Early Stage Report

Aamal Abbas Hussain
CID: 01060230

Department of Computing
Imperial College London
Supervisor: Francesco Belardinelli
June 2020

Abstract

This project aims to study the evolution of a swarm of intelligent 'active particles' under the influence of Model Predictive Controls. We specifically aim to investigate the learning dynamics of a population of agents following a Q-Learning approach. We also aim to study the evolution of such a swarm under Model Predictive Controls generated by field particles. The goal of this study is to bridge the gap between swarm theory and multi-agent reinforcement learning to enable the study of complex populations of agents, who are capable of independent decision making, as well as producing control methodologies to enable the safe operation of intelligent swarms.

Contents

Chapter 1

Introduction

Swarm robotics bases its success upon the collaboration of many agents who contribute towards the overall goal of the system [**?**]. Such a collaboration can take the form of cooperative efforts or even competition, both resulting in behaviours that are far more complex thhan individual robots can achieve. The capacity for complexity is becoming increasingly important as robots are presented with ever more challenging tasks including: search and rescue, cleaning up space debris, and robotic assembly. Such tasks are beyond the capabilities of individual robots and so we must leverage the capacity for agents in a swarm to collectively achieve our goals. Fortunately, a growing subset of the Control Theory community has turned their attention towards this very problem and address the question: how does the dynamical behaviour of a population of agents change under the influence of control? Note that we use the term 'agent' loosely here, an agent can refer to anything from a simple particle [**?**] to a complex robot [**?**]. In answering this question, researchers have gained a stronger understanding of how a swarm may be controlled to achieve desired results and have shown its capabilites in a wide array of domains [**?**].

Another technique which has made incredible strides towards improving the capability of autonomous systems to communally achieve results is Multi-Agent Reinforcement Learning (MARL) [**?**]. This builds on the ideas of classical Reinforcement Learning which allows an agent to optimise its strategies towards tackling a task through repeated exposure to that task. MARL utilises this same principle with the caveat that agents must adapt their strategies based not only on the outcome of the action, but also on the actions of the other agents. This has shown strong results in practical applications [**?**, **?**, **?**]. Whilst MARL was once considered to be non-identifiable (there is no clear way of identifying the algorithm that a particular agent was trained on) and black box (there is no clear way of ensuring a desired result from a learning algorithm), a steady stream of research is emerging to lift this fog [**?**]. This allows for researchers to choose the parameters of their learning algorithms appropriately to ensure that learning agents behave in the desired manner.

1.1 Problem Statement

Based on the successes of the swarm control and MARL communities, a natural question to ask is whether the promises of these disciplines may be unified. Such a combination would radically improve the capabilities of autonomous systems. Ultimately such a system must be driven under the influence of control inputs to perform given tasks. Examples of this might be: forming around individual components of space debris and moving them to a desired location (e.g. a bin), or relocating to multiple areas in a disaster zone to provide immediate relief. For such tasks to be achieved in dynamic environments, which is typically the case, the system must be able to constantly adapt its approach and respond accordingly. Given these considerations, the problem may be stated as follows

1. A population of robots is given a goal which must be achieved, and can often be deconstructed into a set of smaller tasks, to be executed simultaneously.

2. These are to be performed in a dynamic environment with obstacles present and where the number of tasks, and their priorities may be in constant flux.

3. The agents in the system must, therefore, make decisions collectively which will result in the system achieving the given goals. Due to the dynamic quality of the problem, these decisions may need to be evaluated online and in a decentralised fashion, as it may not be possible for a centralised decision maker to communicate with all agents.

4. It is desirable, therefore, to control the dynamical behaviour of a system composed of agents who learn and adapt through interactions with each other. In addition, to ensure the safe operation of the system, properties such as state and control constraints must be satisfied.

With the goal of addressing this problem in mind, the aim of this PhD is to address the question posed by the final point: how does the dynamical behaviour of a swarm of intelligent, adaptive agents evolve under the influence of defined control inputs?

1.2 Objectives and Scope

In particular, we will study

- How the strategies of a population of agents evolves as they iteratively interact with one another. We will examine whether the system evolves to a stable equilibrium, or whether it exhibits complex, even chaotic behaviour.

- The extent to which the dynamical behaviour of a population of learning agents is influenced under given controls. We seek to understand the conditions under which guarantees on theoretical results such as controllability, stability, and well-posedness may be established.

The resultant work will provide novel insights into the behaviour of swarm systems and provide new methods for which they may be practically used.

Relevance to Safe and Trusted AI This study will seek to understand the guarantees that can be placed on the safety of such systems in terms of their stability and constraint satisfaction. If it is possible to determine, a priori, the decision making behaviour of a population through its learning dynamics in terms of known quantities (e.g. the agent parameters), then the system may be considered safe. Similarly, if it can be shown to be robust, and satisfy desired constraints (e.g. controls lying within desired limits), then we may be assured of its long-term safety. In doing so, we can ensure that a user can have confidence in the system's decision making and manipulation.

Chapter 2

Literature Review

In this chapter, we explore the current state of the art in Multi-Agent Systems (MAS). We will consider the main techniques that have been proposed towards task allocation and control. In this review we cover

1. Game Theory: The most well studied model for solving multi-agent decision making problems.
2. Multi-Agent Reinforcement Learning (MARL): A contemporary technique which requires that agents learn and adopt their own strategies.
3. Control Theory: Methods focusing on low-level manipulation of a system (e.g. the position and velocity of an agent) which aims to provide guarantees on its behaviour.
4. Swarms: Systems whose success are founded on the emergent phenomena exhibited by large populations of interacting entities.

There are, of course, a number of techniques which we do not discuss here, as they are not relevant to the proposals in this report. Most notable is that of Decentralised Partially Observable Markov Decision Processes (Dec-POMDPs) which is a multi-agent extension of the MDP framework in which agents aim to maximise a common reward (or utility) function. This method does not scale to large populations and so is not considered relevant to our study. The interested reader should consult [**?**] for an excellent introduction to Dec-POMDPs.

2.1 Multi Agent Systems

We will keep the problem statement defined in Section **??** in mind as we define several metrics which a Multi Agent System (MAS) may be assessed against.

1. *Distributed*. It will often be the case that a multi-agent system must spread out across a large area or operate in regions where communication is limited. As such it is beneficial that such a system does not rely too heavily on communication with a central body. The individual agents must, therefore, be able to make their own decisions and act as independent entities. Aside from the low communication load, this ensures that the system is not tied to a single point of failure; in a system governed by a centralised decision maker, any faults in the central body propagates throughout the group. Of course, this comes at the price of computational load - if the agents are required to act independently, they must possess the resources to do so. The term distributed should not be confused with decentralised. The latter refers to a regime in which agents all communicate with one another before making a decision. Whilst this is preferable to a centralised decision maker in terms of the requirement for one entity to communicate with all agents, it still places a high cost on the requirement that agents be in constant communication with one another. In fact it may pose

the additional risk that if any agent in the team fails, the whole pipeline falls apart. Distributivity, on the other hand, requires that agents make independent decisions so that, though agents may communicate with each other, they are still able to operate when this is not possible.

2. *Generality.* MAS come in many different flavours; sometimes a small group of heterogeneous (agents with different capabilities) agents may be required to perform a delicate task whilst in others, it may be required that a large population of homogeneous (all of the same type) agents perform multiple tasks simultaneously. Indeed the same MAS may have to adapt its strategy whilst in the middle of a task. It would be a hindrance if the agents were required to be reprogrammed with a completely different control methodology each time the situation changes slightly. It would be preferable if the method could apply to all different types of scenarios, with homogenous or heteregeneous agents, with single tasks or multiple, in any environment with only minor customisation required. Better still if the agents are able to adapt themselves to the specifics of the scenario.

3. *Low Computational Load.* As mentioned in the previous chapter, a MAS will often be used in complex situations, many of which will be too dangerous for human intervention. As such, the system will be required to carry out its tasks online. For this reason, it is vital that the control method be one which produces low computational load - especially where the time frame for adjusting agent behaviour is limited. Monetary concerns also play a part in this metric; it would not do to impose that every member of the team be equipped with high performance computing power as this would likely fall outside of the budget of any practical usage.

4. *Robustness.* Particularly for physical robots, a MAS should be able to operate in situations which present sensor noise, communication noise, latency, and environmental disturbances. This is rather important in unseen environments where full a priori knowledge of the map cannot be assumed.

5. *Scalable.* This is mentioned last since the ability for a MAS to scale comes from its distributivity and low computational load. Scalability is the capacity for any method of controlling a MAS to successfully operate as the size of the team increases. This, of course, is related to the complexity of the algorithms involved. However, it should be noted that it is not always required that a MAS be formed of dozens (or even hundreds) of agents. Depending on the task at hand (e.g. robotic surgery), it may be beneficial for fewer, more capable agents to perform the task, rather than a large population.

We begin by presenting a subset of the vast array of literature regarding MAS and focus on those which are relevant to the study at hand. We conclude this chapter by providing some remarks about how these methods hold against the above metrics.

2.2 Game Theory

Game Theory has a rich history when considering an understanding of multi-agent systems and it is impossible to examine MAS without an understanding of Game Theory. These begin in economics but have found a strong application in computation due to the rising need for distributed systems. Game Theory, therefore, branches across all of the categories in this chapter although its synergy with swarms requires development) since Dec-POMDP (Decentralised Partial Observable Markov Decision Processes) and MARL methods have both base their theory upon the foundations of Game Theory.

2.2.1 Partially Observed Stochatic Games (POSG)

POSGs act as a game theoretic corollary to the Dec-POMDP. Here, an optimal solution to a multi-agent decision making problem is found by fixing the strategy (a manner of selecting an action [**?**]) of all agents

except one. The optimal strategy of the chosen agent is then computed. This is fixed so that the next agent may determine its optimal strategy. This process is iterated over all agents in a process known as Alternating Maximisation [**?**]. By choosing strategies which maximise a common payoff, a globally optimal solution can be found.

Similarly Bayesian Games is a game theoretic approach with incomplete information (thus it is sometimes referred to as an 'incomplete information game). This allows for a POSG to be approximated by a sequence of smaller, more tractable games [**?**]. However, iterating through the entire team of agents to find an optimal solution is a lengthy process and does not allow for immediate actions in dynamic situations. To address this, [**?**] propose to place distinct payoffs on each agent and to consider team formation based on social hierarchy as well as preferred partners. This allows the complexity of the problem to be broken down into smaller game units and also provides a clear order in which the games are to be played. This minimises the interference across robots and allows for more immediate action to be taken by agents who choose their strategy early on, while leaving more passive tasks for those later in the sequence.

The Bayesian Game formulation therefore provides a strong candidate for rapid task re-allocation and dynamic decision making. However, as found by Dai et al. [**?**] this is conditional on an understanding by each robot of the strategy of others, which in turn requires adequate communication between robots. Therefore, the game theoretical formulation will fail where communication is not possible amongst team members. However, the addition of heuristics, such as deep learning, may be able to advance an agents ability to recognise the strategies taken by other members of its team without the need for significant explicit communication.

2.2.2 Game Theoretic Control

Game theory can often be applied to problems of control theory (particularly where there are multiple agents) to develop robust controllers which guarantee properties of stability and constraint satisfaction. We define these in our review on Control Theory (see Section **??**).

This idea is explored in [**?**]. Here, a zero-sum game is considered in which the players are a controller and an adversarial environment. The design of the controller must be such that it is able to drive the system to zero error. To illustrate, consider the problem of designing a controller for a re-entry vehicle, as in [**?**], in which vortices seek to destabilise the agent. This will allow us to build stable agents in a much more efficient manner since we can simulate the adversarial environment and hypothetical scenarios the agent may encounter without actually encountering them. The same notion is explored by Bardi et al [**?**].

Mylvaganam et al, in [**?**], consider the N-robot collision avoidance problem, similarly from the point of view of differential game theory. They develop a robust feedback system for the robots which they show to be able to drive the system towards predefined targets whilst providing guarantees of interference from other agents (or lack thereof). In [**?**], Mylvaganam also considers a game theoretic control of multi-agent systems in a distributed manner. Here, agents only consider their own payoff structure and have limited communication with one another. The author shows that an approximate equilibrium can be found using algebraic methods and illustrate the capabilities of the technique using a collision avoidance example. For the sake of brevity the numerous contributions that Mylvaganam has made to this field is not presented here. However, we will conclude with those presented in [**?**]. Here, the author presents approximate solutions to a number of differential games, including linear-quadratic differential games (in which system dynamics are linear functions whilst payoff functions are quadratic), Stackelberg differential games, where a hierarchy is induced across the players (a notion was suggested in the research proposal) and mean-field games, which are discussed under 'MARL'. The importance of the linear-quadratic differential game is the stability of the solution; solutions for the Nash equilibria (NE) are admissable iff they are locally exponentially stable (which the author often shows with the aid of Lyapunov functions). Approximate solutions to the NE are developed which are more feasible to calculate online. The author then shows that this is not simply a theoretical exercise by applying the novel methods towards multi-agent collision

problems and designs dynamic control laws which guarantee that each agent will reach their desired state whilst avoiding collision with the other agents. Similarly, the Stackelberg game is applied to the problem of optimal monitoring by a multi robot system.

2.3 Multi Agent Reinforcement Learning

Reinforcement learning extends the Markov Decision Process problem by considering the case where the reward model is not initially known to the agent. In a similar manner, Multi Agent Reinforcement Learning (MARL) extends the Markov Game setting to one where the payoff structure is not a priori knowledge. MARL has found success in a number of applications, particularly in robotics [**?**] and but also in sensor/communication networks and finance [**?**].

The task of MARL is to determine an optimal joint policy for all agents across the game. This joint policy may be the concatenation of all the individual policy or it may just be options for each agent to take. In either case, optimality is defined through the standard notions of Nash equilibria and so, in this section we consider the broad spectrum of methods which attempt to achieve Nash equilibria.

The largest problem in MARL is the non-stationarity of the environment [**?**]. In single-agent settings, it is assumed that the environment is Markovian. However, this must be lifted in the Multi Agent setting since other agents in the environment will be learning concurrently. As such, we must now consider that the policy for any one agent will depend on the policy of all other agents.

This section begins with a selection of the foundational methods which were developed towards solving the MARL problem. The interested reader may find additional methods and implementations of these techniques in [**?**].

2.3.1 Learning in Two Player Games

The most fundamental method to learning in Matrix games is the simplex algorithm. This is a popular method of linear programming (in which constraints are linear). This will be important in considering more current methods. A similar consideration is given to the infinitesimal gradient ascent algorithm, in which the step size converges to zero. This method guarantees that, in the infite horizon limit, the payoffs will converge to the Nash equilibrium payoff. Note that this does not necessarily mean that both agents will converge to a single Nash equilibrium. This is a particular problem in games where there are multiple Nash equilibria. However, in practice it is difficult to choose a convergence rate of the step size and, without an appropriate choice the strategy may oscillate as shown in the book [**?**]. To address this, a modified approach is presented by Bowling and Veloso which incorporates the notion of Win or Learn Fast (WoLF) to produce WoLF-IGA [**?**]. WoLF is a notion we will come across often in MARL and is shown by the authors to converge to always to a NE. The concern with WoLF methods, however, is that it requires explicit knowledge of the payoff matrix (which is not so much of a problem for model based methods) and the opponent's strategy (which is more of a problem in real-world methods). Finally, the Policy Hill Climbing method (PHC) is shown to converge to an optimal mixed strategy if the other agents are stationary (i.e. are not learning). However, it is shown that, when this is not the case, the algorithm again oscillates. The WoLF-PHC adaptation of this method is shown to converge to a NE strategy for both players with minimal oscillation.

2.3.2 Learning in Stochastic Games

Stochastic Games (or Markov Games) form a basis for MARL settings. However, in this case the agents must learn about the equilibrium strategies by playing the game, which means they do not have a priori knowledge of the reward or transition functions. Schwarz [**?**] considers two properties which should be used for evaluating MARL algorithm: rationality and convergence. The latter simply states that the method should converge to some equilibrium whereas the former suggests that the method should learn

the best response to stationary opponents. A similar set of conditions is considered by Conitzer and Sandholm in [**?**].

A plethora of MARL techniques emerged towards tackling this problem. Most notable of these are the minimax Q-Learning algorithms proposed by Littman [**?**] which focuses on zero-sum games and Nash Q-Learning algorithm put forward by Hu and Wellman [**?**] which extends the framework towards general sum games. The details of such stochastic algorithms are beyond the scope of this review as they consider multi-stage games (i.e. games in which the payoff matrix changes according to state). Establishing the robustness of such systems is an ongoing area of research. However, as of this review, it is not yet possible to place guarantees on the behaviour of MARL algorithms which tackle multi-stage games. This is also true of more contemporary MARL algorithms, including that of Multi-Agent Deep Reinforcement Learning (MDRL or MADRL), which are found to be incredibly sensitive to changes in input [**?**]. Therefore, due to importance we place on the stability and robustness guarantees of our MAS, we focus on those whose dynamics have been studies, as discussed in the following section.

2.3.3 Multi Agent Learning Dynamics

Multi Agent Learning Dynamics (often referred to as Game Dynamics or Learning Dynamics) considers the problem of mathematically modelling Multi Agent Systems who adapt through repeated interact with one another. This model then serves to be able to predict the evolution of the system as well as to understand the trajectory of learning. Typically, this looks at considering whether or not the method is likely to converge towards a Nash equilibrium. This is generally a difficult problem to solve [**?**] for all but toy problems. [**?**] shows that the stable equilbrium and Nash equilibrium (NE) are not necessarily the same and, in fact, argue that stable points are more informative than NEs. Stability provides some guarantees against the stochastic nature of the environment since a stable equilibrium will always be returned to even after perturbations. This feature is extremely important in Safe and Trusted AI as it provides guarantees against undesired behaviour in real world environments.

The area of dynamics which has shown most promise in Multi-Agent Reinforcement Learning is that of evolutionary dynamics. This draws from the principles of Evolutionary Game Theory (EGT) which considers similar assumptions to that of MARL: agents are no longer required to be rational and play the game optimise their expected return through repeated play. Importantly, players have no knowledge of the others' payoffs [**?**]. In [**?**], Tuyls et al. determine the relation between the replicator dynamics concept of EGT (a differential equation defining the evolution of the proportion of a subgroup in an evolving population) and Q-Learning using Boltzmann probabilities as Q-values. The result was a dynamics equation which describes, for each action, the evolution of its selection probability which could even account for random exploration. There have since been a number of works which apply the same insight into different game types and MARL algorithms. In [**?**], these are broken into the following categories

- Stateless games with discrete actions. Here, stateless refers to the idea that the game is static and so the environment has no impact on the result. The aforementioned result [**?**] fits into this category.
- Stateless games with continuous actions. These consider more realistic MARL than the previous category by replacing each agent's strategy vector with a probability density function (pdf) over a continuous action space.
- Stateful games with discrete actions. This mostly considers stochastic games, where there are multiple states with probabilistic (usually Markovian) transitions between them. However, extensive form games, which considers more complex phenomena such as sequential moves and imperfect information, are also briefly mentioned.
- Stateful games with continuous actions. This is one of the more realistic assumptions considered.

However, the authors point out that this area is yet to see results, leaving it open for possible research.

Though the above have seen success through experimental validation in games with 2 players and (in general) 2 to 3 discrete actions, recent work has begun the consideration of improving the models towards more complex scenarios with larger agent populations. A recent example of this is [**?**], in which Hu uses a mean field approximation to model the dynamics of a population of Q-Learning agents. As a reminder, mean field (MF) approaches in MARL consider that an agent updates its strategy based on the mean effect of the population. The result is a system of three equations which describes the evolution of Q-values over a large population in a symmetric bi-matrix game.This presents an important first step in modelling the learning dynamics of large agent populations and has the scope to be expanded to systems of asymmetric games, heterogeneous populations and stateful games.

An advantage of determining the evolutionary dynamics of learning is that it can describe the expected behaviour in different game settings. This is particularly important to understand the convergence of the methods; certain games will often show cyclic behaviour even with the existence of a strict NE. For instance, Imhof et al. show in [**?**] that a repeated prisoner's dilemma game results in cyclic behaviour when considering the options of cooperation or defection.

2.4 Control Theory

The control theoretic perspective considers generating a set of control laws for the system. These are chosen with the aim to satisfy certain properties. The main properties are

- *Stability*: a system will return to the desired setpoint (or within a neighbourhood) if perturbed.
- *Robustness*: a system will perform its function in the presence of uncertainty and noise.
- *Optimality*: the system will achieve its goal in the best way possible. This is determined using a loss function which the control laws must minimise.
- *Feasibility*: the controller will always be able to generate a control law which satisfies the desired properties. Many texts will refer to the well-posedness of the control problem which strictly requires that the control law: a) exist, b) be unique, c) vary continuously with initial conditions.

There are a number of approaches towards control systems. However, the interest of this review lies in multi-agent systems which must operate in the face of uncertainty. As such, we focus on stochastic and distributed control. The literature of the control community is vast and can be divided into a number of sub-fields. We consider Distributed MPC (DMPC) as a topic which is particularly relevant to this study and provide a review in this section.

2.4.1 A Brief Introduction to Model Predictive Control

We begin by providing a brief introduction to the theory underlying model predictive control (MPC). This begins with the theory of Optimal Control which, simply put, is the theoretical study of controls which minimise a given cost function subject to system dynamics. More formally, an optimal control problem is presented as

$$\begin{aligned} &\text{minimise } J(x,u) \\ \text{s.t. } &x(k+1) = f(x(x_0;k,u(k)),u(k)) \\ &u \in U_{ad}, \end{aligned} \tag{2.1}$$

where $J(x, u)$ is a given cost function, which is dependent on the state $x = x(x_0; k, u(k)) \in X$ and control $u(k)$, both of which are functions of the time step k. The first condition gives the dynamics of system (i.e. how the state evolves in time given the controls). Here, we show the dynamical system as an iterated map, though this may also be described through ordinary differential equations (ODEs) or partial differential equations (PDEs). The second condition places constraints on the controls u, which must lie within an admissible set. An example of this is bounded controls, in which case the admissible set is given by

$$U_{ad} := \{u \mid u_a(x) \leq u(x) \leq u_b(x) \ a.e. \ x \in X\} \tag{2.2}$$

Our goal is to find the optimal control $\bar{u}$ which satisfies the condition

$$J(\bar{x}, \bar{u}) \leq J(x, u) \ \forall u \in U_{ad}. \tag{2.3}$$

The trouble with optimal controls is that they are open-loop, in that the controls are determined for the model $x(k+1) = f(x(k; x_0), u(k))$ a priori and simply applied to the system. Of course, in the presence of noise, disturbances, or even errors in parameter estimation in the model itself, such a control scheme is not practical.

The idea of MPC is to reduce the open-loop optimal control problem to a series of finite horizon optimal control problems and solve these. For this reason, it is also referred to as 'Receding Horizon Control'. The control problem is, therefore, transformed to

$$\begin{aligned} \text{minimise } J_N(x, u) &:= \sum_{k=0}^{N-1} l(x(k), u(k)) \\ \text{s.t. } x(k+1) &= f(x(k), u(k)) \\ u &\in U_{ad}, \end{aligned} \tag{2.4}$$

where $l(x, u)$ are 'stage costs', defining the cost function at each time step of the horizon, which is of length N. At each time step, the system updates its measurement of its state x (which we have assumed here to be exact but may only be obtained through a measurement map). Solving this yields the optimal control sequence for this horizon $\bar{u}(0), ..., \bar{u}(N-1)$ and the first of these (i.e. $\bar{u}(0)$) is applied. Then, the system repeats this process at each time step.

2.4.2 Distributed Model Predictive Control

Rawlings et al. [**?**] divide the problem of distributed control into four distinct categories: decentralised control, non-cooperative control, cooperative control, and centralised control. The last of these is not considered in the text since it only considers the case in which a centralised controller has access and can manipulate multiple agents at once. This immediately violates the desired property of distributivity.

Decentralised control is the scheme in which agents do not have information about the actions of other agents and can only optimise for their own objective. This has the advantage of requiring no communication, but can often lead to poor performance when the agents are strongly coupled as each agents model is incomplete. In the non-cooperative setting, each agent optimises their own loss function whilst treating the others actions as a known disturbance. In this setting, each agent has knowledge of the others control laws and their effect. In this case, the agents must communicate their intended actions to one another and iterate to achieve a consensus (or Nash equilibrium). This is the same for cooperative control, except that the loss function is now shared across the team.

Of all of these systems, cooperative control has found greatest applicability in autonomous systems [**?**], perhaps due to the favourable stability properties [**?**]. One particularly strong application of this system is in vehicle platooning. Here, self-driving cars or unmanned aerial vehicles (UAVs) must move

in a certain formation without colliding into one another. A number of examples can be found such as in [?, ?, ?]. Distributed MPC provides the advantage that guarantees can be placed regarding coupled constraint satisfaction and feasibility.

The advances in stochastic and robust MPC, however, do lend themselves towards revisiting the capabilities of decentralised control. Recall that, in this scheme, the agents cannot communicate with one another. However, advances in MARL show us that this disadvantage can be made less prevalent if each agent has a model of the other [?]. To this end, a methodology such as presented in [?] may prove beneficial in the decentralised scheme. Here, the system chooses amongst a family of system models when choosing its control laws. Similarly, agents who determine (or perhaps learn) models of the other agents may be able to leverage this information, minimising the model error when optimising.

2.5 Swarms

Swarm systems comprise of a population of (typically) homogeneous agents who are able to organise themselves into formations using a series of simple local interactions with their neighbours [?]. Whilst the individual agents are generally simplistic, the collective behaviour may exhibit complex phenomena emulating systems observed in biological organisations such as bee or ant colonies [?]. Hybrid algorithms such as in [?] show an accelerated performance in reaching globally optimal solutions in search-based tasks. The advantage of many swarm algorithms is that they are based on local interactions and so are incredibly scalable [?].

Swarms are generally considered to be a special case of multi-agent systems and so, to this end, Sahin [?] puts forward five qualities which must be satisfied for a system to be considered a swarm. Note that Sahin's work considers swarm robotics and its applicationns, though these qualities apply to any agent type (computational or biological).

1. Autonomous agents: the agents can, independently, interact with the environment
2. Large Number of Agents
3. Few Homogenous groups of agents: Agents are considered homogenous if they follow similar rules and perform the same function.
4. Relative incapability of the agent: The agents should not be able to carry out the task on their own, but rather efficient and robust handling of the task should emerge from the collaboration of the agents.
5. Local sensing and communication: The agents should only have local sensing and communication capabilities.

Sahin goes on to describe the typical domains of application for swarm systems, which broadly fall into four categories:

- Tasks that cover a region: Environmental Monitoring and Mapping, as well as Search and Rescue fall into these categories. These leverage the fact that there are many agents, in a distributed setting, to ensure complete coverage of an area.
- Tasks that are dangerous: An example would be scoping out an environment with potential hazards before human operators are involved.
- Tasks that scale-up or down: This leverages the fact that the agents are independent and the control methodology scales. These qualities allow the number of agents to be fluid and so it is possible to add more or less agents to a task as its requirements vary. An example might be providing emergency services to individuals in a disaster relief zone, in which initial population estimates may be revised as the situation progresses.

- Tasks that require redundancy: The fact that the system is not tied to any single point of failure means that a task in which an agent is vulnerable to attack (cyber or physical) may be tackled by a swarm.

2.5.1 Approaches to Swarm Control

Recent work has seen the advancement of swarms controlled in stochastic environments. This is particularly important for swarm intelligence in robots; many systems are developed with the motivation of search-and-rescue, in which the robot swarms will have to operate in environments where accounting for uncertainty is critical. To this end, swarm systems have seen the advent of stochastic control. In this, the swarm is modelled as a diffusive system using a stochastic equation, most often the Kolmogorov Forward Equation [**?**]. In [**?**], the author shows that this equation can be derived by considering local trajectories of smaller subsets of the swarm. Elamvazhuthi and Berman [**?**] extend this idea to other stochastic models for swarms, importantly considering an advection-diffusion-reaction model which allows for hybrid agents to switch between different modes of operation.

The above models come under the term mean-field models [**?**], a series of equations for stochastic forward processes (such as swarm foraging) in which, as the number of agents tends to infinity, the true macroscopic motion of the system tends to these equations. Importantly, however, these stochastic equations allow for an analysis of the swarm, as well as the ability to develop control laws. In [**?**], Li et al. show that these models can be used to develop control laws for robots in a swarm and drive them towards a target distribution. The method is shown to perform accurately both in simulation and on real robots. The advantage is that guarantees can be placed on the convergence and stabilisabiilty of the swarm towards the desired distribution. However, there appears to be significant scope to expand upon this methodology from a safety perspective. To begin with, the method does not consider inter-agent interactions and therefore does not formally guard against collisions. In [**?**], a similar problem is considered, although collisions are avoided by having the robot simply move in the opposite direction when encountering an obstacle. It is here that the mean-field models are not as strong since they do not take these local interactions into account. It must be noted, however, that the authors of [**?**] have identified this problem and are currently working towards its incorporation into the model.

It would, therefore, be of particular interest from a safety perspective to consider agent interactions. A starting point may be [**?**] which extends the dynamical model of a swarm system to include agent interaction. This presents a first step towards considering the swarm as composed of intelligent agents rather than mindless particles and, as the authors suggest, presents the possibility of applying game theoretic approaches towards swarms, which also gives the ability to consider heterogeneous swarm systems who interact with one another through repeated play. Similarly, it would allow for a stronger control perspective on swarm systems such as presented in [**?**], in which a model predictive control (MPC) scheme is presented for a leader-follower swarm system to achieve given tasks.

2.5.2 Decision Making in Swarms

The reduction of the complexity in interactions between agents also allows for the robots to perform other calculations on board. In [**?**], Pini et al. leverage this by considering adaptive task partitioning across swarms. This allows a swarm, in a decentralised manner, to deliberate whether to partition a task into its sub-tasks or to perform the task in its whole. As of now (to the best of my knowledge) the problem of partitioning general tasks into its N sub-tasks is unexplored. This, however, highlights another advantage of swarm systems; they are readily divided into sub-groups (as in [**?**]) to perform a *divide-et-impera* approach to solving problems [**?**]).

Furthermore, swarm systems may be designed in a leaderless manner and so do not require the use of a central controller [**?**]. This presents the advantage that the system can rapidly adapt to the loss of agents or separation of groups throughout the task. However, the assumptions made regarding the homogeneity

of individual agents and the simplicity of their local interactions result in significant limitations placed on the complexity of the tasks that swarm systems can accomplish.

2.6 Discussion

Each of the above methods present different advantages in their approach towards MAS, as well as a particular set of disadvantages.

Game Theoretic methods have long provided a rigorous approach for modelling the behaviour of any MAS. It places no limitation on the heterogeneity of the agents or the task demanded. An appropriate choice of payoff matrices will account for both of these specifications. Recent advances in POSGs have lifted the strong assumptions on prior knowledge that was required to achieve a Nash Equilibrium and Game Theoretic Control allows for such methods to be applied in continuous state spaces. Whilst the latter shows particular promise, game theoretic methods still fall short in terms of distributivity as they are heavily reliant on communication with a centralised decision maker to supply rewards, or within the team to determine each others' actions. They also require a priori knowledge of the situation in order to set up payoff matrices and so cannot be regarded as robust to environmental uncertainty.

Multi-Agent Reinforcement Learning builds on the latter two shortcomings of game theory by allowing the environment to supply rewards to agents who adapt their behaviour accordingly. MARL is proving to be a powerful method to control MAS, and is rapidly improving, but cannot yet be regarded as robust. Studies repeatedly show that small alterations or noise in the environment can result in large deviations in behaviour. Furthermore, few guarantees can be placed on the resulting behaviour of a MARL system, though progress is being made in closing this gap.

Control Theory, on the other hand, takes pride in the guarantees placed on the MAS such as: controllability, stability, and robustness. However, control techniques are often centralised or decentralised. Distributed methods of control is still an evolving field, particularly in terms of robustness to uncertainty. In order to establish guarantees, assumptions are also made on the specifics of the MAS, and so control theoretic methods have yet to evolve into general, heterogeneous systems with any given task.

Swarm systems have their strongest advantage in scalability and, in fact, swarm techniques are built to scale due to the distributivity and low computational load placed on each agent. This has often come with the assumption that agents in the swarm act as simple, homogeneous, particles who follow very simple rules of interaction. Recent work is showing that this is not a necessary assumption and, in fact, swarms can leverage heterogeneity and agent capabilities to ensure robustness and generalisation.

It is clear that each method presents its own set of advantages and disadvantages. In fact, the greater contributions are made from a unification of techniques (e.g., game theoretic control) which leverage the strengths of one technique to mitigate the drawbacks of another. To this end, we consider the unification of swarm control and multi-agent learning. The aim is to unify the strengths of swarm theory, MPC and game theory to study populations of agents who can make decisions on an individual basis but perform tasks through large scale collaboration. In the subsequent chapters, we propose a line of research which explores the properties of a system as well as establishing practical control methodologies along the way.

Proposed Research

Chapter 3

Introduction to Proposals

In the following chapters, we describe our proposed lines of research which aims to study the evolution of swarm systems with independent, learning agents under the influence of predictive controls. We detail the novel lines of research which this work aims to undertake, providing suggestions of the hypotheses posed by these lines. With this in mind the following studies are proposed and developed subsequently:

1. **Stability and Chaos in MARL:** in which we seek to understand the dynamics of agent strategies when using Q-Learning to learn a game through iteration. We will establish the stability, as a function of parameters, when learning general p-player N-action games. This study enables the appropriate selection of parameters and payoff matrices to ensure the stability of the strategies of a finite set of agents, such as leaders in a swarm.

2. **Dynamics of Mean-Field Q-Learning Games:** in which we examine the strategy dynamics for large populations of agents learning through iterated games and mean-field Q-Learning. We seek to understand the long term behaviour of such mean-field systems in terms of its strategy selection. This study extends the previous and allows for the stability of the strategies of a population of agents to be established.

3. **Model Predictive Control of Active Particles through Fields:** in which we investigate the interaction of a swarm of active particles with potential fields. We establish the stabilisability of an MPC scheme with defined stage costs, as well as an analysis of the suboptimality of the method.

4. **Incorporation of Intelligence in Control:** in which we adapt the dynamical system from the above point to include an interaction term, accounting for strategy selection by agents who learn through iterated games against one another. Interactions with an MPC scheme is then to be examined in a similar capacity.

The remainder of this report analyses each of these topics in turn, providing motivation and suggestions for the course of research. The first of these 'Stability and Chaos in MARL' is currently being studied and so is given its own Chapter (Chapter **??**) which includes a summary of the research.

Chapter 4

Stability and Chaos in MARL

This segment of research aims to achieve a deeper understanding of the strategy evolution of agents following a Q-Learning approach. This allows for guarantees to be placed on the behaviour of such agents, in particular the conditions under which the game will converge to a stable equilibrium.

It has long been established that, upon lifting the strong assumptions made by traditional game theory (such as the rationality of agents and complete information), that player strategies can result in much more complex behaviour than convergence to a Nash Equilibrium (NE). In fact, this is even true on what would commonly be regarded as 'simple' games such as tic-tac-toe and prisoner's dilemma [**?**, **?**]. These behaviours include: convergence to a unique equilibrium (though not always to an NE), convergence to one of multiple equilibria, limit cycles and chaos. These behaviours are shown in Figure **??**. Of these, the most preferable is, of course, convergence to a unique equilibrium, although it is still possible to study systems with multiple equilibria or limit cycles [**?**]. However, it would be difficult to control systems whose dynamics are governed by chaos (though research into controlling chaos is ongoing and rife with opportunity [**?**]) and so MARL techniques should avoid this. It would, therefore, be a useful endeavour to determine the conditions under which these sorts of behaviours arise.

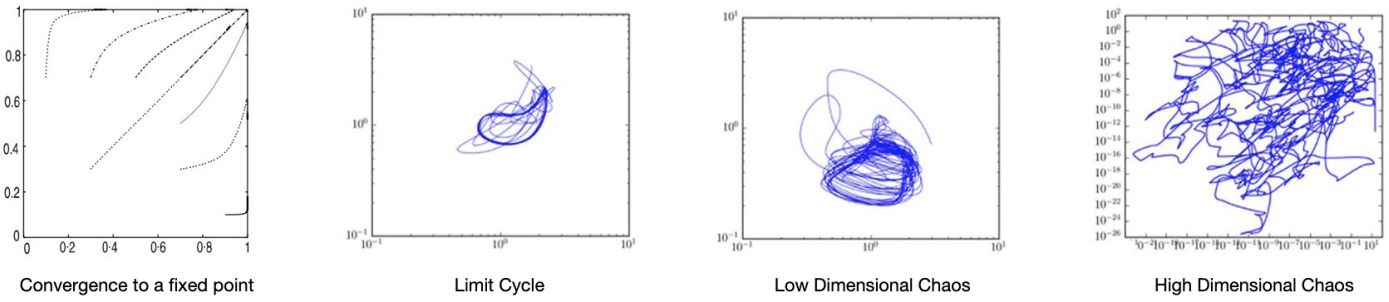


Figure 4.1: Different types of dynamical behaviour displayed by learning agents. a) Figure drawn from [**?**]. Convergence to a unique fixed point in the upper right corner (1, 1). This fixed point is unique, in that all trajectories, regardless of initialisation, will converge to this point. b) Limit Cycle, the trajectories converge to cyclic behaviour c, d) Chaotic behaviour, here small deviatons in the initial conditions can grow exponentially. b-d drawn from [**?**]

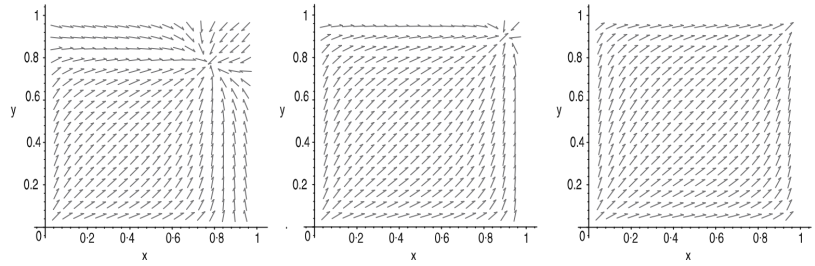
The behaviour of a system may be studied given a model of its dynamics. It is through this process that a wide array of physical systems, from harmonic pendulums to geophysical fluids, can be understood. A growing body of research aims to understand multi-agent reinforcement learning through the lens of its dynamics. In this light, Tuyls et al. [**?**] were able to derive a model of the strategy evolution of agents learning through iterated games. Through this, they were able to arrive at the following model of

Multi-Agent Q-Learning

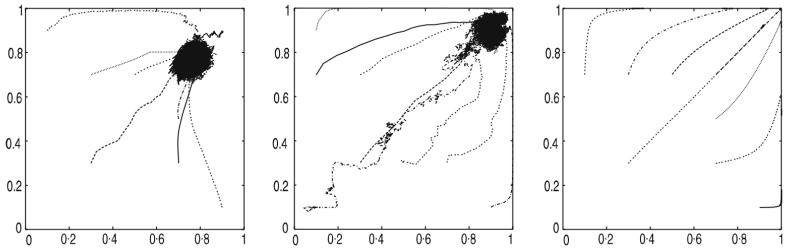
$$\frac{\dot{x}(t)}{x(t)} = \alpha\tau(\sum_j a_{ij}y_j - \sum_{ij} x_i a_{ij} y_j) + \alpha \sum_j x_j ln(\frac{x_j}{x_i}) \tag{4.1a}$$

$$\frac{\dot{y}(t)}{y(t)} = \alpha\tau(\sum_j b_{ij}x_j - \sum_{ij} y_i b_{ij} x_j) + \alpha \sum_j y_j ln(\frac{y_j}{y_i}). \tag{4.1b}$$

Here, α and τ are the parameters of the agent; Sanders et al. refer to these as the memory and intensity of choice parameters respectively. Agent 1 takes action i with probability x_i while Agent 2 takes action j with probability y_j. If these actions are taken, the agents receive payoff a_{ij} and b_{ji} respectively. With these equations, it is possible to predict the expected behaviour of Q-Learning agents, as shown in Figure **??**.



(a)



(b)

Figure 4.2: Figures taken from [**?**] a) Phase plot showing the expected behaviour of Q-Learning agents trained by iterating the Prisoner's Dilemma game as predicted by (**??**). From left to right, the agents have parameter $\tau = 1, 2, 10$. b) Corresponding trajectories of Q-Learning agents with randomised initial conditions displayed through numerical simulation. It is clear that the trajectories in b) follow the predictions in a), after stochasticity is accounted for.

It is clear, both from (**??**) and Figure **??**, that the long-term strategy selection of these agents is determined by the parameters α, τ and the payoffs a_{ij}, b_{ij}. We then pose the question: how do these elements influence the types of behaviours seen during learning on an iterated game? In other words, under what parameter selections are we likely to see convergence to unique equilibria, multiple equilibria, limit cycles or chaos?

With this in mind, the intention of this area of study is to consider the analysis presented in works such as [**?**] and [**?**]. Here, the authors examine the Experience Weighted Attraction (EWA) algorithm, which is regarded as a strong model for the learning behaviour of human players in a game [**?**]. The authors are able to determine the regions of parameter space in which complex behaviour, including cycles and chaos, predominantly occur and those in which a given game converges to a stable equilibrium. Figure **??** reproduces the graphs shown in [**?**] which illustrates the successful derivation of a 'phase line', across which learning shifts from convergent to chaotic.

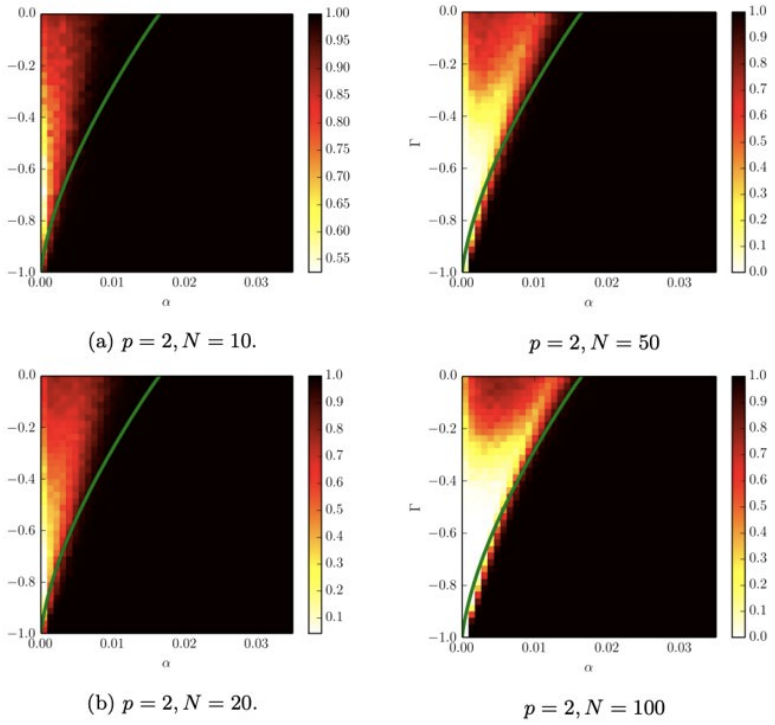


Figure 4.3: Results as produced in the supplementary material of Sanders et al [**?**]. Here, p is the number of players, N is the number of actions, Γ represents the competitiveness of the game (-1 represents zero-sum, 1 represents shared rewards) and α is the memory parameter of the agent as before. Here τ is held constant at 0.05 (though it is referred to as β in the paper). The black region represents where games converged to a fixed point through all simulations, whilst the hotter regions represent the more complex dynamics - red shows the presence of limit cycles and white regions signal the onset of chaos. The green line is a theoretical estimation for the phase line which separates the convergent dynamics from complex.

We propose to bridge the analysis presented by Sanders et al. and Galla et al. towards games learnt using MARL, particularly Q-Learning with Boltzmann exploration as considered by Tuyls et al. This would allow for a characterisation of the expected resultant behaviour under given parameters for a particular game. The vision for this is to provide guarantees on the stability of a finite set of agents within a swarm, such as leaders in a flock. Without this, it would be impossible to ensure the stability of a swarm in which intelligent, social interactions must be accounted for. In addition, this analysis would allow for a characterisation of the conditions under which MARL algorithms may feasibly be applied, thereby supporting the ability of researchers and engineers to choose their payoff matrices and parameters accordingly.

4.1 Summary of Research

In this chapter, we summarise and present the research and results which have been obtained thus far. These tackle the problems described thus far in this chapter. We summarise the derivation of a stability phase line as presented in Sanders et al. [**?**] as well as the numerical simulations performed to verify these results. These details of the derivation are provided in Appendix **??**. The steps to be completed subseqent to this study are then given. Note that the results in the section form the basis of a tentative submission to AAAI 2021.

4.2 Convergence and Chaos in Q-Learning Agents

We begin by considering the evolution of learning with agents who follow a Q-Learning approach. Specifically, we look at characterising the stability of agents learning strategies. This technique is carried out by Sanders et al. [**?**], in which the authors characterise the strategy evolution of agents who learn using an 'Experience Weighted Attraction' (EWA) algorithm, which has been shown to be a strong representation of how people learn in games [**?**, **?**]. The authors are able to determine the regions of parameter space in which learning is likely to converge towards stable equilibria, as opposed to complex behaviours (such as limit cycles or chaotic dynamics). The aim of this study is to format these techniques for the study of computational agents who follow the popular Q-Learning approach [**?**, **?**].

We aim, similarly, to determine the regions of parameter space in which the agents are likely to converge to stable equilibria. In this way it will be possible to, a priori, determine whether, for a particular game, the behaviour is likely to converge and even how to choose the agent or game parameters to ensure the predicatability of the resulting behaviour. To achieve this, we consult the Q-Learning Dynamics proposed by Tuyls et al [**?**]. In this study, the authors were able to derive a continuous time dynamical system describing how agents following a Q-Learning approach adjust the probabilities of choosing actions as they iteratively play a game. These equations are shown in [**?**] to accurately model the expected behaviour of agents as they iterate a game, and the experiments which verified this model are shown in Figures **??**.

With the accuracy of the continuous time dynamical system (**??**) established, we analyse the stability of this system in the following section. Note that whilst the derivation provided is for the particular case of two agents for the sake of brevity, the problem of p-player games is equivalent to that of two players and so the solutions to the former can (and will) be presented.

4.2.1 Dynamics of Q-Learning

We begin with the popular Q-Learning algorithm in which each agent keeps a record of a 'Q-Value' for each action, which is its own estimation of the performance of said action. This estimation is based on the expected reward that the agent will receive if it were to choose this action. In a multi-agent game setting, the reward that an agent receives will be drawn from a payoff matrix and will depend on the behaviour of the other players. Taking this into consideration, an agent updates their Q-Value of action i with the equation

$$Q_{t+1}(a_i) = (1-\alpha)Q_t(a_i) + \alpha(\sum_j a_{ij}y_j + \gamma max_{a_j} Q_t(a_j)). \tag{4.2}$$

Here, $\alpha \in [0,1]$ is a parameter of the agent which is considered the *memory* of the agent. For low values of α, the agent places a higher weight on their pre-existing choice for $Q_t(a)$, and so is considered to have a longer memory, whereas for higher values of α, the agent has a higher propensity to disregard previous estimations in favour of new information. Note that the term $\sum_j a_{ij}y_j$ denotes the expected reward that the agent will receive for selecting action i, drawn from its payoff matrix A, given the probabilities that its opponent chooses any action j. At each iteration of the game, our agent selects an action to play randomly with probabilities given by

$$x_i = \frac{e^{\tau Q_t(a_i)}}{\sum_j e^{\tau Q_t(a_j)}}. \tag{4.3}$$

This introductes a new parameter, $\tau \in [0, \infty)$, which is considered the *intensity of choice* by Sanders et al. [**?**]. When τ is close to zero, each action i is played with the same probability, regardless of its Q-value. However, for large values of τ, the action with the highest Q-value dominates all others and is played with a large probability at each step. The value of τ, then, denotes an agent's propensity to explore its strategy space. Note that in this derivation, as in [**?**], the action probabilities of one agent are

denoted with x and those of its opponent are denoted with y. The payoffs received by these agents are given by a_{ij} and b_{ij} respectively.

Tuyls et al. use this Q-Learning formulation to determine the time evolution of x and y. Given a continuous time approximation of this behaviour, they arrive at the equation **??** which is repeated below for convenience.

$$\frac{\dot{x}(t)}{x(t)} = \alpha\tau(\sum_j a_{ij}y_j - \sum_{ij} x_i a_{ij} y_j) + \alpha \sum_j x_j ln(\frac{x_j}{x_i}) \tag{4.4a}$$

$$\frac{\dot{y}(t)}{y(t)} = \alpha\tau(\sum_j b_{ij}x_j - \sum_{ij} y_i b_{ij} x_j) + \alpha \sum_j y_j ln(\frac{y_j}{y_i}). \tag{4.4b}$$

We now seek to analyse the stability of these dynamics based on the payoff values and agent parameters (α, τ).

4.2.2 Derivation of stability line

The dynamics given by Tuyls et al. describe the time evolution of agent strategies for a given choice of a_{ij}, b_{ij}. However, our goal is to analyse the dynamics of any game, regardless of payoff elements. As such, we follow the approach of Sanders et al. in deriving the full set of 'effective dynamics' averaged over all possible realisations of the payoff matrices. We consider that these elements are drawn from a Multivariate Gaussian such that

$$\begin{aligned}\mathbb{E}[a_{ij}] = \mathbb{E}[b_{ji}] = 0 \\ \mathbb{E}[a_{ij}^2] = \mathbb{E}[b_{ji}^2] = 1 \\ \mathbb{E}[a_{ij}b_{ji}] = \Gamma.\end{aligned}$$

where Γ is a parameter which defines the 'competitiveness' of the game. For a two player game, $\Gamma \in [-1, 1]$ where -1 denotes a zero-sum game and 1 denotes a game with positive correlation across payoff elements. Taking this average (details given in Appendix **??**) yields

$$\begin{aligned}\dot{x}(t) &= x(t)(\Gamma\alpha^2\tilde{\tau}^2 \int dt'[G_y(t,t')x(t')] + \tilde{\alpha}\rho_x(t) + \alpha\tilde{\tau}\eta_x(t) + \tilde{\alpha}\tau\eta_x(t)\eta_y(t) + \sqrt{\Gamma}\tilde{\alpha}\tau\mu_x) \\ \dot{y}(t) &= y(t)(\Gamma\alpha^2\tilde{\tau}^2 \int dt'[G_x(t,t')y(t')] + \tilde{\alpha}\rho_y(t) + \alpha\tilde{\tau}\eta_y(t) + \tilde{\alpha}\tau\eta_x(t)\eta_y(t) + \sqrt{\Gamma}\tilde{\alpha}\tau\mu_y),\end{aligned} \tag{4.5}$$

Here, G, ρ, η, μ are correlation functions, generated due to the averaging process and the notation $\tilde{\cdot}$ denotes a term which has been scaled with respect to N (again, see Appendix **??** for method and notation). We now analyse the stability of this dynamical system by linearising about a fixed point. This yields the requirement that a fixed point x^* satisfy

$$0 = x^*\left[\Gamma\alpha^2\tilde{\tau}^2 x^* \int G_y(t-t')dt' + \tilde{\alpha}\rho_x^* + \alpha\tilde{\tau}\eta_x^* + \tilde{\alpha}\tau\eta_x^*\eta_y^* + \sqrt{\Gamma}\tilde{\alpha}\tau\mu_x\right], \tag{4.6}$$

and similary for y. The reader should be aware that, for the sake of brevity, only the equations for x are presented henceforth as those for y are equivalent.

The implication here is that the resulting value of x^* can take two solutions. However, one of these is $x^* = 0$, which is found to rarely occur [**?**] and we do not consider, while the second, non-trivial solution contains a term $\sqrt{\Gamma}$, which would yield complex values. Such a result, of course, is not possible for a quantity denoting probabilities and so it appears that the theory cannot hold for $\Gamma < 0$. However, numerical experiments will shed light on the likelihood of convergence and the islands of stability in parameter space for this region.

We take the Fourier transform of this system to determine the long term behaviour of the system after a disturbance. To neglect all transient behaviour (high frequency disturbances), we consider the equation at $\omega = 0$

$$\left\langle [|\mathcal{X}(\omega)|^2 \right\rangle = \left\langle |\Gamma\alpha\tilde{\tau}\mathcal{V}_x(\omega) + \tilde{\alpha}\tau\eta_x^*\mathcal{V}_y(\omega) + \tilde{\alpha}\tau\eta_y^*\mathcal{V}_x(\omega) + \sqrt{\Gamma}\tilde{\alpha}\tau\Delta(\omega) + \Xi(\omega)|^2 \right\rangle \frac{1}{\langle |\mathcal{A}(\omega, x^*)|^2 \rangle}, \tag{4.7}$$

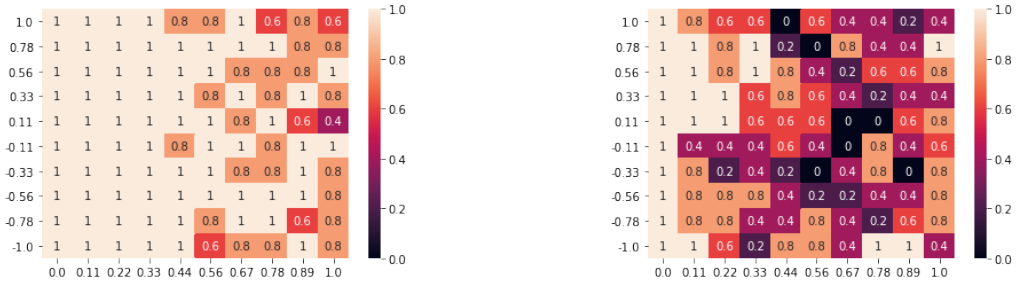
where

$$\mathcal{A}(\omega, x^*) = \frac{i\omega}{x^*} - \Gamma\alpha^2\tilde{\tau}^2\mathcal{G}_y(\omega) \tag{4.8}$$

By considering that $\left\langle |\mathcal{X}(\omega = 0)|^2 \right\rangle \geq 0$ we arrive at an expression for a stability line (i.e. the phase transition between the existence of stable fixed points and unstable behaviour) as the equality (**??**)> 0.

4.2.3 Numerical Experiments

To verify the results of our theory, and to examine the underlying structure of stability and chaos in Multi-Agent Q Learning, we perform a series of numerical experiments by varying the parameters Γ and α whilst keeping τ fixed. This yields the result shown in Figure **??**. We see from this, and from the further experiments shown in Appendix **??**, that the stability of the system is highly dependent on the value of τ. For low values, the system converges almost everywhere, whilst increasing τ to 0.15 decreases the probability of convergence significantly.



(a) $\tau = 0.05$ (b) $\tau = 0.15$

Figure 4.4: $\alpha = 0.1$, $\gamma = 0.1$, $\tau \in [0.1, 10]$, $\Gamma \in [-1, 1]$. Each simulation is run for 1 $\times 10^5$ iterations and tested for convergence. The game is said to have converged based on a tolerance of 1% difference between action probabilities. For each combination of τ, Γ the game is played 5 times, each with random payoff matrices and initial conditions. The average number of converged games (giving an indication of probability of convergence) is shown in each cell of the heatmap.

To generate the numerical simulations in Figure **??** we used the following procedure.

1. Fix the parameters α and γ. The latter is held fixed at 0.1 since it does not affect the long term behaviour of the system (it does not appear in (**??**)).
2. We initialise values of Γ and τ. These will be the variables which we sweep over.
3. Generate payoff matrices for both agents by sampling from a multi-variate Gaussian (variables are the payoff elements) with covariance parameterised by Γ.
4. Initialise the agents with random initial conditions (i.e. random action probabilities).
5. Allow the agents to learn over a maximum of 1×10^5 iterations.

6. Every 100 iterations, check to see if the action probabilities have changed significantly. If not (i.e., the change over the last 100 iterations is less than 1%) the learning process is considered to have converged.

7. This process is repeated 5 times with random payoff matrices generated based on the value of Γ. The probability of convergence is then recorded as $\frac{\text{number of times converged}}{5}$.

8. The values of τ and Γ are then modified and the process is repeated. The heatmap shows the probability of convergence for all values of τ and Γ which are tested.

Future Work

We are currently running a set of numerical experiments varying over all three parameters Γ, α, τ to identify any islands of stability in parameter space. We will then plot (**??**) to verify our results.

As mentioned, the above system considers only a two player game. This was done with the intention of simplifying the notation and reducing the complexity of the experiments. However, the next immediate action is to present the equivalent solutions for the case of a general p-player game.

We then seek to expand this study to a population of agents. For this, we will study the stability of the system presented by Leung et al [**?**] which presents a mean-field model describing the evolution of learning of a population of Q-Learning agents who play co-operative games against one another. This is discussed further in the subsequent chapter on proposed research.

Chapter 5

Proposals for Future Research

This chapter continues to describe the proposed research outined in Chapter **??**. These proposals are to be addressed following the completion of the research in Chapter **??**.

5.1 Large Population Dynamics

This segment of research aims to model the game dynamics of large populations of agents learning through iterated games and mean-field Q-Learning. The aim is to provide similar guarantees of stability as in the previous sections with the caveat that all agents in the population can learn, rather than a finite subset.

One of the main results shown by Sanders et al. [**?**] is that as the number of players in a game increases, the learning behaviour is more likely to be chaotic, regardless of the choice of parameters. This is intuitive since a higher number of players would result in a greater strategy space and more agents for any particular player to learn against and is verified by their presented results as in Figure **??** - the hotter regions occupy a larger area as p increases.

Yet it can be argued that, for large populations of agents (e.g., a crowd), the aggregate behaviour may be predictable, since the individual effect on the overall population is negligible [**?**]. This intuition is the foundation upon which crowd dynamics and flocking systems are based. In fact, the work presented by Leung et al. [**?**] provides a cursory verification of this intuition. Here, the authors present an analysis of the learning dynamics for a large agent population (which they approximate as containing infinite agents) where each agent is an independent Q-Learning using Boltzmann action selection. The result is a system of equations, governed by a Fokker-Planck model which is numerically shown to be a strong predictor of the overall strategy selection of the population. Indeed, rather than examining the strategies of every agent in the population (which the authors approximate as infinitely large), they model the evolution of a probability density function which, for each action, defines the density of agents who retain a given Q-value for that action. This is illustrated in Figure **??**.

As the authors point out, this is the first attempt at considering such a problem, and relies on heavy assumptions. The strongest of these is placed on the game itself, which is always assumed to be cooperative (in Sanders' terms, $\Gamma = 1$). We, therefore, propose extending this model to accomodate an arbitrary choice of games.

The following are suggestions for approaching the estimation of large population dynamics.

- Estimate the population state through Random Finite Sets (RFS) and apply the models from the previous study. The aim here is to determine a state estimation for a swarm of unknown size through a random finite set by solving an optimal estimation problem (typically iterated through a Kalman Filter [**?**]. As the swarm is treated probabilistically, the theory can accomodate an arbitrarily large system. Importantly, this may allow for the results from the previous section to be leveraged towards an arbitrary population.

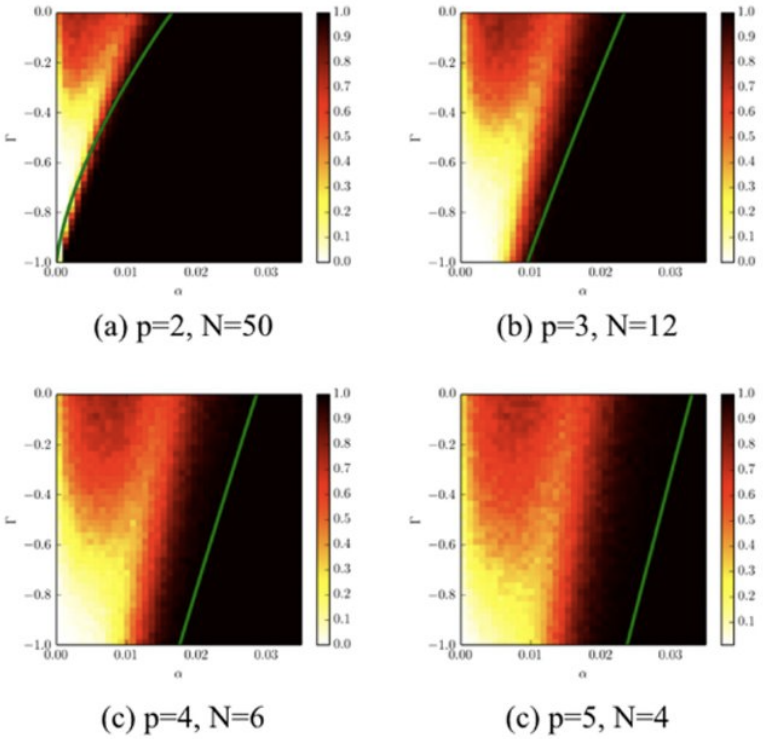


Figure 5.1: Results as produced in the supplementary material of Sanders et al [**?**]. The same items appear here as in Figure **??**, except the value of p (number of players) increases moving from top left to bottom right. It is clear that the hotter region occupies a larger region of parameter space, indicating that as the bumber of players in the game increase, chaos becomes more prevalent.

- Decompose a generic game into a weighted sum of a competitive and a cooperative game. Treat these as two separate agents and invoke the mean field approximation (MFA) so that any given agent is effectively engaged in a three-player game where the strategy of the opponents is the average strategy of the population. This leverages the existing results on the dynamics of three body problems in game theory [**?**] in a mixture of co-operative and competitive settings.

5.2 Swarm Control through Fields

Here we aim to study the interaction of a swarm of 'active particles' with a potential field. In particular this field is to be generated by 'field particles'. A Model Predictive Control (MPC) scheme is to be devised to drive this system to desired configurations, with guarantees placed on stability and satisfaction of input constraints. The sub-optimality of the control scheme is to be studied.

As described in Chapter **??**, the control of swarms has been examined through the use of mean field models. These models resolve the fact that it is impossible to view the entire system as simply the sum of all of the agents and rather model the overall state of the system through density functions. Controls are then applied to this function whose evolution is described through, typically, one of two partial differential equations: the Fokker-Planck Equation (also referred to as the Kolmogorov Forward Equation) and the Vlasov Equation. The former has seen some success in recent literature [**?**, **?**, **?**], though it makes the assumption that the system evolves through 'drifted brownian motion'. This means that the agents are considered to move independently and randomly, though under the influence of a field which affects their velocity. This technique has been shown, both theoretically and experimentally, to drive swarm systems

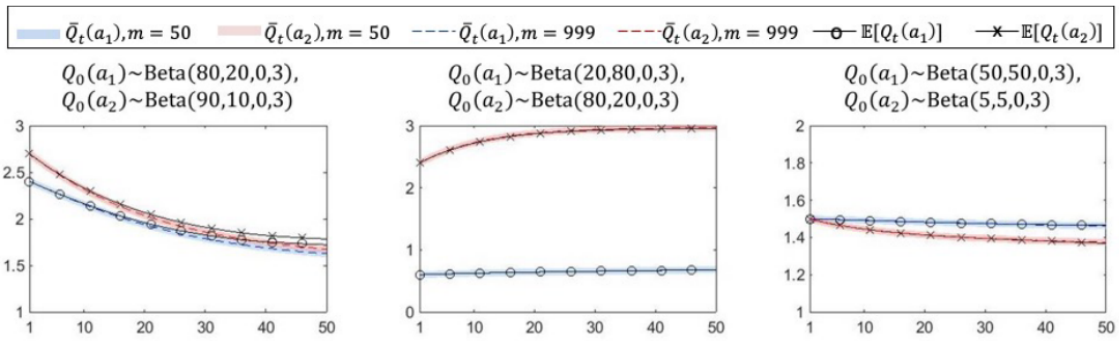


Figure 5.2: Results as produced in Leung et al. [**?**]. Here, the solid and dotted lines represent the evolution of the expectation of Q-values of two actions across a population of 50 or 999 agents respectively who are trained on an iterated Stag Hunt game. The circled and crossed lines represent the dynamics of these Q-values as predicted by the system of equations derived in the paper. These are seen to closely match the results of the numerical experiments.

in a stable manner [**?**]. The latter is beginning to show promise though control is typically through leadership rather than fields [**?**].

The hypothesis of this section is that the distribution of active particles, whose evolution is described through a Vlasov equation, may be controlled through the influence of a scalar field generated by 'field paricles'. This is drawn from the dynamics posed by Bellomo et al. [**?**]

$$\begin{aligned} \partial_t f + \vec{v}\cdot\nabla_{\vec{x}} f + \kappa\nabla_{\vec{v}}\cdot(F_a(f)f) = \epsilon Q(f,f) \quad (\vec{x},\vec{v}) \in \Omega[f]\times D_{\vec{v}}, \quad t>0, \\ F_a[f](t,\vec{x},\vec{v}) = -\int_{\Omega[f]\times D_{\vec{v}}} \psi(|\vec{x}-\vec{x^*}|)(\vec{v}-\vec{x^*})f(t,\vec{x^*},\vec{v^*})d\vec{x^*}d\vec{v^*}, \end{aligned} \tag{5.1}$$

Here, $f = f(t,\vec{x},\vec{v})$ is the one-particle probability density function at phase-space position $(\vec{x},\vec{v})$, $\vec{v}$ and time t which represents the state of the system. $\kappa \geq 0$ is a scalar coefficient, ψ denotes the communication strength between particles, and $(\vec{x^*},\vec{v^*})$ gives the position in phase space of a 'field particle'. It can be seen that the these field particles influence the velocity of the swarm agents.

The contribution that this study presents over those using brownian motion is the inclusion of an interaction operator $Q(f,f)$. This governs how agents interact with one another. With this term included, the dynamics account for the tendency of agents to avoid collisions. However, the term is intended to account also for social interactions between agents, which we aim to leverage in the subsequent section.

The complexity of this interaction term will need to be gradually increased over time. To begin with, we may entirely neglect this term. It is here that we will compare the model proposed by Zhang with that proposed by Bellomo et al. We may then consider only local interactions, which is typical for most swarming systems in the current literature, before then considering non-local interactions. This provides the scope for swarms to interact with a greater number of agents. At this stage, we may expand our study to consider the presence of interaction domains. This places constraints on how agents may interact with one another, allowing for a greater generalisation of inter-agent interactions. Ultimately, we would like to influence the interaction term, though this is discussed in Section **??**. The questions we will consider here are:

- The stablity of the system: Under what conditions is it possible to drive the swarm from one configuration to another in a stable manner? Fortunately, Bellomo et al. present a candidate lyapunov functional upon which dissipation can be evaluated, though it will need to be adapted for the consideration of the consensus models.

- The guarantees of the system: Is it possible to ensure that controls remain within an admissible set (typically a closed, bounded and convex set [**?**]).
- Whether the conservation laws of normalisation and total mass, as presented in [**?**] can be guaranteed.

These results will be established theoretically alongside the sub-optimality of the MPC scheme and verified through numerical simulations in a 2D environment. It will be of interest to perform a similar analysis to Ko and Zuazua [**?**] in which the cost functional is altered to favour particular metrics (e.g. running cost, control time etc.) and also analyse the effect of varying the time horizon.

5.3 Incorporation of Intelligence in Control

This section of the study is perhaps the strongest extension proposed in this chapter, and will likely be the most challenging. Here, we examine the dynamics (**??**) in which the interaction term accounts for the social dynamics of the decision making of individual agents who have learnt through an iterated game. We leverage the strategy evolution dynamics and stability analysis established in **??** and **??**.

Two proposals for incorporation of social interaction:

- Addition of the agent decisions as a constant in the state f. This will require that the decisions remain fixed in the state and so it will be required to make sure that the controls have no influence on this element.
- Weighted summation term in the probability of an agent changing its velocity from v_* to v due to a field particle with velocity v^* $A[f](v_* \to v|v_*, v^*)$. Here, weights are equal to the agent strategy. This extends to both the swarms in which the leaders are making decisions (only leaders have this term) as well as in swarms where the population makes decisions (everyone has this term). This will need to be done with care though to ensure that normalisation holds. Note that, as this a probabilistic approach, we will need to examine the stochastic behaviour of the system and the extent to which constraint and conservation satisfaction can be guaranteed.

The MPC scheme from the previous section will then be adapted to this dynamical system with the same questions of well-posedness and stability considered.

A possible extension of this would be to consider online learning by incorporating the time evolution of the strategy space within the social interaction term, which would now be time-varying. The proper integration of these dynamical systems would depend on the form of the systems derived in Section **??** and it is likely that relaxations would be required in terms of controllability results. An example would be to examine only the behaviour in the long-enough time frame, which means that we could not impose a final time. Of course, this means that we could only consider optimal control rather than model predictive. However, without establishing the models from Section **??**, this extension is speculative and it is likely that such an extension would fall outside the scope of the PhD.

Chapter 6

Research Timeline

This chapter outlines a tentative timeline of the research described thus far. The numbers in brackets denote the estimated months of the PhD that these are likely to take place with zero being the beginning of the PhD (October 2019) and 48 marking the end (September 2023). Note that there is a period of six months at the end which exists to allow for a margin of error.

1. Characterisation of Complex Dynamics in Reinforcement Learning Games (for submission to AAAI 2021) (**9 - 12**)
    - Determining and plotting a simplified expression for a stability line general p-player games using Q-Learning (**9 - 10**).
    - Running large scale simulations of Q-Learning games to experimentally verify results. (**9 - 11**).
    - Characterising the Lyapunov exponent for modes of unstable behaviour in Q-Learning Dynamics (**11 - 14**)
2. Control of Swarms with Independent Agents (**11 - 18**)
    - Setting up a model predictive control problem for the reduced assumptions of particles with no (or limited) interactions (**11 - 12**) and solving the resulting optimisation problem (**12 - 18**).
    - Designing and running simulation code. (**15 - 18**)
3. Characterisation of the Learning Dynamics of Co-operative Mean Field Games (**11 - 18**)
    - Evaluating the stability of co-operative ($\Gamma = 1$) mean-field games (**11 - 13**)
    - Evaluating the Lyapunov exponent of the unstable behaviour (**13 - 16**)
    - Running large scale simulations to verify results (**15 - 18**)
4. Control of Interacting Swarming Agents (**19 - 30**)
    - Setting up and solving MPC optimisation problem on the collisional Vlasov equation with short-range interactions between agents. (**19 - 25**)
    - Designing and running simulation code. (**24 - 30**)
5. Control of Co-operative Intelligent Swarming Agents (**28 - 34**)
    - Setting up and solving MPC optimisation problem on the collisional Vlasov equation with augmented interaction term constrained on learning dynamics. (**28 - 33**)

- Designing and running simulation code. (**32 - 37**)

6. Thesis Completion (**34 - 48**)
    - Writing Up (**34 - 42**)
    - Proof Reading and Editing (**42 - 44**)
    - PhD Completion (**44 - 48**)

Appendices

Appendix A

Research Summary

We start with the two-agent Q-Learning dynamics as presented by Tuyls et al.

$$\frac{\dot{x}(t)}{x(t)} = \alpha\tau(\sum_j a_{ij}y_j - \sum_{ij} x_i a_{ij} y_j) + \alpha \sum_j x_j ln(\frac{x_j}{x_i}) \tag{A.1a}$$

$$\frac{\dot{y}(t)}{y(t)} = \alpha\tau(\sum_j b_{ij}x_j - \sum_{ij} y_i b_{ij} x_j) + \alpha \sum_j y_j ln(\frac{y_j}{y_i}). \tag{A.1b}$$

Here, α and τ are the parameters of the agent; Sanders et al. refer to these as the memory and intensity of choice parameters respectively. Agent 1 takes action i with probability x_i while Agent 2 takes action j with probability y_j. If these actions are taken, the agents receive payoff a_{ij} and b_{ji} respectively.

A.1 Rescaling of Variables

In order to follow the conventions of spin glass theory for the analysis of disordered systems, we rescale the system so that the payoff matrix elements are of order $N^{-1/2}$. The motivation for doing this is that, along the way, we will take the limit of the number of actions N to infinity. However, doing this will result in numerical underflow of the action probabilities (i.e. the probability of each action goes to zero). To compensate for this, we adjust the system so that the sum of the probabilities add to N. The scaling goes as follows:

$$a_{ij} = \sqrt{N}\tilde{a_{ij}} \tag{A.2a}$$

$$b_{ji} = \sqrt{N}\tilde{b_{ji}} \tag{A.2b}$$

We compensate for this change with

$$x_i = \tilde{x_i}/N \tag{A.3a}$$

$$y_i = \tilde{y_i}/N. \tag{A.3b}$$

This gives the original equations as

$$\frac{\dot{\tilde{x}}_i(t)}{\tilde{x}_i(t)} = \alpha\tilde{\tau}\sum_j \tilde{a_{ij}}\tilde{y_j} - \tilde{\alpha}\tau\frac{1}{\sqrt{N}}\sum_{ij}\tilde{x_i}\tilde{a_{ij}}\tilde{y_j} + \tilde{\alpha}\sum_j \tilde{x_j} ln(\frac{\tilde{x_j}}{\tilde{x_i}}) \tag{A.4a}$$

$$\frac{\dot{\tilde{y}}_i(t)}{\tilde{y}_i(t)} = \alpha\tilde{\tau}\sum_j \tilde{b_{ij}}\tilde{x_j} - \tilde{\alpha}\tau\frac{1}{\sqrt{N}}\sum_{ij}\tilde{y_i}\tilde{b_{ij}}\tilde{x_j} + \tilde{\alpha}\sum_j \tilde{y_j} ln(\frac{\tilde{y_j}}{\tilde{y_i}}). \tag{A.4b}$$

The need for the factor of $\frac{1}{\sqrt{N}}$ is to follow the conventions of the saddle point method of integration. This will become clear after taking the expectation of the generating functional. Note that we may drop

the notation on time dependence for x and y. However, these will always be functions of time. Henceforth, we shall not write the tildes on x, y, a_{ij}, b_{ij}. We shall also abbreviate the final term as

$$\rho_{x,i}(t) = \sum_j x_j ln(\frac{x_j}{x_i})$$

A.2 Generating Functional

The generating functional allows us to take a path integral over all possible realisations of learning [**?**]. This is given as:

$$Z = \int D[\vec{x}, \vec{y}] \prod_i \delta(\textit{equation of motion}_i) exp(i \int dt[x_i(t)\psi_i(t) + y_i(t)\phi_i(t)]), \tag{A.5}$$

where the equations of motion are the Lagrange equations of motion given in (**??**) and the fields $\vec{\psi_i}(t)$ and $\vec{\phi_i}(t)$ will be set to zero at the end of the calculation. δ denotes the Dirac delta function. We write this in its Fourier representation, which yields

$$\begin{aligned} Z(\vec{\psi}, \vec{\phi}) = \int D[\vec{x}, \vec{\hat{x}}, \vec{y}, \vec{\hat{y}}] exp(i \sum_i \int dt[\hat{x}_i(\frac{\dot{x_i}(t)}{x_i(t)} - \alpha\tilde{\tau} \sum_j a_{ij} y_j + \tilde{\alpha}\tau \frac{1}{\sqrt{N}} \sum_{ij} x_i a_{ij} y_j - \tilde{\alpha}\rho_{x,i}(t) + h_{x,i}(t))]) \\ \times exp(i \sum_i \int dt[\hat{y}_i(\frac{\dot{y_i}(t)}{y_i(t)} - \alpha\tilde{\tau} \sum_j b_{ij} x_j + \tilde{\alpha}\tau \frac{1}{\sqrt{N}} \sum_{ij} y_i b_{ij} x_j - \tilde{\alpha}\rho_{y,i}(t) + h_{y,i}(t))]) \\ \times exp(i \sum_i \int dt[x_i(t)\psi_i(t) + y_i(t)\phi_i(t))]), \end{aligned} \tag{A.6}$$

Here, the terms a_{ij}, b_{ij} are the payoffs in the game and the term h denotes a field which will be set to zero at the end of the calculation. We recall that these are randomly generated using a multivariate gaussian and then held fixed for the rest of the game. We call this 'quenched disorder'. Isolating these terms allows us to rearrange the above as

$$\begin{aligned} Z(\vec{\psi}, \vec{\phi}) = \int D[\vec{x}, \vec{\hat{x}}, \vec{y}, \vec{\hat{y}}] exp(i \sum_i \int dt[\hat{x}_i(\frac{\dot{x_i}(t)}{x_i(t)} - \tilde{\alpha}\rho_{x,i}(t) + h_{x,i}(t))]) \\ \times exp(i \sum_i \int dt[\hat{y}_i(\frac{\dot{y_i}(t)}{y_i(t)} - \tilde{\alpha}\rho_{y,i}(t) + h_{y,i}(t))]) \\ \times exp(i \sum_i \int dt[x_i(t)\psi_i(t) + y_i(t)\phi_i(t))]) \\ \times exp(i \sum_i \int dt[-\hat{x}_i\alpha\tilde{\tau} \sum_j a_{ij} y_j + \hat{x}_i\tilde{\alpha}\tau \frac{1}{\sqrt{N}} \sum_{ij} x_i a_{ij} y_j - \hat{y}_i\alpha\tilde{\tau} \sum_j b_{ij} x_j + \hat{y}_i\tilde{\alpha}\tau \frac{1}{\sqrt{N}} \sum_{ij} y_i b_{ij} x_j]). \end{aligned} \tag{A.7}$$

The only difference between (**??**) and (**??**) is that we moved the term containing the quenched disorder into a separate exponential. Since our aim is to take an average over all possible realisations of this disorder, we will only need to focus on the last exponential which we rewrite as

$$Q = exp(i \sum_i \int dt[-\hat{x}_i\alpha\tilde{\tau} \sum_j a_{ij} y_j + \hat{x}_i\tilde{\alpha}\tau \frac{1}{\sqrt{N}} \sum_{ij} x_i a_{ij} y_j - \hat{y}_i\alpha\tilde{\tau} \sum_j b_{ij} x_j + \hat{y}_i\tilde{\alpha}\tau \frac{1}{\sqrt{N}} \sum_{ij} y_i b_{ij} x_j]). \tag{A.8}$$

We will then separate the terms so that like sums are paired together

$$Q = exp(-i\alpha\tilde{\tau}\sum_{ij}\int dt[\hat{x}_i a_{ij} y_j + \hat{y}_j b_{ji} x_i]) \times exp(i\tilde{\alpha}\tau\frac{1}{\sqrt{N}}\sum_{ijk}\int dt[\hat{x}_i x_j a_{jk} y_k + \hat{y}_i y_k b_{kj} x_j]) \tag{A.9}$$

It should be noted that we have changed some of the subscripts on the terms. Since these terms are all multiplied together and we sum over the subscripts, the letters we choose are of no importance and we can exchange them freely. We will define both exponentials in Q as Q_1 and Q_2 respectively.

We are now ready to take the expectation of Q. To do this, we will exploit the fact that the payoff elements are Gaussian distributed and use the identity [**?**]

$$\int dz[e^{-A_2(z)+\vec{b}\cdot\vec{z}}] = (2\pi)^{k/2}(det(A))^{-1/2}e^{\omega(b)}, \tag{A.10}$$

where

$$A_2(z) = 1/2\sum_{ij} z_i A_{ij} z_j$$
$$\omega_2(z) = 1/2\sum_{ij} b_i(A)^{-1}_{ij} b_j$$

A.2.1 Expectation of Q_1

We can rewrite Q_1 as

$$Q_1 = \prod_{ij} exp(\vec{b}\cdot\vec{z}),$$

where

$$b \coloneqq [-i\alpha\tilde{\tau}\int dt[\hat{x}_i y_j], -i\alpha\tilde{\tau}\int dt[\hat{y}_j x_i]]^T$$
$$z \coloneqq [a_{ij}, b_{ji}]^T$$
$$A \coloneqq \Sigma^{-1}$$
$$\Sigma_{ij} \coloneqq Cov[z_i, z_j],$$

where Σ is the covariance of $\vec{z}$. We recall that the scaled system has payoff elements chosen so that

$$\mathbb{E}[a_{ij}] = \mathbb{E}[b_{ji}] = 0$$
$$\mathbb{E}[a_{ij}^2] = \mathbb{E}[b_{ji}^2] = 1/N$$
$$\mathbb{E}[a_{ij}b_{ji}] = \Gamma/N.$$

Applying identity (**??**) gives:

$$\mathbb{E}[Q_1] = \prod_{ij} exp(-\alpha^2\tilde{\tau}^2\frac{1}{2N}\int dtdt'[\hat{x}_i(t)\hat{x}_i(t')y_j(t)y_j(t') + \hat{y}_j(t)\hat{y}_j(t')x_i(t)x_i(t') + \Gamma\hat{x}_i(t)x_i(t')y_j(t)\hat{y}_j(t') + \Gamma\hat{y}_j(t)y_j(t')x_i(t)\hat{x}_i(t')]). \tag{A.11}$$

A.2.2 Expectation of Q_2

We take a similar approach with the following definitions

$$\begin{aligned} b &\coloneqq [i\tilde{\alpha}\tau \int dt[\hat{x}_i x_j y_k], i\tilde{\alpha}\tau \int dt[\hat{y}_i x_j y_k]]^T \\ z &\coloneqq [a_{jk}, b_{kj}]^T \\ A &\coloneqq \Sigma^{-1} \\ \Sigma_{ij} &\coloneqq Cov[z_i, z_j], \end{aligned}$$

Following the same procedure as for Q_1 yields

$$\begin{aligned} \mathbb{E}[Q_2] = \prod_{ij} exp(-\tilde{\alpha}^2\tau^2 \frac{1}{2N^2} \int dtdt'[&\hat{x}_i(t)\hat{x}_i(t')x_j(t)x_j(t')y_k(t)y_k(t') \\ &+\hat{y}_i(t)\hat{y}_i(t')x_j(t)x_j(t')y_k(t)y_k(t') \\ &+\Gamma\hat{x}_i(t)\hat{y}_i(t')x_j(t)x_j(t')y_k(t)y_k(t') \\ &+\Gamma\hat{y}_i(t)\hat{x}_i(t')x_j(t)x_j(t')y_k(t)y_k(t')]). \end{aligned} \tag{A.12}$$

We now define the correlation functions

$$\begin{aligned} C_x(t,t') &= N^{-1}\sum_i x_i(t)x_i(t') & C_y(t,t') &= N^{-1}\sum_i y_i(t)y_i(t') \\ L_x(t,t') &= N^{-1}\sum_i \hat{x}_i(t)\hat{x}_i(t') & L_y(t,t') &= N^{-1}\sum_i \hat{y}_i(t)\hat{y}_i(t') \\ K_x(t,t') &= N^{-1}\sum_i x_i(t)\hat{x}_i(t') & K_y(t,t') &= N^{-1}\sum_i y_i(t)\hat{y}_i(t') \\ A_{xy}(t,t') &= N^{-1}\sum_i \hat{x}_i(t)\hat{y}_i(t'). \end{aligned}$$

We then rewrite $\mathbb{E}[Q]$ as

$$\begin{aligned} \mathbb{E}[Q] = exp(-\alpha^2\tilde{\tau}^2\frac{N}{2} \int dtdt'[&L_x(t,t')C_y(t,t') + L_y(t,t')C_x(t,t') + 2\Gamma K_x(t,t')K_y(t',t)] \\ -\tilde{\alpha}^2\tau^2\frac{N}{2} \int dtdt'[&L_x(t,t')C_x(t,t')C_y(t,t') + L_y(t,t')C_x(t,t')C_y(t,t') \\ &+\Gamma A_{xy}(t,t')C_x(t,t')C_y(t,t') + \Gamma A_{xy}(t',t)C_x(t,t')C_y(t,t')]) \end{aligned} \tag{A.13}$$

We can introduce these correlation functions into the expectation which gives

$$\mathbb{E}[Q] = \int D[C_x, \hat{C}_x, L_x, \hat{L}_x, K_x, \hat{K}_x, C_y, \hat{C}_y, L_y, \hat{L}_y, K_y, \hat{K}_y, A_{xy}, \hat{A}_{xy}] exp(N(\Psi, \Phi, \Lambda)), \tag{A.14}$$

where

$$\begin{aligned} \Psi = i \int dtdt'[&\hat{C}_x(t,t')C_x(t,t') + \hat{L}_x(t,t')L_x(t,t') + \hat{K}_x(t,t')K_x(t,t') \\ &+\hat{C}_y(t,t')C_y(t,t') + \hat{L}_y(t,t')L_y(t,t') + \hat{K}_y(t,t')K_y(t,t') + \hat{A}_{xy}(t,t')A_{xy}(t,t')] \end{aligned} \tag{A.15}$$

$$\begin{aligned}\Phi = -\alpha^2\tilde{\tau}^2\frac{N}{2}\int dtdt'[L_x(t,t')C_y(t,t') + L_y(t,t')C_x(t,t') + 2\Gamma K_x(t,t')K_y(t',t)] \\ -\tilde{\alpha}^2\tau^2\frac{N}{2}\int dtdt'[L_x(t,t')C_x(t,t')C_y(t,t') + L_y(t,t')C_x(t,t')C_y(t,t') \\ +\Gamma A_{xy}(t,t')C_x(t,t')C_y(t,t') + \Gamma A_{xy}(t',t)C_x(t,t')C_y(t,t')]\end{aligned} \tag{A.16}$$

$$\begin{aligned}\Lambda = i\sum_i\int dtdt'[\hat{C}_x(t,t')x_i(t)x_i(t') + \hat{L}_x(t,t')\hat{x}_i(t)\hat{x}_i(t') + \hat{K}_x(t,t')x_i(t)\hat{x}_i(t') \\ +\hat{C}_y(t,t')y_i(t)y_i(t') + \hat{L}_y(t,t')\hat{y}_i(t)\hat{y}_i(t') + \hat{K}_y(t,t')y_i(t)\hat{y}_i(t') \\ +\hat{A}_{xy}(t,t')\hat{x}_i(t)\hat{y}_i(t')].\end{aligned} \tag{A.17}$$

We insert this expectation back into the original generating functional which gives

$$\mathbb{E}[Z(\vec{\psi},\vec{\phi})] = \int D[C_x,\hat{C}_x,L_x,\hat{L}_x,K_x,\hat{K}_x,C_y,\hat{C}_y,L_y,\hat{L}_y,K_y,\hat{K}_y,A_{xy},\hat{A}_{xy}]exp(N(\Psi,\Phi,\Omega + (\mathcal{O}(N^{-1})))), \tag{A.18}$$

where Ω includes all terms describing the time evolution of the system and is given by

$$\begin{aligned}\Omega = N^{-1}\sum_i log\int D[x_i,\hat{x}_i,y_i,\hat{y}_i]exp(i\int dt[\hat{x}_i(\frac{\dot{x_i}(t)}{x_i(t)} - \tilde{\alpha}\rho_{x,i}(t) + h_{x,i}(t))]) \\ \times exp(i\int dtdt'[\hat{C}_x(t,t')x_i(t)x_i(t') + \hat{L}_x(t,t')\hat{x}_i(t)\hat{x}_i(t') + \hat{K}_x(t,t')x_i(t)\hat{x}_i(t')]) \\ \times exp(i\int dt[\hat{y}_i(\frac{\dot{y_i}(t)}{y_i(t)} - \tilde{\alpha}\rho_{y,i}(t) + h_{y,i}(t))]) \\ \times exp(i\int dtdt'[\hat{C}_y(t,t')y_i(t)y_i(t') + \hat{L}_y(t,t')\hat{y}_i(t)\hat{y}_i(t') + \hat{K}_y(t,t')y_i(t)\hat{y}_i(t')]) \\ \times exp(i\int dtdt'[\hat{A}_{xy}(t,t')\hat{x}_i(t)\hat{y}_i(t')]]) \times exp(i\sum_i\int dt[x_i(t)\psi_i(t) + y_i(t)\phi_i(t))]).\end{aligned} \tag{A.19}$$

We will evaluate the path integral using the saddle point method for integration. In this method, we consider that the integration is dominated by the maximum of the function

$$f = \Psi + \Phi + \Omega,$$

and we take the limit as N extends to infinity. We therefore determine the relations which maximise this function. We find

$$
\begin{aligned}
\frac{\partial f}{\partial C_x(t,t')} &\implies i\hat{C}_x(t,t') = \frac{\alpha^2\tilde{\tau}^2}{2}L_y(t,t') + \frac{\tilde{\alpha}^2\tau^2}{2}(L_x(t,t')C_y(t,t') + L_y(t,t')C_y(t,t') + 2\Gamma A_{xy}(t,t')C_y(t,t')) \\
\frac{\partial f}{\partial L_x(t,t')} &\implies i\hat{L}_x(t,t') = \frac{\alpha^2\tilde{\tau}^2}{2}C_y(t,t') + \frac{\tilde{\alpha}^2\tau^2}{2}(C_x(t,t')C_y(t,t')) \\
\frac{\partial f}{\partial K_x(t,t')} &\implies i\hat{K}_x(t,t') = \alpha^2\tilde{\tau}^2\Gamma K_y(t',t) \\
\frac{\partial f}{\partial C_y(t,t')} &\implies i\hat{C}_y(t,t') = \frac{\alpha^2\tilde{\tau}^2}{2}L_x(t,t') + \frac{\tilde{\alpha}^2\tau^2}{2}(L_x(t,t')C_x(t,t') + L_y(t,t')C_x(t,t') + 2\Gamma A_{xy}(t,t')C_x(t,t')) \\
\frac{\partial f}{\partial L_y(t,t')} &\implies i\hat{L}_y(t,t') = \frac{\alpha^2\tilde{\tau}^2}{2}C_x(t,t') + \frac{\tilde{\alpha}^2\tau^2}{2}(C_x(t,t')C_y(t,t')) \\
\frac{\partial f}{\partial K_y(t,t')} &\implies i\hat{K}_y(t,t') = \alpha^2\tilde{\tau}^2\Gamma K_x(t',t) \\
\frac{\partial f}{\partial A_{xy}(t,t')} &\implies i\hat{A}_{xy}(t,t') = \tilde{\alpha}^2\tau^2\Gamma C_x(t,t')C_y(t,t').
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
\frac{\partial f}{\partial \hat{C}_x(t,t')} &\implies C_x(t,t') = \lim_{N\to\infty} N^{-1}\sum_i \langle x_i(t)x_i(t')\rangle_\Omega = -\lim_{N\to\infty}\sum_i \frac{\partial^2\mathbb{E}[Z(\psi,\phi)]}{\partial\psi_i(t)\partial\psi_i(t')}\bigg|_{\vec{\phi}=\vec{\psi}=\vec{h}=0} \\
\frac{\partial f}{\partial \hat{L}_x(t,t')} &\implies L_x(t,t') = \lim_{N\to\infty} N^{-1}\sum_i \langle \hat{x}_i(t)\hat{x}_i(t')\rangle_\Omega = -\lim_{N\to\infty}\sum_i \frac{\partial^2\mathbb{E}[Z(\psi,\phi)]}{\partial h_{x,i}(t)\partial h_{x,i}(t')}\bigg|_{\vec{\phi}=\vec{\psi}=\vec{h}=0} \\
\frac{\partial f}{\partial \hat{K}_x(t,t')} &\implies K_x(t,t') = \lim_{N\to\infty} N^{-1}\sum_i \langle x_i(t)\hat{x}_i(t')\rangle_\Omega = -\lim_{N\to\infty}\sum_i \frac{\partial^2\mathbb{E}[Z(\psi,\phi)]}{\partial\vec{\psi}(t)\partial h_{x,i}(t')}\bigg|_{\vec{\phi}=\vec{\psi}=\vec{h}=0} \\
\frac{\partial f}{\partial \hat{C}_y(t,t')} &\implies C_y(t,t') = \lim_{N\to\infty} N^{-1}\sum_i \langle y_i(t)y_i(t')\rangle_\Omega = -\lim_{N\to\infty}\sum_i \frac{\partial^2\mathbb{E}[Z(\psi,\phi)]}{\partial\phi_i(t)\partial\phi_i(t')}\bigg|_{\vec{\phi}=\vec{\psi}=\vec{h}=0} \\
\frac{\partial f}{\partial \hat{L}_y(t,t')} &\implies L_y(t,t') = \lim_{N\to\infty} N^{-1}\sum_i \langle \hat{y}_i(t)\hat{y}_i(t')\rangle_\Omega = -\lim_{N\to\infty}\sum_i \frac{\partial^2\mathbb{E}[Z(\psi,\phi)]}{\partial h_{y,i}(t)\partial h_{y,i}(t')}\bigg|_{\vec{\phi}=\vec{\psi}=\vec{h}=0} \\
\frac{\partial f}{\partial \hat{K}_y(t,t')} &\implies K_y(t,t') = \lim_{N\to\infty} N^{-1}\sum_i \langle y_i(t)\hat{y}_i(t')\rangle_\Omega = -\lim_{N\to\infty}\sum_i \frac{\partial^2\mathbb{E}[Z(\psi,\phi)]}{\partial\vec{\phi}(t)\partial h_{x,i}(t')}\bigg|_{\vec{\phi}=\vec{\psi}=\vec{h}=0} \\
\frac{\partial f}{\partial \hat{A}_{xy}(t,t')} &\implies A_{xy}(t,t') = \lim_{N\to\infty} N^{-1}\sum_i \langle \hat{x}_i(t)\hat{y}_i(t')\rangle_\Omega = -\lim_{N\to\infty}\sum_i \frac{\partial^2\mathbb{E}[Z(\psi,\phi)]}{\partial\partial h_{x,i}(t)\partial h_{y,i}(t')}\bigg|_{\vec{\phi}=\vec{\psi}=\vec{h}=0}.
\end{aligned}
$$

We implement these relations into the expression of Ω, and make the assumption that all actions i are independent and identically distributed (i.i.d.), which gives

$$\Omega = log \int D[x,\hat{x},y,\hat{y}]exp(i\int dt[\hat{x}(t)(\frac{\dot{x}(t)}{x(t)} - \tilde{\alpha}\rho_x(t))])$$
$$\times exp(-\int dtdt'[\frac{1}{2}\alpha^2\tilde{\tau}^2 C_y(t,t')\hat{x}(t)\hat{x}(t') + \frac{1}{2}\tilde{\alpha}^2\tau^2 C_x(t,t')C_y(t,t')\hat{x}(t)\hat{x}(t') + i\alpha^2\tilde{\tau}^2\Gamma G_y(t',t)x(t)\hat{x}(t')])$$
$$\times exp(i\int dt[\hat{y}(t)(\frac{\dot{y}(t)}{y(t)} - \tilde{\alpha}\rho_y(t))])$$
$$\times exp(-\int dtdt'[\frac{1}{2}\alpha^2\tilde{\tau}^2 C_x(t,t')\hat{y}(t)\hat{y}(t') + \frac{1}{2}\tilde{\alpha}^2\tau^2 C_x(t,t')C_y(t,t')\hat{y}(t)\hat{y}(t') + i\alpha^2\tilde{\tau}^2\Gamma G_x(t,t')y(t)\hat{y}(t')])$$
$$\times exp(-\int dtdt'[\tilde{\alpha}^2\tau^2\Gamma C_x(t,t')C_y(t,t')\hat{x}(t)\hat{y}(t')]). \tag{A.20}$$

Since this contains all of the information of the learning evolution, we consider Ω as an effective generating functional (and in fact we see that it has a similar structure to (**??**), without the existence of the fields ψ, ϕ). In particular we recognise this as the generating functional of the 'effective dynamics' given as

$$\begin{aligned}\dot{x}(t) &= x(t)(\Gamma\alpha^2\tilde{\tau}^2\int dt'[G_y(t,t')x(t')] + \tilde{\alpha}\rho_x(t) + \alpha\tilde{\tau}\eta_x(t) + \tilde{\alpha}\tau\eta_x(t)\eta_y(t) + \sqrt{\Gamma}\tilde{\alpha}\tau\mu_x)\\ \dot{y}(t) &= y(t)(\Gamma\alpha^2\tilde{\tau}^2\int dt'[G_x(t,t')y(t')] + \tilde{\alpha}\rho_y(t) + \alpha\tilde{\tau}\eta_y(t) + \tilde{\alpha}\tau\eta_x(t)\eta_y(t) + \sqrt{\Gamma}\tilde{\alpha}\tau\mu_y),\end{aligned} \tag{A.21}$$

with the self-consistency relations

$$G_x(t,t') = \langle\frac{\delta x(t)}{\delta\eta_x(t')}\rangle \qquad G_y(t,t') = \langle\frac{\delta y(t)}{\delta\eta_y(t')}\rangle \tag{A.22}$$
$$\langle\eta_x(t)\eta_x(t')\rangle = C_y(t,t') \qquad \langle\eta_y(t)\eta_y(t')\rangle = C_x(t,t') \tag{A.23}$$
$$\langle\mu_x(t)\mu_y(t')\rangle = C_x(t,t')C_y(t,t') \qquad \langle\mu_x(t)\mu_x(t')\rangle = \langle\mu_y(t)\mu_y(t')\rangle = 0 \tag{A.24}$$

A.3 Stability Analysis

We are now in a position to take the effective dynamics, which describes the evolution of the learning dynamics after averaging over all possible realisations of the payoff elements, and determine the stability of the system at fixed points. We will follow the procedure laid out by Opper et al [**?**]. First, we rewrite $x(t)$, $y(t)$ as perturbations about their fixed points. We will then analyse the stability of these fixed points.

Let

$$x(t) = x^* + \hat{x}(t) \tag{A.25}$$
$$y(t) = y^* + \hat{y}(t) \tag{A.26}$$
$$\eta_x(t) = \eta_x^* + \hat{\nu}_x(t) \tag{A.27}$$
$$\eta_y(t) = \eta_y^* + \hat{\nu}_y(t) \tag{A.28}$$
$$\mu_x(t) = \mu_x^* + \hat{\delta}_x(t) \tag{A.29}$$
$$\mu_y(t) = \mu_y^* + \hat{\delta}_y(t), \tag{A.30}$$

where terms denoted by $\cdot^*$ are the values that are taken at the fixed point and terms denoted by $\hat{\cdot}$ refer to deviations from the fixed point. This is not to be confused with the conjugate variable notation that we used in the previous section. We assume that these perturbations arise from additive noise, $\xi(t)$, $\zeta(t)$, drawn from the unit normal distribution which are applied to the dynamics. Rewriting the dynamics with all of these considerations gives

$$\begin{aligned}\frac{d}{dt}(x^*+\hat{x}(t)) &= (x^*+\hat{x}(t))(\Gamma\alpha^2\tilde{\tau}^2\int dt'[G_y(t,t')(x^*+\hat{x}(t'))] + \tilde{\alpha}\rho_x(t) + \alpha\tilde{\tau}(\eta_x^*+\hat{\nu}_x(t)) \\ &\quad +\tilde{\alpha}\tau(\eta_x^*+\hat{\nu}_x(t))(\eta_y^*+\hat{\nu}_y(t)) + \sqrt{\Gamma}\tilde{\alpha}\tau(\mu_x^*+\hat{\delta}_x(t)) + \xi(t)) \\ \frac{d}{dt}(y^*+\hat{y}(t)) &= (y^*+\hat{y}(t))(\Gamma\alpha^2\tilde{\tau}^2\int dt'[G_x(t,t')(y^*+\hat{y}(t'))] + \tilde{\alpha}\rho_y(t) + \alpha\tilde{\tau}(\eta_y^*+\hat{\nu}_y(t)) \\ &\quad +\tilde{\alpha}\tau(\eta_x^*+\hat{\nu}_x(t))(\eta_y^*+\hat{\nu}_y(t)) + \sqrt{\Gamma}\tilde{\alpha}\tau(\mu_y^*+\hat{\delta}_y(t)) + \zeta(t)).\end{aligned} \tag{A.31}$$

Considering only terms which are linear in the deviations yields

$$\begin{aligned}\frac{d}{dt}\hat{x}(t) &= (x^*+\hat{x}(t))\left[\Gamma\alpha^2\tilde{\tau}^2x^*\int G_y(t-t')dt' + \tilde{\alpha}\rho_x^* + \alpha\tilde{\tau}\eta_x^* + \tilde{\alpha}\tau\eta_x^*\eta_y^* + \Gamma\tilde{\alpha}\tau\mu_x\right] \\ &+x^*\left[\Gamma\alpha^2\tilde{\tau}^2\int G_y(t-t')\hat{x}(t')dt' + \alpha\tilde{\tau}\hat{\nu}_x(t) + \tilde{\alpha}\tau\eta_x^*\hat{\nu}_y(t) + \tilde{\alpha}\tau\eta_y^*\hat{\nu}_x(t) + \sqrt{\Gamma}\tilde{\alpha}\tau\hat{\delta}_x(t) + \xi(t)\right] \\ \frac{d}{dt}\hat{y}(t) &= (y^*+\hat{y}(t))\left[\Gamma\alpha^2\tilde{\tau}^2x^*\int G_x(t-t')dt' + \tilde{\alpha}\rho_y^* + \alpha\tilde{\tau}\eta_y^* + \tilde{\alpha}\tau\eta_x^*\eta_y^* + \Gamma\tilde{\alpha}\tau\mu_y\right] \\ &+y^*\left[\Gamma\alpha^2\tilde{\tau}^2\int G_x(t-t')\hat{y}(t')dt' + \alpha\tilde{\tau}\hat{\nu}_y(t) + \tilde{\alpha}\tau\eta_x^*\hat{\nu}_y(t) + \tilde{\alpha}\tau\eta_y^*\hat{\nu}_x(t) + \sqrt{\Gamma}\tilde{\alpha}\tau\hat{\delta}_y(t) + \zeta(t)\right].\end{aligned} \tag{A.32}$$

If we consider the long term behaviour of the system near a stable fixed point, then $\hat{x}(t)$ goes to zero as $t\to\infty$. This yields the requirement that a fixed point x^* must satisfy

$$0 = x^*\left[\Gamma\alpha^2\tilde{\tau}^2x^*\int G_y(t-t')dt' + \tilde{\alpha}\rho_x^* + \alpha\tilde{\tau}\eta_x^* + \tilde{\alpha}\tau\eta_x^*\eta_y^* + \sqrt{\Gamma}\tilde{\alpha}\tau\mu_x\right]. \tag{A.33}$$

The implication here is that the resulting value of x^* can take two solutions. However, one of these is $x^*=0$, which is found to rarely occur, while the second solution contains a term $\sqrt{\Gamma}$, which would yield complex values. If this is true, then it would imply that the learning algorithm will rarely converge for values of $\Gamma<0$. Numerical experiments will shed more light on the likelihood of convergence. If this ansatz turns out to be incorrect (i.e. we see significant convergence for $\Gamma<0$), it is unlikely that we will be able to solve for the stability line for this case.

We continue to follow the method of Opper et al. by taking the Fourier transform of (**??**). For both agents, the first term in square brackets is constant with respect to time and so will not affect the long term behaviour of the system. We can, therefore, ignore this and focus on the second term. For the sake of brevity, we will only write the equations for Agent 1 (i.e. for x), though the equations for y can be equivalently obtained. Taking the Fourier transform yields

$$\left[\frac{i\omega}{x^*} - \Gamma\alpha^2\tilde{\tau}^2\mathcal{G}_y(\omega)\right]\mathcal{X}(w) = \Gamma\alpha\tilde{\tau}\mathcal{V}_x(\omega) + \tilde{\alpha}\tau\eta_x^*\mathcal{V}_y(\omega) + \tilde{\alpha}\tau\eta_y^*\mathcal{V}_x(\omega) + \sqrt{\Gamma}\tilde{\alpha}\tau\Delta(\omega) + \Xi(\omega). \tag{A.34}$$

Then,

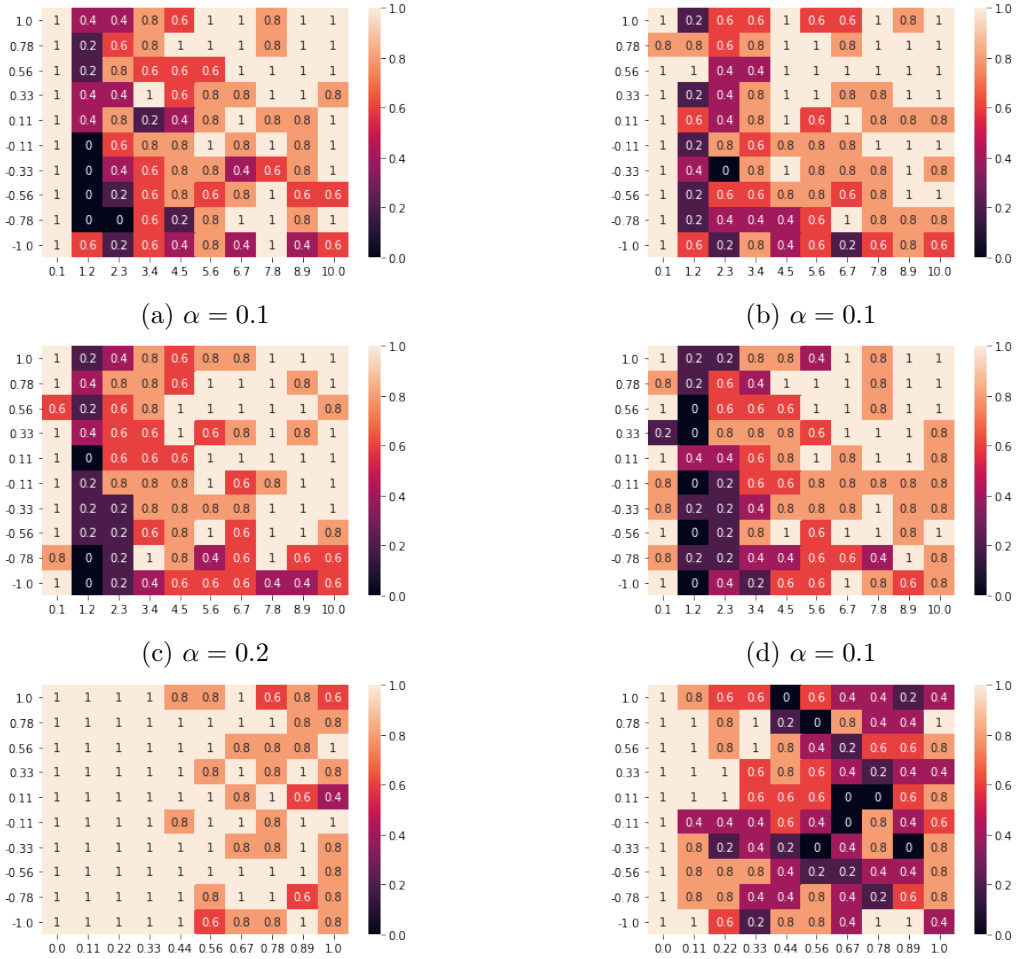
$$\left\langle[|\mathcal{X}(\omega)|^2\right\rangle = \left\langle|\Gamma\alpha\tilde{\tau}\mathcal{V}_x(\omega) + \tilde{\alpha}\tau\eta_x^*\mathcal{V}_y(\omega) + \tilde{\alpha}\tau\eta_y^*\mathcal{V}_x(\omega) + \sqrt{\Gamma}\tilde{\alpha}\tau\Delta(\omega) + \Xi(\omega)|^2\right\rangle \frac{1}{\langle|\mathcal{A}(\omega,x^*)|^2\rangle}, \tag{A.35}$$

where

$$\mathcal{A}(\omega, x^*) = \frac{i\omega}{x^*} - \Gamma\alpha^2\tilde{\tau}^2\mathcal{G}_y(\omega) \tag{A.36}$$

The behaviour of this line for $\omega = 0$ gives the long term behaviour of the system. By considering that, for a stable fixed point to exist, $\left\langle[|\mathcal{X}(\omega = 0)|^2\right\rangle$, we arrive at an expression for a stability line (i.e. the phase transition between the existence of stable fixed points and unstable behaviour).

A.4 Numerical Simulations



(a) $\alpha = 0.1$ (b) $\alpha = 0.1$

(c) $\alpha = 0.2$ (d) $\alpha = 0.1$

(e) $\tau = 0.05$ (f) $\tau = 0.15$

Figure A.1: $\alpha = 0.1$, $\gamma = 0.1$, $\tau \in [0.1, 10]$, $\Gamma \in [-1, 1]$. Each simulation is run for 1e5 iterations and tested for convergence. The game is said to have converged based on a tolerance of 1% difference between action probabilities. For each combination of τ, Γ the game is played 5 times, each with random payoff matrices and initial conditions. The average number of converged games (giving an indication of probability of convergence) is shown in each cell of the heatmap.

To generate the numerical simulations in Figure **??** we used the following procedure.

1. Fix the parameters α and γ. The latter is held fixed at 0.1 since it does not affect the long term behaviour of the system (it does not appear in (**??**)).

2. We initialise values of Γ and τ. These will be the variables which we sweep over.

3. Generate payoff matrices for both agents by sampling from a multi-variate Gaussian (variables are the payoff elements) with covariance parameterised by Γ.

4. Iniitialise the agents with random initial conditions (i.e. random action probabilities).

5. Allow the agents to learn over a maximum of 1×10^5 iterations.

6. Every 100 iterations, check to see if the action probabilities have changed significantly. If not (i.e. the change over the last 100 iterations is less than 1%) the learning is considered to have converged.

7. This process is repeated 5 times with random payoff matrices generated based on the value of Γ. The probability of convergence is then recorded as $\frac{\text{number of times converged}}{5}$.

8. The values of τ and Γ are then modified and the process is repeated. The heatmap shows the probability of convergence for all values of τ and Γ which are tested.