

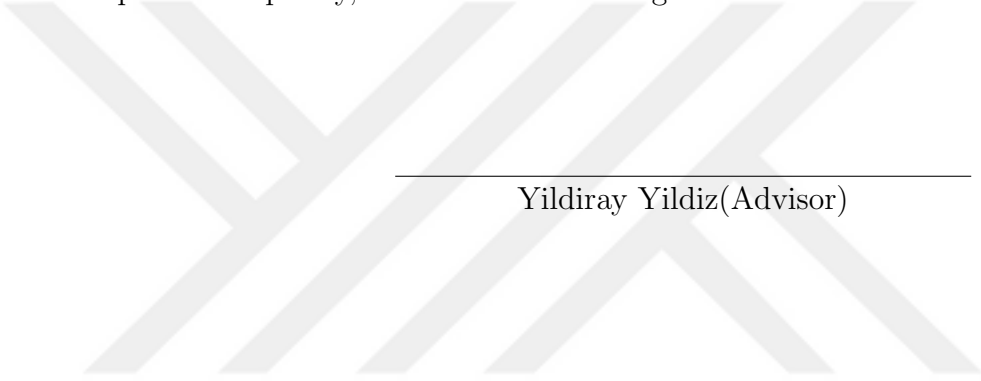
A GAME THEORETICAL FRAMEWORK FOR THE EVALUATION OF UNMANNED AIRCRAFT SYSTEMS AIRSPACE INTEGRATION CONCEPTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

By
Neginsadat Musavi
July 2017

A Game Theoretical Framework for the Evaluation of Unmanned
Aircraft Systems Airspace Integration Concepts
By Neginsadat Musavi
July 2017

We certify that we have read this thesis and that in our opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.



Yildiray Yıldız(Advisor)

Melih Çakmakçı

Ali Türker Kutay

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

A GAME THEORETICAL FRAMEWORK FOR THE EVALUATION OF UNMANNED AIRCRAFT SYSTEMS AIRSPACE INTEGRATION CONCEPTS

Neginsadat Musavi

M.S. in Mechanical Engineering

Advisor: Yildiray Yildiz

July 2017

Predicting the outcomes of integrating Unmanned Aerial Systems (UAS) into the National Aerospace (NAS) is a complex problem which is required to be addressed by simulation studies before allowing the routine access of UAS into the NAS. This thesis focuses on providing 2D and 3D simulation frameworks using a game theoretical methodology to evaluate integration concepts in scenarios where manned and unmanned air vehicles co-exist. The fundamental gap in the literature is that the models of interaction between manned and unmanned vehicles are insufficient: a) they assume that pilot behavior is known a priori and b) they disregard decision making processes. The contribution of this work is to propose a modeling framework, in which, human pilot reactions are modeled using reinforcement learning and a game theoretical concept called level-k reasoning to fill this gap. The level-k reasoning concept is based on the assumption that humans have various levels of decision making. Reinforcement learning is a mathematical learning method that is rooted in human learning. In this work, a classical and an approximate reinforcement learning (Neural Fitted Q Iteration) methods are used to model time-extended decisions of pilots with 2D and 3D maneuvers. An analysis of UAS integration is conducted using example scenarios in the presence of manned aircraft and fully autonomous UAS equipped with sense and avoid algorithms.

Keywords: UAS Integration, Game theory, Reinforcement learning.

ÖZET

TÜRKÇE BAŞLIK

Neginsadat Musavi
Makine Mühendisliği, Yüksek Lisans
Tez Danışmanı: Yildiray Yıldız
Temmuz 2017

İnsansız Hava Araçlarını (İHA), ulusal hava sistemine entegre etmenin sonuçlarını tahmin etmek zordur. Bu nedenle İHAların, ulusal hava sistemine entegre edilmeden önce simülasyon çalışmaları yapılması gerekmektedir. Bu tez, insanlı ve insansız hava araçlarının bir arada bulunduğu senaryolarda, entegrasyon konusunu değerlendirmek için oyun teorisi metodunu kullanarak 2D ve 3D simülasyon çerçevelerini geliştirmeye odaklanmaktadır. Literatürdeki temel boşluk, insanlı ve insansız hava araçları arasındaki etkileşim modellerinin yetersiz olmasıdır. Literatürde, a) pilot davranışının önceden bilindiği varsayılmıştır ve b) pilot karar alma süreçleri görmezden gelinmiştir. Bu boşluğu doldurmak için, insan pilot davranışının seviye-k düşünme (level-k reasoning) kavramı ve pekiştirmeli öğrenme (reinforcement learning) metodu kullanılarak modellendiği bir modelleme çerçevesi önerilerek alana katkıda bulunulması amaçlanmıştır. Seviye-k düşünme kavramı, oyun teorisine ait bir kavramdır ve insanların çeşitli karar verme düzeylerine sahip olduğu varsayımına dayanır. Pekiştirmeli öğrenme, insan öğrenimini taklit eden bir matematiksel öğrenme yöntemidir. Bu çalışmada, pilotların zamanda genişletilmiş kararlarını modellemek için klasik ve yaklaşık pekiştirmeli öğrenme (Nural Fitted Q Iteration) yöntemleri kullanılmıştır. İHA entegrasyon analizi, insanlı ve insansız hava araçlarının bulunduğu örnek senaryolarla, “algıla ve önle” algoritmaları ile donatılmış otonom İHAlar kullanılarak yapılmıştır.

Anahtar sözcükler: İHA Entegrasyonu, Oyun Teorisi, Pekiştirmeli Öğrenme.

Acknowledgement

This effort was sponsored by the Scientific and Technological Research Council of Turkey under grant number 11E282.

I would like to express my sincere appreciation to my adviser Assistant Professor Dr. Yildiray Yildiz for his patience, support, motivation, professional academic behavior and for most for his supervision. I would like to state that I hope for future academic collaborations with him.

I would like to thank to my thesis committee, Assistant Professor Dr. Melih akmakçı and Assistant Professor Dr. Ali Türker Kutay for allocating their time to investigate my work.

I am glad to express my pleasure for having Noushin Salek Faramarzi (Noosh) as a friend. She has always been a sincere and positive friend. I also thank to my vibrant and happy Nasima Afsharimani, dear Ehsan Yousefi and dear Arsalan Nikdoost for their kindness toward me during my graduate studies at Bilkent University. There are so nice friends at Bilkent University that made me feel like I am at home, thank to Başak Avcı, Elif Altıntepe, Tuna Demirbaş and Nurten Bulduk.

Last but not least, I wish to thank to my dear father for his endless support in all of the stages of my life. I thank to my mother for her patience and I thank to my dear friend Mahsa Asgarisabet for her passionate friendship.

I dedicate this thesis to my dear brother Nima. He will survive in my heart and mind forever. May he rest in peace.

Contents

1	Introduction	1
1.1	Objective and Motivation	1
1.2	Approach	3
1.3	Organization	7
2	2D Framework	8
2.1	Modeling Methodology	8
2.1.1	Game Theoretical Modeling of Interactive Decision Making	9
2.1.2	Reinforcement Learning for the Partially Observable Markov Decision Process	9
2.1.3	Combining Game Theory With Reinforcement Learning . .	11
2.2	Components of the Hybrid Airspace Scenario	11
2.2.1	Pilot Observations and Memory	13
2.2.2	Pilot Objective Function	14

2.2.3	Manned Aircraft Model	15
2.2.4	UAS Model	16
2.2.5	Sense And Avoid Algorithms	16
2.2.6	Model Validation	17
2.3	Simulation Results and Discussion	26
2.3.1	Hybrid Airspace Scenarios with a Single Encounter	26
2.3.2	Hybrid Airspace Scenario with Multiple Encounters	28
3	3D Framework	40
3.1	UAS Integration Scenario	40
3.1.1	UAS Conflict Detection And Avoidance Logic	41
3.1.2	Manned Aircraft	43
3.2	Pilot Decision Model	43
3.2.1	Dynamic Level-k Reasoning	44
3.2.2	Neural Fitted Q Iteration	45
3.3	Simulation Results and Discussion	48
3.3.1	Single Encounter Scenarios Between Manned Aircraft	48
3.3.2	Statistical Results For UAS Integration	55
4	Conclusion	60

List of Figures

2.1	Snapshot of the hybrid airspace scenario in the simulation platform. Each square stands for a $5nm \times 5nm$ area.	12
2.2	Pilot observation space.	14
2.3	Comparison of the trajectories created by the validated model and the game theoretical modeling approach for sample encounter number 3.	21
2.4	Comparison of the trajectories created by the validated model and the game theoretical modeling approach for sample encounter number 16.	22
2.5	Comparison of the trajectories created by the validated model and the game theoretical modeling approach for sample encounter number 23.	23
2.6	Comparison of the trajectories created by the validated model and the game theoretical modeling approach for sample encounter number 34.	24
2.7	Comparison of the trajectories created by the validated model and the game theoretical modeling approach for sample encounter number 45.	25

2.8	Level-1 and level-2 pilots interacting with UAS.	29
2.9	Safety vs. performance of Level-k pilot interacting with UAS . . .	30
2.10	Pilot model sensitivity analysis.	32
2.11	Pilot model sensitivity analysis for the single encounter scenario. .	32
2.12	Safety vs. performance in HAS, when SAA1 is employed.	35
2.13	Safety vs. performance in HAS, when SAA2 is employed.	36
2.14	Safety vs. performance in HAS, when SAA1 is employed, in a multi-UAS scenario.	37
2.15	SAA1, SAA2 and safety vs. performance in HAS.	39
3.1	2D snapshot of the airspace scenario in the simulation platform. .	41
3.2	Dynamic level-k reasoning pseudo-code.	45
3.3	Pilot observation space.	49
3.4	Encounter categories.	50
3.5	Separation violation rates. “dst hor” refers to the distance horizon of the pilots. On the columns that shows the separation violation rates for a $10nm$ distance horizon, the cyan color shows the per- centage of violations that occur when the encounters are “difficult to resolve”, meaning that the initial conditions of the encounters do not permit any type of pilot action to avoid a separation violation.	53
3.6	Sample encounter 1: level-1 pilot vs. level-0 pilot.	53
3.7	Sample encounter 1: level-2 pilot vs. level-1 pilot.	53

3.8	Sample encounter 1: level-1 pilot vs. level-1 pilot.	54
3.9	Sample encounter 1: dynamic level-1 pilot vs. level-1 pilot.	54
3.10	Sample encounter 1: horizontal distance.	54
3.11	Sample encounter 1: vertical distance.	55
3.12	Sample encounter 2: level-1 pilot vs. level-0 pilot.	55
3.13	Sample encounter 2: level-2 pilot vs. level-1 pilot.	56
3.14	Sample encounter 2: level-1 pilot vs. level-1 pilot.	56
3.15	Sample encounter 2: dynamic level-1 pilot vs. level-1 pilot.	56
3.16	Sample encounter 2: horizontal distance.	57
3.17	Sample encounter 2: vertical distance.	57
3.18	Average trajectory deviation of manned aircraft.	58
3.19	Average trajectory deviation of UAS.	58
3.20	UAS flight time.	59
3.21	Separation violation between manned aircraft and UAS.	59

Chapter 1

Introduction

1.1 Objective and Motivation

Unmanned Aerial Systems (UAS) refer to aircraft without an on-board human pilot. Instead, UAS can be controlled with an off-board operator or can be programmed to fly autonomously. UAS have operational and cost advantages over manned aircraft in many areas such as surveillance, commercial, scientific and agricultural, and the interest in UAS is increasing rapidly. However, UAS still do not have routine access to National Airspace System (NAS) [1]. Since technologies, standards and procedures for a safe integration of UAS into airspace haven't matured yet, UAS can fly only in segregated airspace with restricting rules in most of the countries. For instance, Federal Aviation Administration (FAA) in USA have regulations which prevents routine UAS operations in the NAS. Aviation industry is very sensitive to risk and for new vehicles such as UAS to enter into this sector, they need to be proven to be safe and it must be shown that they will not affect the existing airspace system in any negative way [2, 3]. Technical issues such as communication between the UAS and air traffic control as well as a trustworthy Sense and Avoid (SAA) system are among the primary impediments that prevent the integration of the UAS into the NAS [1]. Sense and avoid capability refers to the ability of UAS to sense, detect and execute

maneuvers to resolve a conflict with other aircraft and obstacles in the surrounding traffic. Until technologies, standards, requirements and procedures for a safe integration of UAS into airspace are matured, there will not be enough data accumulated about the technical issues and it will be hard to predict the effects of the technologies and concepts that are developed for the integration. Although research efforts exist to develop a safe and efficient real test environment for UAS integration [4], experimental tests are expensive and experimental failures can cause severe economic loss. Therefore, employing simulations is currently the only way to understand the effects of UAS integration on the air traffic system [5]. These simulation studies need to be conducted with hybrid airspace system (HAS) models, where manned and unmanned vehicles coexist.

Before an SAA system is approved, its potential impact on the safety of the surrounding air traffic should be analyzed. To perform an analysis and evaluation of any SAA logic, it is necessary to model the actions that a pilot would take during a conflict [6]. HAS models in the literature are generally based on the assumption that the pilots of manned aircraft always behave as they should, without deviating from the ideal behavior. In their work, [6], Maki *et al.* constructed an SAA logic based on the model developed by the MIT Lincoln Laboratory [7] and Kuchar *et al.* [8] did a rigorous analysis of the traffic control alert system (TCAS) to implement an SAA algorithm for remotely piloted vehicles. TCAS is an on-board collision avoidance system which observes and tracks surrounding air traffic, detects conflicts and suggests avoidance maneuvers to the pilots. In both of these studies, SAA system are evaluated through simulations in a platform which uses MIT's NAS encounter model. In the evaluations, it is assumed that pilot decisions are known a priori and depend on the relative motion of the adversary during a specific conflict scenario. In their work, Perez-Batlle *et al.* [9] classified separation conflicts between manned and unmanned aircraft and proposed separation maneuvers for each class. These maneuvers are tested in a simulation environment, where it is assumed that the pilots will follow these maneuvers 100% of the time. Florent *et al.* [10] developed a SAA algorithm and tested it via simulations and experiments. In both of these tests, it is assumed

that the intruding aircraft does not change its path while the UAS is implementing the SAA algorithm. There are other simulation studies such as [11] that test and evaluate different collision avoidance algorithms, where some predefined actions are used as pilot models. There are also substantial studies with remotely piloted aircraft where the effects of SAA systems on the workload and situational awareness of the pilots are investigated via simulations and flight tests [12], [13]. These HAS models are designed to evaluate and test the performance of collision avoidance systems in single encounter scenarios in which the intruder (generally a manned aircraft) has a pre-defined behavior with no consideration of the decision making process of the pilot. These models are valuable and essential at the initial stages of evaluating a new method but it is not realistic to expect that the pilot, as a decision maker (DM), will always behave deterministically and in a pre-defined manner. It is not always predictable, for example, how pilots will respond to the TCAS [14]. The collision between two aircraft (a DHL Boeing 757 and a Bashkirian Tupolev 154) over Uberlingen, Germany, near the Swiss border On 21:35 (UTC) July 1, 2002, is a good evidence that pilots may decide not to act parallel to TCAS advisory or may ignore traffic controller's commands, during high-stress situations [14]. In addition, in recent studies, it was shown that only 13% of pilot responses match the deterministic pilot model that was assumed for TCAS development [15], [16]. Therefore, incorporating human decision-making processes in HAS models has a strong potential to improve the predictive power of these models.

1.2 Approach

In this thesis, a 2D and a 3D game theoretical Hybrid Airspace System (HAS) modeling framework are built, where pilot reactions are obtained through a decision making process. Below, 2D and 3D frameworks are explained separately.

2D framework: In the 2D HAS model, the pilot behavior is not assumed to be known a priori and decisions are obtained utilizing a) the bounded rationality concept, which helps model imperfect decisions, as opposed to modeling the pilot

as a perfect decision maker, b) reinforcement learning, which helps model time-extended decisions, as opposed to assuming one-shot decision making. In order to predict pilot reactions in complex scenarios where UAS and manned aircraft co-exist, in the presence of automation such as an SAA system, a game theoretical methodology is employed, which is formally known as semi network-form games (SNFGs) [15]. Using this method, probable outcomes of HAS scenarios are obtained that contains interacting humans (pilots) who also interact with UAS equipped with an SAA algorithm. In these scenarios, close encounters are simulated where TCAS and air traffic management instructions can also be incorporated. To obtain realistic pilot reactions, bounded rationality is imposed by utilizing the level-k approach [17, 18], a concept in game theory which models human behavior assuming that humans think in different levels of reasoning. In the proposed framework, pilots optimize their trajectories based on a goal function representing their preferences for system states. During the simulations, UAS fly autonomously based on a pre-programmed flight plan. In these simulations, the effect of certain system variables, such as *horizontal separation requirement* and *required time to conflict* for UAS and the effect of *responsibility assignment for conflict resolutions* on the safety and performance of the HAS are analyzed (see [4] for the importance of these variables and responsibility assignment for UAS integration.). To enable UAS to perform autonomously in the simulations, it is assumed that they employ an SAA algorithm. The simulation results are provided for 2 different SAA methods and, in addition, these 2 methods are compared quantitatively in terms of safety and performance, using the proposed modeling framework.

In prior works, the method exploited in this paper was used to investigate small scale scenarios (in terms of number of agents): In [19], the dynamics between a smart grid operator and a cyber attacker, and in [20, 21, 15], the dynamics between two interacting pilots are modeled. More recently, in [22], a medium scale scenario with 50 interacting pilots is analyzed. In the study with 50 pilots, the simulation environment utilized a gridded airspace where aircraft moved from a grid intersection to another to represent movement. In addition, the pilots could only observe grid intersections to see whether or not another aircraft was nearby.

All these simplifying assumptions decreased the computational cost but also decreased the fidelity of the simulation. In this thesis, a) a dramatically more complex scenario in the presence of manned and unmanned aircraft is investigated, b) the simulation environment is not discretized and the aircraft movements are simulated in continuous time, c) realistic aircraft and UAS physical models are used and d) initial states of the aircraft are obtained from real flight data. Hence, a much more representative simulation environment with the inclusion of UAS equipped with a SAA algorithm is used to obtain probabilistic outcomes of HAS scenarios.

3D framework: The 2D game theoretical approach has two limitations: First, HAS models are developed for a 2D airspace. Second, the policies, i.e. maps from observation spaces to action spaces, obtained for the decision makers remain unchanged during their interaction. To remove these limitations the 2D framework is extended and a 3D HAS model is introduced where the strategic decision makers can modify their policies during interactions between each other. Therefore, compared to the 2D HAS model a much larger class of interactions are modeled.

It is shown in the literature that 1) in repeated strategic interactions, where agents consider other agents' possible actions before determining their own, agents with different cognitive abilities change their behavior during the interaction [25]; and 2) there is a positive relationship between cognitive ability and reasoning levels [26] and [25]. These observations lead to agents with different levels of reasoning who can observe their opponents behavior during repeated interactions, update their beliefs on their opponents reasoning level and change their own level-k rule against them. In their works in [26] and [25], authors introduce a systematic level-k structure where players can update their beliefs about their opponents, and switch their own level rule up one level during their interactions. There are also other level-k rule learning models in the literature such as the one presented in [27] and [28]. Based on the rule learning methods introduced in these works the agent levels can reach up to infinity. This is not a problem for the applications investigated in these studies since they are executed on games in the field of economics in which obtaining level-k rules ($k=0,1,2,\dots,\text{infinity}$) are straight forward due to the number of agents in the games. Since 1) it is

computationally expensive to obtain higher levels, and 2) in certain experimental studies it is shown that humans in general have a maximum reasoning level of 2 [17], the existing level-k rule learning methods may not be suitable for the application considered in this work where more than 180 decision makers are modeled simultaneously in a time extended manner. In this study, we propose a simpler method for modeling level rule update during interactions by a) limiting the levels up to 2 and b) allowing rule update only if a trajectory conflict is detected.

Different from the 2D HAS model developed in [23], [24], in this study, the game theoretical modeling framework is developed for a 3D HAS model which allows to cover a much larger class of integration scenarios. The reinforcement learning algorithm used in the authors' earlier works [23], [24] employs a table to store the Q values of all state (location of the intruder in a girded observation space, approach angle of the intruder, best trajectory action, best destination action and previous action)-action (turn left, turn right, go straight) pairs, which define how preferable to take a certain action given the observations/states. This poses a challenge for the application of the method to systems with a large numbers of state-action pairs such as the proposed 3D HAS model in this study. To circumvent this issue, Neural Fitted Q Iteration (NFQ) method [29], [30] and [31], an approximate reinforcement learning algorithm is utilized. Approximate reinforcement learning methods use function approximators to represent the Q value function [32]. In other words, instead of saving Q values for each state-action pair, Q value function is approximated by a function approximator. In the case of NFQ, a neural network is used as the function approximator. NFQ approach also allowed using a continuous observation space, which also contributed to obtain a more precise definition of the agents observations, compared to [our earlier papers], where a discretized observation space was used.

In the simulations, pilot models, that are obtained using the proposed game theoretical modeling framework, are used in complex scenarios, where UAS and manned aircraft co-exist, to analyze the probable outcomes of HAS scenarios. The HAS scenarios contain interacting humans (pilots) who also interact with multiple UAS with their own sense and avoid (SAA) systems. It is noted that automation

other than SAA systems, such as TCAS, and possible air traffic management instructions can also be incorporated into the proposed framework. During the simulations, UAS fly autonomously based on a pre-programmed flight plan. In these simulations, the effect of responsibility assignment for conflict resolutions on the safety and performance of the HAS are analyzed.

1.3 Organization

The organization of the thesis in the following chapters is described as below: Chapter II is devoted to the 2D game theoretical framework. The modeling methodology as well as details of the scenario consisting of multiple manned aircraft and a UAS is described completely. The obtained 2D HAS model is validated and the results of simulation for the UAS integration is shown at the end of chapter. In Chapter III, the 3D game theoretical pilot model and the modifications that are done upon the 2D HAS model are explained in detail. The rest of chapter is devoted to the results of simulation in single encounter scenarios and UAS integrated scenario. Finally, concluding remarks are provided in Chapter IV.

Chapter 2

2D Framework

2.1 Modeling Methodology

The most challenging problem in the prediction of the outcomes of complex scenarios where manned and unmanned aircraft co-exist is obtaining realistic pilot models. A pilot model in this paper refers to a mapping from observations of the pilot to his/her actions. To achieve a realistic human reaction model, certain requirements need to be met. First, the model should not be deterministic since it is known from everyday experience that humans do not always react exactly the same when they are in a given “state”. Here, “state” refers to the observations and the memory of the pilot. For instance, observing that an aircraft is approaching from a certain distance is an observation and remembering one’s own previous action is memory. Second, pilots should show the characteristics of a strategic Decision Maker (DM) meaning that the decisions must be influenced by the expected moves of other “agents”. Agents can be either the other pilots or the automation logic of UAS. Third, the decisions emanating from the model should not always be the best (or mathematically optimal) decisions since it is known that human actions are less than optimal in many situations. Finally, it should be considered that a human DM’s predictions about other human DMs are not *always* correct. To accomplish all of these requirements, level-k reasoning

and reinforcement learning are used together, forming a non-equilibrium game theoretical solution concept. It is noted that in this study, the UAS are assumed to be fully autonomous.

2.1.1 Game Theoretical Modeling of Interactive Decision Making

Level-k reasoning is a game theoretical solution concept whose main idea is that humans have various levels of reasoning in their decision making process. Level-0 represents a “non-strategic” DM who does not take into account other DMs’ possible moves when choosing his/her own actions. This behavior can also be named as reflexive since it only reacts to the immediate observations. In this study, given a state, a level-0 pilot flies an aircraft with constant speed and heading starting from its initial position toward its destination. A level-1 DM assumes that the other agents in the scenario are level-0 and takes actions accordingly to maximize his/her rewards. A level-2 DM takes actions as though the other DMs are level-1. In a hierarchical manner, a level-k DM takes actions assuming that the other DMs behave as level-(k-1) DMs.

2.1.2 Reinforcement Learning for the Partially Observable Markov Decision Process

Reinforcement learning is a mathematical learning mechanism which mimics the human learning process. An agent receives an observable message of an environment’s state, then chooses an action, which changes the environment’s state, and the environment in return encourages or punishes the agent with a scalar reinforcement signal known as reward. Given a state, when an action increases (decreases) the value of an objective function (reward), which defines the goals of the agent, the probability of taking that action increases (decreases). Reinforcement learning algorithms mostly involve estimating two value functions: State value function V and state-action value function Q [33]. $V(s)$, the value of the

state “s”, is the total amount of reward an agent can expect to gather in the future, starting from that state. It is an estimate of how good it is for an agent to be in a particular state. $Q(s, a)$, the value of taking the action “a” in the state “s”, under a particular policy, is the total amount of reward an agent can expect to gather in the future starting from that state and taking that action by following the given policy. It is an estimate of how good it is for an agent to perform a given action in a given state.

Since the human DM as the agent in the reinforcement learning process is not able to observe the whole environment state (i.e. the positions of all of the aircraft in the scenario), the agent receives only partial state information. By assuming that the environment is Markov, and based on the partial observability of the state information, this problem can be generalized to a Partially Observable Markov Decision Process (POMDP). In this study, the reinforcement learning algorithm developed by Jaakkola *et al.* [34] is used to solve this POMDP. In this approach, different from conventional reinforcement learning algorithms, the agent does not need to observe all of the states, for the algorithm to converge. A POMDP value function and a Q-value function, in which m and a refer to the observable message of the state, s , and action, respectively, are given by:

$$V(m) = \sum_{s \in m} P(s|m)V(s) \quad (2.1)$$

$$Q(m, a) = \sum_{s \in m} P(s|m)Q(s, a). \quad (2.2)$$

A recursive Monte-Carlo strategy is used to compute the value function and the Q-value function. It is noted that the pilot model is given in terms of a policy, π , which is a mapping from observations or messages, m , to actions, a . During reinforcement learning, the policy update is calculated as below, where the policy is updated toward π^1 with ε learning rate, after each iteration:

$$\pi(a|m) \rightarrow (1 - \varepsilon)\pi(a|m) + \varepsilon\pi^1(a|m) \quad (2.3)$$

where, π^1 is chosen such that, $J^{\pi^1} = \max_a (Q^{\pi}(m, a) - V^{\pi}(m))$. For any policy $\pi^1(a|m)$, J^{π^1} is defined as:

$$J^{\pi^1} = \sum_a \pi^1(a|m)(Q^{\pi}(m, a) - V^{\pi}(m)). \quad (2.4)$$

It is noted that the pilot model, or the policy, is obtained once the policy converges during this iterative process.

2.1.3 Combining Game Theory With Reinforcement Learning

The method employed in this study carefully combines the two concepts explained in the above sections: game theory (GT) and reinforcement learning (RL). The method consists of two stages: 1) obtaining pilot reaction models with various levels (level-k) and 2) simulating a given scenario using these models. In the first stage, which can also be considered as the “training” stage, a level-1 type model is trained by assigning level-0 behavior to all of the agents but the one that is being trained. The trainee learns to react as best as he/she can in this environment using RL. Thus, the resulting behavior becomes a level-1 type. Similarly, a level-2 behavior is trained by assigning level-1 behavior to all of the agents but the trainee. This process continues until the highest desired level is reached. Once all of the desired levels are obtained, the first stage ends and in the second stage a given scenario is simulated by assigning certain proportions of these levels to the agents in the scenario. It is noted that in the scenario investigated in this paper, the pilots of the manned aircraft have level-0, level-1 and level-2 behavior types whereas the movements of the UAS are commanded via sense and avoid algorithms.

2.2 Components of the Hybrid Airspace Scenario

The investigated scenario consists of 180 manned aircraft with predefined desired trajectories and an UAS which moves based on its pre-programmed flight plan from one waypoint to another. Fig. 2.1 shows a snapshot of this scenario where the red squares correspond to manned aircraft and the cyan square corresponds

to the UAS. The size of the considered airspace is $600km \times 300km$. The airspace is gridded just to make it easier to visually grasp the dimensions (two neighboring grid points are $5nm$ away), nevertheless all aircraft, manned or unmanned, move in the airspace continuously. Yellow circles show the predetermined waypoints that the UAS is required to pass. The blue lines passing through the waypoints show the predetermined path of the UAS. It is noted that the UAS does not follow this path exactly since it needs to deviate from its original trajectory to avoid possible conflicts using an on-board SAA algorithm. The initial positions, speeds

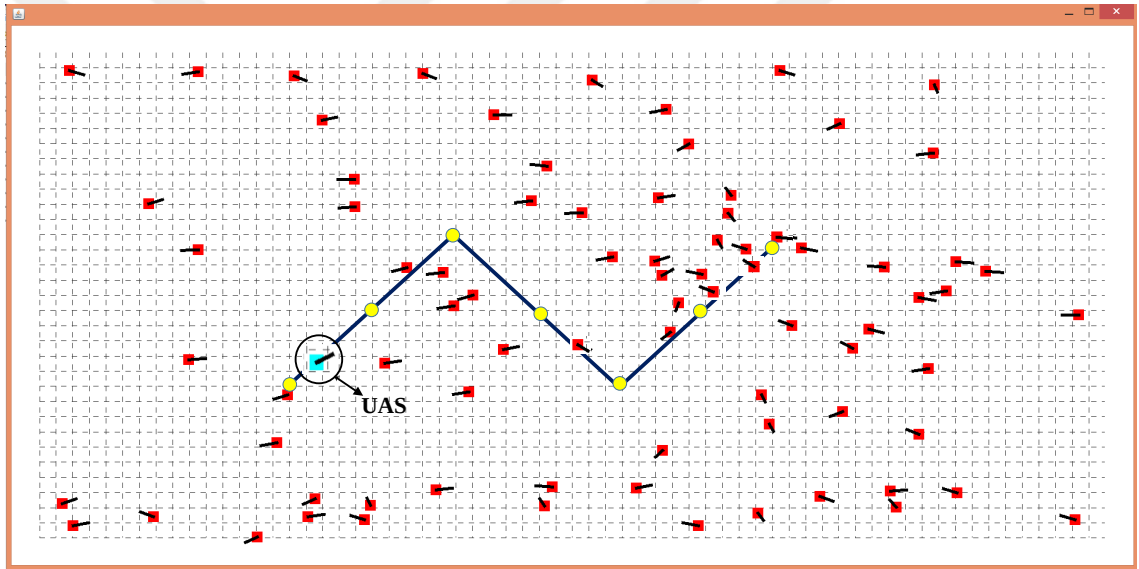


Figure 2.1: Snapshot of the hybrid airspace scenario in the simulation platform. Each square stands for a $5nm \times 5nm$ area.

and headings of the aircraft are obtained from Flightradar24 website which provides live air traffic data (<http://www.flightradar24.com>). The data is collected from air traffic volume on Colorado state, USA airspace on 11 March, 2015. It is noted that in the Next Generation Airspace System (NextGen) air travel demand is expected to increase dramatically, thus traffic density is expected to be much more than it is today. To represent this situation, the number of aircraft in the scenario is increased by projecting various aircraft at different altitudes to a given altitude. To handle the increase in aircraft volume in NextGen, it is expected that new technologies and automation will be introduced such as the automatic dependent surveillance-broadcast (ADS-B), which is a technology that enables an aircraft to receive other aircraft's identification, position and velocity

information and to send its information to others. In the investigated scenario, it is assumed that each aircraft is equipped with the ADS-B. It is noted that the ADS-B can also provide information about the flight path of an aircraft, which is highly relevant to collision avoidance. In our simulations, we provided this crucial information by answering the question “in a given time window, where in my observation space do I expect an intruding aircraft?”, for each agent (see Section 2.2.1 for details).

2.2.1 Pilot Observations and Memory

Although ADS-B provides the positions and the velocities of other aircraft, with his/her limited cognitive capabilities a pilot can not possibly process all this information during his/her decision making process. In this study, in order to model pilot limitations, including the limitations at visual acuity and perception depth, as well as the limited viewing range of an aircraft, it is assumed that the pilots can observe (or process) the information from a limited portion of the nearby airspace. This limited portion is simulated as equal angular portions of two co-centered circles called the “observation space” which is schematically depicted in Fig. 2.2. The radius of the inner circle represents pilot vision range, which is taken as $1nm$ based on a survey executed in [35]. The radius of the outer circle is a variable that depends on the separation requirements. Since standard separation for manned aviation is $3 - 5nm$ [9], this radius is taken as $5nm$. Whenever an intruder aircraft moves toward one of the 6 regions of the observation space (see Fig. 2.2), the pilot perceives that region as “full”. The pilot, in addition, can roughly distinguish the approach angle of the approaching intruder. A “full” region is categorized into four cases; with a) $0^\circ < \text{approach angle} < 90^\circ$, b) $90^\circ < \text{approach angle} < 180^\circ$, c) $180^\circ < \text{approach angle} < 270^\circ$ and d) $270^\circ < \text{approach angle} < 360^\circ$. Fig. 2.2 depicts a typical example, where pilot A observes that aircraft B is moving toward one of the 6 regions that is colored. In this particular example, pilot A perceives the colored region as “full” with approach angle in the interval $[90^\circ, 180^\circ]$ and the rest of the regions as “empty”. The information about emptiness, fullness of a region and approach angle is fed to the reinforcement

learning algorithm simply by assigning 0 to empty regions and 1, 2, 3, 4 to full regions, based on the approach angle classifications explained above. Pilots also know the best action that would move the aircraft closest to its trajectory (BTA: Best Trajectory Action) and the best action that would move the aircraft closest to its final destination (BDA: Best Destination Action). Moreover, pilots have a memory of what their actions were at the previous time step. Given an observation, the pilots can choose between three actions: 45° left, straight, or 45° right, which are coded with numbers 0, 1 and 2. Six ADS-B observations,

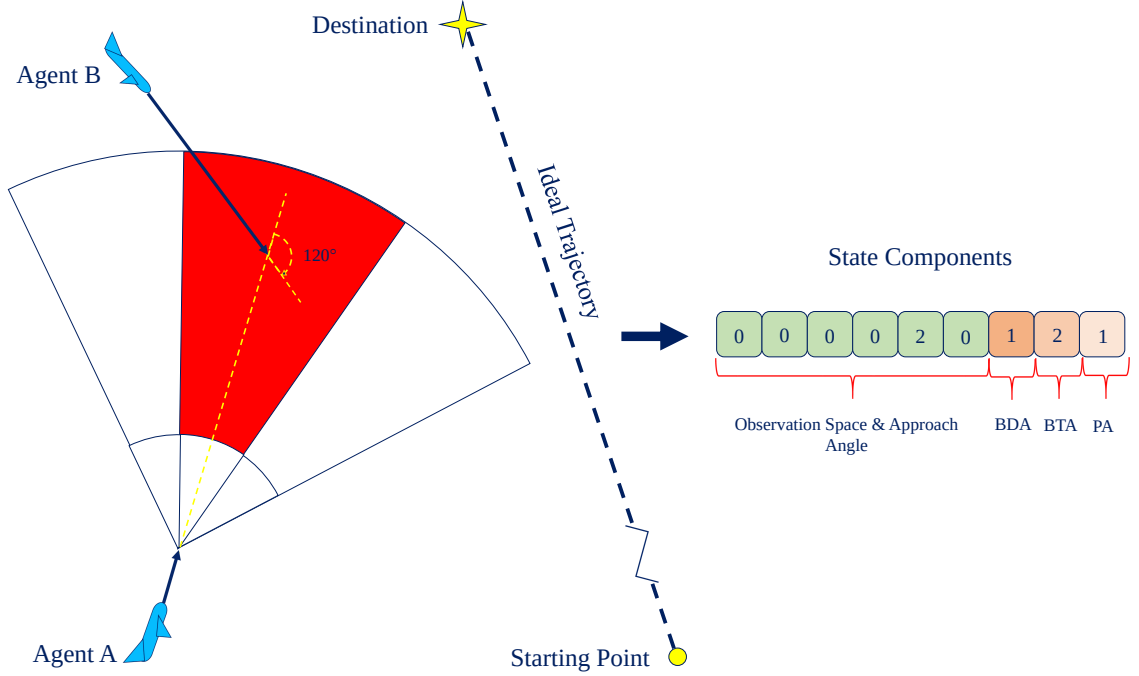


Figure 2.2: Pilot observation space.

one BTA, one BDA, and one previous move make up nine total inputs for the reinforcement learning algorithm. Observations get 5 values, 0, 1, 2, 3 or 4. The previous move, BTA and BDA have three dimensions each: 45° left, 45° right, or straight. Therefore, the number of states for which the reinforcement learning algorithm needs to assign appropriate actions is $5^6 \times 3^3 = 421,875$.

2.2.2 Pilot Objective Function

The goal of the reinforcement learning algorithm is to find the optimum probability distribution among possible action choices for each state. As explained above,

reinforcement learning achieves this goal by evaluating actions based on their return which is calculated via a reward/objective function. A reward function can be considered as a happiness function, goal function or utility function which represents, mathematically, the preferences of the pilot among different states. In this paper, the pilot reward function is defined as

$$reward = w1*(-C) + w2*(-S) + w3*(-CA) + w4*(D) + w5*(-P) + w6*(-E). \quad (2.5)$$

In (2.5), “ C ” is the number of aircraft within the collision region. Based on the definition provided by the Federal Aviation Administration (FAA), the radius of collision is taken as $500ft$ [36]. “ S ” is the number of air vehicles within the separation region. The radius of the separation region is $5nm$ [9]. “ CA ” represents whether the aircraft is getting closer to the intruder or going away from the intruder and takes values 1, for getting closer, or 0, for going away. “ D ” represents how much the aircraft gets closer to or goes away from its destination normalized by the maximum distance it can fly in a time step. The time step is determined based on the frequency of pilot decisions. The average time step during reinforcement learning is determined to be 20 seconds. “ P ” represents how much the aircraft gets closer to or goes away from its ideal trajectory normalized by the maximum distance it can fly in a time step and “ E ” represents whether or not the pilot makes an effort (move). “ E ” gets a value of 1 if the pilot makes a new move and 0 otherwise.

2.2.3 Manned Aircraft Model

The initial positions, speeds and heading angles of the manned aircraft are obtained from Flightradar24 website (<http://www.flightradar24.com>). It is assumed that all aircraft are in their en-route phase of travel with constant speed, $\|\vec{v}\|$, in the range of $[150 - 550]knots$. Aircraft are controlled by their pilots who may decide to change the heading angle for 45° , -45° or may decide to keep it unchanged. Once the pilot gives a heading command, the aircraft moves to the desired heading, ψ_d , in the constant speed mode. The heading change is modeled by a first order dynamics with the standard rate turn: a turn in which an aircraft changes its heading at a rate of 3° per second (360° in 2 minutes) [37]. This is modeled

as a first order dynamics with a time constant of $10s$ ($45 \times (1 - 1/e)/3 \approx 10$). Therefore, the aircraft heading dynamics can be given as

$$\dot{\psi} = -\frac{1}{10} \times (\psi - \psi_d) \quad (2.6)$$

and the velocity, $\vec{v} = (v_x, v_y)$, is then obtained as:

$$v_x = \|\vec{v}\| \sin \psi \quad (2.7)$$

$$v_y = \|\vec{v}\| \cos \psi. \quad (2.8)$$

2.2.4 UAS Model

The UAS is assumed to have the dynamics of a RQ-4 Global Hawk with operation speed of $340knots$ [38]. It moves according to its pre-programmed flight plan and is also equipped with a SAA system. The SAA system can initiate a maneuver to keep the UAS away from other traffic, if necessary, by commanding a velocity vector change. Otherwise, the UAS will continue moving based on its mission plan. Therefore, the UAS always receives a velocity command either to satisfy its mission plan or to protect its safety. Since the UAS has a finite settling time for velocity vector changes, the desired velocity, $\vec{V}_d$ cannot be reached instantaneously. Therefore, the velocity vector variation dynamics of the UAS is modeled by a first order dynamics with a time constant of $1s$ [39] which is represented as:

$$\dot{\vec{v}} = -(\vec{v} - \vec{v}_d). \quad (2.9)$$

2.2.5 Sense And Avoid Algorithms

In order to assure that the UAS can detect probable conflicts and can autonomously perform evasive maneuvers, it should be equipped with a SAA system. In this paper, two different SAA algorithms are investigated. These SAA algorithms are developed by Fasano *et al.* [40] referred as SAA1 and Mujumdar *et al.* [39] referred as SAA2. The algorithms consist of two phases; *conflict detection* phase and *conflict resolution* phase. In the detection phase, the SAA algorithms project the trajectories of the UAS and the intruder aircraft in time, using a predefined time interval, and if the minimum distance between the aircraft during this time is calculated to be less than a minimum required distance,

R , it is determined that there will be a conflict. The same conflict detection logic is used for both of the two SAA algorithms. In order to prevent the conflict, the UAS starts an evasive maneuver in the conflict resolution phase, which is handled differently for SAA1 and SAA2. In SAA1 resolution phase algorithm, a velocity adjustment is suggested that guarantees minimum deviation from the trajectory. The velocity adjustment command, $\vec{v}_A^d$, for the UAS is given in the equation below

$$\vec{v}_A^d = \left[\frac{v_{AB} \cos(\eta - \zeta)}{\sin(\zeta)} \left[\sin(\eta) \frac{\vec{v}_{AB}}{v_{AB}} - \sin(\eta - \zeta) \frac{\vec{r}}{\|\vec{r}\|} \right] \right] + \vec{v}_B \quad (2.10)$$

where, $\vec{v}_A$ and $\vec{v}_B$ refer to the velocity vectors of the UAS and the intruder. $\vec{r}$ and $\vec{v}_{AB}$ denote the relative position and velocity between the UAS and the intruder, respectively. ζ is the angle between $\vec{r}$ and $\vec{v}_{AB}$ and η is calculated as $\eta = \sin^{-1} \frac{R}{\|\vec{r}\|}$. In the case of multiple conflict detection, the UAS will start an evasive maneuver to resolve the conflict that is predicted to happen earliest. In the SAA2 algorithm, the velocity adjustment vector is determined as given in the below equation in which $\vec{r}_m$ stands for the minimum relative position vector between the UAS and the intruder during the conflict:

$$\vec{v}_A^d = \frac{-\vec{v}_A \left(\frac{\vec{r}_0 \cdot \vec{v}_{AB}}{\|\vec{v}_{AB}\|} \right) - (R - \|\vec{r}_m\|) \frac{\vec{r}_m}{\|\vec{r}_m\|}}{\left\| -\vec{v}_A \left(\frac{\vec{r}_0 \cdot \vec{v}_{AB}}{\|\vec{v}_{AB}\|} \right) - (R - \|\vec{r}_m\|) \frac{\vec{r}_m}{\|\vec{r}_m\|} \right\|} \quad (2.11)$$

where, $\vec{r}_0$ refers to the initial relative position vector between the UAS and the intruder. In this solution strategy, UAS moves to resolve the conflict until it retains the minimum safe distance with the intruder. Similar to the SAA1 algorithm, in the case of multiple intruders, the UAS will start an evasive maneuver to resolve the conflict that is predicted to happen earliest.

2.2.6 Model Validation

As noted earlier, since the routine access of UAS into NAS is not a reality yet and thus there is not enough experience accumulated about the issue, it is extremely hard to predict the effects of the technologies and concepts that are developed for the integration. Therefore, employing simulation is currently the only way to understand the effects of UAS integration on the air traffic system [5]. However,

regardless of whether the modeled system exists or whether it is expected to be a reality in the future, the representative model should be validated [41].

Below, we break down the validation task into two steps. In the first step, we explain that the underlying hierarchical game theoretical modeling approach is a useful and valid approach to model complex human interactions, based on earlier experimental studies. In the second step, we investigate the validity of the proposed approach for the UAS integration implementations. Since UAS integration data is not available yet, we take a different approach in this step: We first provide a validation methodology that can be used to validate the proposed approach when the data for UAS integration becomes available. We also explain that the proposed model has enough degrees of freedom that can be used to obtain a predictive model using this data. Then, we proceed to show that the trajectories created by the proposed model is similar to that of a validated encounter model for manned aircraft created using real radar data. Finally, we explain a validation method, called “face validation”, which is commonly used when the modeled system is expected to be a reality in the future, and we argue that our simulation results in the next section can be used to apply this method.

2.2.6.1 Validation of the game theoretical modeling approach

In this study, we utilized a well known game theoretical modeling approach called “level-k” reasoning. The advantage of this approach is its computational simplicity, where the intelligent agent makes behavioral assumptions about others and then produces the best response accordingly based on a reward function. Because of this simplicity, in multi-move scenarios, such as the ones treated in this research, level-k reasoning provides computationally tractable solutions. This approach not only provides a computationally efficient solution but also is shown to be able to model complex human interactions in experimental settings: In [17], several experimental results are conducted using various sizes of subject pools that are made to play different games. Using the data from these games, models of strategic thinking are evaluated and compared and although some perform better than others depending on the game type, “level-k” approach is found to be “behaviorally more plausible”. It is noted that these experimental results are

cited here not to prove that the proposed approach is better than other game theoretical approaches, but to provide real world data showing that the “level- k” modeling approach can represent real world behavior in complex decision making scenarios. In the scenarios studied in this work, the intelligent agents (pilots) are also strategic decision makers as explained in earlier sections and they need to make decisions in a complex environment to maximize their rewards. Therefore, the underlying game theoretical approach can also be considered as a good fit to the problem studied in this work.

2.2.6.2 Validation of the proposed modeling approach for UAS integration concepts

As presented in the previous section, the game theoretical modeling approach is shown to model real world behavior in earlier experimental studies. In this section, we address the question “can this approach be reliably used to model UAS integration scenarios?”.

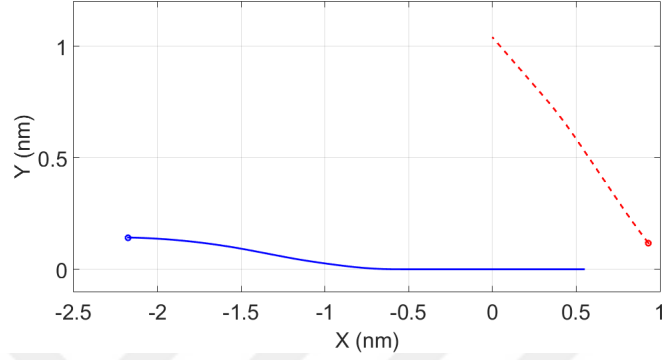
Validation methodology: There exist several model validation methods such as “face validity” , “historical data validation”, “parameter sensitivity analysis” and “predictive validation” [42]. Among these methods, the most definitive technique is predictive validation, where the model is used to predict the system’s behavior and then the outputs of the model and the real system are compared. Here, we explain two main aspects of predictive validation [7] that can be used to validate the proposed model in this study, when the UAS integration data becomes available: First, relevant statistics between the model and the real data should have reasonable agreement. For example, for UAS integration, the average deviation of the UAS from their intended trajectory against the type of the SAA algorithm should be similar between the model and the data. Similarly, the average number of separation violations between manned and unmanned aircraft for different kinds of SAA algorithms should match. Secondly, individual encounters should show similar characteristics. For example, the minimum separation distance between UAS and manned aircraft, and pilot decisions during encounters with similar geometry (approach angle, heading etc.) should be able to be predicted with reasonable accuracy by the model.

Comparison with a validated model: Since UAS integration data is not

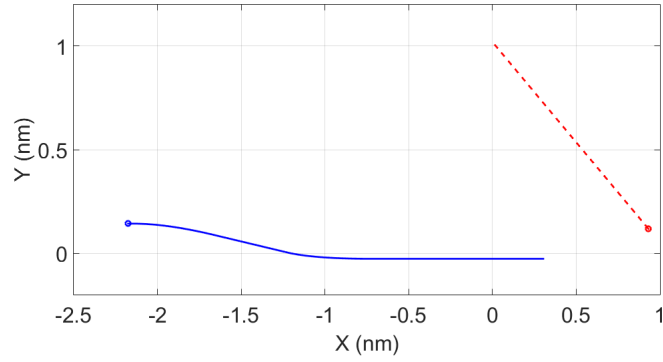
available yet, we compared the results of the proposed model with a manned-aircraft-only encounter model created and validated by the Lincoln Laboratory using real radar data [7]. Sample trajectories are given in two text files that are open to the public: `cor_ac1.txt` and `cor_ac2.txt`. Among the encounters provided, 5 of them (listed as 3rd, 16th, 23rd, 34th and 45th encounters) do not employ altitude or speed variations for conflict resolution and thus can be used for our purposes. The objective of these comparisons are to show that the actions taken by the pilots in the proposed game theoretical model and the validated model are similar. In addition, minimum separation distances experienced between the aircraft and the times that the minimum separation occurs are shown to be reasonably close to each other for the compared models such that the status of separation violation remains the same.

Fig. 2.3a demonstrates aircraft trajectories during encounter number 3, listed in `cor_ac1.txt`. In Fig. 2.3b, the same encounter is regenerated using the proposed game theoretical model by assigning the same initial positions, initial heading angles and initial speeds to two aircraft. It is seen that although the trajectories are not exactly the same, pilot decisions determined by the game theoretical model are similar to the decisions provided by the validated model. In addition, according to Fig. 2.3c, the minimum separation distance is experienced after about 40s and 42s for the validated model and the game theoretical model, respectively, and the difference between the minimum distances is about 0.05nm. In this example, the pilots represented by the solid blue and dashed red curves are modeled as level-1 and level-0 pilots, respectively. Figs. 2.4-2.7 also show similar characteristics where the encounter trajectories and pilot decisions are similar for the validated and proposed models.

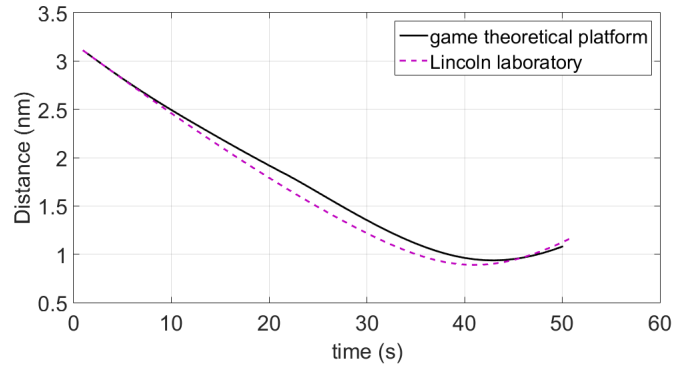
Face validation: “Face validation” is a validation method used for models that are developed for systems that are expected to be a reality in the future, such as UAS integration models [42]. In this method, two aspects of the model are evaluated: 1) Is the logic in the conceptual model correct? and 2) Are the input-output relationships of the model reasonable? The core ideas of the proposed framework, such as level-k game theoretical concept, reinforcement learning and bounded rationality, are supported by several references earlier in this study. In addition, the logic of the objective function utilized during the modeling process



(a) Trajectories created by the validated model.

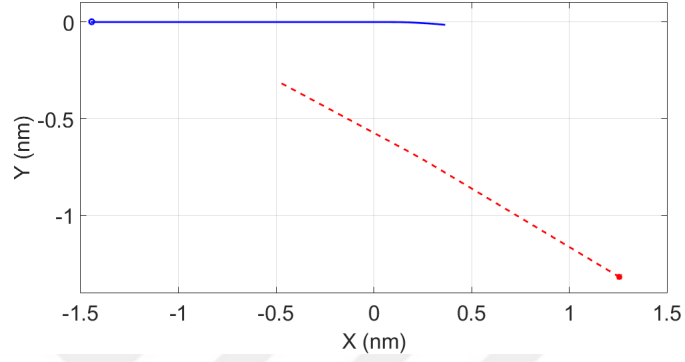


(b) Trajectories created by the proposed model.

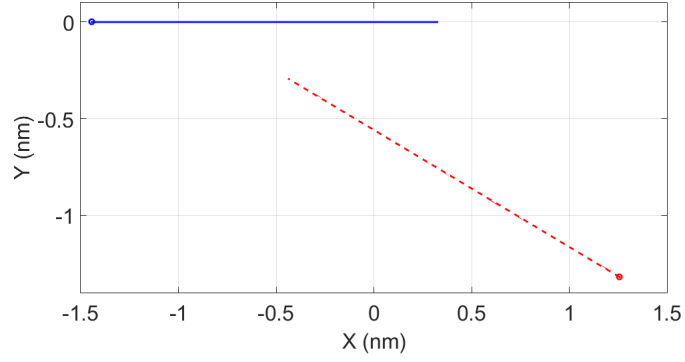


(c) Separation distances for each model.

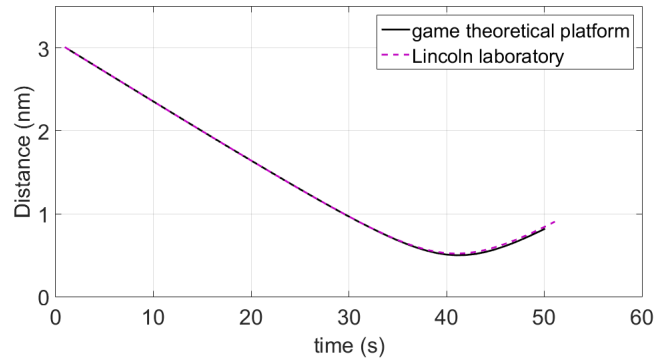
Figure 2.3: Comparison of the trajectories created by the validated model and the game theoretical modeling approach for sample encounter number 3.



(a) Trajectories created by the validated model.

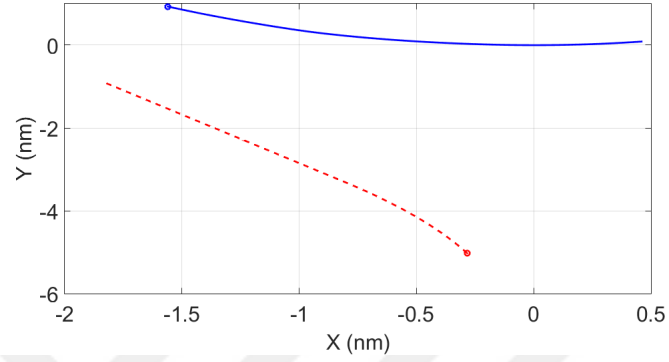


(b) Trajectories created by the proposed model.

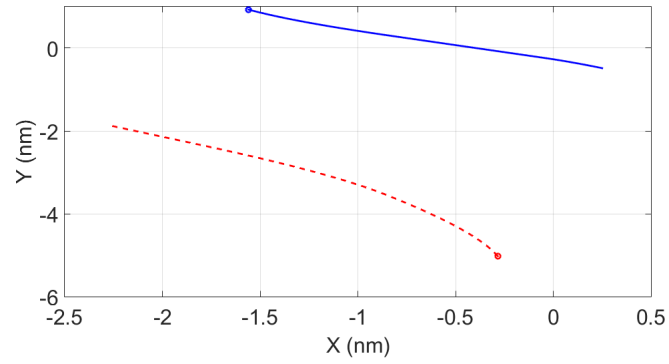


(c) Separation distances for each model.

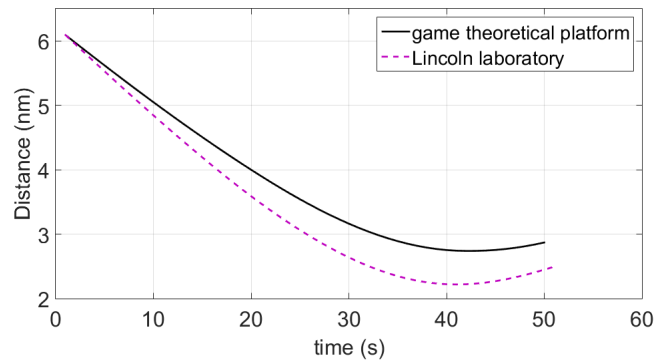
Figure 2.4: Comparison of the trajectories created by the validated model and the game theoretical modeling approach for sample encounter number 16.



(a) Trajectories created by the validated model.

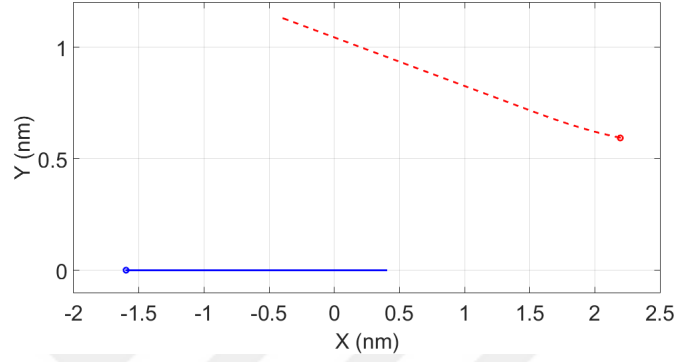


(b) Trajectories created by the proposed model.

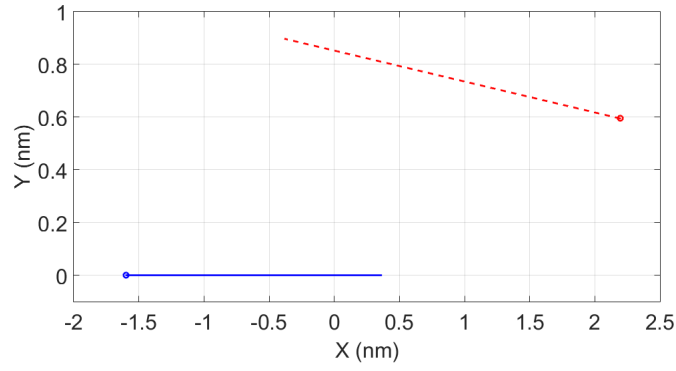


(c) Separation distances for each model.

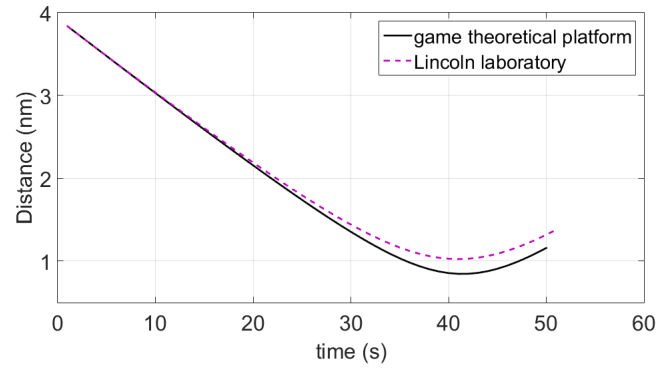
Figure 2.5: Comparison of the trajectories created by the validated model and the game theoretical modeling approach for sample encounter number 23.



(a) Trajectories created by the validated model.

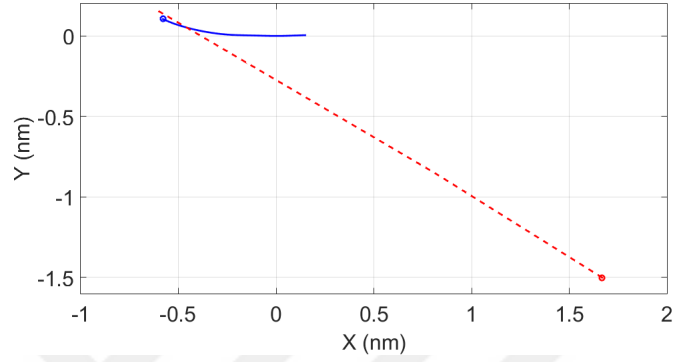


(b) Trajectories created by the proposed model.

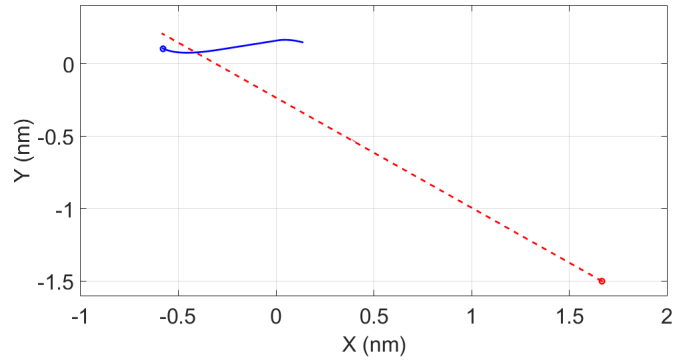


(c) Separation distances for each model.

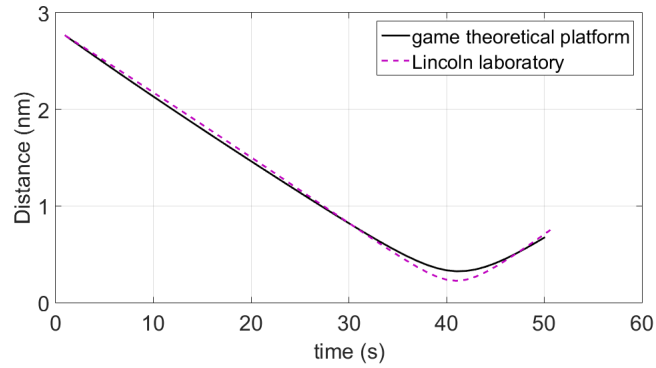
Figure 2.6: Comparison of the trajectories created by the validated model and the game theoretical modeling approach for sample encounter number 34.



(a) Trajectories created by the validated model.



(b) Trajectories created by the proposed model.



(c) Separation distances for each model.

Figure 2.7: Comparison of the trajectories created by the validated model and the game theoretical modeling approach for sample encounter number 45.

is detailed where it is seen that the choice of the terms is logical. Finally, in the simulation results section of this work, input-output relationships of the model are discussed in length to show that they represent reasonable system behavior. Therefore, the steps needed for the face validation of the proposed method are completed.

Remark: It is noted that without collecting, processing and analyzing real HAS data, which may be available in the near future, and carefully comparing the outputs with the model, using available statistical validation tools (see [41]), the validation of the model can not be accepted as completed. It is important for a model to have enough degrees of freedom so that when discrepancies with the real data are detected the model can be modified accordingly to obtain a match with the data with reasonable accuracy [41]. In this regard, the proposed game theoretical framework is a strong candidate for a successful UAS integration model since it contains several degrees of freedom such as the objective function weights representing the importance of each term. In addition, the modular structure of the objective function allows the designers to add/subtract terms to achieve an agreement with the data.

2.3 Simulation Results and Discussion

In this section, the results of a quantitative analysis of a simulation for UAS integration scenarios are presented. Before showing the results for the scenario explained in the previous section, single encounter scenarios, where a single UAS and a single manned aircraft are in a collision path, are investigated. Later, the results for the scenario with multiple encounters in a crowded airspace are shown. A quantitative comparison between the two SAA algorithms in terms of their performance and safety is also presented.

2.3.1 Hybrid Airspace Scenarios with a Single Encounter

In order to investigate the reactions of a level-k pilot during a conflict with an UAS, 4 single encounter scenarios are designed. In these 4 scenarios, level-1 and level-2 policies are used for the manned aircraft pilots and the UAS follows the guidelines of the SAA1 algorithm which may command velocity adjustments in

order for the UAS to avoid the conflict. Apart from pilot levels, the effect of different approach angles, which take the values of 45° , 90° , 135° and 180° , are also investigated. Figure 2.8 depicts the snapshots of four cases, where the red square corresponds to the manned aircraft and cyan triangle corresponds to the UAS. The gray track line right behind the manned aircraft and the UAS represent their traveled path from their initial positions to where they stand in the snapshot. Circles show the initial positions and destinations. The geometric size of the scenarios is $100km \times 50km$. In all cases, the manned aircraft and the UAS are heading toward a conflict which is detected by both the UAS, via its SAA system, and the pilot 20s prior to a probable miss separation. A miss separation is declared when the relative distance becomes less than $5nm$. The pilot then starts an evasive maneuver based on the level-1 and the level-2 reasoning policies, and the UAS implements its own evasive maneuver based on the SAA1 system, whose working principles are explained in Section 2.2.5. Figure 2.9 depicts the separation distance and trajectory deviations during these single encounter scenarios. Comparing the performances of the level-1 and the level-2 pilots, it can be seen that the level-1 pilot maneuvers in a way that he/she provides more separation distance than a level-2 pilot does, except for the case with 90° approach angle. This, in general, is expected since the level-1 pilot assumes that the intruder is a level-0 Decision Maker (DM) who will continue his/her given path without changing his/her direction, and therefore the level-1 DM takes the responsibility of the conflict resolution himself/herself. On the other hand, the level-2 pilot considers the intruder as a level-1 DM who will make maneuvers to avoid the conflict and therefore the conflict resolution responsibility will be shared. That is why the level-2 pilot, in comparison with the level-1 pilot, avoids the UAS with less separation distance. It is noted, however, that the level-1 pilot deviates from its ideal trajectory significantly more than the level-2 pilot does. Even in the case of 90° approach angle, where the minimum separation distance between the level-1 pilot and the UAS is slightly less than the case with the level-2 pilot, the trajectory deviation of the level-2 pilot is less than that of the level-1 pilot. These analyses show that the type of the pilot reactions during a conflict scenario makes a significant impact on the results when evaluating the performances of the SAA algorithms. The same conclusion are derived when the UAS maneuvers based on

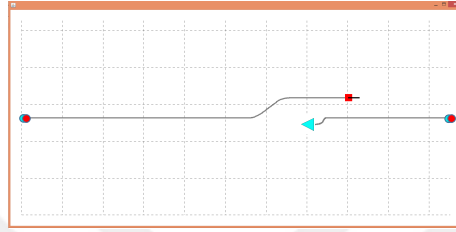
the SAA2 logic, however these results are omitted to save space.

2.3.2 Hybrid Airspace Scenario with Multiple Encounters

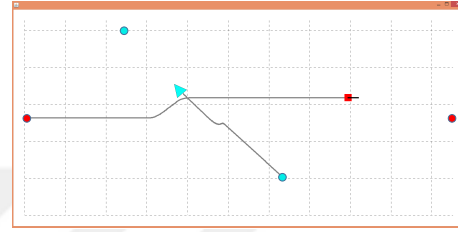
The details of this scenario was explained in Section 2.2. In this section, the scenario is simulated to investigate a) the effect of the variations in the objective function parameters, b) the effect of the distance and the time horizons and c) the effect of responsibility assignment for conflict resolution, on safety and performance. Since the loss of separation is the most serious issue, the safety metric is taken as the number of separation violations between the UAS and the manned aircraft. Performance metrics, on the other hand, include a) averaged manned aircraft trajectory deviations, b) UAS trajectory deviation and c) total flight time of the UAS. In all of the simulations, level-0, level-1 and level-2 pilot policies are randomly distributed over the manned aircraft in such a way that 10% of the pilots fly based on level-0 policies, 60% of the pilots act based on level-1 policies and 30% use level-2 policies. This distribution is based on experimental results discussed in [17]. It is noted that although the given distribution is obtained from human experimental studies, they did not necessarily include pilots and therefore may not be fully representative but can easily be adapted to other distributional data for this framework.

2.3.2.1 Sensitivity analyses of the weighting parameters in the objective function

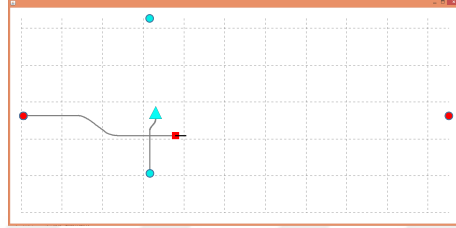
In this section, the sensitivity of the pilot model to its parameters, which are the weight vector components of the objective function in equation (2.5), is investigated. Specifically, the effect of the ratio of the sum of the weights of the safety components of the objective function over the sum of the weights of the performance components, $r = \frac{w1+w2+w3}{w4+w5+w6}$, is investigated for various traffic densities. The results of this analysis for various traffic densities in the HAS is depicted in Fig. 2.10. It is seen that as r increases, the trajectory deviations of both the manned aircraft and the UAS increases, regardless of the traffic density. Cooperation of the manned aircraft and the UAS to resolve the conflict reduces the number of separation violations up to a certain value of r . However, the number



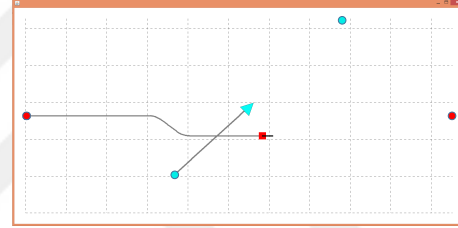
(a) level-1 pilot, approach angle 45°



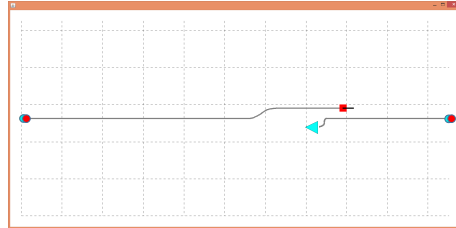
(b) level-1 pilot, approach angle 90°



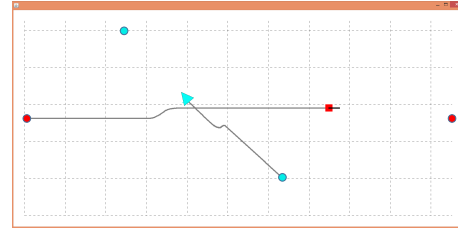
(c) level-1 pilot, approach angle 135°



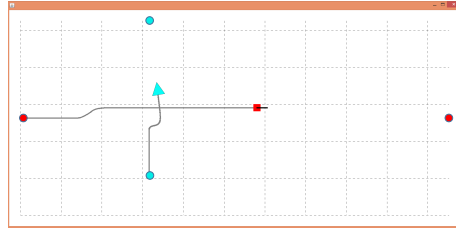
(d) level-1 pilot, approach angle 180°



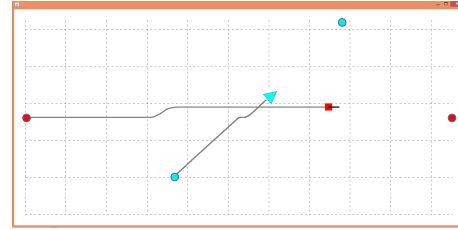
(e) level-2 pilot, approach angle 45°



(f) level-2 pilot, approach angle 90°

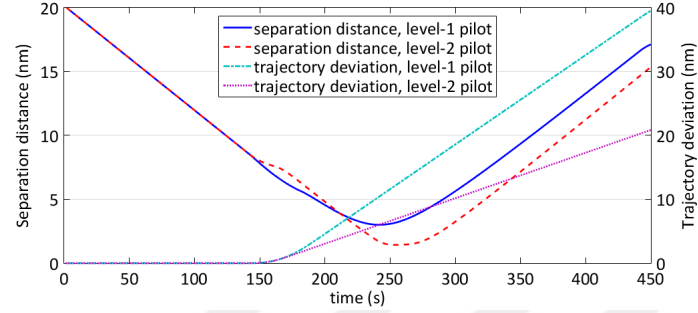


(g) level-2 pilot, approach angle 135°

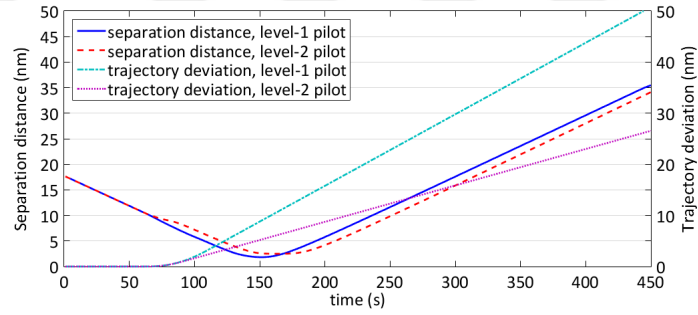


(h) level-2 pilot, approach angle 180°

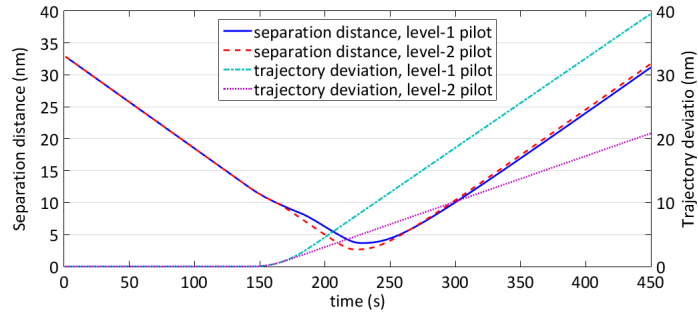
Figure 2.8: Level-1 and level-2 pilots interacting with UAS.



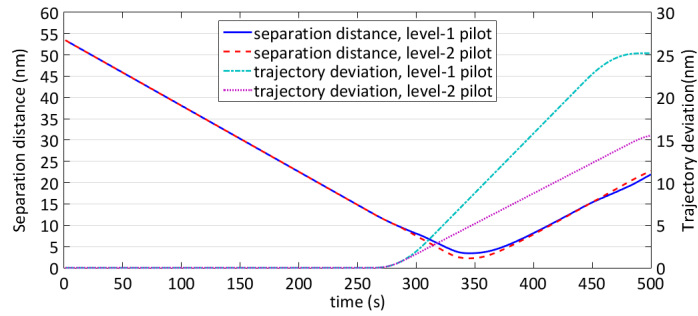
(a) approach angle 45°



(b) approach angle 90°



(c) approach angle 135°



(d) approach angle 180°

Figure 2.9: Safety vs. performance of Level-k pilot interacting with UAS

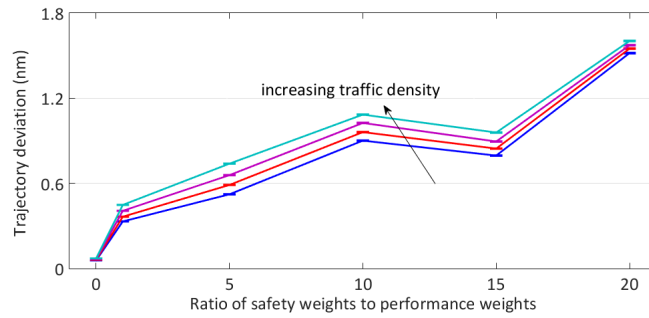
of violations starts increasing with further increase in r . What this means is that, as pilots become more sensitive about their safety and start to overreact to probable conflicts with extreme deviations from their trajectories, the traffic is effected in a negative way.

Fig. 2.11 presents the effect of increasing the ratio r in a single encounter scenario where surrounding traffic does not exist. The percentage values provided in the figure is obtained for 5000 encounters. For each r value these 5000 encounter “episodes” are repeated 1000 times to obtain reliable statistics. As expected, in the absence of surrounding traffic, the increase in the ratio r decreases the number of separation violations.

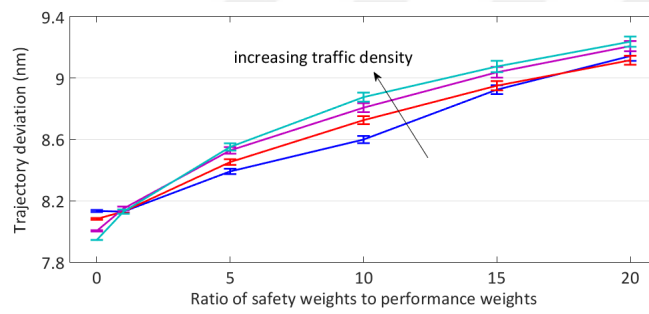
Remark: In this study, it is emphasized that humans are not expected to behave optimally in complex situations in the presence of multiple decision makers and automation due to 1) limited observation space, 2) limited processing power and 3) limited information about other decision makers. The exploited modeling framework captures this suboptimal pilot response using several tools explained in earlier sections, including the RL algorithm which provides convergence guarantees to a local maximum. The pilot behavior we observe in Fig. 2.10 is an example of suboptimal behavior where, in the presence of surrounding traffic, increased safety parameter weights, after a certain point, can cause extreme trajectory deviations and increased separation violations. Fig. 2.11, on the other hand, presents the expected behavior of decreasing violations with increased safety weights, when the scenario is much simpler.

2.3.2.2 The effect of distance and time horizons on performance and safety

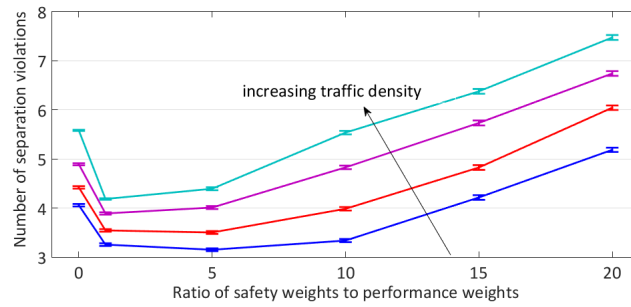
Although standard separation distance for manned aviation is $3 - 5nm$ [9], UAS might require wider separation requirements compared to manned aircraft. In the following analysis, horizontal separation requirement for the UAS is called the *distance horizon*, and the effect of it is reflected into the simulation by defining this value as the “scan radius” for the SAA algorithm: The SAA algorithm considers an intruding aircraft as a possible threat only if the aircraft is within the scan radius. Another variable whose effect is investigated is defined as the time to separation violation and is called the *time horizon*. In the simulation, the time



(a) manned aircraft trajectory deviation



(b) UAS trajectory deviation



(c) number of separation violations

Figure 2.10: Pilot model sensitivity analysis.

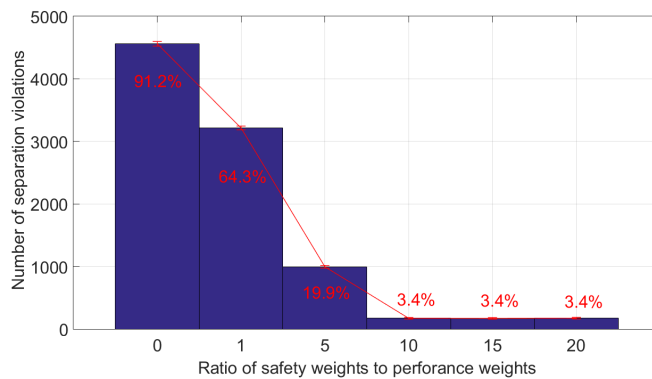


Figure 2.11: Pilot model sensitivity analysis for the single encounter scenario.

horizon is used as the time interval, within which the UAS predicts a probable conflict. Figure 2.12 and Fig. 2.13 show the effects of the time horizon and the distance horizon on the safety and performance of the system, when SAA1 and SAA2 algorithms are used, respectively. When SAA1 algorithm is employed, it is seen in Fig. 2.12 that increasing the distance horizon of the UAS makes the SAA1 system detect probable conflicts from a larger distance, which in turn increases the UAS trajectory deviation. Increasing the time horizon makes a similar effect on trajectory deviation. High UAS trajectory deviation results in higher flight times for the UAS to complete its mission. In addition, higher distance and time horizons reduce the trajectory deviations of the manned aircraft since conflicts are resolved mostly by the UAS. When the UAS foresees the probable conflicts earlier, with increased time and distance horizons, the number of separation violations generally decreases. Increasing the distance and time horizons after a certain point does not improve the safety (number of separation violations), since the UAS starts to disturb the traffic unnecessarily due to the overreactions of the SAA system.

The first observation that strikes the eye, in the case of SAA2 system utilization (see Fig. 2.13), is that the time horizon variations do not affect the results as much as the case of SAA1 system utilization. The second important difference of the SAA2 algorithm is that increasing the distance horizon consistently improves the safety (Fig. 2.13d), unlike the case of SAA1 algorithm utilization where larger distance horizon values do not make a major effect on safety. The reason for this difference can be explained by comparing Figs. 2.12b and 2.13b: SAA1 causes the UAS deviate from its trajectory significantly more than the SAA2 and thus separation violation numbers for the SAA1 do not improve further after a point due to a significant impact on the surrounding traffic. However, it should be noted that, in general, the violation numbers of the SAA1 is lower than that of SAA2 (see Figs. 2.12d-2.13d). After this quantitative analysis, it can be said that the SAA1 system results in a safer flight (less number of violations) whereas the SAA2 system provides a higher performance flight (lower deviations from the trajectory).

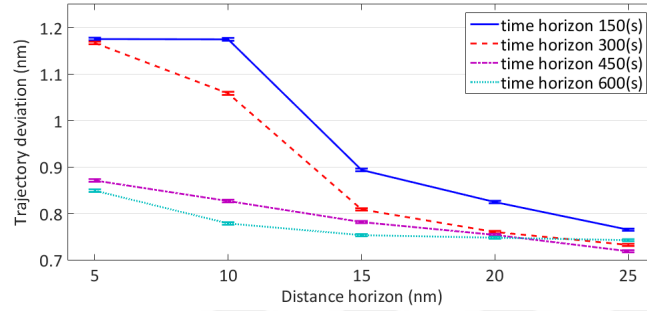
It is important to note that when the technologies and procedures mature

enough to enable full integration of UAS into the NAS, it would not be unrealistic to expect that the ratio of unmanned to manned aircraft will increase dramatically. Since HAS is a complex system where several intelligent agents move simultaneously, it is impossible to predict the effects of increased UAS presence. Therefore, it is important and useful to investigate the response of the overall system to increased number of UAS. Fig. 2.14 shows the effect of increasing the number of UAS in HAS. It is seen that, as the number of UAS increases, trajectory deviations, flight times and separation violations increase. It is noted that no mode/phase changes are observed in the system.

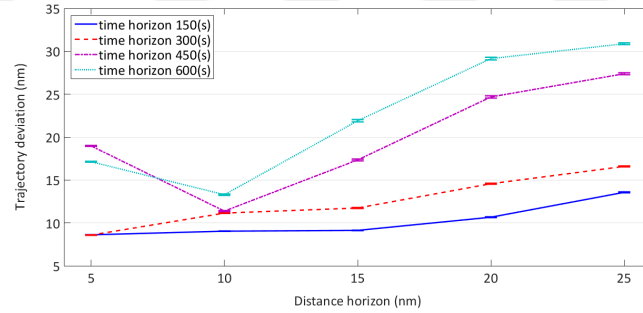
2.3.2.3 Separation responsibility analysis

Another issue to be addressed that is important in studying the integration of UAS into NAS is the separation responsibility [4]: it is crucial to determine which of the agents (manned aircraft or UAS) will take the responsibility of the conflict resolution. Fig. 2.15 depicts a comparison of different resolution responsibility cases: manned aircraft are responsible (blue), both manned aircraft and UAS are responsible (cyan) and only the UAS is responsible (red). In the case when only manned aircraft are responsible for conflict resolution, UAS is forced to continue its path without executing the SAA system and manned aircraft act as level-1 and level-2 DMs. In the case when the UAS is responsible for the conflict resolution, the manned aircraft are forced to continue their path without changing their heading and the UAS executes its SAA system. In the case when both the manned aircraft and the UAS are responsible for the conflict resolution, they both execute their evasive maneuvers. Figure 3.18 shows that manned aircraft deviate more from their trajectory when both the UAS and the manned aircraft share resolution responsibility, compared to the case when only the manned aircraft are responsible. This is true for both the SAA1 (the results on the left) and the SAA2 (the results on the right) algorithms. The reason for increased trajectory deviation for the manned aircraft in the case of shared responsibility is that the pilots' assumptions about possible UAS actions are not always correct which forces the pilots to make additional adjustments in their trajectory, which in turn increases manned aircraft trajectory deviations.

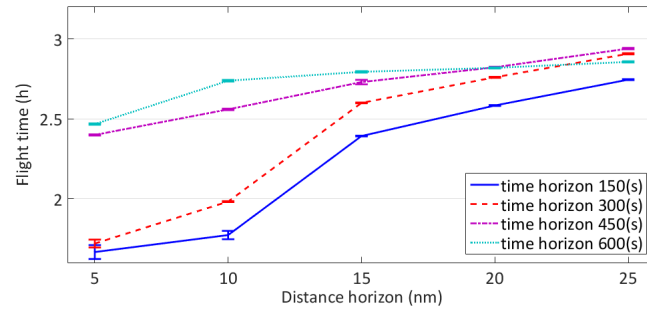
On the other hand, Fig. 3.20 shows that the UAS deviates from its trajectory



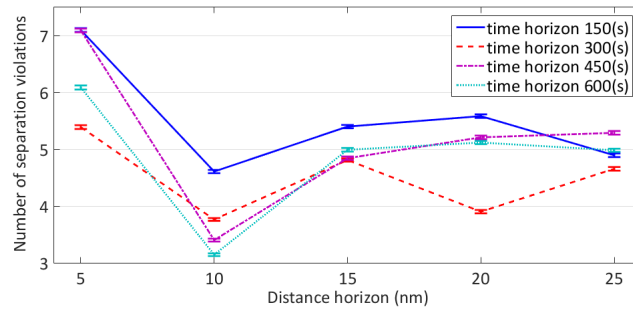
(a) manned aircraft trajectory deviation



(b) UAS trajectory deviation

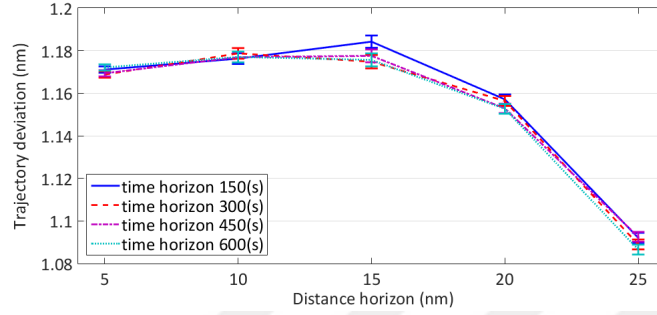


(c) UAS flight time

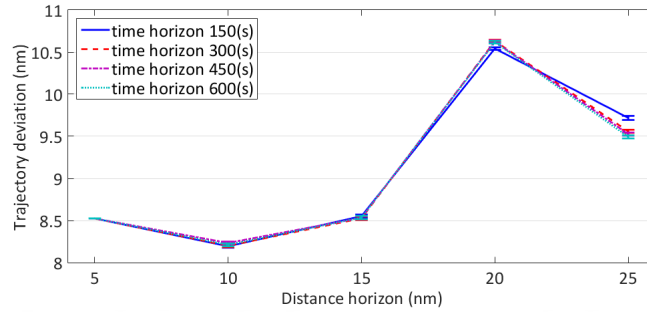


(d) number of separation violations

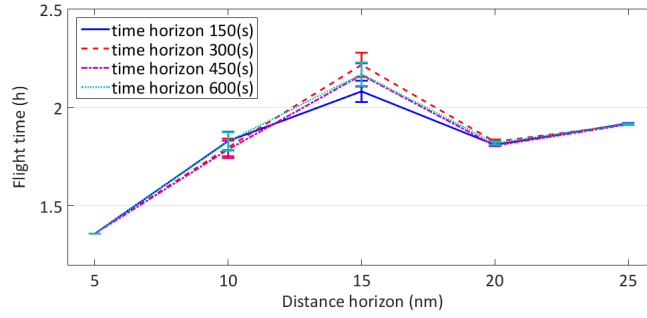
Figure 2.12: Safety vs. performance in HAS, when SAA1 is employed.



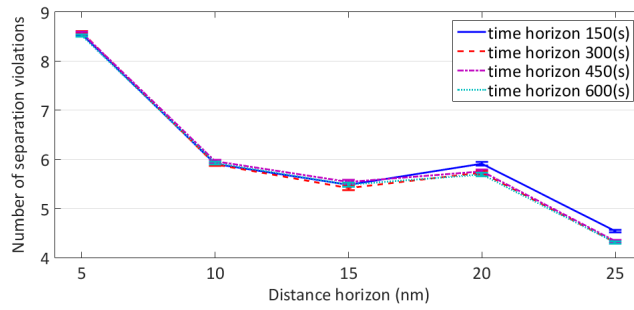
(a) manned aircraft trajectory deviation



(b) UAS trajectory deviation

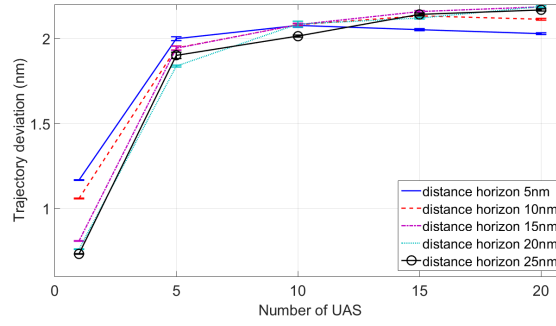


(c) UAS flight time

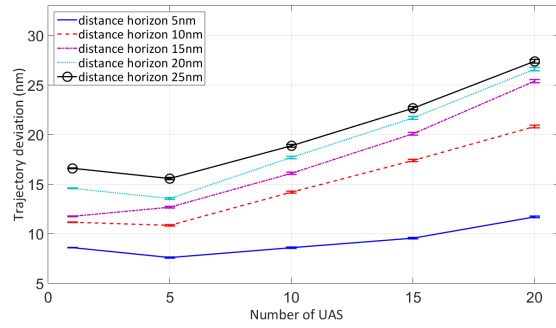


(d) number of separation violations

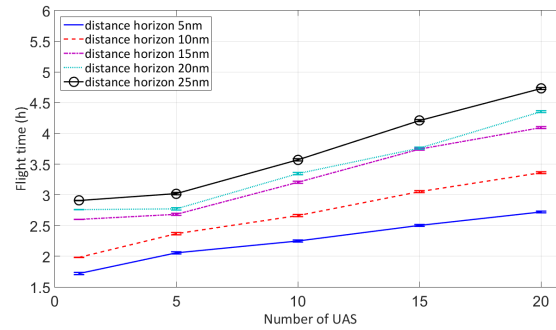
Figure 2.13: Safety vs. performance in HAS, when SAA2 is employed.



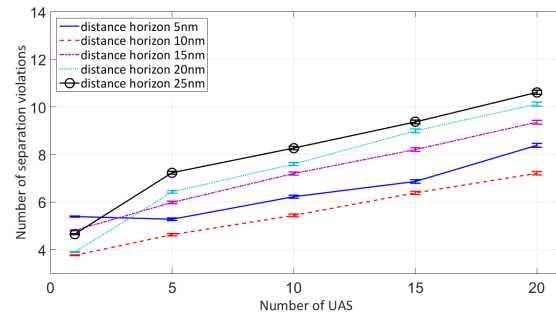
(a) manned aircraft trajectory deviation



(b) UAS trajectory deviation



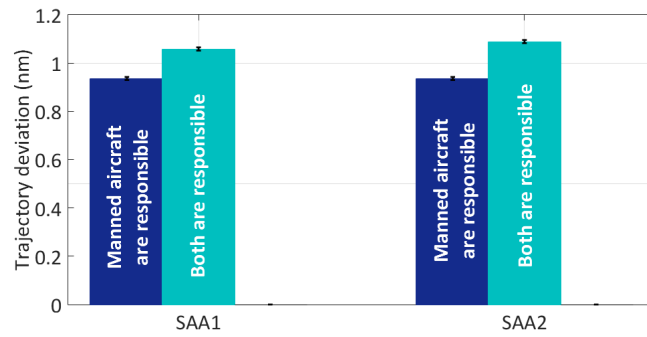
(c) UAS flight time



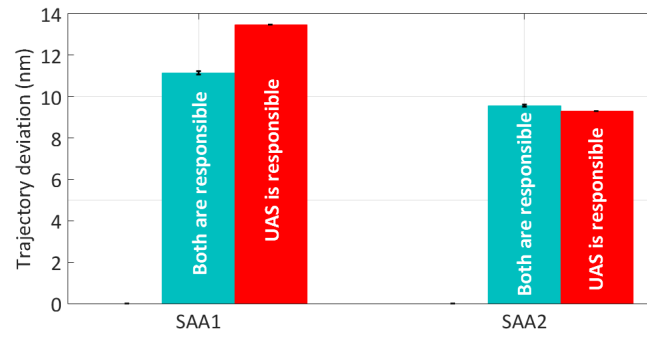
(d) number of separation violations

Figure 2.14: Safety vs. performance in HAS, when SAA1 is employed, in a multi-UAS scenario.

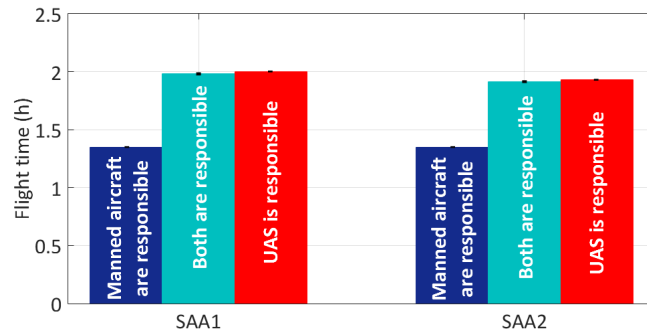
more when it is responsible for the resolution, compared to the case when the responsibility is shared, when SAA1 is utilized. For the case of SAA2 utilization, deviations are less and do not change much based on the responsibility assignments. Figure 3.19 shows, as expected, that for both SAA1 and SAA2, the UAS flight times are the shortest when only the manned aircraft become responsible for the resolution. Perhaps the most important result is given in Fig. 3.21, where it is shown that for both SAA1 and SAA2 utilizations, the safest case is when the resolution responsibility is given to the UAS.



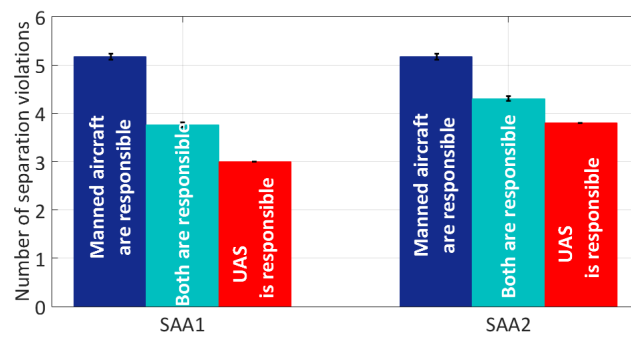
(a) manned aircraft trajectory deviation



(b) UAS trajectory deviation



(c) UAS flight time



(d) number of separation violations

Figure 2.15: SAA1, SAA2 and safety vs. performance in HAS.

Chapter 3

3D Framework

3.1 UAS Integration Scenario

In order to evaluate the outcomes of integrating Unmanned Aircraft Systems (UAS) in to the National Air Space (NAS), a Hybrid Air Space (HAS) scenario, where manned and unmanned aircraft co-exist is designed and explained in this section. The scenario consists of 188 manned aircraft and 3 UAS. The size of the airspace is $600km \times 300km \times 45000ft$. The initial positions, velocities, headings and altitudes of the aircraft are obtained from Flightradar24 website which provides live air traffic data (<http://www.flightradar24.com>). The data is collected from air traffic volume on Colorado state, USA airspace on 11 March, 2015. The manned aircraft in the scenario execute maneuvers based on the pilot model obtained using a combination of reinforcement learning and level-k reasoning, the details of which is explained in the following Section 3.2. Multiple UAS can be randomly located in the airspace and move based on their pre-programmed flight plan from one waypoint to another. Figure 3.1 shows a snapshot of the scenario with multiple manned aircraft and three UAS moving through their multiple waypoints. All aircraft whether manned or unmanned are flying at different altitudes and this snapshot depicts a 2D projection of their configuration, on the horizontal plane. The red squares correspond to manned aircraft and the cyan squares correspond to UAS, which are flying at different altitudes. All aircraft, manned or unmanned have continuous dynamics, which are provided in the following sections. Yellow circles show the predetermined waypoints that the UAS with the

highest altitude is required to pass. The waypoints of other two UAS are not shown in this snapshot. The dashed blue lines passing through the waypoints show the predetermined path of the UAS. It is noted that the UAS do not follow this path exactly since they need to deviate from their original trajectory to avoid possible conflicts using an on-board Sense and Avoid (SAA) algorithm, which is obtained from [40] and [39].

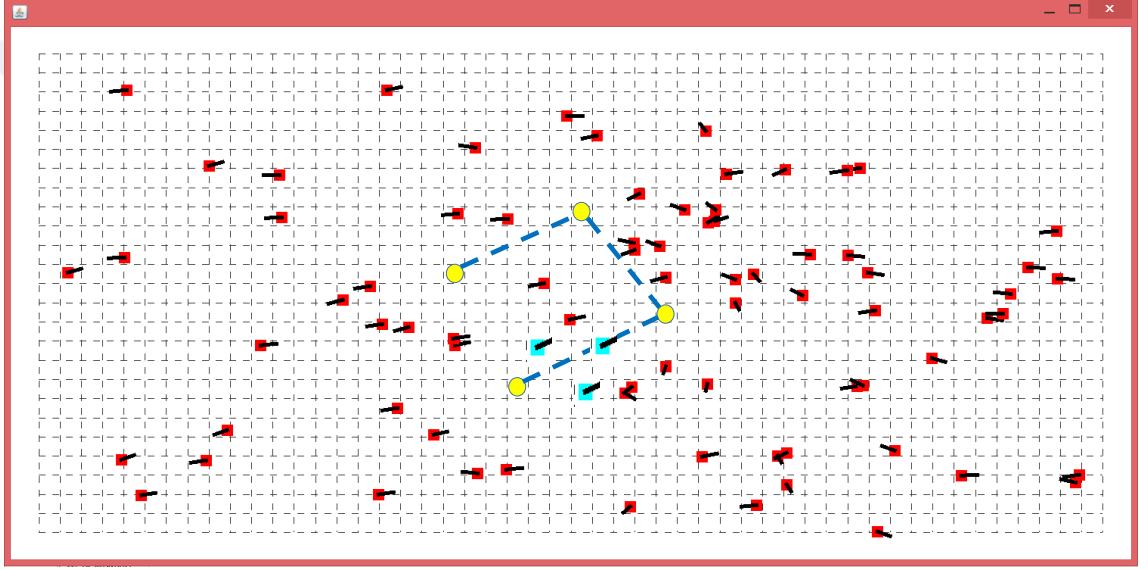


Figure 3.1: 2D snapshot of the airspace scenario in the simulation platform.

In the scenario, it is assumed that each aircraft is able to receive the surrounding traffic information using Automatic Dependent Surveillance-Broadcast (ADS-B) technology. ADS-B technology can provide an own-ship aircraft the identification, position and velocity information of surrounding aircraft that are also equipped with ADS-B.

3.1.1 UAS Conflict Detection And Avoidance Logic

UAS fly according to their pre-programmed flight plans marked by the yellow shown in the Fig. 3.1. UAS are assumed to have the dynamics of RQ-4 Global Hawk with operation speeds of $340knots$, respectively [38]. UAS are also equipped with SAA systems which enable them to detect trajectory conflicts and to initiate evasive maneuvers, if necessary. If no conflict is detected, the UAS continues to follow its mission plan. Either receiving a conflict resolution command from the

SAA system or flying based on their pre-defined flight plan, UAS always receives a velocity command during the flight. The velocity vector variation dynamics of the UAS is modeled by a first order dynamics with a time constant of 1s [39] which is represented as:

$$\dot{\vec{v}} = -(\vec{v} - \vec{v}_d), \quad (3.1)$$

where $\vec{v}$ and $\vec{v}_d$ are the current and the desired/commanded velocity vectors, respectively. The two SAA logic that are utilized in this study are developed by Fasano *et al.* [40], which is referred as SAA1, and Mujumdar *et al.* [39], which is referred as SAA2. Both of the SAA logics are consist of two phases; a *conflict detection* phase and a *conflict resolution* phase. The conflict detection phase is the same for both SAA1 and SAA2. A conflict is detected if the minimum distance between the UAS and the intruder aircraft is calculated to be less than a minimum required distance, R , during a predefined time interval. The minimum distance is calculated by projecting the trajectories of the UAS and the intruder aircraft in time. Once the conflict is detected, SAA1 and SAA2 suggest their own velocity adjustment commands in order to resolve the conflict. The velocity adjustment command of the SAA1 and SAA2 logics, v_A^{d1} and v_A^{d2} are given in the equations below

$$\vec{v}_A^{d1} = \left[\frac{v_{AB} \cos(\eta - \zeta)}{\sin(\zeta)} [\sin(\eta) \frac{\vec{v}_{AB}}{v_{AB}} - \sin(\eta - \zeta) \frac{\vec{r}}{\|\vec{r}\|}] \right] + \vec{v}_B \quad (3.2)$$

$$\vec{v}_A^{d2} = \frac{-\vec{v}_A(\frac{\vec{r}_0 \cdot \vec{v}_{AB}}{\|\vec{v}_{AB}\|}) - (R - \|\vec{r}_m\|) \frac{\vec{r}_m}{\|\vec{r}_m\|}}{\| -\vec{v}_A(\frac{\vec{r}_0 \cdot \vec{v}_{AB}}{\|\vec{v}_{AB}\|}) - (R - \|\vec{r}_m\|) \frac{\vec{r}_m}{\|\vec{r}_m\|} \|} \quad (3.3)$$

where, $\vec{v}_A$ and $\vec{v}_B$ refer to the velocity vectors of the UAS and the intruder. $\vec{r}$ and v_{AB} denote the relative position and velocity between the UAS and the intruder, respectively. ζ is the angle between $\vec{r}$ and v_{AB} and η is calculated as $\eta = \sin^{-1} \frac{R}{\|\vec{r}\|}$. $\vec{r}_0$ refers to the initial relative position vector between the UAS and the intruder. In the case of multiple conflict detection, the UAS will start an evasive maneuver to resolve the conflict that is predicted to happen earliest. The velocity adjustment suggested by the SAA1 logic guarantees minimum deviation from the trajectory, while in the case of the SAA2 logic, UAS moves to resolve the conflict until it retains the minimum safe distance with the intruder.

3.1.2 Manned Aircraft

All manned aircraft are assumed to be in their en-route phase of travel with constant speed, v , in the range of $[150 - 550]knots$. Pilots may decide to change the heading angle for $\pm 45^\circ$, or change the pitch angle for $\pm 10^\circ$, or may decide to keep both the heading and pitch angles unchanged. Once the pilot gives a heading or pitch command, the aircraft moves to the desired heading and pitch, ψ_d and θ_d , in the constant speed mode where, the heading and pitch change is modeled by a first order dynamics with the standard rate turn: a turn in which an aircraft changes its heading at a rate of 3° per second (360° in 2 minutes) [37]. This is modeled as a first order dynamics with a time constant of 10s ($45 \times (1 - 1/e)/3 \approx 10$). Therefore, the aircraft heading and pitch angle dynamics can be given as

$$\dot{\psi} = -\frac{1}{10} \times (\psi - \psi_d) \quad (3.4)$$

$$\dot{\theta} = -\frac{1}{10} \times (\theta - \theta_d) \quad (3.5)$$

and the velocity, $\vec{v} = (v_x, v_y, v_z)$, is then obtained as:

$$v_x = \|\vec{v}\| \sin \psi \cos \theta. \quad (3.6)$$

$$v_y = \|\vec{v}\| \cos \psi \cos \theta. \quad (3.7)$$

$$v_z = \|\vec{v}\| \sin \theta. \quad (3.8)$$

3.2 Pilot Decision Model

The proposed pilot decision making process model in this study is formed by combining of two methodology: the dynamic level-k reasoning and the neural fitted Q iteration (NFQ), which is an approximate reinforcement. A level-k-type model is trained by assigning level-k-1-type behavior to all of the agents (manned aircraft) except the one that is being trained. The trainee learns to react as best as he/she can in this environment using NFQ. Thus, the resulting behavior becomes a level-k type. This process starts with training a level-1 type behavior and continues until the highest desired level is reached. Once all of the desired levels are obtained, the training stage ends and, in the simulation stage,

the obtained level-k reaction models are then used in a airspace scenario explained in Section 3.2 where both manned aircraft and UAS co-exist. In the simulation certain proportions of level-0, level-1 and level-2 behavior type are assigned to the manned aircraft. It is noted that each of level-1 and level-2 agents can change their level-k behavior type based on Dynamic level-k reasoning method after observing their intruder’s behavior.

3.2.1 Dynamic Level-k Reasoning

Level-k reasoning is a game theoretical model where the main idea is that humans have various levels of reasoning in their decision-making process [27]. It has been observed that reasoning levels are related to the cognitive abilities of humans [25]. The level hierarchy is iteratively defined such that the level-k rule is a best response to the level-(k-1) rule. A level-1 decision maker (DM), for example, assumes that the other agents in the scenario are level-0 and takes actions accordingly to provide the best response. A level-2 DM takes actions to give the best response to other DMs that have level-1 reasoning and so on. From a modeling standpoint, the level-0 rule represents an initial point from which more sophisticated rules can be obtained iteratively. A Level-0 here represents a “non-strategic” DM who does not take into account other DMs, possible moves when choosing his/her own actions. This behavior can also be considered as “reflexive” because it only reacts to the immediate observations. In this study, a level-0 pilot flies an aircraft with constant heading and pitch angles starting from its initial position toward its destination.

In its conventional form, level-k reasoning help model the interactions between the DMs where a level-k DM assumes that the interacted DM has level-(k-1) reasoning. Although this approach proved to be successful in predicting short term or one-shot interactions, it misses the point that agents, during their interactions, may update their assumptions about the other agents and therefore update their own behavior. To remedy this problem, we introduce a closed loop algorithm which allows the agents to dynamically update their reasoning levels if a conflict is detected. This algorithm is explained in Fig. 3.2, where a pseudo-code is provided. It is noted that in the simulations, a simpler version of the algorithm provided in Fig. 3.2 is used, where one of the agents are randomly selected to

switch their level, instead of giving each agent a %50 chance for level switching.

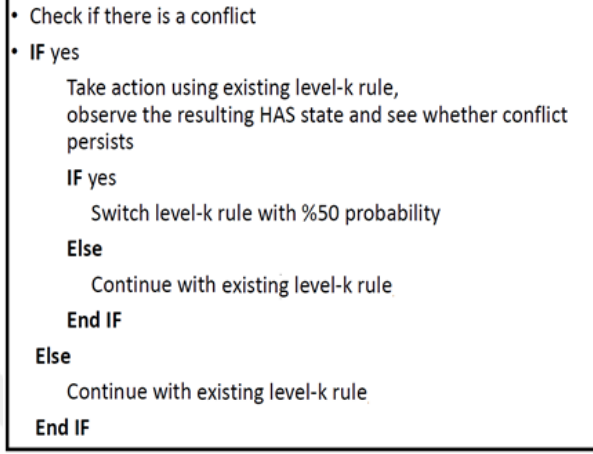


Figure 3.2: Dynamic level-k reasoning pseudo-code.

3.2.2 Neural Fitted Q Iteration

Reinforcement learning is a mathematical learning method based on reward and punishment [43]. The agent interacts with an environment through its observations, actions and a scalar reward signal received from the environment. The agent’s aim is to select actions that maximizes the cumulative future reward. Given a state, when an action increases (decreases) the value of an objective function (reward), which defines the goals of the agent, the probability of taking that action increases (decreases). Reinforcement learning algorithms involve estimating the state-action value function, or “the Q value”, which is a measure of how valuable, in terms of maximizing the total accumulated rewards, taking an action is, given the agent’s state. This estimation is generally conducted in an iterative fashion by updating these Q values after each training step. In classical Q-learning reinforcement learning algorithm, for discrete state and action spaces, the update rule is given by [29] and [31]:

$$Q(s, a) \rightarrow (1 - \alpha)Q(s, a) + \alpha(r(s, a, s') + \gamma \max_a Q(s', a)) \quad (3.9)$$

where, s , a , and s' refer to the state, action, and the successor state, respectively. α is a learning rate γ is a discounting factor. In the case of continuous state space and discrete action space, Neural Fitted Q Iteration (NFQ) method [29] and [31]

approximates the Q value function using a neural network of type multi-layer perceptron. For a given state-action pair the neural network takes the state and action as its input and provides an approximate value of the corresponding Q value for the state-action pair. The method's aim is to minimize the following error function (see [29]):

$$(Q(s, a) - (r(s, a, s') + \gamma \max_a Q(s', a)))^2. \quad (3.10)$$

where, $r(\cdot)$ is the reward signal that agent receives from the environment after the transition from state, s to state, s' by taking action a . This error function measures the deviations between state-action Q values approximated by the multi-layer perceptron ($Q(s, a)$) and the target value ($r(s, a, s') + \gamma \max_a Q(s', a)$). In the NFQ method, Q value functions are updated in a batches which means that the entire set of input patterns ($(s_i, a_i), i = 0, 1, 2, 3, \dots$) and target patterns ($(r(s_i, a_i, s'_i) + \gamma \max_a Q(s'_i, a_i)), i = 0, 1, 2, 3, \dots$) are collected and the update is performed at the end of a full episode, where an episode is defined as Concisely, the NFQ consists of two major steps: the generation of a training set and training a multi-layer perceptron using this set to obtain a Q-value function approximating the optimal state-action Q-values, at the end of each episode. The training is stopped whenever the received average reward per episode converges.

The goal of the reinforcement learning algorithm is to learn the optimal Q values by maximizing the agent's return, which is calculated via a reward/objective function. A reward function can be considered as a happiness function, goal function or utility function which represents, mathematically, the preferences of the pilot. In this paper, the pilot reward function is defined as

$$reward = w1 * (-C) + w2 * (-S) + w3 * (-A) + w4 * (-P) \quad (3.11)$$

In (11) “C” is the number of aircraft within the collision region. Based on the definition provided by the Federal Aviation Administration (FAA), the radius of collision is taken as 500ft in the horizontal direction and 100ft in the vertical direction [36]. “S” is the number of air vehicles within the separation region. The radius of the separation region is 5nm in the horizontal direction [9] and 1000ft in the vertical direction based on the “Reduced vertical separation minima” [36]. “A” represents whether the aircraft is getting closer to the intruder or going away

from the intruder in terms of their approach angle and takes the values of 1, for getting closer, or 0, for going away. “ P ” represents whether the aircraft gets closer to or goes away from its trajectory vector in terms of angle and takes the values of 0, for getting closer, or 1, for going away.

Although ADS-B provides the positions and the velocities of other aircraft, with his/her limited cognitive capabilities a pilot can not possibly process all this information during his/her decision making process. In this study, in order to model pilot limitations, including the limitations at visual acuity and perception depth, as well as the limited viewing range of an aircraft, it is assumed that the pilots can observe (or process) the information from a limited portion of the nearby airspace. This limited portion is called the “observation space”. Since the aircraft are moving in a 3D region, the observation space is a 3D portion of the nearby airspace. This observation space is considered as a portion of a sphere centered at the location of the pilot. In order to illustrate the observation space, it is divided into horizontal and vertical parts and is schematically depicted in Fig. 3.3. Since viewing range of a pilot may be different in horizontal and vertical directions, the observation space in these two directions are shown as different angular portions of a circle. Since the standard separation for manned aviation is $3 - 5nm$ [9], the radius of the observation space is taken as a variable larger than the $5nm$. Whenever an intruder aircraft moves toward the observation space (see Fig. 3.3, where Agent B is the intruder), the approach geometry is defined by two angles: ϕ_H , in the horizontal plane, and ϕ_V , in the vertical plane. The aircraft’s angular orientation with respect to his/her ideal trajectory is also defined by two angles: β_H , in the horizontal plane, and β_V , in the vertical plane. Fig. 3.3 depicts a typical example, where the aircraft B is moving toward the observation space with $\phi_H = -40^\circ$, $\phi_V = -40^\circ$, $\beta_H = -110^\circ$ and $\beta_V = 90^\circ$. Aircraft relative orientations are also coded as different “encounter types”. Figure 3.4 depicts 8 types of encounter geometries projected in the horizontal plane (left column) and four types of encounter projected in the vertical plane (right column). These geometries are indicated as $C\#i, i = 1, 2, \dots, 8$ in the figure. Finally, the observation space includes the pilot’s memory of what their actions were at the previous time step. Given an observation, the pilots can choose between five actions: turn 45° left, go straight, turn 45° right, pitch 10° up, or pitch 10° down. It is noted

that these pilot commands are filtered through the aircraft dynamics provided in Section 3.1.2.

The information of the observations and actions of the pilots fed to the neural network which is in charge of approximating the Q values. The input vector fed into the neural network is $[sign(\beta_H), sign(\beta_V), intruderstatus, sign(\phi_H), sign(\phi_V), encountertype, previousaction, action]^T$, where $sign(.)$ takes the values of +1, -1 or 0 if its argument is a positive value, negative value or zero, respectively. The *intruderstatus* is taken as 1 whenever an intruder is detected by the pilot, and 0, otherwise. The *encountertype* (see Fig. 3.4) is fed to the neural network in the form of a vector with 2 elements indicating the encounter type in horizontal plane and vertical plane. The 1st element take the values of -1, -0.5, 0.5 and +1 for encounter types of *C#1*, *C#2*, *C#3* and *C#4* in horizontal plane, and 0, otherwise. Similarly, the 2nd element take the values of -1, -0.5, 0.5 and +1 for encounter types of *C#5*, *C#6*, *C#7* and *C#8* in vertical plane, and 0, otherwise. The *previousaction* and *action* are fed to the neural network in the form of a vector with 4 elements: actions “turn 45° left”, “go straight”, “turn 45° right”, “pitch 10° up”, or “pitch 10° down” are coded as $[0, 0, 0, 1]$, $[0, 0, 1, 0]$, $[0, 1, 0, 0]$, $[1, 0, 0, 0]$ and $[-1, -1, -1, -1]$, respectively.

3.3 Simulation Results and Discussion

In this section, a quantitative analysis of multiple UAS integration in a crowded airspace is presented. Before presenting these results, single encounter scenarios, where two manned aircraft with different reasoning levels which are in a collision path, are investigated.

3.3.1 Single Encounter Scenarios Between Manned Aircraft

Figure 3.5 presents the separation violation rates of manned aircraft in 5000 random single encounters where pilots are modeled as level-1 and level-2 decision makers. Separation violation occurs when both the horizontal and the vertical

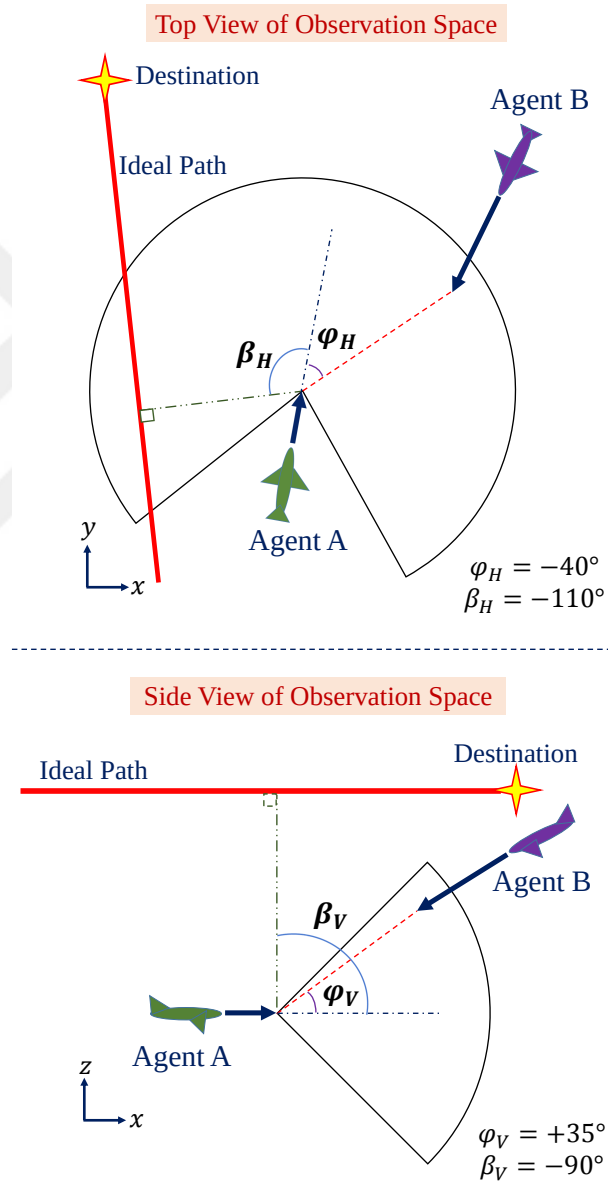


Figure 3.3: Pilot observation space.

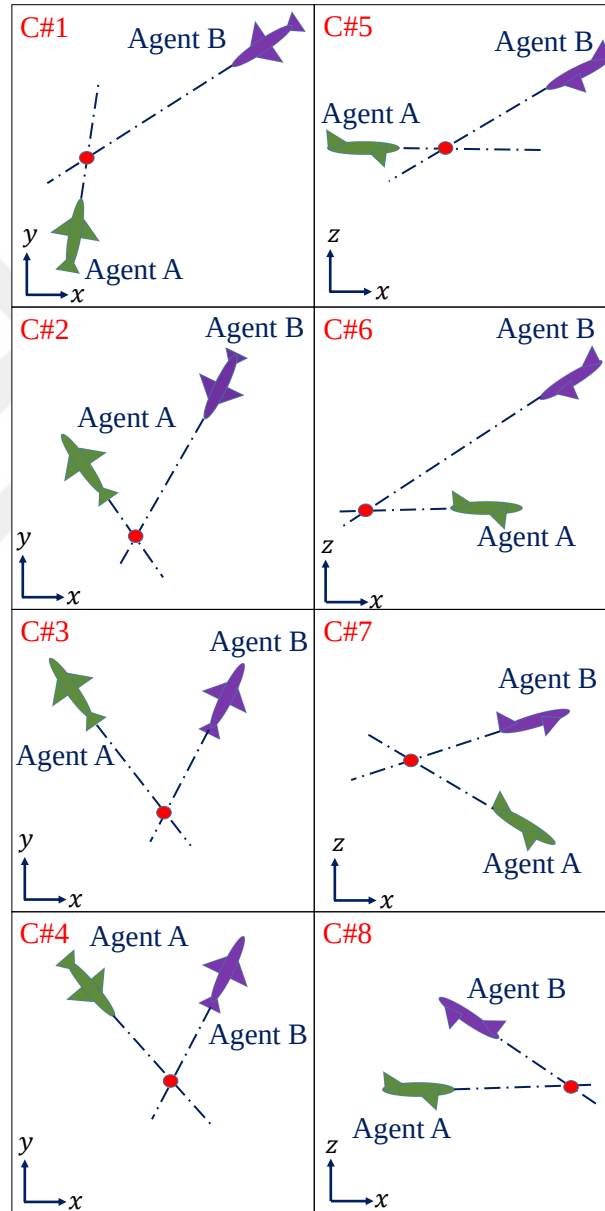


Figure 3.4: Encounter categories.

distances between the two aircraft are less than the horizontal separation requirement, $5nm$ [9] and the reduced vertical separation requirement, $1000ft$ [9], respectively. In the figure, separation violations are shown for 3 different “distance horizon” values, which is the radius of the observation space depicted in Fig. 3.3. Any distance less than the distance horizon is defined as miss separation. Pilots oversee a $20s$ time window prior to a probable miss separation distance with a $5second$ decision frequency for choosing their actions. Distance horizon takes three values: $5nm$ (equal to horizontal separation requirement), $7.5nm$ and $10nm$. Pilots are either level-0, level-1 or level-2 agents. There are 4 possible types of scenarios: 1) level-1 pilot vs. level-0 pilot, 2) level-2 pilot vs. level-1 pilot, 3) level-1 pilot vs. level-1 pilot, and 4) level-2 pilot vs. level-2 pilot. According to the Fig. 3.5, in the 1st type and 2nd type scenarios (level- k vs level- $(k-1)$), by increasing the distance horizon from $5nm$ to $10nm$ the separation violation rate decreases from %96.6 to %11.9. It is noted that %9.8 of the $10nm$ distance horizon separation violations is occur during the scenarios where the encounters are difficult to be resolved. This is because for these type of encounters, no matter how pilots in the collision path maneuver to resolve the conflict the separation violation occur. In the 3rd type and fourth type of scenarios (level- k vs level- k) the separation violation rate decreases by increasing the pilots’ distance horizon, however, separation violation rate is still high (%41.2) even for the $10nm$ case. The reason for this high separation violation rate is that when a level- k agent encounters another agent with the same level, his/her assumption about the other becomes invalid. This problem and its implications were discussed in Section 3.2.1, where a remedy were proposed (see Fig 3.2), which resulted in a “dynamic” level- k reasoning method. According to Fig. 3.5, the separation violation rate, for regular encounters (red), decreases from %11.1 to %2.6, when dynamic level- k reasoning is chosen as the interactive decision making model, for the case of $10nm$ distance horizon.

Two 3D single encounter scenarios are designed to schematically show the performance of pilots: 1) 1st scenario consists of two manned aircraft flying with constant speeds in different altitude levels where, their initial horizontal distance and vertical distances are $21.3nm$ and $1000ft$, respectively, 2) 2nd scenario consists of two manned aircraft where one of the manned aircraft is flying in a

horizontal plane with constant speed and the other manned aircraft is leveling up with a constant vertical rate of $1968\text{ft}/\text{min}$ with initial horizontal distance and vertical distances of 21nm and 1000ft , respectively. In these two scenarios, two manned aircraft are flying in an encounter path with each other and pilots can oversee conflicts in a 20s time window prior to miss separation. All pilot's distance horizon is considered to be 10nm . Both of the aircraft can change the heading angle for $\pm 45^\circ$, or change the pitch angle for $\pm 10^\circ$, or may keep both the heading and pitch angles unchanged. In both of the scenarios 4 cases for pilots' level-k type behavior are considered: 1) a level-1 pilot vs. a level-0 pilot, 2) a level-2 pilot vs. a level-1 pilot, 3) a level-1 pilot vs. a level-1 pilot, and 4) a dynamic level-1 pilot vs. a level-1 pilot.

Figure 3.6, Fig. 3.7, Fig. 3.8 and Fig. 3.9 depict the 2D snapshots the four cases for the 1st scenario. The black squares, red squares and green squares correspond to manned aircraft with level-0 type behavior, level-1 type behavior and level-2 type behavior. The circles stand for the initial location and destination of the aircraft. The gray track line right behind the manned aircraft represent their traveled path from their initial positions to where they stand in the snapshot. Figure 3.10 and Fig. 3.11 depict the horizontal and vertical distances between the two aircraft for all the 4 cases in the 1st scenario. During the conflict resolution the minimum horizontal distance and vertical distance of the two aircraft are 2.7nm and 1825ft in the 1st case scenario, respectively and 4.5nm and 1471ft in 2nd case scenario. Therefore, pilots were able to resolve the conflict.

Taking a look at the 3rd case scenario where two pilots are level-1 type agents and assume that their intruders are level-1 type agents, pilots turn toward each other instead of resolving the encounter. During the encounter the minimum horizontal distance and minimum vertical distances are 0.5nm and 991ft and separation violation occurs. In the 4th case scenario, one of the pilots (the pilot starting its motion from left to right side of the snapshot) is able to observe its intruder's maneuver. This pilot first execute a maneuver such that its intruder is a level-1 type agent and turns toward its intruder. Then the pilot observes its intruder's maneuver within 10s , project their trajectories and detects a critical conflict and changes its level-1 type rule to the level-2 type rule. In this case during the encounter the minimum horizontal distance and the minimum vertical

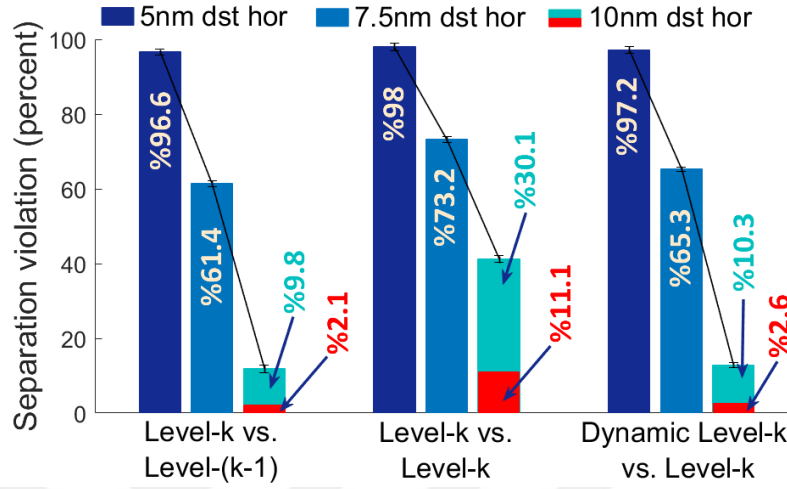


Figure 3.5: Separation violation rates. “dst hor” refers to the distance horizon of the pilots. On the columns that shows the separation violation rates for a 10nm distance horizon, the cyan color shows the percentage of violations that occur when the encounters are “difficult to resolve”, meaning that the initial conditions of the encounters do not permit any type of pilot action to avoid a separation violation.

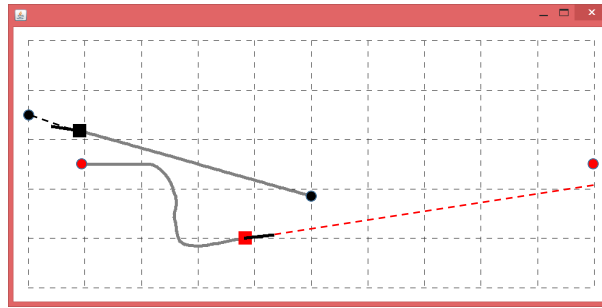


Figure 3.6: Sample encounter 1: level-1 pilot vs. level-0 pilot.

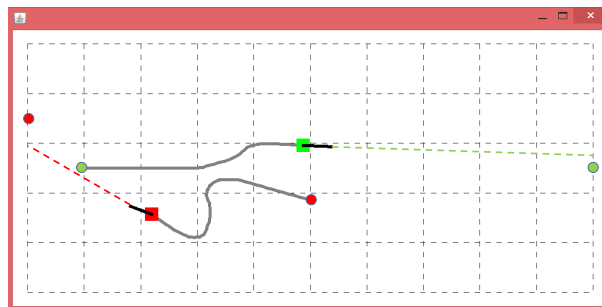


Figure 3.7: Sample encounter 1: level-2 pilot vs. level-1 pilot.

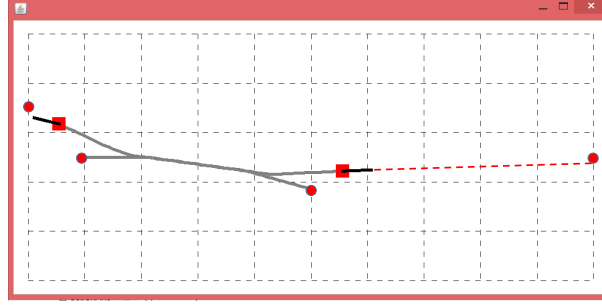


Figure 3.8: Sample encounter 1: level-1 pilot vs. level-1 pilot.

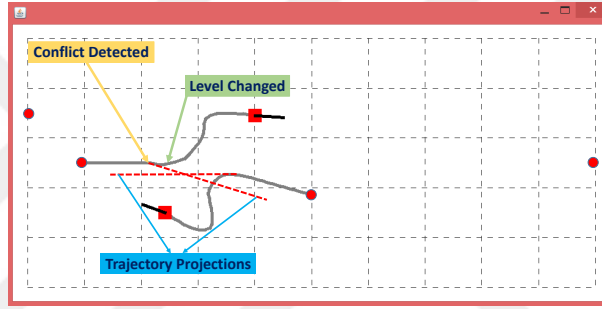


Figure 3.9: Sample encounter 1: dynamic level-1 pilot vs. level-1 pilot.

distances are $3.5nm$ and $1051ft$, therefore, separation violation does not occur. It is noted that if pilots decided to go straight and do not participate in the conflict resolution the minimum horizontal distance and the minimum vertical distances are $0.54nm$ and $1000ft$.

Figure 3.12, Fig. 3.13, Fig. 3.14 and Fig. 3.15 depict the 2D snapshots for the four cases of the 2nd scenario. Figure 3.16 and Fig. 3.17 represent the horizontal and vertical distances of the two manned aircraft. During conflict resolution in the 1st case scenario the minimum horizontal distance and the minimum vertical

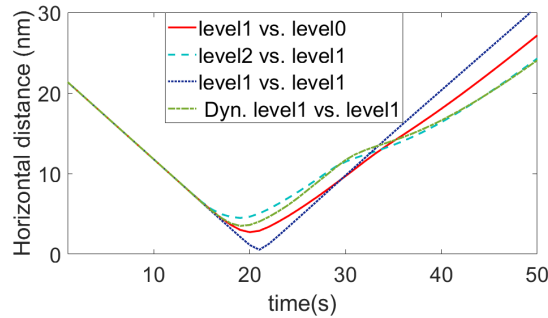


Figure 3.10: Sample encounter 1: horizontal distance.

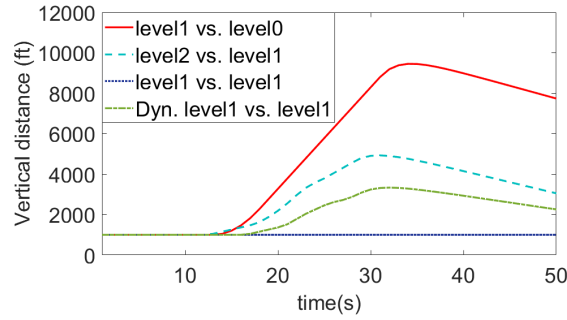


Figure 3.11: Sample encounter 1: vertical distance.

distance of the two aircraft are $2.8nm$ and $1847ft$ and in the 2nd case scenario, the minimum horizontal distance and the minimum vertical distance of the two aircraft are $3nm$ and $2367ft$. Similar to the 1st scenario, pilots resolve the conflict. During conflict resolution in the 3rd case scenario the minimum horizontal distance and the minimum vertical distance of the two aircraft are $0.56nm$ and $966ft$ and in the 4th case scenario, the minimum horizontal distance and the minimum vertical distance of the two aircraft are $3.3nm$ and $1001ft$. Similar to the 1st scenario, utilizing the dynamic level-k rule, pilots are able to resolve the conflict.

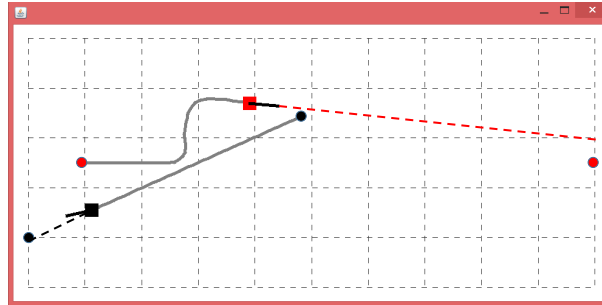


Figure 3.12: Sample encounter 2: level-1 pilot vs. level-0 pilot.

3.3.2 Statistical Results For UAS Integration

In this section, the scenario explained in Section 3.1 is simulated to investigate the effect of *responsibility assignment* for conflict resolution, on safety and performance for various number of UAS integrated in to the crowded airspace. It is noted that the separation responsibility is an important issue in addressing the

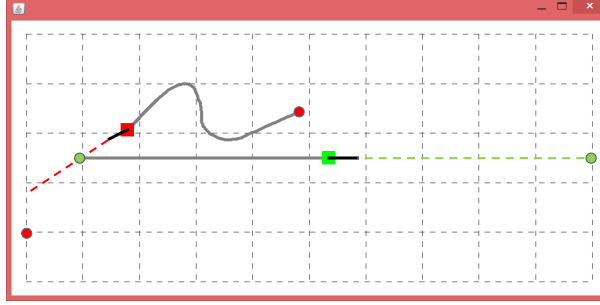


Figure 3.13: Sample encounter 2: level-2 pilot vs. level-1 pilot.

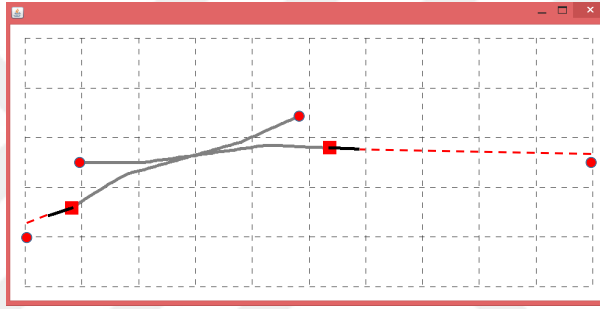


Figure 3.14: Sample encounter 2: level-1 pilot vs. level-1 pilot.

integration of UAS into NAS [4]: it is crucial to determine which of the agents (manned aircraft or UAS) will take the responsibility of the conflict resolution. Since the loss of separation is the most serious issue, the safety metric is taken as the total number of separation violations between all aircraft whether manned or unmanned. Performance metric, on the other hand, is taken as the averaged manned and unmanned aircraft trajectory deviations. In all of the simulations, level-0, level-1 and level-2 pilot policies are randomly distributed over the manned aircraft in such a way that 10% of the pilots fly based on level-0 policies, 60% of

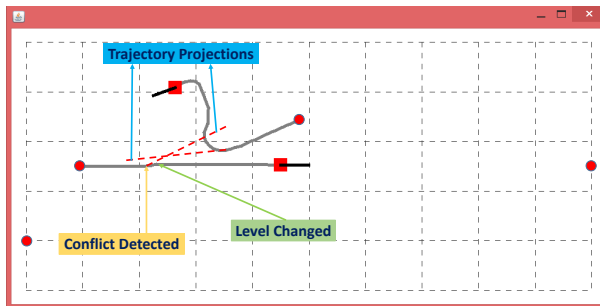


Figure 3.15: Sample encounter 2: dynamic level-1 pilot vs. level-1 pilot.

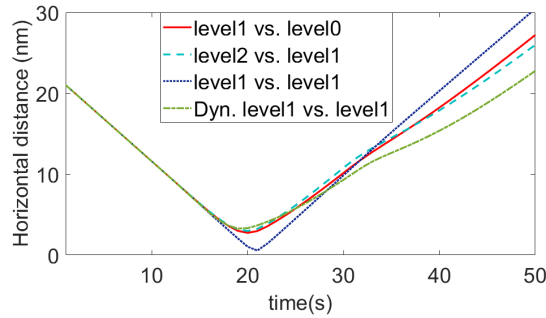


Figure 3.16: Sample encounter 2: horizontal distance.

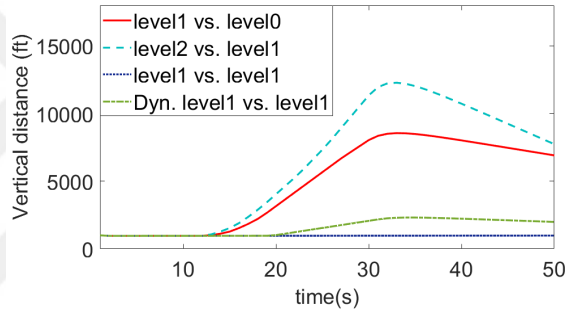


Figure 3.17: Sample encounter 2: vertical distance.

the pilots act based on level-1 policies and 30% use level-2 policies. This distribution is obtained from human experimental studies discussed in [17] and may not necessarily reflect the true distribution for pilots. It is noted, however, that level distribution can easily be adapted to other distributional data in this framework. Level-1 type and level-2 type pilots utilize the dynamic level-k reasoning method.

Figure 3.18, Fig. 3.20, Fig. 3.19, and Fig. 3.21 depict a comparison of different resolution responsibility cases for the two SAA1 and SAA2 logic: manned aircraft are responsible (dark blue), both manned aircraft and UAS are responsible (blue) and only the UAS is responsible (cyan). In the case when only manned aircraft are responsible for conflict resolution, UAS is forced to continue its path without executing the SAA system and manned aircraft act as dynamic level-1 and level-2 DMs. In the case when the UAS is responsible for the conflict resolution, the manned aircraft are forced to continue their path without changing their heading and the UAS executes its SAA system. In the case when both the manned aircraft and the UAS are responsible for the conflict resolution, they both execute their evasive maneuvers. Figure 3.18 shows that manned aircraft

deviate more from their trajectory when only the manned aircraft share resolution responsibility, compared to the case when both of the UAS and the manned aircraft are responsible. This is true for both the SAA1 (the results on the left) and the SAA2 (the results on the right) algorithms. On the other hand, Fig. 3.20 shows that the UAS deviates from its trajectory more when it is responsible for the resolution, compared to the case when the responsibility is shared, when SAA1 is utilized. For the case of SAA2 utilization, the case is the same. Figure 3.19 shows, as expected, that for both SAA1 and SAA2, the UAS flight times are the shortest when only the manned aircraft become responsible for the resolution. According to the Fig. 3.21 for both SAA1 and SAA2 utilizations, the safest case is when the resolution responsibility is shared between the UAS and the manned aircraft..

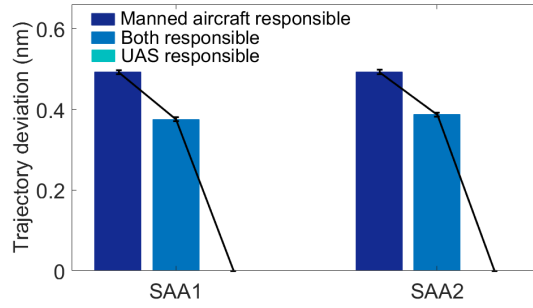


Figure 3.18: Average trajectory deviation of manned aircraft.

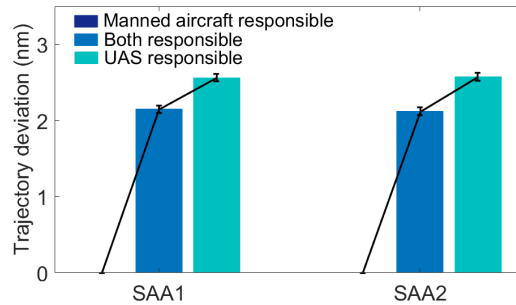


Figure 3.19: Average trajectory deviation of UAS.

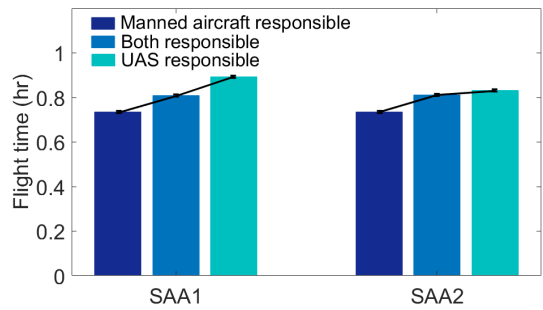


Figure 3.20: UAS flight time.

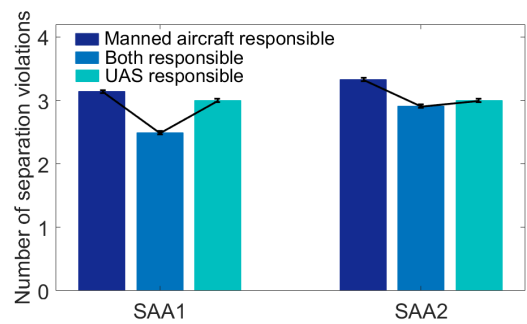


Figure 3.21: Separation violation between manned aircraft and UAS.

Chapter 4

Conclusion

In this thesis, a 2D and a 3D game theoretical modeling framework are proposed for use in the integration of Unmanned Aircraft Systems (UAS) into the National Airspace System (NAS), as a means for concept evaluations. In both of these frameworks, the inclusion of the human decision making process in modeling the human pilot behavior is the main contribution, which fills an important gap in the literature of hybrid airspace systems (HAS) modeling, where in general, the pilot reactions are assumed to be known a priori. The method used in both of the approaches provides probabilistic outcomes of complex scenarios where both manned and unmanned aircraft co-exist. Thus, by being able to provide quantitative analyses, the proposed frameworks prove themselves to be useful in investigating the effect of various system variables, such as separation distances and the utilization of different Sense And Avoid (SAA) algorithms, on the safety and performance metrics of the airspace system. These frameworks can also be used to quantitatively analyze the effect of different responsibility assignments for conflict resolution and SAA design parameters such as UAS distance horizon and UAS time horizon, on safety and performance of the national airspace. These analyses can be helpful to evaluate technologies and concepts related to UAS integration into the NAS. It is noted that, the proposed frameworks are flexible so that any rules and procedures that the pilots are required to follow, for example Traffic Control Alert Systems (TCAS) advisories, can be incorporated into the model. Different manned aircraft and UAS dynamic models can be incorporated to the framework as well.

Bibliography

- [1] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, “On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the national airspace system,” *Progress in Aerospace Sciences*, vol. 44, no. 7, pp. 503–519, 2008.
- [2] U. Roadmap, “Integration of civil unmanned aircraft systems (uas) in the national airspace system (nas) roadmap,” *Federal Aviation Administration, Purpose and Background of Civil UAS Roadmap*, vol. 7, 2013.
- [3] E. R. S. Group *et al.*, “Roadmap for the integration of civil rpas into the european aviation system,” tech. rep., European Comission, Technical Report, 2013, Available: <http://ec.europa.eu/enterprise/sectors/aerospace/files/rpas-roadmap-en-pdf>, 2013.
- [4] C. W. Johnson, “Unmanned aircraft systems (uas) integration in the national airspace system (nas) project,” 2012.
- [5] M. T. DeGarmo, “Issues concerning integration of unmanned aerial vehicles in civil airspace,” *Center for Advanced Aviation System Development*, p. 4, 2004.
- [6] D. Maki, C. Parry, K. Noth, M. Molinario, and R. Miraflor, “Dynamic protection zone alerting and pilot maneuver logic for ground based sense and avoid of unmanned aircraft systems,” in *Infotech@ Aerospace*, pp. 1–10, 2012.
- [7] M. Kochenderfer, L. Espindle, J. Kuchar, and J. D. Griffith, “Correlated encounter model for cooperative aircraft in the national airspace system version 1.0,” *Project Report ATC-344, Lincoln Laboratory*, 2008.

- [8] J. Kuchar, J. Andrews, A. Drumm, T. Hall, V. Heinz, S. Thompson, and J. Welch, “A safety analysis process for the traffic alert and collision avoidance system (tcas) and see-and-avoid systems on remotely piloted vehicles,” in *AIAA 3rd” Unmanned Unlimited” Technical Conference, Workshop and Exhibit, September 20-23 2004, Chicago, Illinois*, 2004.
- [9] M. Pérez-Batlle, E. Pastor, P. Royo, X. Prats, and C. Barrado, “A taxonomy of uas separation maneuvers and their automated execution,” in *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems*, pp. 1–11, IRIT Press, 2012.
- [10] F. Martel, R. R. Schultz, and Z. Wang, “Unmanned aircraft systems sense and avoid flight testing utilizing ads-b transceiver,” in *AIAA Infotech@Aerospace Conference*, pp. 1–8, 2010.
- [11] T. B. Billingsley, “Safety analysis of tcas on global hawk using airspace encounter models,” tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF AERONAUTICS AND ASTRONAUTICS, 2006.
- [12] J. Alfredson, P. Hagström, and B.-G. Sundqvist, “Situation awareness for mid-air detect-and-avoid system for remotely piloted aircraft,” *Procedia Manufacturing*, vol. 3, pp. 1014–1021, 2015.
- [13] J. Alfredson, P. Hagstrom, B. Sundqvist, and J. Pellebergs, “Workload of a collision avoidance system for remotely piloted aircraft,” in *Proceedings of the 46th Annual Nordic Ergonomics Society Conference, IEA Press*, pp. 819–824, 2014.
- [14] E. Salas, F. Jentsch, and D. Maurino, *Human factors in aviation*. Academic Press, 2010.
- [15] R. Lee and D. Wolpert, “Game theoretic modeling of pilot behavior during mid-air encounters,” in *Decision Making with Imperfect Decision Makers*, pp. 75–111, Springer, 2012.
- [16] J. Kuchar and A. C. Drumm, “The traffic alert and collision avoidance system,” *Lincoln Laboratory Journal*, vol. 16, no. 2, p. 277, 2007.

- [17] M. A. Costa-Gomes, V. P. Crawford, and N. Iriberri, “Comparing models of strategic thinking in van huyck, battalio, and beil’s coordination games,” *Journal of the European Economic Association*, vol. 7, no. 2-3, pp. 365–376, 2009.
- [18] D. O. Stahl and P. W. Wilson, “On players’ models of other players: Theory and experimental evidence,” *Games and Economic Behavior*, vol. 10, no. 1, pp. 218–254, 1995.
- [19] S. Backhaus, R. Bent, J. Bono, R. Lee, B. Tracey, D. Wolpert, D. Xie, and Y. Yildiz, “Cyber-physical security: A game theory model of humans interacting over control systems,” *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 2320–2327, 2013.
- [20] R. Lee, D. H. Wolpert, J. W. Bono, S. Backhaus, R. Bent, and B. Tracey, “Counter-factual reinforcement learning: How to model decision-makers that anticipate the future.,” *Decision Making and Imperfection*, vol. 474, no. 4, pp. 101–128, 2013.
- [21] Y. Yildiz, R. Lee, and G. Brat, “Using game theoretic models to predict pilot behavior in nextgen merging and landing scenario,” in *AIAA Modeling and Simulation Technologies Conference*, pp. 1–8, 2012.
- [22] Y. Yildiz, A. Agogino, and G. Brat, “Predicting pilot behavior in medium-scale scenarios using game theory and reinforcement learning,” *Journal of Guidance, Control, and Dynamics*, 2014.
- [23] N. Musavi, D. Onural, K. Gunes, and Y. Yildiz, “Unmanned aircraft systems airspace integration: A game theoretical framework for concept evaluations,” *Journal of Guidance, Control, and Dynamics*, 2016.
- [24] N. Musavi, K. B. Tekelioglu, Y. Yildiz, K. Gunes, and D. Onural, “A game theoretical modeling and simulation framework for the integration of unmanned aircraft systems in to the national airspace,” in *AIAA Infotech@Aerospace*, p. 1001, 2016.

- [25] D. Gill and V. Prowse, “Cognitive ability, character skills, and learning to play equilibrium: A level-k analysis,” *Journal of Political Economy*, vol. 124, no. 6, pp. 1619–1676, 2016.
- [26] D. Gill and V. L. Prowse, “Cognitive ability and learning to play equilibrium: A level-k analysis,” *Analysis*, 2012.
- [27] J.-K. Chong, T.-H. Ho, and C. Camerer, “A generalized cognitive hierarchy model of games,” *Games and Economic Behavior*, vol. 99, pp. 257–274, 2016.
- [28] T.-H. Ho and X. Su, “A dynamic level-k model in sequential games,” *Management Science*, vol. 59, no. 2, pp. 452–469, 2013.
- [29] M. Riedmiller, “Neural fitted q iteration-first experiences with a data efficient neural reinforcement learning method,” in *ECML*, vol. 3720, pp. 317–328, Springer, 2005.
- [30] M. Riedmiller, M. Montemerlo, and H. Dahlkamp, “Learning to drive a real car in 20 minutes,” in *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007*, pp. 645–650, IEEE, 2007.
- [31] T. Gabel, C. Lutz, and M. Riedmiller, “Improved neural fitted q iteration applied to a novel computer gaming and learning benchmark,” in *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on*, pp. 279–286, IEEE, 2011.
- [32] 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.
- [33] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.
- [34] T. Jaakkola, S. P. Singh, and M. I. Jordan, “Reinforcement learning algorithm for partially observable markov decision problems,” in *Advances in neural information processing systems*, pp. 345–352, 1995.

- [35] M. N. ERAST, “Non-cooperative detect, see, & avoid (dsa) sensor study,” tech. rep., NASA ERAST Technical Report, July, 2002.
- [36] J. Planning, “Development office,nextgen. concept of operations for the next generation air transportation system, version 2.0, 2007.”
- [37] P. E. Illman, *The pilot’s handbook of aeronautical knowledge*. Tab Books, 1995.
- [38] K. Dalamagkidis, K. Valavanis, and L. A. Piegler, *On integrating unmanned aircraft systems into the national airspace system: issues, challenges, operational restrictions, certification, and recommendations*, vol. 54. Springer Science & Business Media, 2011.
- [39] A. Mujumdar, R. Padhi, *et al.*, “Reactive collision avoidance using nonlinear geometric and differential geometric guidance,” *Journal of guidance, control, and dynamics*, vol. 34, no. 1, p. 303, costa 2011.
- [40] G. Fasano, D. Accardo, A. Moccia, C. Carbone, U. Ciniglio, F. Corrado, and S. Luongo, “Multisensor based fully autonomous non-cooperative collision avoidance system for uavs,” *AIAA Infotech@ Aerospace*, pp. 7–10, 2007.
- [41] A. M. Law, “How to build valid and credible simulation models,” in *Proceedings of the 40th Conference on Winter Simulation*, pp. 39–47, Winter Simulation Conference, 2008.
- [42] MITRE-Corporation, “Systems engineering guide: Verification and validation of simulation models,” <https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/other-se-lifecycle-building-blocks-articles/verification-and-validation-of-simulation-models>. Accessed 2017-07-21.
- [43] M. Wiering and M. Van Otterlo, “Reinforcement learning,” *Adaptation, Learning, and Optimization*, vol. 12, 2012.