Tseng, L. and Vaidya, N. H., "Asynchronous convex hull consensus in the presence of crash faults," *Proceedings of the 2014 ACM symposium on Principles of distributed computing*, ACM, 2014, pp. 396–405.

[54] Lamport, L., "The part-time parliament," *Concurrency: the Works of Leslie Lamport*, 2019, pp. 277–317.

[55] Lamport, L. et al., "Paxos made simple," *ACM Sigact News*, Vol. 32, No. 4, 2001, pp. 18–25.

[56] Lamport, L., "Fast paxos," *Distributed Computing*, Vol. 19, No. 2, 2006, pp. 79–103.

[57] Lamport, L., "Byzantizing Paxos by refinement," *International Symposium on Distributed Computing*, Springer, 2011, pp. 211–224.

[58] Lamport, L. and Massa, M., "Cheap paxos," *International Conference on Dependable Systems and Networks, 2004*, IEEE, 2004, pp. 307–314.

[59] Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y., and Kim, D. I., "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, Vol. 7, 2019, pp. 22328–22370.

[60] Tschorsch, F. and Scheuermann, B., "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, Vol. 18, No. 3, 2016, pp. 2084–2123.

[61] Miller, A. and LaViola Jr, J. J., "Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin," *University of Central Florida Tech. Report CS-TR-14-01 (accessed 5 June 2019) https://socrates1024. s3. amazonaws. com/consensus. pdf*, 2014.

[62] Garay, J., Kiayias, A., and Leonardos, N., "The bitcoin backbone protocol: Analysis and applications," *Annual international conference on the theory and applications of cryptographic techniques*, Springer, 2015, pp. 281–310.

[63] Nakamoto, S., "Bitcoin: A peer-to-peer electronic cash system," Tech. rep., Manubot, 2019.

[64] Mukhopadhyay, U., Skjellum, A., Hambolu, O., Oakley, J., Yu, L., and Brooks, R., "A brief survey of cryptocurrency systems," *2016 14th annual conference on privacy, security and trust (PST)*, IEEE, 2016, pp. 745–752.

[65] Ølnes, S., Ubacht, J., and Janssen, M., "Blockchain in government: Benefits and implications of distributed ledger technology for information sharing," 2017.

[66] Kuo, T.-T., Kim, H.-E., and Ohno-Machado, L., "Blockchain distributed ledger technologies for biomedical and health care applications," *Journal of the American Medical Informatics Association*, Vol. 24, No. 6, 2017, pp. 1211–1220.

[67] Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., and Wang, F.-Y., "Blockchain-enabled smart contracts: architecture, applications, and future trends," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 49, No. 11, 2019, pp. 2266–2277.

[68] Li, X., Jiang, P., Chen, T., Luo, X., and Wen, Q., "A survey on the security of blockchain systems," *Future Generation Computer Systems*, Vol. 107, 2020, pp. 841–853.

[69] Koo, C.-Y., "Broadcast in Radio Networks Tolerating Byzantine Adversarial Behavior," *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing*, PODC '04, Association for Computing Machinery, New York, NY, USA, 2004, p. 275–282.

[70] Koo, C.-Y., Bhandari, V., Katz, J., and Vaidya, N. H., "Reliable broadcast in radio networks: The bounded collision case," *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, ACM, 2006, pp. 258–264.

[71] Zhang, H. and Sundaram, S., "Robustness of information diffusion algorithms to locally bounded adversaries," *American Control Conference (ACC), 2012*, IEEE, 2012, pp. 5855–5861.

[72] Litsas, C., Pagourtzis, A., and Sakavalas, D., "A graph parameter that matches the resilience of the certified propagation algorithm," *International Conference on Ad-Hoc Networks and Wireless*, Springer, 2013, pp. 269–280.

[73] Pagourtzis, A., Panagiotakos, G., and Sakavalas, D., "Reliable broadcast with respect to topology knowledge," *Distributed Computing*, Vol. 30, No. 2, 2017, pp. 87–102.

[74] Kieckhafer, R. M. and Azadmanesh, M. H., "Reaching approximate agreement with mixed-mode faults," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 5, No. 1, 1994, pp. 53–63.

[75] LeBlanc, H. J., Zhang, H., Koutsoukos, X. D., and Sundaram, S., "Resilient Asymptotic Consensus in Robust Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 31, No. 4, 2013, pp. 766–781.

[76] Dibaji, S. M. and Ishii, H., "Resilient consensus of second-order agent networks: Asynchronous update rules with delays," *Automatica*, Vol. 81, 2017, pp. 123–132.

[77] Dibaji, S. M. and Ishii, H., "Consensus of second-order multi-agent systems in the presence of locally bounded faults," *Systems & Control Letters*, Vol. 79, 2015, pp. 23–29.

[78] Dibaji, S. M., Ishii, H., and Tempo, R., "Resilient randomized quantized consensus," *IEEE Transactions on Automatic Control*, Vol. 63, No. 8, 2017, pp. 2508–2522.

[79] Saldana, D., Prorok, A., Sundaram, S., Campos, M. F., and Kumar, V., "Resilient consensus for time-varying networks of dynamic agents," *American Control Conference (ACC), 2017*, IEEE, 2017, pp. 252–258.

[80] Wang, Y. and Ishii, H., "Resilient Consensus Through Event-Based Communication," *IEEE Transactions on Control of Network Systems*, Vol. 7, 2020, pp. 471–482.

[81] Wang, Y. and Ishii, H., "An event-triggered approach to quantized resilient consensus," *International Journal of Robust and Nonlinear Control*, Vol. 30, No. 11, 2020, pp. 4188–4204.

[82] Wang, Y. and Ishii, H., "A Distributed Model Predictive Scheme for Resilient Consensus with Input Constraints," *2019 IEEE Conference on Control Technology and Applications (CCTA)*, 2019, pp. 349–354.

[83] LeBlanc, H. J., Zhang, H., Sundaram, S., and Koutsoukos, X., "Resilient continuous-time consensus in fractional robust networks," *2013 American Control Conference*, IEEE, 2013, pp. 1237–1242.

[84] LeBlanc, H. J. and Koutsoukos, X., "Resilient first-order consensus and weakly stable, higher order synchronization of continuous-time networked multi-agent systems," *IEEE Transactions on Control of Network Systems*, 2017.

[85] Öksüz, H. Y. and Akar, M., "Resilient Nonlinear Consensus in Continuous Time Networks," *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 3764–3769.

[86] Shang, Y., "Consensus of hybrid multi-agent systems with malicious nodes," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2019.

[87] Usevitch, J. and Panagou, D., "Resilient finite-time consensus: a discontinuous systems perspective," *2020 American Control Conference (ACC)*, IEEE, 2020, pp. 3285–3290.

[88] Wu, Y. and He, X., "Secure consensus control for multiagent systems with attacks and communication delays," *IEEE/CAA Journal of Automatica Sinica*, Vol. 4, No. 1, 2017, pp. 136–142.

[89] Li, Z. and Ding, Z., "Robust Cooperative Guidance Law for Simultaneous Arrival," *IEEE Transactions on Control Systems Technology*, , No. 99, 2018, pp. 1–8.

[90] Sundaram, S. and Gharesifard, B., "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, 2018.

[91] Kikuya, Y., Dibaji, S. M., and Ishii, H., "Fault tolerant clock synchronization over unreliable channels in wireless sensor networks," *IEEE Transactions on Control of Network Systems*, 2017.

[92] Dibaji, S. M., Ishii, H., and Tempo, R., "Resilient randomized quantized consensus," *IEEE Transactions on Automatic Control*, Vol. 63, No. 8, 2018, pp. 2508–2522.

[93] Wang, Y. and Ishii, H., "An event-triggered approach to quantized resilient consensus," *International Journal of Robust and Nonlinear Control*, Vol. 30, 2020, pp. 4188–4204.

[94] Fiore, D. and Russo, G., "Resilient consensus for multi-agent systems subject to differential privacy requirements," *Automatica*, Vol. 106, 2019, pp. 18 – 26.

[95] Fiore, D. and Russo, G., "Resilient and private consensus in multi-agent systems," *2019 18th European Control Conference (ECC)*, 2019, pp. 3478–3483.

[96] LeBlanc, H. J. and Koutsoukos, X. D., "Algorithms for determining network robustness," *Proceedings of the 2nd ACM international conference on High confidence networked systems*, ACM, 2013, pp. 57–64.

[97] Zhang, H., Fata, E., and Sundaram, S., "A notion of robustness in complex networks," *IEEE Transactions on Control of Network Systems*, Vol. 2, No. 3, 2015, pp. 310–320.

[98] Guerrero-Bonilla, L., Prorok, A., and Kumar, V., "Formations for Resilient Robot Teams," *IEEE Robotics and Automation Letters*, Vol. 2, IEEE, 2017, pp. 841–848.

[99] Shahrivar, E. M., Pirani, M., and Sundaram, S., "Robustness and algebraic connectivity of random interdependent networks," *arXiv preprint arXiv:1508.03650*, 2015.

[100] Shahrivar, E. M., Pirani, M., and Sundaram, S., "Spectral and structural properties of random interdependent networks," *Automatica*, Vol. 83, 2017, pp. 234–242.

[101] Zhao, J., Yağan, O., and Gligor, V., "On connectivity and robustness in random intersection graphs," *IEEE Transactions on Automatic Control*, Vol. 62, No. 5, 2017, pp. 2121–2136.

[102] Usevitch, J. and Panagou, D., "r-Robustness and (r, s)-robustness of circulant graphs," *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 4416–4421.

[103] Guerrero-Bonilla, L., Saldana, D., and Kumar, V., "Design Guarantees for Resilient Robot Formations on Lattices," *IEEE Robotics and Automation Letters*, Vol. 4, No. 1, 2018, pp. 89–96.

[104] Saldaña, D., Guerrero-Bonilla, L., and Kumar, V., "Resilient backbones in hexagonal robot formations," *Distributed Autonomous Robotic Systems*, Springer, 2019, pp. 427–440.

[105] Wang, G., Xu, M., Wu, Y., Zheng, N., Xu, J., and Qiao, T., "Using Machine Learning for Determining Network Robustness of Multi-Agent Systems Under Attacks," *Pacific Rim International Conference on Artificial Intelligence*, Springer, 2018, pp. 491–498.

[106] Zhou, K. and Doyle, J. C., *Essentials of robust control*, Vol. 104, Prentice hall Upper Saddle River, NJ, 1998.

[107] Dorato, P., "A historical review of robust control," *IEEE Control Systems Magazine*, Vol. 7, No. 2, 1987, pp. 44–47.

[108] Abdallah, C., Dawson, D. M., Dorato, P., and Jamshidi, M., "Survey of robust control for rigid robots," *IEEE Control Systems Magazine*, Vol. 11, No. 2, 1991, pp. 24–30.

[109] Kwakernaak, H., "Robust control and Hinf-optimization—tutorial paper," *automatica*, Vol. 29, No. 2, 1993, pp. 255–273.

[110] Chen, C., Lewis, F. L., Xie, S., Modares, H., Liu, Z., Zuo, S., and Davoudi, A., "Resilient adaptive and Hinf controls of multi-agent systems under sensor and actuator faults," *Automatica*, Vol. 102, 2019, pp. 19 – 26.

[111] Pequito, S., Ramos, G., Kar, S., Aguiar, A. P., and Ramos, J., "The robust minimal controllability problem," *Automatica*, Vol. 82, 2017, pp. 261 – 268.

[112] Ramos, G., Pequito, S., and Caleiro, C., "The robust minimal controllability problem for switched linear continuous-time systems," *2018 Annual American Control Conference (ACC)*, IEEE, 2018, pp. 210–215.

[113] Cárdenas, A. A., Amin, S., and Sastry, S., "Research Challenges for the Security of Control Systems." *HotSec*, 2008.

[114] Teixeira, A., Pérez, D., Sandberg, H., and Johansson, K. H., "Attack models and scenarios for networked control systems," *Proceedings of the 1st international conference on High Confidence Networked Systems*, 2012, pp. 55–64.

[115] Pasqualetti, F., Dörfler, F., and Bullo, F., "Cyber-physical attacks in power networks: Models, fundamental limitations and monitor design," *2011 50th IEEE Conference on Decision and Control and European Control Conference*, IEEE, 2011, pp. 2195–2201.

[116] Pasqualetti, F., Bicchi, A., and Bullo, F., "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, Vol. 57, No. 1, 2011, pp. 90–104.

[117] Pasqualetti, F., Dörfler, F., and Bullo, F., "Attack detection and identification in cyber-physical systems," *IEEE transactions on automatic control*, Vol. 58, No. 11, 2013, pp. 2715–2729.

[118] Shames, I., Teixeira, A., Sandberg, H., and Johansson, K. H., "Distributed fault detection for interconnected second-order systems with applications to power networks," *First Workshop on Secure Control Systems (SCS), Stockholm, 2010*, 2010.

[119] Shames, I., Teixeira, A. M., Sandberg, H., and Johansson, K. H., "Distributed fault detection for interconnected second-order systems," *Automatica*, Vol. 47, No. 12, 2011, pp. 2757–2764.

[120] Teixeira, A., Shames, I., Sandberg, H., and Johansson, K. H., "Distributed fault detection and isolation resilient to network model uncertainties," *IEEE transactions on cybernetics*, Vol. 44, No. 11, 2014, pp. 2024–2037.

[121] Anguluri, R., Katewa, V., and Pasqualetti, F., "Centralized Versus Decentralized Detection of Attacks in Stochastic Interconnected Systems," *IEEE Transactions on Automatic Control*, 2019.

[122] Fawzi, H., Tabuada, P., and Diggavi, S., "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Transactions on Automatic control*, Vol. 59, No. 6, 2014, pp. 1454–1467.

[123] Candes, E. J. and Tao, T., "Decoding by linear programming," *IEEE transactions on information theory*, Vol. 51, No. 12, 2005, pp. 4203–4215.

[124] Pajic, M., Lee, I., and Pappas, G. J., "Attack-resilient state estimation for noisy dynamical systems," *IEEE Transactions on Control of Network Systems*, Vol. 4, No. 1, 2016, pp. 82–92.

[125] Chang, Y. H., Hu, Q., and Tomlin, C. J., "Secure estimation based Kalman filter for cyber–physical systems against sensor attacks," *Automatica*, Vol. 95, 2018, pp. 399–412.

[126] Shoukry, Y., Nuzzo, P., Puggelli, A., Sangiovanni-Vincentelli, A. L., Seshia, S. A., and Tabuada, P., "Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach," *IEEE Transactions on Automatic Control*, Vol. 62, No. 10, 2017, pp. 4917–4932.

[127] Manshaei, M. H., Zhu, Q., Alpcan, T., Bacşar, T., and Hubaux, J.-P., "Game theory meets network security and privacy," *ACM Computing Surveys (CSUR)*, Vol. 45, No. 3, 2013, pp. 1–39.

[128] Zhu, Q. and Basar, T., "Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: games-in-games principle for optimal cross-layer resilient control systems," *IEEE Control Systems Magazine*, Vol. 35, No. 1, 2015, pp. 46–65.

[129] Nagumo, M., "Über die lage der integralkurven gewöhnlicher differentialgleichungen," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, Vol. 24, 1942, pp. 551–559.

[130] Isaacs, R., *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*, Courier Corporation, 1999.

[131] Friedman, A., *Differential games*, Courier Corporation, 2013.

[132] Botchkarev, O. and Tripakis, S., "Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations," *International Workshop on Hybrid Systems: Computation and Control*, Springer, 2000, pp. 73–88.

[133] Chutinan, A. and Krogh, B. H., "Computational techniques for hybrid system verification," *IEEE transactions on automatic control*, Vol. 48, No. 1, 2003, pp. 64–75.

[134] Chutinan, A. and Krogh, B. H., "Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations," *International workshop on hybrid systems: computation and control*, Springer, 1999, pp. 76–90.

[135] Tomlin, C. J., Mitchell, I., Bayen, A. M., and Oishi, M., "Computational techniques for the verification of hybrid systems," *Proceedings of the IEEE*, Vol. 91, No. 7, 2003, pp. 986–1001.

[136] Dabadie, C., Kaynama, S., and Tomlin, C. J., "A practical reachability-based collision avoidance algorithm for sampled-data systems: Application to ground robots," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 4161–4168.

[137] Bansal, S., Chen, M., Herbert, S., and Tomlin, C. J., "Hamilton-Jacobi reachability: A brief overview and recent advances," *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 2242–2253.

[138] Mitchell, I. M. and Tomlin, C. J., "Overapproximating reachable sets by Hamilton-Jacobi projections," *journal of Scientific Computing*, Vol. 19, No. 1-3, 2003, pp. 323–346.

[139] Girard, A. and Le Guernic, C., "Zonotope/hyperplane intersection for hybrid systems reachability analysis," *International Workshop on Hybrid Systems: Computation and Control*, Springer, 2008, pp. 215–228.

[140] Kochdumper, N. and Althoff, M., "Sparse polynomial zonotopes: A novel set representation for reachability analysis," *IEEE Transactions on Automatic Control*, 2020.

[141] Althoff, M. and Krogh, B. H., "Zonotope bundles for the efficient computation of reachable sets," *2011 50th IEEE conference on decision and control and European control conference*, IEEE, 2011, pp. 6814–6821.

[142] Althoff, M. and Krogh, B. H., "Reachability analysis of nonlinear differential-algebraic systems," *IEEE Transactions on Automatic Control*, Vol. 59, No. 2, 2013, pp. 371–383.

[143] Prajna, S. and Jadbabaie, A., "Safety verification of hybrid systems using barrier certificates," *International Workshop on Hybrid Systems: Computation and Control*, Springer, 2004, pp. 477–492.

[144] Prajna, S., "Barrier certificates for nonlinear model validation," *Automatica*, Vol. 42, No. 1, 2006, pp. 117–126.

[145] Prajna, S., "Barrier certificates for nonlinear model validation," *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, Vol. 3, IEEE, 2003, pp. 2884–2889.

[146] Prajna, S., Jadbabaie, A., and Pappas, G. J., "A framework for worst-case and stochastic safety verification using barrier certificates," *IEEE Transactions on Automatic Control*, Vol. 52, No. 8, 2007, pp. 1415–1428.

[147] Prajna, S. and Rantzer, A., "On the necessity of barrier certificates," *IFAC Proceedings Volumes*, Vol. 38, No. 1, 2005, pp. 526–531.

[148] Panagou, D., Stipanović, D. M., and Voulgaris, P. G., "Distributed coordination control for multi-robot networks using Lyapunov-like barrier functions," *IEEE Transactions on Automatic Control*, Vol. 61, No. 3, 2015, pp. 617–632.

[149] Panagou, D., Stipanovič, D. M., and Voulgaris, P. G., "Multi-objective control for multi-agent systems using Lyapunov-like barrier functions," *52nd IEEE Conference on Decision and Control*, IEEE, 2013, pp. 1478–1483.

[150] Han, D. and Panagou, D., "Robust Multitask Formation Control via Parametric Lyapunov-Like Barrier Functions," *IEEE Transactions on Automatic Control*, Vol. 64, No. 11, 2019, pp. 4439–4453.

[151] Hernández-Martínez, E. G. and Aranda-Bricaire, E., *Convergence and collision avoidance in formation control: A survey of the artificial potential functions approach*, INTECH Open Access Publisher Rijeka, Croatia, 2011.

[152] Merz, A. W., "The homicidal chauffeur," *AIAA Journal*, Vol. 12, No. 3, 1974, pp. 259–260.

[153] Bopardikar, S. D., Bullo, F., and Hespanha, J. P., "A cooperative homicidal chauffeur game," *Automatica*, Vol. 45, No. 7, 2009, pp. 1771–1777.

[154] Exarchos, I., Tsiotras, P., and Pachter, M., "On the suicidal pedestrian differential game," *Dynamic Games and Applications*, Vol. 5, No. 3, 2015, pp. 297–317.

[155] Sgall, J., "Solution of David Gale's lion and man problem," *Theoretical Computer Science*, Vol. 259, No. 1-2, 2001, pp. 663–670.

[156] Mitchell, I. M., Bayen, A. M., and Tomlin, C. J., "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, Vol. 50, No. 7, 2005, pp. 947–957.

[157] Oyler, D. W., Kabamba, P. T., and Girard, A. R., "Dominance in pursuit-evasion games with uncertainty," *2015 54th IEEE Conference on Decision and Control (CDC)*, IEEE, 2015, pp. 5859–5864.

[158] Oyler, D. W. and Girard, A. R., "Dominance regions in the homicidal chauffeur problem," *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 2494–2499.

[159] Oyler, D. W., Kabamba, P. T., and Girard, A. R., "Pursuit–evasion games in the presence of obstacles," *Automatica*, Vol. 65, 2016, pp. 1–11.

[160] Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P., "Control barrier functions: Theory and applications," *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 3420–3431.

[161] Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P., "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, Vol. 62, No. 8, 2016, pp. 3861–3876.

[162] Wieland, P. and Allgöwer, F., "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, Vol. 40, No. 12, 2007, pp. 462–467.

[163] Lin, Y. and Sontag, E. D., "A universal formula for stabilization with bounded controls," *Systems & Control Letters*, Vol. 16, No. 6, 1991, pp. 393–397.

[164] Ames, A. D., Grizzle, J. W., and Tabuada, P., "Control barrier function based quadratic programs with application to adaptive cruise control," *53rd IEEE Conference on Decision and Control*, IEEE, 2014, pp. 6271–6278.

[165] Hsu, S.-C., Xu, X., and Ames, A. D., "Control barrier function based quadratic programs with application to bipedal robotic walking," *2015 American Control Conference (ACC)*, IEEE, 2015, pp. 4542–4548.

[166] Romdlony, M. Z. and Jayawardhana, B., "Uniting control Lyapunov and control barrier functions," *53rd IEEE Conference on Decision and Control*, IEEE, 2014, pp. 2293–2298.

[167] Romdlony, M. Z. and Jayawardhana, B., "Stabilization with guaranteed safety using control Lyapunov–barrier function," *Automatica*, Vol. 66, 2016, pp. 39–47.

[168] Srinivasan, M., Coogan, S., and Egerstedt, M., "Control of multi-agent systems with finite time control barrier certificates and temporal logic," *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 1991–1996.

[169] Chen, Y., Singletary, A., and Ames, A. D., "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: a control barrier function approach," *IEEE Control Systems Letters*, Vol. 5, No. 1, 2020, pp. 127–132.

[170] Borrmann, U., Wang, L., Ames, A. D., and Egerstedt, M., "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, Vol. 48, No. 27, 2015, pp. 68–73.

[171] Garg, K. and Panagou, D., "Control-lyapunov and control-barrier functions based quadratic program for spatio-temporal specifications," *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 1422–1429.

[172] Lindemann, L. and Dimarogonas, D. V., "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, Vol. 3, No. 1, 2018, pp. 96–101.

[173] Lindemann, L. and Dimarogonas, D. V., "Robust control for signal temporal logic specifications using discrete average space robustness," *Automatica*, Vol. 101, 2019, pp. 377–387.

[174] Lindemann, L. and Dimarogonas, D. V., "Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks," *IEEE Control Systems Letters*, Vol. 3, No. 3, 2019, pp. 757–762.

[175] Nguyen, Q., Hereid, A., Grizzle, J. W., Ames, A. D., and Sreenath, K., "3d dynamic walking on stepping stones with control barrier functions," *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 827–834.

[176] Cortez, W. S., Oetomo, D., Manzie, C., and Choong, P., "Control barrier functions for mechanical systems: Theory and application to robotic grasping," *IEEE Transactions on Control Systems Technology*, 2019.

[177] Singletary, A., Chen, Y., and Ames, A. D., "Control Barrier Functions for Sampled-Data Systems with Input Delays," *arXiv preprint arXiv:2005.06418*, 2020.

[178] Jankovic, M., "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, Vol. 96, 2018, pp. 359–367.

[179] Kolathaya, S. and Ames, A. D., "Input-to-state safety with control barrier functions," *IEEE control systems letters*, Vol. 3, No. 1, 2018, pp. 108–113.

[180] Usevitch, J. and Panagou, D., "Resilient Leader-Follower Consensus to Arbitrary Reference Values in Time-Varying Graphs," *IEEE Transactions on Automatic Control*, Vol. 65, No. 4, 2019, pp. 1755–1762.

[181] Usevitch, J. and Panagou, D., "Resilient leader-follower consensus to arbitrary reference values," *2018 Annual American Control Conference (ACC)*, IEEE, 2018, pp. 1292–1298.

[182] Usevitch, J., Garg, K., and Panagou, D., "Finite-time resilient formation control with bounded inputs," *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 2567–2574.

[183] Usevitch, J. and Panagou, D., "Resilient Leader-Follower Consensus with Time-Varying Leaders in Discrete-Time Systems," *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 5432–5437.

[184] Usevitch, J. and Panagou, D., "Determining r- and (r,s)-robustness of digraphs using mixed integer linear programming," *Automatica*, Vol. 111, 2020, pp. 108586.

[185] Usevitch, J. and Panagou, D., "Determining r- and (r,s)-Robustness of Digraphs Using Mixed Integer Linear Programming," *arXiv preprint arXiv:1901.11000*, 2019.

[186] Usevitch, J. and Panagou, D., "Determining r-Robustness of Digraphs Using Mixed Integer Linear Programming," *2019 Annual American Control Conference (ACC)*, IEEE, 2019.

[187] Usevitch, J. and Panagou, D., "Adversarially Resilient Control Barrier Functions in Sampled-Data Systems," *2021 American Control Conference (ACC)*, IEEE, 2021, To appear.

[188] Usevitch, J. and Panagou, D., "Adversarial Resilience Using Control Barrier Functions with High Relative Degree in Sampled-Data Systems," 2021, Submitted.

[189] Boyd, S. and Vandenberghe, L., *Convex optimization*, Cambridge university press, 2004.

[190] LeBlanc, H. J., Zhang, H., Sundaram, S., and Koutsoukos, X., "Resilient continuous-time consensus in fractional robust networks," *2013 American Control Conference*, 6 2013, pp. 1237–1242.

[191] Mitra, A. and Sundaram, S., "Secure distributed observers for a class of linear time invariant systems in the presence of byzantine adversaries," *Decision and Control (CDC), 2016 IEEE 55th Conference on*, IEEE, 2016, pp. 2709–2714.

[192] Zhang, H. and Sundaram, S., "A simple median-based resilient consensus algorithm," *2012 50th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2012*, 2012, pp. 1734–1741.

[193] LeBlanc, H. J. and Hassan, F., "Resilient distributed parameter estimation in heterogeneous time-varying networks," *Proceedings of the 3rd international conference on High confidence networked systems*, ACM, 2014, pp. 19–28.

[194] Mitra, A. and Sundaram, S., "Byzantine-resilient distributed observers for LTI systems," *Automatica*, Vol. 108, 2019, pp. 108487.

[195] Mitra, A. and Sundaram, S., "Secure Distributed State Estimation of an LTI System Over Time-Varying Networks and Analog Erasure Channels," *2018 Annual American Control Conference (ACC)*, June 2018, pp. 6578–6583.

[196] Franceschelli, M., Giua, A., and Pisano, A., "Finite-time consensus on the median value by discontinuous control," *2014 American Control Conference*, IEEE, 2014, pp. 946–951.

[197] Franceschelli, M., Giua, A., and Pisano, A., "Finite-Time Consensus on the Median Value With Robustness Properties," *IEEE Transactions on Automatic Control*, Vol. 62, No. 4, April 2017, pp. 1652–1667.

[198] Cortes, J., "Discontinuous dynamical systems," *IEEE Control Systems Magazine*, Vol. 28, No. 3, 2008, pp. 36–73.

[199] Bhat, S. P. and Bernstein, D. S., "Finite-time stability of continuous autonomous systems," *SIAM Journal on Control and Optimization*, Vol. 38, No. 3, 2000, pp. 751–766.

[200] Mitra, A., Richards, J. A., Bagchi, S., and Sundaram, S., "Resilient distributed state estimation with mobile agents: overcoming Byzantine adversaries, communication losses, and intermittent measurements," *Autonomous Robots*, 2018, pp. 1–26.

[201] Wang, L. and Xiao, F., "Finite-time consensus problems for networks of dynamic agents," *IEEE Transactions on Automatic Control*, Vol. 55, No. 4, 2010, pp. 950–955.

[202] Garg, K. and Panagou, D., "New Results on Finite-Time Stability: Geometric Conditions and Finite-Time Controllers," *2018 Annual American Control Conference (ACC)*, June 2018, pp. 442–447.

[203] Olfati-Saber, R. and Murray, R. M., "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, Vol. 49, No. 9, 2004, pp. 1520–1533.

[204] Chen, G., Lewis, F. L., and Xie, L., "Finite-time distributed consensus via binary control protocols," *Automatica*, Vol. 47, No. 9, 2011, pp. 1962–1968.

[205] Clarke, F. H., *Optimization and nonsmooth analysis*, Vol. 5, Siam, 1990.

[206] Shevitz, D. and Paden, B., "Lyapunov stability theory of nonsmooth systems," *IEEE Transactions on automatic control*, Vol. 39, No. 9, 1994, pp. 1910–1914.

[207] Bacciotti, A. and Ceragioli, F., "Nonsmooth Lyapunov functions and discontinuous Carathéodory systems," *IFAC Proceedings Volumes*, Vol. 37, No. 13, 2004, pp. 841–845.

[208] Filippov, A. F., *Differential equations with discontinuous righthand sides: control systems*, Vol. 18, Springer Science & Business Media, 2013.

[209] Khalil, H. K., *Nonlinear systems*, Vol. 3, Prentice hall Upper Saddle River, NJ, 2002.

[210] Utkin, V. and Lee, H., "Chattering problem in sliding mode control systems," *International Workshop on Variable Structure Systems, 2006. VSS'06.*, IEEE, 2006, pp. 346–350.

[211] Bartolini, G., "Chattering phenomena in discontinuous control systems," *International journal of systems science*, Vol. 20, No. 12, 1989, pp. 2471–2481.

[212] Lee, H. and Utkin, V. I., "Chattering suppression methods in sliding mode control systems," *Annual reviews in control*, Vol. 31, No. 2, 2007, pp. 179–188.

[213] Cichoń, J., Kharazishvili, A., and Weglorz, B., "On sets of Vitali's type," *Proceedings of the American Mathematical Society*, Vol. 118, No. 4, 1993, pp. 1243–1250.

[214] Solovay, R. M., "A model of set-theory in which every set of reals is Lebesgue measurable," *Annals of Mathematics*, 1970, pp. 1–56.

[215] Renganathan, V. and Summers, T., "Spoof resilient coordination for distributed multi-robot systems," *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, IEEE, 2017, pp. 135–141.

[216] LeBlanc, H. J., Zhang, H., Sundaram, S., and Koutsoukos, X., "Consensus of multi-agent networks in the presence of adversaries using only local information," *Proceedings of the 1st international conference on High Confidence Networked Systems*, ACM, 2012, pp. 1–10.

[217] LeBlanc, H. J. and Koutsoukos, X. D., "Low Complexity Resilient Consensus in Networked Multi-Agent Systems with Adversaries," *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, 2012, pp. 5–14.

[218] Vaidya, N. H., "Iterative Byzantine vector consensus in incomplete graphs," *International Conference on Distributed Computing and Networking*, Springer, 2014, pp. 14–28.

[219] LeBlanc, H. J., *Resilient cooperative control of networked multi-agent systems*, Vanderbilt University, 2012.

[220] LeBlanc, H. J., *Resilient Cooperative Control of Networked Multi-Agent Systems*, Ph.D. thesis, Vanderbilt University, 2012.

[221] Boesch, F. and Tindell, R., "Circulants and their connectivities," *Journal of Graph Theory*, Vol. 8, No. 4, 1984, pp. 487–499.

[222] Elspas, B. and Turner, J., "Graphs with circulant adjacency matrices," *Journal of Combinatorial Theory*, Vol. 9, No. 3, 1970, pp. 297–307.

[223] Tindell, R., "Connectivity of Cayley Digraphs," *Combinatorial network theory*, edited by D.-Z. Du and D. F. Hsu, Springer US, Boston, Massachusetts, 1996, pp. 41–64.

[224] Hamidoune, Y. O., "On the Connectivity of Cayley Digraphs," *European Journal of Combinatorics*, Vol. 5, No. 4, 12 1984, pp. 309–312.

[225] LeBlanc, H. J. and Koutsoukos, X., "Resilient asymptotic consensus in asynchronous robust networks," *2012 50th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2012*, 2012, pp. 1742–1749.

[226] Bertsimas, D. and Dunn, J., "Optimal classification trees," *Machine Learning*, Vol. 106, No. 7, 2017, pp. 1039–1082.

[227] Wolsey, L. A., "Mixed integer programming," *Wiley Encyclopedia of Computer Science and Engineering*, 2007, pp. 1–10.

[228] Bollobás, B., *Models of Random Graphs*, Cambridge Studies in Advanced Mathematics, Cambridge University Press, 2nd ed., 2001, p. 34–59.

[229] Koo, C.-Y., Bhandari, V., Katz, J., and Vaidya, N. H., "Reliable broadcast in radio networks: The bounded collision case," *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, 2006, pp. 258–264.

[230] Tseng, L., Wu, Y., Pan, H., Aloqaily, M., and Boukerche, A., "Reliable Broadcast in Networks with Trusted Nodes," *2019 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2019, pp. 1–6.

[231] Tseng, L., "Towards reliable broadcast in practical sensor networks," *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, IEEE, 2017, pp. 1–8.

[232] Loria, A., Dasdemir, J., and Jarquin, N. A., "Leader–follower formation and tracking control of mobile robots along straight paths," *IEEE transactions on control systems technology*, Vol. 24, No. 2, 2015, pp. 727–732.

[233] Meng, Z., Lin, Z., and Ren, W., "Leader–follower swarm tracking for networked Lagrange systems," *Systems and Control Letters*, Vol. 61, No. 1, 2012, pp. 117 – 126.

[234] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., *Robotics: modelling, planning and control*, Springer Science & Business Media, 2010.

[235] Mellinger, D., Shomin, M., Michael, N., and Kumar, V., "Cooperative grasping and transport using multiple quadrotors," *Distributed autonomous robotic systems*, Springer, 2013, pp. 545–558.

[236] Michael, N., Fink, J., and Kumar, V., "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, Vol. 30, No. 1, 2011, pp. 73–86.

[237] Rastgoftar, H. and Atkins, E. M., "Cooperative aerial payload transport guided by an in situ human supervisor," *IEEE Transactions on Control Systems Technology*, Vol. 27, No. 4, 2018, pp. 1452–1467.

[238] Beard, R. W., Lawton, J., and Hadaegh, F. Y., "A coordination architecture for spacecraft formation control," *IEEE Transactions on control systems technology*, Vol. 9, No. 6, 2001, pp. 777–790.

[239] Johnson, W. P., "The curious history of Faà di Bruno's formula," *The American mathematical monthly*, Vol. 109, No. 3, 2002, pp. 217–234.

[240] Mitra, A. and Sundaram, S., "Byzantine-resilient distributed observers for LTI systems," *Autom.*, Vol. 108, 2019.

[241] De Luca, A., Oriolo, G., and Vendittelli, M., "Control of wheeled mobile robots: An experimental overview," *Ramsete*, Springer, 2001, pp. 181–226.

[242] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S., "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, Vol. 12, No. 4, 2020, pp. 637–672.

[243] Usevitch, J. and Panagou, D., "r-Robustness and (r, s)-Robustness of Circulant Graphs," *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 4416–4421.

[244] Mitra, A., Richards, J. A., Bagchi, S., and Sundaram, S., "Resilient distributed state estimation with mobile agents: overcoming Byzantine adversaries, communication losses, and intermittent measurements," *Autonomous Robots*, Vol. 43, 2019, pp. 743–768.

[245] Panagou, D., Turpin, M., and Kumar, V., "Decentralized goal assignment and trajectory generation in multi-robot networks: A multiple lyapunov functions approach," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 6757–6762.

[246] Prautzsch, H., Boehm, W., and Paluszny, M., *Bézier and B-spline techniques*, Springer Science & Business Media, 2013.

[247] Sederberg, T. W., "Computer aided geometric design," 2012.

[248] Jin, W., Deng, C., Li, Y., and Liu, J., "Derivative bound estimations on rational conic Bézier curves," *Applied Mathematics and Computation*, Vol. 248, 2014, pp. 113–117.

[249] Selimovic, I., "New bounds on the magnitude of the derivative of rational Bézier curves and surfaces," *Computer Aided Geometric Design*, Vol. 22, No. 4, 2005, pp. 321–326.

[250] Lindemann, L. and Dimarogonas, D. V., "Decentralized control barrier functions for coupled multi-agent systems under signal temporal logic tasks," *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 89–94.

[251] Glotfelter, P., Cortés, J., and Egerstedt, M., "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE Control Systems Letters*, Vol. 1, No. 2, 2017, pp. 310–315.

[252] Xiao, W. and Belta, C., "Control barrier functions for systems with high relative degree," *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 474–479.

[253] Li, A., Wang, L., Pierpaoli, P., and Egerstedt, M., "Formally correct composition of coordinated behaviors using control barrier certificates," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3723–3729.

[254] Wang, L., Ames, A. D., and Egerstedt, M., "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, Vol. 33, No. 3, 2017, pp. 661–674.

[255] Glotfelter, P., Cortés, J., and Egerstedt, M., "Boolean composability of constraints and control synthesis for multi-robot systems via nonsmooth control barrier functions," *2018 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2018, pp. 897–902.

[256] Guerrero-Bonilla, L. and Kumar, V., "Realization of r-Robust Formations in the Plane Using Control Barrier Functions," *IEEE Control Systems Letters*, Vol. 4, No. 2, 2019, pp. 343–348.

[257] Wang, L., Ames, A. D., and Egerstedt, M., "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 3293–3298.

[258] Pickem, D., Glotfelter, P., Wang, L., Mote, M., Ames, A., Feron, E., and Egerstedt, M., "The robotarium: A remotely accessible swarm robotics research testbed," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1699–1706.

[259] Özkahraman, Ö. and Ögren, P., "Combining control barrier functions and behavior trees for multi-agent underwater coverage missions," *2020 59th IEEE Conference on Decision and Control (CDC)*, IEEE, 2020, pp. 5275–5282.

[260] Park, H. and Hutchinson, S. A., "Fault-tolerant rendezvous of multirobot systems," *IEEE transactions on robotics*, Vol. 33, No. 3, 2017, pp. 565–582.

[261] Saulnier, K., Saldana, D., Prorok, A., Pappas, G. J., and Kumar, V., "Resilient flocking for mobile robot teams," *IEEE Robotics and Automation letters*, Vol. 2, No. 2, 2017, pp. 1039–1046.

[262] Nguyen, Q. and Sreenath, K., "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 322–328.

[263] Blanchini, F., "Set invariance in control," *Automatica*, Vol. 35, No. 11, 1999, pp. 1747 – 1767.

[264] Clarke, F. H., Ledyaev, Y. S., Stern, R. J., and Wolenski, P. R., *Nonsmooth analysis and control theory*, Vol. 178, Springer Science & Business Media, 2008.

[265] Xu, X., Tabuada, P., Grizzle, J. W., and Ames, A. D., "Robustness of Control Barrier Functions for Safety Critical Control**This work is partially supported by the National Science Foundation Grants 1239055, 1239037 and 1239085." *IFAC-PapersOnLine*, Vol. 48, No. 27, 2015, pp. 54 – 61, Analysis and Design of Hybrid Systems ADHS.

[266] Usevitch, J., Garg, K., and Panagou, D., "Strong invariance using control barrier functions: A clarke tangent cone approach," *2020 59th IEEE Conference on Decision and Control (CDC)*, IEEE, 2020, pp. 2044–2049.

[267] Grüne, L. and Pannek, J., "Nonlinear model predictive control," *Nonlinear Model Predictive Control*, Springer, 2017.

[268] Garg, K., Arabi, E., and Panagou, D., "Prescribed-time control under spatiotemporal and input constraints: A QP based approach," *arXiv preprint arXiv:1906.10091*, 2019.

[269] Gauvin, J. and Dubeau, F., "Differential properties of the marginal function in mathematical programming," *Optimality and Stability in Mathematical Programming*, Springer, 1982, pp. 101–119.

[270] Black, M., Garg, K., and Panagou, D., "A Quadratic Program based Control Synthesis under Spatiotemporal Constraints and Non-vanishing Disturbances," *2020 IEEE 59th Conference on Decision and Control (CDC)*, IEEE, 2020.

[271] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B., "Julia: A fresh approach to numerical computing," *SIAM Review*, Vol. 59, No. 1, 2017, pp. 65–98.

[272] Revels, J., Lubin, M., and Papamarkou, T., "Forward-Mode Automatic Differentiation in Julia," *arXiv:1607.07892 [cs.MS]*, 2016.

[273] Chen, M., Herbert, S. L., Vashishtha, M. S., Bansal, S., and Tomlin, C. J., "Decomposition of reachable sets and tubes for a class of nonlinear systems," *IEEE Transactions on Automatic Control*, Vol. 63, No. 11, 2018, pp. 3675–3688.

[274] Liebenwein, L., Baykal, C., Gilitschenski, I., Karaman, S., and Rus, D., "Sampling-Based Approximation Algorithms for Reachability Analysis with Provable Guarantees," *Robotics: Science and Systems*, 2018.

[275] Wurts, J., Stein, J. L., and Ersal, T., "Collision imminent steering at high speed using nonlinear model predictive control," *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 8, 2020, pp. 8278–8289.