Literature Review - Revised

Aamal Hussain

January 6, 2020

Contents

Chapter 1

Introduction

This review attempts to present the current state-of-the-art in regard to solving the 'Multi-Agent Problem', which considers a scenario of multiple agents (such as robots, AI systems etc) must interact, whether cooperatively or competitively, to achieve defined goals. To this end, this review categorises work based on the approach towards solving this problem. Each approach often places assumptions on the type of agents present within the system and so each is appropriate in different settings. Briefly, these categories are

- Swarms, in which the agents are often assumed to be homogeneous and with limited sensing and communication capabilities.
- Decentralised-Partially Observable Markov Decision Processes (Dec-POMDPs), in which agents must choose actions with the aim to optimise a known loss function which is applied to the entire team.
- Game Theory, which allows for each agent to have an individual payoff function which they must optimise with respect to the actions of the other agents.
- Multi-Agent Reinforcement Learning, in which agents do not immediately have access to the payoff function and so must determine it through iterations of exploration.
- Control Theory, which considers the low-level operation of each agent and mathematically defines control laws for each agent which satisfy properties such as stability and controllability.
- Hard Coded, a term of the author's devising to categorise systems which do not fit in any of the formal methods above. These systems are built on a series of if-then statements, allowing agents to reason about their current state and future tasks.

1.1 Objective

The aim of the following sections is not to provide an exhaustive list of all work done in the aforementioned areas. To attempt to do this would be an exercise in futility. Instead, it is to identify research directions which lie within the broad scope of Multi-Agent Systems (MAS). Once identified, these directions will form the basis for the remainder of this review, allowing for particular problems to emerge. It will likely be the case that a medley of these problems will be addressed throughout the course of the PhD and, of course, more will likely be added.

Chapter 2

Swarms

Swarm based systems comprise of multiple homogeneous agents which are able to organise themselves through a formation 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 [**?**].

2.1 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 (divide and conquer) 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.

2.2 Approaches to Swarm Control

Swarms systems are often developed by considering their natural analogues, such as those seen in fish, birds, bacteria, or (most commonly) ants.

TODO: Find typical approaches towards swarm solutions.

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 Inoue et al 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.

2.3 Heterogeneous Swarms and Self-Healing

Recently, there has been an increased interest in introducing heterogeneity into the swarm systems to improve their real world applicability. An example of this which have been shown strong real world success can be found in [**?**]. Here, the authors consider two swarm teams, referred to as 'foot bots' and 'eye bots' who work in unison to explore an environment and solve a navigation task.

TODO: Do more research into heterogeneous swarm systems to pad this out. Perhaps we could introduce heterogeneity from the point of view of hybrid systems in the stochastic control section.

This area of research is sparsely populated and warrants further exploration. This is since the use of co-evolutionary teams can improve the robustness of the swarm optimisation. This is since it will be possible for a team to automatically determine when robots in the other team are not exhibiting expected behaviour and ensure that the other team self corrects. This process is referred to as 'Self-Healing' in [**?**]. Here, the authors allow a user to define the goals of a swarm system. From this, a 'trust' metric can be defined which measures the deviation of each agent from the expected behaviour. The self healing process occurs by limiting communication of all agents with 'untrusted' agents and encouraging communication with 'trusted' agents. The unison of self-healing with co-evolution may present the opportunity for heterogenous swarms to maintain their evolutionary stability, even in the face of environmental disturbances.

2.4 Fault Detection and Recovery

The above sections have considered the fact that swarm systems are robust to losses in the group. However, any MAS system must first be able to recognise that an agent has undergone some failure.

To this end, Tarapore et al. [**?**] develop a robust fault detection approach in which the swarm itself, in a decentralised manner, is capable of assessing deviation from 'normal' behaviour, even when the behaviour of the swarm itself is altered (perhaps by a remote operator). The authors achieve this by requiring that the agents themselves sense and characterise their own behaviour. This characterisation is formulated as a binary feature vector which is then communicated to the agent's neighbours. These neighbours will reach a consensus over whether the agent should be treated as faulty based on their collective behaviour. The results presented in [**?**] show extremely promising results and suggest that their method is, in fact, able to determine faults with high accuracy in the presence of various fault types (including sensor and actuator failures), although poor performance is seen in actuation failures in some instances. It should be noted that this method requires that each robot transmit their feature vector to the nearest neighbours. In environments where communication may be severely limited, this may present further errors. Furthermore, it is unlikely that, when a robot is damaged, only one of its components will be affected. Therefore, it is important to determine the effect on performance in the face of multiple agent failures and in communication losses. This exploration may open the possibility of improving the state of the art in terms of failure detection in swarm systems.

It may be interesting to consider failure from the point of view of stochastic control with jump dynamics (See 'Hybrid Control'). For instance, we could consider the development of systems which are robust to different modes of operation based on failure. Similarly an exploration into adaptive control could allow us to develop swarm systems who can recover from discrete jumps based on belief models.

Chapter 3

Dec-POMDPs

The use of POMDPs in multi-agent settings is formalised as decentralised POMDP (Dec-POMDP) which aims for a team of agents to maximise a common utility. However, it has been found that determining the exact solution to Dec-POMDP problems is NEXP [**?**] and so is intractable for all but toy problems. A number of methods have been presented to attempt to solve Dec-POMDPs. Oliehoek gives a review of these in [**?**]. Most solutions (such as brute force) are intractable for all but toy problems. Solutions which have found some tractability are

- Alternating Maximisation: This is effectively coordinate ascent for determining a joint policy; the optimal policy for one agent is found by keeping the others' actions fixed. This process is iterated across the entire team.
- Approximation as Bayesian Games: A Dec-POMDP can be approximated as a series of Bayesian (or Markov) Games. This is accomplished by giving each agent the same payoff in each game and allowing the payoff vector to transform Markovially at each time-step. By determining the solution to each of the games in turn, an approximate solution to the Dec-POMDP
- Selecting sub-tree policies: This considers standard tree search methods such as breadth first search or depth first search. This is used with the aim to reduce the number of possible policies which are searched over.

Whilst the above solutions are show lower complexity than the brute force approach, they are still limited in their tractability to large scale problems, perhaps with many agents or larger (even continuous) action spaces. This intractability largely occurs due to the fact that Dec-POMDP methods attempt to find optimal actions across the entire team at each time-step. This is in contrast to, for instance, swarm methods in which actions are chosen on a local basis and so computation is not affected by the size of the team.

Approximate solutions to Dec-POMDP have been proposed, perhaps most notable of which is the proposal of MacDec-POMDP [**?**] by Amato et al. Here, macro actions (actions which extend over multiple time steps) are used, as opposed to low-level actions which are re-evaluated at each time step. This allows an exact solution to be found as it does not need to be evaluated at each time step. This method assumes that, once macro-actions are distributed, the policies (sequence of state-action pairs) are known. Since this is not the case, Amato also proposes the use of a Dec-POSMDP [**?**], where 'SMDP' refers to 'Semi-Markov Decision Process, in which a high level model is defined without the underlying Dec-POMDP's actions and observations.

However, this is largely applicable in passive settings where common payoffs can be determined by an offline planner. They also require a significant amount of data with which to allow the system to learn the underlying models and payoff structures. This limits the applicability of the system when communication is limited and the system is presented with environments that it has not seen before. Recent work in MDPs [**?**] has considered learning in the face of Significant Rare Events (SREs) which the system has not yet observed. Currently, it is required that a model of such SREs is known and so it would be interesting to consider the application of Dec-POMDPs in situations where the SRE model is incomplete or erroneous and assess the robustness of the Dec-POMDP framework against such events.

Chapter 4

Game Theory

Game Theory has a rich history when considering an understanding of multi-agent systems. 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 section (although its synergy with swarms is requires significant development) since Dec-POMDP and MARL methods have both used game theory to support their frameworks. In fact, Dec-POMDP is a subset of Partially Observed Stochastic Games (POSG), in which all agents use the same payoff.

4.1 Market Based Methods

Garapati et al. [**?**] define a market based method as the setting where agents "follow their own interests and establish the mechanism of a market for distributing the tasks". Auctioning is the most widely used sub-field of market approaches and so I will use them interchangeably.

Whilst there are different variants to auctioning, the general procedure is that an auctioneer who has knowledge of a task (or multiple tasks) will set up an auction for said task. Agents can then make bids on these tasks and, once the auction is complete, the highest bid will win the task. In the specific application to robotics, a robot's bid will often reflect the costs, suitability or utility their undertaking the task [**?**]. This immediately highlights a few points. The first is that the method is not too heavily reliant upon a single processor to determine some joint policy. Tasks are allocated on a case-by-case basis and the utilities are calculated by the agent themselves. The only centralised process is the auctioneer's assessment of the winner which then relays this information back to them. The downside of this is that the system is heavily reliant upon strong communication channels, without which tasks may not be assigned, incorrect utilities may be communicated and, in general, sub-optimal solutions reached. Furthermore, the requirement that the agents themselves determine the cost of their actions assumes that they have the computational capability to do so. Furthermore, the bids placed by each agent need to be a strong representation of their capability to perform a task which may be hard to estimate without expert knowledge. However, market based methods are well suited to explanation through argumentation (similar to [**?**].

With well chosen payoffs, market based approaches work extremely well. For instance, in [**?**], the authors show that a free market approach (where agents try to maximise their own profits) can lead to a strong collaborative effort across teams. Similarly, in [**?**], Thomas et al. apply the auctioning scheme presented in [**?**] towards a robot construction team. However, it is important to note that these are both passive settings; tasks were assigned before the team were in the field and, in the case of [**?**], the system would repeat the bidding process if a robot failed. While both show strong performance, it cannot be said that either would be applicable in dangerous environments in which dynamic reassignment must happen within strict time constraints. Stancliff et al. [**?**] suggest that a more robust method to planning would be to account for failures a priori, a philosophy which is exemplified in [**?**] who consider the robot's reliability and relevance to a task as well as 'history relevance' which considers the relationship between pairs of robots with the aim of producing more effective teams.

There has also been some interesting work in probabilistic verification of market based approaches. Most notable to me is [**?**] which considers the case of conflict avoidance. Though their method focuses on collision avoidance, it highlights the need for verification of conflict resolution and goal achievement in market based approaches with different payoff structures. Sirigineedi et al. [**?**] make a step in this direction by considering the

verification of cooperative surveillance along a route network. From my understanding, this means that they were able to verify that their agents were able to traverse along the network without interference.

4.2 Stochastic Games

TODO: For the love of God do this already

4.3 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.

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, I will not include all of the numerous strides that Mylvaganam has introduced to the area. However, I must 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 is discussed in 'MARL'. The importance of the linear-quadratic differential game is the stability of the solution; solutions for the Nash equilibria 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.

Chapter 5

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.

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, I will try to consider the broad spectrum of methods which attempt to achieve this 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 chapter begins with a selection of the foundational methods which were developed towards solving the MARL problem. The interested reader may find a stronger review and analysis in [**?**]. I then go on to consider more contemporary approaches, most notably that of agent modelling.

5.1 Learning in Two Player Matrix 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.

5.2 Learning in Multiplayer 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 [**?**], whose algorithm we will consider shortly. Schwarz then presents a review of MARL methods (as of September 2014)

TODO: Consider the case for partially observed/decentralised settings. Find the relevant material within [**?**] which addresses this issue.

Agent Modelling

Returning to the problem of non-stationarity, solutions have been presented in which the agent models the learning of other agents. A noteworthy example of this is found in [**?**] in which the agent performs a one-step lookahead of the other agents' learning and optimises with respect to this expected return. They show that this leads to stable learning and can even lead to emergent cooperation from competition. However, the method requires that both agents have exact knowledge of the others' value functions in order to perform the one step lookahead. Furthermore, it has only been considered for the case of a two agent adversarial game and so the scalability of the system to multiple agents is not yet understood. Another method presented by Mao et al. [**?**] uses a centralised critic to collect the actions and observations of all agents and allows it to model the joint policy of teammates. This is shown to generate cooperative behaviour across four agents and so is more applicable to real world settings. However, its disadvantage over the method presented in [**?**] is that the critic is centralised. In real world settings, this requires the presence of an agent (perhaps a laptop) which is able to handle the computational load of determining a joint policy across all agents and must then communicate the Q-values of all agents back to them. This is both a taxing both in terms of computation and time.

Hong et al [**?**] present a similar system for modelling teammate policies by tasking a CNN with determining the policy features of other agents and then embedding these as features in its own DQN. This shows strong performance in settings where other agents dynamically change their policies. The concerns with this, however, are that, as the number of agents in the field increase, the CNN in each agent must perform another approximation. This places strong requirements on the performance of the CNN since errors in estimation will accumulate as the number of agents increases. Similarly, the complexity of the DQN will increase as more feature vectors are added.

Finally, all of the above methods are not robust to evolving numbers of agents. The problem of agent modelling is an important one to ensure stable learning and to understand the evolution of the system. It also presents a strong challenge and is open to exploration. To put it in context the methods described in this section are all from 2018-19, so its all very new.

The above methods are all centralised techniques, in which a controller determines the optimal joint policy for both agents. However, in real scenarios it is often preferable that each agent learns their own strategy, a task which must be completed without information of the other agent's strategy. The methods presented towards this problem are: linear reward-inaction (L_{R-I}) which guarantees convergence to NEs in games which contain pure NEs, linear reward-penalty (L_{R-P}) which can guarantee convergence to mixed strategies given the appropriate parameters, lagging anchor algorithm which also converges to mixed strategies, and the author's own proposal of the L_{R-I} lagging anchor algorithm which can converge to both pure and mixed NEs.

5.3 Game Dynamics

Game Dynamics (which I often refer to as Multi Agent 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. To extend this applicability into real world settings requires the study of stable equilbrium points; [**?**] 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 extremely important in Safe and Trusted AI as it provides guarantees against undesired behaviour in real world environments.

TODO: Add in the papers from Mendeley which are useful to this part of the study

5.4 Stable Learning

This section considers a dynamical systems approach towards learning agents. Its aim is to develop learning systems which prioritise stability.

Stability may be looked at from the view point of two perspectives. The first is from an optimisation point of view. This considers the dynamics of the learning model, allowing us to better choose our parameters and design our models so that they may converge to a stable result. The second is from the view point of the state-action space of a learnt model. This allows us to determine, before the MAS is deployed, which set of state-action pairs will lead to unstable behaviour. This knowledge allows us to consider which state-action pairs should be avoided. In both cases, stability analysis allows us to build multi agent systems which will learn and act in the way that we expect them to.

In [**?**], Letcher et al. model gradient descent learning of generative-adversarial-networks (GANs) as a two-player differentiable game. A differentiable game is one in which the loss function is twice differentiable. Using this formulation, they are able to analyse the system from new perspectives by considering the current state-of-the-art understanding of differentiable game theory. Whilst, at first glance, this may seem like a purely theoretical exercise, they go on to show that the insights gained allow them to develop a new multi-objective optimisation technique for GANs which shows stronger convergence properties, most notably of which is that it guarantees that the method finds a stable equilibrium (and avoids saddles) between the two players' loss functions.

Jin and Lavaei [**?**] consider the policy of a reinforcement learning agent as a non-linear, time varying feedback controller. Using this notion, they then consider the bounded-input-bounded-output stability of the system. They do this by analysing the ratio between the total output and total input energy (called the L2 gain). If the L2 gain remains finite then the system may be considered to be stable. They then apply these considerations on real-world applications including multi-agent flight formation and obtain stability certificates (essentially confirming that the system will remain stable under certain conditions) for the learned controller.

Berkenkamp et al. [**?**] consider a similar problem from a different definition of stability. Specifically, they look at stability from the point of view of Lyapunov functions. A system is said to be stable if the applying the policy will result in stricly lower evaluations of the function. In other words, a system is stable if its corresponding Lyapunov function is decreasing towards a minimum point. The authors use this idea to define a 'region of attraction' in which the system is stable in the sense of Lyapunov. The goal of Safe Lyapunov Learning, a method which they develop from these insights, is to learn a policy without leaving this region of attraction. They do this by taking measurements within the set and using this to learn about the system dynamics, thereby increasing the safe set. With this, we now have a guarantee that policy optimisation will not result in unsafe behaviour, even in the presence of stochasticity and exploration.

As we can see from the discussion thus far, there are many definitions and perpectives of stability, each lending to a new understanding of learning systems. To add another to the mix, Milchtaich in [**?**], presents a notion of static stability. This means that it is based solely on the incentives of the players and does not require a consideration of the dynamics of the system. Milchtaich's definition of a stable system is one in which, when perturbed, it is more beneficial for an agent to move back towards the equilibrium than it is for them to move away. This is perhaps the most fundamental definition of a stable equilibrium and, as such, Milchtaich shows that it is applicable to all strategic games, within certain assumptions (finite set of player and continuous strategy space) and considers probabilistic perturbations from the original state. This is particular applicable to Multi Robot Reinforcement Learning (or any MARL in continuous strategy spaces) since these systems require random exploration of the strategy space and the dynamics are not known a priori. However, the lack of dynamics means that we do not consider the evolution of learning.

TODO: In the above, look specifically at the assumptions made within the papers. In the multi-agent case, the biggest assumption is that of independent learning. Perhaps consider how advances in coupled dynamical systems may be used to consider learning agents whose actions affect one another.

Chapter 6

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, that a system will return to the desired setpoint (or within a neighbourhood) if perturbed.
- Robustness, that a system will perform its function in the presence of uncertainty and noise
- Optimality, that the system will achieve some defined function
- Feasibility, that the controller will always be able to generate a control law which satisfies the desired properties.

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 particular methodology that this chapter focuses largely on is Model Predictive Control (MPC) as STAI is particularly interested in model-based techniques for autonomous systems.

6.1 Model Predictive Control

Model Predictive Control is a long standing paradigm in AI which looks specifically at the problem of operating real-world agents safely in the face of environmental disturbances. The overarching idea begins with the assumpion that we have a model of the environment. As an example, in the case of autonomous vehicles, we have a model of how adjusting the angle of the front wheels will affect the heading of the car. We then perform a finite horizon look ahead, in which we estimate the environment state for a few time steps ahead, and generate a policy for this horizon. The agent performs the immediate action generated by the policy, and then we repeat the process, after taking measurements of the environment to determine the error in the system. Throughout this process, the controller (a.k.a. policy) must remain stable, but also satisfy constraints. From our previous example, a likely constraint would be that the controller will never result in the car going on the pavement, where it would present a real hazard to pedestrians. It is through these constraints, and the requirement of stability that the system is required to remain safe throughout operation. To that end, mathematical proofs of these properties are provided in the literature ensuring that we can trust in the system's performance.

Deterministic MPC assumes that the environment is completely deterministic and, therefore, assumes knowledge of the exact nature of the disturbances. This somewhat naive assumption simplifies computation and so appears often in the literature, as in [**?**]. However, it does not provide strong guarantees outside of games and simulations. Robust MPC brings this to a higher level of abstraction in which system perturbations, though deterministic, lie on a bounded set. Therefore, we no longer assume the exact nature of the environment and can guard against worst case scenarios. This is extremely effective in closed environments or for simple tasks, but falters in more complex environments.

Stochastic Model Predictive Control (SMPC) lifts the assumption made in MPC, namely that perturbations are deterministic and lie on a bounded set [**?**]. To accomplish this, we define chance constraints, for which probabilistic guarantees must be determined. Furthermore, optimality is defined in terms of the minimisation of the expectation

of a probabilistic cost function. This formulation brings about controllers which are more applicable and robust when deployed. However, the field is in its infancy and presents a number of theoretical challenges. These are best described by Mesbah in [?]

1. The arbitrary form of the feedback control laws
2. The non-convexity and intractability of the chance constraints
3. The complexity of the uncertainty propagations
4. Establishing stability of the control problem

Since the review [?] was written in 2016, there have been a number of approaches towards solving some of these problems, most notably the intractability of the chance contraints. For instance, in [?], Paulson and Mesbah propose the use of joint chance constraints in considering time varying stochastic disturbances as well as model uncertainty. This is shown to be strongly suited to non-linear systems, which itself is an open problem in the MPC sphere.

6.1.1 Distributed MPC

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.

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 agent's 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.

6.1.2 Hybrid Control

Hybrid control considers agents whose behaviour is governed not only by continuous time dynamics but also according to discrete switches [?]. This is particularly useful in systems which may have different modes of operations (such as autonomous vehicles with gear shifts or mechanical systems which undergo collisions). It is particularly useful for the case of systems which undergo discrete failures (e.g. burst tire in an self-driving car). It is important, therefore, that control systems be designed with the aim to account for such switching behaviour.

Mendes et al [?] present one of the first practical implementations of MPC for distributed hybrid systems. Here, a Mixed Integer Quadratic Program (MIQP) is transformed into a set of QPs by generating a controller for each of the feasible combinations of binary variables. An iterative search is then used to determine the best

solution out of this family of distributed control laws. The method is shown to reduce optimality slightly in favour of efficient computation. It would perhaps be interesting to consider distributed control from the perspective of fault detection and recovery, and consider how a system similar to that of Tarapore et al may be used in place of generating all possible controllers. Since Tarapore's method is developed for swarms, it maintains the advantages of low communication overheads but may prove advantageous in terms of communication.

Mendes' method, however, considers deterministic switching control. When considering faults, however, this can no longer be the case, since faults occur stochastically. To this end, it is important to consider stochastic switching dynamics. One proposed method is that of Jump Markovian Linear Systems (JMLS). In this, discrete 'jumps' can occur in a system which, otherwise is governed by continuous time dynamics [**?**]. To this end, Blackmore et al [**?**] develop a method towards predictive control of such systems using a particle filtering approach. The aim of this method is for the resultant system to remain within a safe set with a defined maximum probability which takes into account the discrete changes (which they consider as brake failures in an autonomous vehicle), as well as the usual continuous stochasticity which occurs due to noise and uncertainty. Their method is seen to remain robust to both of these and allows for safe operation (by constraint satisfaction) of an autonomous vehicle. Yin et al [**?**], similarly considering this problem, but for the specific case in which the transition probability of the MJLS is partially unknown. To this end, they similarly develop a controller which is independent of the mode of operation (and so can operate safely in any of those modes) but supplement it with a mode-dependent controller which drives the system initially. This allows for a better implementation than in [**?**] in terms of optimality, but at the expense of robustness and computation.

Stochastic Hybrid systems have yet to find their way into distributed predictive control settings but may prove critical for developing multi-agent systems which are robust to agent failures. Consider, for instance, the case of the platooning problem but with the added complexity that agents within the platoon may undergo failures in their braking system. Similarly, communication failures may occur when driving which may result in an agent having to operate with reduced communication capacity. Developing Stochastic Hybrid Distributed Systems may present the opportunity for autonomous systems to remain robust to known failures.

Chapter 7

Proposed Research

7.1 Considering Intelligence in Swarm Dynamics

Most of the diffusion models, as mentioned in 'Swarms', do not take into consideration the interactions between agents, since it adds a large degree of complexity to the model. However, it would be important from a safety perspective to take these into account by looking at the attractive and repulsive potentials across agents. This would allow us to provide a guarantee of inter-agent collision avoidance whilst maintaining the progression of the swarm towards desired tasks. As mentioned in 'Swarms', advances have been made towards considering the pairwise interactions between swarm agents [**?**]. Importantly, this addition presents the first step towards considering the intelligence of the individual within the swarm.

By expanding the control perspective of stochastic swarms, predictive control methodologies can be developed, similar to that of [**?**] which allow swarms to produce more complex phenomena such as constraint satisfaction. As an example of where this might be important, consider the problem of swarm systems in urban search and rescue, where one swarm sub-group is required to carry a victim away from the scene, whilst another is required to search the area. This can be viewed as from a control perspective where it is required that the latter group's macroscopic model ensures that agents do not enter within a safe region of the former group.

Stochasticity can then be considered from a hybrid perspective through JMLS (see Hybrid Control), allowing the control laws to take into account agent and communication failures. [**?**, **?**, **?**] have begun to make progress in these areas, but it does not seem like the connection of MJD and swarm dynamics has been made yet.

Finally, it may be appropriate to consider stochastic control of heterogenous swarms to allow for a 'marriage' between swarm dynamics and game dynamics. This presents the possibility for swarms sub-groups to adapt to each other's behaviour through repeatedly play and perhaps display elements of cooperative (or non-cooperative) behaviour, all while maintaining the required properties of stability and controllability.

7.2 Stochastic Predictive Control for Decentralised Hybrid Systems

The purpose of this method is to leverage the ideas presented in [**?**] and [**?**]. Here, agents must maintain (and perhaps learn) a model of the other agents within the system. This model provides some representation of the trajectories that the other agents will carry out or an understanding of their future actions. Using their own system models in tandem with these belief models, agents must then determine their trajectory. The challenge here would be proving the stability and controllability of the system whilst maintaining minimal communication requirements. However, it would appear to be a feasible area of exploration.

This system can then be expanded by considering the case of stochastic hybrid systems. This draws on the ideas presented in 'Hybrid Control' where we argue that agents in the system may be subject to stochastic failures. It is, therefore, required that we develop control methodologies which are robust to these switching dynamics to ensure their safe operation.

Similarly, it would be important to consider the learning aspect of these systems. The method so far assumes that each agent knows all possible modes of operation of other agents and their transition probability. However, this is unlikely to be the case. As such, it would be important to consider the ability of agents to learn these models through iterative interaction and learning. This may improve the robustness and optimality of the system,

whilst maintaining its desired properties.

On a similar topic of learning, it would be vital, from a failure perspective, to consider fault detection in a decentralised hybrid system. The above considers learning the system model of other agents who have undergone a faiure. However, it is equally interesting to consider learning *that* they have failed in the first place. The results suggested in 'Swarms' may be applicable here, in an effort to maintain the decentralisation of the system.

Finally, these methods could be extended towards settings where the resultant behaviour is not best described by a Nash Equilibrium. The control theory literature, thus far, has shown a strong assumption that the system will always converge to a Nash Equilibrium. However, in games of repeated play, the resultant dynamics of the game may be best described through cyclic or recurrent behaviour [**?**]. Taking this into consideration would allow distributed control to be applied in a wider array of settings, particularly in those where agent interests are conflicting with one another.

7.3 Stable Exploration in Multi Agent Reinforcement Learning

This builds on the ideas of [**?**, **?**] and an important point mentioned in [**?**] that 'the dynamics implied by multi-agent systems lead to stochastic behaviour resulting sometimes in undesired effects'. In particular, most RL methods require some trade-off between exploration and exploitation. It is, therefore, required that we determine a safe manner in which MARL agents may explore without leading to unstable behaviour. This would require extending the work of [**?**] and/or [**?**] to the multi-agent case. Note that [**?**] does actually consider a multi-agent case, but with independent learners who do not consider the effect that they have on the other agent(s). The particular extension that I propose is to consider the coupled effect between agents as part of the determination of a safe set for exploration.