Multi-Agent Autonomous Operations in Urban Air Mobility with Communication Constraints

Xuxi Yang*, Lisen Deng†, Jia Liu‡, and Peng Wei§
Iowa State University, Ames, Iowa, 50011

Husheng Li¶
University of Tennessee, Knoxville, Tennessee, 37996

Urban Air Mobility (UAM) is an emerging transportation mode, where electrical vertical takeoff and landing (eVTOL) aircraft will transport cargo and passengers within a city. In our previous work, we have introduced computational guidance algorithms for single aircraft and multiple cooperative aircraft to navigate through obstacles and avoid conflicts among aircraft. However, we assumed perfect communications for air-to-air and ground-to-air channels. In this paper, we formulate the computational guidance problem with communication constraints and communication loss. Specifically, our goal in this paper is to (1) modify our computational guidance algorithms given communication constraints, e.g., time, bandwidth, and communication loss; (2) design air-to-air and air-to-ground communication frameworks to facilitate the computational guidance algorithm; and (3) integrate the decision making and communication mechanisms to guide all the aircraft to their respective destinations while avoiding potential conflicts between them. A free-flight airspace simulator in OpenAI Gym environment is used to test the performance of the proposed algorithm.

I. Introduction

A. Motivation

A growing community of interest is forming for the concept of Urban Air Mobility (UAM) including NASA, Uber, Airbus and many other entities around the globe [1–6]. Companies include Airbus, Bell, Embraer, Joby, Zee Aero, Pipistrel, Volocopter, and Aurora Flight Sciences have been working to build and test electric vertical takeoff and landing (eVTOL) aircraft. In UAM, the eVTOL aircraft may be either human piloted or autonomous for passenger transport in personal commute or on-demand air taxi [7–10], which is expected to fundamentally change cities and people's lives to reduce commute time and stress. Some of the barriers for enabling UAM operations include vehicle systems certification, noise impacts from vehicle operations in urban zones, battery technology, communication constraints, social acceptance, and cybersecurity protections [11]. In this paper, we focus on overcoming one specific technical barrier: creating a safe and robust onboard guidance and collision avoidance system for eVTOLs with communication constraints.

In previous work, we have presented computational guidance algorithms for single aircraft and multiple cooperative aircraft [12, 13] by assuming the perfect communications for vehicle-to-vehicle and ground-to-vehicle channels. However, air-to-air and air-to-ground communications will be affected by many factors [14, 15] including bandwidth constraint and communication loss. In this paper we consider these communication constraints in the design of computational guidance algorithms to achieve multi-agent autonomous, safe and efficient UAM operations.

Specifically, our goal is to (1) improve our existing computational guidance algorithms given communication constraints; (2) design air-to-air and air-to-ground communication frameworks to facilitate the computational guidance algorithm; and (3) integrate the decision making and communication mechanisms to guide all the aircraft to their respective destinations while avoiding potential conflicts between them. In this work, the aircraft dynamics are modeled

*Graduate Research Assistant, Department of Aerospace Engineering, xuxiyang@iastate.edu. Student Member AIAA.
†Department of Aerospace Engineering, ldeng@iastate.edu.
‡Assistant Professor, Department of Computer Science, jialiu@iastate.edu.
§Assistant Professor, Department of Aerospace Engineering, pwei@iastate.edu. Senior Member AIAA.
¶Associate Professor, Department of Electrical Engineering and Computer Science, hli31@utk.edu.

based on the tandem tilt-wing eVTOL (Airbus Vahana) from Airbus A^3 [16], as shown in Fig. 1. A free-flight airspace simulator in OpenAI Gym environment will be used to validate the performance of the proposed algorithm.

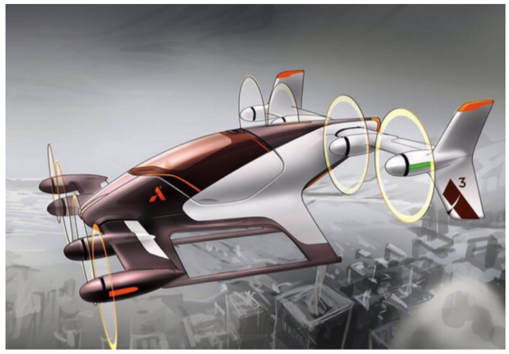


Fig. 1 Airbus Vahana with tandem tilt-wing configuration during the cruise phase [17].

B. Related Work

There have been many important contributions to the topic of guidance algorithm with collision avoidance capability for small unmanned aerial aircraft. And these approaches can be roughly categorized into centralized algorithms and decentralized algorithms.

In centralized methods, the conflicts between aircraft are resolved by a central supervising controller. Under such scenario, the state of each aircraft, the obstacle information, the trajectory constraint as well as the terminal condition is known to the central controller, and the central controller in return designs the individual whole trajectory for all aircraft before the flight, typically by formulating it to an optimal control problem. These methods can be based on semidefinite programming [18], nonlinear programming [19, 20], mixed integer linear programming [21–24], mixed integer quadratic programming [25], sequential convex programming [26, 27], second-order cone programming [28], evolutionary techniques [29, 30], and particle swarm optimization [31]. These centralized methods often pursue the global optimality of the solution. However, as the number of aircraft grows, the computation time of these methods typically scales exponentially. Moreover, these centralized planning approaches typically need to be re-run, as new information in the environment is updated (e.g., a new aircraft enters the airspace).

On the other hand, decentralized methods scale better with respect to the number of agents and are more robust since they do not possess a single point of failure [32]. In decentralized methods, all the conflicts are resolved by each aircraft individually. Researchers have proposed several algorithms under the case where the communication between aircraft can be successfully established [33]. Algorithms in [34, 35] are based on message-passing schemes, which resolve local (e.g., pairwise) conflicts between all members of the team. In [36], every agent is allotted a time slot in which to compute a dynamically feasible and guaranteed collision-free path using MILP. In [37], the author recast the global optimization problem as several local problems, which are then iteratively solved by the agents in a decentralized way. In Decentralized Model Predictive Control approach [38], the aircraft solve their own sub-problem one after the other and send the action to other subsystems through communication.

The proposed algorithm in this paper is built upon our previous work [12, 13], where the computational guidance problem is formulated as a Markov Decision Process (MDP) problem and solved using the Monte Carlo Tree Search (MCTS) algorithm. Comparing with previous work where we assumed perfect communications in air-to-air and ground-to-air channels, the algorithm proposed in this paper will consider practical communication constraints including latency, communication loss, and bandwidth constraint.

The structure of the paper is as follows: in Section II, the background and formulation of MDP, MCTS will be introduced. In Section III, we describe the practical consideration for communication latency, loss, and bandwidth, as well as their impact on the previously proposed computational guidance algorithms. Section IV presents the solution methods with communication constraints. The numerical experiments and results are described in Section V.

II. Problem Formulation

In this section, we briefly introduce the Markov Decision Process Formulation and Monte Carlo Tree Search algorithm.

A. Markov Decision Process (MDP)

Since the 1950s, MDPs [39] have been well studied and applied to a wide area of disciplines, including robotics, automatic control, economics, and manufacturing. In a MDP, the agent may choose any action a that is available based on current state s at each time step. The process responds at the next time step by moving into a new state s' according to the transition probability and given the agent a corresponding reward r.

More precisely, the Markov Decision Process (MDP) includes the following components:

- The state space $\mathcal{S}$ which consists of all the possible states.
- The action space $\mathcal{A}$ which consists of all the actions that the agent can take.
- Transition function $\mathcal{T}(s_{t+1}|s_t, a_t)$ which describes the probability of arriving at state s_{t+1}, given the current state s_t and action a_t.
- The reward function $\mathcal{R}(s_t, a_t, s_{t+1})$ which decides how much reward the agent will collect after a transition s_t, a_t, s_{t+1}. The reward function may only depend on the current state s_t, which will be the case in this paper.

In a MDP problem, a policy π is a mapping from the state to one specific action (known as deterministic policy)

$$\pi : \mathcal{S} \to \mathcal{A}$$

The goal of MDP is to find an optimal policy π^* that, if followed from any initial state, maximizes the expected cumulative rewards:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \mathbb{E}[\sum_{t=0}^{T} R(s_t, a_t)|\pi]$$

B. Mathematical Formulation

Our formulation focuses on horizontal conflict resolution with the aircraft assumed to be co-altitude. Here we assume perfect communications between aircraft and the communication constraint will be introduced in next section. The mathematical formulation of this problem is similar to previous work [12], which includes four components: state space, action space, reward function, and the transition model. We briefly introduce these components in the following.

1. Resolution Advisories

At each time step (5 seconds), the aircraft can choose to change its heading at a certain rate and the advisory of heading angle for each aircraft constitutes the action set $\mathcal{A} = \{-5°/s, 0°/s, +5°/s\}$ where positive corresponds to left turn and negative to right turn. The changing rate of heading angle is determined assuming the aircraft is flying with cruise speed at $190km/h$ [40] with banking angle smaller 25° (the banking angle limit is chosen to be 25° for the passenger comfort consideration).

2. Dynamical Model

The state includes the position, velocity, speed, heading angle, and goal position for all of the aircraft, which is a matrix with each row containing information for one aircraft: $(x, y, v_x, v_y, v, \psi, g_x, g_y)$. Based on the current state and action, Dubin's kinematic model will be used to compute state transition for each aircraft:

$$\dot{x} = v \cos \psi \tag{1}$$

$$\dot{y} = v \sin \psi \tag{2}$$

$$\dot{\psi} = a_\psi \tag{3}$$

where v is the cruise speed, ψ is the heading angle, and a_ψ is the selected action describing the changing rate of heading angle for one aircraft.

After an aircraft executes an advisory, the aircraft speed is held constant between decision stages, and is modeled as a Gaussian distribution centered on the aircraft's cruise speed with a standard deviation of $5m/s$. The changing rate of heading angle distribution is centered on the resolution advisory with a standard deviation of 2°. The noises here aim to account for the uncertainties in the environment and aircraft dynamics.

3. Reward function

For the consideration of safety, the conflict is defined to be when the distance of two aircraft is less than a minimum separation distance $r^{min} = 0.5$ nautical miles [11]. This separation standard was chosen using the definition of well clear for Unmanned Aircraft Systems (UAS) according to Cook and Brooks [41]. For large UAS in high-altitude airspace, the Horizontal Miss Distance (HMD) is defined to be 0.66nmi. Using this value as reference, the separation standard picked for this UAM application is set to 0.5nmi horizontally. This value is tighter than UAS standards because it is assumed that enhanced equipage capabilities will be installed on the eVTOL aircraft [11].

Based on the above separation requirements, we define a state as a conflict state is the distance between any two aircraft is less than r_{min}, and a state as a goal state if an aircraft reaches the goal position.

With the two types of states defined above, the reward function for one aircraft is defined as follows:

$$r(s) = \begin{cases} 1, & \text{if } s \text{ is goal state,} \\ 0, & \text{if } s \text{ is conflict state,} \\ 1 - \frac{d(o,g)}{\max d(o,g)}, & \text{otherwise.} \end{cases} \tag{4}$$

where $d(o, g)$ denotes the distance from the aircraft to its goal position, and $\max d(o, g)$ is the maximum distance from an aircraft to its goal position, which is the diagonal distance of the map. In this way, if an aircraft is not at conflict state (which has reward 0), this aircraft will get a positive reward between 0 and 1, depending on how far this aircraft is from the goal state.

C. Monte Carlo Tree Search (MCTS)

Monte Carlo Tree Search (MCTS) is a method for finding optimal decisions in a given domain by taking random samples in the decision space and building a search tree according to the results [42, 43]. It has already had a profound impact on Artificial Intelligence (AI) approaches for domains that can be represented as trees of sequential decisions, particularly games and planning problems [44–46], including the current state-of-art computer program AlphaZero in the Game of Go [47].

The basic MCTS process is conceptually very simple. A tree is built in an incremental and asymmetric manner. For each iteration of the algorithm, a tree policy is used to find the most urgent node of the current tree. The tree policy attempts to balance considerations of exploration (look in areas that have not been well sampled yet) and exploitation (look in areas which appear to be promising). A simulation is then rolled out from the selected node and the search tree updated according to the result. This involves the addition of a child node corresponding to the action taken from the selected node and an update of the statistics of its ancestors. Moves are made during this simulation according to some default policy, which in the simplest case is to make uniformly random moves. A great benefit of MCTS is that the values of intermediate states do not have to be evaluated, as for depth-limited minimax search, which greatly reduces the amount of domain knowledge required. Only the value of the terminal state at the end of each simulation is required.

In this paper, we use the most popular MCTS algorithm Upper Confidence Bound for Trees (UCT) to solve the formulated MDP. The details of UCT algorithm implementation can be found at [12]. In MCTS, we build a search tree for each aircraft and treat other aircraft as intruder aircraft with the same dynamical model in Equation 1. In the previous paper [13] it was shown the importance of the communication between aircraft and the communication constraint will be discussed in the next section.

III. Communication Constraints to Our Model

When the algorithm is running to issue actions for all the aircraft, the aircraft need to communicate information between each other (position, velocity, destination location, and intended action for next time step). In our previous work [13], we assume all the aircraft can communicate perfectly. However, in real world applications, there are many restrictions for the communication between aircraft [14].

A. Communication Latency

Delay is one of the major concerns in communication systems, which consists of propagation time (limited by light speed) and processing latency (e.g., the computational time overhead of demodulation and decoding). In typical wireless communication systems, the delay in point-to-point communications could be in the order of milliseconds. A significant communication delay could hinder the timely delivery of information and thus cause performance degradation in real-time control systems.

In this paper, we use previous work on aircraft wireless communication systems [15] to decide the communication time between two aircraft or between ground-based centralized controller and aircraft. According to [15], the onboard communication system can provide a full-duplex 2 Mbps/256 kbps (downlink/uplink) connection to a fixed ground station via satellite, where the uplink process will limit the communication time. In the proposed algorithm, the aircraft need to send its own state information including position, velocity, destination, and intended action, and the size of this information is smaller than 256 bits (each number takes 32 bits of memory). Thus the communication can finish under 1ms, which is negligible comparing with the computation time for the online guidance MCTS algorithm.

B. Communication Bandwidth

The data transmission rate is limited by the available bandwidth, namely the frequency band used for the communications. The broader, the faster. Usually bandwidth is expensive due to the congestion in the frequency bandwidth allocation, especially in the proposed computational guidance algorithm, where the information broadcast and communication will be used extensively. Therefore, there is a tradeoff between the bandwidth resource and cost, which means the aircraft needs to try decreasing the communication frequency and only sends the necessary information.

Usually an onboard communication device cannot transmit and receive in the same band at the same time due to high self-interference (the power of transmitted signals is orders of magnitude higher than that of the received signal). Besides, since the broadcast nature of wireless communications, different communications links in a wireless communication network cannot transmit simultaneously [48].

In the proposed algorithm in this paper, the communication between two aircraft includes two processes: (1) one aircraft receives information from all other aircraft; (2) one aircraft transmits the action information to all other aircraft. Due to the factors described above, in this paper we assume aircraft receives/transmits information from/to other aircraft one after another.

C. Communication Loss

During the aircraft en route flight, communication loss between two aircraft or between aircraft and ground-based centralized controller may happen [49], resulting from:

1) Communication equipment problems caused by malfunction or complete failure of aircraft or ground equipment (becoming less of an issue with improved system redundancy);
2) Radio interference where transmissions other than those from authorized users of an RTF frequency interfere with radio reception;
3) Blocked transmissions due to the tall buildings.

When a communication loss happens, the centralized controller will not be able to receive the state information from the aircraft lost communication, which will cause a higher probability of conflict for this aircraft when the centralized controller is making decisions for all of the aircraft. Besides, the centralized controller is not able to send action information to some aircraft, which may cause the aircraft to take action resulting in conflicts with other aircraft or even an unauthorized entry of designated airspace. This may lead to disruption of air traffic, causing risk to other airspace users and increased workload for pilots and controllers.

IV. Solution Method

In our previous work [13], we proposed a coordination mechanism design, where at each decision making step, we set a decision sequence for all aircraft and let them make decisions one after another. After one aircraft selects the action, it will broadcast this action information to all the other aircraft, the aircraft making decisions later can utilize this information to select better action. During this process, the state information and action information of each aircraft needs to be synchronized/communicated.

In this paper, we consider two communication strategies. In the first case, at each decision time step, all of the aircraft send their own information (position, velocity, destination) to a ground-based centralized controller which is located at the center of the airspace. Then the centralized controller will run the proposed algorithm and send the result joint action to each aircraft. In the second case we assume the centralized controller is down and the aircraft needs to use their onboard computer to run the proposed algorithm. In this case the aircraft will communicate the state information and joint action information between each other. We describe the two cases in the following two subsections.

A. Framework 1: Centralized Control and Air-to-ground Communications

In this case, a centralized controller is responsible to receive information from all of the aircraft and issue the result joint action to each aircraft. In this process, communication loss may happen and we assume the loss is two-directional (the aircraft cannot send information to the centralized controller and the centralized controller cannot issue action to the aircraft). Whenever a communication loss happens, the centralized controller will lose the state information of some en route aircraft. In our work the centralized controller will use the state and action information from the last step to predict the current position/velocity of the aircraft:

$$\begin{aligned} v &= v^- \\ \psi &= \psi^- + a^- \\ x &= x^- + v\cos\psi \\ y &= y^- + v\sin\psi \end{aligned} \tag{5}$$

where the "super minus" sign denotes the state and action information from last decision step. Note here the state prediction from last step information is not accurate since there is uncertainty in the aircraft dynamics (described in Section II.B.), where the noises for heading angle and speed are normally distributed with standard deviation 2° and $5m/s$. As shown in Fig. 2, the orange point is the aircraft position from last step, the red point denotes the aircraft position for current step without any uncertainty, and the blue points describe the distribution of current step aircraft position given the uncertainty described above.

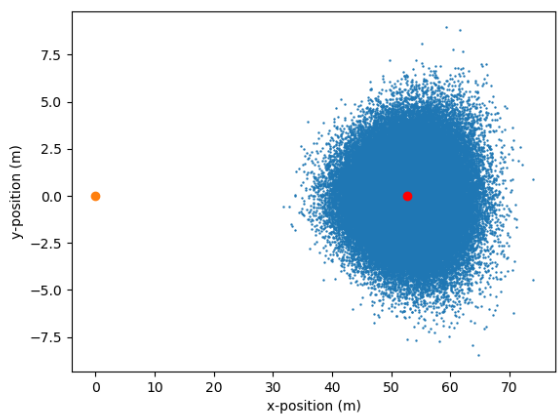


Fig. 2 The simulated distribution of the aircraft position in the current step is shown in blue as point cloud. Current aircraft position without uncertainty is shown in red inside the blue point cloud, and the orange point on the left is the aircraft position from previous step.

To ensure the safe separation in scenarios where the centralized controller has to predict the current state of the aircraft that lost communication, the strategy is to increase the separation distance to mitigate the prediction error. More specifically, we use the property of normal distribution to decide the increased separation distance. Since we have

$$P(-2.33\sigma_\psi < \psi < 2.33\sigma_\psi) = 99\% \tag{6}$$

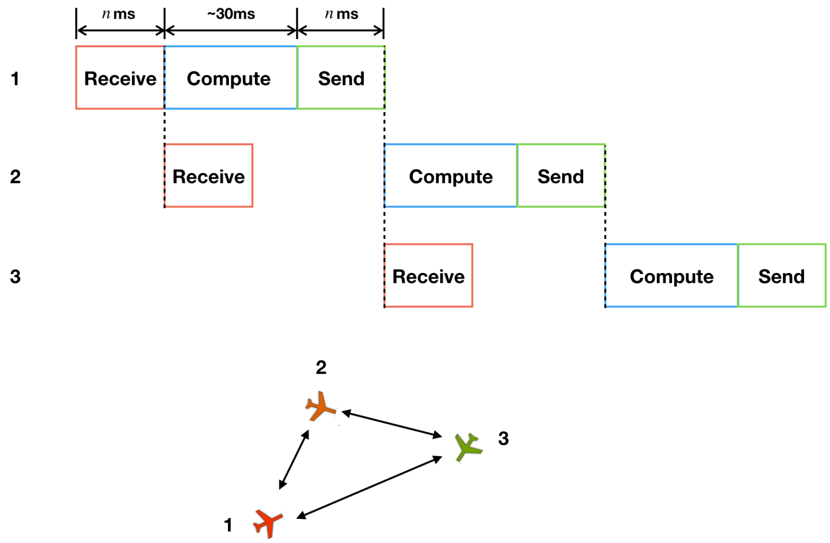
$$P(-2.33\sigma_v < v < 2.33\sigma_v) = 99\% \tag{7}$$

where $\sigma_\psi = 2°$ and $\sigma_v = 5m/s$. If we select a point from the above range, then we can guarantee it covers 98% of the position points. Since it can be calculated that the furthest point in the above range from the red point is 12 m, we will increase the separation distance by 12 m for each time step for our prediction while the communication is lost for this aircraft.

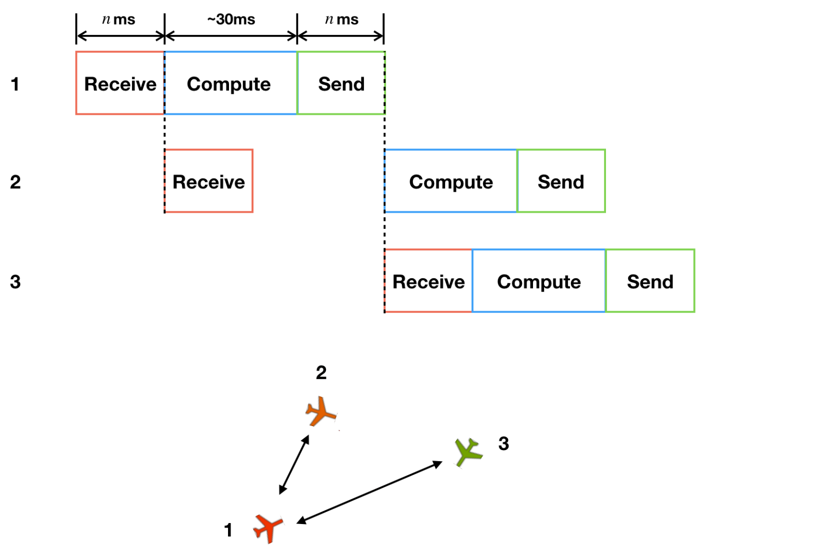
With the increased separation distance for the aircraft lost information and by assuming it fly straight, the centralized controller will issue actions for all of the other aircraft by following the process described in previous work [13].

B. Framework 2: Decentralized Control and Air-to-air Communications

In centralized control scenario, we found some cases where two aircraft close to each other lost communication to the centralized controller simultaneously. In this case, these two aircraft have a higher risk for conflict since the centralized controller is not able to issue actions to these two aircraft. Thus in this case study we propose algorithm to allow aircraft-to-aircraft communication for decentralized conflict resolution, where the aircraft will send its state/action information to other aircraft for cooperative conflict resolution purpose. Since in real world, aircraft close to each other are very unlikely to have communication loss, thus they can avoid potential conflicts through communication and coordination.



(a) Without communication loss



(b) With communication loss

Fig. 3 The algorithm running and communication process for decentralized control.

In the following, we will discuss the algorithm running and communication process with/without communication loss, as shown in Fig. 3. At first, assume currently we have three aircraft flying in the airspace without any communication loss (the arrow means they can send information to each other freely), as shown in Fig. 3a. To run the decentralized

MCTS algorithm, we first need to maintain an order among the aircraft (which is decided according to their order to enter the airspace), then the aircraft can synchronize joint action information according to the maintained order. At the beginning of the process, all of the aircraft (aircraft 2 and aircraft 3 in this case) will send its own information (position, velocity, destination) sequentially according to the maintained order to aircraft 1, which is denoted as the receive block for aircraft 1. Since we have two aircraft flying in the air, the receiving process can be finished in 2 ms, and this number will increase with the number of the aircraft. With all these aircraft information, aircraft 1 can now begin to make its decision by building a MCTS search tree. After aircraft 1 makes a decision and selects an action, it will broadcast this action information to all of the other aircraft sequentially in the same order as the receiving process. As aircraft 1 is making its decision, aircraft 2 begins to receive information from other aircraft. Although aircraft 2 finishes receiving the information from other aircraft in a shorter time compared to aircraft 1's decision process, it cannot start running the MCTS algorithm until it receives action information from aircraft 1, which is necessary information for aircraft 2 to run the onboard MCTS algorithm. Then the same process follows for aircraft 3. After all of the aircraft updates its own action, the algorithm running and communication process finishes and all of the aircraft will take the action according to their own MCTS algorithm result.

It can also happen two aircraft have communication loss and cannot send state/action information to each other, as shown in Fig. 3b, where aircraft 2 and aircraft 3 cannot send information to each other. In this case, the algorithm running and aircraft communication process will be the same as the case in Fig. 3a for aircraft 1 and aircraft 2. Starting from aircraft 3, when the receiving process of aircraft 3 finishes, it will be aware of the communication loss between itself and aircraft 2 since it did not get aircraft 2's information. Then, it can immediately proceed to run the onboard MCTS algorithm without state/action information from aircraft 2. Since communication loss usually happens between aircraft with a large distance, it is safe to run the MCTS algorithm without knowing position/velocity information and action/intent information from aircraft 2. Besides this difference, all the remaining process stays the same with no communication loss case.

V. Numerical Experiments and Results

A. Simulator

To validate the performance of the proposed algorithm, a free flight airspace simulator was built in Python where multiple aircraft can fly freely in the two-dimensional en route airspace above New York City. The airspace has 48km length and 48km width. We envision there will be multiple altitude levels where the eVTOL aircraft are operated. In the scope of this paper, we only focus on one altitude level.

To see the performance of this algorithm in real world applications, we simplify the UAM network by following the generic city model presented in [50, 51]. In this generic city model, seven vertiports are distributed in a "six around one" hexagonal pattern. As shown in Fig. 4, Vertiport 1 is located in the center of the hexagon and is located equidistant from the other six vertiports at a distance of 16km, which will cover the main congestion area of New York City. Overlays of the vertiport network are shown on a Google map image of New York in Fig. 4, which shows typical New York traffic on a Friday at 5 pm [52].

Given the above vertiports network, the demand model will generate flight requests stochastically. At each vertiport, after the taking off of the previous aircraft, the time interval for the next aircraft to take off is uniformly distributed between 1 minute and 3 minutes. Each aircraft will choose a random vertiport as its destination.

In the centralized scenario, the communication loss happens when the aircraft is 3km away from any vertiport. At each time step, there is 5% probability that a communication loss may happen, where the aircraft and the centralized controller cannot communicate information between each other. Whenever a communication loss happened, a uniformly distributed random variable in the range $[2, 5]$ will be generated to represent the communication loss time interval. During the communication loss, the aircraft will take the action to fly straight if it does not receive action information from the centralized controller and the centralized controller will use last step information to predict the aircraft state information at the current time step.

In the decentralized scenario, if two aircraft are at least 6km away from each other, they could have communication loss. The probability of communication loss increases from 0 to 0.5 linearly as the distance between them increases:

$$P(\text{communication loss}) = \text{clip}(\frac{d}{60000} - 1, 0, 0.5) \tag{8}$$

where d denotes the distance between two aircraft. Then the aircraft will make decisions according to the process

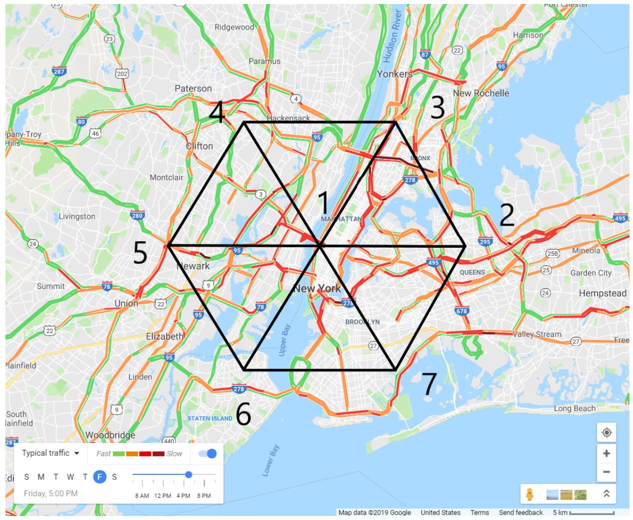


Fig. 4 Network of seven vertiports overlaid on New York city with segment length 16km.

described in Section IV.

The simulator will be kept running until 10,000 aircraft have been generated. During the running of this simulator, the number of conflicts and NMACs (short for near mid-air collision), the total number of aircraft generated, the number of aircraft which reached goals, and the average computation time to make each decision will be recorded and compared. The near mid-air collision (NMAC) standard is defined to be 500 feet by the Aeronautical Information Manual (7-6-3) [53].

B. Results

The numerical experiment described above is conducted over 5 random seeds and in each experiment, there are 10,000 aircraft generated in total. The code implementation of this algorithm is available on GitHub *. The result of the numerical experiment is shown below.

Table 1 and Table 2 show the number of conflicts/NMACs and the number of aircraft reached the goal for centralized control and decentralized control scenarios. In the centralized case, we can see for all the 10,000 aircraft generated, the conflict probability is around 0.23% and the NMAC probability is 0.024%. For the decentralized scenario, the conflict probability is about 0.3% and NMAC probability is 0.006%.

The result shows the proposed algorithm is pretty efficient at ensuring safety separation between aircraft. Comparing with previous work [13] where the conflict/NMAC probability is 0.2% and 0.004%, the performance of the proposed algorithm for centralized control and decentralized control is a little worse since here we considered the communication constraints. Besides, comparing with centralized control, decentralized control is more robust since the aircraft can communicate to each other. This shows that a communication network with more links is beneficial to the whole guidance and collision avoidance system.

Table 1 Performance of the algorithm for centralized control.

	mean	variance	min	max
Number of Conflicts	23	44.8	15	35
Number of NMACs	2.4	2.64	0	5
Number of Aircraft Reached Goal	9995.2	10.56	9990	10000

*`https://github.com/xuxiyang1993/MultiAgent-Guidance-Communication`

Table 2 Performance of the algorithm for decentralized control.

	mean	variance	min	max
Number of Conflicts	30.2	27.8	25	37
Number of NMACs	0.6	0.24	0	1
Number of Aircraft Reached Goal	9999.8	0.96	9998	10000

VI. Conclusion

A computational guidance algorithm for autonomous on-demand free flight operations with communication constraints is proposed in this paper. In this paper we formulate this problem as a Markov Decision Process (MDP) and solve it by an online algorithm Monte Carlo Tree Search (MCTS). In centralized control, we use forward propagation for the centralized controller to predict the state information of aircraft that lost communication. In decentralized control, we design an efficient communication framework for aircraft-to-aircraft communication. With the communication constraints including latency, bandwidth, and loss, we propose a robust computational guidance algorithm for multiple cooperative eVTOL aircraft in urban air mobility. Numerical experiments show that this proposed algorithm has promising performance to help an aircraft reach its destination and avoid potential conflicts with other aircraft.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1565979.

References

[1] Gipson, L., "NASA Embraces Urban Air Mobility, Calls for Market Study," `https://www.nasa.gov/aero/nasa-embraces-urban-air-mobility`, 2017. Accessed: 2018-01-19.

[2] Gipson, L., "Taking Air Travel to the Streets, or Just Above Them," `https://www.nasa.gov/aero/taking-air-travel-to-the-streets-or-just-above-them`, 2018. Accessed: 2018-08-13.

[3] "Uber Elevate | The Future of Urban Air Transport," `https://www.uber.com/info/elevate/`, 2017. Accessed: 2018-08-13.

[4] Holden, J., and Goel, N., "Fast-Forwarding to a Future of On-Demand Urban Air Transportation," *San Francisco, CA*, 2016.

[5] "Urban Air Mobility," `http://publicaffairs.airbus.com/default/public-affairs/int/en/our-topics/Urban-Air-Mobility.html`, 2018. Accessed: 2018-08-13.

[6] "Future of Urban Mobility," `http://www.airbus.com/newsroom/news/en/2016/12/My-Kind-Of-Flyover.html`, 2017. Accessed: 2018-08-13.

[7] Mueller, E. R., Kopardekar, P. H., and Goodrich, K. H., "Enabling Airspace Integration for High-Density On-Demand Mobility Operations," *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 3086.

[8] Antcliff, K. R., Moore, M. D., and Goodrich, K. H., "Silicon Valley as an Early Adopter for On-Demand Civil VTOL Operations," *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016, p. 3466.

[9] Moore, M. D., and Goodrich, K. H., "High speed mobility through on-demand aviation," *2013 Aviation Technology, Integration, and Operations Conference*, 2013, p. 4373.

[10] Vascik, P. D., "Systems-level analysis of On Demand Mobility for aviation," Ph.D. thesis, Massachusetts Institute of Technology, 2017.

[11] Bosson, C., and Lauderdale, T. A., "Simulation Evaluations of an Autonomous Urban Air Mobility Network Management and Separation Service," *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3365. doi:10.2514/6.2018-3365.

[12] Yang, X., and Wei, P., "Autonomous On-Demand Free Flight Operations in Urban Air Mobility using Monte Carlo Tree Search," *International Conference on Research in Air Transportation (ICRAT), Barcelona, Spain*, 2018.

[13] Yang, X., Deng, L., and Wei, P., "Multi-Agent Autonomous On-Demand Free Flight Operations in Urban Air Mobility," *AIAA Aviation 2019 Forum*, 2019, p. 3520.

[14] Rappaport, T. S., *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, 2002.

[15] Orjih, O., "Recent Developments in Aircraft Wireless Networks," *dated Apr*, Vol. 23, 2006.

[16] Pradeep, P., and Wei, P., "Energy efficient arrival with RTA constraint for urban eVTOL operations," *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 2008.

[17] Lovering, Z., "Vahana Configuration Trade Study - Part I – Vahana," , Dec 2016. URL `https://vahana.aero/vahana-configuration-trade-study-part-i-47729eed1cdf`.

[18] Frazzoli, E., Mao, Z.-H., Oh, J.-H., and Feron, E., "Resolution of conflicts involving many aircraft via semidefinite programming," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, 2001, pp. 79–86.

[19] Raghunathan, A. U., Gopal, V., Subramanian, D., Biegler, L. T., and Samad, T., "Dynamic optimization strategies for three-dimensional conflict resolution of multiple aircraft," *Journal of guidance, control, and dynamics*, Vol. 27, No. 4, 2004, pp. 586–594.

[20] Enright, P. J., and Conway, B. A., "Discrete approximations to optimal trajectories using direct transcription and nonlinear programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994–1002.

[21] Schouwenaars, T., De Moor, B., Feron, E., and How, J., "Mixed integer programming for multi-vehicle path planning," *Control Conference (ECC), 2001 European*, IEEE, 2001, pp. 2603–2608.

[22] Richards, A., and How, J. P., "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," *American Control Conference, 2002. Proceedings of the 2002*, Vol. 3, IEEE, 2002, pp. 1936–1941.

[23] Pallottino, L., Feron, E. M., and Bicchi, A., "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE transactions on intelligent transportation systems*, Vol. 3, No. 1, 2002, pp. 3–11.

[24] Vela, A., Solak, S., Singhose, W., and Clarke, J.-P., "A mixed integer program for flight-level assignment and speed control for conflict resolution," *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, IEEE, 2009, pp. 5219–5226.

[25] Mellinger, D., Kushleyev, A., and Kumar, V., "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, 2012, pp. 477–483.

[26] Augugliaro, F., Schoellig, A. P., and D'Andrea, R., "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, 2012, pp. 1917–1922.

[27] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., "Model predictive control of swarms of spacecraft using sequential convex programming," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725–1740.

[28] Acikmese, B., and Ploen, S. R., "Convex programming approach to powered descent guidance for mars landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366.

[29] Delahaye, D., Peyronne, C., Mongeau, M., and Puechmorel, S., "Aircraft conflict resolution by genetic algorithm and B-spline approximation," *EIWAC 2010, 2nd ENRI International Workshop on ATM/CNS*, 2010, pp. 71–78.

[30] Cobano, J. A., Conde, R., Alejo, D., and Ollero, A., "Path planning based on genetic algorithms and the monte-carlo method to avoid aerial vehicle collisions under uncertainties," *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 4429–4434.

[31] Pontani, M., and Conway, B. A., "Particle swarm optimization applied to space trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, 2010, pp. 1429–1441.

[32] Pallottino, L., Scordio, V. G., Frazzoli, E., and Bicchi, A., "Probabilistic verification of a decentralized policy for conflict resolution in multi-agent systems," *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, IEEE, 2006, pp. 2448–2453.

[33] Wollkind, S., Valasek, J., and Ioerger, T., "Automated conflict resolution for air traffic management using cooperative multiagent negotiation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 4992.

[34] Purwin, O., D'Andrea, R., and Lee, J.-W., "Theory and implementation of path planning by negotiation for decentralized agents," *Robotics and Autonomous Systems*, Vol. 56, No. 5, 2008, pp. 422–436.

[35] Desaraju, V. R., and How, J. P., "Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees," *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 4956–4961.

[36] Schouwenaars, T., How, J., and Feron, E., "Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 5141.

[37] Inalhan, G., Stipanovic, D. M., and Tomlin, C. J., "Decentralized optimization, with application to multiple aircraft coordination," *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, Vol. 1, IEEE, 2002, pp. 1147–1155.

[38] Richards, A., and How, J., "Decentralized model predictive control of cooperating UAVs," *43rd IEEE Conference on Decision and Control*, Vol. 4, Citeseer, 2004, pp. 4286–4291.

[39] Bellman, R., "A Markovian Decision Process," *Indiana Univ. Math. J.*, Vol. 6, 1957, pp. 679–684.

[40] "Vahana," , 2018. URL `https://www.airbus.com/innovation/urban-air-mobility/vehicle-demonstrators/vahana.html`, accessed: 2019-10-16.

[41] Cook, S. P., and Brooks, D., "A Quantitative Metric to Enable Unmanned Aircraft Systems to Remain Well Clear," *Air Traffic Control Quarterly*, Vol. 23, No. 2-3, 2015, pp. 137–156. doi:10.2514/atcq.23.2-3.137.

[42] Coulom, R., "Efficient selectivity and backup operators in Monte-Carlo tree search," *International conference on computers and games*, Springer, 2006, pp. 72–83. doi:10.1007/978-3-540-75538-8_7.

[43] Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S., "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, Vol. 4, No. 1, 2012, pp. 1–43.

[44] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., "Mastering the game of Go with deep neural networks and tree search," *nature*, Vol. 529, No. 7587, 2016, p. 484. doi:10.1038/nature16961.

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

[46] Champandard, A. J., "Monte-Carlo tree search in TOTAL WAR: ROME II's campaign AI," *AIGameDev.com: http://aigamedev.com/open/coverage/mcts-rome-ii*, 2014.

[47] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al., "Mastering the game of Go without human knowledge," *Nature*, Vol. 550, No. 7676, 2017, p. 354. doi:10.1038/nature24270.

[48] Li, H., Dai, H., and Li, C., "Communication complexity for distributed scheduling in wireless ad hoc networks," *submitted to Allerton*, 2008.

[49] "Air-Ground Voice Communications," , 2019. URL `https://www.skybrary.aero/index.php/Air-Ground_Voice_Communications`, accessed: 2019-11-30.

[50] Kohlman, L. W., and Patterson, M. D., "System-level urban air mobility transportation modeling and determination of energy-related constraints," *2018 Aviation Technology, Integration, and Operations Conference*, 2018, p. 3677.

[51] Patterson, M. D., Antcliff, K. R., and Kohlman, L. W., "A Proposed Approach to Studying Urban Air Mobility Missions Including an Initial Exploration of Mission Requirements," 2018.

[52] Google, "Google Maps," , 2019. URL `https://www.google.com/maps`, accessed: 2019-03-12.

[53] Administration, F. A., "Near Midair Collision Reporting," , 2017. URL `http://www.faraim.org/aim/aim-4-03-14-530.html`, accessed: 2018-11-14.