Nash Equilibrium Seeking Over Undirected Graphs Via Multi-Agent Agreement

by

Dian Gadjov

A thesis submitted in conformity with the requirements for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

© Copyright 2017 by Dian Gadjov
We consider the problem of distributed Nash equilibrium (NE) seeking over networks. In this setting, agents have limited information about the other players’ actions and are forced to communicate with neighbouring agents. We start with a continuous-time gradient-play dynamics, with perfect information, under a strictly monotone pseudo-gradient assumption. In the partial information case we modify the gradient-play dynamics between players by expanding the action space. We propose an augmented gradient-play dynamics in which players can only communicate locally with their neighbours to compute an estimate of the other players’ actions. We derive new dynamics based on the reformulation of the problem as a multi-agent coordination problem, over an undirected graph. We exploit the incremental passivity properties in the dynamics and show that a Laplacian feedback can be designed using relative estimates of their neighbours. We highlight that there is a trade-off between properties of the game and the communication graph.
Acknowledgements

I would like to thank my supervisor Professor Lacra Pavel. I truly appreciate her rigorous approach to proving theorems. She opened my eyes to the precision and nuance involved in writing papers. Her unique insight and guidance in both game theory and control theory was an invaluable resource that supported me throughout my research. I am extremely fortunate to have had this opportunity to work with her.

I thank the University of Toronto for fostering an environment where students can tackle problems at the edge of science/knowledge. The system’s control group at the University of Toronto has many brilliant minds who are willing to share their insight into the interesting challenges they face.

Last but not least, I would like to thank my family and friends for supporting me throughout my Masters. They have always been there for me and are always there when I need help. Thank you.
Contents

Acknowledgments iii

Contents iv

List of Figures vii

Nomenclature x

1 Introduction 1
   1.1 Motivation ................................................. 1
   1.2 Literature Review ........................................ 2
   1.3 Contributions ............................................. 3
   1.4 Organization .............................................. 5

2 Background 7
   2.1 Mathematical Notations .................................... 7
   2.2 Sets and Set Projection ...................................... 10
   2.3 Dynamical Systems and Stability .......................... 12
   2.4 Passivity .................................................. 19
   2.5 Graph Theory ............................................... 23
   2.6 Consensus Algorithm ....................................... 26
   2.7 Discussion ................................................ 27

3 Continuous Kernel Games 28
   3.1 Game Formulation and Nash Equilibrium .................. 28
   3.2 Gradient-Play Dynamics with Perfect Information ........ 32
       3.2.1 Convergence of Gradient-Play Dynamics with Perfect Information ... 34
   3.3 Projected Gradient Dynamics with Perfect Information .... 35
       3.3.1 Existence and Uniqueness of Projected Gradient Dynamics ........... 36
       3.3.2 Convergence of Projected Gradient Dynamics ..................... 37
4 Augmented Gradient Dynamics with Estimate Correction

4.1 Player State Vector and Action/Estimate Component
4.1.1 Action and Estimate Component Matrices

4.2 Gradient Type Dynamics with Control Input
4.2.1 Gradient-Play Dynamics with Control Input
4.2.2 Projected Gradient Dynamics with Control Input
4.2.3 Pseudo-Gradient vs Extended Pseudo-Gradient
4.2.4 Monotone Extended Pseudo-Gradient and EIP

4.3 Control Input for Consensus of State Vector
4.3.1 Gradient-Play Dynamics with Estimate Correction
4.3.2 Projected Gradient with Estimate Correction

5 Convergence Analysis

5.1 Monotone Extended Pseudo-Gradient
5.1.1 Gradient-Play Dynamics with Estimate Correction
5.1.2 Projected Gradient with Estimate Correction

5.2 Strongly Monotone Pseudo-Gradient
5.2.1 Gradient-Play Dynamics with Estimate Correction
5.2.2 Projected Gradient with Estimate Correction
5.2.3 Gradient-Play Dynamics with Estimate Correction, Alternative Proof
5.2.4 Projection Gradient Dynamics with Estimate Correction, Alternative Proof

5.3 Two-Timescale Singular Perturbation Analysis
5.3.1 Gradient-Play Dynamics with Estimate Correction
5.3.2 Projected Gradient with Estimate Correction

6 Modifying Augmented Gradient Dynamics with Estimate Correction

6.1 Connections with Optimization
6.2 B matrix / weighted Laplacian
6.3 Passive Consensus Dynamics
6.4 Higher-Order Gradient Dynamics
6.4.1 Second-Order LTI System
6.4.2 Second-Order Gradient-Play dynamics with Estimate Correction
6.4.3 Dynamic Game Interpretation

7 Simulation Results

7.1 Unconstrained Action Set and Dynamics
7.2 Compact Action Set and Projected Dynamics
7.3 Passive Consensus Dynamics
7.4 Higher Order Dynamics
7.5 Algorithm Comparison
# 8 Conclusions and Future Work

## 8.1 Future Work

# A Matrix Proofs

# Bibliography
# List of Figures

2.1 Interior, boundary, closure and convex set ........................................... 10  
2.2 Normal cone, tangent cone and projection ........................................... 12  
2.3 Closed convex cone and its polar cone ............................................. 13  
2.4 Feedback Interconnection ................................................................. 19  
2.5 Parallel Interconnection ................................................................. 19  
2.6 Pre-multiply $M$ and post-multiply $M^T$ to system $\Sigma$ ....................... 20  
2.7 Passivity Comparison ....................................................................... 22  
2.8 Connected undirected graph $G_c$ ....................................................... 25  

3.1 Best Response .................................................................................. 29  
3.2 Vector Field .................................................................................... 34  
3.3 Gradient Dynamics (3.5) as a Feedback Interconnection of Two EIP Systems . 34

4.1 Communication Graph .................................................................... 50  
4.2 Laplacian ...................................................................................... 50  
4.3 Augmented gradient dynamics (4.30) over $G_c$ ..................................... 52  

6.1 Block Diagram of Actions and Estimates dynamics in (4.30) ..................... 74  
6.2 Distributed Optimization Block Diagram, Figure 2.a [1] ......................... 74  
6.3 Additional Decomposition and Feedback connections ............................ 75  
6.4 Gradient in feedback with passive system .......................................... 78  
6.5 Block Diagram LTI gradient dynamics .............................................. 82  
6.6 Block Diagram LTI gradient dynamics .............................................. 87  

7.1 (4.32) over random $G_c$ .................................................................. 95  
7.2 (4.32) over cycle $G_c$ .................................................................... 95  
7.3 Random $G_c$, $\lambda_2 = 4.95$ ............................................................. 95  
7.4 Cycle $G_c$ Graph ........................................................................... 95  
7.5 (3.5) perfect information ................................................................ 96  
7.6 (4.32) over complete $G_c$ .............................................................. 96  
7.7 (5.18) over random $G_c$, $\epsilon = 10$ ................................................ 96  
7.8 (5.18) over cycle $G_c$, $\epsilon = 10$ .................................................... 96
7.48 (7.2) Second Order Dynamics ......................................... 108
7.49 (4.30) over cycle $G_c$ .................................................. 110
7.50 Cycle $G_c$ Graph ......................................................... 110
7.51 (7.3) over cycle $G_c$ .................................................. 110
7.52 $[2]$ over cycle $G_c$ .................................................. 110
7.53 (4.30) over random $G_c$ ............................................. 111
7.54 Random $G_c$ Graph ..................................................... 111
7.55 (7.3) over random $G_c$ ............................................. 111
7.56 $[2]$ over random $G_c$ .................................................. 111

8.1 Gradient Dynamics .......................................................... 114
8.2 Payoff Dynamics ............................................................. 114
Nomenclature

$1_N$ An $N$ by 1 column vector of ones

$1_{i \times j}$ An $i$ by $j$ matrix of ones

$0_N$ An $N$ by 1 column vector of zeros

$0_{i \times j}$ An $i$ by $j$ matrix of zeros

⊕ Direct Sum

⊗ The Kronecker Product

$\nabla_i f(x)$ The partial derivative of the function $f(x)$ with respect to $x_i$, i.e., $\nabla_i f(x) = \frac{\partial f(x)}{\partial x_i}$ where $x = (x_i, x_{-i})$

$\nabla f(x)$ The gradient of the function $f(x)$

$\lambda_i(M)$ The $i$th smallest eigenvalue of the matrix $M$ or $\lambda_i$ when the matrix is inferred from the context

$\Sigma_i$ Player $i$’s dynamical system with input and output

$\Sigma$ A dynamical system with input and output

$\mathcal{P}_i$ Player $i$’s dynamics

$\mathcal{P}$ The overall dynamical system

$\Omega_i$ Agent $i$’s action set

$\Omega$ Overall action set, Cartesian product of $\Omega_i$

$\Omega^N$ The enlarged action set when agents have estimates of other agents strategies. It is $N$ Cartesian products of $\Omega$

$\partial \Omega$ The boundary of the set $\Omega$

$\text{int} \Omega$ The interior of the set $\Omega$
\(\Pi_\Omega(x, v)\) The projection of the vector \(v\) at point \(x\) with respect to \(\Omega\)

\(\mathbb{C}^i\) Set of functions with first \(i\) derivatives exist and are continuous

\(C_N^m\) The consensus subspace

\(\text{diag}(A_1, \cdots, A_n)\) Block diagonal matrix with \(A_i\) on the diagonal

\(E\) The edge set of a graph \(G\)

\(E_N^m\) The orthogonal complement of the consensus subspace

\(F\) The pseudo-gradient of the cost function \(J(x)\)

\(\mathbf{F}\) The extended pseudo-gradient of the cost function \(J(x)\)

\(G_c\) A communication graph

\(G(I, J_i, \Omega_i)\) A game defined by the player set, cost functions and action set

\(I\) The set of players / agents

\(I_j\) An \(j\) by \(j\) Identity matrix

\(J_i(x)\) Player \(i\)’s cost function evaluated at \(x\)

\(J(x)\) Stacked vector of cost functions where the \(i\)th row is player \(i\)’s cost function

\(L\) The Laplacian matrix for the communication graph \(G_c\)

\(\mathbf{L}\) The extended Laplacian matrix

\(P_\Omega(x)\) The projection of the point \(x\) onto \(\Omega\)

\(Q\) The incidence matrix of the graph \(G_c\)

\(\mathcal{N}_i\) Neighbours of agent \(i\) on the graph \(G_c\)

NE Nash Equilibrium

\(N_\Omega(x)\) The normal cone of \(\Omega\) at \(x\)

\(\text{Null}(M)\) The null space of matrix \(M\)

\(\text{Range}(M)\) The range space of matrix \(M\)

\(\mathbb{R}\) The set of real numbers

\(\mathcal{R}_i\) Player \(i\)’s action selection matrix

\(\mathcal{R}\) The action selection matrix
$S_i$  Player $i$’s estimate selection matrix
$S$  The estimate selection matrix
$T_{\Omega}(x)$  The tangent cone of $\Omega$ at $x$
$x_i$  Agent or player $i$’s action
$x_{-i}$  Actions of all agents except for agent $i$
$x$  All agents actions stacked into a vector
$x^i$  Agent $i$’s estimate vector
$x^i_i$  Agent $i$’s action
$x^i_{-i}$  Agent $i$’s estimate of all other agents’ actions actions
$x$  Estimate vector of all agents stacked
$x^\parallel$  The consensus component of the vector $x$
$x^\perp$  The orthogonal consensus component of the vector $x$
$\bar{x}, \hat{x}$  The equilibrium point for the dynamical system, $\Sigma$
$x^*, \hat{x}$  A NE point of $G(I, J_i, \Omega_i)$
Chapter 1

Introduction

Game theory describes mathematically the behaviour of players in a setting where players try to minimize their cost, but other players’ actions influence their cost. A simple example is driving to work. Each player is trying to find the best route to get to work that minimizes his/her time on the road. However, other drivers on the road influence how long it will take. Ideally, each player would like all the other drives to not be on the same road as them, so that the time it takes him/her to get to work is minimized. This is unrealistic idea and a better notion of ‘ideal’ needs to be defined. The Nash equilibrium is the strategy/action profile where no players are willing to unilaterally change their current strategy because it would not decrease their cost. In the driving example it would mean that no driver is willing to change their route to work because it wouldn’t make their commute any faster. In a way this is the ‘ideal’ scenario. Alternatively, you could argue that the best outcome would be if the total travel time of all the drives were minimized by some centralized super computer. The problem with this approach is that if drivers have different route objectives (not speed but minimize fuel consumption, best scenery, etc.) the centralized system would have to know all the drivers preferences. This is impractical so it would be better if there was a distributed method for solving this problem. Each driver could try to find the best route for himself/herself and hopefully end up at the Nash equilibrium. However, if a driver would like to find the optimal route to work it would require knowledge of all the other cars. Being able to communicate with every single car would require a vast network where thousand of vehicles would be sending information constantly. This communication scheme is unrealistic but a local communication method between neighbouring cars can be used. This problem, where players search for the Nash equilibrium (NE) based on local communication, is called distributed Nash equilibrium seeking over networks.

1.1 Motivation

In this thesis we consider distributed Nash equilibrium (NE) seeking over networks, where players have limited local information, over a communication network. This is a research topic of recent interest, [3], [4], [5], due to many networked scenarios in which such problems arise, e.g.,
in wireless communication \cite{6, 7, 8}, optical networks \cite{9, 10, 11}, distributed constrained convex optimization \cite{1, 12}, noncooperative flow control problems \cite{13, 14}, etc.

We propose a new continuous-time dynamics for a general class of N-player games and prove convergence to the NE over a undirected graph. Our scheme is derived by reformulating the problem as a multi-agent coordination problem between players. This reformulation appears counter intuitive because agents cooperate in a multi-agent coordination problem, but competing against each other in a game setting. However, players require some information about the other players in order to update their action. In the traffic example above, agents need to know where the other drives are traveling, otherwise there is not enough information for agents to update their route. Therefore, agents cooperate in exchanging information, but they do not cooperate in helping other agents reduce their cost. Specifically, we endow each player (agent) with an auxiliary state variable that provides an estimate of all other players’ actions. Agent’s exchange this estimate information to reach a consensus on the other players’ action. If all the agents have an accurate estimate of all the other agents’ actions then their gradient-type dynamics will move towards their best response and overall the players will reach a NE. Therefore, for each agent we combine its own gradient-type dynamics with an integrator-type auxiliary dynamics, driven by some control signal. We design the control signal for each individual player, based on the relative output feedback from its neighbours. In the limit these auxiliary state variables should agree with one another. At agreement all players will know the actions of all the other players and will be able to reach the NE. The resulting player dynamics has two components: the action component composed of a gradient term (enforcing the move towards minimizing its own cost) and a consensus term, and the estimate component composed with a consensus term. We call this new dynamics gradient-play dynamics with estimate correction (GPEC). We prove that it reaches consensus on the NE of the game, under a monotonicity property of the pseudo-gradient.

1.2 Literature Review

Our work is related to the literature of NE seeking in games over networks. Existing results are almost exclusively developed in discrete-time while we focus on continuous-time dynamics. The problem of NE seeking over networked communication is considered in \cite{15, 16}, specifically for the special class of aggregative games, where each agent’s cost function is coupled to other players’ actions through a single, aggregative variable. In \cite{17}, this approach is generalized to a larger class of coupled games: a gossip-based discrete-time algorithm is proposed and convergence was shown for diminishing step-sizes. Very recently, discrete-time ADMM-type algorithms with constant step-sizes have been proposed and convergence proved under co-coercivity of the extended pseudo-gradient, \cite{18, 19}.

In continuous-time, gradient-based dynamics for the NE computation have been used since the work of Arrow, \cite{20, 21, 22, 23}. Over networks, gradient-based algorithms are designed in
based on information from only a set of neighbouring agents. The games considered are ones with local utility functions, which are proved to be state-based potential games. Continuous-time distributed NE seeking dynamics are proposed for a two-network zero-sum game in [24]. The key assumptions used are the additive decomposition of the common objective function as well as other structural assumptions. Based on the max-min formulation, the system takes the form of the saddle-point dynamics, [20], distributed over the agents for each of the two networks. Their work was inspired by the optimization framework of [1] which connected optimization with control theory. Very recently [2] presented a continuous time consensus based approach. Their algorithm is similar to the gradient-play dynamics with estimate correction (GPEC) presented in this thesis. Their dynamics are obtained by removing a term from the GPEC and replacing the Laplacian with a weighted Laplacian. Additionally, their algorithm takes the gradient using only estimate information while GPEC uses both estimate information and the agents current action. Both their paper and a part of this thesis use a two-timescale singular perturbation method to prove convergence.

Our work is also related to the distributed optimization framework in [1]. However, there are several differences between [1] or [24] and our work. Beside the summable structure of the common cost function ( [24]), in [1] a critical structural assumption is that each agent optimizes its cost function over the full argument. Then, when an augmented (lifted) space of actions and estimates is considered in the networked communication case, a lifted cost function is obtained which can be decomposed as a sum of separable cost functions, individually convex in their full argument. This leads to distributed algorithms, under strict convexity of the individual cost functions with respect to the full argument. In the strategic game context, the individual convexity property with respect to the full argument is too restrictive unless the game is separable to start with. While the game setting has an inherent distributed structure (since each player optimizes its own cost function), individual (player-by-player) optimization is over its own action. In contrast to distributed optimization, each player’s action is only part of the full action profile and its cost function is coupled to its opponents’ actions, which are under their control. This key differentiating structural aspect between games and distributed optimization presents technical challenges. Our main approach is to highlight and exploit passivity properties of the game (pseudo-gradient), gradient-based algorithm/dynamics and the network (Laplacian). Interesting connections between passivity and games have been used in the context of population games in [25].

1.3 Contributions

We consider a general class of N-player games and develop new continuous-time dynamics for NE seeking dynamics under networked information. Our approach is based on reformulating the problem as a multi-agent coordination problem and exploiting the basic incremental passivity properties of the pseudo-gradient map. The reformulated multi-agent coordination problem
is not in a standard form. While the agents (players) have individual heterogeneous, separable, dynamics in the augmented state space, they do not satisfy an individual (incremental) passivity property as typically assumed in distributed multi-agent coordination literature. This individual (incremental) passivity property is typically assumed in coordination control or distributed optimization. In distributed optimization, the standard assumption is that individual cost functions are strictly convex, or that individual gradients are strictly monotone/incrementally passive. In the networked game setting considered herein, such individual gradient assumptions are too restrictive. Nevertheless, we show that under a strict incremental passivity of the pseudo-gradient vector, a synchronizing distributed feedback can be designed using the relative estimates of the neighbours, hence a Laplacian-type feedback. A passivity interpretation highlights the tradeoffs that one has to consider between properties of the game (individual cost functions) and properties of the communication graph. To the best of our knowledge such an approach has not been proposed before. The following is a list of the contributions in this thesis:

- When the incremental passivity (monotonicity) property holds on the augmented space, the dynamics proposed here performs simultaneous consensus of estimates and player-by-player optimization (Chapter 5, Theorem 5.1, 5.2). The technical novelty is showing attractivity of the consensus subspace. Our approach is similar to that in multi-agent coordination control, but unlike those works, herein we cannot exploit individual passivity properties of each agent’s system. The resulting dynamics has the form of a primal-dual type dynamics, where the dual role is taken by the auxiliary states for estimation of the other players’ actions. This is similar to the recent ADMM-type discrete-time algorithms, except that herein we rely on a weaker assumption, namely strict monotonicity of the pseudo-gradient, instead of co-coercivity as used, or individual co-coercivity, and our method converges over any connected graph. Our scheme is different from, due to an extra correction term on the actions’ dynamics that arises naturally from the passivity-based design. Note that the co-coercivity of the pseudo-gradient map used in is equivalent to incremental output passivity, while the strong monotonicity used in for constant-step sizes is equivalent to incremental input passivity. Additionally, we can relax the strict monotonicity assumption to hold just at the NE, recovering a strict-diagonal assumption used in. However since x is unknown, such an assumption cannot be checked a-priori in general, except for quadratic games, (see ).

- When the incremental passivity (monotonicity) property holds only on the action space this corresponds to the standard assumption in gradient-based NE seeking under all-to-all communication. In this case, we show convergence under a two time-scale separation between the consensus dynamics and the gradient-based dynamics (Chapter 5, Theorem 5.13). This is similar to the gossip-based discrete-time algorithm in, where conver-
gence was shown for diminishing step-sizes. Even though such time-scale separation was not explicitly considered in [17], the convergence proof effectively relies on an intermediary result of achieving consensus. Our scheme is different from [17], due to an extra correction term on the actions’ dynamics that arises naturally from the passivity-based design.

We note that a two time-scale analysis was used for a different scheme in [30], for quadratic, potential games and locally for general games. Very recently the results of [30] was extended to the setting considered in this thesis [2]. Our scheme is different from [17], [30], and [2] by having an extra correction term on the actions’ dynamics. Additionally, [2] uses only estimate information in the gradient term while our scheme uses both action and estimate information. Moreover, our passivity-based approach nicely ties all possible cases.

- The passivity-based approach highlights the trade-off between properties of the game and those of the communication graph (All Theorems/Corollaries in Chapter 5). Under a Lipschitz continuity assumption of the extended pseudo-gradient and strong monotonicity assumption of the pseudo-gradient, we quantify how strong does the connectivity of the communication graph need to be to ensure convergence. The key fact is that the Laplacian contribution (or excess passivity) can be used to balance the other terms that are dependent on the game properties.

- Due to the passivity interpretation and the reformulation as a multi-agent coordination problem, the Laplacian-type feedback can be modified and the dynamics converge under the same assumptions (Section 6.3). A weighted Laplacian matrix, where agents select the weights can improve convergence (Section 6.2). Alternatively, an imposed communication protocol can conceal agents’ estimate information. Dynamical extensions of the Laplacian-type feedback can also ensure convergence if the dynamics in the Laplacian system have a passivity property.

- Higher order gradient-type dynamics are shown to converge to the NE if the new dynamics have a passivity property (Section 6.4). These dynamics converge under the same Lipschitz and monotonicity conditions but have the added benefit of filtering out noise that can appear on the communication network. Additionally, some dynamical games can be represented as a higher order gradient-type dynamics and therefore convergence for this class of game can also be proven.

1.4 Organization

The thesis is organized as follows:

- Chapter 2 Background
Chapter 1. Introduction

The notation and definitions from control theory, graph theory, and projection operators are defined. Some well known theorems are stated, which will be used throughout the thesis.

- Chapter 3 Continuous Kernel Games
  This chapter defines what is a game, as well as what is a solution for the game, i.e., the Nash equilibrium. Theorems are provided that ensure that there is a solution for both unconstrained action spaces as well as compact action sets. The gradient-play dynamics for the perfect information setting is presented with the proof of convergence for monotone games.

- Chapter 4 Gradient with Communication Dynamics
  The gradient-play dynamics are modified to deal with the imperfect information case. The action space is expanded so that agents now maintain an estimate of other agents’ actions. The agents communicate with their neighbours to learn what the actions of all the other agents are. The communication between agents converts the problem into a multi-agent coordination problem for the estimates.

- Chapter 5 Convergence Analysis
  In this chapter all the convergence results are presented for different assumption on the pseudo-gradient. This chapter highlights the trade-off between the game conditions and the connectivity of the communication graph.

- Chapter 6 Modified Dynamics
  The dynamics are further modified by changing the Laplacian dynamics with either a weighting (time-varying) matrix or by a dynamical system. The gradient dynamics are also augmented to create higher order dynamics, which have a dynamic game interpretation.

- Chapter 7 Simulation
  Simulation results are presented for various cost functions as well as comparisons between the original gradient-communication dynamics and the modified versions of it.

- Chapter 8 Conclusion
  Lastly, concluding remarks about the work done in this thesis are made and suggestions of future work described.
Chapter 2

Background

In this chapter we introduce the mathematical tools needed to analyze games over a network. Section 2.1 introduces the basic mathematical notation. Section 2.2 discusses the projection operator, convex sets and cones. Section 2.3 defines a dynamical system and the conditions needed for there to be a solution. Additionally, stability of an equilibrium point of a system is defined along with various methods of proving stability. Section 2.4 describes passivity, equilibrium independent passivity, and incremental passivity. Section 2.5 defines the Laplacian matrix, which represents the communication graph of the game. Finally, section 2.6 presents a well known distributed consensus algorithm. This will be important for chapter 4 where the game will be converted into a consensus problem for the players’ estimates.

2.1 Mathematical Notations

In this section the vectors, matrices and basic concepts from linear algebra are defined. Additionally, a variety of function properties such as monotonicity, convexity and Lipschitz are defined. These properties will be used heavily in the convergence results in Chapter 5.

The set \( \mathbb{R}, \mathbb{R}_+, \mathbb{R}_{++} \) is the set of reals, non-negative reals and positive reals respectively. Given a column vector \( x \in \mathbb{R}^n \), \( x^T \) denotes its transpose. The \( i \)th component of the vector \( x \) is denoted as \( x_i \). Let \( x^T y = \langle x | y \rangle \) denote the Euclidean inner product of \( x, y \in \mathbb{R}^n \) and \( \| x \| = \langle x | x \rangle \) the Euclidean norm. Let \( A \otimes B \) and \( A \oplus B \) denote the Kronecker product and direct sum of matrices \( A \) and \( B \) respectively. The all ones vector is \( 1_n = [1, \ldots, 1]^T \in \mathbb{R}^n \), and the all zeros vector is \( 0_n = [0, \ldots, 0]^T \in \mathbb{R}^n \). Additionally, \( 1_{i \times j} \in \mathbb{R}^{i \times j} \) and \( 0_{i \times j} \in \mathbb{R}^{i \times j} \) represent an \( i \) by \( j \) matrix of ones and zeros respectively. Let \( \text{diag}(A_1, \ldots, A_n) \) denote the block-diagonal matrix with \( A_i \) on its diagonal. Given a matrix \( M \in \mathbb{R}^{p \times q} \) the null space is given by \( \text{Null}(M) = \{ x \in \mathbb{R}^q | Mx = 0 \} \) and the range space is \( \text{Range}(M) = \{ y \in \mathbb{R}^p | (\exists x \in \mathbb{R}^q) y = Mx \} \).

Lemma 2.1. Given a matrix \( M \in \mathbb{R}^{p \times q} \), then \( \text{Null}(M) = \text{Null}(M^T M) \)
Chapter 2. Background

Proof. In Appendix A

The element on the $i$th row and $j$th column of a matrix $M$ is denoted $[M]_{i,j}$. A real symmetric matrix $M \in \mathbb{R}^{n \times n}$ is positive semi-definite if $x^T M x \geq 0 \ \forall x \in \mathbb{R}^n$ and positive definite if $x^T M x > 0 \ \forall x \neq 0$.

Definition 2.1 (Definite Functions, [34]). Given a function $V : \Omega \to \mathbb{R}$ where $\Omega \subset \mathbb{R}^n$ and a point $x \in \Omega$. The function $V$ is,

1. **positive definite at** $x$ if $\forall y \neq x \in \Omega$, $V(y) > 0$ and $V(x) = 0$.
2. **positive semi-definite at** $x$ if $\forall y \neq x \in \Omega$, $V(y) \geq 0$ and $V(x) = 0$.
3. **negative definite at** $x$ if $-V(\cdot)$ is positive definite at $x$.
4. **negative semi-definite at** $x$ if $-V(\cdot)$ is positive semi-definite at $x$.

Monotone and convex functions are needed to ensure the existence of a solution for a game (chapter 3). Additionally, the monotone property is important in proving convergence in chapter 5 for the algorithm presented in chapter 4.

Definition 2.2 (Monotone Functions, [32]). Given a function $\Phi : \Omega \to \mathbb{R}$ where $\Omega \subset \mathbb{R}^n$. The function $\Phi$ is,

1. **monotone** if $(x - y)^T (\Phi(x) - \Phi(y)) \geq 0$, for all $x, y \in \Omega$.
2. **strictly monotone** if $(x - y)^T (\Phi(x) - \Phi(y)) > 0$, for all $x \neq y \in \Omega$.
3. **strongly monotone** if there exists $\mu > 0$ such that $(x - y)^T (\Phi(x) - \Phi(y)) \geq \mu \|x - y\|^2$, for all $x, y \in \Omega$.
4. **maximally monotone** if there exists no monotone function $\Psi : \mathbb{R}^n \to \mathbb{R}$ such that the graph of $\Psi$ properly contains the graph of $\Phi$.

Definition 2.3 (Convex Functions, [32]). Given a function $\Phi : \Omega \to \mathbb{R}$ where $\Omega \subset \mathbb{R}^n$. The function $\Phi$ is,

1. **convex** if $f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$, for all $x, y \in \Omega$ and $\forall \alpha \in (0, 1)$.
2. **strictly convex** if $f(\alpha x + (1 - \alpha)y) < \alpha f(x) + (1 - \alpha)f(y)$, for all $x \neq y \in \Omega$ and $\forall \alpha \in (0, 1)$.
3. **strongly convex** if there exists $\mu > 0$ such that $f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) - \frac{\mu}{2} \alpha (1 - \alpha) \|x - y\|^2$, for all $x, y \in \Omega$ and $\forall \alpha \in (0, 1)$.

The following proposition describes how monotone and convex functions are closely related.

Proposition 2.2 (Monotonicity / Convexity relationship). Let $f : \Omega \to \mathbb{R}^n$ be $C^1$ and $\Omega \subset \mathbb{R}^n$ is an open convex set. Then,
1. $f$ is convex if and only if $\nabla f$ is monotone (Prop 17.10 [32]).

2. $f$ is strictly convex if and only if $\nabla f$ is strictly monotone (Prop 17.13 [32]).

3. $f$ is strongly convex with constant $\mu$ if and only if $\nabla f$ is strongly monotone with constant $\mu$ (Ex 22.3 [32]).

**Definition 2.4** (Differentiable Functions, Def 2.45 [32]). Consider a function $f : \Omega \to \Psi$ where $\Omega \subset \mathbb{R}^n$ and $\Psi \subset \mathbb{R}^m$. Let $x$ be an arbitrary point in $\Omega$. The function $f$ is differentiable at $x$ if there exists a linear function $df : \mathbb{R}^n \to \mathbb{R}^m$, called the Fréchet derivative of $f$ at $x$, such that

$$\lim_{\|y\| \to 0} \frac{\|f(x + y) - f(x) - df(x)y\|}{\|y\|} = 0$$

If $f$ is differentiable at every $x \in \Omega$ then $f$ is differentiable. Higher-order Fréchet derivatives are defined inductively.

The Fréchet gradient of a function $f : \Omega \to \mathbb{R}$ at $x$ is the unique vector $\nabla f(x) \in \mathbb{R}^n$ such that

$$df(x)y = \nabla f(x)^T y \quad \forall y \in \mathbb{R}^n$$

If $f$ is differentiable on $\Omega$ then $\nabla f : \Omega \to \mathbb{R}^n$ is the gradient operator (Alternatively denoted as $\frac{\partial f(x)}{\partial x} \triangleq \nabla f(x)$). A function $f$ is continuously differentiable, $C^1$, if it is differentiable and $\nabla f$ is continuous. Generally, the function $f$ is $C^i$ if the first $i$ derivatives are differentiable and continuous.

Let $f : \Omega \to \Psi$ be a differentiable function where $\Omega \subset \mathbb{R}^n$ and $\Psi \subset \mathbb{R}^m$. The function $f_i$, $i \in m$, is the $i$-th component of $f$ [33]. Additionally, denote the natural basis of $\mathbb{R}^n$ by $\{e_1, \ldots, e_n\}$. Then for any $i \in m$ and $j \in n$, define $\frac{\partial f_i}{\partial x_j} : \Omega \to \mathbb{R}$ as

$$\frac{\partial f_i(x)}{\partial x_j} = \lim_{t \to 0} \frac{f_i(x + te_j) - f_i(x)}{t}$$

If the limit exists, then $\frac{\partial f_i}{\partial x_j}(x)$ is the $j$-th partial derivative of $f_i$ at $x$.

**Definition 2.5** (Lipschitz, Defn 1.46 (iii-iv) [33]). Consider a function $f : \Omega \to \Psi$ where $\Omega \subset \mathbb{R}^n$, $\Psi \subset \mathbb{R}^m$ and let $D \subset \Omega$. The function $f$ is

1. **Lipschitz continuous at $x$** if $(\forall \delta, L > 0)(\forall y, z \in \{w \in \mathbb{R}^n | \|w - x\| \leq \delta\}) \quad \|f(y) - f(z)\| \leq L \|y - z\|.$

2. **Locally Lipschitz on $\Omega$** if $f$ is Lipschitz at all points in $\Omega$.

3. **Lipschitz on $D$** if $(\exists L > 0)(\forall y, z \in D) \quad \|f(y) - f(z)\| \leq L \|y - z\|.$

4. **Globally Lipschitz** if $D = \Omega.$
**Definition 2.6** (Cocoercive, Defn 4.4 [32]). Let \( f : \Omega \to \mathbb{R}^n \) where \( \Omega \subset \mathbb{R}^n \). The function \( f \) is \( \mu \)-cocoercive if \( \mu > 0 \) and

\[
(x - y)^T(f(x) - f(y)) \geq \mu \|f(x) - f(y)\|^2 \quad (\forall x \in \Omega)(\forall y \in \Omega)
\]

### 2.2 Sets and Set Projection

In this section we define sets, cones and the projection onto a set. This will be needed when dealing with restricted action sets for each agent in chapter 3, 4 and 5. The following are from [33], [35] and [36].

Let \( \Omega, \Psi \subset \mathbb{R}^n \) denote subsets of the Euclidean \( n \)-space and \( I = \{1, 2, \ldots, N\} \) a finite collection of elements. Given a finite set \( I \) the cardinality of the set is denoted by \( |I| \) and equals the number of elements in the set, i.e, \( N \). The intersection of set \( \Omega \) with the set \( \Psi \) is denoted by \( \Omega \cap \Psi \) and the set of points in \( \Omega \) and not in \( \Psi \) is \( \Omega \setminus \Psi \). An open ball of radius \( r \) centered at \( x \) is defined as \( B_r(x) = \{y \in \mathbb{R}^n|\|y - x\| < r\} \). A point \( x \in \mathbb{R}^n \) is a limit point of the set \( \Omega \) if \((\forall r > 0)(\exists y \in B_r(x), y \neq x) y \in \Omega \). The set \( \Omega \) is closed if it contains all its limit points. The set \( \Omega \) is bounded in \( \mathbb{R}^n \) if there exists an \( x \in \mathbb{R}^n \) and \( r > 0 \) where \( \Omega \subset B_r(x) \). A set in \( \mathbb{R}^n \) is compact if and only if it is closed and bounded. The set \( \Omega \) is convex if \((\forall \delta \in [0,1])(\forall x,y \in \Omega) \delta x + (1-\delta)y \in \Omega \). The interior of the set \( \Omega \) is defined as \( \text{int}\Omega = \{x \in \Omega|\exists r > 0, B_r(x) \subset \Omega\} \). The boundary of the set \( \Omega \) is defined as \( \partial \Omega = \{x \in \mathbb{R}^n|\forall r > 0, \exists y \in B_r(x) \cap \Omega, \exists z \in B_r(x) \setminus \Omega, y \neq x, z \neq x\} \). The closure of \( \Omega \) is denoted \( \overline{\Omega} \) and is equal to \( \Omega \) plus all limit points of \( \Omega \). See figure 2.1 for an example of some of these sets.

![Figure 2.1: Interior, boundary, closure and convex set](image)

**Definition 2.7** (Cones). Given a set \( \Omega \subset \mathbb{R}^n \),
Chapter 2. Background

- The set $\Omega$ is a cone if for any $x \in \Omega$, $\gamma x \in \Omega$ for every $\gamma > 0$
- The polar cone of a convex cone $\Omega$ is defined as $\Omega^o = \{ y \in \mathbb{R}^n | y^T x \leq 0, \forall x \in \Omega \}$
- The normal cone of $\Omega$ at a point $x \in \Omega$ is defined as $N_{\Omega}(x) = \{ y \in \mathbb{R}^n | y^T(x' - x) \leq 0, \forall x' \in \Omega \}$
- The tangent cone of $\Omega$ at $x \in \Omega$ is defined as $T_{\Omega}(x) = \bigcup_{\delta > 0} \frac{1}{\delta}(\Omega - x)$ or $T_{\Omega}(x) = \{ y \in \mathbb{R}^n | y = \alpha(x' - x) \alpha \in \mathbb{R}_+, x' \in \Omega \}$

The point-to-set distance $d(x, \Omega)$ is defined as $d(x, \Omega) \overset{\Delta}{=} \inf \{ \| x - y \| | y \in \Omega \}$. The projection of a point $x \in \mathbb{R}^n$ to the set $\Omega$ is given by $P_{\Omega}(x) \in \Omega$ such that $d(x, \Omega) = \| x - P_{\Omega}(x) \| \leq \| x - x' \|$, for all $x' \in \Omega$, or $P_{\Omega}(x) \overset{\Delta}{=} \arg\min_{x' \in \Omega} \| x - x' \|$. The projection of a vector $v \in \mathbb{R}^n$ at a point $x \in \Omega$ with respect to $\Omega$ is defined by $\Pi_{\Omega}(x, v) \overset{\Delta}{=} \lim_{\delta \to 0^+} \frac{P_{\Omega}(x + \delta v) - x}{\delta}$. Given $x \in \partial \Omega$, let $n(x) \overset{\Delta}{=} \{ y | y \in N_{\Omega}(x), \| y \| = 1 \}$ denote the set of outward unit normals to $\Omega$ at $x$.

**Lemma 2.3** (Vector Projection Decomposition (Lemma 2.1, [35])).

Let $\Omega$ be a closed, convex set. If $x \in \text{int}\Omega$ then,

$$\Pi_{\Omega}(x, v) = v$$

and if $x \in \partial \Omega$ then,

$$\Pi_{\Omega}(x, v) = v + \beta(x)n^*(x) \quad (2.1)$$

where,

$$n^*(x) = \arg\max_{n \in \mathbb{H}(x)} v^T n \quad \beta(x) = \max \{ 0, v^T n^*(x) \}$$

Note that if $v \in T_{\Omega}(x)$ for some $x \in \partial \Omega$, then $\sup_{n \in \mathbb{H}(x)} v^T n \leq 0$ hence $\beta(x) = 0$ and no projection needs to be performed. The operator $\Pi_{\Omega}(x, v)$ is equivalent to the projection of the vector $v$ onto the tangent cone $T_{\Omega}(x)$ at $x$, i.e., $\Pi_{\Omega}(x, v) = P_{T_{\Omega}(x)}(v)$. Figure 2.2 shows the different types of cones and the projection onto a set.

**Theorem 2.4** (Moreau’s Decomposition Theorem, III.3.2.5 [36]). Let $C \subseteq \mathbb{R}^n$ and $C^o \subseteq \mathbb{R}^n$ be a closed convex cone and its polar cone respectively. Let $v \in \mathbb{R}^n$, then the following are equivalent:

1. $v_C = P_C(v)$ and $v_{C^o} = P_{C^o}(v)$.
2. $v_C \in C$, $v_{C^o} \in C^o$, $v = v_C + v_{C^o}$, and $v_C^T v_{C^o} = 0$.

Figure 2.3 depicts the decomposition of a vector.

Notice that $N_{\Omega}(x)$ is a convex cone and the tangent cone is its polar cone, i.e., $N_{\Omega}(x) = (T_{\Omega}(x))^o$, $(N_{\Omega}(x))^o = T_{\Omega}(x)$. By Lemma 2.4 for any $x \in \Omega$, the vector $v \in \mathbb{R}^n$ can be
decomposed into tangent $v_{T\Omega} \in T_{\Omega}(x)$ and normal component $v_{N\Omega} \in N_{\Omega}(x)$,

$$v = v_{T\Omega} + v_{N\Omega} \quad (2.2)$$

with $v_{T\Omega} = P_{T_{\Omega}(x)}(v) = \Pi_{\Omega}(x, v)$ and $v_{N\Omega} = P_{N_{\Omega}(x)}(v)$.

**Definition 2.8 (Variational Inequality (VI), Def 1.1.1 [37])**. Given a set $\Omega \subset \mathbb{R}^n$ and a function $f : \Omega \rightarrow \mathbb{R}^n$, the variational inequality, denoted $VI(\Omega, f)$, is to find a vector $x \in \Omega$ such that

$$(y - x)^T f(x) \geq 0, \quad \forall y \in \Omega$$

Notice that the variational inequality can be written as $(y - x)^T [-f(x)] \leq 0$. From the definition of the normal cone, the variational inequality can be stated as finding a vector $x$ such that $-f(x)$ is in the normal cone of $x$. In chapter 3 the VI is used to characterize the solution of a game.

### 2.3 Dynamical Systems and Stability

In this section a dynamical system is defined and the conditions for the system to have a solution. Then the equilibrium of a dynamical system and the stability of the equilibrium will be described. Lastly, theorems for determining the stability of a dynamical system will be presented. These theorems will be used in Chapter 5 to prove convergence. The follow results are obtained from [34] and [35].
Let $\Sigma$ denote a dynamical system

$$\Sigma : \begin{cases} \dot{x}(t) = f(t, x(t), u(t)), \\ y(t) = h(t, x(t), u(t)) \end{cases} \quad (2.3)$$

with state $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$, input $u(t) \in U \subseteq \mathbb{R}^q$, output $y(t) \in \mathcal{Y} \subseteq \mathbb{R}^q$ and $t \in \mathbb{R}$ is time. The initial state of (2.3) is $x_0 \overset{\Delta}{=} x(t_0)$ where $t_0 \in \mathbb{R}$ is the corresponding time. Throughout the thesis the $(t)$ part of the variable will be dropped to simplify notation, but it should be understood that the state, input and output are a function of time, i.e., $x \overset{\Delta}{=} x(t)$. A dynamical system $\mathcal{P}$ denotes the system $\Sigma$ where $u$ is a function of $x$, e.g., $u = -y = -h(x)$. Consider a differentiable function $V : \mathbb{R}^n \to \mathbb{R}$. The time derivative of $V$ along solutions of (2.3) is denoted as $L_f V \overset{\Delta}{=} \nabla^T V(x) f(x, u) = V(x)$ or just $\dot{V}$.

So far there are no assumptions on the functions $f$, $h$ and $u$. Without having any conditions on $f$ the trajectory of $x$ may not be defined or may not be unique. Therefore, additional assumptions ensure that there exists a unique solution.

**Theorem 2.5** (Local Existence and Uniqueness (Thm 3.1, [34])).

Given a dynamical system,

$$\dot{x} = f(t, x) \quad x(t_0) = x_0$$

Let $f(t, x)$ be piecewise continuous in $t$ and satisfy the Lipschitz condition

$$\|f(t, x) - f(t, y)\| \leq L \|x - y\| \quad \forall x, y \in B_r(x_0), \forall t \in [t_0, t_1]$$
Then, there exists some \( \delta > 0 \) such that the state equation \( \dot{x} = f(t, x) \) with \( x(t_0) = x_0 \) has a unique solution over \([t_0, t_0 + \delta]\).

Therefore, for any system in the form of (2.3) it is assumed that the function \( f \) is locally Lipschitz and \( h \) continuous.

If the state space is restricted to a closed convex set \( \Omega \) then the projection operator \( \Pi_\Omega(x, v) \) can be used to ensure \( x \) doesn’t leave \( \Omega \). The vector \( v \) will be the direction that \( x \) will be changing in, e.g., \( v = f(x) \). The dynamics created by \( \Pi_\Omega(x, v) \) are discontinuous on the boundary and therefore, the standard definition of a solution for a dynamical system doesn’t apply. Therefore, a different solution concept is introduced when dealing with projected dynamics.

**Definition 2.9** (Absolutely continuous, [38]). A function \( x : [a, b] \rightarrow \mathbb{R}^n \) is absolutely continuous if for all \( \epsilon > 0 \), there exists a \( \delta > 0 \) such that, for any integer \( N > 0 \) and collection of disjoint subintervals \([a_i, b_i], i \in [1, 2, \ldots, N]\) of \([a, b]\) where

\[
\sum_{i=1}^{N} (b_i - a_i) < \delta
\]

we have

\[
\sum_{i=1}^{N} |x(b_i) - x(a_i)| < \epsilon
\]

**Definition 2.10** (Solution of Projected Dynamical Systems (Def 2.5, [35])).

A function \( x : [0, \infty) \rightarrow \Omega \) is a solution to \( \dot{x} = \Pi_\Omega(x, -F(x)) \) if \( x(\cdot) \) is absolutely continuous and if \( \dot{x}(t) = \Pi_\Omega(x(t), -F(x(t))) \) holds almost everywhere (a.e.) and \( \Omega \) is a closed convex set.

**Theorem 2.6** (Uniqueness of solution on \( \Omega \) ( Theorem 2.5, [35])).

Let \( -F : \mathbb{R}^n \rightarrow \mathbb{R}^n \) satisfy the following conditions,

\[
\|F(x)\| \leq B(1 + \|x\|) \quad \forall x \in \Omega, \quad B < \infty
\]

\[
(-F(x) + F(y))^T(x - y) \leq B \|x - y\|^2, \quad \forall x, y \in \Omega
\]

(2.4)

Then, for any \( x(0) \in \Omega \) there is a unique solution to \( \dot{x} = \Pi_\Omega(x, -F(x)) \). Furthermore, if \( x_n \rightarrow x_0 \) as \( n \rightarrow \infty \), then \( x_n(t) \) converges to \( x_0(t) \) uniformly on every compact set of \([0, \infty)\).

Note, that if \( F \) is Lipschitz then the conditions in Theorem 2.6 are satisfied and there is a unique solution to the projected dynamics. This is a mild condition for \( F \) when the set \( \Omega \) is compact.

An important question about a dynamical system is what happens to the state \( x \) as it evolves over time. More specifically, does the state \( x \) stay in some region, go to a point or does it go to infinity. The behaviour of \( x \) describes the stability of the system.

**Definition 2.11** (Equilibrium Point).
Consider the nonautonomous system \( \dot{x} = f(t, x) \). A point \( \bar{x} \in \mathcal{X} \) is an \textit{equilibrium point} of the system if when \( x(t_0) = \bar{x} \) then,

\[
\dot{x}(t) = f(t, x(t_0)) = 0 \quad \forall t \geq t_0
\]

**Definition 2.12** (Stability (Defn 4.4 and 4.5, [34])).
Consider the nonautonomous system \( \dot{x} = f(t, x) \). An equilibrium \( \bar{x} = 0 \) is,

1. \textit{stable} if for all \( \epsilon > 0 \),
   \[
   (\exists \delta = \delta(\epsilon, t_0) > 0) \quad \|x(t_0)\| < \delta \implies \|x(t)\| < \epsilon \quad \forall t \geq t_0
   \]

2. \textit{unstable} if it is not stable

3. \textit{asymptotically stable} if it is stable and,

\[
\exists r = r(t_0) > 0 \quad \forall x(t_0) \in B_r(0) \implies \lim_{t \to \infty} x(t) = 0 \quad (2.5)
\]

4. \textit{exponentially stable} if \( \exists r, c_1, c_2 > 0 \) such that,

\[
(\forall x(t_0) \in B_r(0))\quad \|x(t)\| \leq c_1 \|x(t_0)\| e^{-c_2(t-t_0)} \quad \forall t
\]

The condition in (2.5) states that the equilibrium point is \textit{attractive} or an \textit{attractor}.

The theorems below provide a way to prove stability of an equilibrium point.

**Theorem 2.7** (Lyapunov’s stability theorem (Thm 4.1, [34])).
Let \( \bar{x} \) be an equilibrium point of

\[
\dot{x} = f(x)
\]

and \( \Omega \subset \mathcal{X} \subset \mathbb{R}^n \) be a set that contains the equilibrium point. Let \( V : \Omega \to \mathbb{R} \) be a \( C^1 \) function such that,

1. \( V \) is positive definite at \( \bar{x} \) and \( \dot{V} \) is negative semi-definite at \( \bar{x} \). Then, the equilibrium point is stable.

2. \( V \) is positive definite at \( \bar{x} \) and \( \dot{V} \) is negative definite at \( \bar{x} \). Then, the equilibrium point is asymptotically stable.

The function \( V \) is called a Lyapunov function.

**Definition 2.13** (Radially unbounded, [34]). A function \( f : \mathbb{R}^n \to \mathbb{R} \) is \textit{radially unbounded} if

\[
f(x) \to \infty \text{ as } \|x\| \to \infty
\]
Theorem 2.8 (Barbashin-Krasovskii Theorem, (Thm 4.2, [34])).

Let $\bar{x}$ be an equilibrium point of

$$\dot{x} = f(x)$$

Let $V : \mathbb{R}^n \to \mathbb{R}$ be a $C^1$ function such that

- $V(x)$ is positive definite at $\bar{x}$.
- $V(x)$ is radially unbounded.
- $\dot{V}(x)$ is negative definite at $\bar{x}$.

Then $\bar{x}$ is globally asymptotically stable.

Theorem 2.9 (Exponentially Stable Lyapunov (Thm 4.10, [34])).

Consider the nonautonomous system $\dot{x} = f(t, x)$, where $\bar{x} = 0$ is the equilibrium point and $\Omega \subset \mathcal{X} \subset \mathbb{R}^n$ is a set containing $\bar{x} = 0$. Let $V : [0, \infty) \times \Omega \to \mathbb{R}$ be $C^1$ such that for all $t \geq 0$ and $\forall x \in \Omega$,

$$c_1 \|x\|^k \leq V(t, x) \leq c_2 \|x\|^k$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \leq -c_3 \|x\|^k$$

where $c_1, c_2, c_3 > 0$ and $k > 0$. Then, $\bar{x} = 0$ is exponentially stable.

The theorems above both require that the Lyapunov function be positive definite. A very useful theorem for proving attractiveness of a set without requiring a positive definite function is stated below.

Definition 2.14 (Invariant Set, [34]).

A set $\Psi \subset \mathcal{X}$ is

- **positively invariant** if
  $$x(0) \in \Psi \implies x(t) \in \Psi \ \forall \ t > 0$$

- **negatively invariant** if
  $$x(0) \in \Psi \implies x(t) \in \Psi \ \forall \ t < 0$$

- **invariant** if
  $$x(0) \in \Psi \implies x(t) \in \Psi \ \forall \ t \in \mathbb{R}$$

Theorem 2.10 (LaSalle’s Invariance Principle (Thm 4.4, [34])).

Consider the autonomous system $\dot{x} = f(x)$, where $f : \mathcal{X} \to \mathbb{R}^n$ is locally Lipschitz. Let $\Omega \subset \mathcal{X}$ be compact and positively invariant. Let $V : \mathcal{X} \to \mathbb{R}$ be a $C^1$ function such that $\dot{V}(x) \leq 0$, $\forall x \in \Omega$. Let $E = \{x \in \Omega \mid \dot{V}(x) = 0\}$ and let $M$ be the largest invariant set in $E$. Then $\forall x \in \Omega$, as $t \to \infty$, $d(x, M) \to 0$. 
Notice that the theorem doesn’t require that $V(x)$ be positive definite and that it doesn’t prove stability, only that $x$ is attracted to the largest invariant set $E$. When $\dot{V}(x)$ is negative semi-definite then Theorem \ref{thm:laPlace} only showed stability. The power of Theorem \ref{thm:laPlace2} is that if the largest invariant set is just the equilibrium point then by LaSalle’s invariance principle one can prove asymptotic stability, where Theorem \ref{thm:laPlace} could only show stability as stated by Corollary \ref{cor:laPlace}.

**Corollary 2.11** (Corollary 4.1 \cite{34}).

Let $\bar{x} = 0$ be an equilibrium point of,

$$\dot{x} = f(x)$$

Let $V : \Omega \rightarrow \mathbb{R}$ be a $C^1$ function with $\Omega$ containing $\bar{x}$, such that $\dot{V}$ is negative semi-definite at $\bar{x}$. If the largest invariant subset of $E = \{ x \in C \mid \dot{V}(x) = 0 \}$ is $\{ \bar{x} \}$ then $\bar{x}$ is asymptotically stable.

When the dynamical system is not autonomous the set $E$ in LaSalle’s invariance principle is now time dependent. Therefore, the trajectories should approach $E$ as $t \to \infty$. An important lemma for proving attraction for nonautonomous systems is needed to prove asymptotic stability.

**Definition 2.15** (Uniformly Continuous Function, \cite{34}).

Consider a function $f : \Omega \rightarrow \Psi$ where $\Omega \subset \mathbb{R}^n$ and $\Psi \in \mathbb{R}^m$. The function $f$ is uniformly continuous if

$$(\forall \epsilon > 0)(\exists \delta > 0)(\forall x, y \in \Omega) \quad ||x - y|| < \delta \implies ||f(x) - f(y)|| < \epsilon$$

**Lemma 2.12** (Barbalat’s Lemma (lemma 8.2 \cite{34})).

Let $x : \mathbb{R} \rightarrow \mathbb{R}$ be a uniformly continuous function on $[0, \infty)$. Suppose that,

$$\lim_{t \to \infty} \int_0^t x(\tau)d\tau$$

exists and is finite. Then, $x(t) \to 0$ as $t \to \infty$.

Two additional theorems are stated below because they will be used to prove convergence in Chapter \ref{chap:5}. Both theorems prove convergence for a nonlinear system that has been decomposed into two subsystems. One system is the slow system and the other system is the fast system. The idea of the theorem is that if the fast system converges to a point or subspace quick enough and a point is asymptotically stable for the slow subsystem then the point is asymptotically stable for the overall system. The first theorem proves asymptotic stability and the second one proves exponential stability.

**Theorem 2.13** (Singular Perturbation (Thm 11.3, \cite{34})).
Consider the system,

\[
\dot{x} = f(x, y + h(x)) \\
\epsilon \dot{y} = g(x, y + h(x)) - \epsilon \frac{\partial h}{\partial x} f(x, y + h(x))
\]

(2.6)

where \((0, 0)\) is an isolated equilibrium point. Assume that there exist Lyapunov functions \(V(x)\) and \(W(x, y)\) that satisfy,

\[
\frac{\partial V}{\partial x} f(x, h(x)) \leq -a_1 \psi_1^2(x) \quad (2.7)
\]

\[
\frac{\partial W}{\partial y} g(x, y + h(x)) \leq -a_2 \psi_2^2(y) \quad (2.8)
\]

\[
W_1(y) \leq W(x, y) \leq W_2(y) \quad (2.9)
\]

\[
\frac{\partial V}{\partial x} [f(x, y + h(x)) - f(x, h(x))] \leq b_1 \psi_1(x) \psi_2(y) \quad (2.10)
\]

\[
\left[ \frac{\partial W}{\partial x} - \frac{\partial W}{\partial y} \frac{\partial h}{\partial x} \right] f(x, y + h(x)) \leq b_2 \psi_1(x) \psi_2(y) + \gamma \psi_2^2(y) \quad (2.11)
\]

\(\forall x \in \Omega_x, \forall y \in \Omega_y\) where, \(\Omega_x\) and \(\Omega_y\) is the domain of \(x\) and \(y\) respectively. The functions \(\psi_1(x), \psi_2(y), W_1(y)\) and \(W_2(y)\) are positive definite at 0. Let \(\epsilon_d\) and \(\epsilon^*\) be defined as,

\[
\epsilon_d \Delta \frac{a_1 a_2}{a_1 \gamma + \frac{1}{4d(1-d)}[(1-d)b_1 + db_2]^2}
\]

\[
\epsilon^* \Delta \frac{a_1 a_2}{a_1 \gamma + b_1 b_2}
\]

where \(d \in (0, 1)\). Then, the origin of (2.6) is asymptotically stable for all \(0 < \epsilon < \epsilon^*\). Additionally, for \(\epsilon \in (0, \epsilon_d)\) the Lyapunov function for (2.6) is

\[
\nu(x, y) = (1-d) V(x) + d W(x, y)
\]

**Theorem 2.14** (Singular Perturbation, Exponential Stability (Thm 11.4, \([34]\))).

Consider the system

\[
\dot{x} = f(t, x, z, \epsilon) \\
\epsilon \dot{z} = g(t, x, z, \epsilon)
\]

(2.12)

Assume all the following assumptions are satisfied for all \((t, x, \epsilon) \in [0, \infty) \times B_r(0) \times [0, \epsilon_0],\)

1. \(f(t, 0, 0, \epsilon) = 0\) and \(g(t, 0, 0, \epsilon) = 0\)

2. The equation \(0 = g(t, x, z, \epsilon)\) has an isolated root \(z = h(t, x)\) such that \(h(t, 0) = 0\).

3. The functions \(f, g, h\) and their partial derivatives up to the second order are bounded for \(z - h(t, x) \in B_r\).

4. The origin of \(\dot{x} = f(t, x, h(t, x), 0)\) is exponentially stable.
5. The origin of \( \frac{dy}{d\tau} = g(t, x, y + h(t, x), 0) \) is exponentially stable, uniformly in \((t, x)\) (The variable \( \tau \) is defined as \( \tau \triangleq \frac{t}{\epsilon} \)).

Then, there exists \( \epsilon^* > 0 \) such that \( \forall \epsilon < \epsilon^* \), the origin of (2.12) is exponentially stable.

### 2.4 Passivity

In this section we define passivity and variants of passivity. Various properties of passive systems are presented and a connection between passivity, monotonicity and convexity is highlighted. The following are from [39], [28], [27], [40].

Let \( \Sigma \) denote the dynamical system,

\[
\Sigma: \begin{cases} 
\dot{x} = f(x, u) \\
y = h(x, u)
\end{cases}
\]  

(2.13) with state \( x(t) \in \mathcal{X} \subseteq \mathbb{R}^n \), input \( u(t) \in \mathcal{U} \subseteq \mathbb{R}^q \), output \( y(t) \in \mathcal{Y} \subseteq \mathbb{R}^q \).

**Definition 2.16** (Passive, Definition 6.3 [34]).

Consider the dynamical system (2.13) where the equilibrium is \( 0 \in \mathcal{X} \). The system is *passive* if there exists a \( C^1 \) positive semidefinite function \( V(x) \) such that

\[
L_fV = \frac{\partial V}{\partial x} f(x, u) \leq u^T y, \quad \forall (x, u) \in (\mathcal{X}, \mathcal{U})
\]

Moreover, it is

- *output strictly passive* if \( L_fV \leq u^T y - \rho(y) \) and \( \rho \) is positive definite at 0.
- *input strictly passive* if \( L_fV \leq u^T y - \varphi(u) \) and \( \varphi \) is positive definite at 0.

Some useful properties of passive systems is that interconnection of passive systems remain passive and pre/post multiplication of a matrix is also passive.

**Lemma 2.15** (Theorem 2.10 [41]). If \( \Sigma \) and \( \Pi \) are two passive systems then the feedback interconnection, figure 2.4, and the parallel interconnection, 2.5, of \( \Sigma \) and \( \Pi \) is passive.

![Feedback Interconnection](image1)

![Parallel Interconnection](image2)
Lemma 2.16 (Proposition 2.11 [41]). Given a matrix $M$ and a system $\Sigma$ that is passive then if $\Sigma$ is pre-multiplied by $M$ and post-multiplied by $M^T$ it is also passive.

An Equilibrium Independent Passive (EIP) system is a variant of the passivity definition. It compares an arbitrary equilibrium input/output pair to the input/output pair for a trajectory $x(t)$. In order to define an EIP system some additional functions are defined to construct an arbitrary equilibrium input/output pair.

Assume there exists a non empty set $\overline{U} \subset \mathcal{U}$ and a continuous function $k_x : \overline{U} \rightarrow \mathcal{X}$ such that for any constant $\overline{u} \in \overline{U}$, $f(k_x(\overline{u}), \overline{u}) = 0$. The function $k_x$ is referred to as the equilibrium input-state map. Similarly, the equilibrium input-output map is defined as a continuous function $k_y : \overline{U} \rightarrow \mathcal{Y}$ where $k_y(\overline{u}) = h(k_x(\overline{u}), \overline{u})$.

Definition 2.17. System $\Sigma$ (2.13) is Equilibrium Independent Passive (EIP) if it is passive with respect to $u$ and $y$. In other words, there exists a differentiable, positive semi-definite storage function $V : \mathcal{X} \times \overline{\mathcal{X}} \rightarrow \mathbb{R}$ satisfying, $\forall (x, \overline{x}, u) \in \mathcal{X} \times \overline{\mathcal{X}} \times \mathcal{U}$

$$V(x, \overline{x}) \geq 0 \quad V(\overline{x}, \overline{x}) = 0$$

where $\overline{\mathcal{X}} \triangleq \{ x \in \mathcal{X} | \exists \overline{u} \in \overline{U}, \ x = k_x(\overline{u}) \}$ and

$$\dot{V}(x, \overline{x}) \leq (y - \overline{y})^T(u - \overline{u})$$

where $y = h(x, u)$, $\overline{y} = h(k_x(\overline{u}), \overline{u})$.

Other similar notions of input/output strictly passive systems can also be defined. A system $\Sigma$ is output-strictly EIP if there exists a positive definite function $\rho : \mathbb{R}^q \rightarrow \mathbb{R}$ such that

$$\dot{V}(x, \overline{x}) \leq (y - \overline{y})^T(u - \overline{u}) - \rho(y - \overline{y})$$

The system $\Sigma$ is input-strictly EIP if there exists a positive definite function $\varphi : \mathbb{R}^q \rightarrow \mathbb{R}$ such that

$$\dot{V}(x, \overline{x}) \leq (y - \overline{y})^T(u - \overline{u}) - \varphi(u - \overline{u})$$

A slight refinement to the EIP definition can be made to handle the case where $k_y(\overline{u})$ is not a function but is a map. An EIP system with a map $k_y(\overline{u})$ is called maximal EIP (MEIP) when $k_y(\overline{u})$ is maximally monotone, e.g., $\Sigma$ is an integrator, [28].

The interest in EIP systems is that they help in deriving stability and convergence properties for feedback systems without requiring exact knowledge of the equilibrium point [39]. EIP
Lemma 2.17 (Property 2 & 3 [39]). If $\Sigma$ and $\Pi$ are two EIP systems then the feedback interconnection, figure 2.4, and the parallel interconnection, 2.5, of $\Sigma$ and $\Pi$ is an EIP system.

Lemma 2.18. Given a matrix $M$ and a system $\Sigma$ that is EIP then if $\Sigma$ is pre-multiplied by $M$ and post-multiplied by $M^T$ it is also EIP.

Proof. Since $\Sigma$ is EIP it therefore satisfies

$$\dot{V}(x, \bar{\pi}) \leq (y - \bar{y})^T(u - \bar{u})$$

Let $u = Mu_M$ and $y_M = M^Ty$ where $u_M$ and $y_M$ is the input and output to the overall system respectively. Substituting $u = Mu_M$ gives the condition,

$$\dot{V}(x, \bar{\pi}) \leq (y - \bar{y})^TM(u_M - \bar{u}_M)$$

then using $y^T = y_M^TM$ the equation becomes

$$\dot{V}(x, \bar{\pi}) \leq (y_M - \bar{y}_M)^T(u_M - \bar{u}_M)$$

therefore the system from $u_M$ to $y_M$ is EIP.

In the EIP definition, a trajectory $x(t)$ is compared to $\bar{x}$ (fixed). When $\pi$ is replaced by any trajectory $x' \in X$ and if the passivity property holds in comparing any two trajectories of $\Sigma$, the property is called incremental passivity, [40].

Definition 2.18. System $\Sigma$ (2.3) is incrementally passive (IP) if there exists a $C^1$, regular, positive semi-definite storage function $V : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ such that for any two inputs $u, u'$ and with corresponding state $(x, x')$ and output $(y, y')$ the following inequality is satisfied,

$$\dot{V}(x, x') \leq (y - y')^T(u - u')$$ (2.14)

where $\dot{V} = [\frac{\partial}{\partial x} V(x, x')]f(x, u) + [\frac{\partial}{\partial x'} V(x, x')]f(x', u')$.

Output-strictly IP and input-strictly IP can be also be defined by replacing the static $\bar{u}$ and $\bar{y}$ in the output-strictly/input-strictly EIP with an arbitrary input and output. When $u', x', y'$ are constant (equilibrium conditions), this recovers the EIP definition.

The definitions above define different passivity notions and the lemmas explain that interconnecting passive systems maintains the passivity property. The importance of this property is that the sum of the storage functions can be used as a Lyapunov function. The theorem below provides an easy way to construct a Lyapunov function to prove stability for an equilibrium point.
**Theorem 2.19** (Theorem 3.1 [42]). Consider the interconnection of $N$ subsystems, that are described by

\[
\dot{x}_i = f_i(x_i, u_i) \\
y_i = h_i(x_i, u_i)
\]

with $x_i \in \mathbb{R}^{n_i}$ and $u_i, y_i \in \mathbb{R}^{m_i}$ for some integer $n_i > 0$ and $m_i > 0$. The input to each subsystem is dependent on the output of the other subsystems by

\[
u = My
\]

where $u = [u_1^T, \ldots, u_N^T]^T \in \mathbb{R}^m$, $y = [y_1^T, \ldots, y_N^T]^T \in \mathbb{R}^m$, $M \in \mathbb{R}^{m \times m}$, and $m = \sum_i N m_i$. Suppose that the overall interconnected system has an equilibrium $x^* = [(x_1^*)^T, \ldots, (x_N^*)^T]^T \in \mathbb{R}^n$, where $n = \sum_i N n_i$, and each subsystem is EIP. If there exists $p_i > 0$, $i = 1, \ldots, N$ such that

\[
\begin{bmatrix}
M \\
I_m
\end{bmatrix}
\begin{bmatrix}
0_{m \times m} & \frac{1}{2}P \\
\frac{1}{2}P & 0_{m \times m}
\end{bmatrix}
\begin{bmatrix}
M \\
I_m
\end{bmatrix} \leq 0
\]

where $P = \text{diag}(p_1 I_{m_1}, \ldots, p_N I_{m_N})$ then $x^*$ is stable with Lyapunov function

\[
V(x) = \sum_i p_i V_i(x_i, x_i^*)
\]

Another reason for using passive system is highlighted when the system $\Sigma$ is just a static map. There is an interesting connection between convexity, monotonicity and passivity.

![Figure 2.7: Passivity Comparison](image)

When the system $\Sigma$ is just a static map, $y = \Phi(u)$, the system has no state, and therefore $\dot{V} = 0$ in inequality [2.14]. This inequality matches the definition of a monotone function, i.e.,
Definition 2.2(1). Therefore, the incremental passivity property for static maps is equivalent to the static map being monotone. Monotonicity plays an important role in optimization and variational inequalities, while passivity plays as critical a role in the interconnection of dynamical systems. Figure (2.7) compares the different passivity definitions to convex and monotone functions and table (2.1) shows the relationship between convex functions and passivity.

<table>
<thead>
<tr>
<th>Convex Function $f(x)$</th>
<th>Passivity Property of a Static function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nabla f(x)$ is monotone</td>
<td>$\nabla f(x)$ is IP</td>
</tr>
<tr>
<td>$\nabla f(x)$ is strictly monotone</td>
<td>$\nabla f(x)$ is IP, for some $\varphi(x)$</td>
</tr>
<tr>
<td>$\nabla f(x)$ is $\mu$-strongly monotone</td>
<td>$\nabla f(x)$ is input-strictly IP, $\varphi(x) = \mu |x - x'|^2$</td>
</tr>
<tr>
<td>$\nabla f(x)$ is $\beta$-cocoercive</td>
<td>$\nabla f(x)$ is output-strictly IP, $\rho(x) = \beta |\nabla f(x) - \nabla f(x')|^2$</td>
</tr>
</tbody>
</table>

Table 2.1: monotone and passivity comparison

The table is constructed by setting $\dot{V}(x)$ to 0 because the system is just a static map, and selecting appropriate $\varphi(x)$ and $\rho(x)$ for the input/output-strictly definitions. For example $\nabla f(x)$ is $\mu$-strongly monotone if

$$(x - y)^T(\nabla f(x) - \nabla f(y)) \geq \mu \|x - y\|^2 \quad \forall x, y \in \mathbb{R}^n$$

A system is input-strictly IP if

$$\dot{V}(x) \leq (u - u')^T(\nabla f(u) - \nabla f(u')) - \varphi(u - u') \quad \forall u, u' \in \mathcal{U}$$

The system is static so replacing $\dot{V}(x)$ with zero and defining $\varphi(x) = \mu \|x - x'\|^2$ the inequality becomes,

$$0 \leq (u - u')^T(\nabla f(u) - \nabla f(u')) - \mu \|u - u'\|^2 \quad \forall u, u' \in \mathcal{U}$$

$$\mu \|u - u'\|^2 \leq (u - u')^T(\nabla f(u) - \nabla f(u'))$$

This inequality matches the definition of strong monotonicity and are therefore equivalent when the system is a static map. This relationship will used in Chapter 4 to show the similarities between convex optimization and game theory.

### 2.5 Graph Theory

In this section we introduce some basic elements from graph theory and introduce the Laplacian matrix, where the following are from [43] and [44]. The Laplacian will be used in later chapters because it describes the communication between agents over a network. We focus on undirected graphs because we will assume that if agent $i$ can communicate with agent $j$ then agent $j$ can communicate back to $i$. 
A graph $G_c$ is a 2-tuple $G_c = (\mathcal{I}, \mathcal{E})$ with vertex set, $\mathcal{I} = \{1, \ldots, N\}$ where $N$ is the number of vertices, and $\mathcal{E} = \{e_1, \ldots, e_n\} \subseteq \mathcal{I} \times \mathcal{I}$ is the edge set. The edge set contains ordered pairs of distinct vertices such that the pair $e_k = (i, j) \in \mathcal{E}$ represents the $k$th edge which is directed from vertex $i$ to vertex $j$. An undirected graph has an unordered pair of distinct vertices in the edge set. In the game setup, the communication graph is undirected and we associate a vertex with a player/agent. An edge between agents $i, j \in \mathcal{I}$ exists if agent $i$ and $j$ exchange information.

The indegree of vertex $i$ is the number of edges directed to vertex $i$. The outdegree of vertex $i$ is the number of edges directed out from vertex $i$. For an undirected graph the indegree and outdegree are the same and is called the degree of vertex $i$, i.e., $\deg(i)$. The set $\mathcal{N}_i \subset \mathcal{I}$ represents the set of adjacent vertices / neighbours of vertex $i$ and $|\mathcal{N}_i| = \deg(i)$. The degree matrix $D$ contains the degree of each vertex along its diagonal, i.e., $[D]_{i,i} = \deg(i)$ and zero otherwise.

A path from vertex $i$ to vertex $j$ is a sequence of distinct vertices starting at $i$ and ending at $j$ such that consecutive vertices are connected via an edge, i.e.,

$$(v_1, v_2), (v_2, v_3), \ldots, (v_{l-1}, v_l), (v_l, v_{l+1}) \quad \forall i, (v_i, v_{i+1}) \in \mathcal{E}$$

where $v_1$ is the starting vertex and $v_{l+1}$ is the final vertex on the path. A graph is connected if there is a path between every pair of vertices.

**Definition 2.19 (Adjacency Matrix).** Given a graph $G_c$ the adjacency matrix $A \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ is,

$$[A]_{i,j} = \begin{cases} 
1 & (i, j) \in \mathcal{E} \\
0 & \text{otherwise}
\end{cases}$$

The adjacency matrix encodes information about which vertices are connected by an edge. Similarly the incidence matrix describes which vertex is connected to which edge.

**Definition 2.20 (Incidence Matrix of an directed graph).** Given a directed graph $G_c$ the Incidence matrix $Q \in \mathbb{R}^{\mathcal{I} \times |\mathcal{E}|}$ is,

$$[Q]_{i,j} = \begin{cases} 
1 & i = l, \text{ where } e_j = (k, l) \in \mathcal{E} \\
-1 & i = k, \text{ where } e_j = (k, l) \in \mathcal{E} \\
0 & \text{otherwise}
\end{cases}$$

For an undirected graph an arbitrary direction on the edges can be imposed to create the incidence matrix. The Laplacian is constructed from the incidence matrix and the arbitrary direction imposed on the incidence matrix does not change the Laplacian when the graph is undirected.

**Definition 2.21 (Laplacian Matrix).** Given a undirected graph $G_c$ the Laplacian matrix $L \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ is,
Chapter 2. Background

When the graph \( G_c \) is directed, the \( \text{deg}(i) \) will be changed to either the indegree or outdegree of the vertex. When \( G_c \) is an undirected graph the relationship between these matrices is \( L = QQ^T = D - A \). Below is an example of these matrices for a simple graph.

\[
\begin{align*}
D &= \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, & A &= \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, & Q &= \begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, & L &= \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}
\end{align*}
\]

The importance of the Laplacian matrix is that it describes the connectivity of the graph \( G_c \). When \( G_c \) is an undirected and connected graph, 0 is a simple eigenvalue of \( L \) with corresponding eigenvector \( \mathbf{1}_N \) [Theorem 2.1(c) [45]]. Therefore, \( L\mathbf{1}_N = 0 \) and because \( L \) is symmetric \( \mathbf{1}_N^TL = 0 \), Thus,

\[
\begin{align*}
\text{Null}(L) &= \text{span}\{\mathbf{1}_N\}, & \text{Range}(L) &= \text{Null}(\mathbf{1}_N^T) & \text{Range}(L) &= \text{Null}(L)^\perp & (2.15)
\end{align*}
\]

The smallest eigenvalue of a matrix \( L \) is denoted \( \lambda_i(L) \) or simply \( \lambda_i \). The eigenvalues of \( L \) in ascending order are \( 0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N \).

\[
\begin{align*}
\min_{x \neq 0, \ 1_N^Tx = 0} x^TLx &= \lambda_2 \|x\|_2^2 \\
\max_{x \neq 0} x^TLx &= \lambda_N \|x\|_2^2
\end{align*}
\]

which means that the Laplacian is also positive semi-definite matrix. If a graph has \( k \) connected components then the rank of \( L \) is \( N - k \) [Lemma 13.1.1, [43]]. This implies that for a graph
with \( k \) connected components \( \lambda_1 = \cdots = \lambda_k = 0 \) and for a connected graph \( \lambda_2 > 0 \). The second smallest eigenvalue of the Laplacian matrix is also referred to as the algebraic connectivity because it determines if the graph is connected or not.

From [44] lower bounds for the algebraic connectivity for some known graphs are, and lower bounds for the algebraic connectivity for general graphs are,

1. \( \lambda_2 \geq \frac{4}{\text{diam}(G)} \), where \( \text{diam}(G) \) is the diameter of the graph and equals the longest shortest path between two nodes.

2. \( 2e(G)(1 - \cos(\frac{\pi}{n})) \), where \( e(G) \) is the edge connectivity and is equal to the minimum number of edges needed to be removed to disconnect the graph.

From [46], some upper bounds for the largest eigenvalue of the Laplacian, \( \lambda_n \) are,

1. \( \lambda_n \leq N \)

2. \( \lambda_n \leq \max_{(i,j)\in E} |\mathcal{N}_i \cup \mathcal{N}_j| \)

3. \( \lambda_n \leq \max_{(i,j)\in E} \mathcal{N}_i + \mathcal{N}_j - |\mathcal{N}_i \cup \mathcal{N}_j| \)

Lastly, removing an edge from a graph \( G_c \) decreases the algebraic connectivity. The theorem below gives a more general result, showing that all the eigenvalues will decrease.

**Theorem 2.20** (Edge Addition (Thm 13.6.2 [43])). Let \( G \) be a graph with \( n \) vertices and let \( G_2 \) be obtained from \( G \) by adding an edge joining two distinct vertices of \( G \). Then, \( \lambda_i(G) \leq \lambda_i(G_2) \), \( \forall i \in [1,n] \). Additionally, \( \lambda_i(G_2) \leq \lambda_{i+1}(G) \) for \( i < n \).

### 2.6 Consensus Algorithm

If each vertex of the graph \( G_c \) is assigned a value then the Laplacian can be used to measure the sum of the differences between the vertex’s value and its neighbours. Let \( x_i \) represent the value at vertex \( i \) and \( x = [x_1^T, \cdots, x_N^T]^T \) represent the vector of vertex values. The fact that the null space of \( L \) for a connected graph is \( 1_N \) means that \( Lx = 0 \) for some \( x \in \mathbb{R}^N \) when all of the components of \( x \) have the same value, i.e., \( x_1 = x_i = \cdots = x_N = \alpha \) for some \( \alpha \in \mathbb{R} \). The Laplacian measures how much disagreement there is between components in the vector \( x \). This suggests that the Laplacian can be used to adjust \( x_i \)'s value to move towards vertex \( i \)'s neighbours, \( \mathcal{N}_i \), to reach a consensus.

<table>
<thead>
<tr>
<th>Graph ( f(x) )</th>
<th>Algebraic Connectivity, ( \lambda_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete Graph</td>
<td>( \lambda_2 = n )</td>
</tr>
<tr>
<td>( \lambda_2 = 2(1 - \cos(\frac{\pi}{n})) )</td>
<td></td>
</tr>
<tr>
<td>( \lambda_2 = 2(1 - \cos(\frac{2\pi}{n})) )</td>
<td></td>
</tr>
<tr>
<td>Path</td>
<td></td>
</tr>
<tr>
<td>Cycle</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2: Graphs with known \( \lambda_2 \) [44]
The Laplacian consensus algorithm or Average consensus algorithm is,

\[
\dot{x} = -Lx
\]

\[
\dot{x}_i = \sum_{j \in N_i} x_j - x_i \quad \forall i \in \mathcal{I}
\]

It is clear from the equation for \(\dot{x}_i\) that the Laplacian measures the difference between \(x_i\) and the neighbouring vertices. Notice that the difference between \(x_i\)'s value and the average of the neighbouring vertices, \(N_i\), is

\[
\frac{1}{|N_i|} \left( \sum_{j \in N_i} x_j \right) - x_i
\]

When scaling (2.18) by a factor of \(N_i\) this equation becomes exactly the same as \(\dot{x}_i\) in (2.17). Therefore, row \(i\) of the dynamics in (2.17) can be considered as moving towards the weighted averaged of the difference between \(x_i\) and vertex \(i\)'s neighbours.

It can be shown that the dynamics (2.17) convergences to consensus by using the Lyapunov function \(V(x) = x^T Lx\). Additionally, it can be shown that the convergence point is, \(\bar{x} = \text{average}(x(0))\), which is the average of the initial conditions, [Theorem 1, [47]].

2.7 Discussion

The results from this chapter will be used throughout the next chapter to define what is a game and the solution concept for a game. The existence of a solution is established based on the convexity properties (Background from section 2.1) of each players cost function. A simple dynamical system (Background from section 2.3) is presented and shown to converge for games that satisfy a monotonicity condition (Background from section 2.1) and this monotonicity condition allows the dynamical system to be decomposed into two EIP systems (Background from section 2.4). Additionally, games with restricted player actions sets use the projection operator (Background from section 2.2) to ensure actions remain in the action set. The consensus algorithm from section 2.6 will not be used in the next chapter but will used in Chapter 4.
Chapter 3

Continuous Kernel Games

In this chapter we define what is a game and the solution concept for a game, i.e., Nash equilibrium. The Nash equilibrium set is characterized for unconstrained and compact action sets. Theorems are presented that ensure that the Nash equilibrium exists and the assumptions used throughout the thesis are stated. Lastly, the gradient and projected gradient dynamics, for the complete information setting, are proved to converge for games with a monotone pseudo-gradient. Some of the material in this chapter are from [48], [37], [49], [50], [21], [22], [23], [3], [51], [38], and [35].

3.1 Game Formulation and Nash Equilibrium

Let a game be denoted by $G(I, J_i, \Omega_i)$ where $I = \{1, \ldots, N\}$ is the set of $N$ players (agents) involved in a game. The set $\Omega_i \subseteq \mathbb{R}^{n_i}$ is player $i$’s action set and an element of the set, $x_i \in \Omega_i$, is player $i$’s selected action. The dimension of agent $i$’s action set is $n_i$ and can be different for each agent.

For example, consider $N$ firms involved in producing a homogeneous commodity as the players (agents). Each firm may own a few different factories around the world and firm $i$ needs to determine the quantity to produce in each factory. Agent $i$’s action, $x_i$, is the amount produced by all its factories. The amount produced in factory $j$ for firm $i$ is $[x_i]_j$.

The overall action set of all the players is $\Omega = \prod_{i \in I} \Omega_i \subseteq \mathbb{R}^n$, and the overall dimension is $n = \sum_{i \in I} n_i$. Let $x = (x_1, \ldots, x_N) \in \Omega$ denote the action profile of all the agents or $N$-tuple. Equivalently, let $x = (x_i, x_{-i})$ where $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N) \in \Omega_{-i} = \prod_{j \in I \setminus \{i\}} \Omega_j$ is the $(N-1)$-tuple of all agents’ actions except for agent $i$’s. Alternatively, $x$ and $x_{-i}$ can be represented as the stacked vector $x = [x_1^T, \ldots, x_N^T]^T$ and $x_{-i} = [x_1^T, \ldots, x_{i-1}^T, x_{i+1}^T, \ldots, x_N^T]^T$ respectively. Each player (agent) $i$ aims to minimize their own cost function $J_i : \Omega \to \mathbb{R}$, which may depend on all other players’ actions, i.e., $J_i(x_i, x_{-i})$.

**Definition 3.1.** Given a game $G(I, J_i, \Omega_i)$, an action profile $x^* = (x_i^*, x_{-i}^*) \in \Omega$ is a Nash
Equilibrium (NE) of $\mathcal{G}$ if

$$\forall i \in \mathcal{I} (\forall y_i \in \Omega_i) \ J_i(x_{i}^*, x_{-i}^*) \leq J_i(y_i, x_{-i}^*)$$

At a Nash equilibrium, no agent has any incentive to unilaterally deviate from their current action. Any agent $i$, that changes their current strategy given $x_{-i}^*$ will not decrease its cost. If the cost function is $J_i(x) = x_i$ then there is no Nash equilibrium because agents can always pick a larger negative number and decrease their cost. Even if the action set is bounded a Nash equilibrium may not exist. If the action space is $\Omega_i = (0, 1)$ and the cost function is $J_i(x) = x_i$ agents can always pick an action closer to 0 that will lower their cost.

![Figure 3.1: Best Response](image)

For example figure 3.1 shows a two player game where each curve represents the best action player $i$ can make given player $j$’s action. In figure 3.1(a) if player 1 is currently playing 2 then player 2 wants to play 4. If player 2 is playing 4 than play 1 wants to play 6. The two lines never intersect each other and therefore there is no NE for this game. On the other hand, in figure 3.1(b), if player 1 plays 1 and player 2 plays 0 then neither of the players want to change their action and is therefore a NE of the game. To ensure the existence of a Nash equilibrium for a game, additional assumptions on the cost functions and action sets are required.

**Corollary 3.1** (Existence of Nash Equilibria, Unconstrained (Cor 4.2, [48])).

For each $i \in \mathcal{I}$, let agent $i$’s action set be $\Omega_i = \mathbb{R}^{n_i}$. Let the cost function $J_i : \Omega_1 \times \cdots \times \Omega_N \to \mathbb{R}$ be jointly continuous in all its arguments and strictly convex in $x_i$ for every $x_j \in \Omega_j$, for all $j \neq i \in \mathcal{I}$. Furthermore, let $J_i(x_i, x_{-i})$ be radially unbounded with respect to $x_i$, i.e.,

$$\forall i \in \mathcal{I} (\forall x_{-i} \in \Omega_{-i}) \|x_i\| \to \infty \implies J_i(x_i, x_{-i}) \to \infty$$

Then, the associated $N$-person nonzero-sum game admits a Nash equilibrium.
When the action space is closed and bounded then the cost function no longer needs to satisfy the radially unbounded condition because the cost function is bounded. The proof of Corollary 3.1 and the following theorem are based on Brower’s fixed-point theorem.

**Theorem 3.2** (Existence of Nash Equilibria, Compact Set (Thm 4.3, [48])).

For each \( i \in \mathcal{I} \), let \( \Omega_i \) be a closed, bounded and convex subset of a finite-dimensional Euclidean space. The cost function \( J_i : \Omega \times \cdots \times \Omega_N \to \mathbb{R} \) is jointly continuous in all its arguments and strictly convex in \( x_i \) for every \( x_j \in \Omega_j \), for all \( j \neq i \in \mathcal{I} \). Then, the associated \( N \)-person nonzero-sum game admits a Nash equilibrium.

The following two basic convexity and smoothness assumptions are used throughout the thesis to ensure existence of a NE via the corollary or theorem above.

**Assumption 1** (Cost Function Convexity Conditions).

(i) For every \( i \in \mathcal{I} \), the action set for each agent is \( \Omega_i = \mathbb{R}^{n_i} \). The cost function \( J_i : \Omega \to \mathbb{R} \) is jointly continuous in all its arguments; and strictly convex, radially unbounded, and \( C^1 \) in \( x_i \), for every \( x_{-i} \in \Omega_{-i} \).

(ii) For every \( i \in \mathcal{I} \), the action set \( \Omega_i \) is a non empty, convex and compact subset of \( \mathbb{R}^{n_i} \). The cost function \( J_i : \Omega \to \mathbb{R} \) is jointly continuous in all its arguments; and (strictly) convex and \( C^1 \) in \( x_i \), for every \( x_{-i} \in \Omega_{-i} \).

Under Assumption 1(i) from Corollary 3.1 it follows that a NE \( x^* \) exists and under Assumption 1(ii) it follows from Theorem 3.2 that a NE exists. When the cost function \( J_i \) is differentiable with respect to \( x_i \) and the action set \( \Omega_i = \mathbb{R}^{n_i} \) then the NE satisfies

\[
\forall i \in \mathcal{I} \quad \nabla_i J_i(x_i^*, x_{-i}^*) = 0 \quad \text{or} \quad F(x^*) = 0_n
\]  

where \( \nabla_i J_i(x_i, x_{-i}) = \frac{\partial J_i}{\partial x_i}(x_i, x_{-i}) \in \mathbb{R}^{n_i} \), is the gradient of agent \( i \)'s cost function \( J_i(x_i, x_{-i}) \) with respect to its own action \( x_i \). The function \( F : \Omega \to \mathbb{R}^n \) is the pseudo-gradient and is defined as all agents’ partial gradients stacked,

\[
F(x) \triangleq [\nabla_1 J_1^T(x), \ldots, \nabla_N J_N^T(x)]^T
\]

A Nash equilibrium \( x^* \in \Omega \) satisfies the following variational inequality (VI).

**Proposition 3.3** (Nash Equilibrium Characterization (prop 1.4.2, [37])).

Let each \( \Omega_i \) be a closed convex subset of \( \mathbb{R}^{n_i} \). Suppose that for each fixed \( x_{-i} \), the function \( J_i(x_i, x_{-i}) \) is convex and continuously differentiable in \( x_i \). Then, \( x^* \) is a Nash equilibrium if and only if

\[
(x - x^*)^TF(x^*) \geq 0 \quad \forall x \in \Omega
\]  

(3.3)
Additionally, \((3.3)\) can be written as 
\[-F(x^*)^T(x - x^*) \leq 0\] and from the definition of the normal cone (Def. 2.7),

\[-F(x^*) \in N_{\Omega}(x^*) \tag{3.4}\]

Under just convexity of \(J_i\) with respect to \(x_i\), existence of an NE follows based on Kakutani’s fixed point theorem. Note that under just convexity, the NE point may no longer be unique, as shown in figure 3.1(b). The number of the Nash equilibriums can be determined by the monotonicity of \(F(x)\) and the action set \(\Omega_i\).

**Theorem 3.4** (Multiplicity of Nash Equilibrium (Theorem 3. [49])).

Given a game \(G(I, J_i, \Omega_i)\), let \(\Omega_i \subset \mathbb{R}^{n_i}\) be nonempty, closed and convex for each \(i \in I\). The function \(J_i(x_i, x_{-i})\) is continuously differentiable on \(\Omega\) and convex in \(x_i\) for every \(x_{-i} \in \Omega_{-i}\). Then, the following hold,

1. If the set \(\Omega_i\) is bounded \(\forall i \in I\) then the NE set is nonempty and compact.
2. If \(F(x)\) is monotone then the NE set is convex and possibly empty.
3. If \(F(x)\) is strictly monotone then the NE set has at most one point.
4. If \(F(x)\) is strongly monotone then the NE set is a unique point.

The above setting refers to players’ strategic interactions, but it does not specify what knowledge or information each player has. Since each players cost function, \(J_i\), may depend on all the players’ actions, an introspective calculation of an NE requires complete information where each player knows the cost functions and strategies of all other players (see Definition 3.1 and equation (3.1)). In this case, agents could immediately play the NE strategy because each agent knows all the cost functions. A game with incomplete information refers to players not fully knowing the cost functions or strategies of the other players, [50]. Throughout the thesis, we assume \(J_i\) is known by player \(i\) only. In a game with incomplete but perfect information, each agent has knowledge of the actions of all other players, \(x_{-i}\). We refer to the case when players are not able to observe the actions of all the other players, as a game with incomplete and imperfect or partial information.

This is the setting we consider in this thesis: we assume players can communicate only locally, with their neighbours. Our goal is to derive dynamics for seeking an NE in the incomplete, partial information case, over a communication graph \(G_c\). To ensure that information about all agents strategies is eventually obtained an assumption on the communication graph is imposed. The information sharing between agents is described by an undirected graph \(G_c = (I, E)\) or \(G_c\).

**Assumption 2.** The communication graph \(G_c\) is connected.
Without this assumption if agents $i, j \in \mathcal{I}$ are not connected via a path on the communication graph they will never learn each others action. Without having all the actions, players will have to use an algorithm that uses payoff information to update their strategy to reach the NE.

Now that a game is defined and the conditions for the existence of a Nash equilibrium are stated, the following sections will present dynamics that converge to the NE in the perfect information setting.

### 3.2 Gradient-Play Dynamics with Perfect Information

From the definition of the Nash equilibrium (Def 3.1) each agent is playing an action that minimize their cost function such that no other action, given the other players’ actions, can reduce you reduce their cost. From the perspective of agent $i$, he is trying to solve the optimization problem of finding an $x_i$ that minimizes $J_i(x_i, x_{-i})$ given $x_{-i}$. Since each player is trying to find the minimum of a function, the problem looks similar to an optimization problem. Therefore, it is natural to see what methods are using in optimization to find the minimum of a cost function. The convex optimization problem to find the minimum of an objection function $J(x)$ is,

$$\min_x J(x)$$

$$s.t. \quad x \in \Omega$$

where $x \in \mathbb{R}^n$, $\Omega \subset \mathbb{R}^n$ is a convex set, and $J : \mathbb{R}^n \to \mathbb{R}$ is convex. A method used for solving this problem when $\Omega = \mathbb{R}^n$ is gradient descent,

$$\dot{x} = -\frac{\partial}{\partial x} J(x)$$

The Nash equilibrium and its characterizations (3.1) and (3.3) are given by an equality/inequality of the gradient of the cost function. This suggests that by using a gradient method, that the cost of each agent will decrease and potentially reach a Nash Equilibrium. There are however some major differences between the convex optimization problem and the game problem,

$$(\forall i \in \mathcal{I}) \min_{x_i} J_i(x_i, x_{-i})$$

$$s.t. \quad x_i \in \Omega_i$$

In convex optimization, gradient descent methods work because of the convexity of the cost function with respect to the whole input variable, $x$. In the game setting, the cost function for agent $i$ only needs to be convex with respect to $x_i$ (Corollary 3.1 Proposition 3.3 Theorem 3.2 3.4). In convex optimization the state variable $x$ is updated by a centralized process. Even in distributed convex optimization, the entire state variable $x$ is updated by each agent solving its
sub optimization problem. In the game setting, each agent can only update their part of the full action profile of all agents, i.e., $x_i$ and not $x_i$. In a convex optimization problem, the solution is the minimum of a single cost function. In the game setting, the minimum of $N$ optimization problems with respect to just their action, $x_i$, is the Nash Equilibrium. Additionally, the cost function for each agent is dependent on the actions of the other agents. Therefore, every time an agent updates his action all agents need to solve a “new” optimization problem.

Even with these noticeable differences a gradient method like the one using in convex optimization is employed. In a game with perfect information a typical gradient-based dynamics for $\Omega_i = \mathbb{R}^{n_i}$, (21, 22, 23, 37), can be used by each player,

\[ P_i : \dot{x}_i = -\nabla_i J_i(x_i, x_{-i}), \quad \forall i \in I \]  \hspace{1cm} (3.5)

or,

\[ P : \dot{x} = -F(x) \]  \hspace{1cm} (3.6)

which represents the overall system, with all the agents’ dynamics stacked together. Assumption 1(i) ensures existence and uniqueness of solutions of (3.5). Note that the dynamics of all the players are coupled one with another. Additionally, the dynamics (3.5) requires all-to-all instantaneous information exchange between players or over a complete communication graph.

With just Assumption 1 there is no guarantee that (3.5) will be able to converge to the Nash Equilibrium. For example in a two player game let the cost functions be,

\[ J_1(x_1, x_2) = \frac{1}{2}(x_1 + 2x_2)^2 \]
\[ J_2(x_1, x_2) = \frac{1}{2}(x_2 + 3x_1)^2 \]

where $x_1, x_2 \in \mathbb{R}$ and the unique NE is $(0, 0)$. These cost functions satisfy Assumption 1(i) but when looking at the vector field of the pseudo-gradient,

\[ F(x) = Ax = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} x \quad \text{eig}(A) = 1 \pm \sqrt{6} \]

we see in figure 3.2 that that the NE point is a saddle point and therefore will be unstable.

The example above shows that additional assumptions must be made on the cost functions to ensure convergence to the NE. Convergence of (3.5) is typically shown under strict (strong) monotonicity on the pseudo-gradient $F$, (37, 22), or under strict diagonal dominance of its Jacobian evaluated at $x^*$, (3).

**Assumption 3 (Pseudo-Gradient Conditions).**

(i) The pseudo-gradient $F : \Omega \to \mathbb{R}^n$ is strictly monotone, $(x - x')^T(F(x) - F(x')) > 0$, $\forall x \neq x'$.

(ii) The pseudo-gradient $F : \Omega \to \mathbb{R}^n$ is strongly monotone, $(x - x')^T(F(x) - F(x')) \geq \mu \|x -
Figure 3.2: Vector Field

\[ x'\|_{2}^{2}, \forall x, x' \in \Omega, \text{ for } \mu > 0; \text{ and Lipschitz continuous, } \|F(x) - F(x')\| \leq \theta\|x-x'\|, \forall x, x' \in \Omega, \text{ where } \theta > 0. \]

Under Assumption 3(i) or 3(ii), the game has a unique NE via Theorem 3.4

3.2.1 Convergence of Gradient-Play Dynamics with Perfect Information

The gradient dynamics under Assumption 3 has a passivity interpretation. The gradient dynamics \( P (3.6) \) is the feedback interconnection between a bank of \( N \) integrators and the static pseudo-gradient map \( F(\cdot) \), as depicted in figure 3.3. The bank of integrators, \( \Sigma \), is MEIP with

\[
\dot{x} = u \\
y = x
\]

Figure 3.3: Gradient Dynamics (3.5) as a Feedback Interconnection of Two EIP Systems

storage function \( V(x) = \frac{1}{2}\|x - \pi\|^2 \). The function \( F(\cdot) \) is a static function and, under Assumption 3(ii), is monotone. Therefore, \( F(\cdot) \) is an incrementally passive (EIP) function. Hence, their interconnection is also EIP by Lemma 2.17 and asymptotic stability can be shown using \( \Sigma \)'s
storage function.

Lemma 3.5. Consider a game $G(\mathcal{I}, J, \Omega_i)$ with perfect information and under Assumption 3(i). The equilibrium point, $\bar{x}$, of the gradient dynamics, (3.5) or (3.6), is the NE of the game, i.e., $\bar{x} = x^*$. Moreover, under Assumption 3(i) the NE is globally asymptotically stable. Alternatively, under Assumption 3(ii), $x^*$ is globally exponentially stable.

Proof. The equilibrium point, $\bar{x}$, of (3.6) satisfies the equation, $F(\bar{x}) = 0$. From the Nash equilibrium characterization given by equation (3.1), $\bar{x} = x^*$ and is a NE point of $G(\mathcal{I}, J, \Omega_i)$.

Consider the quadratic Lyapunov function $V : \Omega_i \to \mathbb{R}$, where $V(x) = \frac{1}{2} \|x - \bar{x}\|^2$. Along the solution trajectory of (3.6) and using $F(\bar{x}) = 0$,

$$\dot{V}(x) = -(x - \bar{x})^T (F(x) - F(\bar{x}))$$

$$< 0 \quad \forall x \neq \bar{x} \in \Omega$$

where the last line follows from Assumption 3(i). Therefore, $\dot{V}(x) \leq 0$ and $\dot{V}(x) = 0$ only if $x = \bar{x} = x^*$. Since $V$ is radially unbounded, the conclusion follows from LaSalle’s invariance principle, i.e., Corollary 2.11 Under Assumption 3(ii), $\dot{V}(x) \leq -\mu \|x - \bar{x}\|^2$, $\forall x \in \Omega$ and global exponential stability follows from Theorem 2.9.

3.3 Projected Gradient Dynamics with Perfect Information

In the previous section, the action set for each player is $\mathbb{R}^n_i$. In this section we deal with compact action sets, under Assumption 1(ii), by using projected gradient dynamics. We highlight the major steps for proving convergence to the NE. The steps used to prove convergence for the perfect information case will be used to prove convergence for the dynamics introduced in the next chapter. Some of the following results come from [35].

In a game $G(\mathcal{I}, J_i, \Omega_i)$ with perfect information and a compact $\Omega_i$ action set, each player $i \in \mathcal{I}$ runs the projected gradient-based dynamics, given as [22], [23],

$$\mathcal{P}_i : \quad \dot{x}_i = \Pi_{\Omega_i}(x_i, -\nabla_i J_i(x_i, x_{-i})), \quad x_i(0) \in \Omega_i$$

(3.7)

In the perfect information case each agent can use the projected gradient dynamics (3.7) because the actions of all other players is known. Notice that the projected gradient dynamics move in the direction of the negative gradient when $x_i \in \text{int}(\Omega_i)$. The projected gradient dynamics are exactly the same as the dynamics in (3.6) when inside the interior of the action set. However, on the boundary of the action set, if $-\nabla_i J_i(x_i, x_{-i})$ points out of $\Omega_i$ then the projection operator will be active. The gradient will be projected to the closest vector in the tangent cone at $x_i$.

The overall system, of all agents’ projected dynamics, in stacked vector notation is given by

$$\mathcal{P} : \quad \dot{x}(t) = \Pi_{\Omega}(x(t), -F(x(t))), \quad x(0) \in \Omega$$

(3.8)
or, equivalently,

\[ \mathcal{P} : \dot{x}(t) = P_{T_\Omega(x(t))}[-F(x(t))] \quad x(0) \in \Omega \quad (3.9) \]

where equivalence follows from Lemma 2.4 in Chapter 2, or directly by Proposition 1 and Corollary 1 in [51]. Furthermore, this is equivalent to the differential inclusion

\[ -F(x(t)) - \Pi_\Omega(x(t), -F(x(t))) \in N_\Omega(x(t)). \quad (3.10) \]

For all the dynamics above, the projection operator is discontinuous on the boundary of \( \Omega \). We therefore use the standard definition of a solution of a projected dynamical system, i.e., Definition 2.10.

### 3.3.1 Existence and Uniqueness of Projected Gradient Dynamics

The existence of a unique solution of (3.8) is guaranteed for any \( x(0) \in \Omega \), under Lipschitz continuity of \( F \) on \( \Omega \) by Theorem 2.6. Note that any solution must necessarily lie in \( \Omega \) for almost every \( t \). Alternatively, existence holds under continuity and (hypo) monotonicity of \( F \), i.e., for some \( \mu \leq 0 \),

\[ (x - x')^T (F(x) - F(x')) \geq \mu \|x - x'\|^2, \quad \forall x, x' \in \Omega \]

(see Assumption 2.1 in [35], and also Theorem 1 in [51] for an extension to non-autonomous systems). This is similar to the QUAD relaxation in [52], i.e., a function \( f : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^n \) is QUAD(\( \Delta, \omega \)) if

\[ (x - y)^T [f(x, t) - f(y, t)] - (x - y)^T \Delta(x - y) \leq -\omega(x - y)^T (x - y) \quad \forall x, y \in \mathbb{R}^n \]

for some \( \omega > 0 \) and diagonal matrix \( \Delta \in \mathbb{R}^{n \times n} \). This means that \( F(x) - \mu x \) is monotone, where \( \mu \leq 0 \). Note that \( F(x) + \eta x \) is strongly monotone for any \( \eta > -\mu \), with \( \eta + \mu > 0 \) monotonicity constant. When \( \mu > 0 \) we can take \( \eta = 0 \) and recover \( \mu \)-strong monotonicity of \( F \). Thus under Assumption \( \Pi \) and \( \beta \), for any \( x(0) \in \Omega \), there exists a unique solution of (3.8) and moreover, a.e., \( \dot{x}(t) \in T_\Omega(x(t)) \) and \( x(t) \in \Omega \).

Equilibrium points of (3.8) coincide with the Nash equilibria, which are solutions of the VI(\( \Omega, F \)), by Theorem 2.4 in [35]. To see this, let \( \bar{x} \) be an equilibrium point of (3.8) such that \( \{x \in \Omega | 0_n = \Pi_\Omega(x, -F(x))\} \). By Lemma 2.3, if \( \bar{x} \in \text{int}\Omega \), then \( 0_n = \Pi_\Omega(\bar{x}, -F(\bar{x})) = -F(\bar{x}) \) and is clearly a NE point. If \( \bar{x} \in \partial\Omega \), then

\[ 0_n = \Pi_\Omega(\bar{x}, -F(\bar{x})) = -F(\bar{x}) - \beta n \quad (3.11) \]

for some \( \beta > 0 \) and \( n \in n(\bar{x}) \subset N_\Omega(\bar{x}) \). Equivalently, by (3.10), \( -F(\bar{x}) \in N_\Omega(\bar{x}) \). Thus
\(-F(\overline{x}) = \beta n, \) for \(\beta > 0\) and \(n \in N_{\Omega}(\overline{x}).\) Using the definition of \(N_{\Omega}(\overline{x}),\) it follows that

\[-F(\overline{x})^T(x - \overline{x}) \leq 0, \quad \forall x \in \Omega\]

Comparing this inequality to \((3.3),\) or \((3.4),\) it follows that \(\overline{x} = x^*.\) Thus, the equilibria points of \((3.8)\) \(\{x^* \in \Omega \mid 0_n = \Pi_{\Omega}(x^*, -F(x^*))\}\) coincide with Nash equilibria \(x^*\).

### 3.3.2 Convergence of Projected Gradient Dynamics

The last section showed that there is a solution to \((3.8)\) and that the equilibrium points of this system are NE points. In this section we show that the projected gradient dynamics \((3.8)\) converge to the NE points.

**Lemma 3.6.** Consider a game \(G(I, J, \Omega_i)\) with perfect information, under Assumptions \(I(ii)\) and \(J(i).\) Then, for any \(x_i(0) \in \Omega_i,\) the solution of \((3.7)\) or \((3.8)\) converges asymptotically to the NE of the game \(x^*.\) Under Assumption \(J(ii)\) convergence is exponential.

**Proof.** The proof follows from Theorem 3.6 and 3.7 in [35]. Consider any \(x(0) \in \Omega\) and \(V(t, x) = \frac{1}{2} \|x(t) - x^*\|^2,\) where \(x^*\) is the Nash equilibrium of the game, and \(x(t)\) is the solution of \((3.8).\) Then, the time derivative of \(V\) along solutions of \((3.8)\) is

\[\dot{V} = (x(t) - x^*)^T \Pi_{\Omega}(x(t), -F(x(t)))\]  

(3.12)

Since \(T_{\Omega}(x(t)) = [N_{\Omega}(x(t))]^o,\) by Theorem 2.4, the pseudo-gradient \(-F(x(t))\) can be decomposed as in \((2.2)\) into normal and tangent components, \(N_{\Omega}(x(t))\) and \(T_{\Omega}(x(t))\). From \((3.8)/(3.9)\) the equivalence shows that \(\Pi_{\Omega}(x(t), -F(x(t)))\) is in the tangent cone \(T_{\Omega}(x(t))\). It follows from \((3.10)\) that

\[-F(x(t)) \in N_{\Omega}(x(t)) + T_{\Omega}(x(t))\]

\[-F(x(t)) - \Pi_{\Omega}(x(t), -F(x(t))) \in N_{\Omega}(x(t))\]  

(3.13)

From the definition of the normal cone \(N_{\Omega}(x(t))\) this means,

\[(x' - x(t))^T(-F(x(t)) - \Pi_{\Omega}(x(t), -F(x(t)))) \leq 0 \quad \forall x, x' \in \Omega\]  

(3.14)

\[(x(t) - x')^T \Pi_{\Omega}(x(t), -F(x(t))) \leq -(x(t) - x')^T F(x(t))\]  

(3.15)

Thus, it follows that for \(x' = x^*\) and \(\forall x(t) \in \Omega,\)

\[(x(t) - x^*)^T \Pi_{\Omega}(x(t), -F(x(t))) \leq -(x(t) - x^*)^T F(x(t))\]  

(3.16)

From \((3.3),\) at the Nash equilibrium \(F(x^*)^T(x(t) - x^*) \geq 0, \forall x(t) \in \Omega.\) Therefore, adding \((3.3)\)
to the right-hand side of (3.16)

\[(x(t) - x^*)^T \Pi \Omega (x(t), -F(x(t))) \leq -(x(t) - x^*)^T (F(x(t)) - F(x^*))\]

Substituting the above inequality into (3.12) yields that along solutions of (3.8), for all \( t \geq 0 \),

\[
\dot{V} = (x(t) - x^*)^T \Pi \Omega (x(t), -F(x(t))) \\
\leq -(x(t) - x^*)^T (F(x(t)) - F(x^*)) \\
< 0 \quad \forall x(t) \neq x^* \in \Omega
\]

where the strict inequality follows from Assumption 3(i). Hence \( V(t) \) is monotonically decreasing and non-negative, and thus there exists \( \lim_{t \to \infty} V(t) = \underline{V} \). As in Theorem 3.6 in [35], a contradiction argument can be used to show that \( \underline{V} = 0 \), hence for any \( x(0) \in \Omega, \|x(t) - x^*\| \to 0 \) as \( t \to \infty \).

Under Assumption 3(ii), along solutions of (3.8), \( \dot{V} \leq -\mu \|x(t) - x^*\|^2 = -\mu V(t) \), for all \( t \geq 0 \) and for any \( x(0) \in \Omega \). From Theorem 2.9, exponential convergence follows immediately.

In this chapter gradient-play dynamics and projected gradient dynamics are shown to converge for perfect information games. The perfect information condition is paramount for the use of these algorithms. Both (3.5) and (3.7) contain the term \( \nabla_i J_i(x_i, x_{-i}) \), which is dependent on \( x_{-i} \). This term means that all agents must know the actions of all other players. In a large network setting, having knowledge of all the other players actions is not practical/possible. In the next chapter the perfect information requirement is relaxed to partial information. In this setting agents can communicate to neighbouring players under a connected communication graph assumption.
Chapter 4

Augmented Gradient Dynamics with Estimate Correction

In this chapter we address the primary objective of this thesis.

Problem:

Given a game \( G(I, J_i, \Omega_i) \) satisfying assumption \( \text{(1)} \) and assumption \( \text{(3)} \), redesign a gradient-based dynamics such that it converges to a Nash equilibrium of the game under incomplete and partial information over a connected undirected communication graph \( G_c \) (Assumption \( \text{(2)} \)).

In other words, this thesis determines how we can modify the gradient dynamics (3.5) such that it converges to a NE in a networked information setting, over some connected communication graph \( G_c \). The communication graph is not complete and there is no all-to-all instantaneous information exchange.

To approach this problem, we first recognize that if agents don’t have knowledge of the other agents actions, but still know their own cost function, agents are still able to use the gradient. Players will now require to communicate with their neighbours to learn the actions of the other agents.

We endow each player (agent) with a state vector that provides an estimate of all other players’ actions. In section \( \text{(4.1)} \) we define this state space, the extended pseudo-gradient and define some matrices for separating the action and estimate component from the state vector. Initially players will not know what the other players actions are but over time will learn the actions of others. Agents then update their estimate via some connected communication graph as in Assumption \( \text{(2)} \). This assumption ensures that agents are able to learn the actions used by all other agents. Without this assumption agents have no guarantee that they will reach a
Nash equilibrium. If all the agents are able to obtain the actions of all other players then the
dynamics will be the same as the perfect information case.

In Section 4.2 we reformulate the Nash equilibrium problem as a multi-agent agreement
problem between the players. If all the agents are able to come to an agreement for their state
variable then all agents will have perfect information of the other players actions. We exploit
the incremental passivity property of the extended pseudo-gradient.

In Section 4.3 we design a new signal for each player, based on the relative feedback from
its neighbours, such that these estimates agree with one another and converge to the NE. In
the following chapter we prove convergence of this algorithm by exploiting the monotonicity
condition on the pseudo-gradient (Assumption 3) and the Lipschitz condition on the extended
pseudo-gradient.

### 4.1 Player State Vector and Action/Estimate Component

Assume that player $i$ maintains an estimate of all players actions, $x^i = [(x^i_1)^T, \ldots, (x^i_N)^T]^T \in \Omega$ and $x^i_j \in \mathbb{R}^{n_j}$ is the estimate of player $j$’s action $\forall j \in \mathcal{I}$. Note that $x^i_i = x_i \in \mathbb{R}^{n_i}$ is player $i$’s actual action and is called player $i$’s action component (vector). We call the augmented vector of estimates and actions, player $i$’s state vector $x^i = [(x^i_1)^T, \ldots, (x^i_N)^T]^T = (x^i_i, x^i_{-i})$. The vector $x^i_{-i} = [(x^i_1)^T, \ldots, (x^i_{i-1})^T, (x^i_{i+1})^T, \ldots, (x^i_N)^T]^T \in \mathbb{R}^{n-n_i}$ is called the estimate component (vector) and represents player $i$’s state vector without their own action component ($x^i_i$). All agents’ estimate vectors are stacked into a single vector denoted $x = [(x_1)^T, \ldots, (x_N)^T]^T \in \Omega^N$ is called the overall state vector. The state space is now the $N$ Cartesian product of the action space, $\Omega^N = \prod_{i \in \mathcal{I}} \Psi_i \in \mathbb{R}^{Nn}$, where $\Psi_i = \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_{i-1}} \times \Omega_i \times \mathbb{R}^{n_{i+1}} \times \cdots \times \mathbb{R}^{n_N}$ and the total dimension of the expanded state space is $Nn$. When the action set $\Omega_i = \mathbb{R}^{n_i}, \forall i \in \mathcal{I}$ then $\Omega^N = \prod_{i \in \mathcal{I}} \Omega = \mathbb{R}^{Nn}$. Note that if all agents have the same estimate, $x^1 = \cdots = x^N = x \in \Omega$ then $x = 1_N \otimes x$. Let $x^*$ be a Nash equilibrium of the perfect information game, then $x^* = 1_N \otimes x^*$ in the expanded space.

To be able to talk about a player’s action and the estimate component separately, additional
notation is introduced. A set of matrices are defined that extract the action, $x^i = x_i$, and
estimate component, $x^i_{-i}$, from $x^i$. Additionally, operations on an (estimate) action vector
with dimension $(\mathbb{R}^{(N-1)n}) \mathbb{R}^n$ that result in a vector with the appropriate dimension, $\mathbb{R}^{Nn}$, is
defined. These matrices ensure that vectors of appropriate type are compared.

#### 4.1.1 Action and Estimate Component Matrices

Let Agent $i$’s action selection matrix is the non-square matrix defined as,

$$
\mathcal{R}_i \triangleq \begin{bmatrix}
0_{n_i \times n_i < i} & I_{n_i} & 0_{n_i \times n_i > i}
\end{bmatrix} \in \mathbb{R}^{n_i \times n} \tag{4.1}
$$
where,

\[
\begin{align*}
    n_{<i} &= \sum_{j<i} n_j, \quad n_{>i} = \sum_{j>i} n_j \\
\end{align*}
\]  

(4.2)

The matrix \( R_i \) returns only agent \( i \)’s action when applied to his estimate vector,

\[
R_i x^i = x_i
\]

The transpose of \( R_i \), \( R_i^T \), is agent \( i \)'s action alignment matrix. This matrix returns a vector where all elements are zero except for the action component that is equal to the input. The action selection matrix is defined as \( R \triangleq \text{diag}(R_1, \ldots, R_N) \). This matrix similarly returns the action vector of all the agents when applied to \( x \),

\[
R x = [(x_1^1)^T, \ldots, (x_N^N)^T]^T = [(x_1)^T, \ldots, (x_N)^T]^T = x
\]  

(4.3)

Example. Let \( G(I, J, \Omega_i) \) be a three player game where \( n_i = 1, \forall i \in I \). An element from the extended action space is \( x \in \mathbb{R}^9 \) and the actual action vector is \( x \in \mathbb{R}^3 \)

\[
R = \begin{bmatrix}
R_1 & 0_{1 \times 3} & 0_{1 \times 3} \\
0_{1 \times 3} & R_2 & 0_{1 \times 3} \\
0_{1 \times 3} & 0_{1 \times 3} & R_3
\end{bmatrix}, \quad R_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \quad R_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}
\]

\[
R_1 x^1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1^1 \\
x_2^1 \\
x_3^1
\end{bmatrix} = x_1^1 = x_1, \quad R_2^T x_2^2 = \begin{bmatrix} 0 \\
1 \\
0
\end{bmatrix} x_2^2 = \begin{bmatrix} 0 \\
x_2^2 \\
0
\end{bmatrix}
\]

\[
R x = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x_1^1 \\
x_2^1 \\
x_3^1 \\
x_1^2 \\
x_2^2 \\
x_3^2 \\
x_1^3 \\
x_2^3 \\
x_3^3
\end{bmatrix} = x \quad R^T x = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix} x = \begin{bmatrix}
x_1^1 \\
x_2^2 \\
x_3^3
\end{bmatrix}, \quad R^T x = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix} x = \begin{bmatrix}
x_1^1 \\
x_2^2 \\
x_3^3
\end{bmatrix}
\]

When applied to \( x^i \) the matrix \( S_i \) removes his own action component, \( x_i^i = x_i \), from the estimate.

Let Agent \( i \)’s estimate selection matrix be defined as,

\[
S_i \triangleq \begin{bmatrix}
I_{n<i} & 0_{n<i \times n_i} & 0_{n<i \times n_i} \\
0_{n>i \times n<g} & I_{n<g} & 0_{n<i \times n_i}
\end{bmatrix} \in \mathbb{R}^{(n-n_i) \times n}  
\]  

(4.4)
vector $x^i$, 
\[ S_i x^i = x^i_{-i} \]

The transpose of $S_i$, $S_i^T$, is agent $i$’s estimate alignment matrix. This matrix returns a vector where the action component is zero and the estimate component is equal to the input. The estimate selection matrix is defined as $\mathcal{S} \overset{\Delta}{=} \text{diag}(S_1, \ldots, S_N)$. This matrix similarly returns the estimate component vector of all the agents, i.e.,

\[ \mathcal{S}_x \overset{\Delta}{=} \left[ (x^1_{-1})^T, \ldots, (x^N_{-N})^T \right]^T = \mathcal{S} x \tag{4.5} \]

**Example.** Let $\mathcal{G}(I, J, \Omega)$ be a three player game where $n_i = 1$, $\forall i \in I$. An element from the extended action space is $x \in \mathbb{R}^9$ and the estimate component vector is $z = \left[ (x^1_1)^T, (x^2_2)^T, (x^3_3)^T \right]^T = \left[ x^1_2 \ x^1_3 \ x^2_1 \ x^2_3 \ x^3_1 \ x^3_2 \right]^T \in \mathbb{R}^6$

\[ \mathcal{S} = \begin{bmatrix} S_1 & 0_{2 \times 3} & 0_{2 \times 3} \\ 0_{2 \times 3} & S_2 & 0_{2 \times 3} \\ 0_{2 \times 3} & 0_{2 \times 3} & S_3 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad S_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad S_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \]

\[ S_1 x^1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^1_1 \\ x^1_2 \\ x^1_3 \end{bmatrix} = \begin{bmatrix} x^1_2 \\ x^1_3 \end{bmatrix}, \quad S_2^T x^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x^2_1 \\ x^2_3 \end{bmatrix} = \begin{bmatrix} x^2_1 \\ 0 \end{bmatrix} \]

\[ S x = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = z \]

\[ S^T z = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^1_1 \\ x^1_2 \\ x^1_3 \\ x^2_1 \\ x^2_2 \\ x^2_3 \end{bmatrix} = \begin{bmatrix} 0 \\ x^2_1 \\ x^2_2 \\ x^2_3 \end{bmatrix} \]
The matrix, \[ A = R^T R \] (4.6)
is the action filter matrix. Consider \( y = Ax \). Then, \( y^i = x^i \) and \( y^j = 0, \forall i \neq j \in \mathcal{I} \), i.e., it selects the action component and sets the estimate components to zero. The matrix \( E = S^T S \) is the estimate filter matrix. Then, \( y = Ex \) selects the estimate component and sets the action components to 0, i.e., \( y^i = x^i, \forall i \neq j \in \mathcal{I} \) and \( y^j = 0 \).

**Example.** Let \( G(\mathcal{I}, J_i, \Omega_i) \) be a three player game where \( n_i = 1, \forall i \in \mathcal{I} \). An element from the extended action space is \( x \in \mathbb{R}^9 \).

\[
R^T Rx = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} = \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
x_8 \\
x_9 \\
\end{bmatrix}
\]

\[
S^T Sx = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} = \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
x_8 \\
x_9 \\
\end{bmatrix}
\]

These matrices have very nice properties and will be used throughout the remainder of the thesis. One property is that the matrix norm of both the action and estimate selection matrices are equal to 1, i.e, \( \|R\| = \|S\| = 1 \). Some useful identities of these matrices are given below:

\[
R R^T = I_n, \quad S S^T = I_{(N-1)n}, \quad R^T R + S^T S = I_{Nn}, \quad R S^T = 0, \quad S R^T = 0 \quad (4.7)
\]

\[
\text{Null}(R) = \text{Range}(S^T), \quad \text{Null}(S) = \text{Range}(R^T)
\]

\[
\text{Null}(R^T) = 0, \quad \text{Null}(S^T) = 0
\]

\[
R_i R_i^T = I_n, \quad S_i S_i^T = I_{n-n_i}, \quad R_i S_i^T = 0, \quad S_i R_i^T = 0 \quad (4.9)
\]

\[
R_i R_j^T = 0, \quad S_i S_j^T = 0, \quad R_i S_j^T = \begin{bmatrix} 0 & I_{n_i} \end{bmatrix}, \quad \forall i \neq j \in \mathcal{I} \quad (4.10)
\]

\[
R_i S_j^T S_j R_k^T = I_{n_i}, \quad R_i S_j^T S_j R_k^T = 0_{n_i}, \quad \forall i \neq j, i \neq k \in \mathcal{I} \quad (4.11)
\]

---

**Chapter 4. Augmented Gradient Dynamics with Estimate Correction**

43
\[ x = \mathcal{R}(1_N \otimes x) = \mathcal{R}(1_N \otimes I_n)x \implies \mathcal{R}(1_N \otimes I_n) = I_n \] (4.12)

From (4.7), \( \mathcal{R}^T \mathcal{R} + S^T S = I_{Nn} \), a useful decomposition of the vector \( x \) into actions and estimates can be derived.

\[ x = I_{Nn}x = (\mathcal{R}^T \mathcal{R} + S^T S)x \]

From the definition of the action component, \( x \), and the estimate component, \( z \), given by equation (4.3) and (4.5), \( x \) can be written as,

\[ x = \mathcal{R}^T (\mathcal{R}x) + S^T (Sx) = \mathcal{R}^T x + S^T z \] (4.13)

**Lemma 4.1.** Let \( S = \text{diag}(S_1, \ldots, S_N) \in \mathbb{R}^{(N-1)n \times Nn} \) where \( S_i \) be defined in equation (4.4). Then, \( \|S(1_N \otimes I_n)\| = \sqrt{N-1} \).

**Proof.** Proof is provided in Appendix A, Lemma A.3.

### 4.2 Gradient Type Dynamics with Control Input

In the enlarged space the estimate components are different initially, but in the limit all players’ estimate vectors should be in consensus. Thus, we modify the gradient dynamics such that player \( i \) updates \( x^i \) to reduce their own cost function and updates \( x_{-i} \) and and \( x^i \) to reach a consensus with the other players.

#### 4.2.1 Gradient-Play Dynamics with Control Input

Let each player combine his gradient-type dynamics (3.5) with an integrator-type auxiliary dynamics, driven by some control signal, \( \tilde{\Sigma}_i \):

\[
\begin{aligned}
\dot{x}_i &= -\nabla_i J_i(x_i, x_{-i}) + B_i^i u_i \\
\dot{x}_{-i} &= 0 \\
y_i &= (B^i)^T x^i
\end{aligned}
\] (4.14)

where \( B^i \) is a full rank \( n \times n \) matrix. For each player, \( u_i \in \mathbb{R}^n \) is a control signal to be designed based on the output from his neighbours, such that in the limit \( x^i = x_j \), for all \( i, j \), and the estimate vectors converge to the NE, \( x^i \to x^* \), \( \forall i \in I \).

Thus, we have reformulated the design of NE dynamics over \( G_c \) as a multi-agent agreement problem. This continuous time, multi-agent agreement problem for continuous kernel games is novel and gives new insight for games over undirected communication graphs. To proceed, we first analyze properties of \( \tilde{\Sigma}_i \) and the overall agents’ dynamics \( \tilde{\Sigma} \). We can write \( \tilde{\Sigma}_i \) (4.14) in a compact form

\[
\tilde{\Sigma}_i : \begin{cases}
\dot{x}^i = -\mathcal{R}_i^T \nabla_i J_i(x^i) + B_i^i u_i \\
y_i = (B^i)^T x^i
\end{cases}
\] (4.15)
Note that agent dynamics in (4.15) are heterogenous, separable, but do not satisfy an individual passivity property as typically assumed in multi-agent literature, e.g. [27], [26]. In the networked game context such individual convexity assumptions are too restrictive. The overall agents’ dynamics denoted by \( \tilde{\Sigma} \) can be written in stacked form as

\[
\tilde{\Sigma} : \begin{cases} \\
 \dot{x} = -R^T F(x) + Bu \\
y = B^T x
\end{cases}
\]

where \( x = [(x_1)^T, \ldots, (x_N)^T]^T \in \mathbb{R}^{Nn}, \ u = [u_1^T, \ldots, u_N^T]^T \in \mathbb{R}^{Nn}, \ B = diag(B_1, \ldots, B_N) \in \mathbb{R}^{Nn \times Nn} \) and,

\[
F(x) : \Omega^N \rightarrow \mathbb{R}^n \\
F(x) \triangleq [\nabla_1 J_i^T(x_1), \ldots, \nabla_N J_N^T(x_N)]^T
\]

is the continuous extension of the pseudo-gradient \( F, (3.2) \) to the augmented space. Note that when the estimate vectors of all the agents are in agreement then \( F(1_N \otimes x) = F(x) \).

### 4.2.2 Projected Gradient Dynamics with Control Input

When each action set \( \Omega_i \) is compact we modify each player’s dynamics \( \tilde{\Sigma}_i \), in (4.14) using projected dynamics for the action components to \( \Omega_i \) as,

\[
\tilde{\Sigma}_i : \begin{cases} \\
 \dot{\bar{x}}_i = R_i (x_i, -\nabla_i J_i(x_i, x_{-i}) + R_i B^i u_i(t)) \\
y_i = (B^i)^T \bar{x}_i
\end{cases}
\]

where \( u_i(t) \in \mathbb{R}^n \) is a piecewise continuous function, to be designed.

The projected dynamics (4.18) are similar to (4.14), but a portion of the input is inside the projection while the estimate component dynamics are not. The action \( x_i \) must remain inside of the constraint set so the input that affects agent \( i \)'s action must be inside the projection. For example, if the constraint set is the output voltage level, leaving the constraint set temporarily could damage the system and must be avoided. Other constraint sets could be physical limitations on the system, where actions outside of \( \Omega \) can’t be actualized in the real world.

The estimate \( \bar{x}_{-i} \) does not need to be inside of a projection operator because the estimates are not real actions being taken. This allows the estimate \( \bar{x}_{-i} \) to be any value in \( \mathbb{R}^{(n-n_i)} \). Additionally, in order to use the projection for the estimates, knowledge of every agents’ action set is required. In some cases this may be possible, but it is more general to assume that this
information is not available. The dynamics $\tilde{\Sigma}_i$ in (4.18) can be written more compactly as

$$\tilde{\Sigma}_i : \begin{cases} \dot{x}_i = R_i^T \Pi_i (x_i, -\nabla_i J_i(x_i) + R_i B_i u_i) + S_i^T S_i B_i u_i \\ y_i = (B_i^i)^T x^i \end{cases}$$

(4.19)

and the overall dynamics for all players in stacked form is,

$$\tilde{\Sigma} : \begin{cases} \dot{x} = R^T \Pi (R x(t), -F(x(t)) + R B u(t)) + S^T S B u(t) \\ y = B^T x(t) \end{cases}$$

(4.20)

where $u(t) \in \mathbb{R}^{Nn}$ is piecewise continuous. Notice that when the projection is not active in (4.20) the dynamics are exactly the same as (4.16). The only difference between the dynamics is when $x \in \partial \Omega^N$.

### 4.2.3 Pseudo-Gradient vs Extended Pseudo-Gradient

In section 3.2.1 and section 3.3.2 Assumption 3, on the pseudo-gradient $F(x)$, was used to prove convergence. The monotonicity condition also ensured existence and uniqueness of the solution for the dynamics in (3.8). Additionally, the monotonicity condition shows how much each agent’s action effects the gradient of the other agents. For a two player game the monotone condition on $F(x)$ is,

$$(x_1 - y_1)^T [\nabla_1 J_1(x_1, x_2) - \nabla_1 J_1(y_1, y_2)] + (x_2 - y_2)^T [\nabla_2 J_2(x_2, x_1) - \nabla_2 J_2(y_2, y_1)] > 0$$

Observe that changing $x_1$ or $x_2$ affects both $\nabla_1 J_1$ and $\nabla_2 J_2$. Therefore, the monotonicity condition describes the amount of coupling between agents. In the expanded action space this monotonicity condition no longer holds on $F(x)$.

$$(x_1^1 - y_1^1)^T [\nabla_1 J_1(x_1^1, x_2^1) - \nabla_1 J_1(y_1^1, y_2^1)] + (x_2^2 - y_2^2)^T [\nabla_2 J_2(x_2^2, x_1^2) - \nabla_2 J_2(y_2^2, y_1^2)] \not> 0$$

Each agent’s estimate is no longer coupled with the other players’ estimate. This is apparent from the fact that the estimates in the first term have the superscript 1 and the second term have the superscript 2. Therefore, additional assumptions are needed to prove convergence and to ensure existence and uniqueness for the projected gradient dynamics. In the rest of the thesis we consider one of the following two assumptions on the extended pseudo-gradient $F$.

**Assumption 4.** [Expanded Pseudo-Gradient Conditions]

(i) The extended pseudo-gradient $F : \Omega^N \rightarrow \mathbb{R}^n$ is monotone,

$$(x - x')^T (F(x) - F(x')) \geq 0, \quad \forall x, x' \in \Omega^N$$
(ii) The extended pseudo-gradient \( F : \Omega^N \rightarrow \mathbb{R}^n \) is Lipschitz continuous,

\[
\| F(x) - F(x') \| \leq \theta \| x - x' \| , \quad \forall x, x' \in \Omega^N
\]

where \( \theta > 0 \).

Thus, existence of a unique solution of (4.20) is guaranteed under Assumption 4(i) or (ii) by Theorem 2.6 in Chapter 2 or Theorem 1 in [51]. Some examples of cost functions where Assumption (4)(i) hold are,

\[
J_i(x) = \sum_{k \in \Psi_i} x_i^2, \quad \forall i \in \mathcal{I}, \; \Psi_i \subset \mathcal{I}
\]

\[
J_i(x) = f_i(x_i) + g_i(x_{-i}) \quad \forall i \in \mathcal{I}, \; f_i(\cdot) \text{ is convex}
\]

\[
J_i(x) = \left( 1 + \sum_{k \in \Psi_i} x_k^2 \right) x_i \quad \forall i \in \mathcal{I}, \; \Psi_i \subset \mathcal{I}
\]

\[
J_i(x) = f_i(x_i) + h_i(x_{-i}) x_i \quad \forall i \in \mathcal{I}, \; f_i(\cdot) \text{ is convex}
\]

\[
J_i(x) = f_i(x_i) + h_i(x_{-i}) j_i(x_i) \quad \forall i \in \mathcal{I}, \; f_i(\cdot), j_i(\cdot) \text{ is convex}
\]

\[
h_i(y) \geq 0 \quad \forall y \in \Omega_{-i}
\]

When \( F(x) \) is \( C^1 \) and \( \Omega \) is compact then assumption 4(ii) will hold.

We compare these assumptions to similar ones used in distributed optimization and in multi-agent coordination control, respectively. First, note that Assumption 4(i) on the extended pseudo-gradient \( F \) holds under individual joint convexity of each \( J_i \) with respect to the full argument. In distributed optimization problems, each objective function is assumed to be strictly (strongly) jointly convex in the full vector \( x \) and the gradient to be Lipschitz continuous, e.g. [1]. Similarly, in multi-agent coordination control, it is standard to assume that individual agent dynamics are separable and strictly (strongly) incrementally passive, e.g. [27].

However, in a game context the individual joint convexity of \( J_i \) with respect to the full argument is too restrictive, unless we have a trivial game with separable cost functions. In general, \( J_i \) is coupled to other players’ actions while each player has under its control only its own action. This is a key difference versus distributed optimization or multi-agent coordination, one which introduces technical challenges.

However, we show that under Assumption 4(i), the overall \( \tilde{\Sigma} \) (4.16) is incrementally passive, hence EIP. Based on this, we design new dynamics which converges over any connected \( G_c \) (Theorem 5.1). Under the weaker conditions, Assumption 3(ii) on \( F \) and Assumption 3(ii) on \( F \), we show that the new dynamics converges over any sufficiently connected \( G_c \) (Theorem 5.3). We also note that Assumption 4(i) is similar to those used in [18], [19], while Assumption 4(ii) is weaker. Assumption 4(i) is the extension of Assumption 3(i) to the augmented space, for local communication over the connected graph \( G_c \). The weaker Assumption 4(ii) on \( F \), is the extension of Lipschitz continuity of \( F \) in Assumption 3(ii). We also note that these assumptions
could be relaxed to hold only locally around \( x^* \) in which case all results become local.

### 4.2.4 Monotone Extended Pseudo-Gradient and EIP

In this section we prove that using a monotonicity condition on the extended pseudo-gradient \( F \) that the dynamics (4.16) and (4.20) are EIP. The proof for (4.16) is straightforward and the proof for (4.20) provides the steps for obtaining a useful inequality that will be used throughout this thesis.

**Lemma 4.2.** Under Assumption 4(i), the overall system \( \tilde{\Sigma}, (4.16) \), is incrementally passive, hence EIP.

**Proof.** Consider two inputs \( u, u' \in U \). Let \( x, x' \in \Omega^N \) be the trajectories and \( y, y' \in \mathbb{R}^{Nn} \) by the outputs of \( \tilde{\Sigma}, (4.16) \) for the inputs \( u, u' \). Let the storage function be \( V(x, x') = \frac{1}{2}\|x - x'\|^2 \). Then, along solutions of (4.16),

\[
\dot{V} = -(x - x')^T \left[ R^T(F(x) - F(x')) + B(u - u') \right] 
\]

\[
= -(x - x')^T (F(x) - F(x')) + (y - y')^T(u - u') \tag{4.21}
\]

where the last line follows from equation (4.3) and \( y = B^T x \). Using Assumption 4(i) the first term is less than zero and

\[
\dot{V} \leq (y - y')^T(u - u')
\]

Thus by Definition 2.18, \( \tilde{\Sigma} \), is incrementally passive, hence EIP.

**Lemma 4.3.** Under Assumption 4(i), the overall system \( \tilde{\Sigma}, (4.20) \), is incrementally passive, hence EIP.

**Proof.** Consider two inputs \( u(t), u'(t) \in U \). Let \( x(t), x'(t) \in \Omega^N \) be the state trajectories and \( y(t), y'(t) \in \mathbb{R}^{Nn} \) the outputs of \( \tilde{\Sigma}, (4.20) \) for the inputs \( u(t), u'(t) \). Let the storage function be \( V(t, x, x') = \frac{1}{2}\|x(t) - x'(t)\|^2 \). Then, along solutions of (4.20),

\[
\dot{V} = (x(t) - x'(t))^T R^T \left[ \Pi_{\Omega}(Rx(t), -F(x(t)) + RBu(t)) \right. 
\]

\[
- \left. \Pi_{\Omega}(Rx'(t), -F(x'(t)) + RBu'(t)) \right] 
\]

\[
+ (x(t) - x'(t))^T S^T S B(u(t) - u'(t)) \tag{4.22}
\]

Using the properties of the matrix \( R \), (4.7), and following the steps in the proof of Lemma 3.6, the following holds for (4.20),

\[
\dot{x}(t) = R\dot{x}(t) = \Pi_{\Omega}(Rx(t), -F(x(t)) + RBu(t)) \in T_\Omega(Rx(t)) 
\]

\[
= \Pi_{\Omega}(x(t), -F(x(t)) + RBu(t)) \in T_\Omega(x(t)) \tag{4.23}
\]
Adding (4.24) and (4.25) using 4.23 results in (4.22) yields for $\dot{V}$,

$$\dot{V} \leq - (x(t) - x'(t))^T \mathcal{R}^T \mathcal{R}^T (F(x(t)) - F(x'(t)))$$

$$+ (x(t) - x'(t))^T \mathcal{R}^T \mathcal{R} B(u(t) - u'(t))$$

$$+ (x(t) - x'(t))^T S^T S (u(t) - u'(t))$$

When using the matrix identity $\mathcal{R}^T \mathcal{R} + S^T S = I$ from (4.7), the inequality becomes,

$$\dot{V} \leq -(x(t) - x'(t))^T \mathcal{R}^T (F(x(t)) - F(x'(t))) + (x(t) - x'(t))^T B(u(t) - u'(t))$$

Finally, using Assumption 4(i), the first term is less than 0, and $y(t) = B(u(t)$ it follows that

$$\dot{V} \leq (y(t) - y'(t))^T (u(t) - u'(t))$$

and $\tilde{\Sigma}$ is incrementally passive, hence EIP.
4.3 Control Input for Consensus of State Vector

In section 4.1, equation (4.16) and (4.20), is the reformulation of NE dynamics over $G_c$ as a multi-agent agreement problem. The dynamics did not define the control signal ($u_i$) as a function of the other players' action. In this section we define the control input that each agent uses to reach a consensus in their state vector while maintaining the NE point.

4.3.1 Gradient-Play Dynamics with Estimate Correction

Given agent dynamics $\tilde{\Sigma}_i$, (4.15), for each individual player we design $u_i \in \mathbb{R}^n$ based on the relative output feedback from its neighbours, such that the auxiliary state variables (estimates) agree one with another. Thus, all agents reach the consensus subspace, $x^i = x^j$, for all $i, j \in I$. Additionally, the dynamics need to converge to the NE $x^*$. For simplicity, we will assume from now on that $B_I = I_n$ so that $y = x$.

From section 2.6, the Laplacian matrix $L \in \mathbb{R}^{n \times n}$ was used to measure the difference between the components of an input vector $x \in \mathbb{R}^n$, i.e., $[Lx]_i = \sum_{j \in N_i}(x_i - x_j) \in \mathbb{R}$. When the graph is connected the Laplacian is positive semi-definite and $Lx = 0$ only when $x_1 = \cdots = x_n = \alpha$ for some $\alpha \in \mathbb{R}$. Notice that the Laplacian is comparing single elements in $x$ but we require consensus of a whole vector. Let $L = L \otimes I_n \in \mathbb{R}^{Nn \times Nn}$ denote the augmented Laplacian matrix. This matrix compares vectors and has the property that it is equal to zero on a connected graph only when the vectors are at consensus.

Example 4.4. Let $G(\mathcal{I}, J_i, \Omega_i)$ be a three player game where $n_i = 1$, $\forall i \in \mathcal{I}$. The communication graph is given in figure 4.1 and the corresponding Laplacian in figure 4.2. The augmented

$$L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

![Figure 4.1: Communication Graph](image1)

![Figure 4.2: Laplacian](image2)
Laplacian is $L \triangleq (L \otimes I_3)$ and output from vector $x \in \Omega^N$ is,

$$Lx = \begin{bmatrix}
2 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & 2 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & 2 & 0 & 0 & -1 & 0 & 0 & -1 \\
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
x_8 \\
x_9 \\
\end{bmatrix} = \begin{bmatrix}
(x_1 - x_2) + (x_1 - x_3) \\
(x_2 - x_2) + (x_1 - x_3) \\
(x_3 - x_3) + (x_1 - x_3) \\
(x_1 - x_1) \\
(x_2 - x_2) \\
(x_3 - x_1) \\
(x_4 - x_1) \\
(x_5 - x_3) \\
(x_6 - x_3) \\
\end{bmatrix}$$

From the last line in the example above it is obvious that $Lx = 0$ only when $x_1 = x_2 = x_3$. Therefore, the augmented Laplacian can be used to determine if the estimate vectors are at consensus. Some properties $L$ inherits from the Laplacian $L$ are

$$Null(L) = Range(I_N \otimes I_n) \quad Range(L) = Null(I_N^T \otimes I_n)$$

following from $L1_N = 0_N$. Additional, since $L$ is symmetric, $Range(L) = Null(L)^{\perp}$ and $Range(I_N \otimes I_n)^{\perp} = Null(I_N^T \otimes I_n)$. For any $W \in \mathbb{R}^{q \times n}$, and any $x \in \mathbb{R}^{Nn}$ by using the properties of the Kronecker product,

$$(I_N^T \otimes W)Lx = (I_N^T \otimes W)(L \otimes I_n)x = ((I_N^T L) \otimes (W I_n))x = 0_q$$

where $I_N^T L = 0$ because of the symmetry of the Laplacian. Just like the Laplacian, the matrix $L$ is also positive semi-definite.

With respect to the overall dynamics $\Sigma$, (4.16), the objective is to design $u$ such that $x$ reaches consensus, i.e., $1_N \otimes x$, for some $x \in \Omega$ and $x$ converges towards the NE $x^*$. The consensus condition is written as $Lx = 0$. If Assumption 4(i) holds then $\Sigma$, (4.16) is incrementally passive by Lemma 4.2 and since $L$ is positive semi-definite, a passivity-based control design, e.g. [26, 28], suggests taking $u = -Lx$. The resulting closed-loop system which represents the new overall system dynamics $\bar{P}$ is given in stacked notation as

$$\bar{P} : \dot{x} = -R^T F(x) - Lx$$

shown in Figure 4.3 as the feedback interconnection between $\Sigma$ and $L$.

The new individual player dynamics $\bar{P}_i$ from (4.15) with $u_i = -\sum_{j \in N_i} (x_i - x_j)$ as used in multi-agent agreement problems [17]. In the standard multi-agent agreement setting agents'
functions are assumed to satisfy an incremental passivity condition which doesn’t necessarily hold in the game setting. This is caused by the coupling in each players’ cost function.

\[ \tilde{P}_i : \dot{x}^i = -R_i^T \nabla_i J_i(x^i) - \sum_{j \in N_i} (x^i - x^j) \]  

or, separating the action \( x^i_j = x_i \) and estimate \( x^i_{-i} \) dynamics,

\[
\tilde{P}_i : \begin{cases}
\dot{x}_i &= -\nabla_i J_i(x_i, x^i_{-i}) - R_i \sum_{j \in N_i} (x^i - x^j) \\
\dot{x}^i_{-i} &= -S_i \sum_{j \in N_i} (x^i - x^j)
\end{cases}
\]  

(4.32)

For player \( i \), \( \tilde{P}_i \), (4.31) or (4.32) is clearly distributed over \( G_c \). The input is the relative difference between its estimate and its neighbours’. The overall stacked action / estimate representation is,

\[
\tilde{P} : \begin{cases}
\dot{x} &= -F(R^T x + S^T z) - RL[R^T x + S^T z] \\
\dot{z} &= -SL[R^T x + S^T z]
\end{cases}
\]  

(4.33)

where \( x = R x \) and \( z = S x \) as defined in (4.3) and (4.5) respectively. From equation (4.13), the term \( R^T x + S^T z \) is equal to \( x \).

In standard consensus terms, dynamics (4.32) show that agent \( i \) can share the estimate information to move in the direction of the average value of its neighbours, while the gradient term enforces movement towards minimizing their own cost. Compared to the gossip-based algorithm in [17], the action part of (4.32) has an extra correction term. The next result shows that the equilibrium of (4.30) or (4.32) occurs when the agents are at a consensus and at NE.

**Lemma 4.5.** Consider a game \( G(\mathcal{I}, J_i, \Omega_i) \) over a communication graph \( G_c \) under Assumption 1(i) and 2. Let the dynamics for each agent \( \tilde{P}_i \) be as in (4.31), (4.32), or overall \( \tilde{P} \), (4.30). At an equilibrium point \( \bar{x} \in \Omega^N \) the estimate vectors of all players are equal \( \bar{x}^i = \bar{x}^j, \forall i, j \in \mathcal{I} \) and equal to the Nash equilibrium profile \( x^* \), hence the action components of all players coincide with the optimal actions, \( \bar{x}^i_i = x^*_i, \forall i \in \mathcal{I} \).
Proof. Let $\bar{x}$ denote an equilibrium of (4.30),

$$0_{Nn} = -\mathcal{R}^T \mathbf{F}(\bar{x}) - L\bar{x} \quad (4.34)$$

Pre-multiplying both sides by $(1_T^T \otimes I_n)$, yields $0_n = -(1_T^T \otimes I_n)\mathcal{R}^T \mathbf{F}(\bar{x})$, where $(1_T^T \otimes I_n)L\bar{x} = 0$ was obtained by using (4.28) and the symmetry of the Laplacian. Using (4.12) we can simplify $(1_T^T \otimes I_n)\mathcal{R}^T = I_n$ which gives

$$0_n = F(\bar{x}), \quad \text{or} \quad \nabla_i J_i(\bar{x}) = 0, \quad \forall i \in \mathcal{I} \quad (4.35)$$

by (4.17). Substituting (4.35) into (4.34) results in $0_{Nn} = -L\bar{x}$. From this it follows that $\bar{x}^i = \bar{x}^j, \forall i, j \in \mathcal{I}$ by Assumption 2 and (4.28). Therefore $\bar{x} = 1_N \otimes \bar{x}$, for some $\bar{x} \in \Omega$. Substituting this back into (4.35) yields $0_n = F(1_N \otimes \bar{x})$ or $\nabla_i J_i(\bar{x}) = 0, \forall i \in \mathcal{I}$. Notice that $\nabla_i J_i(\bar{x}_i, \bar{x}_{-i}) = 0$, for all $i \in \mathcal{I}$ if and only if $0_n = F(\bar{x})$. Therefore, by the Nash equilibrium characterization (3.1), $\bar{x} = x^*$, hence $\bar{x} = 1_N \otimes x^*$ and for all $i, j \in \mathcal{I}$, and $\bar{x}^i = \bar{x}^j = x^*$ the NE of the game.

\[\square\]

### 4.3.2 Projected Gradient with Estimate Correction

A similar argument from the previous section applies for projection dynamics in the case when the action set is compact. If Assumption 4(i) holds then $\Sigma$, (4.20) is incrementally passive by Lemma 4.3, and $L$ is positive semi-definite. Therefore, just as in Section 4.3.1 we consider $u(t) = -Lx(t)$. The resulting closed-loop system, represents the new overall system dynamics $\tilde{P}$ and is given in stacked notation as

$$\tilde{P}: \dot{x} = \mathcal{R}^T \Pi_\Omega (Rx(t), -\mathbf{F}(x(t)) - \mathcal{R}Lx(t)) - S^T SLx(t) \quad (4.36)$$

Alternatively, using $x = R\bar{x}$ and $z = S\bar{x}$, the dynamics can be split into action and estimate components as in (4.33),

$$\tilde{P}: \begin{cases} \dot{x} = \Pi_\Omega (x, -\mathbf{F}(\mathcal{R}^T x + S^T z) - \mathcal{R}L[\mathcal{R}^T x + S^T z]) \\ \dot{z} = -SL[\mathcal{R}^T x + S^T z] \end{cases} \quad (4.37)$$

Existence of a unique solution of (4.36) or (4.37) is guaranteed under Assumption 4(i) or (ii) by Theorem 1 in [51]. From (4.36) or (4.37) after separating the action $x_i^o = x_i$ and estimate $x_i^e$ dynamics, the new projected player dynamics $\tilde{P}_i$ are,

$$\tilde{P}_i: \begin{cases} \dot{x}_i = \Pi_{\mathcal{N}_i} \left(x_i - \nabla_i J_i(x^i) - \mathcal{R}_i \sum_{j \in \mathcal{N}_i} (x^i - x_j)\right) \\ \dot{x}_i^e = -S_i(\sum_{j \in \mathcal{N}_i} x^i - x_j) \end{cases} \quad (4.38)$$
Compared to $\bar{x}$, $\tilde{P}$, in (4.32) has projected action components. The next result shows that the equilibrium of (4.36) or (4.38) occurs when the agents are at consensus and at the NE.

**Lemma 4.6.** Consider a game $\mathcal{G}(\mathcal{I}, J_i, \Omega_i)$ over a communication graph $G_c$ under Assumptions 1(ii), 2 and 4(i) or (ii). Let the dynamics for each agent $\tilde{P}_i$ be as in (4.38), or overall $\tilde{P}$, (4.36). At an equilibrium point $\bar{x}$ the estimate vectors of all players are equal $\tilde{x}_i^j = \tilde{x}_i^j$, $\forall i, j \in \mathcal{I}$ and equal to the Nash equilibrium profile $x^*$, hence the action components of all players coincide with the optimal actions, $\tilde{x}_i^j = x^*_i$, $\forall i \in \mathcal{I}$.

**Proof.** Let $\bar{x}$ denote an equilibrium of (4.36),

$$0_{Nn} = R^T \Pi_\Omega (R\bar{x}, -F(\bar{x}) - RL\bar{x}) - SL\bar{x}$$

(4.39)

Pre-multiplying both sides by $R$ and using (4.7) simplifies to,

$$0_n = \Pi_\Omega (R\bar{x}, -F(\bar{x}) - RL\bar{x})$$

(4.40)

Substituting (4.40) into (4.39) results in $0_{Nn} = -SL\bar{x}$ which implies $L\bar{x} \in \text{Null}(S^T S)$. From Lemma 2.1 the null space of $S^T S$ is the same as $\text{Null}(S)$. Therefore, $\bar{x} \in \text{Null}(S^T SL)$ if and only if $L\bar{x} \in \text{Null}(S)$, which is equivalent to $L\bar{x} \in \text{Range}(R^T)$ following from (4.8). Thus, $\bar{x} \in \text{Null}(SL)$ if and only if there exists $q \in \mathbb{R}^n$ such that $L\bar{x} = R^T q$. Then for $q \in \mathbb{R}^n$ and for all $w \in \mathbb{R}^n$, $(1_N^T \otimes w^T)L\bar{x} = (1_N^T \otimes w^T)R^T q$. Using $L(1_N^T \otimes w^T) = 0$ from (4.29) and $(1_N^T \otimes w^T)R^T = w^T$ by (4.12), this means $0 = w^T q$, for all $w \in \mathbb{R}^n$. Therefore, $q = 0$ and $L\bar{x} = 0$. By equation (4.28), $\bar{x} = 1_N \otimes x$, for some $x \in \Omega$.

From this it follows that $\tilde{x}_i^j = \tilde{x}_i^j$, $\forall i, j \in \mathcal{I}$ by Assumption 2 and (4.28). Therefore, $\bar{x} = 1_N \otimes \bar{x}$, for some $\bar{x} \in \Omega$. Substituting this back into (4.40) yields $0_n = \Pi_\Omega (R(1_N \otimes \bar{x}), -F(1_N \otimes \bar{x}))$ or $0_n = \Pi_\Omega (\bar{x}, -F(\bar{x}))$ by using (4.17). Therefore, as in (3.11), it follows that $-F(\bar{x}) \in N_\Omega(\bar{x})$ hence, by (3.4), $\bar{x} = x^*$ the NE. Thus, $\tilde{x} = 1_N \otimes x^*$ and for all $i, j \in \mathcal{I}$, and $\tilde{x}_i^j = \tilde{x}_i^j = x^*$ the NE of the game.

In this Chapter we stated the main objective of this thesis. The goal is to find gradient-based dynamics such that a game under partial information over a connected undirected graph converges to the Nash Equilibrium. To tackle this problem each agent is given a state variable which contains an estimate of the other agents actions. Agents communicate to neighbouring players to update their estimates and reach a consensus in their state variable, i.e., $x_1^1 = \cdots = x^N$. The control signal $u = -Lx$ encapsulates the communication between agents. With this control signal the overall system equilibria points coincide with the NE points of the game. In the next chapter, dynamics (4.30) and (4.36) are proved to converge if the communication graph $G_c$ is sufficiently connected.
Chapter 5

Convergence Analysis

In the previous chapter we reformulated the Nash equilibrium problem as a multi-agent agreement problem. Using the input control signal $u = -Lx$ the dynamical system (4.30)/(4.36) have their equilibria at the Nash equilibria of the game. Although the equilibria points of the dynamical coincide with the NE points there is no guarantee that the NE point is (asymptotically) stable. In this chapter we analyze the convergence of the players’ new dynamics $\tilde{P}_i$ (4.31) or overall dynamics $\tilde{P}$ (4.30) for unconstrained action sets, and player dynamics (4.38) or overall dynamics (4.36) for compact action sets.

In Section 5.1 we analyze convergence of (4.30) and (4.36) on a single timescale. Theorem 5.1 proves convergence of (4.30) under Assumptions 1(i), 2, 3(i), and 4(i), while Theorem 5.2 proves convergence of (4.36) under Assumptions 1(ii), 2, 3(i), and 4(i). We exploit the incremental passivity (EIP) property of $\tilde{\Sigma}$, (Lemma 4.2 or 4.3), and diffusive properties of the Laplacian.

In Section 5.2, Theorem 5.3 proves convergence under Assumptions 1(i), 2, 3(ii) and 4(ii) for compact action sets. Similarly, for the compact action set, Theorem 5.4 shows convergence under Assumptions 1(ii), 2, 3(ii) and 4(ii). The monotonicity assumption on the pseudo-gradient and the monotonicity assumption on the extended pseudo-gradient in Theorem 5.1/5.2 is replaced with strong monotonicity of the pseudo-gradient and a Lipschitz condition on the extended pseudo-gradient. These assumptions are a relaxation of the assumptions used in Section 5.1. The dynamics are shown to converge when an inequality relating the graph connectivity, monotonicity constant $\mu$ and the Lipschitz constant of the extended pseudo-gradient, is satisfied.

In Section 5.3, we modify the estimate component of dynamics (4.32) to be much faster, and in Theorem 5.13 prove convergence under Assumptions 1(i), 2, 3(ii) and 4(ii), using a two-timescale singular perturbation approach. By allowing the estimate component dynamics to be faster, the graph connectivity can be relaxed. Therefore, there is a trade-off between graph connectivity and the speed of the estimate dynamics to ensure convergence. The projection dynamics are not analyzed on two-timescales and the shortcoming of this approach is highlighted. Additionally, a minor remark about just scaling the Laplacian to improve convergence is mentioned.
5.1 Monotone Extended Pseudo-Gradient

5.1.1 Gradient-Play Dynamics with Estimate Correction

Theorem 5.1 shows that, under Assumption 4(i), that dynamical system (4.32),

$$\dot{x}_i = \begin{bmatrix} \dot{x}_i \\ x_i^* \end{bmatrix} = \begin{bmatrix} -\nabla_i J_i(x_i, x_i^* - \mathcal{R}_i \sum_{j \in \mathcal{N}_i} (x_i^* - x_j) \\ -\mathcal{S}_i \sum_{j \in \mathcal{N}_i} (x_i^* - x_j) \end{bmatrix}$$

converges to the NE for games with unconstrained action sets, over any connected $G_c$.

**Theorem 5.1.** Consider a game $\mathcal{G}(\mathcal{I}, J_i, \Omega_i)$ over a communication graph $G_c$ under Assumptions 1(i), 2, 3(i) and 4(i). Let each player’s dynamics $\dot{\mathcal{P}}_i$, be as in (4.31), (4.32)/(5.1), or overall $\mathcal{P}$, (4.30), as in Figure 4.3. Then, any solution of (4.30) is bounded and asymptotically converges to $1_N \otimes x^*$, and the actions components converge to the NE of the game, $x^*$.

**Proof.** By Lemma 4.5, the equilibrium of $\mathcal{P}$ (4.30) is $\bar{x} = 1_N \otimes x^*$. We consider the quadratic storage function $V(x) = \frac{1}{2}||x - \bar{x}||^2$, $\bar{x} = 1_N \otimes x^*$ as a Lyapunov function. Following the steps in the proof of Lemma 4.2 and substituting $u = -Lx$ and $x' = \bar{x}$, we obtain that along the solutions of (4.30),

$$\dot{V} = -(x - \bar{x})^T \mathcal{R}^T (F(x) - F(\bar{x})) - (x - \bar{x})^T L (x - \bar{x})$$

By Assumption 4(i) and from the property of the augmented Laplacian $L$ being positive semi-definite it follows that $\dot{V} \leq 0$, for all $x \in \Omega^N$, hence all trajectories of (4.30) are bounded and $\bar{x}$ is stable. To show convergence we use LaSalle’s invariance principle, i.e., Theorem 2.10.

From (5.2), $\dot{V} = 0$ when both terms in (5.2) are zero, i.e., $(x - \bar{x})^T \mathcal{R}^T (F(x) - F(\bar{x})) = 0$ and $(x - \bar{x})^T L (x - \bar{x}) = 0$. By Assumption 2 and the properties of the Laplacian (4.28), $(x - \bar{x})^T L (x - \bar{x}) = 0$ is equivalent to $x - \bar{x} = 1_N \otimes x$, for some $x \in \mathbb{R}^n$. Since the equilibrium $\bar{x} = 1_N \otimes x^*$, this implies that $x = 1_N \otimes x$, for some $x \in \mathbb{R}^n$. Using the definition of the extended pseudo-gradient (4.17), $F(1_N \otimes x) = F(x)$, the first term in (5.2) is zero when,

$$0 = -(x - \bar{x})^T \mathcal{R}^T (F(x) - F(\bar{x}))$$

$$= -((1_N \otimes x - 1_N \otimes x^*)^T \mathcal{R}^T (F(1_N \otimes x) - F(1_N \otimes x^*)))$$

$$= -((1_N \otimes x - 1_N \otimes x^*)^T \mathcal{R}^T [F(x) - F(x^*)])$$

$$= -(x - x^*)^T [F(x) - F(x^*)] < 0 \quad \forall x \neq x^*$$

(5.3)

where the last line follows from equation (4.12) and the strict inequality follows from Assumption 3(i). Therefore $\dot{V} = 0$ in (5.2) only if $x = x^*$ and hence $x = 1_N \otimes x^*$. Since $V$ is radially unbounded, the conclusion follows from Theorem 2.10 or Theorem 2.8 that the equilibrium $1_N \otimes x^*$ is asymptotically stable. \qed
5.1.2 Projected Gradient with Estimate Correction

The following results show convergence of \[ \text{4.36} \]

\[
\tilde{P} : \dot{x} = R^T \Pi_{\Omega} (R x(t), -F(x(t)) - RLx(t)) - S^T S L x(t) \quad (5.4)
\]

to the NE for games with compact action sets over any connected \( G_c \), under Assumption \[ \text{4(i)} \] when the action space is compact.

**Theorem 5.2.** Consider a game \( G(I, J, \Omega_i) \) over a communication graph \( G_c \) under Assumptions \[ \text{4(ii), 2(iii) and 2(i)} \]. Let each player’s dynamics \( \tilde{P}_i \) be as in \[ \text{4.38} \], and overall \( \tilde{P} \), \[ \text{4.36} \] or \[ \text{4.37} \]. Then, for any \( x(0) \in \Omega \) and any \( z(0) \), the solution of \[ \text{4.37} \] asymptotically converges to \( 1_N \otimes x^* \), and the actions components converge to the NE of the game, \( x^* \).

**Proof.** The proof is similar to the proof in Theorem \[ \text{5.1} \] except that the argument is based on Barbalat’s Lemma, (Lemma \[ 2.12 \]), since the system is time-varying. Let \( V(t, x) = \frac{1}{2} \| x(t) - \bar{x} \|^2 \), where \( \bar{x} = 1_N \otimes x^* \) by Lemma \[ 4.6 \]. Using equation \[ 4.27 \] in Lemma \[ 4.3 \] with \( x'(t) = \bar{x} \), \( u(t) = -Lx(t) \), and \( u'(t) = -L\bar{x} \), it follows that for any \( x(0) \in \Omega \) and any \( z(0) \), along \[ 4.36 \],

\[
\dot{V} \leq - (x(t) - \bar{x})^T R^T (F(x(t)) - F(\bar{x})) - (x(t) - \bar{x})^T L(x(t) - \bar{x}) \quad (5.5)
\]

Under Assumption \[ 4(i) \] and from the positive semi-definite Laplacian \( L \), \( \dot{V} \leq 0 \), for all \( R x(t) \in \Omega \). Thus, \( V(t, x(t)) \) is non-increasing and bounded from below by 0, hence it converges as \( t \to \infty \) to some \( \bar{V} \geq 0 \). Then, under Assumption \[ 4(i) \], it follows that \( \lim_{t \to \infty} \int_0^t (x(\tau) - \bar{x})^T L(x(\tau) - \bar{x}) d\tau \) exists and is finite. Since \( x(t) \) is absolutely continuous, hence uniformly continuous, from Barbalat’s Lemma it follows that \( L(x(t) - \bar{x}) \to 0 \) as \( t \to \infty \). Since \( \bar{x} = 1_N \otimes x^* \), this means that \( x(t) \to 1_N \otimes x \), as \( t \to \infty \) for some \( x \in \Omega \). Then, \( V(t, x(t)) = \frac{1}{2} \| x(t) - \bar{x} \|^2 \to \frac{1}{2} \| 1_N \otimes (x - x^*) \|^2 = \bar{V} \) as \( t \to \infty \). If \( \bar{V} = 0 \) the proof if completed.

Using the strict monotonicity assumption \[ 3(i) \], it can be shown by a contradiction argument that \( x = x^* \) and \( \bar{V} = 0 \). Assume that \( x \neq x^* \) and \( \bar{V} > 0 \). Then from \[ 5.5 \] there exists a sequence \( \{ t_k \} \), where \( t_k \to \infty \), as \( k \to \infty \), such that \( \dot{V}(t_k) \to 0 \) as \( k \to \infty \). Suppose this claim is false. Then there exists a \( d > 0 \) and a \( T > 0 \) such that \( \dot{V}(t) \leq -d \), for all \( t > T \), which contradicts \( \bar{V} > 0 \), hence the claim is true. Substituting \( \{ t_k \} \) into \[ 5.5 \] yields

\[
\dot{V}(t_k) \leq - (x(t_k) - \bar{x})^T R^T (F(x(t_k)) - F(\bar{x})) - (x(t_k) - \bar{x})^T L(x(t_k) - \bar{x})
\]

where the left-hand side converges to 0 as \( k \to \infty \). Hence,

\[
0 \leq - \lim_{k \to \infty} (x(t_k) - \bar{x})^T R^T (F(x(t_k)) - F(\bar{x})) - (x(t_k) - \bar{x})^T L(x(t_k) - \bar{x})
\]

Using \( \lim_{k \to \infty} x(t_k) = 1_N \otimes x \in \text{Null}(L) \), this leads to

\[
0 \leq - (1_N \otimes (x - x^*))^T R^T (F(1_N \otimes x) - F(1_N \otimes x^*))
\]
or, \(0 \leq -(x - x^*)^T(F(x) - F(x^*)) < 0\), by the strict monotonicity Assumption 3(i), since we assumed \(x \neq x^*\). This is a contradiction, hence \(x = x^*\) and \(V = 0\). Therefore, \(1_N \otimes x^*\) is asymptotically stable.

### 5.2 Strongly Monotone Pseudo-Gradient

In this section we replace the monotonicity assumption on the pseudo-gradient and monotonicity assumption on the extended pseudo-gradient with a strong monotonicity condition on the pseudo-gradient and a Lipschitz condition on the extended pseudo-gradient. Additionally, an alternative method is shown that provides a different bound, but is dependent on the number of players.

#### 5.2.1 Gradient-Play Dynamics with Estimate Correction

Next we show that, under a weaker Lipschitz property of \(F\) (Assumption 4(ii)) and strong monotonicity of \(F\) (Assumption 3(ii)), (4.32) converges over any sufficiently connected \(G_c\).

**Theorem 5.3.** Consider a game \(\mathcal{G}(I, J_i, \Omega_i)\) over a communication graph \(G_c\) under Assumptions 1(i), 2, 3(ii) and 4(ii). Let each player’s dynamics \(\tilde{P}_i\) be as in (4.31), (4.32) or overall \(\tilde{P}\) (4.30). Then, if \(\lambda_2(L) > \frac{\theta_2^2}{n} + \theta\), any solution of (4.30) converges asymptotically to \(1_N \otimes x^*\), and the actions components converge to the NE of the game, \(x^*\). Furthermore, the convergence is exponential.

**Proof.** We decompose \(\mathbb{R}^{Nn}\) into the consensus subspace \(C_N^n = \{1_N \otimes x \mid x \in \mathbb{R}^n\}\) and its orthogonal complement \(E_N^n\), such that \(\mathbb{R}^{Nn} = C_N^n \oplus E_N^n\). Let the two projection matrices be defined as,

\[
P_C = \frac{1}{N} 1_N \otimes 1_N^T \otimes I_n \quad P_E = I_{Nn} - \frac{1}{N} 1_N \otimes 1_N^T \otimes I_n \quad I_{Nn} = P_C + P_E
\]

where \(P_C\) is the projection onto the consensus subspace and \(P_E\) is the orthogonal complement. Then any \(x \in \mathbb{R}^{Nn}\) can be decomposed as \(x = I_{Nn}x = (P_C + P_E)x = x^\parallel + x^\perp\), where \(x^\parallel = P_Cx \in C_N^n\) and \(x^\perp = P_Ex \in E_N^n\), with \((x^\parallel)^T x^\perp = 0\). Thus \(x^\parallel = 1_N \otimes x\), for some \(x \in \mathbb{R}^n\) and \(Lx^\parallel = 0\) because \(x^\parallel\) is in the consensus subspace which is the Laplacian’s null space. Additionally, \(\min_{x^\perp \in E_N^n} (x^\perp)^T L x^\perp = \lambda_2(L)\|x^\perp\|^2\).

Consider the Lyapunov function \(V(x) = \frac{1}{2}\|x - \overline{x}\|^2\), where \(\overline{x} = 1_N \otimes x^*\). Using the decomposition, \(x = x^\parallel + x^\perp\), the Lyapunov function can be written as,

\[
V(x) = \frac{1}{2} \|x - \overline{x}\|^2
= \frac{1}{2} \|(x^\parallel + x^\perp) - \overline{x}\|^2
= \frac{1}{2} \|x^\perp + (x^\parallel - \overline{x})\|^2
= \frac{1}{2} \left(\|x^\perp\|^2 + 2(x^\perp)^T (x^\parallel - \overline{x}) + \|x^\parallel - \overline{x}\|^2\right)
\]
where \( \overline{x} \) can’t be decomposed because it is in the consensus subspace and the middle term in the second last line is removed because \( (x^\parallel)^T \overline{x} = 0 \).

The Lyapunov function \( V(x) \) is the same as in Lemma 4.2. Therefore, following the same steps as in Lemma 4.2 and replacing \( u = -Lx \) and \( u' = -L\overline{x} \) we get the relation as in (5.2), i.e.,

\[
\dot{V} \leq -(x - \overline{x})^T \mathcal{R}^T [F(x) - F(\overline{x})] - (x - \overline{x})^T L(x - \overline{x})
\]

Replacing \( x \) with its decomposed components \( x^\perp, x^\parallel \), \( \dot{V} \) can be written as,

\[
\dot{V} \leq -\left[ x^\perp + (x^\parallel - \overline{x}) \right]^T \mathcal{R}^T [F(x) - F(\overline{x})] - \left[ x^\perp + (x^\parallel - \overline{x}) \right] L \left[ x^\perp + (x^\parallel - \overline{x}) \right]
\]

\[
\leq -\left[ x^\perp \right]^T \mathcal{R}^T [F(x) - F(\overline{x})] - \left[ x^\parallel - \overline{x} \right]^T \mathcal{R}^T [F(x) - F(\overline{x})] - \left[ x^\perp \right] L \left[ x^\perp \right]
\]

(5.6)

where \( Lx^\parallel = 0 \) and \( L\overline{x} = 0 \) was used on the last line.

A decomposition of the extended pseudo-gradient can be represented as the difference between the consensus and the orthogonal complement component,

\[
F(x) - F(\overline{x}) = F(x^\perp + x^\parallel) - F(\overline{x})
\]

\[
= \underbrace{F(x^\perp + x^\parallel) - F(x^\parallel)}_{\text{first term}} + \underbrace{F(x^\parallel) - F(\overline{x})}_{\text{second term}}
\]

where the term \(-F(x^\parallel) + F(\overline{x}) = 0\) was added to \( F(x) - F(\overline{x}) \). Looking at the first term we see that it is measuring the change in \( F \) for a change in \( x^\perp \) and the second term measures the change in \( F \) from the consensus component of the current action to the equilibrium. Applying this decomposition to (5.6), \( \dot{V} \) becomes,

\[
\dot{V} \leq -\left[ x^\perp \right]^T \mathcal{R}^T \left[ F(x^\perp + x^\parallel) - F(x^\parallel) + F(x^\parallel) - F(\overline{x}) \right] - \left[ x^\perp \right] L \left[ x^\perp \right]
\]

\[
- \left[ x^\parallel - \overline{x} \right]^T \mathcal{R}^T \left[ F(x^\perp + x^\parallel) - F(x^\parallel) + F(x^\parallel) - F(\overline{x}) \right]
\]

\[
\leq -\left[ x^\perp \right]^T \mathcal{R}^T \left[ F(x^\perp + x^\parallel) - F(x^\parallel) \right] - \left[ x^\perp \right] L \left[ x^\perp \right]
\]

\[
- \left[ x^\parallel - \overline{x} \right]^T \mathcal{R}^T \left[ F(x^\perp + x^\parallel) - F(x^\parallel) \right] - \left[ x^\parallel - \overline{x} \right]^T \mathcal{R}^T \left[ F(x^\parallel) - F(\overline{x}) \right]
\]

Using \( (x^\perp)^T Lx^\perp \geq \lambda_2(L) \| x^\perp \|^2 \), \( F(x^\parallel) = F(1_N \otimes x) = F(x) \), and \( F(\overline{x}) = F(1_N \otimes x^*) = F(x^*) \) yields,

\[
\dot{V} \leq -\left[ x^\perp \right]^T \mathcal{R}^T \left[ F(x^\perp + x^\parallel) - F(x^\parallel) \right] - \left[ x^\perp \right] L \left[ x^\perp \right] - \left[ x^\parallel - \overline{x} \right]^T \mathcal{R}^T \left[ F(x) - F(x^*) \right] - \lambda_2(L) \| x^\perp \|^2
\]
The Lipschitz condition in Assumption 3(ii) bounds \(\|\theta\|\) norm of \(R\) Chapter 5. Convergence Analysis

60

Rearranging terms and defining \(w\).

Substituting these bounds into the relation above,

\[
\dot{V} \leq -\theta \|x^\perp\|^2 - [x^\perp]^T \mathcal{R}^T [F(x) - F(x^*)] - \lambda_2(L) \|x^\perp\|^2
\]

Under Assumption 4(ii), the Lipschitz condition bounds \(\|F(x^\perp + x^\parallel) - F(x^\parallel)\| \leq \theta \|x^\parallel\|\). The norm of \(R\) can also be bounded because \(\|\mathcal{R}\| = 1\) and therefore, \(\|\mathcal{R}x^\perp\| \leq \|\mathcal{R}\| \|x^\perp\| = \|x^\perp\|\). Applying these bounds,

\[
\dot{V} \leq \theta \|x^\perp\|^2 - \theta \|x^\perp\| \|x - x^*\| - \lambda_2(L) \|x^\perp\|^2
\]

The Lipschitz condition in Assumption 3(ii) bounds \(\|F(x) - F(x^*\| \leq \theta \|x - x^*\| \leq \theta \|x - x^*\|^2\) and the monotonicity condition in Assumption 3(ii) bounds \(\langle x - x^*\| [F(x) - F(x^*)] \geq \mu \|x - x^*\|^2\).

Substituting these bounds into the relation above,

\[
\dot{V} \leq \theta \|x^\perp\|^2 + \theta \|x^\perp\| \|x - x^*\| - \lambda_2(L) \|x^\perp\|^2
\]

Rearranging terms and defining \(w^T = \begin{bmatrix} \|x - x^*\| & \|x^\perp\| \end{bmatrix}\) then,

\[
\dot{V} \leq -w^T \Theta w = -w^T \begin{bmatrix} \mu & -\theta \\ -\theta & \lambda_2(L) - \theta \end{bmatrix} w
\]

If the matrix \(\Theta\) is positive definite then \(\dot{V} \leq 0\). The matrix is positive definite when \(\mu(\lambda_2(L) - \theta) > \theta^2\) \(\Rightarrow \lambda_2(L) > \frac{\theta^2}{\mu} + \theta\). Hence \(\dot{V} \leq 0\) and \(\dot{V} = 0\) only if \(x^\perp = 0\) and \(x = x^*\), hence \(x = 1_N \otimes x^*\). Asymptotic stability now follows from the radially unbounded Lyapunov function by Lyapunov’s stability Theorem refthm:lyapDirect in Chapter 2.

Exponential convergence follows from substituting \(\|x - x^*\| = \frac{1}{\sqrt{N}} \|x^\parallel - \bar{x}\|\), into \(w\) above,

\[
\dot{V} \leq -w^T \Theta w
\]

\[
= \begin{bmatrix} \|x - x^*\| & \|x^\perp\| \end{bmatrix} \begin{bmatrix} \mu & -\theta \\ -\theta & \lambda_2(L) - \theta \end{bmatrix} \begin{bmatrix} \|x - x^*\| \\ \|x^\perp\| \end{bmatrix}
\]

\[
= \frac{1}{\sqrt{N}} \|x^\parallel - \bar{x}\| \|x^\perp\| \begin{bmatrix} \mu & -\theta \\ -\theta & \lambda_2(L) - \theta \end{bmatrix} \frac{1}{\sqrt{N}} \|x^\parallel - \bar{x}\| \|x^\perp\|
\]
\[ v^T \Theta_N v = \left[ \| \mathbf{x} \| - \mathbf{x} \right] \left[ \begin{array}{cc} \frac{1}{\sqrt{N}} \mu & -\frac{1}{\sqrt{N}} \theta \\ \frac{1}{\sqrt{N}} \lambda_2(L) - \theta \end{array} \right] \left[ \| \mathbf{x} \| - \mathbf{x} \right] \]

where \( v^T \Delta = \left[ \| \mathbf{x} \| - \mathbf{x} \right] \left[ \begin{array}{cc} \frac{1}{\sqrt{N}} \mu & -\frac{1}{\sqrt{N}} \theta \\ \frac{1}{\sqrt{N}} \lambda_2(L) - \theta \end{array} \right] \left[ \| \mathbf{x} \| - \mathbf{x} \right] \) and \( V(x) = \frac{1}{2} \| v \|^2 \). For \( \Theta_N \) to be positive definite \( \mu(\lambda_2(L) - \theta) > \theta^2 \) \( \Rightarrow \) \( \mu(\lambda_2(L) - \theta) > \theta^2 \). This is the same condition to prove asymptotic stability. This implies that \( \dot{V}(x(t)) \leq -\eta V(x(t)) \), for some \( \eta > 0 \). Therefore, using Theorem 2.9, \( x(t) \) converges exponentially to \( x^* \) for the same condition on \( \lambda_2 \).

**Remark 5.1.** Since \( \theta, \mu \) are related to the coupling in the players’ cost functions and \( \lambda_2(L) \) to the connectivity of the graph, Theorem 5.3 highlights the tradeoff between properties of the game and those of the communication graph \( G_c \). This suggests that the Laplacian contribution can be used to balance the other terms in \( \dot{V} \). Alternatively, \( L \) on feedback path in Figure 4.3 has excess passivity which compensates the lack of passivity in the \( F \).

**Remark 5.2.** We note that we can relax the monotonicity assumption to hold just at the NE \( x^* \), recovering a strict-diagonal assumption used in [3]. However, since \( x^* \) is unknown, such an assumption cannot be checked a-priori except for special cases such as quadratic games. Local results follow if assumptions for \( F(\cdot) \) hold only locally around \( x^* \), and for \( F(\cdot) \) only locally around \( x^* = 1_N \otimes x^* \). We note that the class of quadratic games satisfies Assumption 3(ii) globally.

### 5.2.2 Projected Gradient with Estimate Correction

The corresponding Theorem for a compact action set is shown next.

**Theorem 5.4.** Consider a game \( G(I, J, \Omega_i) \) over a communication graph \( G_c \) under Assumptions 1(ii), 2, 3(ii) and 4(ii). Let each player’s dynamics \( \tilde{P}_i \) be as in (4.38) or overall \( \tilde{P} \) (4.36).

Then, if \( \lambda_2(L) > \frac{\theta^2}{\mu} + \theta \), for any \( x(0) \in \Omega \) and any \( z(0) \), the solution of (4.36) asymptotically converges to \( 1_N \otimes x^* \), and the actions converge to the NE of the game, \( x^* \). Furthermore, the convergence to the NE is exponential.

**Proof.** The proof is similar to the proof of Theorem 5.3. Based on Lemma 4.6, using \( V(t, x, \mathbf{x}) = \frac{1}{2} \| x(t) - \mathbf{x} \|^2 \) and equation (4.27) in Lemma 4.3, for \( u(t) = -Lx(t), u'(t) = -L\mathbf{x}(t) \) one can obtain (5.5) along (4.36), for any \( x(0) \in \Omega \) and any \( z(0) \). Then further decomposing \( x(t) \) into \( x^\perp(t) \) and \( x^{\|}(t) \) components as in the proof of Theorem 5.3 leads to inequality (5.7), where \( \Theta \) is positive under the conditions in the theorem. Then invoking Barbalat’s Lemma as in the proof of Theorem 5.2 shows that \( x(t) \to x^* \) and \( x^\perp(t) \to 0 \) as \( t \to \infty \). 

### 5.2.3 Gradient-Play Dynamics with Estimate Correction, Alternative Proof

In this section an alternative convergence proof for the unconstrained action space is provided. The bounds are similar and may provide a better bound in some situations. If \( N - 1 < \frac{4}{\theta} \), then
the following theorem will provide a better bound than Theorem 5.3.

**Theorem 5.5.** Consider a game $G(I, J, \Omega)$ over a communication graph $G_c$ under Assumptions 1(i), 2(ii) and 3(ii). Let each player’s dynamics $\tilde{P}_i$ be as in (4.31), (4.32) or overall $\tilde{P}$ (4.30). Then, if $\lambda_2(L) > \frac{N}{4}\theta^2$ and $\lambda_2(SLST) \geq \frac{\lambda_2(L)}{N}$, any solution of (4.30) converges asymptotically to $1_N \otimes x^*$, and the actions components converge to the NE of the game, $x^*$.

**Proof.** Let the dynamics of (4.30) be decomposed into action and estimate components as in equation (4.33), i.e.,

\[
\tilde{P} : \begin{cases} 
\dot{x} = -F(R^T x + S^T z) - RL[R^T x + S^T z] \\
\dot{z} = -SL[R^T x + S^T z] 
\end{cases} \tag{5.8}
\]

Let $v \triangleq z - S(1_N \otimes x)$ and make a change of coordinates so that the system is in terms of $x$ and $v$. First note that $x = R^T x + S^T z$ can be written as,

\[
R^T x + S^T z = \left(\begin{array}{c} R^T(1_N \otimes x) + S^T(v + S(1_N \otimes x)) \\
R^T R + S^T S \end{array}\right)
\]

\[
= (R^T R + S^T S)(1_N \otimes x) + S^T v
\]

\[
= 1_N \otimes x + S^T v
\]

where equation (4.12), $x = R(1_N \otimes x)$, and equation (4.7), $R^T R + S^T S = I_{Nn}$, were used. Therefore, the system dynamics are,

\[
\tilde{P} : \begin{cases} 
\dot{x} = -F(1_N \otimes x + S^T v) - RL[1_N \otimes x + S^T v] \\
\dot{v} = -SL[1_N \otimes x + S^T v] - S(1_N \otimes \dot{x}) 
\end{cases} \tag{5.9}
\]

The term $L(1_N \otimes x) = 0$ so the system can be further simplified to,

\[
\tilde{P} : \begin{cases} 
\dot{x} = -F(1_N \otimes x + S^T v) - RLST v \\
\dot{v} = -LSST v - S(1_N \otimes \dot{x}) 
\end{cases} \tag{5.10}
\]

The following Lemma is used to characterize the $SLST$ matrix and help determine what the equilibrium point of $\dot{v}$ is.

**Lemma 5.6.** If a communication graph $G_c$, satisfies Assumption 3 then $SLST$ is positive definite.

**Proof.** The matrix $SLST v = 0$ only if $S^T v \in \text{Null}(SL)$. Recall that $\text{Null}(SL) = \text{Null}(L) = (1_N \otimes w)$, for any $w \in \Omega$. Note that to be in $\text{Null}(L)$, $y = [y^1 \ldots y^N]^T = S^T v$ has to have $y^i = y^j$, $\forall i, j \in I$, but $y$ has a 0 in component $y^i$ $\forall i \in I$, due to the definition of $S$. Therefore $y^i \neq y^j$ unless $y^i = y^j = 0$, for all $j$. Therefore $y = S^T v$ has to be equal to 0 to be
in the $\text{Null}(L)$. Since $\text{Null}(ST) = \{0\}$ this implies $v = 0$, hence $\text{Null}(LS^T) = 0$ and $SLS^T$ is positive definite.

From Lemma 5.6 we see that the first term in $\dot{v}$ is negative definite and the equilibrium point for $\dot{v} = 0$ if $v = 0$ and $x = x^*$. Now, consider the Lyapunov function $V(x) = \frac{1}{2} \|x - \bar{x}\|^2$ and use the properties of $R$ and $S$ to write the Lyapunov function in terms of $x$ and $v$,

$$V(x) = \frac{1}{2} \|x - \bar{x}\|^2 = \frac{1}{2} \|R^T R + STS) - (R^T R + STS)\|\|^2 = \frac{1}{2} \|R^T x + ST - R^T R\| - S^T S\|\|^2 = \frac{1}{2} \|R^T (x - R\bar{x}) + S^T (z - S\bar{x})\|^2 = \frac{1}{2} \|x - R\bar{x}\|^2 + \frac{1}{2} \|z - S\bar{x}\|^2 = \frac{1}{2} \|x - x^*\|^2 + \frac{1}{2} \|v + S(1_N \otimes x) - S\bar{x}\|^2 = \frac{1}{2} \|x - x^*\|^2 + \frac{1}{2} \|v + S(1_N \otimes [x - x^*])\|^2$$

The third last line follows from equation (4.7), i.e., $R^T S = 0$, $R^T R = I$ and $S^T S = I$. Along the solutions of (4.33),

$$\dot{V} = - (x - x^*)^T [F(1_N \otimes x + ST) - F(1_N \otimes x^*)] - (x - x^*)^T RLS^T v + (v + S(1_N \otimes \bar{x}))^T [\dot{v} + S(1_N \otimes \bar{x})]$$

Substituting $\dot{v}$ and eliminating terms give,

$$\dot{V} = - (x - x^*)^T [F(1_N \otimes x + ST) - F(1_N \otimes x^*)] - (x - x^*)^T RLS^T v + v^T [-SLS^T v - S(1_N \otimes \bar{x})] + (1_N \otimes [x - x^*])^T ST [-SLS^T v - S(1_N \otimes \bar{x})] + v^T S(1_N \otimes \bar{x}) + (1_N \otimes [x - x^*])^T ST S(1_N \otimes \bar{x}) = - (x - x^*)^T [F(1_N \otimes x + ST) - F(1_N \otimes x^*)] - (x - x^*)^T RLS^T v - v^T SLS^T v - (1_N \otimes [x - x^*])^T ST SLS^T v$$

The term $(1_N \otimes [x - x^*]) = (1_N \otimes I_n)[x - x^*]$ and therefore,

$$(1_N \otimes [x - x^*])^T ST SLS^T v = [x - x^*][1_N \otimes I_n]^T ST SLS^T v$$
The term \((1_N \otimes I_n)^T S^T S L S^T v\) can be simplified further with the following lemma.

**Lemma 5.7.** If \(G_c\) is a connected graph then \((1_N \otimes I_n)^T S^T S L = -R L\)

**Proof.** Given the matrix \((1_N \otimes I_n)^T S^T S L\) and using the identity (4.7), the equation can be written as,

\[
(1_N \otimes I_n)^T S^T S L = (1_N \otimes I_n)^T (I_{Nn} - R^T R) L
\]

\[
= (1_N \otimes I_n)^T L - (1_N \otimes I_n)^T R^T R L
\]

\[
= -(1_N \otimes I_n)^T R^T R L
\]

\[
= -R L
\]

Where the first term in the second line is eliminated because of equation (4.28) and the third line is simplified because of equation (4.12). \(\square\)

Using this lemma, \(\dot{V}\) can be reduced further to,

\[
\dot{V} = -(x - x^*)^T \left[ F(1_N \otimes x + S^T v) - F(1_N \otimes x^*) \right] - (x - x^*)^T R L S^T v
\]

\[
- v^T S L S^T v + [x - x^*]^T R L S^T v
\]

\[
= -(x - x^*)^T \left[ F(1_N \otimes x + S^T v) - F(1_N \otimes x^*) \right] - v^T S L S^T v
\]

\[
(5.12)
\]

Similar to Theorem 5.3, the extended pseudo-gradient is decomposed into action and estimate terms.

\[
F(1_N \otimes x + S^T v) - F(1_N \otimes x^*)
\]

\[
= F(1_N \otimes x + S^T v) - F(1_N \otimes x) + F(1_N \otimes x) - F(1_N \otimes x^*)
\]

\[
= \underbrace{F(1_N \otimes x + S^T v) - F(1_N \otimes x)}_{\text{first term}} + \underbrace{F(x) - F(x^*)}_{\text{second term}}
\]

From Assumption 4(ii) the first term is bounded by the Lipschitz condition,

\[
\|F(1_N \otimes x + S^T v) - F(1_N \otimes x)\| \leq \theta \|S^T v\| \leq \theta \|S^T\| \|v\| = \theta \|v\|
\]

and from Assumption 3(ii), the second term is bounded by \((x - x^*)^T (F(x) - F(x^*)) \geq \mu \|x - x^*\|^2\). Combining these conditions and using the assumption that \(\lambda_2(S L S^T) \geq \frac{\lambda_2(L)}{N}\) yields,

\[
\dot{V} \leq -\mu \|x - x^*\|^2 + \theta \|x - x^*\| \|v\| - \frac{\lambda_2(L)}{N} \|v\|^2
\]

\[
\leq -\left[ \|x - x^*\| \|v\| \right] \left[ \frac{\mu}{\theta} - \frac{\theta}{2} \frac{\lambda_2(L)}{N} \right] \left[ \|x - x^*\| \|v\| \right]
\]

\[
\leq -w^T \Theta w
\]
where \( w^T = \left[ \| x - x^* \| \; \| v \| \right] \) and \( \Theta = \begin{bmatrix} \frac{\mu}{\theta} & -\frac{\theta}{2} \\ -\frac{\theta}{2} & \frac{\lambda_2(L)}{N} \end{bmatrix} \). For the matrix \( \Theta \) to be positive definite \( \lambda_2(L) > \frac{N}{4\mu} \theta^2 \). When \( \dot{V} = 0 \) then \( x = x^* \) and \( v = 0 \). Therefore, using Lyapunov’s stability theorem, the equilibrium is asymptotically stable.

**Remark 5.3.** In Theorem 5.5 the bound \( \lambda_2(SLST) \geq \frac{\lambda_2(L)}{N} \) was assumed. Unfortunately, a proof for this bound was not able to be obtained. If we denote \( L_i \) as the Laplacian matrix with row and column \( i \) removed, then the smallest eigenvalue of \( SLST \) is \( \min_i \lambda_1(L_i) \). The closest result that was obtain comes from Theorem 4.8 and Theorem 4.11 in [53], that states that given a connected graph \( G_c \),

\[
\frac{1}{N} \lambda_2(L)\lambda_3\lambda_2(L) \cdots \lambda_N\lambda_2(L) = \det(L_i) \quad \forall i \in [1, \ldots, N]
\]  

(5.14)

This result doesn’t tell what the minimum eigenvalue of \( L_i \) is but the product of the eigenvalues. Testing random Laplacian matrices shows that the bound is \( \lambda_2(SLST) \geq \frac{\lambda_2(L)}{N} \) but this is not a decisive argument.

**Conjecture 5.8.** If \( G_c \) satisfies Assumption 2 then \( \lambda_2(SLST) \geq \frac{\lambda_2(L)}{N} \).

If this conjecture is correct than this assumption can be removed. Additionally, this bound appears to be very conservative for some graphs that were tested. For example a cycle graph the bound appears to satisfy \( \lambda_2(SLST) \geq \frac{\lambda_2(L)}{4} \) regardless of the number of agents in the network.

The next corollary proves convergence of (4.33) when \( -RL[R^T x + S^T z] \) is removed, i.e., (5.15). The following Lemma is used to in the following corollary to created the inequality where the dynamics will converge.

**Lemma 5.9.** If \( G_c \) is a connected graph then \( \| RLST \| = \max_{i \in I} \sqrt{|N_i|} \)

*Proof.* The proof can be found in Appendix A, Lemma A.2.

**Corollary 5.10.** Consider a game \( \mathcal{G}(I, J_i, \Omega_i) \) over a communication graph \( G_c \) under Assumptions 1(i), 2, 3(ii) and 4(ii). Let the overall \( \tilde{P} \) system be,

\[
\tilde{P}: \begin{cases} \dot{x} = -F(R^T x + S^T z) \\ \dot{z} = -SL[R^T x + S^T z] \end{cases}
\]  

(5.15)

Then, if \( \lambda_2(L) > \frac{N(\theta + d)^2}{4\mu} \) where \( d = \max_{i \in I} \sqrt{|N_i|} \) and \( \lambda_2(SLST) \geq \frac{\lambda_2(L)}{N} \), any solution of (4.30) converges asymptotically to \( 1_N \otimes x^* \), and the actions components converge to the NE of the game, \( x^* \).

*Proof.* The dynamics, \( \tilde{P} \), given by (5.15) is the same as (4.33) except that \( -RL[R^T x + S^T z] \) is removed from \( \dot{x} \), i.e., no correction term in the action component. Therefore, following the
Chapter 5. Convergence Analysis

66

Theorem 5.5

\[ \dot{V} = - (x - x^*)^T [F(1_N \otimes x + S^Tv) - F(1_N \otimes x^*)] \]
\[ - v^T SL^Tv + [x - x^*]^T RL^Tv \]

where \( \dot{V} \) is similar to equation (5.12) except there is an extra \([x - x^*]^T RL^Tv\) term. Using Lemma 5.9, this term is bounded by \([x - x^*]RL^Tv \leq d \|x - x^*\| v\) where \(d = \max_{i \in I} \sqrt{|N_i|}\).

Applying the same bounds on the other terms as done in Theorem 5.5 gives,

\[ \dot{V} \leq -w^T \Theta w \]

where \(w^T = [\|x - x^*\| \parallel v\|]\) and \(\Theta = \begin{bmatrix} \mu & -\frac{\theta + d}{2} \\ \frac{\theta + d}{2} & \frac{\lambda_2(L)}{N} \end{bmatrix}\).

Therefore, \(\Theta\) is positive definite if \(\lambda_2(L) > \frac{N(\theta + d)^2}{4\mu}\) and the rest of the proof follows from Lyapunov’s stability theorem.

Theorem 5.11. Consider a game \(G(I, J, \Omega_i)\) over a communication graph \(G_c\) under Assumptions 1(ii), 2, 3(ii) and 4(ii). Let each player’s dynamics \(\tilde{P}_i\) be as in (4.38) or overall \(\tilde{P}\) (4.36). Then, if \(\lambda_2(L) > \frac{N(\theta + d)^2}{4\mu}\) and \(\lambda_2(SL^Tv) \geq \frac{\lambda_2(L)}{N}\), any solution of (4.30) converges asymptotically to \(1_N \otimes x^*\), and the actions components converge to the NE of the game, \(x^*\).

Proof. The proof is similar to the proof of Theorem 5.5 using \(V(t, x, \bar{x}) = \frac{1}{2} \|x(t) - \bar{x}\|^2\). Following the same steps and substituting (4.26) for the \(\frac{1}{2} \|x(t) - x^*\|\) part of the Lyapunov function gives \(\dot{V}\) as in equation 5.11. The rest of the proof follows Theorem 5.5 and then invokes Barbalat’s Lemma to show that \(x(t) \to x^*\) as \(t \to \infty\).

The corresponding Corollary follows the exact steps as Corollary 5.10 but for the projection dynamics.

Corollary 5.12. Consider a game \(G(I, J, \Omega_i)\) over a communication graph \(G_c\) under Assumptions 1(i), 2(iii) and 4(ii). Let the overall \(\tilde{P}\) system be,

\[
\tilde{P} : \begin{cases} 
\dot{x} &= \Pi \Omega (x, -F(R^T x + S^T z)) \\
\dot{z} &= -S L [R^T x + S^T z] 
\end{cases} \tag{5.16}
\]
Then, if \( \lambda_2(L) > \frac{N(\theta + \mu)^2}{4\mu} \) where \( d = \max_{i \in I} \sqrt{|N_i|} \) and \( \lambda_2(\mathbf{SL} \mathbf{ST}) \geq \frac{\lambda_2(L)}{N} \), any solution of (4.30) converges asymptotically to \( 1_N \otimes x^* \), and the actions components converge to the NE of the game, \( x^* \).

Proof. The proof follows the steps of Corollary 5.10 for the projection case in Theorem 5.11.

5.3 Two-Timescale Singular Perturbation Analysis

In this section we show a trade-off between the speed of the estimate dynamics and the connectivity on \( G_c \) in Theorem 5.3, based on a time-scale separation argument. The idea is to modify the dynamics of the estimates in \( \tilde{\mathcal{P}} \) (4.30) or (4.33), such that the system approaches quickly to the consensus subspace. The condition that ensures convergence in the previous section is dependent on \( \theta, \mu \) and \( \lambda_2(L) \), e.g., \( \lambda_2(L) > \frac{\theta^2}{\mu} + \theta \). If the graph is not sufficiently connected this inequality will not hold true. By speeding up the estimate component dynamics this condition will be modified by a gain term that describes the speed of the consensus dynamics. If the estimate components converge quickly then the system will behave more like the perfect information case. Under Assumption 3(ii) and 4(ii), we show convergence to the NE over a sufficiently connected \( G_c \) by using a time-scale decomposition approach.

Recall the equivalent representation of \( \tilde{\mathcal{P}} \) in (4.33) and modify the estimate component of the dynamics, such that it is much faster than the action component,

\[
\tilde{\mathcal{P}}_\epsilon : \begin{cases}
\dot{x} &= -\mathbf{F}(\mathbf{R}^T x + \mathbf{S}^T z) - \mathbf{RL}[\mathbf{R}^T x + \mathbf{S}^T z] \\
\epsilon \dot{z} &= -\mathbf{SL}[\mathbf{R}^T x + \mathbf{S}^T z]
\end{cases}
\] (5.17)

The constant \( \epsilon \) is greater than zero and controls the speed of the \( z \) dynamics. Thus, player \( i \)'s dynamics is as follows:

\[
\tilde{\mathcal{P}}_{i,\epsilon} : \begin{bmatrix}
\dot{x}_i \\
\dot{x}_i^{{\perp}_i}
\end{bmatrix} = \begin{bmatrix}
\nabla_i J_i(x_i, x^i_i) - \mathcal{R}_i \sum_{j \in N_i} (x^i_i - x^j) \\
-\frac{1}{\epsilon} \mathcal{S}_i \left( \sum_{j \in N_i} x^i_i - x^j \right)
\end{bmatrix}
\] (5.18)

with the \( \frac{1}{\epsilon} \) high gain factor on the estimate component. Equation (5.17) is in the standard form of a singularly perturbed system,

\[
\tilde{\mathcal{P}}_\epsilon : \begin{cases}
\dot{x} &= f(x, z) \\
\epsilon \dot{z} &= g(x, z)
\end{cases} \quad \text{or} \quad \tilde{\mathcal{P}}_\epsilon : \begin{cases}
\dot{x} &= f(t, x, z, \epsilon) \\
\epsilon \dot{z} &= g(t, x, z, \epsilon)
\end{cases}
\] (5.19)

The estimate dynamics and the action dynamics are the fast and the slow systems, respectively.
5.3.1 Gradient-Play Dynamics with Estimate Correction

**Theorem 5.13.** Consider a game $G(I,J,\Omega_i)$ over a communication graph $G_c$ under Assumptions (i), (ii) and (iii). Let each players’ dynamics, $\bar{P}_i,\epsilon$, be given by equation (5.18), or overall dynamics, $\bar{P}_\epsilon$, as in equation (5.17).

Then, there exists $\epsilon^* > 0$, such that for all $\epsilon \in (0,\epsilon^*)$, $(x^*,S(1_N \otimes x^*))$ is asymptotically stable. Additionally, the system is asymptotically stable, for all $\epsilon \in (0,1)$ if

$$\lambda_2(L) > \epsilon N \sqrt{N-1}(\frac{\theta}{\mu} + 1)(\theta + d))$$

where $d = \max_{i \in I} \sqrt{|N_i|}$ and $\lambda_2(SLS^T) \geq \frac{\lambda_2(L)}{N}$. Furthermore, if $J(x)$ is $C^2$ then there exists $\epsilon^* > 0$, such that for all $\epsilon \in (0,\epsilon^*)$, $(x^*,S(1_N \otimes x^*))$ is exponentially stable.

**Proof.** We analyze (5.17) by examining the reduced and the boundary-layer systems. First we find the roots of $SL[R^T x + S^T z] = 0$, or $SLx = 0$. By equation (4.8) $x \in \text{Null}(SL)$ if and only if $Lx \in \text{Null}(S)$, which is equivalent to $Lx \in \text{Range}(R^T)$. Thus, $x \in \text{Null}(SL)$ if and only if there exists a $q \in \mathbb{R}^n$ such that $Lx = R^T q$. Then, for $q \in \mathbb{R}^n$ and for all $w \in \mathbb{R}^n$, $(1_N \otimes w^T)Lx = (1_N \otimes w^T)R^T q$. By using (4.29) the term $(1_N \otimes w^T)Lx = 0$. Additionally, applying identity (4.12) the equation becomes $w^T q = 0$, for all $w \in \mathbb{R}^n$ and therefore $q = 0$ and $Lx = 0$. By property (4.28), $x = 1_N \otimes x$ and $x \in \Omega$. Hence, the roots of $SL[R^T x + S^T z] = 0$ are $x = (1_N \otimes x)$ for any $x \in \Omega$. This implies that $SL[R^T x + S^T z] = 0$ when $z = Sx = S(1_N \otimes x)$.

We make a change of coordinates from $z$ to $v$ via $v = z - S(1_N \otimes x)$, to shift the equilibrium of the boundary-layer system to the origin. First, we use $z = v + S(1_N \otimes x)$ and $x = R(1_N \otimes x)$ to rewrite the term $R^T x + S^T z$ that appears in (5.17) as follows,

$$R^T x + S^T z = R^T R(1_N \otimes x) + S^T (v + S(1_N \otimes x)) = (R^T R + S^T S)(1_N \otimes x) + S^T v = 1_N \otimes x + S^T v$$

where identity (4.7) was used. Substituting this and the change of variables $v = z - S(1_N \otimes x)$ into equation (5.17) yields,

$$\dot{x} = -F(1_N \otimes x + S^T v) - RLS^T v$$

$$\tilde{P}_\epsilon : \begin{cases} \dot{\epsilon} = -SLS^T v + \epsilon S(1_N \otimes RLS^T v) \\ + \epsilon S(1_N \otimes F(1_N \otimes x + S^T v)) \end{cases}$$

where $L(1_N \otimes x) = 0$ was utilized to simplify the equation. Note that $v = 0$ is the quasi-steady state of $\dot{\epsilon}$ and substituting this in $\dot{x}$ gives the reduced system,

$$\dot{x} = -F(1_N \otimes x) = -F(x)$$

which is exactly the gradient dynamics and has equilibrium $x^*$, which is the NE. By Lemma
the gradient dynamics, \((5.21)\), is exponentially stable.

The boundary-layer system on the \(\tau = t/\epsilon\) timescale is

\[
\frac{dv}{d\tau} = -SLS^Tv
\]

where \(SLS^T\) is positive definite as shown in Lemma \(5.6\) and therefore is exponentially stable. By Theorem \(2.14\), it follows that there exists an \(\epsilon^* > 0\), such that for all \(\epsilon < \epsilon^*\), the equilibrium point \((x^*, 0)\) is exponentially stable for system \((5.20)\). Equivalently, \((x^*, S(1_N \otimes x^*))\) is exponentially stable for system \((5.17)\).

Alternatively, Theorem \(2.13\) can be applied to \((5.20)\) to show asymptotic stability. The two Lyapunov functions are \(U(x) = \frac{1}{2} \|x - x^*\|^2\) and \(W(v) = \frac{1}{2} \|v\|^2\), and along the reduced and the boundary layer-systems, \((5.21), (5.22)\), the following hold

\[
-x^T F(1_N \otimes x) \leq -\mu \|x - x^*\|^2
\]

\[
-v^T SLS^Tv \leq -\frac{\lambda_2(L)}{N} \|v\|^2
\]

Therefore, condition 1, \((2.7)\) and condition 2 \((2.8)\) in Theorem \(2.13\) hold where \(\alpha_1 = \mu\), \(\psi_1(x) = \|x - x^*\|\), \(\alpha_2 = \frac{\lambda_2(L)}{N}\), and \(\psi_2(v) = \|v\|\). Condition 3, \((2.9)\), is trivially satisfied and condition 4, \((2.10)\),

\[
(x - x^*)^T \left[ -F(1_N \otimes x + S^Tv) - RLs^Tv + F(1_N \otimes x) \right]
\]

\[
\leq (\theta + d) \|x - x^*\| \|v\|
\]

is met for \(\beta_1 = \theta + d\) and where \(d = \max_i \sqrt{|N_i|}\) from Lemma \(5.9\). The left hand side of condition 5, \((2.11)\), is,

\[
v^T S(1_N \otimes RLs^Tv) + v^T S(1_N \otimes F(1_N \otimes x + S^Tv)) \]

\[
= v^T S(1_N \otimes I_n)RLs^Tv + v^T S(1_N \otimes I_n)F(1_N \otimes x + S^Tv)
\]

Adding the term \(-F(1_N \otimes x^*) = 0\) and following the steps in \(5.13\) we bound \(5.23\)

\[
\leq v^T \|S(1_N \otimes I_n)\| \|RLs^T\| \|v + v^T S(1_N \otimes I_n) \left[ F(1_N \otimes x + S^Tv) - F(1_N \otimes x^*) \right]\
\leq \|S(1_N \otimes I_n)\| \|RLs^T\| \|v\|^2 + \|v\| \|S(1_N \otimes I_n)\| [\theta \|x - x^*\| + \theta \|v\|]
\]

Using Lemma \(4.1\) and the known bounds on the other terms, equation \(5.23\) is bounded by,

\[
\leq \sqrt{N - 1} \theta \|x - x^*\| \|v\| + \sqrt{N - 1}(\theta + d) \|v\|^2
\]

and condition 5, \((2.11)\) holds for \(\beta_2 = \sqrt{N - 1} \theta\), \(\gamma = \sqrt{N - 1}(\theta + d)\).

Therefore, all conditions are satisfied and we can apply Theorem \(2.13\). In fact the Lyapunov
function \( V(x) \) used is,

\[
V(x) = (1 - \delta)U(x) + \delta W(v)
\]

\[
= (1 - \delta) \frac{1}{2} \|x - x^*\|^2 + \delta \frac{1}{2} \|v\|^2
\]

where \( \delta = \frac{\beta_1}{\beta_1 + \beta_2} \). The maximum value of epsilon is \( \epsilon^* = \frac{\alpha_1 \alpha_2}{\alpha_1^2 + \beta_1 \beta_2} \) and is given as

\[
\epsilon^* = \frac{\lambda_2(L)\mu}{N\sqrt{N - 1}(\theta + \mu)(\theta + d)}
\]

Furthermore, any \( 0 < \epsilon < 1 \) such that

\[
\lambda_2(L) > \epsilon N\sqrt{N - 1}(\frac{\theta}{\mu} + 1)(\theta + d)
\]

the point \((x^*, 0)\) is asymptotic stable for (5.20), hence \((x^*, S(1_N \otimes x^*))\) is asymptotic stable for (5.17).

**Remark 5.4.** For \( \tilde{P}_{i, \epsilon} \) in equation (5.18) the estimate dynamics is made faster with the gain \( \frac{1}{\epsilon} \). It can be shown that for the upper bound on \( N \), \( \frac{1}{\epsilon} > N\sqrt{N - 1}(1 + \frac{\theta}{\mu}) \) and for small \( \epsilon \), the bound on \( \lambda_2(L) \) in Theorem 5.13 is lower than the bound on \( \lambda_2(L) \) in Theorem 5.3.

Alternatively, we can consider a gain parameter \( \frac{1}{\epsilon} > 0 \) on the estimates in (4.30) to improve the lower bound to \( \lambda_2(L) > \epsilon(\frac{\theta^2}{\mu} + \theta) \), as shown in the next section. Thus, a smaller \( \epsilon \) can relax the connectivity bound on \( L \), but \( \epsilon \) is a global parameter. This highlights another aspect of the trade-off between the properties of the game (coupling) and properties of the communication graph.

The following Corollary removes the \(-\mathcal{R}_L\mathcal{R}^T x + \mathcal{S}^T z\) term from (5.17). Removing this term has the effect of making the action dynamics completely determined by the gradient.

**Corollary 5.14.** Consider a game \( \mathcal{G}(I, J_i, \Omega_i) \) over a communication graph \( G_c \) under Assumptions 1(i), 2, 3(ii) and 4(ii). Let overall dynamics be,

\[
\tilde{P}_{i, \epsilon}:
\begin{align*}
\dot{x} &= -F(R^T x + S^T z) \\
\epsilon \dot{z} &= -SL[R^T x + S^T z]
\end{align*}
\]

(5.24)

where \( \epsilon > 0 \). Then, there exists an \( \epsilon^* > 0 \), such that for all \( \epsilon \in (0, \epsilon^*) \), the equilibrium point \((x^*, S(1_N \otimes x^*))\) is exponentially stable. Alternatively, \((x^*, S(1_N \otimes x^*))\) is asymptotically stable for all \( \epsilon \in (0, 1) \) if

\[
\lambda_2(L) > \epsilon N\sqrt{N - 1}(\frac{\theta^2}{\mu} + \theta).
\]

**Proof.** The proof follows the same steps as Theorem 5.13 except that the \( \mathcal{R}_L S^T \) term has been removed. The effect of removing this term is that the \( d \) term in the inequality will be removed. \( \square \)
Remark 5.5. The steps and structure of the proof for Theorem 5.3 and 5.11 look like a singular perturbation method. Unfortunately, if the Lyapunov function in Theorem 5.5 is used, \( \epsilon \) will appear in a term with coupling, i.e.,

\[
\dot{V} \leq -\left[\|x - x^*\| \|v\|\right] \left[-\theta - \frac{\mu}{2} - \frac{1}{2}(1 - 1)d - \frac{1}{2}\left(\frac{1}{\epsilon} - 1\right)d\right] \left[\|x - x^*\|\right]
\]

where \( d = \max_{i \in I} \frac{1}{\sqrt{|N_i|}} \) and \( \frac{1}{\epsilon} \geq 1 \). This coupling limits the range of \( \epsilon \in (\epsilon_l^*, \epsilon_u^*) \) with some lower bound. Having the epsilon restricted to such an interval is nearly impossible to know. It would require agents to know the network configuration and the properties of all agents’ cost functions, rendering the bound for this method impractical at best.

5.3.2 Projected Gradient with Estimate Correction

Doing the two-timescale analysis for projection dynamics fails to follow through since the Lipschitz condition doesn’t hold on the boundary. There doesn’t seem to be any obvious way to get a Lipschitz bound so the analysis fails. The following is not a two-timescale decomposition, but highlights that scaling the Laplacian improves convergence. Consider a gain parameter \( \frac{1}{\epsilon} > 0 \) on the Laplacian in (4.36) to improve the lower bound condition on \( \lambda_2(L) \). The system with this gain parameter is,

\[
\tilde{P}_\epsilon : \dot{x} = R^T \Pi \Omega(x, -F(x) - \frac{1}{\epsilon}RLx) - \frac{1}{\epsilon}S^TSLx
\]  

(5.25)

Thus, player \( i \)'s dynamics is as follows:

\[
\tilde{P}_{i,\epsilon} : \begin{bmatrix} \dot{x}_i \\ \dot{x}^i \\ \dot{x}^{i-1} \end{bmatrix} = \begin{bmatrix} \Pi \Omega_i(x_i, -\nabla_i J_i(x_i, x^i) - \frac{1}{\epsilon}R_i \sum_{j \in N_i}(x^j - x^i)) \\ -\frac{1}{\epsilon}S_i(\sum_{j \in N_i}x^j - x^i) \end{bmatrix}
\]  

(5.26)

Following the proof of Theorem 5.3, the matrix \( \Theta \) becomes

\[
\Theta = \begin{bmatrix} \mu & -\theta \\ -\theta & \frac{1}{\epsilon} \lambda_2(L) - \theta \end{bmatrix}
\]

where the condition for \( \Theta \) to be positive definite is \( \lambda_2(L) > \epsilon(\frac{\theta^2}{\mu} + \theta) \).

In this chapter, (4.30) and (4.36) are proved to converge for monotone extended pseudo-gradient and strongly monotone pseudo-gradient games. Various Lyapunov functions were used to prove convergence under different conditions, i.e., \( \lambda_2(L) > \frac{\theta^2}{\mu} + \theta \). In the second part of this chapter the estimate dynamics are made faster so that the dynamics can converge under weaker graph connectivity conditions. Speeding up the estimate dynamics makes it possible for (4.30) and (4.36) to con-
verge in cases where the graph connectivity is low.
Chapter 6

Modifying Augmented Gradient Dynamics with Estimate Correction

In this chapter other changes to (4.30) and (4.36) are made to improve transient behaviour, ensure convergence, or deal with dynamical games. In Section 6.1 dynamics (4.30) are put into a block diagram representation and compared with block diagrams from optimization, to motivate which blocks to change. In Section 6.2 the $B$ matrix in (4.16) is modified,

$$\dot{x} = -\mathcal{R}^T \mathbf{F}(x) + Bu$$

$$= -\mathcal{R}^T \mathbf{F}(x) - BLx$$

The $B$ matrix can be changed to make $BL$ a weighted Laplacian to improve convergence. Section 6.3 changes the static feedback $u = -Lx$ to a dynamical system. Section 6.4 changes (4.30) from a second order system into a higher order system. Having higher order dynamics improves the transient of the system because it has the effect of passing $-\mathcal{R}^T \mathbf{F}(x) - BLx$ through a filter. Additionally, higher order dynamics have the interpretation of using (4.30) to play a dynamical game.

6.1 Connections with Optimization

The dynamics $\tilde{P}$, (4.30), reveal an interesting connection between distributed optimization and passivity based control, [1], [26]. Using $L = L \otimes I_n$ and $L = QQ^T$, where $Q$ is the incidence matrix, the interconnected block-diagram of (4.30) is shown in Figure 6.1. In distributed optimization (e.g. [1]) and passivity based control (e.g. [26]) the dynamics are decoupled as shown in figure 6.2.

In figure 6.1 the dynamics on the top path are coupled, due to the inherent coupling in players’ cost functions in the game $\mathcal{G}(I, J, \Omega)$, i.e., $z$ affecting $\mathbf{F}(\cdot)$. Recall from chapter 3 that $\mathbf{F}(x) = \mathbf{F}(\mathcal{R}^T x + \mathcal{S}^T z) = [\nabla_1 J_1^T(x^1), \ldots, \nabla_N J_N^T(x^N)]^T$ and player $i$’s gradient is
\[ \dot{\mathbf{x}} = -\mathbf{F}(\mathbf{R}^T \mathbf{x} + \mathbf{S}^T \mathbf{z}) + \mathbf{u} \]

\[ \mathbf{z} = \begin{bmatrix} \frac{1}{s} \\ \vdots \\ \frac{1}{s} \end{bmatrix}, \quad \mathbf{S}^T \mathbf{z} \]

\[ \mathbf{Q} \otimes \mathbf{I}_n \rightarrow \mathbf{Id} \rightarrow \mathbf{Q}^T \otimes \mathbf{I}_n \rightarrow \mathbf{x} \]

\[ \nabla_i J_i(\mathbf{x}^i) = \frac{\partial J_i(\mathbf{x}^i)}{\partial \mathbf{x}^i} \text{ and not } \nabla_{\mathbf{x}^i} J_i(\mathbf{x}^i) = \frac{\partial J_i(\mathbf{x}^i)}{\partial \mathbf{x}^i} \text{ as in the optimization case. Therefore, in figure 6.2, each agent is taking the gradient of their cost function and changing all components in the state vector } \mathbf{x}^i \text{ and in figure 6.1 player } i \text{'s gradient only changes } \mathbf{x}_i^i. \]

Additionally, notice that pre-multiplication by } \mathbf{Q} \text{ and post-multiplication by } \mathbf{Q}^T \text{ preserves passivity of the system, i.e., Lemma 2.18. Figure 6.1 and figure 6.2 suggests that a passive dynamical system can be designed for the identity block on the feedback path. One such dynamical system can be obtained by substituting the static feedback through } \mathbf{L}, \text{ with an integrator or proportional-integrator term through } \mathbf{L}, \text{ which preserves passivity, as in [1]. Thus, a passivity interpretation for the Nash equilibrium seeking dynamics, allows for a control theoretic method for deriving new dynamics/algorithms. Figure 6.3 shows further decomposition of figure 6.1 as well as introducing an additional feedback path. The new dynamics for the Laplacian are contained in } \mathcal{P}_r. \text{ System } \mathcal{P}_v \text{ was a bank of integrators for the original system, i.e, } \dot{v} = u_v \text{ and } y_v = v. \text{ The integrators can be replaced by some other system to create a new set of dynamics. For example, the integrators can be replaced by } \frac{1}{s(s+d)} \text{ for some } d > 0 \text{ and the Nash equilibrium of the overall system will} \]
Figure 6.3: Additional Decomposition and Feedback connections

remain the same. From the passivity viewpoint, if the system is stable and the feedback $\mathcal{P}_w$ also has some passivity properties, then convergence follows directly from a passivity based argument. As long as the feedback doesn’t change the equilibrium point away from the NE, $\mathcal{P}_w$ could also be designed for other performance objectives, i.e., disturbance rejection.

In Section 6.2, we modify the $B$ matrix that was introduced in (4.16). In Section 6.3, we change the static Laplacian feedback to be a dynamical system. Lastly, in Section 6.4, we design $\mathcal{P}_v$ to create higher order dynamics.

### 6.2 B matrix / weighted Laplacian

In section 4.2.1 equation (4.16) is,

$$
\bar{\Sigma} : \begin{cases} 
\dot{x} = -R^T F(x) + Bu \\
y = B^T x 
\end{cases}
$$

(6.1)

and the $B$ matrix was dropped to simplify the analysis. If the $B$ matrix was kept then the Laplacian term would become $BLB^T$. If $B$ is additionally assumed to be non-singular then
from *Sylvester’s Law of Inertia* the number of positive, zero and negative eigenvalues stay the same \[54\]. Unfortunately, if \(B_1\) is any matrix and \(B_2 = \cdots = B_N = I_n\) then the nullspace will change and no longer be the consensus subspace. Even if \(B_i = b_i I_n\) for some \(b_i \in \mathbb{R}\), the null space will change. If \(B_1 = \cdots = B_N = M \in \mathbb{R}^n\), then the null space will remain the same but it requires that all agents know what this transformation matrix \(M\) is.

Therefore, the pre/post multiplication of \(B\) is not practical without having a shared \(M\) matrix. However, this could be implemented as some communication protocol that all agents must use. The protocol applies some time varying transform on the action vector.

**Example 6.1.** Consider a two player game \(\mathcal{G}\), where \(B_i = M(t)\) is some time varying communication protocol. Then, the Laplacian with the communication protocol is,

\[
L = \begin{bmatrix}
1 & 0 & -1 & 0 \\
0 & 1 & 0 & -1 \\
-1 & 0 & 1 & 0 \\
0 & -1 & 0 & 1
\end{bmatrix}
\]

\[
M(t) = \begin{bmatrix}
M_{1,1}(t) & M_{1,2}(t) & 0 & 0 \\
M_{2,1}(t) & M_{2,2}(t) & 0 & 0 \\
0 & 0 & M_{1,1}(t) & M_{1,2}(t) \\
0 & 0 & M_{2,1}(t) & M_{2,2}(t)
\end{bmatrix}
\]

\[
N(t) = \begin{bmatrix}
M^2_{1,1}(t) + M^2_{1,2}(t) & M_{1,1}(t)M_{1,2}(t) + M_{1,2}(t)M_{2,2}(t) \\
M_{1,1}(t)M_{1,2}(t) + M_{1,2}(t)M_{2,2}(t) & M^2_{2,1}(t) + M^2_{2,2}(t)
\end{bmatrix}
\]

\[
M(t)LMT(t) = \begin{bmatrix}
N(t) & -N(t) \\
-N(t) & N(t)
\end{bmatrix}
\]

Using this method, each row in \([M(t)LMT(t)]x\) is now taking the difference between some linear combination of an agent’s estimate vector against other agents’ linear combination. Without knowing what \(M(t)\), each agent can’t determine what the estimate vector of the other agents.

The example above shows that the \(M(t)\) matrix can be used as a security method to hide what the other agents’ estimates are. This is an interesting possibility for the \(B\) matrix, but is not looked at in this thesis. An alternative is to remove the \(B\) matrix from the output equation and just have the system,

\[
\Sigma : \begin{cases}
\dot{x} = -R^T F(x) - BLx \\
y = x
\end{cases} \tag{6.2}
\]

If each agent selects \(B_i = \text{diag}(b_{i,1}, \ldots, b_{i,n})\), and \(b_{i,j} > 0 \in \mathbb{R}\) then \(\text{null}(BL) = \text{null}(L)\) and the NE doesn’t change. This \(B\) matrix has the effect of scaling the relative difference of each estimate component, i.e., \([BLx]_1 = b_{1,1} \sum_{i \in N_i}(x^*_i - x^*_1)\).

All the theorems in Chapter \[3\] prove convergence when a inequality between the algebraic connectivity of the graph and the Lipshitz and monotonicity constant are satisfied, i.e., \(\lambda_2(L) > \frac{\theta^2}{\mu} + \theta\). To converge in communication graphs with small algebraic connectivity, a gain \(\epsilon\) was introduced to increase the convergence speed to the consensus subspace. If agents select an
appropriate $B_i$ it will have the same effect as the gain term $\epsilon$.

If there is a graph structure that is imposed that agents can determine, then $\lambda_2(L)$ can be calculated. If on the other hand, the graph is completely random, but still connected, then the lowest possible algebraic connectivity of a connected graph is a path and is bounded by, $2(1 - \cos(\frac{\pi}{N})) \leq \lambda_2(L)$ as given in table 2.2. Since agents are storing estimates of the other agents actions, the value of $N$ is known by all agents. Therefore, all agents can calculate $2(1 - \cos(\frac{\pi}{N}))$ without having to know what the communication graph is.

Notice that the Lipschitz constant $\theta$ can be determined by the components of the pseudo-gradient $F$

$$
\|F(x) - F(x')\|^2 = \sum_{i}^{N} \|\nabla_i J_i(x^i) - \nabla_i J_i(x'^i)\|^2 
\leq \sum_{i}^{N} \theta_i^2 \|x^i - x'^i\|^2
$$

which shows that

$$
\|F(x) - F(x')\| \leq \theta_I \|x - x'\|
$$

where $\theta_I = \max_{i \in I} \theta_i$. With an additional assumption that the agents’ cost functions are strongly monotone with respect to their own action then the strong monotone condition can be bounded by,

$$
\|F(x) - F(x')\| \geq \mu_I \|x - x'\|
$$

where $\mu_I = \min_{i \in I} \mu_i$. Therefore, if agents select $B_i = b_i I_n$ where $b_i > \frac{1}{2(1 - \cos(\frac{\pi}{N}))} \left( \frac{\theta_i^2}{\mu_i} + \theta_I \right)$, then $\lambda_2(L) > \frac{\theta_i^2}{\mu_i} + \theta$. Unfortunately, from this equation you can see that everything can be calculated except for $\theta_I$ and $\mu_I$. If each agent knows the structure of all other agents’ cost function then these quantities can be computed a priori, but if the cost functions are completely unique with no underlying structure then this information must be passed through the network.

Similar to the effects of the $B$ matrix, the Laplacian can be replaced by a weighted Laplacian $L_w \in \mathbb{R}^{Nn \times Nn}$, defined as

$$
[L_w]_{i,j} = \begin{cases} 
\sum_{k=1}^{N_n} w_{i,k} & i = j \\
-w_{i,k} & i \neq j \\
0 & \text{Otherwise}
\end{cases}
$$

where $w_{i,j} = w_{j,i} > 0$. When $B_i = \text{diag}(b_{i,1}, \ldots, b_{i,n})$, and $b_{i,j} > 0 \in \mathbb{R}$, the $n(i-1) + j$ row of $BL$ is $n(i-1) + j$ row of $L$ scaled by $b_{i,j}$. This scales all estimate components equally. However, agents might have an neighbour with a more valuable estimate and would like to put
more weight on their estimate. The benefit of using the weighted Laplacian is similar to the $B$ matrix, making the consensus dynamics converge faster. For example all agents value $x_i^i$ more than $x_j^i$ where $j \neq i$ because $x_j^i$ is not an estimate, it is player $i$’s actual action. Therefore, players could weigh other agent’s action component $\alpha > 1$ and estimate components 1, i.e.,

$$
L_w = \begin{bmatrix}
2\alpha & 0 & 0 & -\alpha & 0 & 0 & -\alpha & 0 & 0 \\
0 & \alpha + 1 & 0 & 0 & -\alpha & 0 & 0 & -1 & 0 \\
0 & 0 & \alpha + 1 & 0 & 0 & -1 & 0 & 0 & -\alpha \\
-\alpha & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 \\
0 & -\alpha & 0 & 0 & \alpha & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\
-\alpha & 0 & 0 & -1 & 0 & 0 & \alpha + 1 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & -\alpha & 0 & 0 & 0 & 0 & 0 & \alpha
\end{bmatrix}
$$

Another possible weighted Laplacian is for games where $\forall i \in I$, player $i$’s cost function is dependent on a set of players $\Psi_i \subset I$. Additionally, $\forall j \in \Psi_i$, player $j$’s cost function is dependent on $i$’s action. Players could give higher weight to neighbours actions that directly effect their cost function instead of giving higher weight for all neighbouring agents.

### 6.3 Passive Consensus Dynamics

The feedback equation used to ensure consensus is the static map $u = -Lx$, as shown in figure 4.3. The Laplacian matrix is equal to the pre and post multiplication of the incidence matrix $Q \otimes I_n$ with the identity function as shown in figure 6.1. The identity function can be replaced by a dynamical system to improve transient behaviour and convergence as shown in figure 6.4.

Let the identity function in figure 6.1 or $P_r$ in figure 6.4 be replaced by the LTI system,

$$
\dot{x} = -RF(x) + u_x \\
y_x = x
$$

$Q \otimes I_n$

$Q^T \otimes I_n$

$P_r : \begin{cases}
\dot{r} = f_r(r, u_r) \\
y_r = h(r, u_r)
\end{cases}$

$u_r$

Figure 6.4: Gradient in feedback with passive system
where the matrices $A \in \mathbb{R}^{q \times q}$, $B \in \mathbb{R}^{q \times |E|n}$, $C \in \mathbb{R}^{|E|n \times q}$ and $D \in \mathbb{R}^{|E|n \times |E|n}$ are the matrix dimensions\(^1\) and $r \in \mathbb{R}^q$. The identify function is equivalent to the LTI system where $A = 0, B = 0, C = 0,$ and $D = I$. A popular option for consensus algorithms (\cite{1, 30}) is to use,

$$
\mathcal{P}_r : \begin{align*}
\dot{r} &= 0 + (Q \otimes I)u_r \\
y_r &= (Q^T \otimes I)r + Iu_r
\end{align*}
$$

The input to the gradient dynamics is $u = -(Q \otimes I)y_r$ and the input to the LTI system is $u_r = (Q^T \otimes I)y = (Q^T \otimes I)x$. Substituting these values into \(6.3\) the dynamics from $x$ to $u$ is,

$$
\dot{r} = Lx \\
u = -Lr - Lx
$$

The overall dynamics of this system is given by,

$$
\mathcal{\tilde{P}} : \begin{align*}
\dot{x} &= -\mathcal{R}^T\mathbf{F}(x) - Lr - Lx \\
\dot{r} &= Lx
\end{align*}
$$

If $G_c$ is a connected communication graph, then $\dot{r} = 0$ when $x = (1_n \otimes x) \in null(L)$. This implies that $\dot{x} = 0$ when $\mathcal{R}^T\mathbf{F}(x) = -Lr$. Following the same steps as in Lemma 4.5 this can only occur if $r \in null(L)$ and $F(x) = 0$. Using the candidate Lyapunov function $V(x, r) = \frac{1}{2} \|x - x^*\|^2 + \frac{1}{2} \|r - r^*\|^2$ we see that,

$$
\dot{V}(x, r) = (x - x^*)^T (-\mathcal{R}^T\mathbf{F}(x) - \mathbf{F}(x^*) - L(x - x^*)) \\
- (x - x^*)^T L(r - r^*) + (r - r^*)^T L(x - x^*)
$$

where the first line is the same as $\dot{V}(x)$ in equation (5.2) and the second line is 0. Therefore, this system can be shown to converge to the NE following any of the methods in Chapter 5.

Generally, using a LTI system $\mathcal{P}_r$, the overall dynamics will be,

$$
\mathcal{\tilde{P}} : \begin{align*}
\dot{x} &= -\mathcal{R}^T\mathbf{F}(x) - (Q \otimes I)C r - (Q \otimes I)D(Q^T \otimes I)x \\
\dot{r} &= Ar + B(Q^T \otimes I)x
\end{align*}
$$

**Theorem 6.2 (LTI Laplacian Dynamics).** Consider a game $\mathcal{G}(I, J, \Omega_i)$ over a communication graph $G_c$ under Assumptions 1(i), 2 and either Assumption 3(i), 4(i) or Assumption 3(ii),

\(^1\)The input, $u_r$, is coming from $Q^T \otimes I_n$ and the output $y_r$ is feeding into $Q \otimes I_n$, where $Q^T \in \mathbb{R}^{|E| \times N}$. As a reminder $|E|$ is the number of edges in the graph. This is why the dimension has the term $|E|n$.\]
Let the Laplacian dynamics be given by equation (6.3) and the overall interconnected system dynamics as (6.5). If the LTI system (6.3) satisfies the following conditions:

1. The matrix $D$ has full rank and $x^T(D + D^T)x \geq \lambda_d \|x\|$, where $\lambda_d > 0$, i.e., positive definite.

2. If $A \neq 0$ then,
   
   (a) The matrix $A$ is invertible.
   
   (b) The matrix $CA^{-1}B + D$ has full rank
   
   (c) Satisfies the inequality, $PA + A^T P < 0$, for some positive definite matrix, $P$.

2'. If $A = 0$
   
   (a) The null space of the $B$ matrix is empty, i.e., $\text{Null}(B) = \{0\}$.

3. The equality, $C^T(Q^T \otimes I) = PB(Q^T \otimes I) + (Q \otimes I)B^TP$ holds for some positive definite matrix, $P$.

Then, the action vector $x$ converges to $1_N \otimes x^*$ and the $x$ component is asymptotically stable.

Proof. The first step is to show that the equilibrium, $\bar{x}$ of (6.5) is the Nash equilibrium. When $A \neq 0$ the equilibrium condition $\dot{x} = 0$ is,

$$0 = -R^T F(x) - (Q \otimes I)Cr - (Q \otimes I)D(Q^T \otimes I)x$$

Pre-multiply both sides by $(1_N^T \otimes I_n)$ and following the steps as in Lemma 4.5 we get $0 = -F(x)$. Plugging this result back into the equation above we get,

$$0 = -(Q \otimes I)Cr - (Q \otimes I)D(Q^T \otimes I)x$$

The equilibrium condition for $\dot{r} = 0$ is $r = A^{-1}B(Q^T \otimes I)x$. Substituting $r$ into the equation above gives,

$$0 = -(Q \otimes I)CA^{-1}B(Q^T \otimes I)x - (Q \otimes I)D(Q^T \otimes I)x$$

$$= -(Q \otimes I)(CA^{-1}B + D)(Q^T \otimes I)x \quad (6.6)$$

The matrix $(Q \otimes I)$ has full rank and $(CA^{-1}B + D)$ has full rank by condition (2). Therefore, the equation is equal to zero when $x \in \text{Null}(Q^T \otimes I) = \text{Null}(L)$, which implies $\bar{x} = 1_N \otimes x^*$. When the matrix $A$ is zero then $\dot{r} = 0 = B(Q^T \otimes I)x$. The matrix $B$ has full rank by condition (2') and just as before the equation is equal to zero when $x \in \text{Null}(Q^T \otimes I) = \text{Null}(L)$ and therefore $\bar{x} = 1_N \otimes x^*$.

Next, convergence to the Nash equilibrium is established. Let the candidate Lyapunov function be $V(x, r) = \frac{1}{2} \|x - x^*\|^2 + (r - r^*)^T P(r - r^*)$ where $P$ satisfies condition (2c) when
\( A \neq 0 \) and \( P = 0 \) if \( A = 0 \). Then, along the solution trajectory of (6.5),

\[
\dot{V}(x, r) = -(x - x^*)^T R^T [F(x) - F(x*)] - (x - x^*)^T (Q \otimes I) D (Q^T \otimes I) (x - x^*) \\
- (x - x^*)^T (Q \otimes I) C (r - r^*) + (r - r^*)^T [PB (Q^T \otimes I) + (Q \otimes I) B^T P] (x - x^*) \\
+ (r - r^*)^T (PA + AT P) (r - r^*) \\
= -(x - x^*)^T R^T [F(x) - F(x*)] - (x - x^*)^T (Q \otimes I) \left[ \frac{D + D^T}{2} \right] (Q^T \otimes I) (x - x^*) \\
+ (r - r^*)^T (PA + AT P) (r - r^*) \\
= -(x - x^*)^T R^T [F(x) - F(x*)] - \frac{\lambda_d}{2} (x - x^*)^T L (x - x^*) \\
+ (r - r^*)^T (PA + AT P) (r - r^*)
\]

where \( \lambda_d \) is the minimum eigenvalue of \( D + D^T \), i.e., condition (1). The first line in the last equation is the same as equation (5.2) except that \( \lambda_d \) term is scaling the Laplacian part. The second term in the last line is negative definite if \( A \neq 0 \) and \( 0 \) if \( A = 0 \). Following the steps of any theorem in Chapter 5 with the appropriate assumptions, the first term is \( \leq 0 \) and the second term is \( < 0 \). Therefore, using LaSalle’s invariance principle the equilibrium \( \bar{x} = 1_n \otimes x^* \) is asymptotically stable.

Notice that the condition (2.c) and condition (3) are essentially the conditions for a LTI system to be passive by Positive Real Lemma (Lemma 6.2 [34]). The additional assumptions are used to ensure that the NE point doesn’t change. This suggests that using a passive system in the Laplacian dynamics, that also ensure the NE doesn’t change, will also lead to other viable dynamics. Let the storage function be \( V_1(x) = \frac{1}{2} \| x - x^* \| \) and \( f_x(x) = R^T F(x) \). Then solution along the trajectory of the feed forward path is,

\[
\dot{V}_1 = -(x - x^*)^T (f_x(x) - f_x(x*)) + (x - x^*)^T (u_x - u_x^*) \\
= -(x - x^*)^T (f_x(x) - f_x(x*)) + (y_x - y_x^*)^T (u_x - u_x^*)
\]

Similarly, let \( V_2(r) \) be the storage function of \( P_r \) such that \( V_2 \leq (y_r - y_r^*)^T (u_r - u_r^*) \). Creating the Lyapunov function \( V(x, r) = V_1(x) + V_2(r) \), the solution along the trajectory of the overall interconnected system is,

\[
\dot{V} \leq -(x - x^*)^T (f_x(x) - f_x(x*)) + (x - x^*)^T (u_x - u_x^*) + (y_r - y_r^*)^T (u_r - u_r^*) \\
\leq -(x - x^*)^T (f_x(x) - f_x(x*)) + (y_x - y_x^*)^T (u_x - u_x^*) + (y_r - y_r^*)^T [Q^T \otimes I_n] (y_x - y_x^*) \\
\leq -(x - x^*)^T (f_x(x) - f_x(x*)) + (x - x^*)^T (u_x - u_x^*) + (u_x - u_x^*)^T (y_x - y_x^*) \\
\leq -(x - x^*)^T (f_x(x) - f_x(x*))
\]

The last equation is exactly the same \( \dot{V} \) obtained in Chapter 5 and therefore the system can be shown to converge to the Nash equilibrium.
Remark 6.1. There is an additional assumption that is not explicitly stated, but is required for the system $P_r$ to be implementable. The dynamics must not have information directly shared between two agents that are not connected by an edge on the communication graph $G_c$. For example, if $A = B = C = 0$ and $D = L$ in the linear system, $P_r$, then the feedback to the gradient system will be $u = -LLx$. The matrix $LL$ will appear to have agents who are not connected to each other, being able to communicate. However, this matrix is showing that agents communicate two times over the network before updating their strategy. As long as $P_r$ can be broken down into a sequence of valid communications between agents, and the dynamics of an agent’s auxiliary variable $r$ is not known by others, then this would be a valid distributed algorithm.

Remark 6.2. The dynamical system $P_w$ in figure 6.3 is connected in negative feedback with the rest of the system. Just like the Laplacian, there can be passive dynamical systems that connect to the rest of the system. As long as this system doesn’t change the NE point and doesn’t involve direct communication between agents who are not connected on the communication graph $G_c$, then this method can also be used to construct new algorithms.

6.4 Higher-Order Gradient Dynamics

The gradient dynamics with perfect information, $\dot{x} = -F(x)$, can be represented as a bank of integrators with the gradient in negative feedback as depicted in figure 3.3. This is a first order dynamical system because of the integrator block. When decomposing the system into a static gradient feedback term and a dynamic feedforward system, what other feedforward dynamics can be used such that the overall system converges to the NE? The simplest extension would be to replace the integrator by some higher order LTI system with the hope of getting better performance. Figure 6.5 shows the new system where the integrator has been replaced by the LTI system $P_v$. The original system can be represented by $P_v$ is the matrices $A = D = 0$ and $B = C = I_n$. The system $P_v$ is a linear system and therefore has a corresponding transfer
function. Let the transfer function of the LTI system be denoted as \( G(s) \in \mathbb{R}^{n \times n} \),

\[
G(s) = \begin{bmatrix}
G_T(s) & 0 & \cdots & 0 \\
0 & \ddots & \cdots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & G_T(s)
\end{bmatrix}
\]

\[
G_T(s) = \frac{c_n s^{n-1} + c_{n-1} s^{n-2} + \cdots + c_2 s + c_1}{s^n + d_{n-1} s^{n-1} + \cdots + d_1 s + d_0}
\]

where the off diagonal elements of \( G(s) \) are zero and the diagonal elements are \( G_T(s) \). In the original system there was a bank of integrators, \( \frac{1}{s} I_n \), and now there is a bank of transfer functions, \( G_T(s)I_n \). Of course, each agent could use their own transfer function, \( G_i(s) \) instead of \( G_T(s) \). Additionally, agents with vector actions can have transfer functions on the off-diagonal elements. We will assume an identical transfer function along the diagonal for all action components to keep the analysis simple. The transfer function \( G_T(s) \) can also be converted into the controllable canonical form,

\[
\dot{v}^i = \begin{bmatrix}
0_{q-1} & I_{q-1} \\
-d_0 & -d
\end{bmatrix} v^i + \begin{bmatrix}
0_{q-1} \\
1
\end{bmatrix} u_i
\]

\[y_i = e^T v^i
\]

where \( d = [d_1, \ldots, d_{q-1}] \in \mathbb{R}^{1 \times q-1} \), \( e^T = [c_1, \ldots, c_q] \in \mathbb{R}^{1 \times q} \), \( v^i \in \mathbb{R}^q \) and \( q \) is the dimension of \( v^i \). The overall state, input and output vector is \( v = [(v^1)^T, \ldots, (v^n)^T]^T \), \( u_v = [(u_1)^T, \ldots, (u_n)^T]^T \) and \( y_v = [(y_1)^T, \ldots, (y_n)^T]^T \) respectively.

In this representation, the component of \( v^i \) corresponding to the action of an agent needs to be determined. When at the equilibrium \( v^i_1 = v^i_2 = \cdots = v^i_n = 0 \) and \( d_0 v^i_1 = u_i = -[F(y_v)]_i = -[F(c_1 v^x)]_i \) where \( v^x = [v^1_1, \ldots, v^n_1]^T \). In order for the equilibrium of the system to be equal to the NE, we require the constant \( d_0 = 0 \) and \( c_1 = 1 \). The resulting equation is \( 0 = -F(v^x) \) which is the NE condition. Therefore, the state variable \( v^i_1 \) is defined as the \( i \)th component of the action vector \( x \), i.e., \( [x]_i = \Delta v^i_1 \) and \( x = \Delta v^x \). Looking back at the transfer function \( G_T(s) \), we can see that when \( d_0 = 0 \) that the transfer function must contain an integrator. The constant \( c_1 = 1 \) states that the action \( [x]_i = \Delta v^i_1 \) must be in the output without being scaled.

The overall \( \dot{v} \) dynamics can be written compactly as,

\[
\dot{v} = \left( I_n \otimes \begin{bmatrix}
0_{q-1} & I_{q-1} \\
0 & -d
\end{bmatrix} \right) v - \left( I_n \otimes \begin{bmatrix}
0_{q-1} \\
1
\end{bmatrix} \right) F(e^T v)
\]

An alternative representation is created by applying the coordinate transform \( w_i^T = \begin{bmatrix} v^1_i & \cdots & v^n_i \end{bmatrix} \).
and \( w^T = \begin{bmatrix} w_1^T & \ldots & w_n^T \end{bmatrix} \). The overall system can be written as,

\[
\dot{w} = \left( \begin{bmatrix} 0_{q-1} & I_{q-1} \\ 0 & -d \end{bmatrix} \otimes I_n \right) w - \left( \begin{bmatrix} 0_{q-1} \\ 1 \end{bmatrix} \otimes I_n \right) F((c \otimes I_n) w)
\]

\[
= \begin{bmatrix} 0_{n(q-1) \times n} & I_{n(q-1)} \\ 0_{n \times n} & -d \otimes I_n \end{bmatrix} w - \begin{bmatrix} 0_{n(q-1) \times n} \\ I_n \end{bmatrix} F((c \otimes I_n) w)
\]

\[
= A_w w - B_w F((c \otimes I_n) w)
\]

The requirement that \( d_0 = 0 \) means that the transfer function \( \frac{1}{s + d_0} \) will not maintain the NE. Therefore, the next simplest transfer function is \( \frac{1}{s(s + d)} \). The example below shows that this transfer function can be used for the perfect communication case as long as \( d \) is selected appropriately.

**Example 6.3.** Consider a game \( G(I, J, \Omega) \) with perfect communication that satisfies Assumption 1(i) and 3(ii). Let the transfer function be \( G_T(s) = \frac{1}{s(s + d)} \) so that the overall interconnected systems is,

\[
\dot{v} = \left( I_n \otimes \begin{bmatrix} 0 & 1 \\ 0 & -d \end{bmatrix} \right) v - \left( I_n \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) F(v^*)
\]

Applying the coordinate transform \( w^T_i = \begin{bmatrix} v_1^i & \ldots & v_n^i \end{bmatrix} \) and \( w^T = \begin{bmatrix} w_1^T & \ldots & w_n^T \end{bmatrix} \) the overall system can be written as,

\[
\dot{w} = \left( \begin{bmatrix} 0 & 1 \\ 0 & -d \end{bmatrix} \otimes I_n \right) w - \left( \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes I_n \right) F(w_1)
\]

\[
= \begin{bmatrix} 0_{n \times n} & I_n \\ 0_{n \times n} & -dI_n \end{bmatrix} w - \begin{bmatrix} 0_{n \times n} \\ I_n \end{bmatrix} F(w_1)
\]

\[
= A_w w - B_w F(w_1)
\]

Let the candidate Lyapunov function be \( V(w) = (w - w^*)^T P (w - w^*) \) where \( P = \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix} \otimes I_n \), and \( P_i \in \mathbb{R} \). Along the solutions of (6.7),

\[
\dot{V}(w) = (w - w^*)^T [PA_w + A_w^T P] (w - w^*) - 2(w - w^*)^T PB_w (F(w_1) - F(w_1^*))
\]

\[
= (w - w^*)^T \begin{bmatrix} 0 & P_1 - dP_2 \\ P_1 - dP_2 & 2(P_2 - dP_3) \end{bmatrix} (w - w^*)
\]

\[
- 2 [(w_1 - w_1^*)^T P_2 + (w_2 - w_2^*)^T P_3] (F(w_1) - F(w_1^*))
\]
From Assumption 3(ii), the Lipschitz and strongly monotone condition bounds $\dot{V}$,

$$
\dot{V}(w) \leq z^T \begin{bmatrix}
-2P_2\mu & \theta P_3 + \|P_1 - dP_2\| \\
* & 2(P_2 - dP_3)
\end{bmatrix} z
$$

$$z = \begin{bmatrix}
\|w_1 - w_1^*\| \\
\|w_2 - w_2^*\|
\end{bmatrix}^T$$

Let $P_1 = K_1P_2 = dP_2$ and $P_3 = K_3P_2$ then the matrix is negative definite if

$$[-4\mu(1 - dK_3) - \theta^2 K_3^2]P_2^2 > 0$$

The term in front of $P_2^2$ is a quadratic in $K_3$ and the maximum occurs when

$$4\mu d - 2\theta^2 K_3 = 0$$

$$K_3 = \frac{2\mu d}{\theta^2}$$

Substituting this value of $K_3$ into the inequality gives

$$-4\mu + 4\mu d \left( \frac{2\mu d}{\theta^2} \right) - \theta^2 \left( \frac{2\mu d}{\theta^2} \right)^2 > 0$$

$$\frac{4\mu (\mu d^2 - \theta^2)}{\theta^2} > 0$$

$$d^2 > \frac{\theta^2}{\mu}$$

Therefore, if $d^2 > \frac{\theta^2}{\mu}$ then $\dot{V} < 0$. For the matrix $P$ to be positive definite, $P_1P_3$ must be greater than $P_2^2$. Substituting the values we obtained for $P_i$, the condition for $P$ to be positive definite is $\frac{2\mu d^2}{\theta^2} > 1$ which is satisfied if $d^2 > \frac{\theta^2}{\mu}$.

Curiously, the condition for the LTI system to converge to the NE looks similar to the imperfect information case over a communication network. This suggests that the factor $\frac{\theta^2}{\mu}$ is strongly related to any game dynamics and not just games with communication.

**Remark 6.3.** Consider the case where the gradient dynamics in (6.7) were used on a distributed optimization problem,

$$\min_x J(x)$$

and each agent solves the sub problem,

$$\min_{x_i} J(x_i, x_{-i})$$

Agent $i$ would update $x_i$ by using the gradient and share $x_i$ with all the other agents. In this case the Lyapunov candidate $V(w) = (w - w^*)^T P(w - w^*) + 2P_3[J(w_1) - J(w_1^*)]$ can be used. Notice that $\frac{\partial}{\partial w_1} J(w_1) = F(w_1)\dot{w}_1 = F(w_1)w_2$ and this term will cancel out the coupling term.
from \((w - w^*)^T P(w - w^*)\). Therefore, selecting any \(P_2 > 0\) and setting \(P_1 = dP_2\), there will always be a \(P_3\) sufficiently large, such that \(P\) is positive definite and

\[
\dot{V}(x) \leq -2P_2(x_1 - x_1^*)(F(x_1) - F(x_1^*)) + 2(P_2 - dP_3) \|x_2 - x_2^*\|^2 \leq 0
\]

The above example suggests that as long as \(d\) is selected appropriately, putting any polynomial on the denominator and having the numerator equal to one in the transfer function will work. The example below shows that this is not necessarily the case.

**Example 6.4.** Consider a game \(G(I, J, \Omega_i)\) with perfect communication that satisfies Assumption 1(i) and 3(ii). Let the transfer function be \(G_T(s) = \frac{1}{s^3 + d_2s^2 + d_1s}\) where \(d_i > 0\) and the poles are in the open left half plane. Let the Lyapunov function be \(V(x) = (x - x^*)^T P(x - x^*)\) where,

\[
P = \begin{bmatrix}
P_1 & P_2 & P_4 \\
P_2 & P_3 & P_5 \\
P_4 & P_5 & P_6
\end{bmatrix} \otimes I_n
\]

then following the same steps as in the previous example, the bound on \(\dot{V}\) is,

\[
\dot{V}(x) \leq z^T \begin{bmatrix}
-2P_4\mu & \theta P_5 + \|P_1 - d_1P_4\| & \theta P_6 + \|P_2 - d_2P_4\| \\
* & 2(P_2 - d_1P_5) & P_3 + P_4 - d_2P_3 - d_1P_6 \\
* & * & 2(P_5 - d_2P_6)
\end{bmatrix} z
\]

\[
z = [\|w_1 - w_1^*\| \|w_2 - w_2^*\| \|w_3 - w_3^*\|]^T
\]

Let all \(P_i = K_iP_4\) and set \(P_5 = 0\), \(P_1 = d_1P_4\) and \(K_3 = d_1K_6 - 1 > 0\). The matrix becomes,

\[
\dot{V}(x) \leq z^T \begin{bmatrix}
-2P_4\mu & 0 & \theta P_6 - P_2 + d_2P_4 \\
* & 2P_2 & 0 \\
* & * & -2d_2P_6
\end{bmatrix} z
\]

where \(P_2 < 0\) and \(P_4 > 0\). The matrix is negative definite if,

\[
4\mu d_2 K_6 P_4^2 > (\theta K_6 - K_2 + d_2)^2 P_4^2
\]

The inequality is a quadratic in \(K_6\) and the max occurs at,

\[
4\mu d_2 - 2\theta(\theta K_6 - K_2 + d_2) = 0
\]

\[
K_6 = \frac{d_2(2\mu - \theta) + \theta K_2}{\theta^2}
\]

substituting back into the inequality gives,

\[
4\mu d_2 \frac{d_2(2\mu - \theta) + \theta K_2}{\theta^2} > \left(\frac{2\mu d_2}{\theta}\right)^2
\]
\[
8\mu^2d_2^2 - 4\mu d_2^2\theta + 4\mu d_2\theta K_2 > 4\mu^2d_2^2 \\
4\mu d_2(\mu d_2 - d_2\theta + \theta K_2) > 0
\]

This requires that \(\mu > \theta\) which is impossible because \(\theta\) is an upper bound and \(\mu\) is a lower bound.

One way to understand why the conditions are worse than the first example is by moving the \(\frac{1}{s}\) term in the LTI dynamics into the gradient term. Figure 6.6 shows the transfer function

![Block Diagram LTI gradient dynamics](image)

\[G(s) = G_r(s)[\frac{1}{s} \otimes I_n]\] being split into an integrator and a reduced block, \(G_r(s)\). The gradient dynamics in the feedback are passive because \(F(x)\) is assumed to be monotone. The reduced block \(G_r(s)\) is still a linear system just like before. In the previous example \(G_{Tr}(s) = \frac{1}{s^2 + d_2s + d_1}\) is a linear system of relative degree 2 and therefore, this system is not passive. Just because the linear system isn’t passive doesn’t mean it won’t converge but it does show that the analysis is much more difficult because it no longer has a passivity property.

### 6.4.1 Second-Order LTI System

The simple transfer function \(G_T(s) = \frac{1}{s(s+d)}\) works for the perfect information case, but has a requirement that involves the strong monotone constant \(\mu\) and the Lispchitz constant. Increasing the denominator to a second order polynomial didn’t improve the requirements but actually made it worse. Therefore, adding terms in the numerator is considered next.

**Example 6.5.** Consider a game \(G(I, J, \Omega_i)\) with perfect communication that satisfies Assumption 1(i). Let the transfer function for the LTI system be given as \(G_T(s) = \frac{cs+1}{s(s+d)}\). The state space equation for this transfer function is,

\[
\begin{align*}
\dot{v}^i &= \begin{bmatrix} 0 & I \\ 0 & -d \end{bmatrix} v^i + \begin{bmatrix} 0 \\ I \end{bmatrix} u_i \\
y_i &= \begin{bmatrix} 1 & c \end{bmatrix} v^i
\end{align*}
\]
and with the feedback, \( u = -F(y) \), the overall dynamics are,
\[
\dot{w} = \begin{bmatrix} 0_{n \times n} & I_n \\ 0_{n \times n} & -dI_n \end{bmatrix} w - \begin{bmatrix} 0_{n \times n} \\ I_n \end{bmatrix} F(w_1 + cw_2) \tag{6.8}
\]
\[= A_w w - B_w F(w_1)\]

Let the candidate Lyapunov function be 
\[V(w) = (w - w^{*})^T P (w - w^{*}) \] where 
\[P = \begin{bmatrix} d & 1 \\ 1 & c \end{bmatrix} \otimes I_n, \]
then along the solutions of 6.8,
\[
\dot{V}(w) = (w - w^{*})^T P \left( \begin{bmatrix} 0 & I \\ 0 & -dI \end{bmatrix} (w - w^{*}) - \begin{bmatrix} 0 \\ I \end{bmatrix} (F(w_1 + cw_2) - F(w_1^{*})) \right)
+ \left( \begin{bmatrix} 0 & I \\ 0 & -dI \end{bmatrix} (w - w^{*}) - \begin{bmatrix} 0 \\ I \end{bmatrix} (F(w_1 + cw_2) - F(w_1^{*})) \right)^T P (w - w^{*})
= (w - w^{*})^T \begin{bmatrix} 0 & 0 \\ 0 & (1-cd)I \end{bmatrix} (w - w^{*}) - (w - w^{*})^T \begin{bmatrix} I \\ cI \end{bmatrix} (F(w_1 + cw_2) - F(w_1^{*}))
= (1-cd) \|w_2 - w_2^{*}\|^2 - (w_1 + cw_2 - w_1^{*})^T (F(w_1 + cw_2) - F(w_1^{*}))
\]

If \( c > 0, d > 0 \) and \( cd > 1 \) then the matrix \( P \) is positive definite. The first term in the last line will be negative definite and the second term is negative semi-definite because of the monotonicity condition on the pseudo-gradient and therefore, \( \dot{V}(x) \leq 0 \). When \( \dot{V}(w) = 0 \) then \( w_2 = w_2^{*} = 0 \) and the second term of \( \dot{V} \) becomes \( (w_1 - w_1^{*})^T (F(w_1) - F(w_1^{*})) = 0 \). From LaSalles invariance principle, the largest invariant set is \( w_2 = 0 \) and \( F(w_1) = 0 \) and therefore, these dynamics will converge to the NE.

These dynamics have better conditions than the transfer function \( G_T(s) = \frac{1}{s(s+d)} \) because it does not require the constants \( c \) and \( d \) to be a function of \( \theta \) or \( \mu \). The dynamics work for the perfect communication setting and require mild conditions on the constants. Therefore, this system has good potential of working for the partial information case and is considered in the next section.

### 6.4.2 Second-Order Gradient-Play dynamics with Estimate Correction

The previous section showed that under mild conditions of the parameters, \( c \) and \( d \), the gradient dynamics with perfection information will converge. In this section we show convergence for the incomplete information case over a connected communication graph.

**Theorem 6.6.** Consider a game \( \mathcal{G}(I, J, \Omega_i) \) over a communication graph \( G_c \) under Assumptions \#(i)\#, \#(ii)\# and \#(ii)\#. Let each player’s dynamics \( \bar{P}_i \) be,
\[
\dot{\bar{v}}_i = \begin{bmatrix} \bar{v}_1^i \\ \bar{v}_2^i \end{bmatrix} = \begin{bmatrix} 0_{n \times n} & I_n \\ 0_{n \times n} & -dI_n \end{bmatrix} \begin{bmatrix} \bar{v}_1^i \\ \bar{v}_2^i \end{bmatrix}
\]
then any solution of (6.9) converges asymptotically to
\[ \bar{w} \]
The solution along the trajectory of (6.9) is,\[ \dot{\bar{w}} = \begin{bmatrix} 0_{n \times n} & I_{Nn} \\ 0_{Nn \times Nn} & -dI_{Nn} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \begin{bmatrix} 0_{Nn \times Nn} \\ I_{Nn} \end{bmatrix} [R^T F(w_1 + cw_2) + L(w_1 + cw_2)] \]
where \( c > 0, d > 0 \) and \( v_i = [(v_i^1)^T, (v_i^2)^T]^T \in \mathbb{R}^{2n} \). The vector \( v_i^1 \in \mathbb{R}^n \) is agent \( i \)'s estimate vector and \( v_i^2 \in \mathbb{R}^n \) is the auxiliary variable. Agent \( i \)'s actual action is \( R_i v_i^1 \) and the estimate of the other agents actions is \( S v_i^1 \). Alternatively, the overall \( \bar{P} \) dynamics are,
\[ \dot{w} = \begin{bmatrix} 0_{Nn \times Nn} & I_{Nn} \\ 0_{Nn \times Nn} & -dI_{Nn} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} - \begin{bmatrix} 0_{Nn \times Nn} \\ I_{Nn} \end{bmatrix} [R^T F(w_1 + cw_2) + L(w_1 + cw_2)] \]
where \( w = [w_1^T, w_2^T]^T \in \mathbb{R}^{2Nn} \). The vector \( w_1 = [(v_1^1)^T, \ldots, (v_N^1)^T]^T \in \mathbb{R}^{Nn} \) is the stacked estimate vector of all agents and \( w_2 = [(v_1^2)^T, \ldots, (v_N^2)^T]^T \in \mathbb{R}^{Nn} \) is the auxiliary variable. If \( \lambda_2(L) > \frac{\theta^2}{\mu} + \gamma > 0 \) and,
\[ d > \frac{c[(\lambda_n + 2\theta)^2 - \gamma^2]}{\gamma^2} \]
then any solution of (6.9) converges asymptotically to \( \bar{w}_1 = 1_N \otimes x^* \), which implies that the action components, \( R w_1 \), converge to the NE of the game, \( x^* \).

**Proof.** Let the Lyapunov function be given as \( V(w) = \frac{1}{2}(w - \bar{w})^T P(w - \bar{w}) \) where \( P = \begin{bmatrix} d & 1 \\ 1 & c \end{bmatrix} \otimes I_{Nn} \), or more explicitly,
\[ V(w) = \frac{d}{2}(w_1 - \bar{w}_1)^T(w_1 - \bar{w}_1) + (w_1 - \bar{w}_1)^T(w_2 - \bar{w}_2) + \frac{c}{2}(w_2 - \bar{w}_2)^T(w_2 - \bar{w}_2) \]
The solution along the trajectory of (6.9) is,
\[ \dot{V}(w) = d(w_1 - \bar{w}_1)^T(w_2 - \bar{w}_2) + (w_2 - \bar{w}_2)^T(w_2 - \bar{w}_2) + d(w_1 - \bar{w}_1)^T(w_2 - \bar{w}_2) \\
- (w_1 - \bar{w}_1)^T R^T[F(w_1 + cw_2) - \bar{F}(w_1 + cw_2)] \\
- (w_1 - \bar{w}_1)^T L[(w_1 - \bar{w}_1) + c(w_2 - \bar{w}_2)] \\
- c (w_2 - \bar{w}_2)^T R^T[F(w_1 + cw_2) - \bar{F}(w_1 + cw_2)] \\
- c (w_2 - \bar{w}_2)^T L[(w_1 - \bar{w}_1) + c(w_2 - \bar{w}_2)] \\
- cd (w_2 - \bar{w}_2)^T (w_2 - \bar{w}_2) \\
= (1 - dc)(w_2 - \bar{w}_2)^T (w_2 - \bar{w}_2) \\
- [(w_1 - \bar{w}_1) + c(w_2 - \bar{w}_2)]^T L[(w_1 - \bar{w}_1) + c(w_2 - \bar{w}_2)] \\
- ((w_1 - \bar{w}_1) + c(w_2 - \bar{w}_2))^T R^T[F(w_1 + cw_2) - \bar{F}(w_1 + cw_2)] \]

(6.10)
Let the two state variables $w_1$ and $w_2$ be decomposed into parallel and perpendicular components as in Theorem 5.3, i.e., $w_1 = w_1^1 + \bar{w}_1$ and $w_2 = w_2^1 + \bar{w}_2$. Let $z$ be defined as,

$$z \triangleq \left[ \|w_1^1 - \bar{w}_1\|, \|w_1^{[\|]} - \bar{w}_1^{[\|]}\|, \|w_2^1 - \bar{w}_2\|, \|w_2^{[\|]} - \bar{w}_2^{[\|]}\| \right]^T$$

then the first term, $T_1(w) = (1 - dc)(w_2 - \bar{w}_2)^T(w_2 - \bar{w}_2)$ in equation (6.10) is bounded by,

$$T_1(w) = (1 - dc)(w_2^1 - \bar{w}_2^1)^T(w_2^1 - \bar{w}_2^1) + (1 - dc)(w_2^{[\|]} - \bar{w}_2^{[\|]})(w_2^{[\|]} - \bar{w}_2^{[\|]})$$

$$\leq -z^T \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & dc - 1 & 0 \\ 0 & 0 & dc - 1 \end{bmatrix} z$$

where the minus sign is brought in front of the matrix. The second term in equation (6.10), $T_2(w) = -[(w_1 - \bar{w}_1) + c(w_2 - \bar{w}_2)]^T \mathbf{L}[(w_1 - \bar{w}_1) + c(w_2 - \bar{w}_2)]$ is bounded by,

$$T_2(w) = -[(w_1^1 - \bar{w}_1^1) + c(w_2^1 - \bar{w}_2^1)]^T \mathbf{L}[(w_1^1 - \bar{w}_1^1) + c(w_2^1 - \bar{w}_2^1)]$$

$$\leq -z^T \begin{bmatrix} \lambda_2 & 0 & -c\lambda_n & 0 \\ 0 & 0 & 0 & 0 \\ -c\lambda_n & 0 & c^2\lambda_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} z$$

and the third term $T_3(w) = -[(w_1 - \bar{w}_1) + c(w_2 - \bar{w}_2)]^T \mathbf{R}^T \mathbf{R}^T [(w_1 + cw_2) - F(\bar{w}_1 + cw_2)]$ in equation (6.10) is equal to,

$$T_3(w) = -[(w_1 - \bar{w}_1)^T \mathbf{R}^T [F(w_1 + cw_2) - F(\bar{w}_1 + cw_2)]]$$

$$- c[w_2 - \bar{w}_2]^T \mathbf{R}^T [F(w_1 + cw_2) - F(\bar{w}_1 + cw_2)]$$

$$= -[F(w_1 + cw_2) - F(\bar{w}_1 + cw_2)]^T \Psi(w) - (w_1^{[\|]} - \bar{w}_1^{[\|]})^T \Psi(w)$$

$$= -[F(w_1 + cw_2) - F(\bar{w}_1 + cw_2)]^T \Psi(w) - (w_2^{[\|]} - \bar{w}_2^{[\|]})^T \Psi(w)$$

where

$$\Psi(w) \triangleq \mathbf{R}^T [F(w_1 + cw_2) - F(\bar{w}_1 + cw_2)]$$

$$= \mathbf{R}^T [F(w_1^{[\|]} + cw_2^{[\|]} + w_1^1 + cw_2^1) - F(w_1^{[\|]} + cw_2^{[\|]} + w_1^1)]$$

$$+ \mathbf{R}^T [F(w_1^{[\|]} + cw_2^{[\|]} + w_1^1 + cw_2^1) - F(w_1^{[\|]} + cw_2^{[\|]} + w_1^1)]$$

$$+ \mathbf{R}^T [F(w_1^{[\|]} + cw_2^{[\|]} - F(w_1^{[\|]} + cw_2^{[\|]} + w_1^1)]$$

$$+ \mathbf{R}^T [F(w_1^{[\|]} + cw_2^{[\|]} - F(w_1^{[\|]} + cw_2^{[\|]} + w_1^1)]$$

By writing $\mathbf{R}^T [F(w_1 + cw_2) - F(\bar{w}_1 + cw_2)]$ into component changes, as done in $\Psi(w)$, we can use the same technique as in Theorem 5.3. By applying monotonicity and Lipschitz condition,
Chapter 6. Modifying Augmented Gradient Dynamics with Estimate Correction

\( T_3(w) \) can be bounded by,

\[
T_3(w) \leq -z^T \begin{bmatrix}
-\theta & -\theta & -c\theta & -c\theta \\
-\theta & \mu & -c\theta & -c\theta \\
-c\theta & -c\theta & -c^2\theta & -c^2\theta \\
-c\theta & -c\theta & -c^2\theta & c^2\mu
\end{bmatrix} z
\]

Therefore, \( V(w) \leq T_1(w) + T_2(w) + T_3(w) \) is bounded by,

\[
\dot{V}(x, r) \leq -z^T \begin{bmatrix}
\lambda_2 - \theta & -\theta & -c(\lambda_n + \theta) & -c\theta \\
-\theta & \mu & -c\theta & -c\theta \\
-c(\lambda_n + \theta) & -c\theta & c^2(\lambda_2 - \theta) + (cd - 1) & c^2\theta \\
-c\theta & -c\theta & c^2\theta & c^2\mu + (cd - 1)
\end{bmatrix} z
\]

\[
= z^T \begin{bmatrix}
\Theta_1 & \Theta_2 \\
\Theta_2 & \Theta_3
\end{bmatrix} z = z^T \Theta z
\]

In order for the matrix \( \Theta \) to be positive definite, the Schur Complement of \( \Theta_1 \) in \( \Theta \) must be positive definite as well as \( \Theta_1 \). The matrix \( \Theta_1 \) is positive definite if \( \mu(\lambda_2 - \theta) > \theta_2^2 \), which is true from the assumption \( \lambda_2 > \frac{\theta_2^2}{\mu} + \theta \). The Schur Complement of \( \Theta_1 \) in \( \Theta \) is,

\[
\Theta_3 - \Theta_2 \Theta_1^{-1} \Theta_2 > 0
\]

Notice that \( \Theta_3 = c^2 \Theta_1 + (dc - 1)I \) and only \( \Theta_3 \) has the constant \( d \) in it. Therefore, the matrix inequality becomes,

\[
(dc - 1)I + c^2 \Theta_1 - \Theta_2 \Theta_1^{-1} \Theta_2 > 0
\]

It is obvious that the value of \( d \) can be increased until this inequality is satisfied, therefore there is always a \( c \) and \( d \) that ensures convergence to the Nash equilibrium.

To derive a bound for \( d \), let the largest magnitude eigenvalue of \( -\frac{1}{c} \Theta_2 \) be determined by the characteristic polynomial,

\[
\det \left( \phi I - \frac{1}{c} \Theta_2 \right) = \det \left( \begin{bmatrix}
\phi - \lambda_n - \theta & -\theta \\
-\theta & \phi - \theta
\end{bmatrix} \right)
\]

\[
= (\phi - \lambda_n - \theta)(\phi - \theta) - \theta^2
\]

\[
= \phi^2 - (\lambda_n + 2\theta)\phi + \theta^2 + \lambda_n\theta
\]

\[
= \phi^2 - (\lambda_n + 2\theta)\phi + \lambda_n\theta
\]

\[
\phi_{\text{large}} = \frac{1}{2} \left[ (\lambda_n + 2\theta) + \sqrt{(\lambda_n + 2\theta)^2 - 4(\lambda_n\theta)} \right]
\]

\[
\leq \frac{1}{2} \left[ (\lambda_n + 2\theta) + \sqrt{(\lambda_n + 2\theta)^2} \right]
\]

\[
\leq \lambda_n + 2\theta
\]
Therefore, the largest magnitude eigenvalue of $\Theta_2$ is less than $c(\lambda_n + 2\theta)$. Let $\gamma = \min \text{eig}(\Theta_1)$ which has the close form solution,

$$\gamma = \frac{\lambda_2 + \mu - \theta}{2} - \sqrt{\frac{(\lambda_2 - \mu - \theta)^2}{4} + \theta^2}$$

The Schur Complement condition can be satisfied if,

$$(cd - 1) + c^2 \gamma - \frac{c^2(\lambda_n + 2\theta)^2}{\gamma} > 0$$

and a bound for the constant $d$ is,

$$d > \frac{c \left[ (\lambda_n + 2\theta)^2 - \gamma^2 \right]}{\gamma} + \frac{1}{c}$$

Using this selected value for $d$ the Lyapunov function, $\dot{V}(w)$ is negative semi-definite. Therefore, using LaSalle’s invariance principle the NE equilibrium is asymptotically stable.

The convergence of the second order gradient consensus dynamics followed similarly to the perfect information case, but with additional assumptions on $c$ and $d$ related to the graph connectivity, the Lipschitz constant, and monotonicity condition. This suggests that if higher order gradient dynamics for the perfect information case works, then it will also work for the imperfect information case. All passive systems $P_v$ can be used and convergence will hold as long as the NE point isn’t change. Additional conditions on the parameters $c$ and $d$ will need to be set so that the algorithm can converge in the imperfect information setting.

**Remark 6.4.** From an agent’s perspective the dynamics (6.9) can be interpreted as a gradient method with prediction. The term $w_2$ is the derivative of the state vector $w_1$. Therefore, $w_1 + cw_2$ is the current state $w_1$, plus a term that predicts that the state variable will continue to move in the direction of $cw_2 = cw_1$. The parameter $c$ controls how large of a step to make in the predicted direction. The parameter $d$ controls how much to weight the past direction of $-RTF(w_1 + cw_2) - L(w_1 + cw_2)$. A small $d$ means there is a high weight on past directions and large $d$ means high weight on the recent $-RTF(w_1 + cw_2) - L(w_1 + cw_2)$ direction.

**Remark 6.5.** The parameters $c$ and $d$ have to satisfy an inequality that is dependent on $\lambda_2$, $\lambda_N$, $\theta$ and $\mu$. Either this information available because of conditions on the game, or agents need to communicate to each other to determine these parameters. Additionally, agents could select their own $c_i$ and $d_i$, and the proof will require changing $c$ to a matrix $C = \text{diag}(c_1, \ldots, c_N)$. A similar proof to the one above can be made to prove convergence.

At first glance, the higher order dynamics do not appear to be that useful because the condition $\lambda_2 > \frac{\theta^2}{\mu} + \theta$ is the same condition that the first order system had. The convergence conditions don’t appear to have improved, so why would a higher order system be desirable? The purpose for introducing these additional dynamics is to show that more complex dynamics
can be introduced that still work. These additional dynamics can be used to meet other design objectives. For example, the second order method mentioned above will have some effect on reducing noise. This noise can be introduced by the communication network, so the second order dynamics ensures that agents don’t sporadically change their actions all of the time because of the noise.

### 6.4.3 Dynamic Game Interpretation

The higher order dynamics also have a different interpretation and is not used to improve performance but to represent dynamic games. Let $p \in \mathbb{R}^N$ be the cost vector and $p_i$ represents the cost received by player $i$. Originally the cost that players received for action profile $x$ is $p = J(x)$. If the cost received is via a dynamical system is $\dot{p} = J(x) - \lambda p$, then the payoff is constantly changing. The dynamics for the original gradient system, with complete information, can be written as $\dot{x} = -F(x) = -\nabla_p p$ where $\nabla_p p = [(\partial p/\partial x_1)^T, \ldots, (\partial p/\partial x_N)^T]^T$. Differentiating $\dot{x}$ again gives $\ddot{x} = -\nabla_p \dot{p} = -\nabla_p (J(x) - \lambda p) = -F(x) - \lambda \dot{x}$. The state dynamics for the dynamic cost function is a second order system and can we written as,

$$
\begin{bmatrix}
\dot{x} \\
\ddot{x}
\end{bmatrix} =
\begin{bmatrix}
0 & I \\
0 & -\lambda I
\end{bmatrix}
\begin{bmatrix}
x \\
\dot{x}
\end{bmatrix} -
\begin{bmatrix}
0 \\
\lambda
\end{bmatrix}
[F(x)] \quad (6.11)
$$

This equation is the same as the second order dynamics generated from the LTI system $G_T(s) = \frac{1}{s(s+\lambda)}$. Therefore, changing the LTI system $G_T(s) = \frac{1}{s}$ to some other transfer function can be interpreted as transforming a static game into a higher order dynamic game. This means the bounds determined for the constants $c$ and $d$ actually give a bound on the type of dynamical second order games the gradient dynamics with imperfect information can converge in. This also means that if given a dynamical LTI game, the agent can design $P_v$ to change the pole/zero placement of the game’s dynamics to a range where the gradient dynamics can converge.
Chapter 7

Simulation Results

In this chapter, we present a set of examples showing that the dynamics mentioned throughout this thesis do converge to the Nash equilibrium under various cost functions, connectivity conditions of $G_c$, and parameters, i.e., $\epsilon$. In section 7.1 dynamics (4.30) are shown to converge for three different examples:

1. Example 7.1. Assumption 3(i) and 4(i) hold.

2. Example 7.2. Assumption 3(ii) and 4(ii) hold locally. Additionally, the gain $1/\epsilon$ is used to ensure convergence and improve convergence speed.

3. Example 7.3. Assumption 3(ii) and 4(ii) hold locally. Additionally, the gain $1/\epsilon$ is used to ensure convergence and improve convergence speed.

In section 7.2 the projection dynamics (4.36) are simulated on a compact action set on 4 examples:

1. Example 7.4. Assumption 3(i) and 4(i) hold.

2. Example 7.5. Assumption 3(ii) and 4(ii) hold.

3. Example 7.6. Assumption 3(ii) and 4(ii) hold.

4. Example 7.7 where the NE is on the boundary, and Assumption 3(ii) and 4(ii) hold.

In section 7.3 the static Laplacian feedback is compared to a dynamic Laplacian. In section 7.4 the second order dynamics are compared to (4.30) when subjected to sinusoidal noise. The higher order dynamics is better at filtering out noise when compared to (4.30). Lastly, in section 7.5 the gradient dynamics with estimate correction (4.30)/(4.33) is compared to (5.15), i.e., (4.30) with the $-\mathbf{R}[\mathbf{R}^T \mathbf{x} + \mathbf{S}^T \mathbf{z}]$ term removed. Additionally, our dynamics are compared to a recent continuous time algorithm [2].
Chapter 7. Simulation Results

7.1 Unconstrained Action Set and Dynamics

Example 7.1. Consider a N-player quadratic game from economics, where 20 firms are involved in the production of a homogeneous commodity. The quantity produced by firm $i$ is denoted by $x_i$. The overall cost function of firm $i$ is $J_i(x_i, x_{-i}) = c_i(x_i) - x_i f(x)$, where $c_i(x_i) = [20 + 10(i - 1)]x_i$ is the production cost and $f(x) = 2200 - \sum_{i \in \mathcal{I}} x_i$ is the demand price, as in [19]. We investigate the proposed dynamics (4.32) over a communication graph $G_c$ via simulation. The initial condition for each estimate component is selected randomly from $[0, 20]$. Assumption 3(i) and 4(i) hold, so by Theorem 5.1 the dynamics (4.32) will converge even over a minimally connected graph. Figures 7.1 and 7.2 show the convergence of (4.32) over a randomly generated communication graph $G_c$ (Fig. 7.3) and over a cycle $G_c$ graph (Fig. 7.4), respectively.

Next we provide, for reference, the results for running the gradient dynamics in the perfect information case (Fig. 7.5), and the augmented gradient with estimate correction over a com-
Chapter 7. Simulation Results

Notice that (4.32), or (5.18) with $\epsilon = 1$, is able to converge to a NE with a sparsely connection communication graph. In fact, Assumption 3(ii) holds so we can apply Theorem 5.13 and use (5.18), with $\epsilon > 0$ to speed up convergence. Running (5.18) with the gain term $\epsilon = 10$ the convergence speed drastically improves for the random graph and cyclic graph as shown in Figure 7.7, Figure 7.8.

Remark 7.1. The $1/\epsilon$ term describes how much faster the estimate dynamics is compared to the gradient. If this algorithm is converted into a discrete time algorithm then the $1/\epsilon$ term
describes how many iterations of estimate information sharing is done before doing the gradient. Generally, as $1/\epsilon$ increases the number of iterations will also increase. In exchange for doing more iterations of estimate exchange the information that each agent has is closer to the actual action, the perfect information case, and therefore improves convergence for weakly connected graphs.

**Example 7.2.** Consider a second example of an 8 player game with $J_i(x_i, x_{-i}) = c_i(x_i) - x_i f(x)$, $c_i(x_i) = (10 + 4(i - 1)) x_i$, and $f(x) = 600 - \sum_{i \in \mathcal{I}} x_i^2$, as in [17]. The initial conditions are selected randomly from $[0, 20]$. Here Assumption 4(i) on $\mathbf{F}$ does not hold globally, so we cannot apply Theorem 5.1, but Assumption 4(ii) holds locally. By Theorem 5.3, (4.32) will converge depending on $\lambda_2(L)$. Figure 7.11 shows the convergence of (4.32) over a sufficiently connected, randomly generated communication graph $G_c$ as depicted in Fig. 7.9. However, on a cyclic graph $G_c$, (4.32) does not converge. Alternatively, by Theorem 5.13, a high $1/\epsilon$ gain (time-scale decomposition) can balance the connectivity loss. Fig. 7.12 shows convergence for (5.18) with $1/\epsilon = 200$, over a cycle $G_c$ graph as shown in Fig. 7.10.

![Figure 7.9: Random $G_c$, $\lambda_2 = 1.67$](image1.png)

![Figure 7.10: Cycle $G_c$ Graph](image2.png)

Both figure 7.11 and figure 7.12 have comparable convergence speed but the cyclic graph requires a high $1/\epsilon$ gain because of the connectivity of the graph. The difference in the connectivity requirement in this example compared to the previous example, is due to the difference in coupling strength in the cost function. This example has the gradient dependent on the square of the agents actions but the last example is linear.
Next we provide for reference the results for running the gradient dynamics in the perfect information case (Fig. 7.13), and the augmented gradient with estimate correction, (5.18) for $1/\epsilon = 200$, over a complete $G_c$, (Fig. 7.14). The convergence speed of perfect information case is comparable to a randomly generated graph. If the graph is not well connected then the results are comparable if the gain $1/\epsilon$ is high.

Figure 7.11: (4.32) over random $G_c$  
Figure 7.12: (5.18) cycle $G_c$, $1/\epsilon = 200$

Figure 7.13: (3.5) perfect information  
Figure 7.14: (5.18) complete $G_c$, $1/\epsilon = 200$
Example 7.3. Consider an 8 player game where $J_i(x_i, x_{-i}) = c_i(x_i) - x_i f_i(x)$, $c_i(x_i) = (10 + 4(i - 1))x_i$, and $f_i(x) = x_i (600 - \sum_{i \in \mathcal{I}} x_i^2)$, where the initial conditions are selected randomly from $[0, 20]$. Here Assumption 4(i) does not hold, but Assumption 4(ii) holds locally. By Theorem 5.3, (4.32) will converge depending on $\lambda_2(L)$. Figure 7.17 shows the convergence of (4.32) over a randomly generated, sufficiently connected, communication graph $G_c$ as in Fig. 7.15. Over a cycle $G_c$ graph, (4.32) does not converge. Alternatively, by Theorem 5.13, a high $1/\epsilon$ gain (time-scale decomposition) can balance the connectivity loss. In fact running (5.18) with $1/\epsilon = 20$ over a cycle $G_c$ graph (Fig. 7.16) gives better performance, and is shown in Fig. 7.18.

Figure 7.15: Random $G_c$, $\lambda_2 = 2$

Figure 7.16: Cycle $G_c$ Graph

Figure 7.17: (4.32) over random $G_c$

Figure 7.18: (5.18) cycle $G_c$, $1/\epsilon = 20$

Next we provide for reference, the results for running the gradient dynamics in the perfect information case (Fig. 7.19), and the augmented gradient with estimate correction, (4.32), over
a complete \( G_c \), (Fig. 7.20). Notice that the perfect information case is significantly faster than the random \( G_c \) case. However, figure 7.18 suggests that using a higher gain \( 1/\epsilon \) for a random \( G_c \) will give comparable results.

![Figure 7.19: (3.5) perfect information](image)

![Figure 7.20: (4.32) over complete \( G_c \)](image)

### 7.2 Compact Action Set and Projected Dynamics

In this section we go through the examples of the previous section except we constraint the action space to a compact set. Additionally, there is an extra example showing the projected graph method converging when the NE is on the boundary of the action set.

**Example 7.4.** Let the cost function be defined in Example 7.1 and this time \( \Omega_i = [0, 200] \). We investigate the projected augmented gradient dynamics (4.38) over a graph \( G_c \). The initial conditions for the action components are selected randomly from \([0, 200]\), while the estimate component is selected from \([-2000, 2000]\). Assumption (i) and (ii) hold, so by Theorem 5.2 the dynamics (4.38) will converge even over a minimally connected graph. Figures 7.23 and 7.24 show the convergence of (4.38) over a randomly generated communication graph \( G_c \) (Fig. 7.21) and over a cycle \( G_c \) graph (Fig. 7.22), respectively.
Next we provide for reference, the results for running the projected gradient dynamics in the perfect information case (Fig. 7.25), and the projected augmented gradient with estimate correction over a complete $G_c$, (Fig. 7.26). The perfect information case is significantly faster than the imperfect information case. The reason for this difference is because the estimates are very far away from the initial actions. In some cases the action set of all the other players is known and the initial conditions can be set more appropriately, improving the convergence time.
Example 7.5. Let the cost function be defined as in Example 7.2 and the initial conditions for the action components are selected randomly from [0, 20] and the estimate components from [−200, 200]. Here Assumption 4(i) on $F$ does not hold globally, so cannot apply Theorem 5.2. Under Assumption 3(ii) and 4(ii) by Theorem 5.3 (4.38) will converge depending on $\lambda_2(L)$. Figure 7.25 shows the convergence of (4.38) over a sufficiently connected, randomly generated communication graph $G_c$ as in Fig. 7.27. A high $1/\epsilon$ gain on the estimates can balance the lack of connectivity. Fig. 7.26 shows results for (5.26) with $1/\epsilon = 200$, over a cycle $G_c$ graph as in Fig. 7.28. Similar to example 7.2, using $1/\epsilon = 200$ for the cyclic graph gives similar convergence speed as the randomly generated graph. Both graphs remain at 0 for a small amount of time because of how far outside of the action set the estimate initial conditions are.

Figure 7.27: Random $G_c$, $\lambda_2 = 0.83$
Figure 7.28: Cycle $G_c$ Graph
Next we provide for reference, the results for running the projected gradient dynamics (3.7) in the perfect information case (Fig. 7.31), and the projected augmented gradient with estimate correction, (5.26) for $1/\epsilon = 200$, over a complete $G_c$, (Fig. 7.32). All plots have very similar performance independent of the graph and $1/\epsilon$. This is the same characteristics that are seen in example 7.2.

Example 7.6. Let the cost function be defined as in Example 7.3, where the initial conditions
for the action components are selected randomly from $[0, 20]$, while the estimate components are selected from $[-200, 200]$. Here Assumption 4(i) does not hold, but Assumption 4(ii) holds. By Theorem 5.4, (4.38) will converge depending on $\lambda_2(L)$. Figure 7.35 shows the convergence of (4.38) over a randomly generated, sufficiently connected, communication graph $G_c$ as in Fig. 7.33. Over a cycle $G_c$ graph, (4.38) do not converge. A high $1/\epsilon$ gain on the estimates can balance the connectivity loss. In fact running (5.26) with $1/\epsilon = 20$ over a cycle $G_c$ graph (Fig. 7.34) gives better performance than the random graph, and is shown in Fig. 7.36.

Next we provide for reference, the comparison of results for the projected gradient dynamics (3.7) in the perfect information case (see Fig. 7.37), and the projected augmented gradient with estimate correction, (4.38), over a complete $G_c$, (see Fig. 7.38). Just like in example 7.3, the perfect information case is significantly faster but can be made comparable to any connected
graph by using a large $1/\epsilon$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Figure7.37.png}
\caption{Perfect information $J_i(x_i, x_{-i}) = c_i(x_i) - x_i f(x)$, where $c_i(x_i) = [20 + 40(i-1)]x_i$ and $f(x) = 1200 - \sum_i(x_i)$. The action set is restricted to $\Omega_i = [0, 200]$ and the initial conditions for the action components are selected randomly from $[0, 200]$, while the estimate components are selected from $[-2000, 2000]$. The NE of this game is on the boundary $[200, 200, 183.3, 143.3, 103.3, 63.3, 23.3, 0, \ldots, 0]$ while the previous examples are in the interior of the action set. Figure 7.41 shows convergence of (4.38) over a randomly generated communication graph $G_c$ (Fig. 7.39). When $G_c$ is a cycle graph (Fig. 7.40), dynamics (4.38) give similar convergence result (Fig. 7.42), but is twice as slow.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{Figure7.39.png}
\caption{Random $G_c$, $\lambda_2 = 1.87$}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.2\textwidth]{Figure7.40.png}
\caption{Cycle $G_c$ Graph}
\end{figure}

Example 7.7. Consider an 20-player game where the cost function is $J_i(x_i, x_{-i}) = c_i(x_i) - x_i f(x)$, where $c_i(x_i) = [20 + 40(i-1)]x_i$ and $f(x) = 1200 - \sum_i(x_i)$. The action set is restricted to $\Omega_i = [0, 200]$ and the initial conditions for the action components are selected randomly from $[0, 200]$, while the estimate components are selected from $[-2000, 2000]$. The NE of this game is on the boundary $[200, 200, 183.3, 143.3, 103.3, 63.3, 23.3, 0, \ldots, 0]$ while the previous examples are in the interior of the action set. Figure 7.41 shows convergence of (4.38) over a randomly generated communication graph $G_c$ (Fig. 7.39). When $G_c$ is a cycle graph (Fig. 7.40), dynamics (4.38) give similar convergence result (Fig. 7.42), but is twice as slow.
Next we provide for reference a comparison of the results for the projected gradient dynamics (3.7) in the perfect information case (see Fig. 7.43), and the projected gradient with estimate correction, (4.38), over a complete $G_c$, (see Fig. 7.44).
7.3 Passive Consensus Dynamics

In this section we compare the difference between the gradient dynamics (4.32) with the gradient dynamics where the static Laplacian is replaced by a dynamic one (6.4), i.e.,

\[
\dot{x} = -R^T F(x) - Lr - Lx \\
\dot{r} = Lx
\]

Consider again the N-player quadratic game from economics, where 20 firms are involved in the production of a homogeneous commodity. The overall cost function of firm \(i\) is \(J_i(x_i, x_{-i}) = c_i(x_i) - x_i f(x)\), where \(c_i(x_i) = [20 + 10(i - 1)]x_i\) is the production cost, \(f(x) = 2200 - \sum_{i \in \mathbb{I}} x_i\) is the demand price, i.e, Example 7.1.

We investigate (4.32) and (6.4) over a cyclic graph \(G_c\). The initial condition for each component of \(x\) is selected randomly from \([0, 20]\). The initial condition for the auxiliary variable \(r\) is set to zero. In Example (7.1) the gradient dynamics converged for all connected graphs, therefore we expect to see convergence for dynamics (6.4). The figures below show how these two sets of dynamics compare. Figures 7.45 and 7.46 show the convergence of (4.32) and (6.4) respectively over a cyclic communication graph \(G_c\).

![Figure 7.45: (4.32) Static Laplacian](image1.png) ![Figure 7.46: (6.4) Dynamic Laplacian](image2.png)

From these plots both the static method and dynamic method give comparable results. From the analysis of the dynamic Laplacian in Section 6.3, the proof of convergence required the same conditions as the static Laplacian, therefore it is expected that both methods would have similar convergence speeds.
### 7.4 Higher Order Dynamics

In this section we will study the effects of noise on the dynamics of (4.32) and (6.9). Consider again the game, where 20 firms are involved in the production of a homogeneous commodity where the cost function of firm \(i\) is 
\[
J_i(x_i, x_{-i}) = c_i(x_i) - x_i f(x).
\]
The function 
\[
c_i(x_i) = 20 + 10(i-1)x_i
\]
is the production cost and 
\[
f(x) = 2200 - \sum_{i \in \mathcal{I}} x_i
\]
is the demand price. The dynamics of (4.32) with noise is,
\[
\dot{x} = -\mathcal{R}F(x) - \mathbf{L}x + n(t)
\]  
(7.1)
and the higher order dynamics (6.9), with noise is,
\[
\mathbf{w} = \begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \end{bmatrix} = \begin{bmatrix} w_2 \\ -d w_2 - \mathcal{R}^T \mathbf{F}(\mathbf{w}_1 + c\mathbf{w}_2) - \mathbf{L}(\mathbf{w}_1 + c\mathbf{w}_2) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ n(t) \end{bmatrix}
\]  
(7.2)
where the constant \(c = 1\) and \(d = 2\). The noise function for both dynamical systems is 
\[
n(t) = 50 \sin(20t) + 50 \sin(23t)
\]
The initial conditions are selected randomly from \([0, 20]\) and with a cyclic communication graph. Figure 7.47 shows the response of (7.1) and figure 7.48 shows the response of (7.2).

![Figure 7.47: (7.1) Gradient Method](image1)

![Figure 7.48: (7.2) Second Order Dynamics](image2)

It is clear that the higher order system is filtering out the sinusoidal noise. This shows that in situations where the communication network has noise, that changing (4.32) to a higher order system can reduce the noise considerably. The oscillation of the higher order system varies by about 0.6, while the regular gradient method changes by about 12 around the equilibrium.
7.5 Algorithm Comparison

Most of the existing literature that deals with Nash equilibrium seeking over a network is done in discrete time [4], [5], [15], [17], [18], [19], etc. It is not possible to compare the discrete time algorithm to the continuous time algorithm in this thesis. Additionally, these discrete time algorithms have terms that don’t have a corresponding continuous time counterpart. For example, the algorithm can be an asynchronous, probabilistic update rule and the estimate update takes the average of the neighbors’ values, i.e., \( x^1(t+1) = \frac{x^1(t)+x^2(t)}{2} \). The algorithms that are in continuous time don’t deal with the network setting or are for a class of game that have assumptions that are incompatible with the assumptions in this thesis, [24], [25].

Very recently [2] proposed a continuous time algorithm \(^1\) that is similar to the dynamics (5.15) in Corollary 5.10,

\[
\dot{\bar{P}} : \begin{cases} \\
\dot{x} = -F(R^T x + S^T z) \\
\dot{z} = -SL[R^T x + S^T z]
\end{cases} \tag{7.3}
\]

The difference is that the Laplacian, \( L = L \otimes I_n \) is replaced by a weighted Laplacian, \( L_w \). When replacing the Laplacian in (7.3) with the weighted Laplacian the estimate dynamics are,

\[
\dot{x}_j^i = \begin{cases} \\
-\left[\sum_{k \in N_i}(x^i_j - x^k_j)\right] & j \notin N_i \\
-\left[\sum_{k \in N_i}(x^i_j - x^k_j)\right] - \left[(x^i_j - x_j^i) - \text{extra term}\right] & j \in N_i
\end{cases} \tag{7.4}
\]

for all \( i \neq j \in \mathcal{I} \) (Note that the Laplacian is only in the \( \dot{z} \) dynamics which is the estimate components).

Consider the 20 player game in example 7.1 where the cost function \( J_i(x_i, x_{-i}) = c_i(x_i) - x_i f(x) \), \( c_i(x_i) = [20 + 10(i-1)]x_i \) and \( f(x) = 2200 - \sum_{i \in \mathcal{I}} x_i \). The initial conditions are set randomly from [0, 20] and the communication graph is a cyclic graph, i.e., figure 7.50. Figure 7.49 shows the response of the gradient-play dynamics with estimate correction, (4.30). Figure 7.51 shows the response of (7.3) and figure 7.52 shows the response of (7.3) with the weighted Laplacian, i.e., algorithm from [2].

\(^1\) The exact algorithm written in [2] does not converge. However, it is unclear if agent \( i \) uses \( x_i \) in the gradient or \( x^i_i \). In [2] \( x_i \neq x^i_i \) and using \( x^i_i \) the algorithm will fail to converge. Therefore, we assume that \( x^i_i \triangleq x_i \).
The response of the three dynamics over a cyclic graph suggests that all the methods are approximately the same. However, if the graph is more connected than the differences become more apparent. Figure 7.53 shows the response of (4.30) over a random communication graph, figure 7.54. Figure 7.55 shows the response of (7.3) and figure 7.56 shows the response of the algorithm in [2].
Using a random graph is clear that (4.30) is slower than the other two methods. The algorithm in [2] is still slightly faster than (7.3) but is because the weighted Laplacian is essentially scaling the Laplacian by a gain. Therefore, if the graph is strongly connected (7.3) becomes faster than (4.30). If the graph is weakly connected than both methods are comparable.
Chapter 8

Conclusions and Future Work

In this thesis, we have investigated how gradient dynamics with a consensus term can be used to converge to the Nash equilibrium over a sufficiently connected graph. In Chapter 2, we introduced the necessary preliminary background for this thesis. It describes passivity, the project operator, graph theory and stability results from control theory. In Chapter 3, we define what is a game and the most well known and solution concept, the Nash equilibrium. Necessary and sufficient conditions for the existence and uniqueness of the Nash equilibrium for both unconstrained and compact action spaces are stated and corresponding assumptions are made to ensure that the game does have an NE. Throughout this thesis we assumed and dealt with games that have a monotone pseudo-gradient. We begin by assuming that agents have full knowledge of all the other agents actions and show that gradient and projected gradient dynamics work.

In Chapter 4, we remove the assumption that agents know all other agents strategies and try to develop a method for solving the NE problem which still relies on the gradient. In order to accomplish this task the action space was enlarged so that all agents maintained an estimate of the other agents strategies. A consensus type feedback represented the communication that agents made over the network to update their estimates.

In Chapter 5, the dynamics were shown to converge if the extended pseudo-gradient was monotone. This condition was shown to have an interesting connection to distributed optimization and passivity. Under a weaker condition, convergence was proven if the communication graph was sufficiently connected, relative to the Lipschitz constant and the strong monotonicity constant $\mu$. The condition derived showed that if agents have strong coupling with other agents cost functions then the connectivity of the graph must be stronger. The two-time scale decomposition highlighted that agents can increase the speed of the consensus dynamics in order to reach the consensus subspace faster. The faster consensus dynamics can overcome the strong coupling forces between agents.

In Chapter 6, modifications to the dynamics were suggested for different parts of the system. The first such modification was allowing agents to use a weighted Laplacian, which could help improve the conditions required for convergence. Additionally, a communication protocol
Chapter 8. Conclusions and Future Work

perspective was suggested to provide a security / anonymity to agent’s estimate information that is being shared over the network. Modifications to the Laplacian dynamics were highlighted to have a passivity property and allowed for deriving a class of LTI passive dynamics on the Laplacian. These dynamics maintained the NE and still converged to the NE. Next, higher order dynamics were suggested that also contains some passivity property. Second order dynamics were shown to converge with the same conditions on the connectivity of the graph with some mild conditions on the constants for the LTI system. Lastly, this chapter showed that higher order gradient dynamics can be interpreted as dynamical game. This relationship shows that the conditions on the parameters of the LTI system dictate what type of dynamical game the gradient method can converge in.

In Chapter 7, simulation results of all the dynamics are shown for a set of different problems, over different graphs.

8.1 Future Work

In this thesis we used the gradient as the base component of the dynamics and then made modifications to the gradient until it worked in a network setting. In Chapter 6 we changed everything of the original system except for the function $F(x)$ itself. In the optimization literature, gradient is the basic method used and over the years more powerful methods have been developed. Therefore, alternative “driving” components should be analyzed such as replacing the extended pseudo-gradient with a best response map, or a regularized gradient descent. More specifically, replacing the pseudo-gradient with a proximal / forward-backward splitting method. Unfortunately, the techniques used in optimization are for discrete time systems and there doesn’t appear to be a straightforward or equivalent method in continuous time. By replacing the gradient with another method hopefully the dynamics will be able to converge under weaker assumptions.

The Lipschitz condition is a very mild and standard assumption in optimization, so this condition will probably remain. The strong monotonicity assumption should be weakened to a (strict) monotone assumption without any additional assumptions. For example, the strong monotonicity condition can be reduced to just strict monotonicity but require the pseudo-gradient to be cocoercive. This assumption can be weakened by restricting the class of game or exploiting additional properties of the pseudo-gradient.

Another possible avenue is to broaden the class of games that the dynamics can converge in. In this thesis the game was assumed to be monotone. One way of tackling this problem is to add dynamics into system $P_w$ in figure 6.3. The feedback dynamics behave like a regulation term and can change the gradient dynamics considerably. The feedback could have anticipatory dynamics which make predictions about other agents future actions.

Instead of weakening the assumptions of the game an alternative is to weaken the assumption on the knowledge that agents have. In this thesis agents are assumed to know their own cost
function. A weaker condition is to assume that agents don’t know their cost function but only know the outcome of their selected action. The figures below show the gradient being transformed into a payoff based learning dynamics.

\[
\begin{align*}
\dot{x} &= -F(u) \\
y &= x
\end{align*}
\]

\[
\begin{align*}
\dot{p} &= f(p, u) \\
x &= h(p, u)
\end{align*}
\]

Figure 8.1: Gradient Dynamics  
Figure 8.2: Payoff Dynamics

The dynamics in figure 8.1 is the original gradient dynamics with perfect information. Figure 8.2 assumes that players no longer know what their cost function \( J(x) \) is, but instead can measure the resulting cost. The function \( f(p, u) \) is the dynamics that use the payoff information and the function \( h(p, u) \) selects the action. The dynamics are now on the payoffs instead of the actions. Maybe by using the communication network agents can get additional information to ensure convergence to the NE.
Appendix A

Matrix Proofs

Lemma A.1. Given a matrix $M \in \mathbb{R}^{p \times q}$, then $\text{Null}(M) = \text{Null}(M^T M)$

Proof. Let $x \in \text{Null}(M)$, then $Mx = 0 \implies M^T Mx = 0$ and therefore, $\text{Null}(M) \subseteq \text{Null}(M^T M)$. If $x \in \text{Null}(M^T M)$, then $M^T Mx = 0 \implies x^T M^T Mx = (Mx)^T (Mx) = 0 \implies Mx = 0$ and therefore, $\text{Null}(M^T M) \subseteq \text{Null}(M)$. Hence, $\text{Null}(M) = \text{Null}(M^T M)$. □

Lemma A.2. If $G_c$ is a connected graph then $\|RLS^T\| = \max_{i \in I} \sqrt{|N_i|}$

Proof. Let the Laplacian $L$ be represented as,

$$
L = \begin{bmatrix}
L_1 \\
\vdots \\
L_i \\
\vdots \\
L_N \\
\end{bmatrix} = \begin{bmatrix}
l_{1,1} & \cdots & l_{1,j} & \cdots & l_{1,N} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
\vdots & \ddots & l_{i,j} & \ddots & \vdots \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
l_{N,1} & \cdots & l_{N,j} & \cdots & l_{N,N}
\end{bmatrix}
$$

where $l_{i,j} \in \mathbb{R}$ is an element of the matrix and $L_i \in \mathbb{R}^{1 \times N}$ is the $i$th row of the matrix.
Additionally, the matrix $\mathbf{L}$ can be expressed as,

$$
\mathbf{L} = \begin{bmatrix}
\mathbf{L}_{1,1} & \mathbf{L}_{1,N} & \vdots & \mathbf{L}_{i,1} & \vdots & \mathbf{L}_{i,N} & \vdots & \mathbf{L}_{N,1} & \mathbf{L}_{N,N}
\end{bmatrix}
= \begin{bmatrix}
\mathbf{L}_{1,1} \otimes \mathbf{R}_{1} & \mathbf{L}_{1,N} \otimes \mathbf{R}_{N} & \vdots & \mathbf{L}_{i,1} \otimes \mathbf{R}_{1} & \mathbf{L}_{i,2} \otimes \mathbf{R}_{1} & \cdots & \mathbf{L}_{i,N} \otimes \mathbf{R}_{1} & \mathbf{L}_{i,1} \otimes \mathbf{R}_{N} & \mathbf{L}_{i,2} \otimes \mathbf{R}_{N} & \cdots & \mathbf{L}_{i,N} \otimes \mathbf{R}_{N} & \vdots & \vdots & \vdots & \mathbf{L}_{N,1} \otimes \mathbf{R}_{1} & \mathbf{L}_{N,2} \otimes \mathbf{R}_{1} & \cdots & \mathbf{L}_{N,N} \otimes \mathbf{R}_{1}
\end{bmatrix}
$$

where $\mathbf{L}_{i,j} \in \mathbb{R}^{n_i \times n}$ is a row vector / matrix and $\mathbf{R}_i$ is defined as in equation (4.1). An interesting property of stacking the $\mathbf{R}_i$ matrices is that it is equal to an identity matrix,

$$
\begin{bmatrix}
\mathbf{R}_1 \\
\mathbf{R}_2 \\
\vdots \\
\mathbf{R}_N
\end{bmatrix}
= \begin{bmatrix}
\mathbf{I}_{n_1} & 0 & \cdots & 0 \\
0 & \mathbf{I}_{n_2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \mathbf{I}_{n_N}
\end{bmatrix}
= \begin{bmatrix}
\mathbf{R}_1^T & \mathbf{R}_2^T & \cdots & \mathbf{R}_N^T
\end{bmatrix}
$$

where $\mathbf{R}_i$ behaves like a block identity matrix. If one of the $\mathbf{R}_i$ is set to zero it would be equivalent to removing a block identity element. Therefore,

$$
\begin{bmatrix}
\alpha_1 \mathbf{R}_1 \\
\alpha_2 \mathbf{R}_2 \\
\vdots \\
\alpha_N \mathbf{R}_N
\end{bmatrix}
= \alpha_1 \mathbf{R}_1^T \alpha_2 \mathbf{R}_2^T \cdots \alpha_N \mathbf{R}_N^T
$$

(A.1)

for all $\alpha_i \in \mathbb{R}$. From equation (4.6) and from example 4.1.1 the matrix $\mathbf{R}^T \mathbf{R}$ sets the rows that are estimate components to zero. Notice that the vector $\mathbf{x} \in \Omega^N$ has the component $x^i_j$ in the same row as $\mathbf{L}_{i,j}$. Therefore, if $\mathbf{R}^T \mathbf{R}$ sets the estimate component $x^i_j$ to zero when $i \neq j$ then
the matrix row \( \mathbf{L}_{i,j} \) will also be set to zero. Hence,

\[
\mathbf{R}^T \mathbf{R} \mathbf{L} = \begin{bmatrix}
\mathbf{L}_{1,1} \\
0 \\
\vdots \\
0 \\
\vdots \\
\mathbf{L}_{N,N}
\end{bmatrix} = \begin{bmatrix}
\mathbf{L}_1 \otimes \mathbf{R}_1 \\
\mathbf{L}_2 \otimes \mathbf{R}_2 \\
\vdots \\
\mathbf{L}_i \otimes \mathbf{R}_i \\
\vdots \\
\mathbf{L}_N \otimes \mathbf{R}_N
\end{bmatrix} = \begin{bmatrix}
l_{1,1} \mathbf{R}_1 \\
l_{1,2} \mathbf{R}_1 \\
\vdots \\
l_{i,i} \mathbf{R}_i \\
\vdots \\
l_{N,N} \mathbf{R}_N
\end{bmatrix}
\]

and is equal to \( \mathbf{L} \) where \( \mathbf{L}_{i,j} = 0 \), for all \( i \neq j \). Grouping the \( \mathbf{R}^T \mathbf{R} \mathbf{L} \) matrix into \( n \) by \( n \) block, \( \begin{bmatrix} 0 \\ l_{i,j} \mathbf{R}_i \\ 0 \end{bmatrix} \), and using equation (A.1) by setting some of the \( \alpha \)'s to 0, this matrix can be rewritten as,

\[
\mathbf{R}^T \mathbf{R} \mathbf{L} = \begin{bmatrix}
l_{1,1} \mathbf{R}_1^T \\
0 \\
\vdots \\
0 \\
\mathbf{L}_2 \otimes \mathbf{R}_2 \\
\vdots \\
\mathbf{L}_i \otimes \mathbf{R}_i \\
\vdots \\
\mathbf{L}_N \otimes \mathbf{R}_N
\end{bmatrix} = \begin{bmatrix}
l_{1,1} \mathbf{R}_1^T \\
l_{1,2} \mathbf{R}_1^T \\
\vdots \\
l_{i,i} \mathbf{R}_i^T \\
\vdots \\
l_{N,N} \mathbf{R}_N^T
\end{bmatrix}
\]

Notice that when multiplying a row \( \mathbf{L}_{i,j} = \begin{bmatrix} l_{i,1} \mathbf{R}_j \\ l_{i,2} \mathbf{R}_j \\
\vdots \\
l_{i,N} \mathbf{R}_j \end{bmatrix} \) with a column of \( \mathbf{R}^T \mathbf{R} \mathbf{L} \) all products with the term \( \mathbf{R}_i \mathbf{R}_j^T \) are \( I_{n_i} \) when \( i = j \) and zero otherwise, following from equation (4.9) and (4.10). Since each row of \( \mathbf{R}^T \mathbf{R} \mathbf{L} \) has only one \( \mathbf{R}_i \) the product \( \mathbf{L} \mathbf{R}^T \mathbf{R} \mathbf{L} \) will be,

\[
\mathbf{L} \mathbf{R}^T \mathbf{R} \mathbf{L} = \begin{bmatrix}
l_{1,1} \mathbf{L}_{1,1} \\
l_{1,2} \mathbf{L}_{2,2} \\
\vdots \\
l_{i,j} \mathbf{L}_{j,j} \\
\vdots \\
l_{N,N} \mathbf{L}_{N,N}
\end{bmatrix} \quad \mathbf{S} \mathbf{L} \mathbf{R}^T \mathbf{R} \mathbf{L} = \begin{bmatrix}
l_{1,2} \mathbf{L}_{2,2} \\
l_{1,3} \mathbf{L}_{3,3} \\
\vdots \\
l_{i,j} \mathbf{L}_{j,j} \\
\vdots \\
l_{N,N-1} \mathbf{L}_{N-1,N-1}
\end{bmatrix}
\]

The matrix \( \mathbf{S} \) removes the rows in the action component, therefore rows that contain \( l_{i,i} \mathbf{L}_{i,i} \) are
removed from $L R^T RL$ to give $S L R^T RL$. Lastly multiplying this matrix by $S^T$ gives,

\[
S L R^T R L S^T = \begin{bmatrix}
  l_{1,2} L_{2,2} S^T \\
l_{1,3} L_{3,3} S^T \\
  & \vdots \\
l_{1,N} L_{N,N} S^T \\
l_{2,1} L_{1,1} S^T \\
l_{2,3} L_{3,3} S^T \\
  & \vdots \\
l_{i,j \neq i} L_{j,j} S^T \\
  & \vdots \\
l_{N,N-1} L_{N-1,N-1} S^T
\end{bmatrix}
\]

Notice that since the matrix is in the form $M^T M$ the matrix is symmetric. If agent $i$ doesn’t communicate with $j$ then $l_{i,j} = l_{j,i} = 0$ and the row $l_{i,j} L_{j,j} S^T$ and $l_{j,i} L_{i,i} S^T$ will be zero. Therefore, $n_i + n_j$ eigenvalues will be zero because of these two terms. If agent $i$ communications to both agent $j$ and agent $k$ then $l_{j,i} L_{i,i} S^T = l_{k,i} L_{i,i} S^T$ will be the same. If agent $i$ communicates with $k$ different agents there will be $(k - 1)n_i$ eigenvalues which are 0.

Since there are no leading $l_{i,i}$ terms in $S L R^T R L S^T$ all the leading $l_{i,j}$ terms are either 0 or $-1$. The matrix product $L_{i,i} S^T$ is,

\[
L_{i,i} = L_i \otimes R_i = [l_{i,1} R_i, \ldots, l_{i,i} R_i, \ldots, l_{i,N} R_i]
\]

\[
S^T = \begin{bmatrix}
  S_1^T & 0 & 0 \\
  0 & S_2^T & 0 \\
  0 & 0 & \ddots
\end{bmatrix}
\]

\[
L_{i,i} S^T = [l_{i,1} R_i S_1^T \ldots l_{i,i} R_i S_i^T \ldots l_{i,N} R_i S_N^T]
\]

Therefore, from equation (4.9) the term $l_{i,i} R_i S_i^T = 0$ and from equation (4.11) all other terms are $l_{i,j} \begin{bmatrix} 0 & I_{n_i} \end{bmatrix}$ for $i \neq j$. As mentioned before all the leading terms, $l_{i,j}$, in $S L R^T R L S^T$ are either equal to $-1$ or 0. Therefore, the overall matrix has entries of either 0 or 1. Notice
that if a vector $x = -S l_{i,i}^T$ multiplies $L R^T R L S^T$ then,

$$L R^T R L S^T x = \begin{bmatrix}
    l_{1,1} L_{1,1} S^T \\
l_{1,2} L_{1,2} S^T \\
\vdots \\
l_{1,N} L_{1,N} S^T \\
l_{2,1} L_{2,1} S^T \\
l_{2,2} L_{2,2} S^T \\
\vdots \\
l_{i,j} L_{i,j} S^T \\
\vdots \\
l_{N,N} L_{N,N} S^T
\end{bmatrix}
\begin{bmatrix}
    x \\
    \vdots \\
- l_{1,i} |N_i| I_{n_i} \\
0 \\
\vdots \\
- l_{j,i} |N_i| I_{n_i} \\
0 \\
\vdots \\
- l_{N,i} |N_i| I_{n_i} \\
0
\end{bmatrix} = \begin{bmatrix}
    - l_{1,i} |N_i| R_i^T \\
\vdots \\
- l_{j,i} |N_i| R_i^T \\
0 \\
\vdots \\
- l_{N,i} |N_i| R_i^T
\end{bmatrix} \quad \text{(A.3)}$$

following from the fact that $-L_{i,j} S^T S l_{i,i}^T = -\sum_{k \in I} l_{i,k} l_{j,k} R_{i,k} S_{k}^T S_{k} R_{i,k}^T$ and from equation (4.11) is 0 if $i \neq j$. Additionally, If $i = k$ the term $R_{i,k}^T = 0$ therefore $-L_{i,i} S^T S l_{i,i}^T = -\sum_{j \in I} l_{i,j} R_{i,j} S_{j}^T S_{j} R_{i,j}^T = -\sum_{j \in I} l_{i,j}^2 I_{n_i} = -I_{n_i} \sum_{j \in I} l_{i,j}^2$. The term $l_{i,j}^2 = 1$ if agent $i$ communicates to agent $j$ and 0 otherwise. Therefore, the sum is the number of agents that $i$ communicates to, i.e., $|N_i|$. Multiplying this matrix by $S$ gives,

$$S L R^T R L S^T x = \begin{bmatrix}
    S_1 & 0 & 0 \\
0 & S_2 & 0 \\
0 & 0 & \vdots
\end{bmatrix}
\begin{bmatrix}
    -l_{1,i} |N_i| R_i^T \\
\vdots \\
- l_{j,i} |N_i| R_i^T \\
0 \\
\vdots \\
- l_{N,i} |N_i| R_i^T
\end{bmatrix} = |N_i| x \quad \text{(A.4)}$$

Therefore, all vectors of the form $x = -S l_{i,i}^T$ are eigenvectors of $S L R^T R L S^T$ with eigenvalue of $|N_i|$. The norm of a matrix is defined as $\| R L S^T \| = \sqrt{\lambda_{\max}(S L R^T R L S^T)}$ where $\lambda_{\max}(\cdot)$ returns the max eigenvalue. Since, all the eigenvalues are 0 or $|N_i|$ the norm is therefore $\| R L S^T \| = \max_{i \in I} \sqrt{|N_i|}$.

\textbf{Lemma A.3.} Let $S = \text{diag}(S_1, \ldots, S_N) \in \mathbb{R}^{(N-1)n \times n}$ where $S_i$ be defined in equation (4.4). Then, $\| S(I_N \otimes I_n) \| = \sqrt{N-1}$.

\textbf{Proof.} Given a vector $x^T = (1_N \otimes x)^T = [x^1, \ldots, x^N]$ where $x^1 = \cdots = x^N = x$, the matrix $S$
removes the $x_i^i$ component. For example when $N = 3$ and $n = 3$ then,

$$Sx = \begin{bmatrix} x_{-1}^1 \\ x_0^2 \\ x_2^2 \\ x_2^3 \\ x_3^1 \\ x_3^2 \\ x_3^3 \end{bmatrix} = \begin{bmatrix} x_1^1 \\ x_2^2 \\ x_3^3 \\ x_1^1 \\ x_2^2 \\ x_3^3 \\ x_1^1 \end{bmatrix}$$ \hspace{1cm} (A.5)

Let $P$ be a permutation matrix such that the components of $x_N^N$ are moved into $x_i^j$, $\forall i \neq N$ to restore the component that was removed by $S$. Vector component $x_1^1$ is removed by $S$ so $x_1^N$ will be moving into $x_1^1$’s position, i.e.,

$$PSx = PS(1_N \otimes I_n)x = \begin{bmatrix} x_1^1 \\ x_2^2 \\ x_3^3 \\ x_1^1 \\ x_2^2 \\ x_3^3 \\ x_1^1 \end{bmatrix} = \begin{bmatrix} x_1^1 \\ x_2^2 \\ x_3^3 \end{bmatrix} = (1_{N-1} \otimes x) = (1_{N-1} \otimes I_n)x$$ \hspace{1cm} (A.6)

where $x_1^1 = x_1^1$ and $x_2^2 = x_2^2$ because $x_i^j = x_j^j$ for all $i, j$. The norm of a matrix pre-multiplied by a permutation matrix doesn’t change the norm of the matrix. Therefore,

$$\|S(1_N \otimes I_n)\| = \|PS(1_N \otimes I_n)\| = \|(1_{N-1} \otimes I_n)\| = \|1_{N-1}\| \|I_n\| = \sqrt{N-1}$$
Bibliography


BIBLIOGRAPHY


[54] J. J. Sylvester, “A demonstration of the theorem that every homogeneous quadratic polynomial is reducible by real orthogonal substitutions to the form of a sum of positive and negative squares,” *Philosophical Magazine*, vol. 4, no. 23, pp. 138 – 142, 1852.