Eleventh USA/Europe Air Traffic Management Research and Development Seminar (ATM2015)

Autonomous Air Traffic Control for Non-Towered Airports

Zouhair Mahboubi
Aeronautics and Astronautics
Stanford University
Stanford, California, USA
zouhair@stanford.edu

Mykel J. Kochenderfer
Aeronautics and Astronautics
Stanford University
Stanford, California, USA
mykel@stanford.edu

***Abstract*—Half of all reported near mid-air collisions involve at least one general aviation aircraft. More than half of NTSB reports of mid-air collisions occur in the vicinity or in the traffic pattern of an airport, and a majority of them occur at non-towered airports. This paper proposes a concept for traffic collision prevention targeted for general aviation aircraft operating in the vicinity of non-towered airports. We envision an autonomous air traffic control system as a non-intrusive, ground based system with no additional requirements to participating aircraft except for radio communication. We outline how such a system can be modeled and solved as a Markov decision process and present simulation results for aircraft in the traffic pattern.**

Keywords: Separation; Air-ground integrated concepts; Autonomous systems and operations; Air Traffic Control; Markov decision process.

I. Introduction

The Aircraft Owners and Pilots Association (AOPA) publishes a yearly report that summarizes general aviation (GA) accident trends and factors in the U.S. In 2000, the report highlighted the fact that most mid-air collisions happen within 10 miles of airports. Of the collisions that occurred in the traffic pattern, the majority were at non-towered airports [1]. Large aircraft benefit from the use of the Traffic Alert and Collision Avoidance System (TCAS), but the system was not designed for use for smaller aircraft and is prohibitively expensive. Prior research has explored ways to extend collision protection to GA aircraft [2]. Papers have presented modifications of both TCAS [3], and its successor, ACAS X [4], for use in GA aircraft. Research has largely focused on on-board systems for detection, alert, and resolution, but solutions that require new equipment are not likely to be adopted. Although the availability of Automatic Dependent Surveillance-Broadcast (ADS-B) will make collision avoidance more tractable for GA aircraft [5], there is already resistance from the GA community due to privacy concerns and the high costs involved [6].

A ground-based surveillance concept would eliminate the need for specialized equipment on GA aircraft. Radar systems that measure energy pulses reflected off aircraft are prohibitively expensive, but a system based on the TCAS surveillance sensors may be feasible [7]. The TCAS surveillance system can only detect replies from aircraft transponders, but many GA aircraft are equipped with a transponder [8]. Recently, Lincoln Laboratory has investigated a system with an array of ground-based TCAS antennas for use at class D (towered) airports and high density non-towered airports [9].

This paper proposes a concept for air traffic collision prevention in the vicinity of airports with a focus on GA aircraft. The idea is inspired by the Advanced Airspace Concept, which uplinks 4D trajectories to aircraft [10], and recent work on an automated system for communicating between unmanned aircraft and air traffic control (ATC) [11]. We envision an autonomous ATC system (auto-ATC) at a non-towered airport that is advisory in nature. The pilot would still be responsible for maintaining separation, but the system would be able to provide advice to participating aircraft. The system would issue high-level recommendations to aircraft tracked through ground based sensors in the immediate vicinity of an airport. The system would aim to reduce collision risk with minimal radio transmissions. The aircraft are expected to be flying according to a model within the airport pattern that can be influenced through recommendations over the radio.

In the following sections, we show how such a system can be modeled as a Markov decision process (MDP). The outline of the paper is as follows: Section II defines the notation for MDPs and how the problem of collision prevention in the traffic pattern can be formulated as one. Section III presents simulations results with sensitivity analysis to some of the parameters, and Section IV concludes with a summary and suggestions for future work.

II. Problem Formulation

MDPs model sequential problems where decisions need to be made under uncertainty. They have been successfully used in the context of aircraft collision avoidance [4], [12]. In this section, we briefly introduce the concept of MDPs and explain how they can be applied to developing the decision-making component of the auto-ATC system. Because our goal is to investigate this new concept, we make a few simplifying assumptions in our modeling of the problem.

An MDP is defined by a state space $\mathcal{S}$, action space $\mathcal{A}$, transition function T, and reward function R [13]. If the current state of the process is $s \in \mathcal{S}$ and we execute action $a \in \mathcal{A}$, then the next state will be $s' \in \mathcal{S}$ with probability $T(s' \mid s, a)$. The reward associated with executing a from s is given by $R(s, a)$.

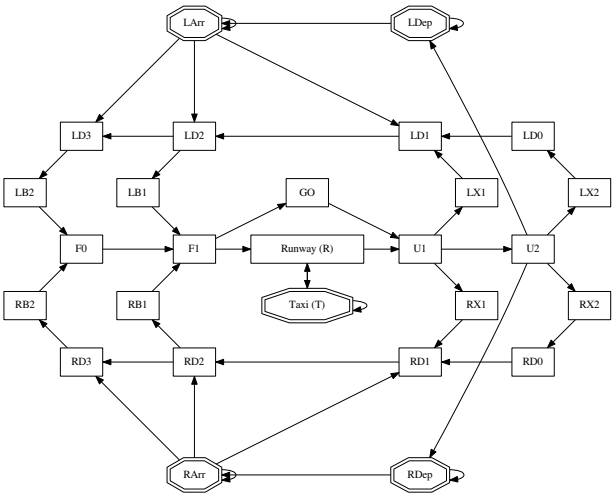


Fig. 1: Aircraft states in the pattern.

The goal in an MDP is to select actions in a way that maximizes the expected discounted reward:

$$\mathrm{E}[\sum_{k=0}^{\infty} \gamma^k r_k] \tag{1}$$

where r_k is the reward received in step k and $\gamma \in [0, 1)$ is a discount factor. The discount factor decays the value of rewards received in the future and is set to 0.95 in this paper. A decision making policy that is dependent on the current state is denoted π, and the action recommended at state s is denoted $\pi(s)$.

A. State Space

The state space $\mathcal{S}$ is composed of a discrete set of states specifying the locations of K aircraft in the traffic pattern. The experiments in this paper focus on $K = 4$. A particular state s is represented by a tuple $(s^{(1)}, \ldots, s^{(K)})$, where $s^{(i)} \in \{\ell_1, \ldots, \ell_n\}$ represents the location of the ith aircraft. In this formulation, there are a set of $n = 27$ possible locations, e.g., Taxi ($\ell_1 = \mathrm{T}$) and Runway ($\ell_2 = \mathrm{R}$). Hence, if there are K aircraft, then $|\mathcal{S}| = n^K$. The set of possible states that can immediately follow a particular location ℓ_i is denoted $\mathcal{N}(\ell_i)$. Figure 1 shows a representation of the various pattern locations along with possible transitions between them.

B. Action Space

The action space $\mathcal{A}$ is composed of a discrete set of actions specifying a particular aircraft and a location. A particular action $a = (a_i, a_l)$ involves commanding aircraft $a_i \in \{1, \ldots, K\}$ to transition immediately to location a_l. If no aircraft is to be addressed, we use $a = (0, \varnothing)$. The valid set of actions depends on the current state. We denote the set of actions available from state s as $A(s)$. For example, if $s = (\mathrm{R}, \mathrm{U2})$, then $A(s) = \{(0, \varnothing), (1, \mathrm{T}), (1, \mathrm{U1}), (2, \mathrm{LX2}), (2, \mathrm{RX2})\}$. In this formulation, it is not possible to request an aircraft to depart the pattern.

C. Transition Function

The transition function specifies the probability of transitioning to some next state given the current sate and action taken. The probabilities governing the aircraft transitions are independent from each other:

$$T(s' \mid s, a) = \prod_{i=1}^{K} T(s'^{(i)} \mid s^{(i)}, a) \tag{2}$$

The transition model assumes the following:

- If an aircraft is not being addressed by an action, all of the possible next states are equally likely with probability $1/|\mathcal{N}(s^{(i)})|$.
- If the aircraft is being addressed, the commanded state a_l‘ is selected with probability α (a cooperation factor), while other possible states are selected with probability $(1 - \alpha)/(|\mathcal{N}(s^{(i)})| - 1)$
- A pilot will fly the pattern without detailed instructions from the tower, but will generally take the runway only when instructed by the system: $T(\mathrm{R} \mid \mathrm{T}, (0, \varnothing)) = 1 - \alpha$.

D. Reward Functions

The reward function is designed to increase aircraft separation while minimizing intervention. We make the following assumptions:

- The rewards are additive over states and actions, i.e. $R(s, a) = R(s) + R(a)$
- Two aircraft occupying the same state result in a cost C_c, unless they are in one of the states considered safe (Taxi, Departures, and Arrivals). Each aircraft in the T state incurs a cost $C_t < C_c$ to avoid the system grounding all aircraft.
- There is a cost C_a for issuing an advisory.

In this paper, we set the collision cost $C_c = 1000$ and taxi cost $C_t = 10$. The experiments vary the advisory-cost ratio $\beta = C_a/C_c$. The reward function can be written by defining $\tilde{s} = (s^{(i)} \in s : s^{(i)} \notin \{\mathrm{T}, \mathrm{LDep}, \mathrm{RDep}, \mathrm{LArr}, \mathrm{Rarr}\})$:

$$\begin{aligned} R(s, a) = &- C_c(|\tilde{s}| - |\mathrm{unique}(\tilde{s})|) \\ &- C_t|(s^{(i)} \in s : s^{(i)} = \mathrm{T})| \\ &- \begin{cases} 0 & \text{if } a = (0, \varnothing) \\ C_a & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

where we use the list comprehension notation $(x \in X : F(x))$ to mean the list of all elements in X that satisfy the logical expression $F(x)$. The notation $|y|$ denotes the number of elements in the list y.

E. Solution Approach

There are different approaches for finding the optimal policy, but this paper uses a dynamic programming algorithm known as Gauss-Seidel value iteration [13]. We define a value function U that assigns 0 to all states. We then iterate through

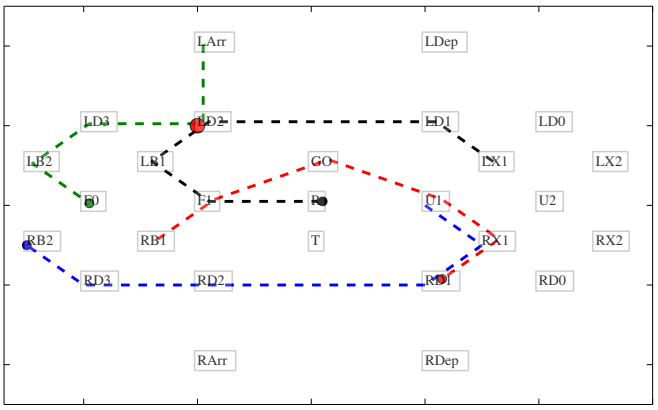


Fig. 2: Sample simulation with one collision at RD2.

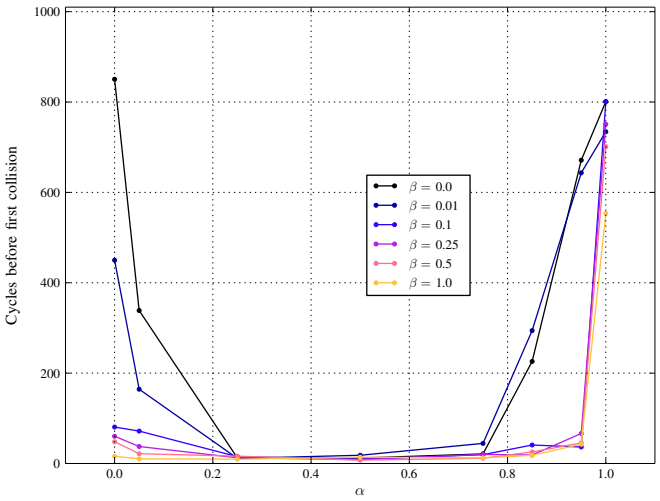


Fig. 3: Number of steps to first collision.

all the states, updating the value function as we go along according to

$$U(s) \leftarrow \max_{a \in A(s)} \left[R(s,a) + \gamma \sum_{s' \in \mathcal{S}} T(s' \mid s,a) U(s') \right] \quad (4)$$

Gauss-Seidel value iteration sweeps over the states repeatedly until convergence. Once converged, an optimal policy π^* can be extracted from U as follows:

$$\pi^*(s) = \arg\max_{a \in A(s)} \left[R(s,a) + \gamma \sum_{s' \in \mathcal{S}} T(s' \mid s,a) U(s') \right] \quad (5)$$

The complexity of the algorithm is polynomial in the state space size, which is exponential in the number of aircraft. However, the transition probabilities are sparse, and we can leverage the structure of the problem to achieve computationally tractable solutions. With $K = 4$, there are over 500,000 states and 100 actions, yet an optimal solution can be computed in six minutes using a single thread on a 1.9 GHz Intel i7 CPU.

III. Results

This section presents results of the MDP solution and their sensitivity to the cooperation factor α and cost fraction β. We then introduce a 3D simulation for the aircraft in the pattern and present results that show how the auto-ATC leads to a reduction in collision risk.

A. Sensitivity analysis

We initially simulate the performance of the policies using the MDP model. Each aircraft is initialized at random positions in the pattern. The states are then simulated forward K steps using the optimal policy and transition probabilities. Figure 2 shows an example of such a simulation with $K = 5$.

We can use these simulations to determine the sensitivity of the performance to some of the parameters. For each pair (α, β), we ran 20 simulations with 1000 time steps. Fig. 3 reports the average time before the first collision is encountered, while Fig. 4 shows the expected number of collisions per 1000 steps with error bars representing the standard errors.

The first thing we notice is that as β increases, so does the number of collision. This relationship is expected since larger β discourage the auto-ATC from issuing commands. As $\beta \to \infty$, we expect the auto-ATC to always issue $(0, \varnothing)$.

A surprising result is the fact that as the cooperation factor $\alpha \to 0$, the performance mirrors that of $\alpha \to 1$ for small values of β. At first glance this is counter-intuitive; we expect that non-cooperating aircraft would result in the worst performance. However, it turns out that for many states there are only two possible states at the next cycle. If $\alpha = 0$ and we would like the aircraft to transition to one of the states to avoid a collision, we should advise it to transition to the other state to trick the non-cooperating pilot. In practice, this would not be a desirable policy, but it is an interesting corner case that arises given the modeling assumptions.

Performance is worst when $\alpha = 0.5$. If we consider the case where there are only two states at the next time step, issuing an advisory results in the same transition probabilities as not issuing an advisory. If issuing advisories has a non-zero cost, we expect the optimal policy to never issue any advisories in that scenario. Indeed, as we let $\beta \to 1$, we trade-off the mean number of collisions with the intrusiveness of the auto-ATC. We associated a cost with this when designing the reward function, but we can also look at the resulting policy and define a verbosity index: $|\{s : \pi(s) \neq (0, \varnothing)\}|/|\mathcal{S}|$ which is the fraction of states where the system issues an advisory.

Figure 5 gives the verbosity index for different values of β and the associated mean collisions per 1000 steps. For cooperation factors that result in situations where advisories have little effect on the outcome, the optimal policy is silent and has zero verbosity. We also see that even if we have no advisory cost ($\beta \to 0$), the verbosity $\to 0.8$, instead of 1, because there are always some states where no commands are issued since there is only one possible transition. For the case of $\alpha \to 1$, we can cut the verbosity by a factor of 2 if we let the number of collisions increase by an order of magnitude.

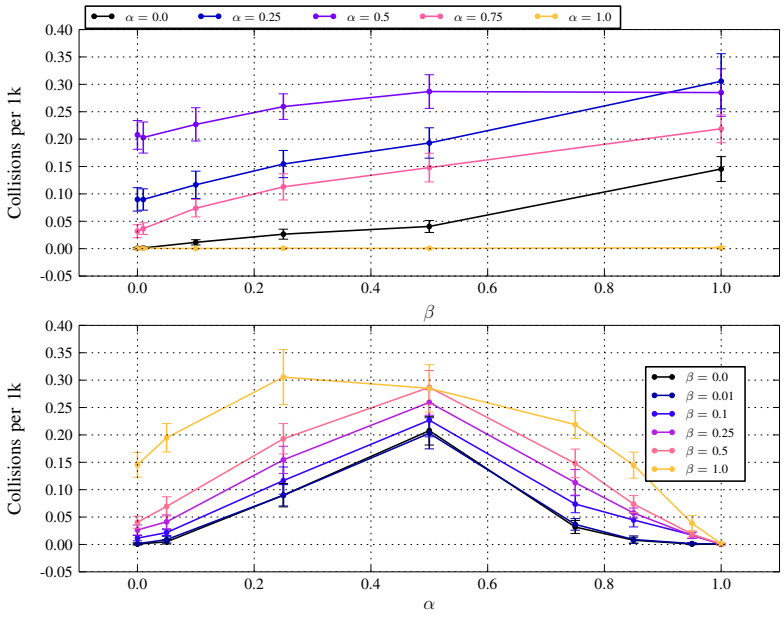


Fig. 4: Number of collisions per 1000 steps.

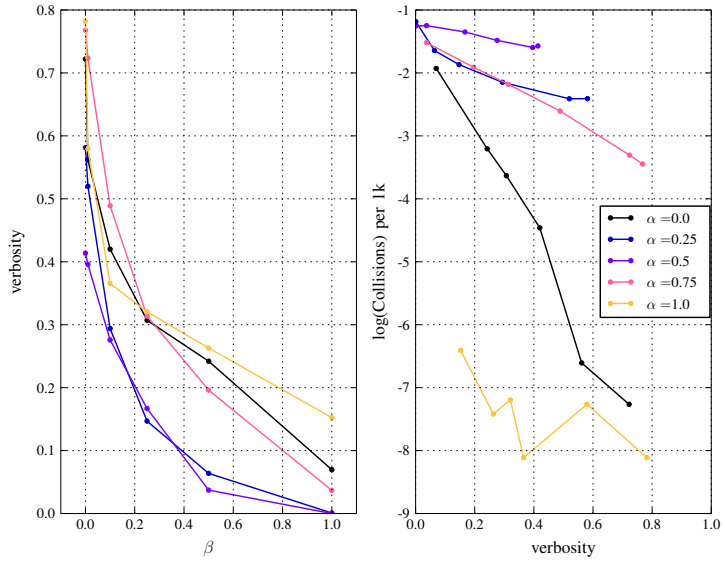


Fig. 5: Policy verbosity.

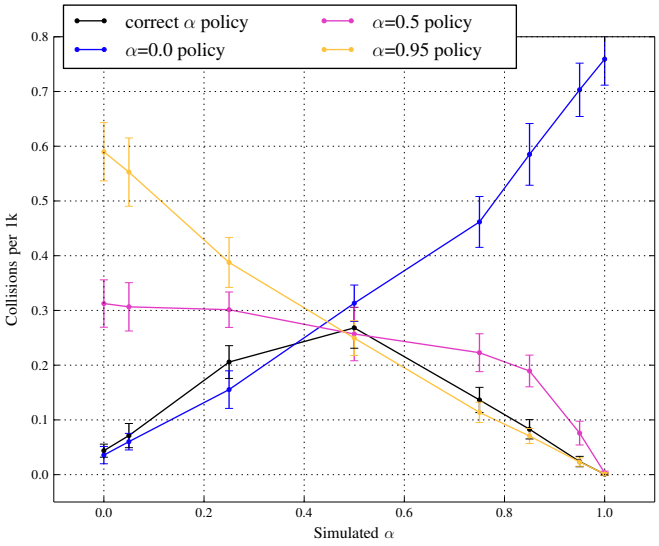


Fig. 6: Collision sensitivity to incorrect α, $\beta = 0.5$.

All the results we have presented so far involved evaluating policies in simulation using the same parameters that were used for optimization. We will refer to the optimal policy obtained from using values for α and β as $\pi_{\alpha,\beta}$. Differences between the α used for generating the policy and the α used for evaluation can result in degraded performance. The cost fraction β affects the policy, but it does not affect the simulation of policies since it does not appear in the transition function. In order to evaluate the sensitivity of the resulting policies to the inaccuracy in α, we compute the number of collisions per 1000 steps for the policies $\pi_{0,0.5}$, $\pi_{0.5,0.5}$, and $\pi_{0.95,0.5}$ for different true values of α. Figure 6 shows the results when compared to the policies that have been optimized using the correct α. We can see that $\pi_{0.95,0.5}$ yields similar performance to the policies with the correct α so long as α is approximately greater than 0.5, and likewise for $\pi_{0,0.5}$ so long as α is approximately less than 0.5. The plot indicates that the performance of the policy in terms of collisions is not overly sensitive to knowing the correct α.

B. 3D Simulation of the Pattern

The analysis presented so far has used the simple transition model from Section II. Although using this model is useful for understanding the problem conceptually, it is not very realistic when it comes to properly evaluating the performance of the algorithm. In practice, aircraft in the pattern do not all move to the next states at the same time, and the amount of time they spend in each leg of the pattern depends on several factors. Therefore, we developed a higher fidelity 3D aircraft model that would capture some of these factors. We describe the details and modeling assumptions we made to construct this model.

Each aircraft is parametrized with the following states:

- $x = [x_N, x_E, x_D]^\top$, aircraft position in north-east-down world coordinates,
- V, aircraft airspeed, which is assumed to be along the aircraft longitudinal axis, and
- ψ, aircraft heading in world coordinates.

Additionally, we assume that the pilot perfectly regulates the aircraft roll ϕ and its glide path angle γ. The resulting equations of motion are:

$$\dot{x} = V \begin{bmatrix} \cos\psi \\ \sin\psi \\ -\sin\gamma \end{bmatrix}, \dot{\psi} = \frac{g\tan\phi}{V} \tag{6}$$

The equations are integrated using the Euler method with a time step of $dt = 0.25\,\text{s}$. This model does not account for any wind effects, assumes that the pilot is maintaining coordinated flight, and neglects any dynamics associated with achieving the necessary roll and glide path angles. These are reasonable assumptions for this model as we are not concerned with the details of the flight dynamics, but rather with the motion of the aircraft in the 3D world coordinates.

In order to make the aircraft fly around the pattern, we implemented a two-layer logic controller to emulate a pilot:

- *Navigate.* The pilot flies towards a waypoint in the world coordinates by setting a desired bearing ψ_d and altitude h_d.
- *Aviate.* The pilot regulates the aircraft at h_d by commanding γ and steers towards ψ_d by commanding ϕ. We impose a maximum achievable climb rate $\gamma_{\max} = 5°$ and minimum turn radius $\phi_{\max} = 45°$. This control is implemented as two simple proportional controllers with gains $k_{p,h_{err}} = 0.1\,°/\text{m}$ and $k_{p,\psi_{err}} = 3$.

We define target spatial waypoints for each of the locations in the pattern shown in Fig. 1 with the runway fixed at $(0, 0, 0)$. The simulation assumes that the pilot flies from their current waypoint to their next waypoint, and whenever the aircraft is within 50 m of their destination waypoint, the pilot chooses the next leg he will be flying according to the MDP model introduced previously. A random position error with Gaussian distribution $\mathcal{N}(0, (150\,\text{m})^2)$ horizontally and $\mathcal{N}(0, (20\,\text{m})^2)$ vertically is added to the east-north-down coordinates of the destination waypoint.

A sample of the resulting trajectories can be seen in Fig. 7. Some states require special handling, in particular:

- *Taxi.* The aircraft speed is reduced to 5 m/s $\approx$ 10 knots and the equations of motion are modified to no longer assume coordinated flight. Instead, we assume that the pilot is able to directly steer the aircraft to the desired heading.
- *Runway.* When the aircraft is on the runway, it accelerates to either its flight or taxi speed depending on whether it is landing or taking-off. The flight speed is assumed to be uniformly distributed between 43 m/s and 49 m/s $\approx$ 83 knots and 95 knots.
- *Final.* No navigation noise is added when the aircraft is on the final leg, and the altitude gains for the pilot controller model are increased by a factor of 10 to ensure that the aircraft touches down on the runway.
- *Departure state.* When an aircraft is in the departure state and transitions into departure again, we generate a random trajectory that takes 30 seconds to execute and remains outside of the pattern.

Unlike the MDP case, not all aircraft transition states at the same time. In order to incorporate the ATC commands into the 3D simulation, an approximation must be made. We explored two different schemes to select when to issue an ATC command: event-driven and periodic. In the event-driven scheme, we issue a command whenever any of the aircraft reaches its destination waypoint and is about to transition. However, this scheme may lead to advisories being issued very close together, which would be problematic in practice. In the second scheme, we simply issue commands periodically (every 10 s). In both instances, we only issue a new command after the addressed aircraft has acted on the previous command.

We use as a baseline a silent policy which issues no advisories. A consequence of this policy is that the aircraft eventually end in the taxi state. Hence, the silent policy has an artificial advantage over the auto-ATC policy in long-running simulations because collisions are not counted in the taxi state.

In the MDP simulation, the aircraft have discrete states, and we define collisions as two aircraft occupying the same state. However, in the 3D simulation, the aircraft have continuous states and they might never exactly collide. We define near mid-air collision (NMAC) events as when two aircraft come within 150 m $\approx$ 500 ft horizontally or 30 m $\approx$ 100 ft vertically of each other. To mirror the MDP formulation, we ignore events between aircraft that are in the taxi, departure, or arrival states. The 3D simulation can result in transitions that we normally would not model in the MDP simulation. In the 3D simulation, it is possible for one aircraft to move to the next leg while the other one is still flying the same leg. This was not modeled in the MDP since all aircraft changed states in the same cycle. This could be addressed by using a continuous-time MDP formulation [14].

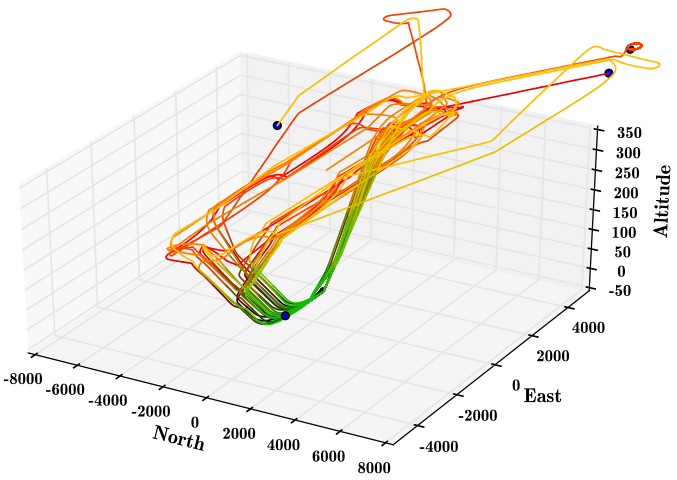


Fig. 7: 3D simulation of four aircraft in the pattern.

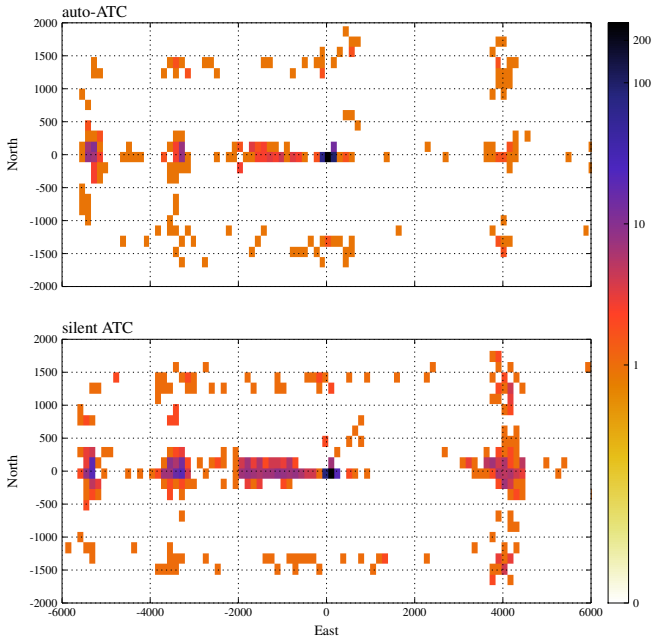


Fig. 8: Position distribution of NMAC events.

C. 3D Simulation Results

Similar to the MDP case, we present the results from running Monte Carlo simulations. For each policy, a total of 1000 cases are initialized with four aircraft in a non-NMAC configuration at random locations in the pattern and with random airspeeds distributed with a mean of 46 m/s $\approx$ 90 knots. The states are then simulated forward until either an NMAC event occurs or 20 hours is elapsed. We use $\alpha = 0.95$ for the simulations.

Figure 8 shows the location of all the NMACs as a heat map on the pattern for both the event driven auto-ATC with $\beta = 0.5$ and the silent ATC. The majority of events occur when the aircraft are turning from base to final and on the runway. The other hotspots are other convergence points such as when aircraft are arriving in the pattern or cutting the base turn when following an upwind aircraft. These observations are consistent with the analysis of actual NMAC events [5] and is not surprising given our modeling assumptions. The NMACs between these hotspots (e.g., on the downwind leg) are due to faster aircraft overtaking slower aircraft. Because the simulation horizon is relatively long, both auto-ATC and silent ATC have roughly the same number of NMAC events (900 and 1000 respectively). However, while the number of events are similar and the 2D distribution of the events look similar, there is a difference in how long it takes for each event to occur.

Figure 9 shows the inverse cumulative density function (CDF) of the time to first NMAC for different policies. We can see that the difference between the event-driven and the periodic policies is not very significant. Although it performed well on the MDP transition model, the $\beta = 0.01$ auto-ATC policy performs poorly in the 3D simulation and is worse than the silent ATC. However, $\beta = 0.5$ and $\beta = 1.0$ policies perform better than the silent ATC. This is counter to what we observed in the MDP simulations where lower β values (i.e., more verbose policies) lead to less collisions.

The difference in performance between the MDP model and the 3D simulation can be explained by the modeling errors between the two formulations. As a reminder, while the silent ATC allows aircraft to remain in the safe taxi state, the reward function for auto-ATC was chosen to encourage aircraft to leave the taxi state. Since the policy obtained with a low β does not associate a cost to issuing ATC commands, it is more likely to encourage aircraft to take the runway. In the MDP simulation, the transitions follow the model assumed by the policy, and it is therefore able to avoid most collisions. However, when evaluated in 3D, the system does not prevent NMACs as well and the trade-off between taxi and NMAC is different. In contrast, the policies with larger β do not encourage aircraft to take the runway, but they also do not encourage them to remain in taxi as discussed later. These policies still however intervene in flight states, and even though they do not prevent all of the NMACs, they perform better than the silent ATC.

A policy can increase the time to NMAC by either preventing collisions or grounding the aircraft. To verify that aircraft are not being grounded, we show in Fig. 10a the distribution of flight hours obtained from the simulations. We define flight hours as the total time spent in the air, i.e., not in taxi. The auto-ATC policies with $\beta = 0.5$ and $\beta = 1.0$ are indeed achieving more flight hours than the silent ATC, which means the auto-ATC policies are delaying the onset of NMAC, not grounding the aircraft in the safe taxi state. This subtle behavior is an artifact of our modeling assumptions regarding the transitions between the runway and the taxi. These transitions deserve more attention in future work, including a sensitivity analysis of taxi cost relative to collision and alert costs.

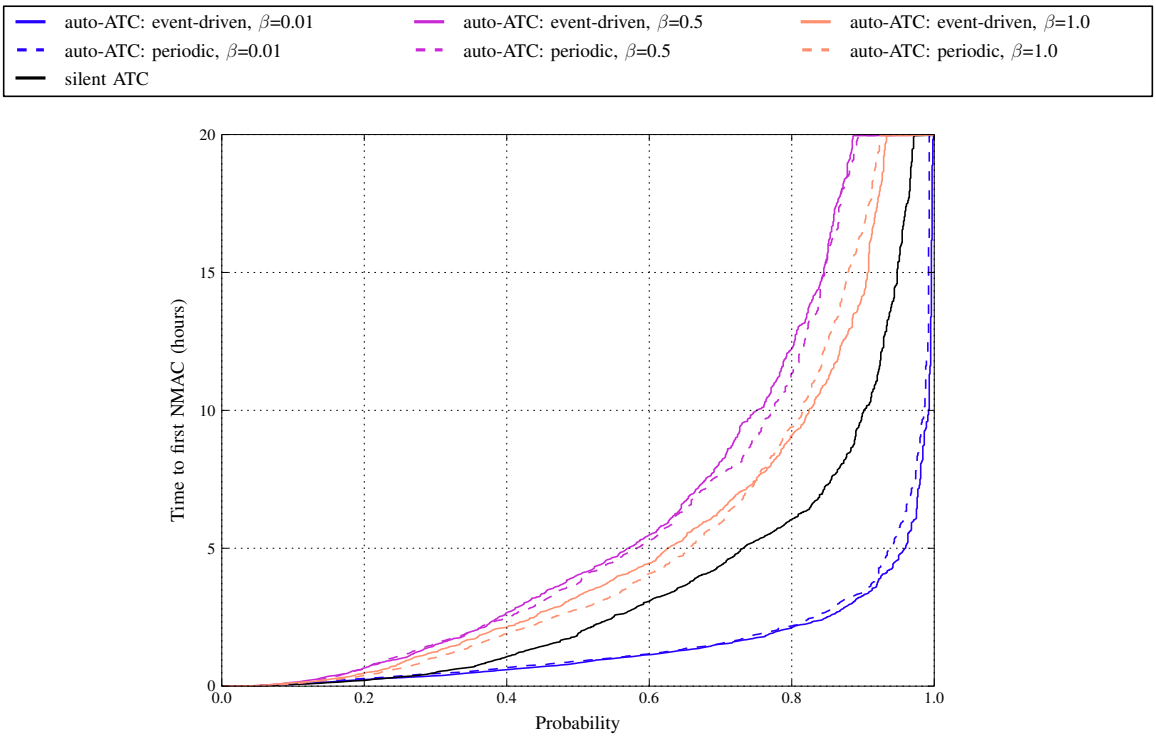
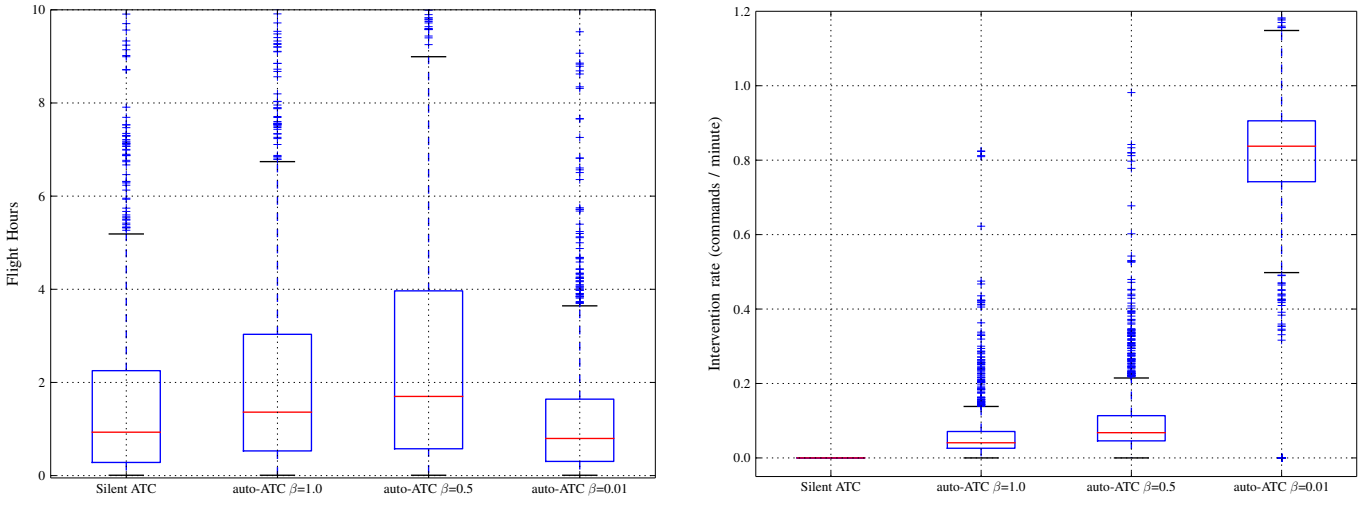


Fig. 9: Inverse CDF for NMAC events.



(a) Flight hours.

(b) Alert rate.

Fig. 10: 3D simulation results.

In the MDP formulation we defined the verbosity index for each policy. In the 3D simulation, we keep track of the intervention rate, the number of ATC commands issued per minute. Figure 10b shows the intervention rate for the policies under consideration. As expected, the silent ATC does n ot issue any commands, while policies with smaller values for β have larger intervention rates. The rates are on the order of one command per minute, which seems acceptable given that there are four aircraft in the pattern.

Given all the modeling errors and approximations that had to be made to go from the MDP formulation to the 3D simulation, the fact that we decrease the time before NMAC events occur while increasing the flight hours is an encouraging result.

We are primarily interested in the performance of the policies relative to each other. The actual probability of NMAC events per hour obtained in these simulations are not an accurate representation of reality. We expect a higher level of safety as pilots' ability to see and avoid should result in a lower probability of NMAC events.

All simulation and analysis scripts that were used to generate the experimental results are under version control and are available online at https://github.com/sisl/autoATC.git.

IV. Conclusions and Further Work

We introduced a concept for an autonomous ATC that could help reduce the risk of air traffic collisions in the vicinity of non-towered airports. We showed how the system can be posed as an MDP by defining the states, actions, transitions, and rewards. Value iteration was used to compute optimal policies for a range of cooperation levels and cost parameters. We assessed performance in terms of safety and the verbosity of the system using Monte Carlo simulation. In a 3D simulation of a traffic pattern, we found a system operating point that reduces the probability of near mid-air collision with an acceptable intervention rate compared to a silent system.

Although the results are promising, this first formulation makes many simplifying modeling assumptions. First of all, the number of states in the pattern and available actions should be extended. For example, if one aircraft is on left base and another is on right base, they are on a collision course but there are no actions in our formulation to prevent collision. A possible extension would be to add commands for S turns. In addition, our model has the taxi state acting as a sink. To enhance realism, it is necessary to incorporate a better model of aircraft behavior when transitioning between the runway and taxi states.

In real life, the transitions do not all occur at the same time as modeled by the MDP and the actions need to be taken at potentially non-uniform time steps. This could be accounted for by modeling the dynamics using continuous-time Bayesian networks (CTBN). The decision making can then be posed as a CTMDP [14].

A major assumption made in this paper was that the positions of the aircraft are exactly known by the system. In practice, we will need to estimate the location of each aircraft from the ground sensors. Doing so would require an observation model and applying Bayes' rule to track a probability distribution over the aircraft positions. The problem can then be reformulated as a partially observable Markov decision process (POMDP) [15]. Additionally, the parameters for the 3D simulation (airspeeds, turning radius, controller gains, etc.) were chosen in this paper using engineering judgment. The parameters of a higher fidelity model could be derived from radar data.

We envision the system issuing commands over a Common Traffic Advisory Frequency. Hence, there needs to be a way to identify the aircraft in the pattern. One way to achieve this is to refer to aircraft by their position, transponder code, or their call-sign inferred through speech recognition [11]. Finally, a practical implementation would require the ability to handle special cases such as varying number of aircraft in the pattern, aircraft overflying the runway above the traffic pattern altitude, and change of runway direction due to shifting wind.

References

[1] AOPA Air Safety Foundation, "Nall Report: Accident Trends and Factors for 1999," *AOPA Air Safety Foundation*, 2000.

[2] D. Diefes and R. Deering, "Concept Design for a Low Cost Cockpit Display/Collision Avoidance System for General Aviation Aircraft," in *IEEE Position, Location and Navigation Symposium (PLANS)*, 1996.

[3] C. E. Lin and Y.-Y. Wu, "TCAS Solution for Low Altitude Flights," in *Integrated Communications Navigation and Surveillance Conference (ICNS)*, 2010.

[4] T. B. Billingsley, M. J. Kochenderfer, and J. P. Chryssanthacopoulos, "Collision Avoidance for General Aviation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 27, no. 7, pp. 4–12, 2012.

[5] F. Kunzi and R. J. Hansman, "Mid-Air Collision Risk And Areas Of High Benefit For Traffic Alerting," in *AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, 2011.

[6] AOPA, EAA, *et al.*, *Open Letter to Federal Aviation Administration*, 2015. [Online]. Available: download.aopa.org/advocacy/150122-Huerta-ADS-B-FINAL.pdf.

[7] E. J. Shea, "Air Surveillance for Smart Landing Facilities in the Small Aircraft Transportation System," Master's thesis, Virginia Polytechnic Institute and State University, 2002.

[8] FAA, *General Aviation and Part 135 Activity Surveys CY 2012*, 2012. [Online]. Available: www.faa.gov/data_research/aviation_data_statistics/general_aviation/CY2012/.

[9] S. D. Campbell, "Small Airport Surveillance Sensor," in *MIT Lincoln Laboratory ATC Workshop*, 2014.

[10] T. Nikoleris, H. Erzberger, R. A. Paielli, and Y.-C. Chu, "Autonomous System for Air Traffic Control in Terminal Airspace," in *AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, 2014.

[11] S. Gunawardana, "A Rule-Based Dialog Management System for Integration of Unmanned Aerial Systems into the National Airspace System," PhD thesis, Stanford, 2012.

[12] J. E. Holland, M. J. Kochenderfer, and W. A. Olson, "Optimizing the Next Generation Collision Avoidance System for Safe, Suitable, and Acceptable Operational Performance," *Air Traffic Control Quarterly*, vol. 21, no. 3, pp. 275–297, 2013.

[13] M. J. Kochenderfer, *Decision Making Under Uncertainty: Theory and Application*. Cambridge, MA: MIT Press, 2015.

[14] K. F. Kan and C. R. Shelton, "Solving Structured Continuous-Time Markov Decision Processes," in *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2008.

[15] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains," *Artificial Intelligence*, vol. 101, no. 1–2, pp. 99–134, 1998.

AUTHOR BIOGRAPHIES

Zouhair Mahboubi is a graduate student in Aeronautics and Astronautics at Stanford University. He received his B.Eng. in Mechanical Engineering from McGill University in 2008, and his M.S. in Aeronautics at Stanford University in 2010. He worked as a Guidance, Navigation, and Controls Engineer at Zee.Aero (an aerospace startup in Silicon Valley) for four years before returning to Stanford to pursue his Ph.D.

Mykel Kochenderfer is Assistant Professor of Aeronautics and Astronautics at Stanford University. Prior to joining the faculty in 2013, he was at MIT Lincoln Laboratory where he worked on airspace modeling and aircraft collision avoidance. He received his Ph.D. from the University of Edinburgh in 2006. He received B.S. and M.S. degrees in computer science from Stanford University in 2003.