Predicting Pilot Behavior in Medium Scale Scenarios Using Game Theory and Reinforcement Learning

Yildiray Yildiz* and Adrian Agogino †
U. C. Santa Cruz, Moffett Field, California, 95035, USA

Guillaume Brat‡
Carnegie Mellon University, Moffett Field, California, 95035, USA

Effective automation is critical in achieving the capacity and safety goals of the Next Generation Air Traffic System. Unfortunately creating integration and validation tools for such automation is difficult as the interactions between automation and their human counterparts is complex and unpredictable. This validation becomes even more difficult as we integrate wide-reaching technologies that affect the behavior of different decision makers in the system such as pilots, controllers and airlines. While overt short-term behavior changes can be explicitly modeled with traditional agent modeling systems, subtile behavior changes caused by the integration of new technologies may snowball into larger problems and be very hard to detect.

To overcome these obstacles, we show how integration of new technologies can be validated by *learning* behavior models based on goals. In this framework, human participants are not modeled explicitly. Instead, their goals are modeled and through reinforcement learning their actions are predicted. The main advantage to this approach is that modeling is done within the context of the entire system allowing for accurate modeling of all participants as they interact as a whole. In addition such an approach allows for efficient trade studies and feasibility testing on a wide range of automation scenarios. The goal of this paper is to test that such an approach is feasible. To do this we implement this approach using a simple discrete-state learning system on a scenario where 50 aircraft need to self-navigate using Automatic Dependent Surveillance-Broadcast (ADS-B) information. In this scenario, we show how the approach can be used to predict the ability of pilots to adequately balance aircraft separation and fly efficient paths. We present results with several levels of complexity and airspace congestion.

I. Introduction

A key element to meet the continuing growth in air traffic is the increased use of automation. Decision support systems, computer-based information acquisition, trajectory planning systems, high level graphic display systems and all advisory systems are considered to be automation components related to Next Generation (Next-Gen) airspace.[1] In the Next-Gen Air System, a larger number of interacting human and automation systems are expected as compared to today. Improved tools and methods are needed to analyze this new situation and predict potential conflicts or unexpected results, if any, due to increased human-human and human-automation interactions. In a recent NASA report,[1] among others, "Human-Automation Function Allocation", "Methods for Transition of Authority and Responsibility as a Function of Operational Concept" and "Transition from Automation to Human Control" are mentioned as "Highest Priority Research Needs" for Next-Gen airspace development.

*Associate Scientist, University Affiliated Research Center, NASA Ames Research Center, MS 269-1, Moffett Field, CA, AIAA Senior Member.
†Scientist, University Affiliated Research Center, NASA Ames Research Center, MS 269-1, Moffett Field, CA.
‡Technical Lead, Silicon Valley Campus, NASA Ames Research Center, MS 269-1, AIAA Senior Member.

There have been several methods developed for modeling, optimizing and making predictions in airspace systems. Brahms agent modeling[2] framework has been successfully used to model human behavior but it is not used to predict possible outcomes of large scale complex systems with human-human and human-automation interactions. For optimization Tumer and Agogino[3] used agent-based learning to optimize air traffic flow but they did not model pilot behavior, which is critical for being able to predict system outcomes.

In the proposed approach, we firstly mathematically define pilot goals in a complex system. These goals can constitute, for example, "staying on the trajectory", "not getting close to other aircraft" or "having a smooth landing". We then use game theory and machine learning to model the outcomes of the overall system based on these pilot goals together with other automation and environment variables.

Formally, we utilize of a game-theoric framework known as Semi Network-Form Games (SNFG),[4] to obtain probable outcomes of a Next-Gen scenario with interacting humans (pilots) in the presence of advanced Next-Gen technologies. Our focus is to show how this framework can be scaled to larger problems that will make it applicable to a wide range of air traffic systems. Earlier implementations of this framework[4–7] proved useful for investigating strategic decision making in scenarios with two humans. In this paper, for the first time, we investigate a dramatically larger scenario which includes 50 aircraft corresponding to 50 human decision makers. The method presented in the paper is a step towards predicting the effect of new technologies and procedures on the air space system by investigating pilot reactions to the new medium. These predictions can be utilized to evaluate the performance vs efficiency trade-offs.

In section II, we explain how game theory is employed in predicting the complex system behavior. In this section, we also present the two components of this approach: "Level-K reasoning" and "Reinforcement learning". In section III, we present the main components of the Next-Gen scenario that we investigate. In this section, we explain the airspace and aircraft models together with pilot goals and a general description of the scenario. In section IV, we provide simulation set-up details. In section V, we show the simulation results where we investigate 4 different variations of the Next-Gen scenario with different levels of complexity and congestion. Finally, in section VI, we conclude the paper by giving a summary and take-away notes of this study together with future research directions.

II. Game Theory Based Prediction

Game theory is used to analyze strategic decision making amongst a group of "players". Typically, players represent human decision makers, though the concept of a player can be expanded to other decision makers including animals in evolutionary game theory or complex automated decision makers. In this paper, players are pilots. In the context of this paper, the key aspect of players is that they observe the environment, they take actions based on these observations, and the actions they take influence the environment and the other players (See Figure 1). The goal of game theory is to predict the actions of these players based on their goals. These goals are represented as "reward functions" which are some function of the system state. We assume that the players are trying (though imperfectly) to maximize their reward functions.

Given a set of goals represented as reward functions, we can then try to predict the actions of the players. However, several challenges need to be overcome:

- Figuring out how a player can attempt to maximize their reward function can be a difficult inverse problem.
- Players may not be able to perfectly maximize their reward functions.
- The best action of a player will depend on the actions of all the other players. Multiple solutions may exist, and many solutions may be unstable.

The best ways of handling these issues heavily depend on the number of players, the size of the state space, the size of the action space and on the complexity of the reward functions. In this paper, we utilize a concept called "Level-K" reasoning combined with reinforcement learning.

Our goal is to predict the behavior of a particular player, yet how this player behaves depends on the behavior of other players. Level-K reasoning helps us address this problem through a hierarchical approach, where we begin by assigning basic behaviors to every player. Then, given the reward function of a player, and basic behaviors of other players, we predict how a player will behave. Reinforcement learning helps us to make these predictions in an iterative manner for games with multiple stages. This approach is explained in more detail below.

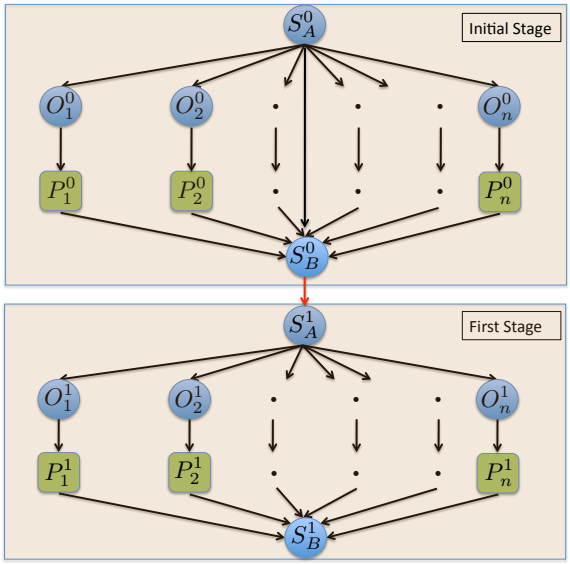


Figure 1: Schematic representation of the Next-Gen scenario with n number of aircraft, as a multi-stage game. The initial and the first stage of the game is shown in the figure. S, O and P represent states, observations and pilots, respectively.

A. Level-K reasoning

The basic idea in level-K reasoning[8,9] is that humans show different levels of reasoning in games. The lowest level, level-0 reasoning, is non-strategic, meaning that a level-0 player does not take other players' possible moves into consideration. Level-0 strategies can be random or can be constructed using expert system knowledge. A level-1 player assumes that other players have level-0 reasoning and tries to maximize his/her reward function based on this assumption. Similarly, a level-2 player assumes that other players have level-1 reasoning, and so on. It is noted that once a player makes a certain level assumption about the other players, other players simply becomes a part of the environment and the problem reduces to single agent decision making.

B. Reinforcement learning

SNFG framework[6] extends the standard level-K reasoning to model time-extended scenarios. In a time extended scenario with N steps, a player makes N action choices. Therefore, the player need to optimize his/her policy - his map from observations/memory to actions - to maximize the average reward $\sum_{i=1}^{N}(r_i/N)$, where r_i represents the reward at time step i. Reinforcement learning (RL) is a tool that is used to tweak player policies at each time step towards maximizing the reward without knowing the underlying model of the system. RL algorithm takes system states as inputs and gives an appropriate action (agent move) as the output. When the actions are performed, the system states change. The reward is calculated based on these new states and RL algorithm uses this reward to update the agent policy. In the next round, the updated policy is used to produce the next action given the new states. See Fig. 2. This process continues until the average reward converges to a certain value.

There are various reinforcement learning methods that can be utilized for this purpose.[10] In this paper, we use a method developed by Jaakkola.[11] The reason for this choice is that the Jaakkola algorithm has local converge guarantees for scenarios where the player can not observe all of the system states, which is the case for the scenario investigated in this paper. The details of the scenario is explained in the following sections.

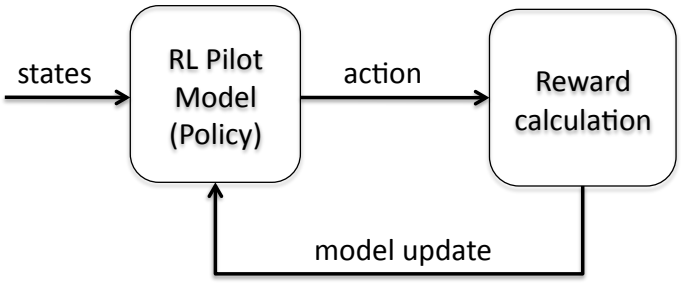


Figure 2: Reinforcement Learning (RL) schematic diagram. System states are the inputs to the RL algorithm. Given system states, the existing pilot model produces a corresponding action. A reward is calculated based on the new states created as a result of this action. Based on this reward, RL algorithm updates the pilot model. This process continues until the average reward converges to a fixed value.

III. Next-Gen Scenario Model

We test the game theoric approach on an air traffic scenario, where 50 aircraft have to space themselves efficiently using Automatic Dependent Surveillance-Broadcast (ADS-B). ADS-B is a satellite-based technology that provides aircraft the ability to receive other aircraft ID, position and velocity. This technology is expected to support Next Generation (Next-Gen) airspace operations where the volume of operations are projected to be dramatically higher than what it is now. In the scenario, 50 aircraft are approaching to a single sector. (In the existing airspace system, sector capacities are much lower, but it is expected that to achieve Next-Gen airspace goals, sector capacities will need to be increased dramatically.) Thanks to the ADS-B technology, pilots are aware of other aircraft, to a certain degree. Given this ADS-B information, pilots are supposed to continue flying on their assigned trajectory while at the same time protecting separation from other aircraft.

A. Airspace model

Aircraft are assumed to be at the en-route phase of the flight, flying level at the same altitude, throughout the scenario. Accordingly, the airspace is approximated as a two dimensional Cartesian grid.

B. Aircraft model

Aircraft are assumed to be controlled by an automatic pilot in velocity control mode. This is approximated by allowing aircraft to move to a neighboring intersection in the grid, either diagonally or straight, at every time step.

C. Scenario Description

1. At time $t = t_0$, aircraft have their initial positions and directions, p_0^i and d_0^i, $i = 1, 2, .., 50$, where 50 is the number of aircraft in the scenario. Initial positions p_0^i are either randomly or with a certain structure assigned on the grid with the exclusion of a sector region in the center. Initial directions d_0^i are assigned in such a way that each aircraft aims towards the center of the sector. As an example for random initial position assignment, see Fig. 3. At time $t = t_0$ a goal position, gp^i, which is where the aircraft is supposed to reach, is also assigned to each aircraft. This goal position gp is simply where the initial direction arrow intersects an edge of the grid.

2. At times $t = t_k, k = 1, 2, ..$, aircraft move towards the center of the sector, and towards their goal position gp. Pilots observe surrounding aircraft and tries to protect separation while following their assigned trajectory. The assigned trajectory is a straight-line from the initial position p_0 to the goal position gp.

D. Pilot Reward Function

Pilot's reward function, or "goal function", U_i, is a mathematical representation of the preferences of the pilot about different states of the system. For the investigated scenario, it is assumed that the following factors plays a role in pilot decisions:

1. Preventing a separation violation

The most important task for the pilots is to keep a safe distance from other aircraft. In the simplified scenario, a separation violation is modeled as two or more aircraft sharing the same intersection in the grid. Therefore, the first term of the reward function is formed as:

$$u_1 = -N_{violation}, \tag{1}$$

where $N_{violation}$, "number of separation violations", represents the number of aircraft existing in the same intersection with the considered aircraft. The minus sign reveals that this term needs to be minimized to maximize the overall reward function.

2. Decreasing the probability of a separation violation

Pilots' second important task is keeping the aircraft at a safe distance from other aircraft and therefore decreasing the probability of a separation violation. The aircraft that are at the neighboring intersections of the aircraft in consideration are assumed to be at a "non-safe" distance and hence increase the likelihood of a separation violation. The pilots' goal is to minimize the number of these surrounding aircraft during flight. The second term, modeling this goal, is given as:

$$u_2 = -N_{neighbor}, \tag{2}$$

where $N_{neighbor}$ stands for "number of neighboring aircraft".

3. Staying on the assigned trajectory

Pilots' third task is to stay at their assigned trajectories. This task is divided into two components. The first component is approaching to the final goal point. The second component is staying as close as possible to the assigned path. An aircraft can approach to it's final destination without staying very close to the assigned path. Similarly, an aircraft can stay exactly on the assigned path and not approach the final destination, if, for example, it goes on the opposite direction. So, the mutual existence of these two subtasks are necessary.

The first task, getting close to the final destination, is modeled by an indicator function which gets the value 1 or 0 depending on whether after each step they are closer (1) or not (0) to their final destination in the grid. This is expressed as:

$$u_{31} = I_{close}, \tag{3}$$

where, I_{close} stands for "the indicator function for getting close to the final destination".

The second subtask, staying on the assigned path, is modeled by the negative of the distance of the aircraft to the closest point on the assigned path. This is expressed as

$$u_{32} = D_{path}, \tag{4}$$

where, D_{path} stands for "the distance to the assigned path".

4. Minimizing Effort

As human beings, pilots tend to choose inaction or the action that needs the least effort, if possible. This final term is modeled as:

$$u_4 = -I_{effort}, \tag{5}$$

where I_{effort} takes the value 1 if pilots change aircraft heading and 0 otherwise.

Combining the above components, the reward function U for a given pilot can be given as

$$U = \omega_1(-N_{violation}) + \omega_2(-N_{neighbor}) + \omega_{31}(I_{close}) + \omega_{32}(D_{path}) + \omega_4(-I_{effort}), \tag{6}$$

where ω_is are the weighting assigned to each component.

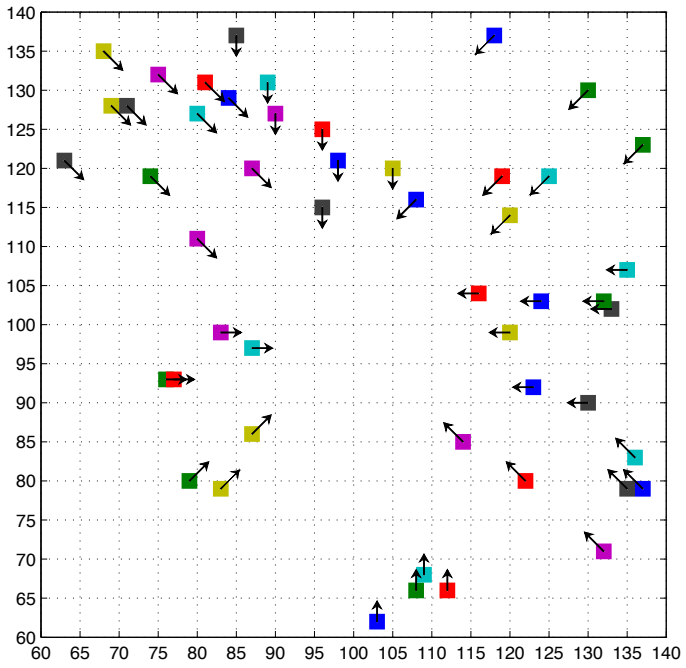


Figure 3: Initial positions and directions of aircraft. 50 aircraft are randomly distributed on an 80x80 Cartesian grid, excluding a 20x20 sector region in the center. Their directions are assigned in such a way that all aircraft aims toward this center sector. In the figure, axes are marked by the increments of 5 for the clarity of representation.

IV. Simulation Setup

To represent the airspace, an 80x80 Cartesian grid is used. At time $t = t_0$, 50 aircraft are distributed on this grid, either randomly or with a certain structure, excluding a central region, which represents a sector. Aircraft directions are assigned in such way that all aircraft head towards the sector. See Fig. 3 for a random initial distribution.

A. Pilot move space

In this model, pilots are assumed to have 3 actions: diagonal right, diagonal left and straight.

B. Pilot observations and memory

ADS-B technology can provide pilots the information, position, velocity etc. of other aircraft. However, a pilot has limited ability to utilize all this information for his/her decision making. For this scenario, we model these pilot limitations by assuming that pilots can observe a limited section of the grid in front of them. Pilot observations on the grid are presented in Fig. 4, where Pilot A observes whether or not any aircraft is headed towards the regions that are marked by an "x" sign. In this particular example, another aircraft is heading towards one of these regions that is marked with a green x sign. Therefore, Pilot A will see this section on the grid as "full", while the rest of his observation space, the red x signs, will be "empty".

In addition to these ADS-B observations, pilots also know their configuration, i.e. "diagonal" or "straight", their "best directional move" (M_{BD}) and "best trajectory move" (M_{BT}). M_{BD} is the move that would make the aircraft approach to its final destination more than any alternative move would. Similarly, M_{BD} is the move that would make the aircraft approach to its trajectory more than any alternative move. Finally, pilots have a memory of what their actions were at the previous time-step.

8 ADS-B observations, 1 configuration, 1 M_{BD}, 1 M_{BD} and 1 previous move make up totally 12 inputs for the reinforcement learning algorithm. Observations and configuration have binary values, 1 or 0. Previous

move, M_{BD} and M_{BD} have 3-dimensions each: diagonal left, diagonal right or straight. Therefore, the number of states for which the reinforcement learning algorithm need to assign appropriate actions is $2^9 \times 3^3 = 13824$.

Figure 5 shows a schematic diagram of RL pilot model inputs and outputs.

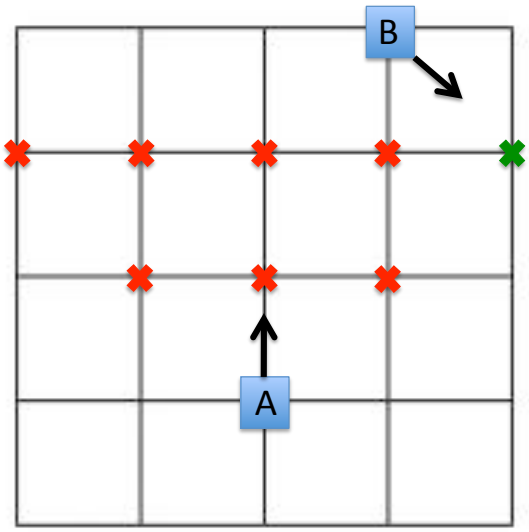


Figure 4: Pilot observations. Pilots can observe 8 points in front of them. If any other aircraft can occupy any of these observation points in the next time-step, assuming that they will keep moving in their current direction, that point is considered "full". In the example given in this figure, Pilot A observes all 8 points marked by "x". Pilot B is heading towards one of these points, the green x, and will occupy that point in the next time-step if he/she continues to fly with his/her current direction. Therefore, for Pilot A, the green x is considered as "full" (1) while the rest of his observation points, the red x, are considered "empty" (0).

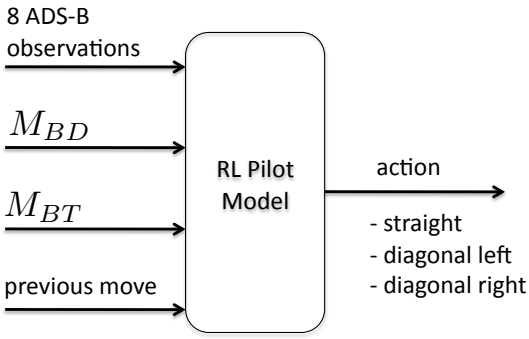


Figure 5: Reinforcement Learning pilot model inputs and output. The model gets ADS-B observations, best directional move, best trajectory move and the previous move as inputs and chooses one of the possible actions among "straight", "diagonal left" and "diagonal right" as the output.

C. Level-0 pilot

In general, level-0 players are modeled as uniformly random, i.e. they do not have any preference over any moves. However, depending on the application, this selection may vary. One important property of level-0 players is that they need to be non-strategic: their actions should be independent of other players' actions. In this scenario, we modeled level-0 players as pilots that fly with a predetermined fixed heading, regardless of other pilots' positions or intents.

V. Simulation Results

In this section, 4 safety-related scenarios are investigated to show the predictive capabilities of the proposed approach. In these scenarios, we explore the safety issues such as loss of separation and deviations from the assigned trajectories, together with pilot performances via "average rewards" pilots obtain during their flight. We also make predictions on how high-density air traffic effect these issues.

We first use reinforcement learning (RL) to obtain level-1 and level-2 pilot policies, which are mappings from system states to actions. RL training simulations are conducted by initializing the player's policy with a uniform random distribution over actions and then running the RL algorithm which tweaks the player policy in certain episodes to increase the average reward. These runs are stopped when the average reward converges to a fixed value. When a level-1 pilot is being trained, level-0 behavior is assigned to the remaining pilots. Similarly, when a level-2 pilot is being trained, level-1 behavior is assigned to the remaining pilots.

After level-1 and level-2 pilot policies are determined, we simulated the following scenarios to make system level predictions.

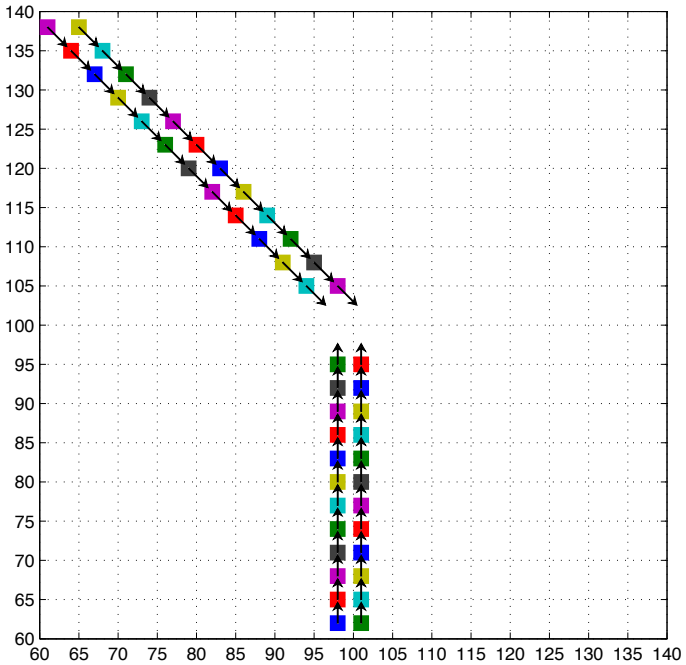
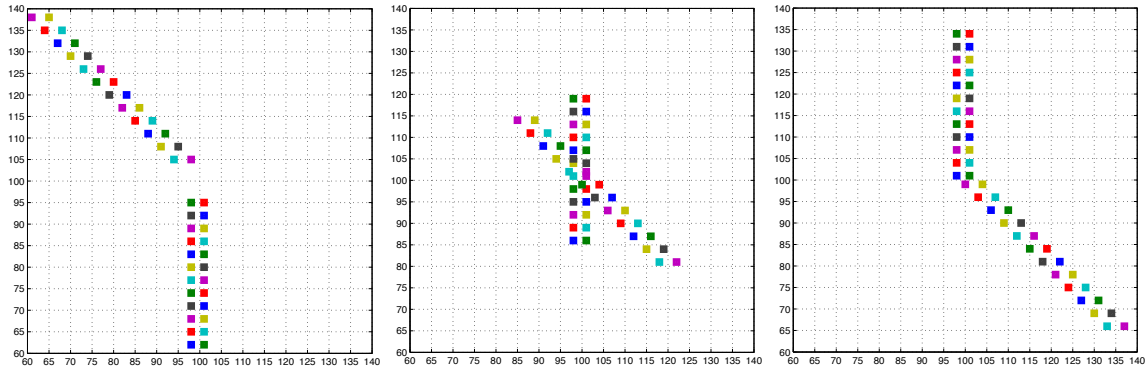


Figure 6: Initial positions and headings of the aircraft for Scenario 1. Colored squares represent aircraft and arrows indicate aircraft heading. The aircraft are initialized in such a way that if they do not change their initial heading and they fly with an equal constant speed, there will be no separation violations.

A. Scenario 1: Introducing self-navigating aircraft to airspace - Configuration 1

In this scenario, two sets of aircraft are flying in fixed trajectories towards a sector located at the center of the air space grid. Figure 6 shows the initial positions of the aircraft together with their heading. The aircraft are located in such a way that there is no danger of separation violation if aircraft follow the assigned trajectories, which are straight lines, perfectly. It is reminded that a separation violation is modeled as two or more aircraft sharing the same grid intersection. Level-0 pilots are defined as pilots flying with a predetermined fixed heading, regardless of the surrounding aircraft presence. In this scenario, we start with assigning level-0 behavior to all pilots and then replace these pilots, in increasing numbers, with level-1 pilots.

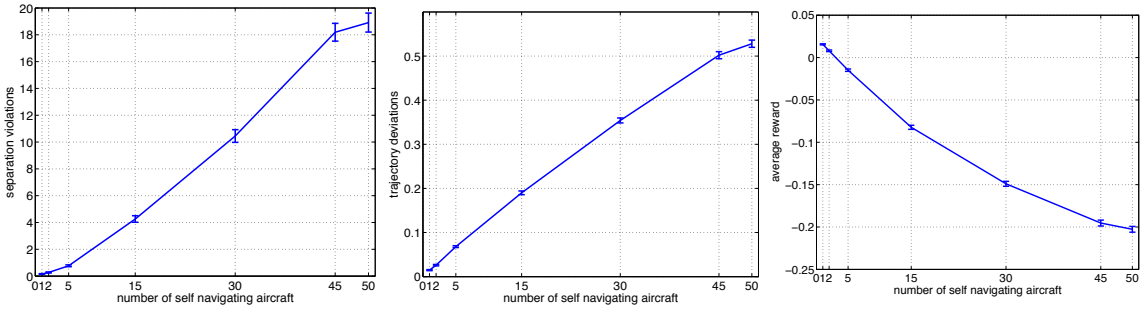
Figure 7 presents the evolution of this scenario, when all pilots are level-0. As expected, the aircraft follow perfect fixed trajectories and no separation violation event occurs. This is not an interesting result, since we already knew the outcome: The pilots were given trajectories and spaced such that no safety violations would occur. The real question we are after is "What happens if we start replacing these perfectly spaced



(a) Initial positions (b) Positions at simulation step 24 (c) Positions at simulation step 39

Figure 7: The evolution of Scenario 1 when all the aircraft have fixed paths with constant heading. Since the initial positions are set to prevent a separation violation in the case of all fixed heading aircraft, no such violations occur during the scenario.

pilots with self-navigating pilots?". It is noted that the outcome of this may be unpredictable as the original solution is somewhat brittle. As explained earlier, self navigating pilots have ADS-B technology onboard and they can observe their surroundings as depicted in Fig. 4. In this scenario, we modeled self navigating aircraft pilots as level-1 strategic thinkers: They assume that other pilots are level-0 and then they try to choose optimal actions that will maximize their reward functions. Pilot reward function was explained in Section III.D.



(a) Separation violations (b) Trajectory deviations (c) Average rewards

Figure 8: The effect of introducing self-navigating aircraft into a perfectly structured airspace (config 1), in terms of a) Separation violations, b) Trajectory deviations and c) Average rewards. Separation violations are represented by two or more aircraft sharing the same grid. Trajectory deviations are the average distance, in terms of unit grid length, of the aircraft to their respective assigned trajectories, averaged over all aircraft. Average rewards are the average value of the reward functions averaged over all aircraft.

We simulated the system after replacing various number of fixed trajectory aircraft with self-navigating aircraft. Figure 8 shows the effects of this newly introduced ADS-B equipped aircraft into the system, in increasing numbers. It is seen that as the number of self navigating aircraft in the system increases, the number of separation violations and trajectory deviations increases, as expected. As a consequence, the average pilot reward decreases. It is noted that the original scenario is a special one where the aircraft trajectories are very close to each other and therefore to prevent separation violations, these trajectories are very carefully assigned. Under these circumstances, self-navigating pilots' assignments are very challenging: The system is brittle, there is no room for even small deviations from the trajectories. On the other hand, self-navigating pilots can not observe the whole airspace and they do not get any guidance from the ground. They operate only with the observations they obtain from ADS-B.

The quantitative analysis so far may suggest that the introduction of self-navigating pilots in a tightly

spaced airspace without ground control holds serious risks. It also shows that, in general, the speed of increase in separation violations increases as the number of self-navigating aircraft increases, whereas the speed of increase in trajectory deviations decreases. This may reveal that unpredictable aircraft behavior may be less of a concern compared to separation violations.

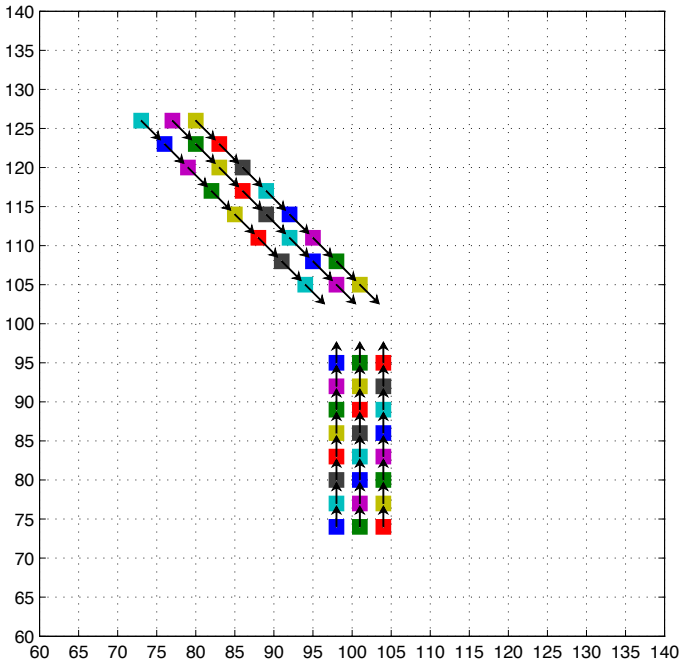


Figure 9: Initial positions and headings of the aircraft for Scenario 2. Colored squares represent aircraft and arrows indicate aircraft heading. The aircraft are initialized in such a way that if they do not change their initial heading and they fly with an equal constant speed, there will be no separation violations.

B. Scenario 2: Introducing self-navigating aircraft to airspace - Configuration 2

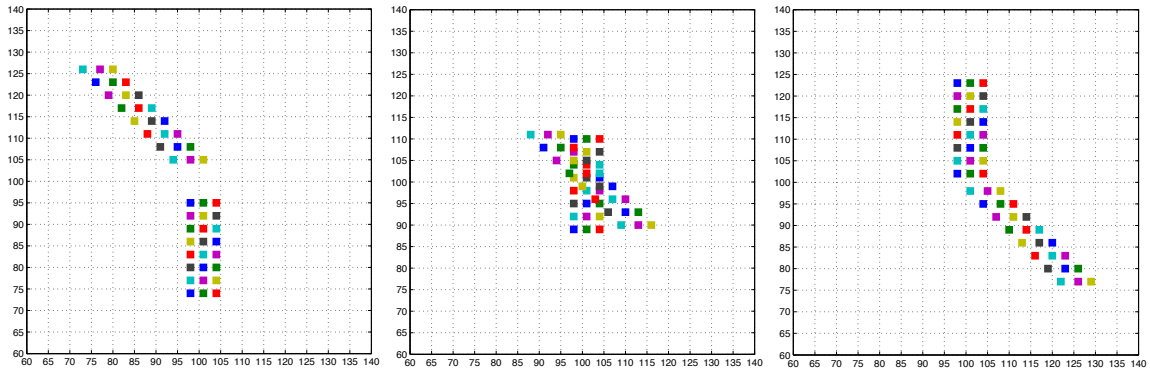
This scenario is similar to the first one in that we replace fixed trajectory aircraft with self-navigating aircraft in an airspace scenario and observe the effects on separation violations, trajectory deviations and average pilot rewards. However, we now use a different flying configuration, which is shown in Fig. 9. This configuration is more brittle than the first one since there is less free space around the aircraft, on average. Figure 10 presents the evolution of this scenario when all aircraft have fixed trajectories (level-0 pilot). As in the previous scenario, when there is no self-navigating aircraft there occurs no separation violations, by design.

Figure 11 presents a comparison between the effects of replacing the fixed trajectory aircraft with self-navigating aircraft in configurations 1 and 2, in terms of separation violations, trajectory deviations and average rewards. Since the second configuration is more brittle, the number of separation violations and trajectory deviations are larger, which translates in to lower average rewards.

The quantitative analysis of the effect of brittle trajectories on safety and efficiency may be useful for future design of aircraft routes. For example, although configuration 2 causes more separation violations, in general, it may be more efficient to design the trajectories as such due to some other considerations. This quantitative analysis may help find a "sweet spot" or a balance between brittleness of the system and efficiency, which will result a safe and efficient, in term of throughput, for example, airspace.

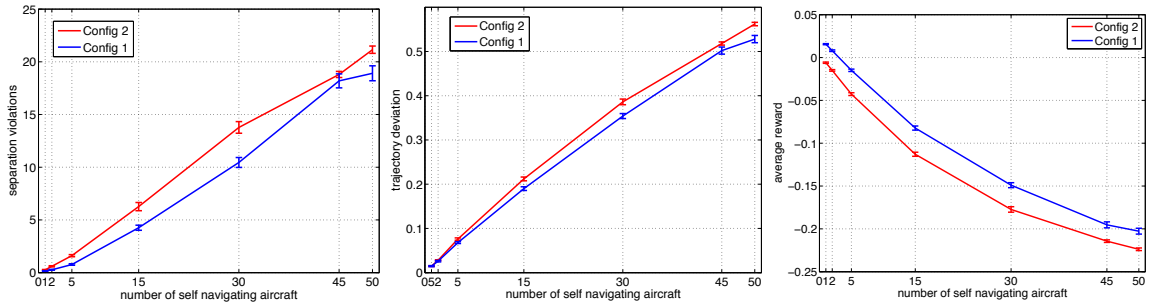
C. Scenario 3: Introducing self-navigating aircraft, with a different ADS-B setting, to airspace

In the previous two scenarios, we investigated the effect of introducing ADS-B equipped self navigating aircraft to airspace. We assumed that the ADS-B data link provided these pilots the positions of nearby aircraft. Their observation space was given in Fig. 4. In this scenario, we assume that the set of observations that a pilot can use is smaller: self-navigating pilots can only use the observations at 3 points in front of



(a) Initial positions (b) Positions at simulation step 15 (c) Positions at simulation step 28

Figure 10: The evolution of Scenario 2 when all the aircraft have fixed paths with constant heading. Since the initial positions are set to prevent a separation violation in the case of all fixed heading aircraft, no such violations occur during the scenario.



(a) Separation violations (b) Trajectory deviations (c) Average rewards

Figure 11: Comparison of the effects of introducing self-navigating aircraft for two different flying configurations in terms of a) Separation violations, b) Trajectory deviations and c) Average rewards. The second configuration is more brittle than the first one since there is less free space around the aircraft, on average. As a result, average number of separation violations and trajectory deviations are larger, which also translates into lower average rewards.

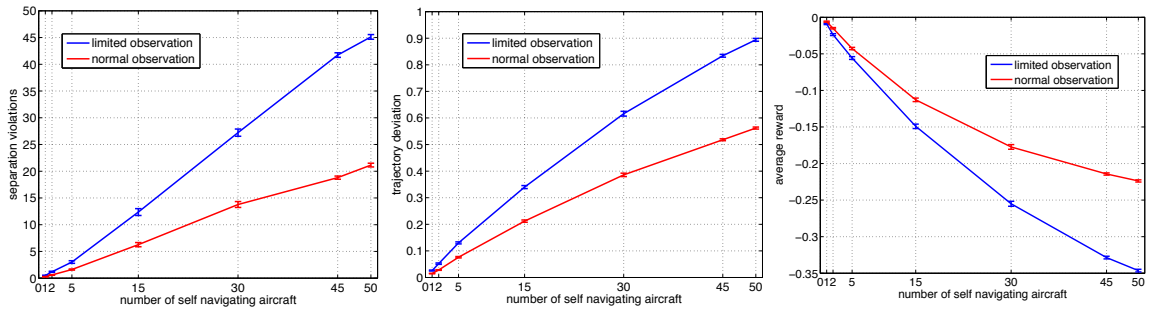
them: straight ahead, diagonal right and diagonal left grid points. This may correspond to a different ADS-B setting that gives more limited information to the pilot, or pilots not being able to handle more information.

Figure 12 shows the effects of having an ADS-B system that provides less information about surrounding aircraft, by comparing the results with the previous scenario, where the pilots had a larger observation space. Number of separation violations, trajectory deviations and average rewards are effected negatively, as expected. What is interesting is that the system deterioration is faster than linear with the increase in the number of self-navigating aircraft.

The results of this investigation may give clues on the amount of information that is needed to be provided to the pilots with ADS-B equipped aircraft. This goes without saying that the results presented here are obtained from simulating a simplified scenario to show the capabilities of the game theoric approach. In real applications, the assumptions and simplifications should be carefully tailored depending on the complexity of the problem.

D. Scenario 4: Changing airspace density

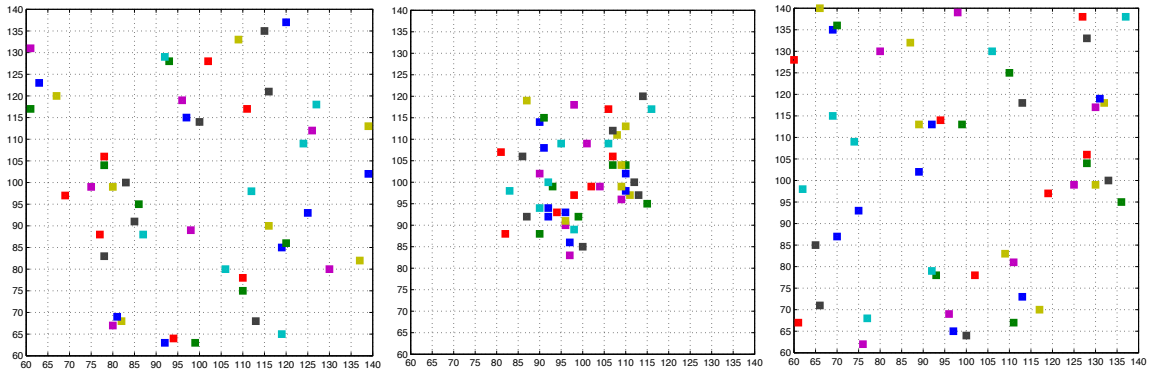
In this scenario, we investigate how airspace density makes a quantitative effect on system safety. The scenario begins with aircraft randomly initialized in the airspace with assigned trajectories that will make them fly on straight lines towards a sector located in the middle of the airspace grid and continue flying



(a) Separation violations (b) Trajectory deviations (c) Average rewards

Figure 12: Comparison of the effects of introducing self-navigating aircraft to airspace for two different ADS-B settings which results in two different observation spaces. "Normal observation" corresponds to the observation space shown in Fig. 4. In "limited observation" setting, pilots can observe only the first 3 grid points in front of them, the one that is directly in front of them, one that is in front diagonal right and one in front diagonal left. The limitation of surrounding aircraft information causes dramatic safety problems as observed from the figures. It is interesting to see that the deterioration of the airspace system safety is faster than linear with the increase in the number of self navigating aircraft.

straight until they reach the boundaries of the grid. Figure 3 shows an example initialization with 50 aircraft. Figure 13 shows the evolution of this scenario when we use 50 aircraft with level-0 pilots. It is reminded that by design, level-0 pilots never change their initial heading and fly on straight line trajectories.



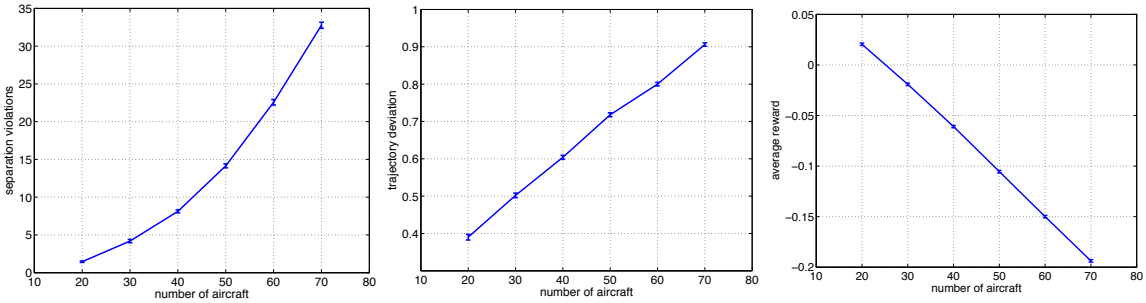
(a) Initial positions (b) Positions at simulation step 29 (c) Positions at simulation step 50

Figure 13: The evolution of scenario 4 when all the aircraft have fixed paths with constant heading. Aircraft are randomly initialized on the grid excluding a 20x20 central sector. All aircraft are assigned directions that will make them go towards the central sector on a straight line, and continue in the same direction until they reach the boundaries of the grid. In this simulation, all aircraft pilots obey the initial directions and they never change their heading. These pilots, as explained earlier, are "level-0" pilots.

To make the scenario more realistic, we used a mixture of pilot types level-0, level-1 and level-2. Some experimental studies[12] show that, in general, level-0 type has minimum frequency and level-1 types are more frequent then level-2 types. Level-3 types are rare. This is intuitive since the amount of reasoning gets unreasonably taxing for humans as levels increase. These type distributions are regarded as behavior parameters. Existing data or previous analysis can be used for estimating type distributions. For our simulations, we used the following type distributions: 10% level-0, 60%level-1 and 30% level-2.

Figure 14 presents simulation results where the effect of airspace density variations on separation violations, trajectory deviations and average rewards is quantitatively investigated. As expected, as the air density increases, all these variables are negatively effected. An interesting result is that although trajectory deviation and average reward varies linearly with airspace density, separation violations shows an almost

quadratic increase. These quantitative estimation analysis may serve as a useful tool for designing Next-Generation airspace structure where dramatically increased airspace densities are expected. However, the scenario investigated here is a simpler version of reality since the focus is to show the capabilities of the approach. In real applications, the assumptions should be carefully tailored for the specific scenarios.



(a) Separation violations (b) Trajectory deviations (c) Average rewards

Figure 14: The effect of airspace density variations on a) Separation violations, b) Trajectory deviations and c) Average rewards. In this scenario, a mixture of pilot types is used to obtain a more realistic collective behavior. Separation violations are represented by two or more aircraft sharing the same grid. Trajectory deviations are the average distance, in terms of unit grid length, of the aircraft to their respective assigned trajectories, averaged over all aircraft. Average rewards are the average value of the reward functions averaged over all aircraft. As airspace density increases, all these variables are negatively affected. However, although trajectory deviations and average rewards show linear change, separation violations increase almost quadratically.

VI. Discussion and Conclusion

For next generation automation technology to be implemented, a compelling safety analysis will have to be made. This is a daunting task, especially in large systems where there is extensive human/automation interaction and where a human participant may change his/her behavior in unexpected ways based on the actions of the other elements in the system. We believe that the best way to validate this technology integration is to model the human/automation interaction implicitly through learning algorithms. In this paradigm, the goals of the participants are modeled explicitly, but the behavior of the participants are modeled through reinforcement learning, allowing us to predict behavior in a large integrated system.

In this paper, we test an implementation of this framework on a simplified scenario where ADS-B information is being integrated into a 50 aircraft system, allowing some of the aircraft to self-navigate . We show how the framework can be used to predict various safety aspects in scenarios that include human-human and human-automation interactions. We provide simulation results that present the quantitative effect of introducing self-navigating aircraft into the airspace on separation violations, trajectory deviations and pilot performances. In addition, we show results that present the effect of airspace density increases on the same variables.

The focus of this work is to show the predictive capabilities of the proposed approach for midscale airspace scenarios, using simplified system models. In the future, we plan to investigate more complex integration tasks. These tasks will likely involve continuous variables, large-scale simulation and modeling behavior at multiple resolutions of detail.

References

[1]Sheridan, T. B., Corker, K. M., and Nadler, E. D., "Final report and recommendations for research on human-automation interaction in the Next Generation Air Transportation System," Technical Report (DOT-VNTSC-NASA- 06-05), U.S. Department of Transportation, Research and Innovative Technology Administration., Cambridge, MA, USA, 2006.

[2]Acquisti, A., Sierhuis, M., Clancey, W. J., and Bradshaw, J. M., "Agent based modeling of collaboration and work

practices onboard the international space station," *Proc. Eleventh Conference on Computer-Generated Forces and Behavior Representation*, Vol. 8, 2002, pp. 315–337.

[3]Tumer, K. and Agogino, A., "Distributed agent-based air traffic flow management," *Proc. 6th International Joint Conference On Autonomous Agents And Multiagent Systems*, No. 255, 2007.

[4]Lee, R. and Wolpert, D., *Chapter: Game theoretic modeling of pilot behavior during mid-air encounters*, in Decision making with multiple imperfect decision makers. Intelligent Systems Reference Library Series. Springer, 2011.

[5]Yildiz, Y., Lee, R., and Brat, G., "Using Game Theoretic Models to Predict Pilot Behavior in NextGen Merging and Landing Scenario," *Proc. AIAA Modeling and Simulation Technologies Conference*, No. AIAA 2012-4487, Minneapolis, Minnesota, Aug. 2012.

[6]Lee, R., Wolpert, D., Bono, J., Backhaus, S., Bent, R., and Tracey, B., "Counter-factual reinforcement learning: How to model decision-makers that anticipate the future," *CoRR*, Vol. abs/1207.0852, 2012.

[7]Backhaus, S., Bent, R., Bono, J., Lee, R., Tracey, B., Wolpert, D., Xie, D., and Yildiz, Y., "Cyber-Physical Security: A Game Theory Model of Humans Interacting over Control Systems," *CoRR*, Vol. abs/1304.3996, 1304.3996.

[8]Stahl, D. and Wilson, P., "On players models of other players: Theory and experimental evidence," *Games and Economic Behavior*, Vol. 10, No. 1, 1995, pp. 218254.

[9]Costa-Gomes, M. and Crawford, V., "Cognition and behavior in two-person guessing games: An experimental study," *American Economic Review*, Vol. 96, No. 5, 2006, pp. 17371768.

[10]Wiering, M. and van Otterlo, M., editors, *Reinforcement Learning, State-of-the-art*, Springer, 2012.

[11]Jaakkola, T., Satinder, P. S., and Jordan., I., "Reinforcement learning algorithm for partially observable Markov decision problems," *Advances in Neural Information Processing Systems 7: Proceedings of the 1994 Conference*, 1994.

[12]Costa-Gomes, M. A., Crawford, V. P., and Iriberri, N., "Comparing Models Of Strategic Thinking In Van Huyck, Battalio, And Beil's Coordination Games," *Games and Economic Behavior*, Vol. 7, No. 2-3, 1995, pp. 365–376.