Reinforcement Learning for Cybersecurity Risk Assessment of Advanced Air Mobility Systems

by

Brenton A. Pieper

B.S. Cyber Operations, United States Naval Academy, 2022

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2024

Authored by: Brenton A. Pieper
Operations Research Center
May 10, 2024

Certified by: Saurabh Amin
Professor of Civil and Environmental Engineering, Thesis Supervisor

Accepted by: Georgia Perakis
John C Head III Dean (Interim), MIT Sloan School of Management
Professor, Operations Management, Operations Research & Statistics
Codirector, Operations Research Center

Reinforcement Learning for Cybersecurity Risk Assessment of Advanced Air Mobility Systems

by

Brenton A. Pieper

Submitted to the Sloan School of Management
on May 10, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

ABSTRACT

Modern AI/ML tools have significant potential to accelerate the development of Advanced Air Mobility (AAM) systems that use unmanned aerial systems for providing mobility services. The efficacy of these systems relies on highly granular, reliable, and trustworthy sensor data. This thesis is motivated by the need to assess safety risks due to cyber vulnerabilities in the surveillance components of AAM systems such as Automatic Dependent Surveillance-Broadcast (ADS-B) and the Airborne Collision Avoidance System (ACAS).

We focus on spoofing attacks targeted at specific AAM agents and develop a computational approach to evaluate the impact of such attacks on the performance of cooperative agents modeled in a Multi-Agent Reinforcement Learning (MARL) framework. Our threat model is particularly suited for quantifying the safety risks of nominally trained MARL algorithms under attacks by an adversary capable of compromising observational data of a single target agent. In contrast to prior work in Adversarial RL, our approach to creating adversarial perturbations does not require access to learning and control mechanisms internal to the compromised agent. We show how realistic spoofing attacks can be successfully constructed using a simulated MARL-based AAM system, called AAM-Gym. We then conduct a safety risk analysis of such attacks using commonly accepted aviation safety metrics. Specifically, we find that safety compliance decreases across multiple aircraft densities under a spoofing attack to a single agent, owing to higher risk of Near Mid-Air Collision (NMAC).

Finally, to understand possible algorithmic defenses, we take inspiration from Safe RL and show how AAM agents can be made more robust, and hence more safety compliant, to observational spoofing by using a minimax training criterion. Our work highlights the need to rigorously study the safety risks of AAM systems under realistic cyber threat models. Our findings can benefit efforts to develop practical defense techniques, such as signal validation

and filtering, to detect the presence of adversarial perturbations, and control algorithms to adapt and respond to safety compromises in a timely manner.

Thesis supervisor: Saurabh Amin
Title: Professor of Civil and Environmental Engineering

Acknowledgments

I would like to thank my advisor, Professor Saurabh Amin for all the time and attention that he provided to this work as well as to my development as an academic researcher. Not only did he provide insightful feedback throughout the course of this Thesis, but he also encouraged me to pursue anything that interests me and explore my passion in cybersecurity. I am deeply grateful for all the guidance, patience, and care that he has shown over the past two years.

I would also like to thank Andrew Weinert at MIT Lincoln Laboratory. His expertise in all things aviation was pivotal to the development of this Thesis. I am grateful for all of his support and encouragement to pursue challenging topics and push the edge of research. I would also to like to thank Luis Alvarez and Kara Breeden at MIT Lincoln Laboratory, whose assistance in simulations for this Thesis proved invaluable. Special thanks to MIT Lincoln Laboratory as well for the financial support that has allowed me to pursue this opportunity.

Lastly, I would like to thank everyone who has encouraged me in my academic journey to this point. I would like to thank my parents for demonstrating curiosity, compassion, and resiliency and allowing me to follow in their example. I am forever grateful to my wife, Erica, for her constant love, support, and always encouraging me to pursue knowledge to help other people. Finally, I would like to thank all of my friends and family who have supported me over the years and inspired me in this Thesis and beyond.

Contents

Chapter 1

Introduction

With the proliferation of AI, society is witnessing a rise in autonomous agents that interact with an environment with little-to-no human input or assistance. From self-regulating industrial control systems [1] to self-driving cars [2] to AI-enabled personal assistants [3], systems that learn through interaction with an environment and take autonomous actions are widely implemented or in development across the world today. Additionally, with the capability to share training data and observations between autonomous agents and centralized servers, we see the connection of autonomous agents to achieve common goals via federated machine learning methods [4]. One such example is Advanced Air Mobility (AAM) systems which rely on interconnected small unmanned aerial systems to provide mobility services. Due to their autonomous nature, these systems rely on intelligent algorithmic design that allows for safe and reliable automation.

Reinforcement Learning (RL) methods have been proposed and developed for the control and operation of these systems. Modern Reinforcement Learning methods, however, have been shown to be brittle and vulnerable to manipulation which can lead to decreases in system performance [5], [6]. In safety-critical environments, such as Advanced Air Mobility, this decrease can have serious safety risk implications. This necessitates a safety risk assessment of systems that utilize these methods. Moreover, this safety risk assessment should study the specific impact of targeted manipulations using known cyber attack vectors in these systems. This Thesis provides a template for Advanced Air Mobility systems and other similar RL-based systems through which system designers can simulate cyber attacks and study their impact on system safety.

In Chapter 2, we provide a safety risk assessment that considers the cyber threat to Advanced Air Mobility systems. We characterize this threat by providing an overview of system concepts in AAM designed to enable safe operations and introducing a cyber threat model that targets these system concepts and their vulnerabilities. Our approach develops

this threat model for an adversary whose goal is to decrease system safety using a cyber attack, specifically a spoofing attack.

In Chapter 3, we map this threat model to a Multi-Agent Reinforcement Learning framework which allows us to simulate this threat model and measure specific system safety metrics. Our approach is to develop an algorithm for simulating this spoofing attack that considers a realistic adversary in a game theoretic setting using observational perturbations developed by Adversarial Reinforcement Learning. Our approach of implementing an RL-based adversary differs from current methods for creating observational perturbations in which the adversary has full access to the cost function that the victim agent uses for training. Our algorithm provides a novel method for developing optimal observational perturbations in Adversarial Reinforcement Learning that simulates a realistic spoofing attack on a system.

In Chapter 4 we apply Robust Reinforcement Learning methods in AAM against our threat model in order to study possible algorithmic defenses to spoofing attacks on MARL-based systems. Additionally, we propose practical cyber defense techniques that can be applied to AAM and other systems to defend against these attacks.

This case study in Advanced Air Mobility allows us to provide a generalized cyber threat model and safety risk assessment for Autonomous Networked Systems. Autonomous networked systems are a unique form of autonomous systems in that they are multi-agent systems who cooperate to achieve a common goal. Unlike more traditional AI agents, these systems are comprised of multiple AI agents operating within the same environment.

In order to define autonomous networked systems, we identify systems with the following properties:

1. **Distributed Sensing Capabilities**: Autonomous networked system agents are equipped with the ability to observe data from their environment and this ability is distributed across all agents in the system.

2. **Existence of Data-Sharing Channels**: Observations, actions, and messages can be shared either directly from agent to agent or via a centralized server.

3. **Decentralized Autonomous Execution**: Each agent interacts with the environment independently without input from other agents or a centralized server.

With these properties we can assess how networks use these components to achieve autonomy and to analyze the safety implications of attacks to system features.

We define safety as the minimization of unintended or adverse effects to the primary objectives and safety-critical constraints of the environment that the autonomous networked system operates in. In other words, safety of these systems should ensure low risk of violating

safety-critical constraints as imposed by the human designer. In this work we view safety through the lens of minimizing risk metrics in environments of interest.

1.1 Unmanned and Autonomous Systems

A common example of an autonomous networked system is unmanned vehicles. Unmanned Aerial Systems (UAS), Unmanned Underwater Vehicles (UUV), and self-driving cars are technologies already being utilized and developed by industry and the military. UAS networks for intelligence, surveillance, and reconnaissance have been utilized widely in national security missions where UAS have engaged in autonomous path planning to maximize coverage [7] while corporations such as Tesla have developed self-driving cars capable of communicating and providing autonomous transportation for human passengers.

These systems fit the framework for autonomous networked systems, as they have *distributed sensing capabilities* in the form of GPS location sensing through WGS 84 satellite navigation as well as LiDAR and computer vision techniques for self-driving cars. Unmanned vehicle systems also utilize *data-sharing channels* for navigation and communication. UAS have also been widely discussed as avenues for 5G wireless communication to link UAS as well as ground-based users [8], [9] while self-driving cars utilize deep federated learning to share collected training data with other vehicles [10]. Lastly, these systems rely on *decentralized autonomous execution*. While both systems recommend human operators to monitor and take control of the agent in any situation deemed unsafe, the agents are designed to take actions solely based on their own sensing of the environment and data received from the network.

1.2 Advanced Air Mobility

Advanced Air Mobility (AAM) is the use of autonomous aviation to aid in human transportation. In order to aid in the adoption and implementation of these systems in the next decade, The Federal Aviation Administration (FAA) has published guidance and implementation plans as systems are being designed and implemented in the next decade. In 2020, the FAA published Unmanned Aircraft Systems Traffic Management (UTM) Concept of Operations for UAS that operate in low altitude airspace, a major safety concern for AAM. In 2023, the FAA published the Urban Air Mobility (UAM) Concept of Operations to provide guidance for AAM systems in or near urban environments. Through these documents we can identify critical components in AAM and their safety considerations.

UTM is a framework designed to manage the safe and efficient integration of drones into

the airspace, particularly in low-altitude environments where traditional air traffic control is less present. It coordinates flights, resolves conflicts, shares real-time airspace information, and helps enforce regulations and standards. AAM envisions a future with new types of air vehicles, such as electric Vertical Takeoff and Landing (VTOL) aircraft, also known as air taxis. These vehicles will transport passengers and cargo within and between cities. UTM and AAM work in tandem, with UTM providing the necessary infrastructure to enable the complex operations envisioned by AAM.

AAM requires robust data-sharing channels. UTM is dependent on layers of information sharing and data exchange to coordinate and safely separate trajectories [11]. Additionally, the FAA requires a Provider of Services for UAM that provides communication between federated UAM actors and data and advisory centers [12]. Federation is a key component of the cybersecurity and interoperability of UAM, as it allows individual agents to perform their own computation capabilities while allowing for cooperative operation between all UAM agents.

Additionally, AAM relies on independent autonomous execution. Boeing's UAM Concept of Operations describes autonomous UAM aircraft as being able to make decisions to achieve a given objective without operation intervention, [13] while the FAA envisions mature operations for UAM allowing for Human-Over-the-Loop (HOVTL), that is the human's main role is passive monitoring of the system.

Lastly, AAM systems will rely on independent sensing capabilities for their navigation and operation. The FAA stipulates that UAM operators are responsible for maintaining separation. This requirement coupled with the autonomous nature of AAM necessitates Detect-and-Avoid (DAA) equipment to achieve separation autonomously. This may be done through the Airborne Collision Avoidance System sXu (ACAS sXu) for autonomous aircraft or Autonomous Flight Rules (AFR) that require autonomous sensing abilities to determine separation distance.

Since AAM demonstrates the key properties of autonomous networked systems, AAM represents a relevant and timely case study in autonomous networked system safety. In order to fully assess system safety, we must understand the cyber threat to AAM systems, and assess its potential impacts to safe operations.

Chapter 2

Cyber Threats to Advanced Air Mobility Safety

In this chapter we discuss the cyber threat to AAM systems. Firstly, we provide an overview of key concepts for AAM safety for developing a cyber threat model. To adapt AAM to the existing airspace environment, some concepts to enable safe operations and organization have been developed and proposed. We first introduce some common metrics used to design and evaluate airborne systems and then introduce two systems concepts envisioned to enable safe AAM operations and simulation methods for AAM safety. Next we introduce a cyber threat model that targets these concepts using known attack vectors in their current implementations.

2.1 System Concepts for AAM Safety

2.1.1 Aviation Safety Metrics

In order to analyze safe operations in aviation, we introduce key concepts in aviation safety and relevant metrics with which we can provide a robust assessment of safety. The International Civil Aviation Organization (ICAO) defines two processes to satisfy these metrics: separation provision and collision avoidance. Separation provision requires aircraft to maintain an appropriate minimum distance from any potential hazards, while collision avoidance requires avoiding immediate hazards in close proximity to the aircraft. In order to satisfy these requirements, aircraft may use DAA systems to adhere to separation provision and collision avoidance. Any DAA system must be capable of performing these capabilities to conduct safe operations, regardless of environmental conditions.

One metric that can be used to assess the capabilities of these DAA systems is the rate

of Near Mid-Air Collision (NMAC). NMACs, instances where aircraft separation is less than 100 ft vertically and 500 ft horizontally, can be used to assess the safety requirements in both separation provision and collision avoidance by describing the efficacy of DAA systems in preventing a Mid-Air Collision (MAC) [14]. This allows us to utilize NMAC as a surrogate metric for MACs. In AAM, however, we must a utilize a tailored surrogate metric for sUAS. Therefore, we can calculate Small Near Mid-Air Collision (sNMAC) for UAS which describes instances of less than 15 ft vertically and 50 feet horizontally of separation. We can utilize NMAC rates to assess the threat of collisions in encounters between UAS and larger manned aircraft and similarly utilize sNMAC for encounters between UASs [15].

In order to evaluate NMAC and sNMAC rates with DAA systems, we can utilize risk ratios. Risk ratios allow us to compare the probability of NMACs and sNMACs in systems equipped with DAA by comparing them to unequipped systems. We define the risk ratio for NMACs as:

$$RR_{NMAC} = \frac{\mathbb{P}(\text{NMAC})_{equipped}}{\mathbb{P}(\text{NMAC})_{unequipped}} \tag{2.1}$$

and similarly for sNMACs as:

$$RR_{sNMAC} = \frac{\mathbb{P}(\text{sNMAC})_{equipped}}{\mathbb{P}(\text{sNMAC})_{unequipped}} \tag{2.2}$$

In addition to safety metrics, operational suitability and acceptability metrics can be utilized to assess performance. Operational suitability metrics can include the rates of non safety-critical alerts that disrupt normal operation. Acceptability metrics include the reversal rates, the rates at which DAA systems provide alert reversals to change the direction of initial alerts, as well as altitude crossing rates in which DAA systems predict aircraft should cross altitudes with an intruder [14]. These metrics are useful in assessing overall DAA system performance.

2.1.2 Urban Air Mobility

According to the FAA's UAM Concept of Operations, corridors provide an environment for AAM operations in and around urban areas to utilize their capabilities without the need for ground-based air traffic control guidance. Corridors allow for structured traffic management in areas with dense AAM operations and other airspace classifications. Since sufficient separation will be required within these dense areas, conflict management will be largely dependent on vehicle-to-vehicle data exchanges and AFR mechanisms. An example concept for UAM structured airspace shows the necessity for an autonomous networked systems architecture to operate under the operational guidelines published by the FAA. Figure 2.1 shows

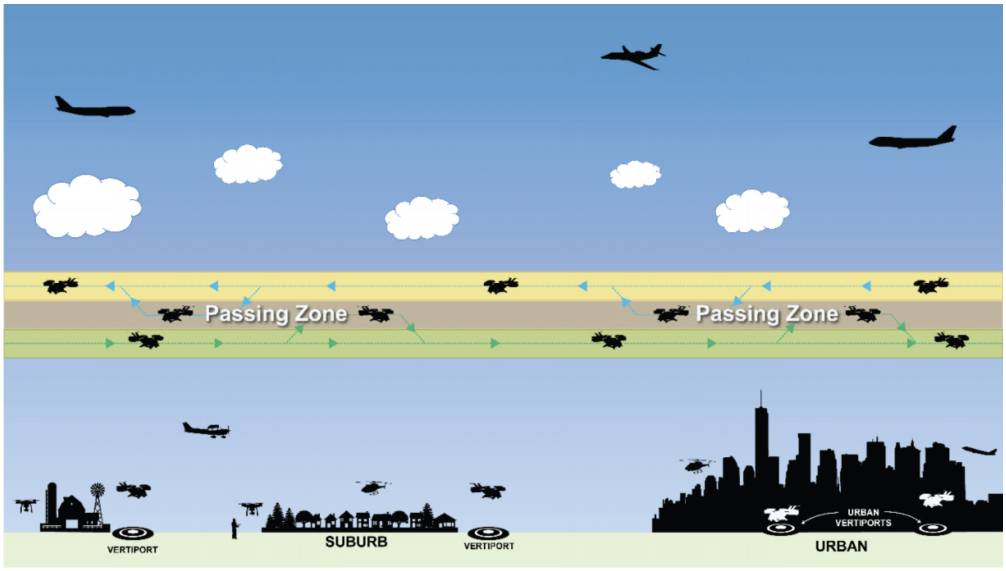


Figure 2.1: UAM Concept with UTM Corridors (FAA, UAM Concept of Operations 2.0, 2023)

a concept for UAM where the properties of autonomous networked systems are necessary for safe operation in a UAM environment, such as taking off from vertiports at low altitudes, or operating in passing zones near dense UTM corridors. This airspace structure allows for reducing the burden on AAM agents to adhere to safety metrics such as avoiding NMACs by providing provisioned airspace traffic management in dense environments.

2.1.3 ACAS sXu

UAM services are primarily designed to mitigate the likelihood of an NMAC through strategic mitigation and separation provisions. If these mitigations fail and two aircraft are within a couple minutes of potentially colliding, DAA systems are designed as an electronic means of compliance to the primarily visual-based separation responsibilities of a pilot and to comply with applicable operating rules of Title 14 of the Code of Federal Regulations (14 CFR). Specifically, ACAS sXu is a standardized DAA systems developed by RTCA for small UAS [16]. It is an evolution of the Traffic Alert and Collision Avoidance System (TCAS) that has been used to minimize the risk of an NMAC for decades. The ACAS X variants use dynamic programming [17] to develop an offline lookup table that an equipped agent can use to quickly and autonomously respond to a detected intruding aircraft to prevent an NMAC. ACAS sXu relies on a broad range of surveillance equipage to detect intruding aircraft which is why distributed sensing capability is crucial in operating in safety critical encounters.

2.1.4 Evaluating in Simulation and Encounter Models

In order to assess systems such as ACAS, we must use realistic simulations in various operational settings so that we can robustly examine the aviation safety metrics previously introduced. We can do this using encounter sets that assume existing safety layers such as air traffic control and UTM corridors have failed. Specifically, previous work has developed encounter sets to test DAA system behavior in a simulation that approximates actual operations and measure the safety metrics in this simulation [18]. Utilizing these encounter sets allows us to provide an assessment of aviation safety metric adherence of relevant technologies in AAM such as ACAS sXu that will aid us in providing a safety risk assessment of AAM.

2.2 Cyber Safety Risk Assessment

While some work has been done in characterizing specific threats to UAS or other systems such as self-driving cars, there has not been a large-scale safety risk assessment of autonomous networked systems under a cyber threat model. This work bridges the gap between the work that has been done to describe potential threat vectors in systems such as UAS and domain-specific safety metric evaluations in systems such as AAM. We do this by introducing a general cyber threat model for autonomous networked systems that can be used to conduct a cyber safety risk assessment.

Since autonomous networked systems rely on accurate sensing, data-sharing, and execution, they also depend on trustworthy environmental sensing, low network latency, and high component reliability. Without resiliency to failures in these areas, autonomous networked systems may fail. Recent work has analyzed autonomous networked systems to improve this resiliency. For example, UAS networks under a Byzantine Generals problem can adapt using attritable learning to maintain a high mission probability success [19]. To provide a cyber safety risk assessment of autonomous networked systems, we introduce a cyber threat model for AAM.

2.2.1 Cyber Threat Model

In order to assess the cyber safety risk of AAM, we must characterize the potential threats that may target known vulnerabilities in these systems. As with all computer systems, autonomous networked systems have features that are vulnerable to cyber attacks, however, some of these features are uniquely vulnerable to autonomous networked systems [20].

For cyber-physical systems, to include autonomous networked systems, we are interested in protecting the integrity, authenticity, and availability of systems. One way that we can analyze the cyber threat to these attributes of cybersecurity is by applying the STRIDE model developed by Microsoft for computer security threat modeling. The STRIDE model categorizes six threats to security: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. We are specifically interested in the spoofing and denial of service threats to autonomous networked systems as these violate the *integrity* and *authenticity* of shared data used by the system and the *availability* of these systems respectively.

While other cyber attacks to components used for control such as propulsion or Vertical Takeoff and Landing (VTOL) equipment would violate the *integrity* of these systems, this would likely lead to total system failure, so this problem is trivial for a cyber attacker if they can gain access to the system. Therefore, spoofing and denial of service attacks are more relevant to study as these allow for an attacker to impact system safety while also maintaining stealth. In this work, we will model the effect of spoofing attacks on the safety metrics of autonomous networked systems, however, the potential impact of denial of service attacks should also be noted.

2.2.2 Spoofing

Spoofing is an attack on a systems in which an attacker provides information to a target system that did not come from the source that the system believes it came from [21]. Spoofing violates the *authenticity* of systems by removing the confidence and validity in data received by the system. This is relevant to autonomous networked systems that depend on both data channels and environment sensing. Additionally, in the context of autonomous networked systems, spoofing represents a threat to the *integrity* of the system as well, as the system's interaction with the environment depends on the validity of the input signals coming from data-sharing channels and environment sensing.

In the context of AAM, spoofing attacks on autonomous aviation have had a history in both research and in real life operations. GPS navigation signals have long been known to be able to be spoofed as seen in 2011 when Iran successfully used GPS spoofing to capture a US military UAS [22]. Furthermore, the Automatic Dependent Surveillance-Broadcast (ADS-B) system that is widely implemented in NextGen aviation systems to include AAM, has also been shown to be vulnerable to spoofing attacks [23]. Since GPS and ADS-B, which is built off GPS, are two main input sources for sensing capabilities for AAM, spoofing attacks are a relevant attack vector that violate the *authenticity* and *integrity* of AAM systems.

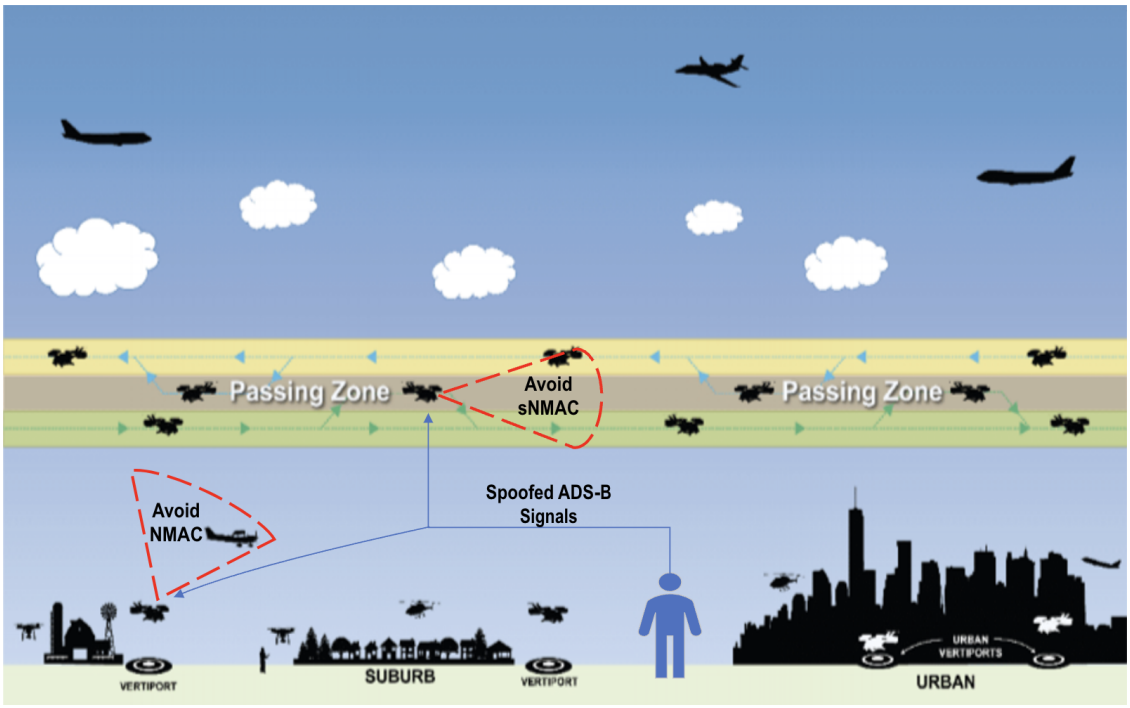


Figure 2.2: Threat Model for AAM Operations. Adapted from (FAA, UAM Concept of Operations 2.0, 2023)

Figure 2.2 shows scenarios in the UAM concept where this threat model can have safety-critical impacts. For example, agents rely on accurate sensing to avoid sNMACs when operating in corridors or avoiding NMACs with other aircraft in the airspace. Additionally, operational constraints such as safe separation distance relies on this architecture as well.

2.2.3 Denial of Service

Denial of service describes any attack vector in which the attacker prevents access of a system to users, operators, or other systems. In the context of autonomous networked systems, whose operation depends on availability of data from the environment and other agents, denial of service attacks prevent the safe and reliable operation of systems, thereby violating the *availability* of autonomous networked systems.

GPS and ADS-B input can lead to jamming attacks in which attackers flood the target with signals and prevent the legitimate signal from being received by the target. Jamming attacks have been used to interrupt GPS communication for decades and is still a problem today for aircraft that fly in geopolitical hotzones [24], [25]. This represents a relevant attack vector for AAM as users who will use these systems for transportation will expect the availability of safe and reliable operations. Under this threat model, we will now outline some of the work that is being done to mitigate these attacks and work to study their effects

on autonomous networked systems.

2.2.4 Related Work and Simulation Environments

Here we describe how recent work has tested this security threat in similar settings through simulation using the attacks previously described. Analysis of self-driving car fleets using open-source simulation environments measured the likelihood and impact of such attacks [26], [27]. Similarly, tests on open-source UAS and aviation simulations such as AirSim and FlightGear have measured the likelihood and impact of these attacks on autonomous aerial vehicles [28]–[30]. Additionally work in both academia and by the FAA has developed custom simulation environments for testing ADS-B and GPS spoofing attacks on UAS with the goals of improving safety and resiliency through collision avoidance and improved incident response [31]–[33].

We build on successful proof of concept attacks in these simulation environments by demonstrating a proof of concept attack in AAM. While some of the previous work has shown the impact of spoofing attacks in manned aircraft, this attack has not been shown in an autonomous system such as AAM. Furthermore, simulation of this attack will allow us to provide a safety risk assessment using aviation safety metrics to quantify this attack's impact.

Chapter 3

Safety Risk Analysis Through Multi-Agent Reinforcement Learning

With independent execution, autonomous networked systems will have to rely on machine learning and autonomous decision making to operate safely during normal operation without a human in the control loop. For example, machine learning will be used for sensing capabilities such as computer vision for object recognition in self-driving cars. The networked nature of these systems will also necessitate shared learning of algorithms to improve system performance and expedite training time. One such widely proposed machine learning method for training and operating these systems is Reinforcement Learning (RL).

This chapter provides a safety risk analysis of autonomous networked systems under the lens of RL. We provide an overview of modern RL and Multi-Agent Reinforcement Learning (MARL) and map our cyber threat model for autonomous networked systems to MARL-based systems. Next we develop an algorithm for simulating realistic spoofing attacks in MARL systems in which the attacker can only influence the sensing capabilities of the system and does not have access to the internal training mechanisms of the system. Finally, we test this threat model on an autonomous networked system simulation using state-of-the-art MARL algorithms and for an RL-based AAM environment and measure safety metrics.

3.1 Introduction to Multi-Agent Reinforcement Learning

This section provides a background on RL, MARL, and various algorithms and methods being implemented to improve RL-based system's safety and reliability. RL is the branch of Machine Learning that deals with autonomous agents that interact with an environment, receive a reward signal from the environment, and learn from that signal to take optimal actions to achieve some defined goal. RL is particularly adept at operating under uncertainty

where it is able to make sequential decisions and can outperform humans in some tasks as demonstrated in gaming [34]. RL has also been used to design autonomous systems such as in transportation, as well as to teach AI models such as ChatGPT to interact with humans more optimally [35]. Moreover, RL is a natural choice for AAM as it allows for automation of both system control as well as Detect-and-Avoid (DAA) capabilities of agents.

3.1.1 Dynamic Programming and RL

The underlying dynamics to solve RL problems rely on Dynamic Programming. Dynamic Programming can be used to solve a problem by breaking it into sequential subproblems to recursively solve the larger problem. This is useful for RL where we want to solve a sequential decision process in which we can solve smaller subproblems to choose actions optimally at each timestep. Dynamic Programming is applied to RL by utilizing a Markov Decision Process (MDP) to model the RL agent and its environment. An MDP is described by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ where $\mathcal{S}$:= the set of possible states $s \in \mathcal{S}$ of the environment and agent, $\mathcal{A}$:= the set of possible actions $a \in \mathcal{A}$ of the agent, $\mathcal{P}$:= the transition probability of a state and action $\mathbb{P}(s'|s,a)$, $\mathcal{R}$:= the rewards for all state-action pairs (s,a), and γ := the discount factor $\in [0,1)$. Given an MDP, an agent can estimate the value of states $V(s)$ by estimating the expected sum of discounted rewards:

$$V(s_t) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t R(s_t, a_t, s_{t+1}) \middle| s_0 = s, a_t\right] \tag{3.1}$$

An RL agent optimizes over some time horizon T by choosing a policy π that is the solution to the optimal Bellman equation:

$$\pi^*(s) = \arg\max_a \left\{ R(s,a) + \gamma \sum_{s'} \mathbb{P}(s'|s,a) V^{\pi^*}(s') \right\} \tag{3.2}$$

ACAS sXu, uses Dynamic Programming to provide DAA capabilities for sUAS. It builds off of the Dynamic Programming approached developed for TCAS in which the system uses a cost function $C(s,a)$ where the system receives negative reward, and thus has a negative value, $J_k(s)$, that we seek to minimize when (s,a) results in an NMAC and alerts an optimal action to lower this cost, thereby avoiding the risk of NMAC.

$$J_k(s) = \min_a \left[C(s,a) + \sum_{s'} \mathbb{P}(s'|s,a,) J_{k-1}(s') \right] \text{[17]} \tag{3.3}$$

sXu tailors this approach to sUAS by optimizing over the action space for sUAS and by avoiding sNMACs. These actions are computed in an offline table and each sXu equipped aircraft can utilize this lookup table to automatically respond given any aviation state space.

In most scenarios, however, the state and action space is too large to enumerate and solve directly with dynamic programming, so RL relies on approximate dynamic programming techniques to estimate an MDP and solve the Bellman equation.

3.1.2 RL Algorithms

With the recent improvements of deep learning [36], several methods have been developed that have leveraged deep learning's speed and universality to improve the performance of these techniques. One such popular method is Deep Q Networks (DQN), in which the value V of a state is given by a state-action pair Q and is estimated by a deep neural network. Other techniques include policy optimization methods that learn parameterized policies π_θ, and actor-critic methods that utilize both value estimation and policy optimization for RL learning.

Soft Actor-Critic

Soft Actor-Critic (SAC) is a high performing actor-critic algorithm for RL. SAC addresses the main challenges of policy optimization problems, high sample complexity and brittle convergence guarantees, by utilizing an entropy measure when estimating the value of a policy.

$$V(\pi) = \mathbb{E}_{(s_t,a_t)\sim p_\pi}\left[\sum_{t=0}^{T} R(s_t,a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t))\right] \text{ [37]} \tag{3.4}$$

By maximizing this dual objective, SAC is able to ensure that it is maximizing its reward while also acting as randomly as possible. This aids in smoother and stable convergence and reduces the sample complexity needed, allowing for very high performance across many domains.

3.1.3 POMDPs and MARL

Multi-Agent Reinforcement Learning (MARL) is an extension of RL to systems in which multiple agents interact with a single environment cooperatively, competitively, or independently [38]. Similarly to single-agent RL, We can extend MDPs to a multi-agent setting

using a Markov Game in which the value can be estimated as:

$$V^i_{\pi^i,\pi^{-i}}(s) = \mathbb{E}\left[\sum_{t\geq 0}\gamma_t R^i(s_t, a_t, s_{t+1})\middle| a^i_t, s_0 = s\right] \tag{3.5}$$

Here, $-i$ is the index of all agents except i and the optimal solution is a function of all players' simultaneous actions. This solution is known as a Nash Equilibrium [39] of a Markov Game where it has the property

$$V^i_{\pi^{i*},\pi^{-i*}}(s) \geq V^i_{\pi^i,\pi^{-i*}}(s), \forall \pi^i \tag{3.6}$$

Due to the scalability issues of most real-world environments, however, this solution is infeasible to compute directly, so most MARL algorithms converge to an estimated equilibrium using techniques applied from single-agent RL.

Finally, decision processes in reality are often made under uncertainty where the true state information is not fully known. The Partially Observable Markov Decision Process (POMDP) is an extension of the MDP to account for this uncertainty which is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$ where $\Omega :=$ the set of possible observations and $\mathcal{O} :=$ the set of observational probabilities. When an agent completes a transition $(s, a) \to s'$, it receives an observation $o \in \Omega$ with probability $\mathcal{O}(o|s', a)$ [40]. The agent can then compute a Bayesian belief $b(s)$ update about the state:

$$b'(s') = \frac{\mathcal{O}(o|s', a)\sum_{s\in\mathcal{S}}\mathbb{P}(s'|s, a)b(s)}{\sum_{s'\in\mathcal{S}}\mathcal{O}(o|s', a)\sum_{s\in\mathcal{S}}\mathbb{P}(s'|s, a)b(s)} \tag{3.7}$$

The agent can then use this belief state to solve the Bellman optimality equation

$$V^*(b) = \max_{a\in\mathcal{A}}\left\{\sum_{s\in\mathcal{S}} b(s)R(s, a) + \gamma\sum_{o\in\Omega}\mathbb{P}(o|b, a)V^*(\tau(b, a, o))\right\} \tag{3.8}$$

where $\tau(b, a, o) = \sum_{o\in\Omega}\mathbb{P}(b'|b, a, o)\mathbb{P}(o|a, b)$. We can apply this to the multi-agent setting similarly to MDPs to get a Bayesian Game and estimate a Bayesian Nash Equilibrium. Estimating this Bayesian Nash Equilibrium is helpful in analyzing multi-agent settings in cooperative or competitive environments. We now introduce a selection of state-of-the-art MARL algorithms

3.1.4 MARL Algorithms

Recent work on MARL algorithms has developed high-performing algorithms to provide optimal control for MARL settings. Here we describe three of these algorithms that cover the various techniques for deep learning.

Multi-Agent Deep Deterministic Policy Gradients

Multi-Agent Deep Deterministic Policy Gradients (MADDPG) is an actor-critic method that utilizes both policy optimization, the actor, and a Deep Q Network to estimate state-action value pairs, the critic [41]. MADDPG leverages the actor-critic architecture for the multi-agent setting by implementing an actor network for each agent and a centralized critic network for all agents. MADDPG combines the state information of each agent $i \in N$ into a vector $\mathbf{x}$ and computes the gradient of expected reward for each agent, $J(\theta_i)$ as:

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{s\sim p^{\mu}, a_i \sim \pi_i} \left[\nabla_{\theta_i} \log \pi_i(a_i|o_i) Q_i^{\pi}(\mathbf{x}, a_1, ..., a_N)\right] \tag{3.9}$$

MADDPG choose the policy π_i for each agent that is going to maximize this reward based on each agent's observations. MADDPG depends on the *integrity* of observations to maintain high performance. Additionally, the decentralized policy network is trained using a centralized critic network, so even though execution is fully decentralized, agents' actions implicitly influence other agents which could cause global divergence if even one agent does not act deterministically.

Multi-Agent Proximal Policy Optimization

Multi-Agent Proximal Policy Optimization (MAPPO) is another actor-critic method built on PPO which is a high-performing policy optimization algorithm that uses a surrogate objective and a clipping function to achieve high-performing and stable convergence of the algorithm [42], [43]. MAPPO computes the gradient for an objective function for each agent as:

$$L^{CLIP}(\theta_i) = \mathbb{E}_t \left[\min(r_t(\theta_i)\hat{A}_t, \text{clip}(r_t(\theta_i), 1-\epsilon, 1+\epsilon)\hat{A}_t\right] \tag{3.10}$$

Here, $r_t(\theta_i)$ is the "trust region" for each agent where $r_t(\theta_i) = \frac{\pi_{\theta_i}(a_{i,t}|s_{i,t})}{\pi_{\theta_i,\text{old}}(a_{i,t}|s_{i,t})}$ and $\hat{A}_t$ is a centralized advantage function for the agents. MAPPO has shown high-performance in various MARL testbeds, however, it suffers from the same need for correct observations to compute $r_t(\theta_i)$. MAPPO also implicitly considers all agent's actions through the centralized advantage function similarly to MADDPG.

QMix

QMix is a value-based MARL method that uses value decomposition networks to perform centralized training and decentralized execution.

$$Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \sum_{i=1}^{n} Q_i(\tau^i, u^i; \theta^i) \text{ [44]} \tag{3.11}$$

Here, $\boldsymbol{\tau}$ is the joint observation history of all agents and $\mathbf{u}$ is the joint actions of all agents. QMix then chooses the joint action that maximizes Q_{tot}. Since this is the same as selecting the individual actions that maximize each Q_i, this allows for scalable training of the agents and maintaining high performance during decentralized execution.

3.2 Cyber Threats to MARL Systems

The observations that the POMDP receives and uses to guide the system comes from agent sensing of the environment, either by direct observation or via a centralized information server. Additionally, during execution the actions of a MARL agent are autonomous and decentralized; each agent takes its own action based solely on its own learned control mechanisms. Therefore we can see that the properties of an autonomous networked systems can be modeled using MARL, and furthermore the cyber threat model for autonomous networked systems can be mapped to the MARL framework. For example, in AAM, navigational data received via ADS-B is used as the observation to guide the AAM agent. Since we have shown that these signals can be spoofed, we can also see that observations in a MARL-based system can be spoofed. Therefore, we can use this mapping to develop an RL algorithm for observational spoofing in a cooperative MARL environment and we will use this algorithm to conduct a safety risk analysis through MARL.

3.2.1 Mapping a Threat Model to MARL

The performance of RL algorithms is directly dependent on the reliability and accuracy of environmental observations. Without consistent correct observations, agents are not guaranteed to converge to an optimal solution [5]. This property is further magnified by the non-stationarity of MARL systems. This dependency represents an attack vector on MARL-based autonomous systems through manipulating the observational data that MARL systems rely on to operate optimally and safely.

Since an agent's action selection depends on the *integrity* and *availability* of the observation, adversarial perturbations in the input data or loss of input data can alter a MARL

agent's performance due to the underlying POMDP guiding the agent's decision making. Recent work has utilized *adversarial examples*, perturbations to the observations received by an RL agent at test time that cause the largest possible decrease in the agent's performance from what it would predict using its trained policy [6]. In other words, adversarial examples allow for an adversary to spoof the observations of an MARL agent and alter the agent's behavior. Additionally, in the cooperative MARL setting, empirical studies have shown that flipping the reward signal of a single agent during test time can cause global divergence from the Bayesian Nash Equilibrium of all cooperative agents [45]. Our cyber threat model can capture this behavior by considering adversarial examples, and thereby lower bounding the worst possible system performance under our threat model during test time.

3.2.2 An Algorithm for Observational Spoofing in MARL

Previous work has shown that data poisoning attacks in which the observational perturbations are provided during training time, as well as Adversarial RL in which training an adversarial agent to decrease another agent's reward, can lead to sub-optimal performance. Neither of these methods, however, are practical for our cyber threat model. A realistic adversary would only have access to an already trained agent during test time, and in an environment such as AAM, we are interested in a ground-based adversary that does not have access to the physical action space of the agents.

Additionally, while methods such as the Fast Gradient Sign Method (FGSM) [46] have been used to generate quick and optimal adversarial examples, they require access to the trained neural network and differentiable inputs. While this is useful for designing robust systems, we are interested in an adversarial environment in which the networks are not open source and inputs may include non-differentiable categorical features. Therefore, our method for developing adversarial examples is to train an RL agent to select perturbations to decrease the reward or increase the cost of the victim agent.

We can model this additional adversarial agent in the existing cooperative setting by modeling the environment as a Bayesian Game. The adversary can select perturbations $\delta \in \Delta$ and provide these perturbations to the observations o of the agent. Therefore, if the victim agent's reward signal is based on minimizing the cost of actions in the environment, the adversary's goal is the maximize the costs of the agent. The adversary's value V_{ADV} in this Bayesian Game can be modeled as a max-min problem in which the adversary selects the perturbation δ first, and the agent takes its action a next based on the adversary-provided

state. The best response for the adversary would then be given as:

$$V^*_{ADV}(s) = \max_{\delta \in \Delta} \min_{a \in \mathcal{A}} \mathbb{E}\left[C(s,a) + \gamma \sum_{s'} \mathbb{P}(s'|s,a) J^{\pi^*}(s') \right] \tag{3.12}$$

where $s := (o + \delta)$. This best response can be modeled as a Bayesian Nash Equilibrium for the adversary by fixing the victim agent's actions based on the observation it receives. If the victim agent does no exploration during test time and acts according to its trained weights, then this Bayesian Nash Equilibrium allows us to lower bound system performance during test time given our spoofing adversary.

In the cooperative MARL setting, the adversarial agent's goal is to maximize the cost of all cooperative agents. Similarly to previous work, we do this by targeting a single cooperative agent. The best response for the adversary in the multi-agent setting can be given as:

$$V^*_{MADV}(\mathbf{s}) = \max_{\delta \in \Delta} \sum_{i \in \mathcal{I}} \min_{a_i \in \mathcal{A}} \mathbb{E}\left[C(s_i,a_i) + \gamma \sum_{s'_i} \mathbb{P}^i(s'_i|s_i,a_i) J^{\pi^*}_i(s'_i) \right] \tag{3.13}$$

where $s_i := (o_i + \delta)$ for the the spoofed agent i and $s_i := (o_i)$ for all other agents. Similarly, this can be modeled as a Bayesian Nash Equilibrium by fixing all inner actions a_i and computing this best response.

To solve for this Bayesian Nash Equilibrium we train the cooperative agents using the inner cost-minimization problem until all cooperative agents converge to some global optimum. We then fix the actions of all cooperative agents, $\mathbf{a}_t$, using these trained policies and train the adversary to solve the outer maximization problem by providing perturbations to a single trained victim agent, so that solving the outer problem is a single-agent RL problem. When the outer agent converges to some optimum, we declare that optimum to the be the estimated Bayesian Nash Equilibrium V^*_{MADV}. This algorithm is described in Algorithm 1 in Appendix A.

3.3 Simulating a Cyber Safety Risk Assessment With MARL

We can now conduct safety-critical modeling using our threat model by simulating attacks on AAM. Firstly, to show the impact on a generalized autonomous networked system, we can show the effects of our threat model on the PettingZoo Multi-Particle Environments

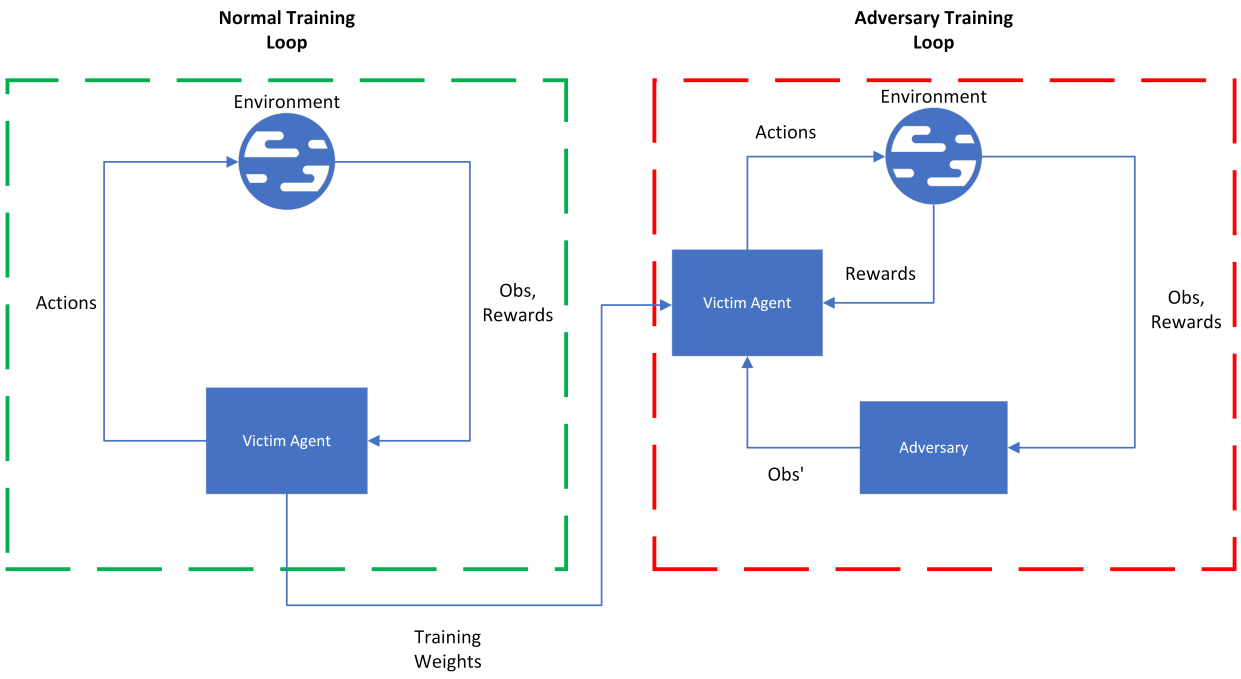


Figure 3.1: Training a Spoofing Adversary

(MPE) developed by OpenAI [47]. Next, we conduct a simulated attack in the AAM-Gym testbed developed by MIT Lincoln Laboratory [48]. This approach allows us to establish the link between the safety risk analysis of autonomous networked systems and the properties and performance of MARL algorithms used to model to such systems.

3.3.1 Multi-Particle Environments

MPE allows us to test our threat model on state-of-the-art MARL algorithms. MPE is a common testbed for MARL algorithm benchmarking and development. Previous work utilizing MPE has analyzed algorithm performance by collecting the reward convergence of agents in different scenarios. We build on this work by simulating our spoofing attack within an MPE environment and assessing system performance using various algorithms.

We test the Simple Spread environment from MPE in which the goal of the cooperative agents is to minimize the distance of the closest agent to a set of landmarks without colliding with each other. This environment is a suitable choice for our safety risk analysis as we can test the performance of the algorithms in an adversarial environment in which there is a specific safety risk associated with the reward signal. The adversary is trained with a single-agent version of the multi-agent algorithm used, so PPO is used for the MAPPO test, DDPG for MADDPG, and DQN for QMix. The adversary is trained to spoof the location

of other cooperative agents and is rewarded for causing collisions. The code used to test the MARL algorithms was adapted from [49] and standard hyperparameters were used. All MPE experiments were computed on the MIT SuperCloud [50].

3.3.2 AAM-Gym

AAM-Gym allows us to test our threat model on an RL-based AAM environment that includes agent equipage with ADS-B, GPS, ACAS sXu, and UTM corridors over the New York City metro area. Since the observation space for AAM-Gym utilizes ADS-B, we can train an adversary to spoof the ADS-B input to a victim agent in order to conduct our safety risk analysis. Specifically, we spoof latitude and longitude measurements in ADS-B. Additionally, we can analyze the dynamic programming approach of sXu as well as Discrete Soft Actor-Critic (SACD) in a realistic UAM scenario. Firstly, we adapt Algorithm 1 for adversarial observational spoofing to AAM by spoofing ADS-B signals for an sXu equipped sUAS. We then test this algorithm to calculate the risk ratio for NMACs using different aircraft densities.

Our algorithm for ADS-B spoofing in AAM-Gym utilizes a layered agent in which the inner agent receives observations from the environment, then passes its actions and observations to an outer agent that then interacts with the environment using both agents' actions. In our adversarial environment, the inner agent is the AAM agent, either sXu or RL equipped, seeking to maintain separation assurance, while the outer agent is the adversarial agent with the ability to add noise to the inner agent's observations. This agent and environment setup can be seen in Figure 3.2. The adapted algorithm to apply our model for adversarial observational spoofing to a layered agent aircraft in an AAM environment is given by Algorithm 2 in Appendix A.

We utilize the encounter sets from [48] to test with aircraft densities of 100, 200, 300, 400, and 500. We train the adversary with SACD to spoof the latitude and longitude observations of intruding aircraft received from ADS-B and reward our adversary for causing NMACs and sNMACs. All of our experiments for AAM-Gym were run using the MIT Lincoln Laboratory Supercomputing Center [50].

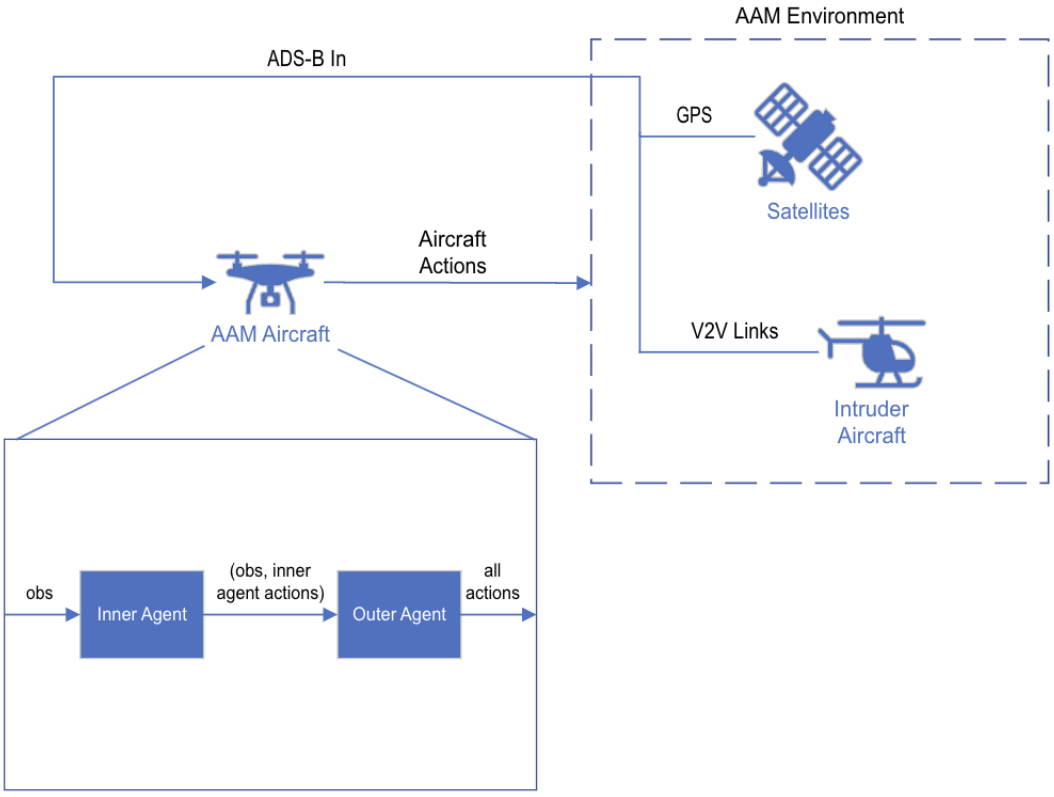
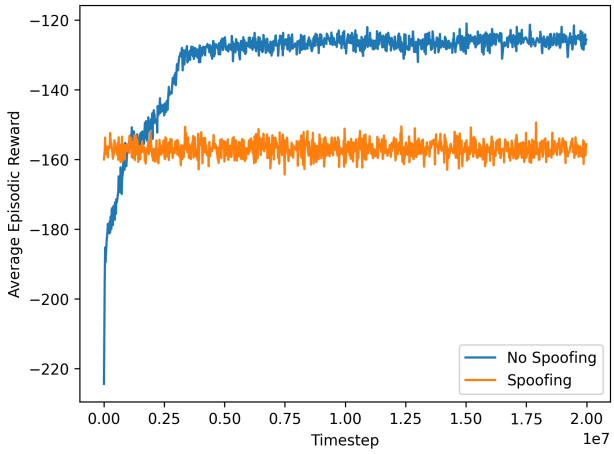


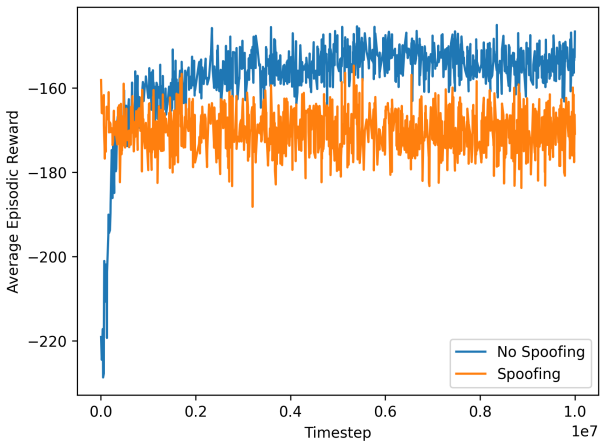
Figure 3.2: Layered AAM Agent

3.4 Simulation Results and Analysis

3.4.1 MPE Results



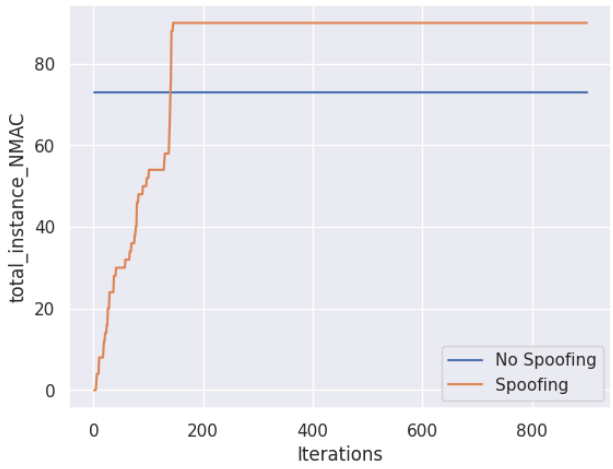
(a) MAPPO Performance



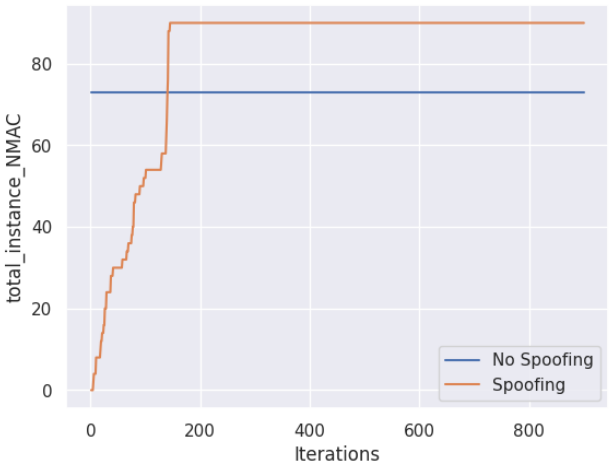
(b) QMix Performance

We see that the spoofing adversary caused a $\sim 25\%$ increase in total costs to the cooperative agents for MAPPO, and a $\sim 10\%$ increase in total costs to the cooperative agents for QMix. As noted in [41], MADDPG does not perform well in this safety-critical environment to begin with, so spoofing has no noticeable impact on system safety as system safety is already poor without spoofing for MADDPG. We also notice that a DQN-trained adversary increases the instability of the QMix agents. Overall, we can see that in both MAPPO and QMix, overall system performance is decreased due to a loss in safe separation and thus higher frequency of collisions by spoofing a single agent's observations. This shows that our proof of concept attack is successful in a generalized autonomous networked system.

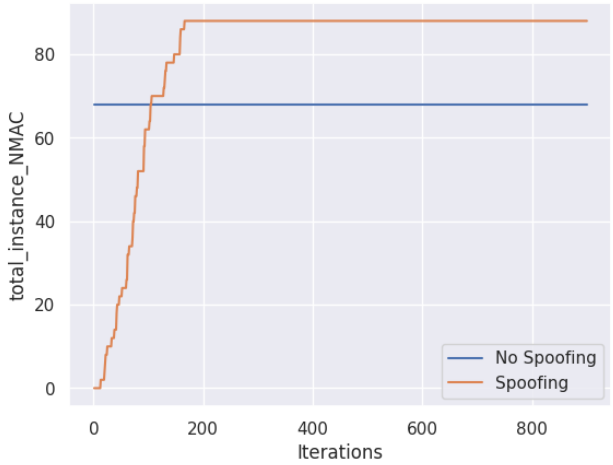
3.4.2 AAM-Gym Results



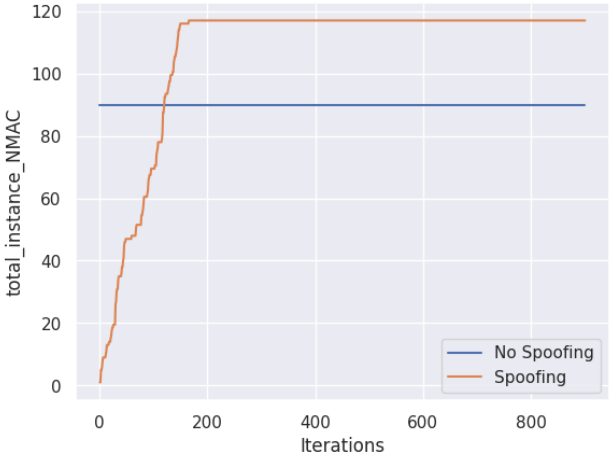
(a) sXu NMACs 100 Density



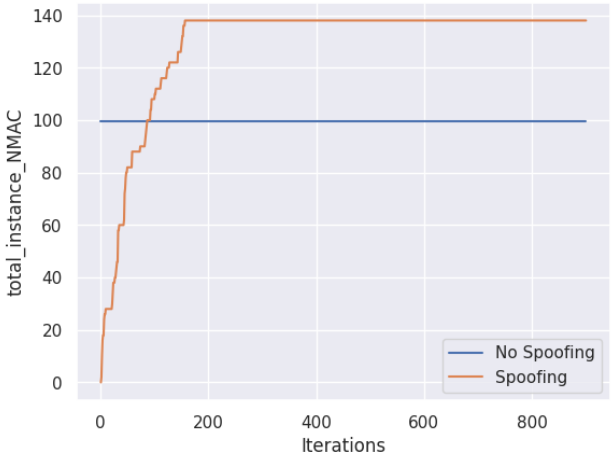
(b) sXu NMACs 200 Density



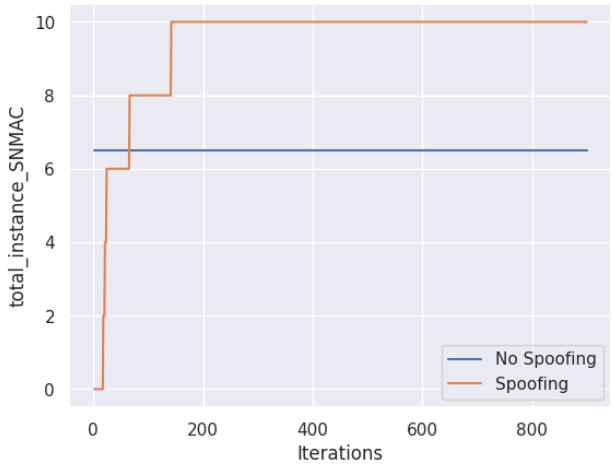
(c) sXu NMACs 300 Density



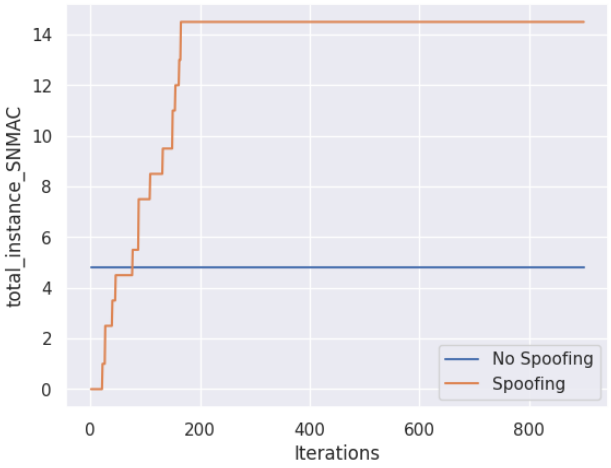
(d) sXu NMACs 400 Density



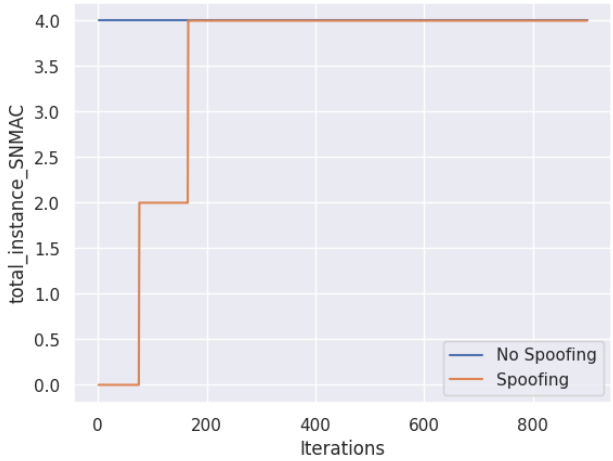
(e) sXu NMACs 500 Density



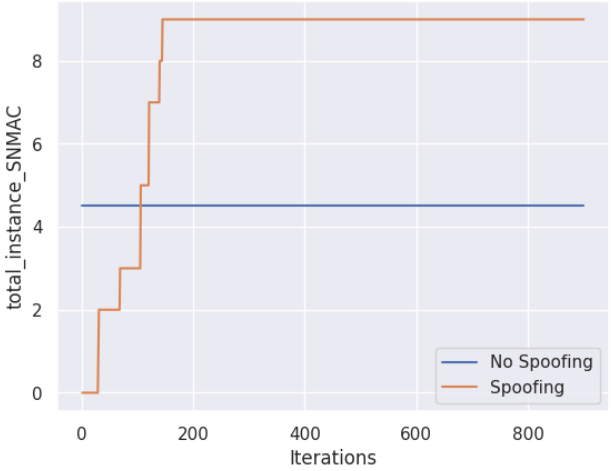
(a) sXu sNMACs 100 Density



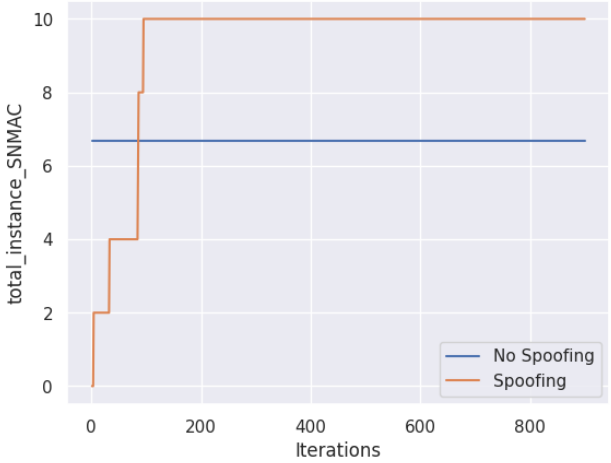
(b) sXu sNMACs 200 Density



(c) sXu sNMACs 300 Density



(d) sXu sNMACs 400 Density



(e) sXu sNMACs 500 Density

We can see that in AAM, our spoofing attack caused an increase in NMACs at all aircraft densities, and an increase in sNMACs at all densities except 300 aircraft where it achieves the same number of sNMACs as the baseline. While we see an increase in sNMACs, the encounter sets used for these experiments were designed to test for NMACs, not sNMACs. Because of this we can not definitively say that our attack increases the sNMAC risk ratio of sXu, however, we can say that our proof of concept attack is successful in increasing the quantity of both NMACS and sNMACs across multiple aircraft densities.

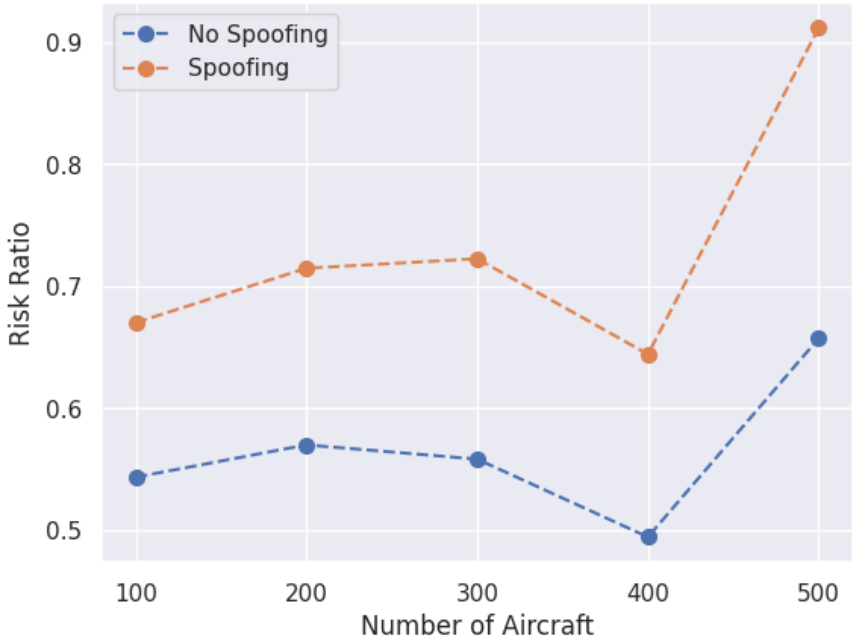
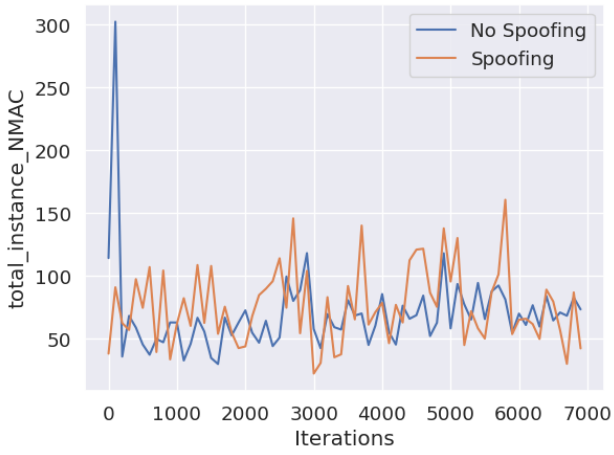
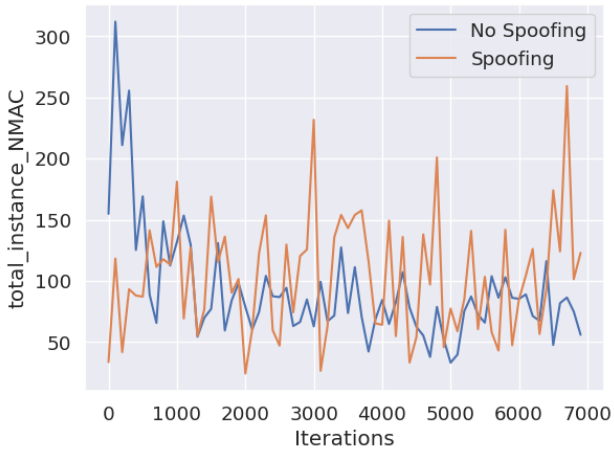


Figure 3.6: ACAS sXu NMAC Risk Ratio

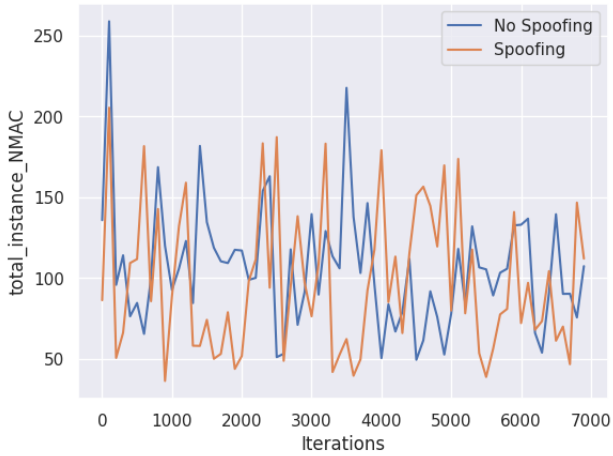
We can see that the NMAC risk ratio increases under the presence of a spoofing attack. At a density of 400 aircraft, we see that both the baseline and spoofing risk ratios decrease, however, due to the symmetric decrease for both the baseline and spoofing risk ratios, it is likely that this decrease is caused by the encounter set, not sXu. Lastly, we see that at a density of 500 aircraft we have very large risk ratios due to both the spoofing attack as well as the dense environment leading to even more NMACs.



(a) SACD NMACs 100 Density



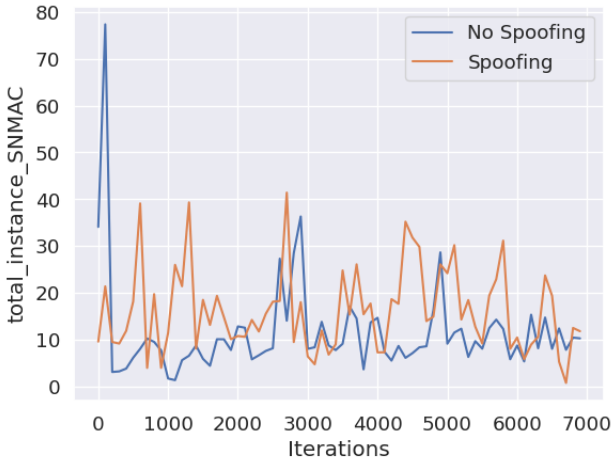
(b) SACD NMACs 200 Density



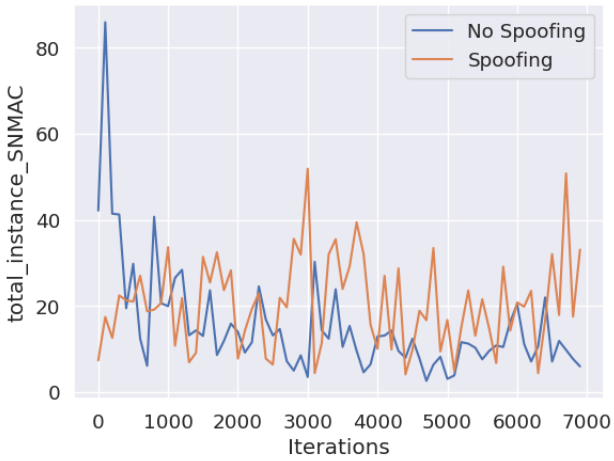
(c) SACD NMACs 300 Density

Figure 3.7: SACD NMAC Spoofing Simulations (Running Average per 100 Iterations)

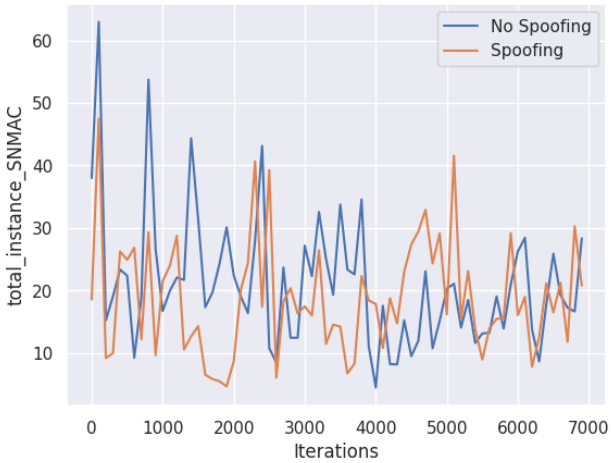
Given the instability caused by very dense environments at aircraft densities of 400 and 500, we test the 100, 200, and 300 aircraft density environments for SACD. The layered approach using SACD for both the victim and the adversary, does not converge as quickly as ACAS sXu, as the environment resets often, likely due to instances of system failure caused by NMACs as can be seen by the large spikes in NMACs before a reset. The increase in NMACs, however, shows that the proof of concept attack is successful against RL-based methods in AAM at increasing NMACs.



(a) SACD sNMACs 100 Density



(b) SACD sNMACs 200 Density



(c) SACD sNMACs 300 Density

Figure 3.8: SACD sNMAC Spoofing Simulations (Running Average per 100 Iterations)

Similarly, we see larger spikes in the number of sNMACs across all 3 aircraft densities with spoofing. Though SACD is clearly not as adept at avoiding sNMACs as ACAS sXu, the increase in sNMACs across all densities shows that the proof of concept attack is successful against RL-based methods in AAM at increasing sNMACs.

3.4.3 Discussion

Overall, we see that our attack is successful at decreasing safety metric compliance and therefore increasing safety risk. We see that in AAM, our attack has a significant impact on increasing the number of overall NMACs by spoofing the ADS-B observations of only a single sUAS. We observe that the adversarial agent notably increases the NMAC risk ratio with sXu which indicates that the agent is reducing relative separation. These results also

directly motivate the need for development of sNMAC encounter sets so that this threat can be further modeled and understood in an AAM environment. The observed reduced separation due to spoofing and increase in NMACs and sNMACs, however, do indicate that the proof of concept attack was successful against both sXu and SACD in decreasing system safety.

Chapter 4

Design of Robust and Secure Systems

In order to improve the safety of Advanced Air Mobility and other autonomous networked systems, design changes must be made to ensure safety even in the presence of cyber attacks. Designers of these systems can do this by implementing both algorithmic and practical improvements to systems and analyze system safety under our safety risk analysis. In this chapter we introduce Safe and Robust RL, apply these concepts to the MPE and AAM-Gym experiments, and provide an analysis of safety metrics with these improvements. Lastly, we suggest a few practical cyber defense techniques to defend against spoofing attacks.

4.1 Safe MARL

Since autonomous networked systems rely on MARL for their control, we can leverage RL techniques to improve safety of these systems. Our threat model introduces uncertainty into the environment by making the agent's observations uncertain via spoofing attacks. Much work has been done in decision making under uncertainty, specifically in the fields of Safe and Robust RL.

4.1.1 Safe RL

Safe RL is the field of RL that deals with training agents to maximize their expected reward while ensuring system safety and adherence to safety constraints [51]. Many Safe RL techniques rely on adjusting the optimization problem guiding the RL algorithm to ensure system safety. For example control barrier functions may be utilized to ensure that only safe actions can be taken by agents when maximizing their reward [52], [53]. Furthermore, many implementation methods have been developed to improve system safety such as teacher advice and demonstrations. One component of Safe RL is Robust RL, in which the opti-

mization problem is modified to deal with state uncertainty. Given our threat model, we are interested in this subdomain in which spoofing attacks have a direct impact on this state uncertainty.

4.1.2 Robust RL

Robust RL aims to maximize the expected reward of the RL agent using a model that assumes state uncertainty. Many methods for robust RL optimize the minimax criterion by modeling the maximum expected reward in the worst-case scenario:

$$V^*(s) = \min_{p \in \mathcal{P}} \max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi_t, s_{t+1})\right] \quad [54] \tag{4.1}$$

This worst-case criterion can be applied to RL algorithms such as $\hat{Q}$ Learning that applies the minimax criterion to Q Learning:

$$\hat{Q}(s_t, a_t) = \min\left(\hat{Q}(s_t, a_t), r_{t+1} + \gamma \max_{a_{t+1} \in \mathcal{A}} \hat{Q}(s_{t+1}, a_{t+1})\right) \quad [55] \tag{4.2}$$

This minimax criterion can also be applied to adversarial learning by maximizing over the worst-case adversarial perturbations. In other words, a new Bayesian Nash Equilibrium can be estimated by allowing the victim agent to compute a best response to these optimal perturbations:

$$V^*(s) = \min_{a \in \mathcal{A}} \max_{\delta \in \Delta} \mathbb{E}\left[C(s, a) + \gamma \sum_{s'} \mathbb{P}(s' | o + \delta, a) J^{\pi^*}(s')\right] \tag{4.3}$$

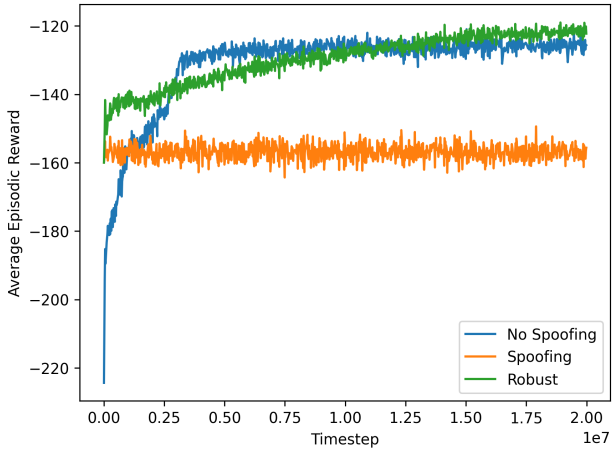
This approach, called Adversarial RL, has been explored in both the single-agent setting [56] as well as the multi-agent setting with algorithms such as M3DDPG that utilizes the minimax criterion for observational perturbations to provide a robust algorithm to perform MADDPG [57]. While these works have built on the offline adversarial example creation method developed by [46], our work provides adversarial examples by training an adversary to provide the perturbations, thus treating this as a multi-player Bayesian Game.

Another advantage of this approach is that it allows both the adversarial agent to improve their performance through self-play by selecting actions with the latest neural network parameters [58]. Self-play allows the victim agent to adapt to the last and likely best adversary, which concurrently develops perturbations in accordance with the last victim agent. This approach has shown the ability to achieve superhuman performance in several adversarial multi-agent domains and can closely estimate a Nash Equilibrium in an extensive form

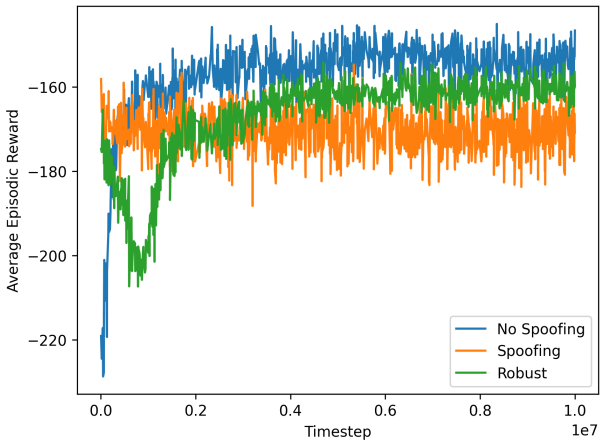
game [59]. We can use this model to develop a robust self-play approach and conduct a safety risk assessment utilizing Robust RL under our cyber threat model.

4.2 Adversarial Improvements to MARL-Based Systems

Firstly, we can simulate the MPE Simple Spread tests again by retraining the cooperative agents in addition to an adversary and measuring system performance.



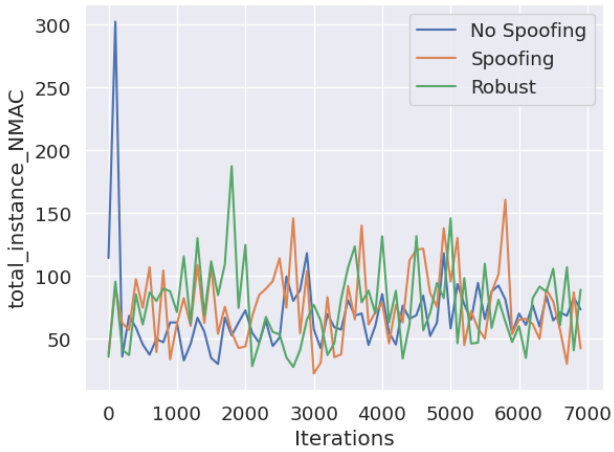
(a) Robust MAPPO Performance



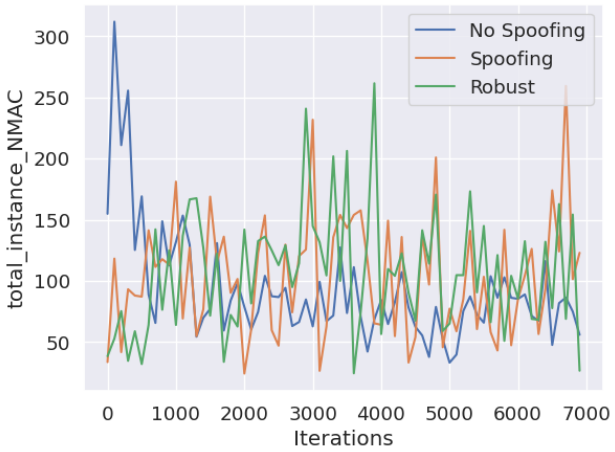
(b) Robust QMix Performance

We can see that utilizing Robust RL for both MAPPO and QMix decreases the total costs of the network. Notably, for MAPPO we actually see a slight performance increase from the baseline using a robust method. This performance improvement is likely due to the fact that collisions still frequently occur during the baseline as the reward signal for collisions in Simple Spread is only -1. The adversarial improvements therefore force MAPPO to be slightly more conservative in its actions leading to a slight decrease in collision frequency.

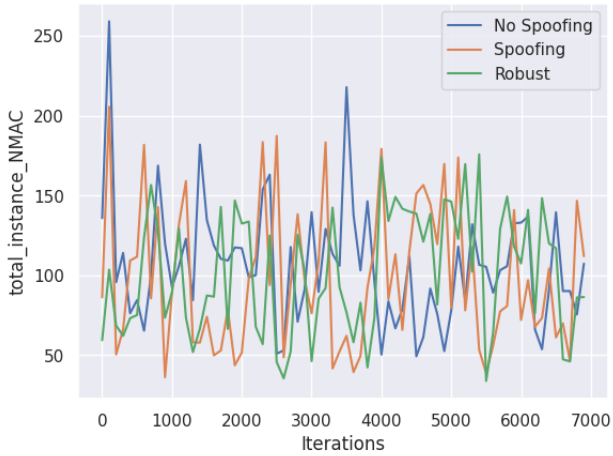
We can also apply this method to an RL-based algorithm in AAM-Gym. We will retrain our layered SACD agents with an adversary and observe NMAC and sNMAC rates.



(a) Robust SACD NMACs 100 Density



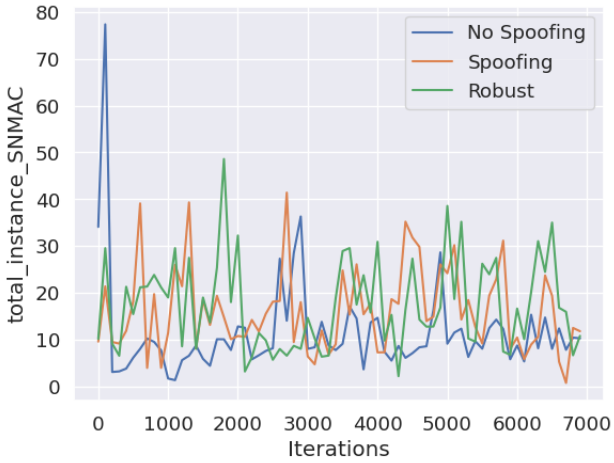
(b) Robust SACD NMACs 200 Density



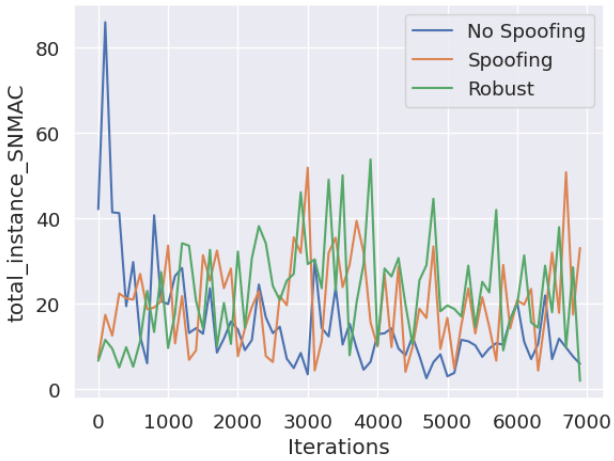
(c) Robust SACD NMACs 300 Density

Figure 4.2: Robust SACD NMAC Spoofing Simulations (Running Average per 100 Iterations)

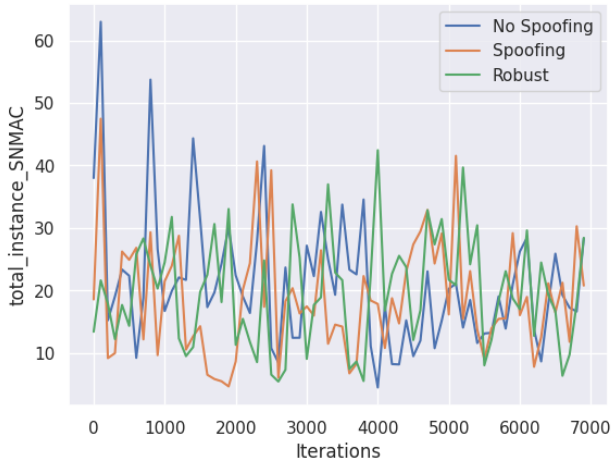
While Robust SACD does not achieve the same performance as the baseline, we can see that in lower aircraft densities, the number of NMACs at environment resets decreases from the non-robust level. Since SACD is not as refined for DAA capabilities, it is likely that ACAS sXu integration could improve this performance.



(a) Robust SACD sNMACs 100 Density



(b) Robust SACD sNMACs 200 Density



(c) Robust SACD sNMACs 300 Density

Figure 4.3: Robust SACD sNMAC Spoofing Simulations (Running Average per 100 Iterations)

Similarly, robust SACD does not improve the stability of the algorithm or improve the system's ability to prevent sNMACs. As with preventing NMACs, sXu integration may improve the system's performance at preventing sNMACs.

4.2.1 Discussion

Overall we find that Robust RL can aid in decreasing the severity of spoofing attacks by considering a cyber adversary during system training, however it does not guarantee system safety alone. Further investigation into robust integration with DAA systems as well as its effect on operational metrics should be conducted, but these results at least suggest a baseline for RL-based systems that could be considered for mitigating spoofing attacks.

4.3 Cyber Defense Techniques

System designers of AAM and other autonomous networked systems should consider practical cyber defenses to protect against spoofing attacks. The success of the spoofing attack depends on the ability of the adversary to send signals to the victim that the victim believes originate from a legitimate source. Since ADS-B does not validate received signals, any received signals are treated as legitimate which creates the vulnerability used in this attack. Two potential defenses that could integrated with AAM are signal validation and filtering.

4.3.1 Signal Validation

In network security, authentication is provided by Public Key Infrastructure (PKI) which ensures that any message encrypted with a private key can only be decrypted with a public key. Therefore if a user sends a message and it is decrypted with their public key, the message must have been signed with their private key, and therefore must have been sent by them. Additionally, Message Authentication Codes (MACs) can be used to ensure integrity of the message by computing a one-time hash of the message that is sent along with the message. When the receiver recomputes a hash of the message it is compared to the transmitted hash to see if they match and the message integrity is preserved. These cryptographic methods provide defenses against a man-in-the-middle attack such as our spoofing attack.

These methods should be implemented for ADS-B spoofing to help improve ACAS sXu. One such proposed method is the Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol [60] which utilizes PKI and MACs to authenticate broadcasted signals. One challenge for this approach is that it does not defend against ghost injection in which a fake AAM agent sends spoofed ADS-B signals, which could be associated with a legitimate private key. Other concepts described in aviation safety literature could be implemented such as multi-constellation and multi-frequency technologies which can help to protect against GPS injection attacks [61]. Additionally, with autonomous networked systems, a trusted third party can be used to aid in message authentication and integrity similarly to how certificate authorities operate in web security [62].

4.3.2 Filtering

We can also leverage the POMDP structure of RL-based systems to provide observational filtering. Using Bayes theorem we can compute a belief state about the true state of the system as shown in Equation 3.7. If we receive an observation that is unlikely given our belief state, we can weight our actions by that knowledge. One method to do this is to

implement particle filters by using sequential importance sampling [63]. Additionally to deal with uncertainty, traditional methods from control theory such as Kalman filters can be used to guide POMDPs by dealing with uncertainty in the observations [64]. This is a process that has been adapted to surveillance systems such as GPS to improve their state estimation capabilities [65]. These filtering techniques could help an RL-based system deal with the added uncertainty introduced by the adversary and thus help defend against potential spoofing attacks.

Chapter 5

Conclusion and Future Work

5.1 Conclusions

This Thesis assessed the safety risk of Advanced Air Mobility systems under the presence of a cyber attack, specifically a spoofing attack. We showed that a spoofing attack is a realistic and relevant attack vector in Advanced Air Mobility systems due to their reliance on unsecure sensing capabilities such as ADS-B. Furthermore, we demonstrated that a spoofing attack can affect the Dynamic Programming and Reinforcement Learning algorithms used for autonomous decision making in Advanced Air Mobility systems. We developed an algorithm for an RL-based spoofing adversary to generate adversarial examples and, by introducing a Bayesian game for our adversary, we demonstrated the worst-case scenario for a trained agent.

Using this algorithm, we showed that a spoofing attack on a single AAM agent can cause an increase in the NMAC risk ratio across multiple aircraft densities. We also showed that this spoofing attack can lead to a noticeable increase in the quantity of sNMACs across multiple aircraft densities as well. Additionally, by demonstrating the loss of system performance in an autonomous networked system in an abstract safety-critical environment through MPE, we can generalize our results in AAM to the safety risks of all autonomous networked systems under this cyber threat model. Lastly, we showed that implementing adversarial reinforcement learning methods integrated with DAA systems such as ACAS sXu could potentially mitigate the impact of observational spoofing attacks. We also recommend cyber defense techniques for addressing this attack vector such as signal validation and filtering.

5.2 Future Work

This Thesis demonstrated a proof of concept attack by targeting a single agent in an AAM network. The threat model that we proposed can be further developed by considering a larger class of potential attacks, namely considering denial of service attacks and broadcasted spoofing attacks that target a physical area rather than a specific agent. Our results also directly motivate the need for the development of sNMAC encounter sets to prove the statistical significance of cyber threats to sUAS safety. Lastly, aviation standards development organizations should consider the the threat of ADS-B spoofing to DAA systems and potentially enact stricter cyber defense standards as these systems are more widely implemented.

Glossary

AAM Advanced Air Mobility. 3, 9, 11

ACAS Airborne Collision Avoidance System. 3

ACAS sXu Airborne Collision Avoidance System sXu. 12

ADS-B Automatic Dependent Surveillance-Broadcast. 3, 17

AFR Autonomous Flight Rules. 12

DAA Detect-and-Avoid. 12, 21

DQN Deep Q Networks. 22

FAA Federal Aviation Administration. 11

FGSM Fast Gradient Sign Method. 26

GPS Global Positioning System. 11

HOVTL Human-Over-the-Loop. 12

ICAO International Civil Aviation Organization. 13

LiDAR Light Detection and Ranging. 11

MAC Mid-Air Collision. 14

MACs Message Authentication Codes. 43

MADDPG Multi-Agent Deep Deterministic Policy Gradients. 24

MAPPO Multi-Agent Proximal Policy Optimization. 24

MARL Multi-Agent Reinforcement Learning. 3, 20, 22

MDP Markov Decision Process. 21

MPE Multi-Particle Environments. 27, 28

NMAC Near Mid-Air Collision. 3, 14

PKI Public Key Infrastructure. 43

POMDP Partially Observable Markov Decision Process. 23

RL Reinforcement Learning. 9, 20

RTCA Radio Technical Commission for Aeronautics. 15

SAC Soft Actor-Critic. 22

SACD Discrete Soft Actor-Critic. 29

sNMAC Small Near Mid-Air Collision. 14

TCAS Traffic Alert and Collision Avoidance System. 15

TESLA Timed Efficient Stream Loss-tolerant Authentication. 43

UAM Urban Air Mobility. 11

UAS Unmanned Aerial Systems. 11

UTM Unmanned Aircraft Systems Traffic Management. 11

UUV Unmanned Underwater Vehicles. 11

VTOL Vertical Takeoff and Landing. 12, 17

WGS 84 World Geodetic System. 11

Appendix A

Algorithm 1 Observational Spoofing Using An Adversarial Agent

$\tau \leftarrow$Environment Transition Dynamics
while $t < T_{\max}$ **do**
  for $i \in \mathcal{I}$ **do**
    $a_i \leftarrow$Train(o_i) ▷ Use MARL algorithm of your choice
  end for
  $\mathbf{a} \leftarrow (a_1, a_2, ..., a_{|\mathcal{I}|})$
  $\mathbf{o}_{t+1} \leftarrow \tau(\mathbf{s}, \mathbf{a}, \mathbf{s}')$
  $t \leftarrow t + 1$
end while
$t \leftarrow 0$
while $t < T_{\max,adv}$ **do**
  $\delta \leftarrow$Train$(\mathbf{o}_t)$ ▷ Use RL algorithm of your choice
  for $i \in \mathcal{I}$ **do**
    if $i =$Victim Agent **then**

$$a_i \leftarrow \min_{a \in \mathcal{A}} \mathbb{E}\left[C(o_i + \delta, a_i) + \gamma \sum_{o'} \mathbb{P}(o'|o_i + \delta, a_i) J^*(o')\right]$$

    else

$$a_i \leftarrow \min_{a \in \mathcal{A}} \mathbb{E}\left[C(o_i, a_i) + \gamma \sum_{o'} \mathbb{P}(o'|o_i, a_i) J^*(o')\right]$$

    end if
  end for
  $\mathbf{a}_t \leftarrow (a_{1,t}, a_{2,t}, ..., a_{|\mathcal{I}|,t})$
  $\mathbf{o}_{t+1} \leftarrow \tau(\mathbf{s}, \mathbf{a}_t, \mathbf{s}')$
  $t \leftarrow t + 1$
end while

Algorithm 2 Adversarial ADS-B Spoofing Attack in AAM-Gym

$T \leftarrow$ Rollout Length
Inner Agent $\leftarrow o_t$
$a_{\text{inner},t} \leftarrow \min_a \mathbb{E}\left[\sum_{\tau=t}^{T} \gamma^t R_{\text{inner}}(o_t, a_t, o_{t+1})\right]$
Outer Agent $\leftarrow (o_t, a_{\text{inner},t})$
if Aircraft is spoofed **then**
    $a_{\text{outer},t} \leftarrow \max_\delta \mathbb{E}\left[\sum_{\tau=t}^{T} \gamma^t R_{\text{outer}}(o_t, a_{\text{inner},t}, o_{t+1} + \delta)\right]$
    if NMAC exists **then**
        $R_{\text{outer}} \leftarrow 1$
        $R_{\text{inner}} \leftarrow -1$
    else
        $R_{\text{outer}} \leftarrow \alpha \cdot \beta \cdot \min$ [Intruder Distances]
        $R_{\text{inner}} \leftarrow -\alpha \cdot \beta \cdot \min$ [Intruder Distances]
    end if
else
    $a_{\text{outer},t} \leftarrow 0$
    if NMAC exists **then**
        $R_{\text{outer}}, R_{\text{inner}} \leftarrow -1$
    else
        $R_{\text{outer}}, R_{\text{inner}} \leftarrow -\alpha \cdot \beta \cdot \min$ [Intruder Distances]
    end if
end if
$\tilde{a}_t \leftarrow a_{\text{inner},t}$
$o_{t+1} \leftarrow T(s_t, \tilde{a}_t) + a_{\text{outer},t}$
$t \leftarrow t + 1$

References

[1] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1130–1151, 2005.

[2] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[3] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, *et al.*, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[6] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," *arXiv preprint arXiv:1702.02284*, 2017.

[7] J. Tisdale, Z. Kim, and J. K. Hedrick, "Autonomous uav path planning and estimation," *IEEE Robotics & Automation Magazine*, vol. 16, no. 2, pp. 35–42, 2009.

[8] B. Li, Z. Fei, and Y. Zhang, "Uav communications for 5g and beyond: Recent advances and future trends," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, 2018.

[9] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[10] A. Nguyen, T. Do, M. Tran, B. X. Nguyen, C. Duong, T. Phan, E. Tjiputra, and Q. D. Tran, "Deep federated learning for autonomous driving," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2022, pp. 1824–1830.

[11] Federal Aviation Administration, "Unmanned Aircraft Systems (UAS) Traffic Management (UTM) Concept of Operations," Department of Transportation, 2020.

[12] Federal Aviation Administration, "Urban Air Mobility (UAM) Concept of Operations," Department of Transportation, 2023.

[13] Wisk and Boeing, "Concept of Operations for Uncrewed Urban Air Mobility," The Boeing Company, 2022.

[14] J. E. Holland, M. J. Kochenderfer, and W. A. Olson, "Optimizing the next generation collision avoidance system for safe, suitable, and acceptable operational performance," *Air Traffic Control Quarterly*, vol. 21, no. 3, pp. 275–297, 2013.

[15] A. Weinert, L. Alvarez, M. Owen, and B. Zintak, "Near midair collision analog for drones based on unmitigated collision risk," *Journal of Air Transportation*, vol. 30, no. 2, pp. 37–48, 2022.

[16] L. E. Alvarez, I. Jessen, M. P. Owen, J. Silbermann, and P. Wood, "Acas sxu: Robust decentralized detect and avoid for small unmanned aircraft systems," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019, pp. 1–9. DOI: 10.1109/DASC43569.2019.9081631.

[17] M. Kochenderfer and J. Chryssanthacopoulos, "Robust airborne collision avoidance through dynamic programming," Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, Massachusetts, Tech. Rep. ATC-371, Jan. 2011, p. 116.

[18] A. J. Weinert, E. P. Harkleroad, J. Griffith, M. W. Edwards, and M. J. Kochenderfer, "Uncorrelated encounter model of the national airspace system version 2.0," *Project Report ATC-404, Massachusetts Institute of Technology, Lincoln Laboratory*, 2013.

[19] G. Cybenko and R. A. Hallman, "Attritable multi-agent learning," in *Disruptive Technologies in Information Sciences V*, SPIE, vol. 11751, 2021, pp. 92–97.

[20] K. L. Best, J. Schmid, S. Tierney, M. J. Awan, N. M. Beyene, M. A. Holliday, R. Khan, and K. Lee, *How to Analyze the Cyber Threat from Drones: Background, Analysis Frameworks, and Analysis Tools*. Santa Monica, CA: RAND Corporation, 2020. DOI: 10.7249/RR2972.

[21] K. Jindal, S. Dalal, and K. K. Sharma, "Analyzing spoofing attacks in wireless networks," in *2014 Fourth International Conference on Advanced Computing Communication Technologies*, 2014, pp. 398–402. DOI: 10.1109/ACCT.2014.46.

[22] S. Peterson and P. Faramarzi, "Exclusive: Iran hijacked US drone, says Iranian engineer," *Christian Science Monitor*, Dec. 15, 2011. [Online]. Available: https://www.csmonitor.com/World/Middle-East/2011/1215/Exclusive-Iran-hijacked-US-drone-says-Iranian-engineer (visited on 12/15/2011).

[23] A. Costin and A. Francillon, "Ghost in the Air (Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices," *Black Hat USA*, vol. 1, pp. 1–12, 2012.

[24] E. Tegler, "Could Hamas Be Jamming GPS In and Around Gaza?" *Forbes*, Oct. 30, 2023. [Online]. Available: https://www.forbes.com/sites/erictegler/2023/10/30/could-hamas-be-jamming-gps-in-and-around-gaza/?sh=4f6d39276aaa (visited on 10/30/2023).

[25] B. Cole, "NATO Nation Hit By 'Unprecedented' GPS Attack," *Newsweek*, Jan. 2, 2024. [Online]. Available: https://www.newsweek.com/finland-nato-gps-attack-unprecedented-1857098 (visited on 01/04/2024).

[26] S. Malik and W. Sun, "Analysis and Simulation of Cyber Attacks Against Connected and Autonomous Vehicles," in *2020 International Conference on Connected and Autonomous Driving (MetroCAD)*, 2020, pp. 62–70. DOI: 10.1109/MetroCAD48866.2020.00018.

[27] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 11, pp. 2898–2915, 2017.

[28] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, "Cyber security threat analysis and modeling of an unmanned aerial vehicle system," in *2012 IEEE conference on technologies for homeland security (HST)*, IEEE, 2012, pp. 585–590.

[29] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics: Results of the 11th International Conference*, Springer, 2018, pp. 621–635.

[30] A. R. Perry, "The flightgear flight simulator," in *Proceedings of the USENIX annual technical conference*, vol. 686, 2004, pp. 1–12.

[31] M. Strohmeier, G. Tresoldi, L. Granger, and V. Lenders, "Building an avionics laboratory for cybersecurity testing," in *Proceedings of the 15th Workshop on Cyber Security Experimentation and Test*, 2022, pp. 10–18.

[32] S. Crow, B. Farinholt, B. Johannesmeyer, K. Koscher, S. Checkoway, S. Savage, A. Schulman, A. C. Snoeren, and K. Levchenko, "Triton: A {software-reconfigurable} federated avionics testbed," in *12th USENIX Workshop on Cyber Security Experimentation and Test (CSET 19)*, 2019.

[33] D. Ingegneri, D. Timoteo, P. Hyle, F. Parraga, and A. Reyes, "A cybersecurity test and evaluation facility for the next generation air transportation system (nextgen)," in *9th Workshop on Cyber Security Experimentation and Test (CSET 16)*, 2016.

[34] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[35] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[37] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, PMLR, 2018, pp. 1861–1870.

[38] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

[39] J. F. Nash Jr, "Equilibrium points in n-person games," *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[40] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[41] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Neural Information Processing Systems (NIPS)*, 2017.

[42] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[44] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

[45] J. Lin, K. Dzeparoska, S. Q. Zhang, A. Leon-Garcia, and N. Papernot, "On the robustness of cooperative multi-agent reinforcement learning," in *2020 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2020, pp. 62–68.

[46] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[47] I. Mordatch and P. Abbeel, "Emergence of grounded compositional language in multi-agent populations," *arXiv preprint arXiv:1703.04908*, 2017.

[48] M. Brittain, L. E. Alvarez, K. Breeden, and I. Jessen, "AAM-Gym: Artificial Intelligence Testbed for Advanced Air Mobility," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, IEEE, 2022, pp. 1–10.

[49] G. Papoudakis, F. Christianos, L. Schäfer, and S. V. Albrecht, "Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks," *arXiv preprint arXiv:2006.07869*, 2020.

[50] A. Reuther, J. Kepner, C. Byun, *et al.*, "Interactive supercomputing on 40,000 cores for machine learning and data analysis," in *2018 IEEE High Performance extreme Computing Conference (HPEC)*, IEEE, 2018, pp. 1–6.

[51] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[52] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 3387–3395.

[53] Z. Cai, H. Cao, W. Lu, L. Zhang, and H. Xiong, "Safe multi-agent reinforcement learning through decentralized multiple control barrier functions," *arXiv preprint arXiv:2103.12553*, 2021.

[54] J. A. Bagnell, A. Y. Ng, and J. G. Schneider, "Solving uncertain markov decision processes," 2001.

[55] M. Heger, "Consideration of risk in reinforcement learning," in *Machine Learning Proceedings 1994*, Elsevier, 1994, pp. 105–111.

[56] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *International Conference on Machine Learning*, PMLR, 2017, pp. 2817–2826.

[57] S. Li, Y. Wu, X. Cui, H. Dong, F. Fang, and S. Russell, "Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 4213–4220.

[58] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *arXiv preprint arXiv:1712.01815*, 2017.

[59] J. Heinrich and D. Silver, "Deep reinforcement learning from self-play in imperfect-information games," *arXiv preprint arXiv:1603.01121*, 2016.

[60] A. Perrig, D. Song, R. Canetti, J. Tygar, and B. Briscoe, "Timed efficient stream loss-tolerant authentication (tesla): Multicast source authentication transform introduction," Tech. Rep., 2005.

[61] Joint Authorities for Rulemaking on Unmanned Systems, "Annex E - JARUS guidelines on Specific Operations Risk Assessment," JARUS, 2022.

[62] Z. Wu, A. Guo, M. Yue, and L. Liu, "An ads-b message authentication method based on certificateless short signature," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 1742–1753, 2019.

[63] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, pp. 197–208, 2000.

[64] M. Geist and O. Pietquin, "Kalman temporal differences," *Journal of artificial intelligence research*, vol. 39, pp. 483–532, 2010.

[65] X. Gao, H. Luo, B. Ning, F. Zhao, L. Bao, Y. Gong, Y. Xiao, and J. Jiang, "Rl-akf: An adaptive kalman filter navigation algorithm based on reinforcement learning for ground vehicles," *Remote Sensing*, vol. 12, no. 11, p. 1704, 2020.