

MIT Open Access Articles

Optimization-Based Autonomous Air Traffic Control for Airspace Capacity Improvement

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Bas #pin, Baris # et al. "Optimization-Based Autonomous Air Traffic Control for Airspace Capacity Improvement." IEEE Transactions on Aerospace and Electronic Systems, 56, 6 (December 2020): 4814 - 4830 © 2020 The Author(s)

As Published: 10.1109/TAES.2020.3003106

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: <https://hdl.handle.net/1721.1/130406>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



Optimization-Based Autonomous Air Traffic Control for Airspace Capacity Improvement

Barış Başpınar, Hamsa Balakrishnan, and Emre Koyuncu

Abstract

In order to handle increasing demand in air transportation, high-level automation support seems inevitable. This paper presents an optimization-based autonomous air traffic control (ATC) system and the determination of airspace capacity with respect to the proposed system. We model aircraft dynamics and guidance procedures for simulation of aircraft motion and trajectory prediction. The predicted trajectories are used during decision process and simulation of aircraft motion is the key factor to create a traffic environment for estimation of airspace capacity. We define the interventions of an air traffic controller (ATCo) as a set of maneuvers that is appropriate for real air traffic operations. The decision process of the designed ATC system is based on integer linear programming (ILP) constructed via a mapping process that contains discretization of the airspace with predicted trajectories to improve the time performance of conflict detection and resolution. We also present a procedure to estimate the airspace capacity with the proposed ATC system. This procedure consists of constructing a stochastic traffic simulation environment that includes the structure of the evaluated airspace. The approach is validated on real air traffic data for en-route airspace, and it is also shown that the designed ATC system can manage traffic much denser than current traffic.

Index Terms

Air traffic control, conflict detection and resolution, airspace capacity estimation, integer linear programming.

I. INTRODUCTION

It is expected that the number of commercial flights will almost double from 26 million to 48.7 million, and 13.5 trillion passenger-kilometer will be flown by 2030, which is almost three times what is flown by airlines today [1]. Therefore, the airspace capacity should also increase accordingly to accommodate the increase in the air traffic volume. One of the major barriers in the expansion of the airspace capacity is the workload on the air traffic controllers (ATCo). Therefore, the future air traffic management (ATM) operations are going to require enhanced and high-level automation support for routine decision-making procedures, which contain the conflict detection and separation assurance.

Barış Başpınar and Emre Koyuncu are with the Department of Aeronautical Engineering, Istanbul Technical University, Istanbul, 34469 Turkey. This work was conducted while Barış Başpınar was with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA (as a Visiting Student Researcher) (e-mail: {baspinarb,emre.koyuncu}@itu.edu.tr)

Hamsa Balakrishnan is with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA (e-mail: hamsa@mit.edu)

Manuscript received April 19, 2005; revised August 26, 2015.

Development of automation tools for large-scale ATM scenarios is a challenging subject. Firstly, developed system should contain realistic models that can be implemented to real operation. Overly simplified aircraft models endanger the operation because of the inaccurate predictions. The simplified conflict resolution strategies also limit the airspace capacity. Secondly, the algorithm should be highly scalable with respect to the number of aircraft in order to cope with the increasing air traffic volume. Then, an optimization process should improve the efficiency of the system, which is desirable for all stakeholders. Finally, the algorithm should be verified on the real air traffic data to demonstrate its applicability to the real world problems. The main objective of this paper is to develop highly scalable automation tool that addresses the challenges described above for performing separation provision task of air traffic controller for en-route operation.

A. Related Studies

There are several studies in the literature associated with conflict detection and separation assurance problems. All of these studies focus on either free flight concept or ground based operation. The free flight concept does not contain a centralized traffic controller, and aircraft are responsible from their conflict resolutions that are performed airborne. The ground based operation corresponds to a centralized air traffic control mechanism. The main operational difference between the free flight and centralized mechanism is that the flight path intent information is not used in the free flight approach, whereas it is utilized in the ground based operation during conflict detection and resolution processes.

In the literature, some studies are centred on conflict detection problem without conflict resolution process. Most of these works are based on the free flight concept. The work in [2] defines protected zones around aircraft, and then uses these zones for conflict detection. These zones are propagated for a fixed time horizon to check the potential collisions. The propagation process is limited to simplified aircraft dynamics. Shewchun et al. [3] consider more complex dynamics, such as along track and cross track fluctuations, which translates to bearing and acceleration uncertainties. The authors benefit from Linear Matrix Inequalities and positive semi-definite programming to solve the conflict detection problem. The probability theory is also studied in free flight to present the conflict detection as a probability distribution. The work in [4] models the trajectory prediction error as a normal distribution, with zero mean and a covariance matrix with eigenvectors in the along-track and cross-track directions. The conflict probability is computed in the horizontal plane according to these stochastic error dynamics. In the study [5], the authors utilize the predicted trajectories during conflict detection process in horizontal plane by focusing on the ground based operation unlike the aforementioned probabilistic approach. They focus on finding possible conflict locations. They use the idea that the conflict detection process can be sped up via a transformation from trajectories to bins. Vink et al. [6] also utilize the predicted trajectories for conflict detection. In addition, the authors present unpredictable aircraft dynamics as uncertainty regions around the trajectories, and the conflict detection can be achieved for 3D trajectories. Both studies assume that predicted trajectories are received from another source such as Flight Data Processing System (FDPS), hence they mainly work on generated trajectories without using an aircraft model. The probabilistic methods have also been applied to the ground-based approaches. For instance, the work [7] constructs a mathematical model by solving a partial differential equation with Dirichlet boundary

conditions to calculate the conflict probability. The study [8] uses the reachability analysis for evaluating the possible trajectories and proposes a probabilistic conflict detection algorithm based on the cumulative distribution function approximation benefiting from Laurent series expansion. The work [9] proposes the generalized polynomial chaos method to solve the stochastic optimal control problem and estimate the conflict probability. This method is presented with a simplified aircraft model in horizontal plane. Note that the aforementioned works are limited to conflict detection, while the automation of the air traffic control system would require algorithms that can perform both conflict detection and resolution.

There are also some methods that focus only on the separation assurance without conflict detection. The work in [10] benefits from the potential field method to ensure the safety in free flight. The approach is computationally cheap, however the potential field methods have inherent limitations, such as being stuck in the local minima and oscillating solutions in the presence of narrow passages and dense environments [11]. Therefore, it can cause loss of separation in realistic ATM scenarios. Tomlin's works [12], [13] present hybrid system frameworks for conflict resolution in horizontal plane. Although the paper does not directly address the conflict detection problem, the alert zones and protected zones terms are defined to acknowledge the issue. The resolution maneuvers are generated by solving the Hamilton-Jacobi-Bellman (HJB) equation. The authors use simplified aircraft dynamics and consider only conflict resolution in the horizontal plane because of high computational complexity of solving the HJB equation. In addition to these studies, the work [14] also develops a hybrid automaton for separation assurance in the horizontal plane. The aforementioned work is mainly about air traffic flow, however the presented automaton is used to ensure the separation. Overall, these methods have limited scalability regarding the time they take to generate solution because of the high computational demand, thus it is difficult to use these algorithms in real airspace. Furthermore, generating resolution maneuvers only on horizontal plan (2D) causes a bottleneck in the airspace capacity, and it is far from real ATM operation. In many practical situations, vertical maneuvers might be the only option to ensure the safety. Hence, it is important for an automated ATC system to operate in 3D for realistic applications.

There are also several studies that address conflict detection and resolution together. In the study [15], the authors use quadratic programming to model the conflict resolution process, and then the optimization problem is solved via semi-definite programming combined with a randomization scheme. The algorithm is capable of conflict detection and separation assurance and also uses flight plans during optimization process. However, the algorithm has exponential complexity with respect to the number of aircraft, which limits its applicability to large-scale ATM scenarios. The work [16] follows a similar approach that performs separation assurance via solving a quadratic program. The problem is also formulated as a mixed-integer programming in the study [17]. In these two studies, the problem is limited to the horizontal plane, and simplified aircraft dynamics is used to construct the optimization problem. The study [18] proposes a conflict detection and resolution algorithm based on a stochastic reachability analysis approach. In this study, the problem is evaluated in the horizontal plane, and the algorithm has poor scalability regarding the time it takes to generate solution for realistic ATM scenarios. For example, it takes 265 seconds to generate resolution maneuvers for a scenario that contains 8 aircraft. To separate the aircraft, the work [19] constructs binary linear programs with the altitude change and speed change actions. It does not use any

aircraft model and assumes that the actions are instantaneous. There is no aircraft dynamics in the method, and it is not possible to extend the method with different actions rather than the ones implemented. The authors of the study [20] benefit from the genetic algorithm for conflict resolution after detecting the conflicts via an algorithm based on minimum bounding box. However, in this method, the computation time is a real issue for large-scale ATM scenarios. Overall, it can be observed that most of the algorithmic approaches either tend to use simplified aircraft dynamics and limit the conflict resolution maneuvers to horizontal plane for the sake of reducing the computational complexity, or tend to incorporate more realistic conflict resolution maneuvers and aircraft dynamics by sacrificing computational complexity and hence limiting the scalability. There are also some studies that use heuristics to improve the scalability. In the works [21], [22], the authors create algorithms that emulate the decision process of an air traffic controller based on the finite state machine. The developed algorithms detect conflicts and ensure separations in 3D without an optimization process.

In addition to the algorithmic works above, there are also software tools developed for providing decision support. The tool developed by Yang and Kuchar [23] is based on the free flight concept. The authors benefit from the algorithm in [4] to achieve conflict detection. The tool supports the decision process of the pilot with a probability map of conflicts and suggests elemental maneuvers such as heading change, speed change, climb or descent to resolve the potential conflicts. For the ground-based control systems, NASA's Center-TRACON Automation System (CTAS) [24] and MITRE's URET [25] are developed for providing decision support to air traffic controllers. Both tools have conflict detection capability without resolution advisory and use flight plans to assist the conflict detection process.

The other issue is the estimation of the airspace capacity, which is limited by controller workload. In the literature, there are several studies focus on evaluating the controller workload via complexity metrics. Complexity is described as "hard to separate, analyze, or solve" harmonizing with most people's intuition [26]. According to [27], complexity is composition of situation complexity and cognitive complexity. Situation complexity is mainly about configuration of traffic, whereas cognitive complexity is closely related to the decision making process. Cognitive complexity is also affected from situation complexity. According to [26], [28], complexity drives the workload of an air traffic controller that is a limiting factor on the airspace capacity. The studies in the literature are mainly focused on situation complexity. Among all the complexity metrics given in the literature, traffic density is the most basic and most associated with complexity. As stated by [26], the body of literature seems at the same time to praise the concept of traffic density as the best available indicator of complexity, and to criticize it that it does not capture the richness of what controllers find complex. As an alternative, the model of Dynamic Density (DD) is described as aggregation of different complexity metrics. DD models are presented as either linear combination of different complexity metrics [29], [30] or non-linear combination of complexity factors as in work [31], which uses neural networks for non-linear regression. These studies represent several complexity metrics to construct more detailed mathematical model representations of situation complexity, where traffic density is one of complexity factors. Besides them, the work [32] proposes fractal dimension as a measure of the situation complexity of the traffic pattern. The fractal dimension evaluates the number of degree of freedom in the traffic flow. In the [33], air traffic complexity is defined as the required control effort to ensure the safety when a new aircraft enters into the airspace.

The authors describe an input-output framework and present a complexity map to evaluate the complexity of a traffic situation. Despite this lack of inclusion in metrics, the airspace structure is considered as an important factor for understanding complexity.

B. Overview and Summary of Contributions

This paper mainly focuses on providing a scalable and fully automated ATC system that can be used in the existing ATM system and redetermining the airspace capacity by the help of designed ATC system. To achieve these objectives, first of all, we design an aircraft model and guidance procedure to simulate and predict the motion of the aircraft. Secondly, we model a set of maneuvers that represents the possible choices of ATCo during decision process. Then, we develop a mapping process based on discretization of the airspace to improve the performance of conflict detection and resolution and create an integer linear programming (ILP) that uses the sets and matrices derived from mapping process to ensure the safety in the airspace. Next, we focus on a methodology to redetermine the airspace capacity that consists of creating a stochastic traffic environment to simulate traffics at different complexities and defining breaking point of an airspace in terms of different metrics. In consequence of these approaches, the designed autonomous ATC system has better scalability than previous algorithmic approaches such as [13], [16], [15]. Moreover, it presents a more realistic approach to the automated air traffic control problem by generating separation maneuvers in 3D and using detailed aircraft motion models in opposition to some aforementioned studies such as [2], [14], [17]. It also contains an optimization-based approach to improve the efficiency of the ATC process, which is a contribution when compared with the previous highly scalable algorithms such as [21], [22]. The optimization process is also a contribution in comparison with the existing software tools such as [24], [25]. Besides, the majority of the existing software tools do not generate a resolution advisory, whereas the proposed system generates an optimum solution. Furthermore, the effects of the designed system on the airspace capacity are also investigated, and it is shown that the designed ATC system can manage traffic much denser than current traffic. Besides, the sub-components of the proposed system can be used individually in different ATM applications. For example, the trajectory prediction model can be used when dealing with trajectory management, or the mapping process can be utilized when dealing with mitigation of airspace congestion.

The rest of the paper is organized as follows. Section II explains the aircraft model and the guidance algorithms that are used in simulation environment and trajectory prediction process. Section III contains the set of maneuvers that presents the interventions of the optimization-based autonomous ATCo. In Section IV, optimization-based separation assurance process is presented. Then, the airspace capacity redetermination process is explained in Section V. Finally, implementations and discussions are given in Section VI.

II. MODELING OF THE AIRCRAFT MOTION AND TRAJECTORY PREDICTION

An aircraft follows its planned trajectory according to flight plan and the interventions of air traffic controller. To simulate and predict this process, an aircraft model and trajectory tracking algorithms are required. This section presents the aircraft model and guidance algorithms that will be used in simulation and trajectory prediction process. The block diagram of the process is illustrated in Fig. 1. The motion of aircraft can be simulated using this structure

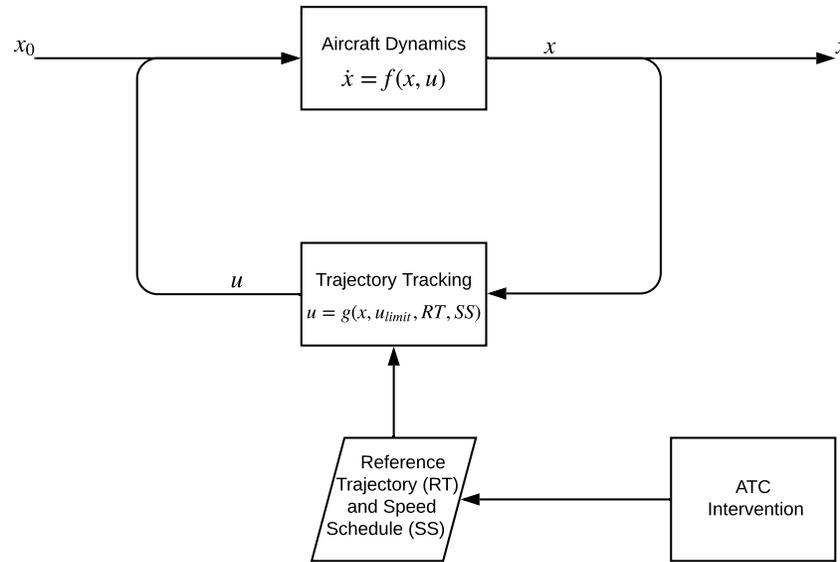


Fig. 1: Simulation of Aircraft Motion and Trajectory Prediction Process. In this diagram, the set of state variables (x) consists of the horizontal position, altitude, true airspeed, heading angle, and the mass of the aircraft. The set of control inputs (u) contains the engine thrust, bank angle, and flight path angle.

via numerical integration. In this structure, the aircraft dynamics corresponds to a set of differential equations that is driven by the control inputs (u) to calculate the state variables (x). The trajectory tracking layer is used to generate the necessary control inputs to follow the reference trajectory and adjust the speed. The reference trajectory and speed schedule can be modified by ATC interventions. ATC interventions will be explained in the next section. This section explains the details of aircraft dynamics, trajectory tracking guidance, reference trajectory and speed schedule. The proposed structure will be used for the purpose of simulating motion of the aircraft and predicting trajectories. Whereas simulation corresponds to integration of the system for a small time step, trajectory prediction refers to calculation of the system variables over a time horizon. The difference between them is also explained in this section.

A. Aircraft Dynamics

General form of the time-invariant dynamics of the aircraft is considered as:

$$\dot{x}(t) = f(x(t), u(t)) \quad x(0) = x_0 \quad (1)$$

where the $x(t) \in X \subseteq \mathbb{R}^n$, $u(t) \in U \subseteq \mathbb{R}^m$ such that $n, m \in \mathbb{N}$ and the state $x_0 \in X$ is the initial state of the aircraft. We use a point mass model for the aircraft dynamics, which can be easily derived from basic aerodynamics and was used in several air traffic control applications [34], [35]. The model can capture dynamics of a non-aggressive aircraft, so it is suitable for modeling the aircraft in air traffic control applications. Besides, the performance parameters of this model can be obtained for almost all aircraft types in civil aviation by using the

Base of Aircraft Data (BADA). The model is a nonlinear dynamical system with three control inputs and six state variables. The state variables are the horizontal position (x and y), altitude (h), the true airspeed (v), the heading angle (ψ) and the mass of the aircraft (m). The control inputs are the engine thrust (T), the bank angle (ϕ) and the flight path angle (γ). The wind acts as a disturbance on the aircraft dynamics, which is modeled by the wind speed, $W = [w_1, w_2, w_3]$. The equations of aircraft motion are:

$$\dot{x} = v \cos(\psi) \cos(\gamma) + w_1 \quad (2)$$

$$\dot{y} = v \sin(\psi) \cos(\gamma) + w_2 \quad (3)$$

$$\dot{h} = v \sin(\gamma) + w_3 \quad (4)$$

$$\dot{v} = -\frac{C_D S \rho v^2}{2m} - g \sin(\gamma) + \frac{T}{m} \quad (5)$$

$$\dot{\psi} = \frac{C_L S \rho v \sin(\phi)}{2m \cos(\gamma)} \quad (6)$$

$$\dot{m} = -F \quad (7)$$

In the equation set above, aerodynamic lift and drag coefficients are denoted by C_L and C_D , total wing surface area is S , air density is indicated as ρ , and the fuel consumption is indicated as F . These coefficients and other parameters such as bounds on the speed, mass, bank angle, and flight path angle are obtained from the Base of Aircraft Data (BADA) as in paper [36]. In general, the availability of the performance parameters of different aircraft types is a problem. Therefore, the BADA has been developed by EUROCONTROL to overcome this issue. BADA Revision 3.11 [37] provides operations and procedures data for a total of 405 aircraft types. Hence, a model that is constructed benefiting from BADA covers performance parameters of almost all aircraft types in Air Traffic Management.

In BADA, the thrust and fuel consumption are presented as functions of altitude, speed, flight phase, and engine type. The fuel consumption is also affected by the thrust. Note that the thrust and fuel consumption rates are presented implicitly. For example, the aircraft's altitude changes gradually when climbing and the fuel consumption is defined as a function of the altitude, so the fuel consumption varies gradually with regard to the change in the altitude. For a specific type of aircraft, these functions depend on the flight phase. Therefore, it is necessary to describe the flight phase of the aircraft as a discrete state that helps to use the appropriate functions. In this case, the system involves the interaction of continuous and discrete dynamics, which is the fundamental property of a hybrid system [38]. A hybrid automaton is a model and specification language for hybrid system. It can be used to evaluate the system dynamics with phased operations.

A hybrid automaton H is a collection $H = \{Q, X, f, I, D, E, G, R\}$, with

- Q is a set of discrete states
- X is a set of continuous variables
- $f : Q \times X \rightarrow TX$ is a vector field
- $I \subseteq Q \times X$ is a set of initial states
- $D : Q \rightarrow P(X)$ is a domain
- $E \subset Q \times Q$ is a set of discrete transitions

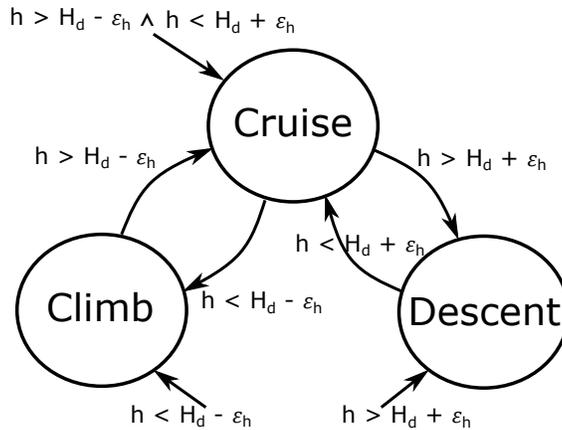


Fig. 2: Flight Phase Transitions.

- $G : E \rightarrow P(X)$ assigns to each $e = (q, q') \in E$ a guard
- $R : E \times X \rightarrow P(X)$ assigns to each $e = (q, q') \in E$ and $x \in X$ a reset relation

where $P(X)$ indicates the power set or set of all subsets of X . Hybrid automata define possible evaluations of their state. Roughly speaking, an automaton starts from an initial value $(q_0, x_0) \in I$ and continuous variable x is updated according to differential equation

$$\dot{x} = f(q_0, x) \quad (8)$$

$$x(0) = x_0 \quad (9)$$

the discrete state q remains constant as $q(t) = q_0$ until continuous variable x reaches the guard $G(q_0, q_1) \subseteq \mathbb{R}^n$ of some edge $(q_0, q_1) \in E$. Then, the discrete state changes value to q_1 and the continuous variable may get reset to some value in $R(q_0, q_1, x) \subseteq \mathbb{R}^n$. After this discrete transition, continuous evaluation resumes and the whole process is repeated.

The hybrid automaton for aircraft dynamics with respect to flight phases is presented in Fig. 2. The set of discrete states Q is $\{Climb, Cruise, Descent\}$, and the set of guard conditions G is described in terms of the altitude of aircraft h , the desired altitude H_d , and the buffer for altitude transitions ϵ_h . The same set of equations (2)-(7) is used for vector field f in each flight phase, however the functions that calculate the thrust and fuel consumption in f are differ from each other. The detailed information about these functions can be found in [37], [39].

B. Reference Trajectory and Speed Schedule

The reference trajectory corresponds to a set of way-points $WP = \{wp_0, wp_1, \dots, wp_k\}$ and $wp_i \in \mathbb{R}^3$, which contains the position information. During the flight, an aircraft follows its reference trajectory and this trajectory can be modified by Air Traffic Control Operator (ATCo). The speed schedule defines the desired speed of the aircraft. In BADA [37], the airline procedure model is defined according to engine model that gives the speed schedule as a

function of altitude and flight phase. In this study, desired speed is defined as the combination of the BADA airline procedure model and the speed intervention of ATCo as follows:

$$V_d = f_v(h, q) + V_c \quad (10)$$

where, V_d and V_c are the desired speed and the speed change, respectively. $f_v(h, q)$ corresponds to the speed schedule of the airline according to altitude and flight phase. In the rest of the paper, the flight plan term implies both reference trajectory and speed schedule. H_d corresponds to the desired altitude. When an aircraft flies between wp_i and wp_{i+1} , H_d defines the sum of the altitude information in wp_{i+1} and the altitude change H_c given by ATCo.

C. Trajectory Tracking Guidance

Trajectory Tracking corresponds to generate the required control inputs (e.g. bank angle and flight path angle) for an aircraft to follow the reference trajectory. In trajectory tracking, two control functions are combined for lateral and longitudinal motions. In the longitudinal part, the desired velocity and target altitude are reached by calculating the required flight path angle. The lateral control part includes the straight line controller, turn controller, and heading controller. These controllers calculate required bank angle to follow a horizontal path. In addition to them, a speed controller adjust thrust of the aircraft to keep the speed at the desired level during cruise.

1) *Longitudinal Controller*: The longitudinal controller determines the flight path angle. There are two different controllers, and one of them is used as longitudinal controller according to the flight phase. In climb and descent phases, the controller calculates the flight path angle according to desired speed with the following equation:

$$\gamma = \max \left\{ \gamma_{min}, \min \left\{ \gamma_{max}, \left(\frac{T - D}{m} - \frac{V_d - v}{\Delta t} \right) \frac{1}{g} \right\} \right\} \quad (11)$$

where, D is the drag. V_d is a function of h , so this controller calculates the necessary flight path angle to reach the desired altitude and keep the desired speed.

In cruise phase, there is a proportional controller to compensate the steady state error of the altitude as follows:

$$\gamma = \max \left\{ \gamma_{min}, \min \left\{ \gamma_{max}, k_a(H_d - h) \right\} \right\} \quad (12)$$

2) *Lateral Controller*: This controller calculates the bank angle to follow the path on the horizontal plane. Two different controllers are used with a switching mechanism as lateral controller to generate the bank angle, which are straight-line controller and turn controller. In addition to them, there is a heading controller that is used in a special case.

a) *Straight-Line Controller*: This controller generates required bank angle, ϕ , to follow the straight line between two way-points in horizontal plane. PLOS (pursuit and line-of-sight) guidance algorithm [40] is used to perform this controller. In Figure 3a, the straight-line between two way-points, $W_i = [x_i, y_i]$ and $W_{i+1} = [x_{i+1}, y_{i+1}]$, and the position of aircraft, $p = [x, y]$, are shown. S is the target location that aircraft tries to reach. If S is taken as W_{i+1} , heading angle rate, $\dot{\psi}_1$, can be expressed as (14) by pure pursuit guidance law.

$$\psi_d = \arctan 2(y_{i+1} - y, x_{i+1} - x) \quad (13)$$

$$\dot{\psi}_1 = k_1(\psi_d - \psi) \quad (14)$$

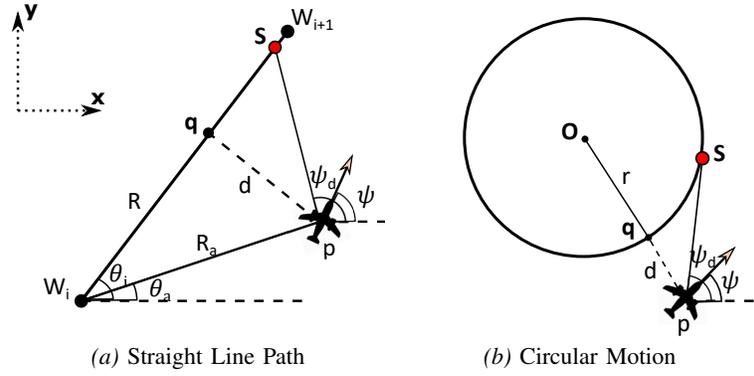


Fig. 3: Straight-Line and Arc Geometry.

where k_1 is the gain.

The LOS (line-of-sight) guidance law ensures that the angle between W_i and the aircraft is the same as that of the angle between W_i and W_{i+1} . The LOS guidance law is given by expression (19) with gain k_2 .

$$\theta_i = \arctan 2(y_{i+1} - y_i, x_{i+1} - x_i) \quad (15)$$

$$\theta_a = \arctan 2(y - y_i, x - x_i) \quad (16)$$

$$R_a = \|W_i - p\| \quad (17)$$

$$d = R_a \sin(\theta_i - \theta_a) \quad (18)$$

$$\dot{\psi}_2 = k_2 d \sin(\theta_i - \theta_a) \quad (19)$$

The combined guidance law is given as (20) and bank angle is calculated as (21).

$$\dot{\psi} = \dot{\psi}_1 + \dot{\psi}_2 \quad (20)$$

$$\phi = \max \left\{ -\phi_{max}, \min \left\{ \phi_{max}, \arctan \left(\frac{\dot{\psi} v}{g} \right) \right\} \right\} \quad (21)$$

b) Turn Controller: Turn controller provides necessary bank angle to follow a circular trajectory that is centered in $W_o = [x_o, y_o]$ with radius r as shown in Figure 3b where the position of aircraft is $p = [x, y]$ and S is the target location that aircraft tries to reach. If S is taken as tangent point to the line joining the aircraft and the loiter circle center, then desired heading angle is calculated by expression (27). Then, heading angle rate, $\dot{\psi}_1$, for desired heading is ensured by expression (28).

$$\theta_a = \arctan 2(y - y_o, x - x_o) \quad (22)$$

$$\alpha = \begin{cases} \arccos\left(\frac{r}{d+r}\right) & \text{for left turn} \\ -\arccos\left(\frac{r}{d+r}\right) & \text{for right turn} \end{cases} \quad (23)$$

$$S = [x_s, y_s] \quad (24)$$

$$x_s = x_o + r \cos(\theta_a + \alpha) \quad (25)$$

$$y_s = y_o + r \sin(\theta_a + \alpha) \quad (26)$$

$$\psi_d = \arctan 2(y_s - y_o, x_s - x_o) \quad (27)$$

$$\dot{\psi}_1 = k_1(\psi_d - \psi) \quad (28)$$

Another heading angle rate, $\dot{\psi}_2$, is generated to decrease the cross-track error (d) with gain k_2 as expressed below.

$$d = \|W_o - p\| - r \quad (29)$$

$$\dot{\psi}_2 = k_2 d \quad (30)$$

The combined heading angle rate is provided as (31) and bank angle is calculated as (32).

$$\dot{\psi} = \dot{\psi}_1 + \dot{\psi}_2 \quad (31)$$

$$\phi = \max \left\{ -\phi_{max}, \min \left\{ \phi_{max}, \arctan \left(\frac{\dot{\psi} v}{g} \right) \right\} \right\} \quad (32)$$

c) Heading Controller: Heading controller turns the aircraft to desired heading by arranging the necessary bank angle and keeps this heading value as stable. The heading angle rate and bank angle are calculated as follows:

$$\dot{\psi} = k(\psi_d - \psi) \quad (33)$$

$$\phi = \max \left\{ -\phi_{max}, \min \left\{ \phi_{max}, \arctan \left(\frac{\dot{\psi} v}{g} \right) \right\} \right\} \quad (34)$$

This controller is only used to direct the aircraft to a new way-point that is not found in the original flight plan.

d) Transitions between Horizontal Controllers: In any stage of the flight, only one of the straight-line, turn and heading controllers is operated as lateral controller as presented in Fig. 4. The system is initialized with straight-line controller. Assume that the aircraft has a list of way-points and it is flying between wp_i and wp_{i+1} , where i symbolizes the current flight segment in the flight plan. All of the segments of the flight can be presented via lines between consecutive way-points, and the transition between two lines can be presented as an arc as in fly-by transition approach. In this case, the reference trajectory consists of lines and arcs. The position $[x_{i+1,in}, y_{i+1,in}]$ defines the entry point of the arc between the line $l_{wp_i, wp_{i+1}}$ and line $l_{wp_{i+1}, wp_{i+2}}$. The straight-line controller calculates the bank angle to follow the line $l_{wp_i, wp_{i+1}}$ until the transition condition $d_{i+1,in} \leq \epsilon_d$, which describes reaching the entry point of the arc, and $d_{i+1,in}$ is computed as follows:

$$d_{i+1,in} = \sqrt{(x_{i+1,in} - x)^2 + (y_{i+1,in} - y)^2} \quad (35)$$

where, $[x, y]$ is the current position of the aircraft. Then the turn controller is activated, and it generates the bank angle to follow the arc until catching the course of the next line $l_{wp_{i+1}, wp_{i+2}}$, which is symbolized as θ_{i+1} . There is also a reset relation $i := i + 1$ in the transition from *Turn* to *Straight – Line* that specifies passing the current trajectory segment. The heading controller is only triggered when a new way-point is added to the reference trajectory as the next way-point wp_{i+1} or the next way-point wp_{i+1} is modified. The switch σ turns to 1 when

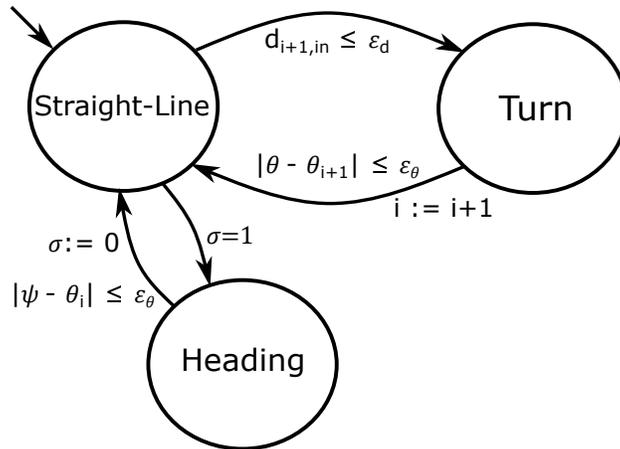


Fig. 4: Transitions between Horizontal Controllers.

the next way-point wp_{i+1} is changed, then the heading controller adjusts the direction of the aircraft towards to the next way-point until the heading of the aircraft gets closer to the course of the line $l_{wp_i,wp_{i+1}}$. Afterwards, the switch of the special case σ turns to 0, and the straight-line controller takes over as the lateral controller. When the aircraft reaches the last way-point of the reference trajectory, the controllers are deactivated and the process is terminated.

3) *Speed Controller*: The last controller aims to hold the speed at the desired value for the cruise phase. It arranges the thrust to keep the speed with proportional control. The speed controller is only active during cruise phase because the longitudinal controller already adjusts the speed in climb and descent phases.

D. Simulation and Trajectory Prediction

It is important to emphasize the difference between simulation and trajectory prediction from the stand point of this study. The simulation refers to the proceed one step at a time. Let us consider a simulation environment that contains several aircraft, which fly according to different flight plans. The simulation process contains movements of the all aircraft in the environment, and ATCo can intervene any aircraft at any time step of the simulation. During the decision process, ATCo uses the trajectory prediction as a tool to calculate the future states of the aircraft and decide which action is better for an aircraft. The trajectory prediction can be performed via the reference trajectory to check the possible conflicts or a modified trajectory to forecast the consequences of an action. Of course we will use same aircraft model and trajectory tracking algorithms for both simulation and trajectory prediction. However, they correspond to the different parts of the real operation. While the simulation environment corresponds to the virtual version of the real operation in an airspace, the trajectory prediction refers to the calculations of the future states of the aircraft during decision process.

III. ATCo INTERVENTIONS

The purpose of air traffic control is to ensure the safe and efficient flow of air traffic. To prevent collisions, an air traffic controller (ATCo) ensures that each aircraft maintains a minimum amount of empty space around it at all

times. The interventions of the ATCo are denoted as maneuvers. At any time during the flight, the ATCo can give a maneuver to an aircraft to ensure its safety. In this section, we define the set of maneuvers that will be used for separation assurance. Note that the term 'Null Action' is used to refer to following the current trajectory without any change; it can also be given by an ATCo as a maneuver.

A. Speed Change

A speed change corresponds to the modification of the desired speed. As presented before, the speed schedule contains the speed information according to the flight level, and the desired speed of the aircraft is the combination of the speed schedule and the speed change as in Eq. (10). At any time during the flight, ATCo can give a speed change to an aircraft by adjusting the V_c . The desired true airspeed of the aircraft is updated according to the speed change, and the true airspeed is held at the desired value with the controllers.

B. Altitude Change

As the speed change, the altitude change includes the adjustment of the schedule. Let us consider that the aircraft is flying between wp_i and wp_{i+1} , then the planned altitude of the aircraft is presented in wp_{i+1} as h_{i+1} . The desired altitude of the aircraft refers to the sum of h_{i+1} and the altitude change H_c . And the ATCo can modify the H_c . When ATCo changes the value of H_c or V_c for an aircraft, this modification continues until the end of the flight without affected from the other interventions. These variables can be set to another value only with another intervention of the ATCo to these variables.

C. Direct Routing

An ATCo can make an aircraft skip a sequence of way-points, a maneuver known as Direct Routing. Direct routing can involve skipping next way-point or the next 2-3 way-points as shown in Fig. 5. In Fig. 5, three different scenarios are illustrated. An action is given to the aircraft at time $t = 40s$ for each scenario. These actions, 'Null Action', 'Direct Routing 1' and 'Direct Routing 2' correspond to following the current trajectory, skipping the next way-point, and skipping the next two consecutive way-points, respectively. Before implementing the action, feasibility is always checked. If the action is not feasible, then 'Null Action' is implemented instead of the given action. For example, it is not possible to skip the next four consecutive way-points in the scenario in Fig. 5, so 'Null Action' is performed when 'Direct Routing 4' is requested.

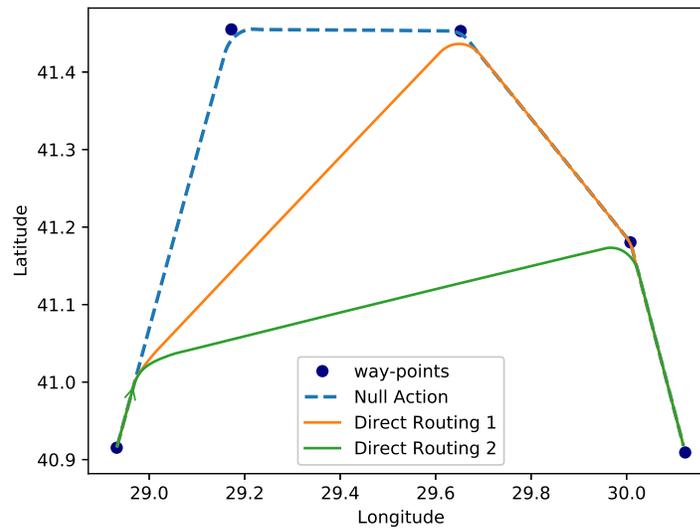


Fig. 5: Direct Routing Action.

D. Course Change

Course change refers to changing the current reference course of the aircraft. It is implemented by modifying the next way-point wp_{i+1} in the flight plan, using Equation (40).

$$\theta_i = \arctan 2(y_{i+1} - y_i, x_{i+1} - x_i) \quad (36)$$

$$d_{i+1} = \sqrt{(x_{i+1} - x)^2 + (y_{i+1} - y)^2} \quad (37)$$

$$x_{new} = x + d_{i+1} \cos(\theta_i - \theta_c) \quad (38)$$

$$y_{new} = y + d_{i+1} \sin(\theta_i - \theta_c) \quad (39)$$

$$wp_{i+1} = [x_{new}, y_{new}, h_{i+1}] \quad (40)$$

where, θ_i , d_{i+1} and θ_c define the current reference course, distance from the aircraft to the next way-point, and course change, respectively. The course change θ_c can be positive or negative depending on the direction of the motion. While one of them decreases the length of the path and acts like a direct routing action, the other one causes delay, as illustrated in the scenarios in Fig. 6.

E. Vector for Spacing

Vector for spacing (VFS) is a kind of delaying motion. It deviates the aircraft from the reference trajectory to cause a delay. It is implemented by inserting a new way-point between the current wp_i and wp_{i+1} , as calculated

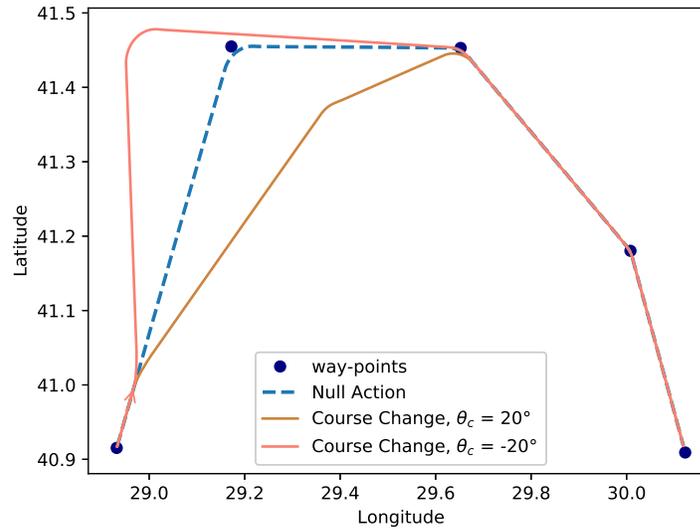


Fig. 6: Course Change.

via Equation (43). After adding the new way-point to the flight plan, the new way-point will be the wp_{i+1} .

$$x_{new} = x + l_c \cos(\theta_i - \theta_c) \quad (41)$$

$$y_{new} = y + l_c \sin(\theta_i - \theta_c) \quad (42)$$

$$wp_{new} = [x_{new}, y_{new}, h_{i+1}] \quad (43)$$

where, l_c and θ_c symbolize the distance from the aircraft to the new way-point and course change, respectively. Both l_c and θ_c are the inputs of VFS action. An example scenario is illustrated in Fig. 7. This maneuver always contains a feasibility assessment before implementation. Before adding wp_{new} to the flight plan, the inequality (44) is checked. If this inequality is ensured then the action is given, otherwise 'Null Action' is implemented.

$$\sqrt{(x_{i+1,in} - x)^2 + (y_{i+1,in} - y)^2} \geq 1.5l_c \quad (44)$$

F. Holding Pattern

Holding defines a racetrack pattern based on a reference position. In this study, this reference position is given as the position of the aircraft when the holding maneuver is activated. The procedure begins with a semi circle, followed by a straight-line, a semi-circle, and another straight-line. A standard holding pattern uses the right-side as the turn direction and takes approximately 4 minutes to complete. Each semi-circle takes a minute; in the same way, we also standardize the flight duration for the 180° turn as approximately a minute via appropriate turn radius, which is calculated using Equation (46). However, the flight duration of straight ahead sections t_l is given as input parameter for the pattern. The example scenario presented in Fig. 7 contains a holding pattern for $t_l = 30s$, which takes approximately 3 minutes.

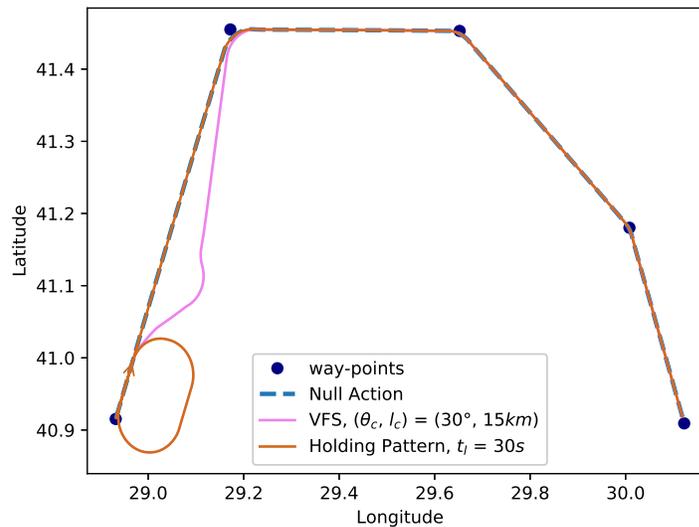


Fig. 7: Vector for Spacing and Holding Pattern.

$$\phi_r = \max \left\{ -\phi_{max}, \min \left\{ \phi_{max}, \arctan \left(\frac{\dot{\psi}_r v}{g} \right) \right\} \right\} \quad (45)$$

$$r_h = \frac{v^2}{g \tan(\phi_r)} \quad (46)$$

$$l_h = vt_l \quad (47)$$

where, ϕ_r and $\dot{\psi}_r$ are the reference bank angle and turn rate for the calculation of the radius. $\dot{\psi}_r$ is chosen as 3° per second to complete 180° turn in a minute. l_h defines the length of the straight ahead sections, which is adjusted via t_l . The holding pattern is implemented in level flight, whereas all of the other maneuvers can be implemented during climb, descent or cruise. The aircraft can climb or descent after completing the holding pattern.

IV. OPTIMIZATION-BASED SEPARATION ASSURANCE

An airspace can be discretized using a grid structure to speed up the conflict detection and resolution process. In an en-route airspace, the vertical separation minima d_s^v is 1000 ft where Reduced Vertical Separation Minima (RVSM) is applied, and the horizontal separation minima d_s^h is 5 NM. These values are also used as minimum separation distances in this study. Therefore, in discretized airspace, the flight levels are defined in thousands of feet (e.g., FL290, FL300, FL310 etc.), and the edge length of the square cells is chosen as 10 NM. In this study, we use double-layer grids for each flight level in sector to discretize the airspace. A double-layer grid consists of two non-overlapping grids where the edge lengths are equal, and the corners of the cells in the first grid overlap the centers of the cells in the second grid. Assume that two aircraft are flying north south direction at two neighboring cells in the first grid with violation of the separation, in this case, the conflict cannot be detected by checking the

cell occupancy in this grid. Thus, we use double-layer grids to catch overlooked loss of separation at the borders of the cells. We maps a predicted trajectory into the grid structure via matrices, where each element of a matrix corresponds to a specific cell in the grid. We introduce the mapping process for a grid, then same process can be applied for the other grid. Firstly, we focus on the mapping process, and then we continue with formulating the optimization problem.

The trajectory tracking algorithm computes an action trajectory \tilde{u} , which is a function of the current situation and flight plan. Starting from some initial state $x(t_0)$ at time t_0 , a state trajectory is derived from an action trajectory \tilde{u} as:

$$x(t_0 + T) = x(t_0) + \int_{t_0}^{t_0+T} f(x(t), u(t))dt \quad (48)$$

which integrates the state transition equation $\dot{x} = f(x, u)$ from the initial condition $x(t_0)$. Let \mathcal{A} be the set of aircraft and \mathcal{I} be the set of interventions or maneuvers. For an aircraft $a \in \mathcal{A}$, the \tilde{u} is also affected from maneuver $m \in \mathcal{I}$ at time t_0 . Let $\tilde{x}(a, m)$ denote the state trajectory or predicted trajectory over the interval $[t_0, t_0 + T]$, obtained by integrating (48), for the aircraft $a \in \mathcal{A}$ with the maneuver $m \in \mathcal{I}$, where the planning horizon is T .

Let $\mathcal{M}_{k \times l}$ be the set of $k \times l$ real matrices. A matrix $A \in \mathcal{M}_{k \times l}$ is called a Boolean matrix if its entries $a_{i,j} \in \mathcal{D}$, where $\mathcal{D} = \{1, 0\}$. The set of $k \times l$ Boolean matrices is donated by $\mathcal{B}_{k \times l}$. Let $A = (a_{i,j}), B = (b_{i,j}) \in \mathcal{B}_{k \times l}$. Then, $\neg A = (\neg a_{i,j})$, $A \wedge B = (a_{i,j} \wedge b_{i,j})$ and $A \vee B = (a_{i,j} \vee b_{i,j})$. Let $L = (l_{i,j}) \in \mathcal{M}_{k \times l}$, then the Hadamard product $B \circ L$ is given as $B \circ L = (b_{i,j} \cdot l_{i,j})$. Let $N = (n_{i,j}) \in \mathcal{M}_{k \times l}$, then the min and max operations between L and N are described as $\min(L, N) = (\min(l_{i,j}, n_{i,j}))$ and $\max(L, N) = (\max(l_{i,j}, n_{i,j}))$, respectively.

Let \mathcal{F} be the set of flight levels (FL). Let us define $g_{a,m}^f \in \mathcal{B}_{k \times l}$, $s_{a,m}^f \in \mathcal{M}_{k \times l}$ and $e_{a,m}^f \in \mathcal{M}_{k \times l}$ that contain location, entry time and exit time information for the corresponding cell of the grid, respectively. These matrices are calculated for the aircraft $a \in \mathcal{A}$ using $\tilde{x}(a, m)$. Let \mathcal{G} denote a grid and $c_{i,j,k}$ denote a specific cell in this grid, where (i, j) corresponds to horizontal location of the cell and k corresponds to flight level $f \in \mathcal{F}$. The cell is a three-dimensional object, and k specifies the center of the object along altitude axis. The height of the cell is $2\Delta h - 2\epsilon_h$, where Δh defines the altitude difference between two consecutive flight levels and ϵ_h symbolizes the altitude buffer. In this case, the two consecutive cells $c_{i,j,k}$ and $c_{i,j,k+1}$ have overlapping volume between FL_k and FL_{k+1} . Therefore, an aircraft is assigned to both cells when climbing or descending between FL_k and FL_{k+1} . Let the entry in the i^{th} row and j^{th} column of the matrix $g_{a,m}^f$, $s_{a,m}^f$ or $e_{a,m}^f$ be denoted as $(\cdot)_{i,j}$. The entries of these matrices are expressed as follows:

$$(g_{a,m}^f)_{i,j} = \begin{cases} 1 & \tilde{x}(a, m) \cap c_{i,j,f} \neq \emptyset \\ 0 & otherwise \end{cases} \quad (49)$$

$$(s_{a,m}^f)_{i,j} = \begin{cases} t_{i,j,f}^{in} - t_b & \tilde{x}(a, m) \cap c_{i,j,f} \neq \emptyset \\ M & otherwise \end{cases} \quad (50)$$

$$(e_{a,m}^f)_{i,j} = \begin{cases} t_{i,j,f}^{out} + t_b & \tilde{x}(a, m) \cap c_{i,j,f} \neq \emptyset \\ 0 & otherwise \end{cases} \quad (51)$$

where M is a large enough number, typically chosen greater than an upper bound on the planning horizon. The parameters $t_{i,j,f}^{in}$ and $t_{i,j,f}^{out}$ denote the entry time to the cell $c_{i,j,f}$ and exit time from the cell $c_{i,j,f}$, respectively. The parameter t_b is the time buffer.

Let us consider the case in which maneuver is the null action, which is symbolized as m_0 . Let a_p and a_q be the different aircraft in the airspace. The matrix g_{a_p,q,m_0}^f presents the intersections of the predicted trajectories without any interventions, which is given as follows:

$$g_{a_p,q,m_0}^f = g_{a_p,m_0}^f \wedge g_{a_q,m_0}^f \quad (52)$$

Let us introduce two new matrices s_{a_p,q,m_0}^f and e_{a_p,q,m_0}^f that contain the maximum values of the entry times and the minimum values of the exit times for the simultaneously occupied cells. The matrix δ_{a_p,q,m_0}^f contains the time differences between s_{a_p,q,m_0}^f and e_{a_p,q,m_0}^f . These matrices are calculated as follows:

$$s_{a_p,q,m_0}^f = \max(g_{a_p,q,m_0}^f \circ s_{a_p,m_0}^f, g_{a_p,q,m_0}^f \circ s_{a_q,m_0}^f) \quad (53)$$

$$e_{a_p,q,m_0}^f = \min(g_{a_p,q,m_0}^f \circ e_{a_p,m_0}^f, g_{a_p,q,m_0}^f \circ e_{a_q,m_0}^f) \quad (54)$$

$$\delta_{a_p,q,m_0}^f = s_{a_p,q,m_0}^f - e_{a_p,q,m_0}^f \quad (55)$$

The entry $(\delta_{a_p,q,m_0}^f)_{i,j}$ of the matrix δ_{a_p,q,m_0}^f corresponds to the time difference between aircraft a_p and a_q for the occupation of the cell $c_{i,j,f}$. If $(\delta_{a_p,q,m_0}^f)_{i,j} \geq 0$, then there is no conflict between them in the cell $c_{i,j,f}$, because they are not in the cell simultaneously. In the case of $(\delta_{a_p,q,m_0}^f)_{i,j} < 0$, there may be a conflict between them in the corresponding cell. Let τ_{a_p,q,m_0}^f and τ_{a_p,q,m_0} denote the sets of ordered pairs (t_1, t_2) with first element from s_{a_p,q,m_0}^f and second element from e_{a_p,q,m_0}^f for $f \in F$. These sets are defined as follows:

$$\tau_{a_p,q,m_0}^f = \{((s_{a_p,q,m_0}^f)_{i,j}, (e_{a_p,q,m_0}^f)_{i,j}) : (\delta_{a_p,q,m_0}^f)_{i,j} < 0\} \quad (56)$$

$$\tau_{a_p,q,m_0} = \bigcup_{f \in F} \tau_{a_p,q,m_0}^f \quad (57)$$

The set τ_{a_p,q,m_0} contains all time pairs that are candidate time intervals for the loss of separation between the aircraft a_p and a_q with m_0 . If $\tau_{a_p,q,m_0} = \emptyset$, then the two aircraft have already necessary separation for the planning horizon T , otherwise there may be a loss of separation at the candidate time intervals between the aircraft a_p and a_q . The set of conflicted aircraft \mathcal{A}_c is described as follows:

$$\begin{aligned} \mathcal{A}_c = & \{a_p : \forall a_p \in \mathcal{A}, \forall a_q \in \mathcal{A} \setminus \{a_p\}, \forall (t_1, t_2) \in \tau_{a_p,q,m_0}, \\ & \exists t \in [t_1, t_2] (\|h_{a_p,m_0}(t) - h_{a_q,m_0}(t)\| < d_s^v \wedge \\ & \|(x(t), y(t))_{a_p,m_0} - (x(t), y(t))_{a_q,m_0}\| < d_s^h)\} \end{aligned} \quad (58)$$

where, d_s^v and d_s^h symbolize vertical separation minima and horizontal separation minima, respectively. The parameters $(x(t), y(t), h(t))_{a_p,m_0}$ and $(x(t), y(t), h(t))_{a_q,m_0}$ belong to $\tilde{x}(a_p, m_0)$ and $\tilde{x}(a_q, m_0)$, respectively. The set of separated aircraft \mathcal{A}_s , which have separation without ATCo's intervention, can be presented as the relative complement of \mathcal{A}_c with respect to the set \mathcal{A} as follows:

$$\mathcal{A}_s = \mathcal{A} \setminus \mathcal{A}_c \quad (59)$$

Let $a_n \in \mathcal{A}_c$ and $a_r \in \mathcal{A}_c$ be two different aircraft in the airspace. Let matrix $g_{a_n}^f$ be the logical disjunction of the all matrices $g_{a_n,m}^f$ as in Eq. (60) that presents the all visitable cells at the flight level f for the aircraft a_n with the all possible maneuvers in the set \mathcal{I} . By using the matrices $g_{a_n}^f$ and $g_{a_r}^f$, the matrix g_{a_n,a_r}^f can be expressed as in (61), which contains intersections of the all possible trajectories of the aircraft a_n and a_r at flight level f via the maneuver set \mathcal{I} .

$$g_{a_n}^f = \bigvee_{m \in \mathcal{I}} g_{a_n,m}^f \quad \text{and} \quad g_{a_r}^f = \bigvee_{m \in \mathcal{I}} g_{a_r,m}^f \quad (60)$$

$$g_{a_n,a_r}^f = g_{a_n}^f \wedge g_{a_r}^f \quad (61)$$

Let $m_u \in \mathcal{I}$ and $m_v \in \mathcal{I}$ be interventions for the aircraft a_n and a_r , respectively. Then, the time matrices $s_{a_n,r,m_u,v}^f$, $e_{a_n,r,m_u,v}^f$ and $\delta_{a_n,r,m_u,v}^f$ can be calculated via $g_{a_n,r}^f$ as follows:

$$s_{a_n,r,m_u,v}^f = \max(g_{a_n,r}^f \circ s_{a_n,m_u}^f, g_{a_n,r}^f \circ s_{a_r,m_v}^f) \quad (62)$$

$$e_{a_n,r,m_u,v}^f = \min(g_{a_n,r}^f \circ e_{a_n,m_u}^f, g_{a_n,r}^f \circ e_{a_r,m_v}^f) \quad (63)$$

$$\delta_{a_n,r,m_u,v}^f = s_{a_n,r,m_u,v}^f - e_{a_n,r,m_u,v}^f \quad (64)$$

As mentioned before, there is no conflict in the corresponding cell between the aircraft a_n and a_r with the maneuver m_u and m_v if $(\delta_{a_n,r,m_u,v}^f)_{i,j} \geq 0$, there may be a loss of separation otherwise. Let us consider a special case that the aircraft a_n visits a specific cell $c_{i,j,f}$ with a specific maneuver m_1 , whereas it does not visit this cell with another maneuver m_2 and the aircraft a_r also visits this cell with maneuver m_v . In this case, the $(g_{a_n,r}^f)_{i,j}$ turns to 1. When aircraft a_n flies with maneuver m_2 , this situation does not cause any problem in terms of conflict detection because of the large number M in the $(s_{a_n,m_2}^f)_{i,j}$. In any case, $(\delta_{a_n,r,m_2,v}^f)_{i,j} \geq 0$ because of the large number M in the $(s_{a_n,m_2}^f)_{i,j}$, where the aircraft a_n does not visit the cell $c_{i,j,f}$ with maneuver m_2 .

The set $\tau_{a_n,r,m_u,v}$ can be obtained as previously presented:

$$\tau_{a_n,r,m_u,v}^f = \{((s_{a_n,r,m_u,v}^f)_{i,j}, (e_{a_n,r,m_u,v}^f)_{i,j}) : (\delta_{a_n,r,m_u,v}^f)_{i,j} < 0\} \quad (65)$$

$$\tau_{a_n,r,m_u,v} = \bigcup_{f \in F} \tau_{a_n,r,m_u,v}^f \quad (66)$$

Let us define a new set P_{cc} that consists of ordered quadruples. Each quadruple contains a specific maneuver pair for two aircraft that cause a loss of separation when these maneuvers are implemented together. The set P_{cc} is given by the following expression:

$$\begin{aligned} P_{cc} = & \{(a_n, a_r, m_u, m_v) : \forall a_n \in \mathcal{A}_c, \forall a_r \in \mathcal{A}_c \setminus \{a_n\}, \\ & \forall m_u \in \mathcal{I}, \forall m_v \in \mathcal{I}, \forall (t_1, t_2) \in \tau_{a_n,r,m_u,v}, \\ & \exists t \in [t_1, t_2] \quad (\|h_{a_n,m_u}(t) - h_{a_r,m_v}(t)\| < d_s^v \wedge \\ & \|(x(t), y(t))_{a_n,m_u} - (x(t), y(t))_{a_r,m_v}\| < d_s^h)\} \end{aligned} \quad (67)$$

Let us consider the case in which the aircraft $a_n \in \mathcal{A}_c$, whereas $a_r \in \mathcal{A}_s$. ATCo prefers null action than other maneuvers when an aircraft has separation. Therefore, in this study, the aircraft $a_r \in \mathcal{A}_s$ takes null action, while one of the maneuvers in the set \mathcal{I} can be given to the aircraft $a_n \in \mathcal{A}_c$. The matrix $g_{a_n,r}^f$ can be calculated with the Eq. (61) as previously presented. However, the matrix $g_{a_r}^f$ should be described as in the expression (68) instead of the expression (60). The rest of the matrices and sets are calculated via expressions from (62) to (66), where m_v is taken as m_0 .

$$g_{a_n}^f = \bigvee_{m \in \mathcal{I}} g_{a_n,m}^f \quad \text{and} \quad g_{a_r}^f = g_{a_r,m_0}^f \quad (68)$$

Then, the new set P_{cs} that consists of the restricted maneuvers for specific aircraft in the set \mathcal{A}_c because of the violation of the separation with an aircraft $a_r \in \mathcal{A}_s$ can be described as follows:

$$\begin{aligned} P_{cs} = & \{(a_n, m_u) : \forall a_n \in \mathcal{A}_c, \forall a_r \in \mathcal{A}_s, \forall m_u \in \mathcal{I}, \\ & \forall (t_1, t_2) \in \tau_{a_n,r,m_u,0}, \exists t \in [t_1, t_2] \\ & (\|h_{a_n,m_u}(t) - h_{a_r,m_0}(t)\| < d_s^v \wedge \\ & \|(x(t), y(t))_{a_n,m_u} - (x(t), y(t))_{a_r,m_0}\| < d_s^h)\} \end{aligned} \quad (69)$$

By using the defined sets and matrices, an integer linear programming (ILP) is formulated to ensure the safety in the airspace, while optimizing the separation assurance process as follows:

$$\min \sum_{m \in \mathcal{I}} \sum_{a \in \mathcal{A}_c} (f_{a,m} + CI t_{a,m}) x_{a,m} - \sum_{a \in \mathcal{A}_c} c_0 x_{a,m_0} \quad (70)$$

subject to

$$x_{a_n,m_u} + x_{a_r,m_v} \leq 1 \quad \forall (a_n, a_r, m_u, m_v) \in P_{cc} \quad (71)$$

$$x_{a_n,m_u} = 0 \quad \forall (a_n, m_u) \in P_{cs} \quad (72)$$

$$\sum_{m \in \mathcal{I}} x_{a,m} = 1 \quad \forall a \in \mathcal{A}_c \quad (73)$$

$$x_{a,m} \in \{0, 1\} \quad \forall a \in \mathcal{A}_c, \forall m \in \mathcal{I} \quad (74)$$

where, $x_{a,m}$ for all $a \in \mathcal{A}_c$ and $m \in \mathcal{I}$ are binary decision variables such that the variable $x_{a,m}$ is equal to 1 if aircraft a takes maneuver m as an action, and zero otherwise. The parameters $f_{a,m}$ and $t_{a,m}$ define the fuel consumption and travel duration, respectively. The parameter CI is the cost index. The combination of the $f_{a,m}$ and $t_{a,m}$ is defined as the total cost of the flight and presented as the first term of the objective function (70), whereas the second term of the objective function prioritizes the null action. In implementations, the c_0 will be chosen much greater than CI to prevent intervention if it is unnecessary in terms of separation. Constraints (71) ensure that restricted maneuver pair, which cause loss of separation, are not given together. Constraints (72) ensure that an action that causes a conflict with an aircraft $a_r \in \mathcal{A}_s$, which already has separation, is not chosen.

V. AIRSPACE CAPACITY REDETERMINATION

In the current ATM system, the maximum amount of workload that ATCos are able to sustain imposes the limits on the capacity of the airspace. In an automated ATM system, the capacity of the airspace should be redetermined

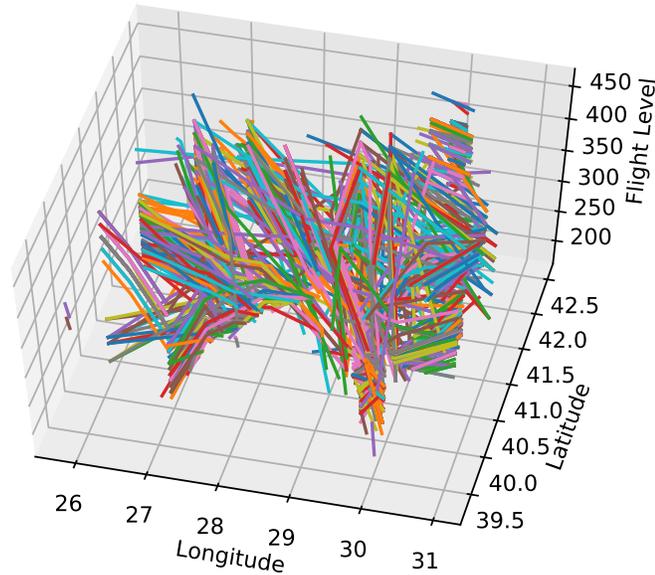


Fig. 8: Flight Plans in ISTANBUL ACC for a Standard Day. This figure illustrates the reference trajectories of all aircraft that operate during a standard day in the ISTANBUL ACC.

according to designed algorithms. This section presents the identification method of the airspace capacity according to developed autonomous air traffic control system. The capacity estimation bases on making simulations for different traffic densities to obtain the critical capacity of the system.

A. Stochastic Traffic Simulation Environment

We construct a traffic simulation environment to simulate the daily traffic in an airspace with desired throughput. The traffic simulator consists of two stochastic processes. The first process determines which reference trajectory is followed by an aircraft and the second stochastic process generates the entry time of the aircraft to the airspace.

Let \mathcal{R}_t be the set of the reference trajectories such that $\mathcal{R}_t = \{WP_1, WP_2, \dots, WP_M\}$, where $WP_j = \{wp_0, wp_1, \dots, wp_k\}$ and $wp_i \in \mathbb{R}^3$. Let \mathcal{R}_p denote the set of usage probabilities of the reference trajectories in \mathcal{R}_t such that $\mathcal{R}_p = \{p_1, p_2, \dots, p_M\}$ and $\sum_{i=1}^M p_i = 1$. A reference trajectory $r_t \in \mathcal{R}_t$ is assigned to the aircraft a randomly based on the probabilities in \mathcal{R}_p . The set \mathcal{R}_t and \mathcal{R}_p are extracted from the real flight plan data of the aircraft, which operate in the corresponding airspace. In Fig. 8, the reference trajectories of the all aircraft that operate during a standard day in the ISTANBUL ACC (area control centre) are illustrated. Some of the reference trajectories are flown more than others. The utilization frequency of a reference trajectory is used to generate the usage probability of this flight plan and all of the reference trajectories in the real operation are assigned to the set \mathcal{R}_t by removing duplicates.

After constructing the simulation environment for a sector with the sets \mathcal{R}_t and \mathcal{R}_p , the only input for the environment is the traffic density N that denotes the total number of flights during the day. The second stochastic process determines the entry times of the flights into the airspace according to N . We want to keep the workload

of the ATCo approximately at the same level during operation and operating with a flat demand profile rather than having peak and off-peak hours is a fact when the demand is close to the capacity of the airspace. Therefore, the entry times of the flights during the day are modelled as a Poisson process. It is more convenient to define a Poisson process in terms of the sequence of interarrival times, X_1, X_2, \dots , which are defined to be independent and identically distributed (i.d.d.). A renewal process is an arrival process for which the sequence of interarrival times is a sequence of i.d.d random variables and a Poisson process is a renewal process in which the interarrival intervals have an exponential distribution function; i.e., for some real $\lambda > 0$ each X_i has the density $f_X(x) = \lambda \exp(-\lambda x)$ for $x \geq 0$. To simulate a Poisson process with rate λ , the i.d.d. random variables X_1, X_2, \dots, X_N are generated, where $X_i \sim \text{Exponential}(\lambda)$. Then, the arrival times are given by $T_i = \sum_{k=1}^i X_k$. In our case, the arrival times correspond to the entry times of the aircraft into the airspace and λ equals to $86400/N$, where N is the traffic density and 86400 refers to 24 hours in terms of seconds.

In the simulation environment, each aircraft enters into the airspace at its entry time, however there is also a collision detection mechanism at the border, which shifts the entry time of a problematic aircraft. If the incoming aircraft leads to a collision with another aircraft when it enters into the airspace, the incoming aircraft is held at the entrance until a clearance. For the initialization of the state variables of an incoming aircraft, the initial position, altitude and heading angle are obtained from its reference trajectory. As previously mentioned, the speed schedule is presented as a function of altitude, and the initial speed of the aircraft is determined according to its initial altitude. And, the initial mass of the aircraft is taken as the nominal mass presented in BADA.

B. Breaking Point Analysis

The breaking point corresponds to the critical capacity of the airspace in terms of a specific metric, where the probability of the metric greater than the acceptable metric level has a sharp transition. This metric can be about safety, performance, efficiency, etc. Let S be a specific metric and S_a be the acceptable level of this metric. The number of aircraft n is the breaking point or the metric-specific capacity of the airspace if $P(S(n-1) > S_a) < b$ and $P(S(n+1) > S_a) > 1-b$, where $b \in (0, 0.5)$ and $S(n)$ is a non-decreasing function [41]. The capacity of an airspace can be affected from several metrics. In this case, a critical capacity value n_i is generated for each metric S_i and minimum of them is taken as the capacity of the airspace, $n = \min(n_1, n_2, \dots, n_l)$. It is also possible to define critical capacity as a range instead of a single number. In this case, the n is defined as a range $[n^l, n^u]$ and it corresponds to the critical capacity of the airspace if $P(S(n^l-1) > S_a) < b$ and $P(S(n^u+1) > S_a) > 1-b$, where $b \in (0, 0.5)$.

In this study, we use four different metrics to determine the airspace capacity. The collision per hour is presented as safety metric that corresponds to the number of collision in the airspace during an hour and the acceptable limit of this metric is taken as 0. The second and third metrics are about efficiency that are defined as extension of total cost per flight and extension of travel duration per flight. As previously presented, the total cost corresponds to the combination of the fuel consumption and travel duration as $f_{a,m} + \mathcal{C}It_{a,m}$. The acceptable limits of them are defined as 0.1% and 3%, respectively. The last metric is about the time performance of the optimization process. In the simulation environment, the presented optimization problem is repeatedly solved with time period δ to ensure

TABLE I: The Set of Maneuvers.

Maneuver ID	Maneuver
0	Null Action
1	Direct Routing 1
2	Direct Routing 2
3	Speed Change, $V_c = 10$ m/s
4	Speed Change, $V_c = -10$ m/s
5	Altitude Change, $H_c = 2000$ ft
6	Altitude Change, $H_c = 4000$ ft
7	Altitude Change, $H_c = -2000$ ft
8	Vector for Spacing, $(\theta_c, l_c) = (30^\circ, 15 \text{ km})$
9	Vector for Spacing, $(\theta_c, l_c) = (45^\circ, 15 \text{ km})$
10	Course Change, $\theta_c = 20^\circ$
11	Course Change, $\theta_c = -20^\circ$
12	Holding Pattern (Circular)

the safety in the airspace. If the execution time of the optimization process is smaller than δ seconds, it will be applicable with time period δ . Therefore, the last metric is chosen as execution time of the optimization process and the acceptable limit is taken as δ seconds. In implementations, the optimization-based ATCo generates actions every 20 seconds, so the acceptable limit is taken as 20 seconds. If the time period of the algorithm creates a bottleneck in the capacity, it can be increased easily or more powerful hardware can be used to improve the time performance of the algorithm. However, evaluation of the time performance is necessary to check the applicability of the chosen time period.

VI. IMPLEMENTATIONS

In this section, firstly, several example scenarios are analyzed to show the working principle and validity of the optimization-based autonomous ATC system, and then the results of the simulations in the stochastic traffic environment are evaluated to determine the capacity of a specific airspace and analyze the solution strategies of the autonomous system. In simulations, the performance parameters of Boeing 737-800 are used for all aircraft in the sector, if the aircraft type is not expressed. During the implementations, the set of interventions \mathcal{I} consists of 13 different maneuvers as presented in Table I. This set can be rearranged by adding new maneuvers or removing some of the maneuvers. The chosen maneuvers are rational and applicable to the real ATC system. When adding a new maneuver, it is important to consider the constraints in the real system. For example, the speed change cannot be high because of the dynamic limitations and high fuel consumption, or the altitude change should be proportional to the amount of 2000 ft because of the fact that eastbound flights use odd flight levels, whereas westbound flights use even flight levels with 1000 ft vertical separation during standard cruise operation. In this section, we also perform an analysis about the efficiency of the chosen maneuvers.

A. Example Scenarios

We perform three different scenarios to demonstrate the basic working principles and validity of the autonomous ATCo as illustrated in Fig. 9. In the scenario 1, four aircraft fly towards each other in the FL320 as shown in Fig. 9a, where dashed lines correspond to reference trajectories of the aircraft. The ATCo is activated at $t = 50$ s and it generates the optimum actions, then aircraft follow the intervened trajectories after the ATCo's action request that are direct routing for the green and orange aircraft and altitude change for the red aircraft as illustrated in Fig. 9d. After the ATCo's intervention, the conflicts are resolved and there is no collision during simulation. As presented in Fig. 9d, the ATCo decreases the altitude of the red aircraft to prevent the collision. It prefers descent to climb because of less fuel consumption during descent. The scenario 2 contains an additional aircraft in the airspace as shown in Fig. 9b and this aircraft is in the set of separated aircraft \mathcal{A}_s , which has separation with the rest of the traffic. As presented in Fig. 9e, the ATCo increases the altitude of the red aircraft instead of decrease as in previous scenario. It is shown in this scenario that the intervention to a conflicted aircraft does not cause a loss of separation with a separated aircraft. In the scenario 3, which is illustrated in Fig. 9c, we perform a complex scenario that consists of four conflicted aircraft in the four different flight levels and altitude change of an aircraft can cause another conflict with an aircraft in the other flight levels if this situation is not evaluated during conflict resolution process. The autonomous ATCo resolves the conflicts in this scenario at $t = 50$ s and there is no collision during simulation after conflict resolution as presented in Fig. 9f. Let us focus on the aircraft that operate in FL360. The blue aircraft is climbing from FL340 to FL380 that has a trajectory intersection with the purple aircraft in FL360. The potential collision between them is prevented by giving vector for spacing to the purple aircraft. Because of the vector for spacing, the purple aircraft has also separation with the brown aircraft and the red aircraft in the same flight level. As presented in this example, the conflict resolution mechanism considers climbing and descending traffic when ensuring the safety in a flight level and an intervention to a specific aircraft does not cause an additional conflict with another aircraft. Moreover, as shown in this scenario, the autonomous ATCo can manage a complex traffic that is really challenging for a human operator.

We also perform an additional scenario with different types of aircraft to show how the proposed method handles several aircraft models. We use the same scenario that was presented previously as scenario 1. However, in this case study, the types of aircraft are different. The flight plans are illustrated in Fig. 10a, where Blue is a Boeing 737-800, Red and Orange are Boeing 747-8F, Green is a Cessna Citation CJ3. In the airspace, there are 4 aircraft from 3 different wake turbulence categories. Red and Orange are heavy jets, Blue is a medium aircraft, and Green is a light aircraft. The aircraft are intervened by the ATCo at $t = 50$ s, and the ATCo gives direct routing for the red and orange aircraft and altitude change for the blue aircraft as shown in Fig. 10b. After the intervention, the conflicts are resolved and there is no collision during simulation. Although we use the same scenario in the current case study and the first case study in Fig. 9, the ATCo generates different actions for some of aircraft because of different performance models. For example, the red aircraft obtains altitude change in the first case study, while it obtains direct routing now. In the current scenario, the ATCo intervenes the heavy aircraft (i.e., red and orange) with direct routing, while the light aircraft (i.e., green) has no intervention. And the ATCo decreases the altitude

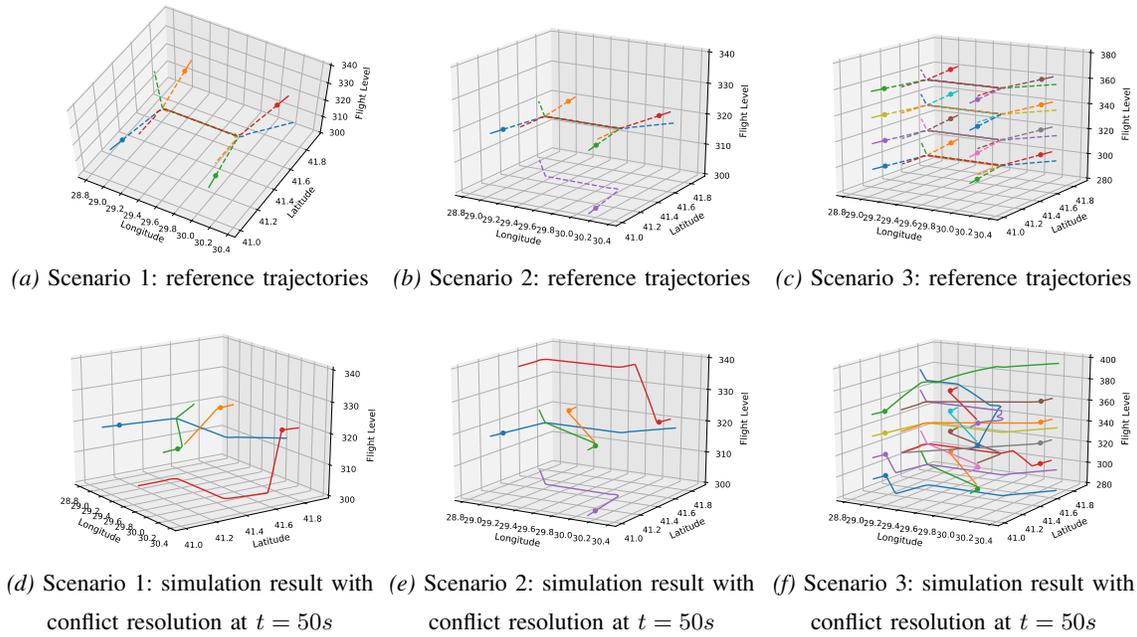


Fig. 9: Example Scenarios.

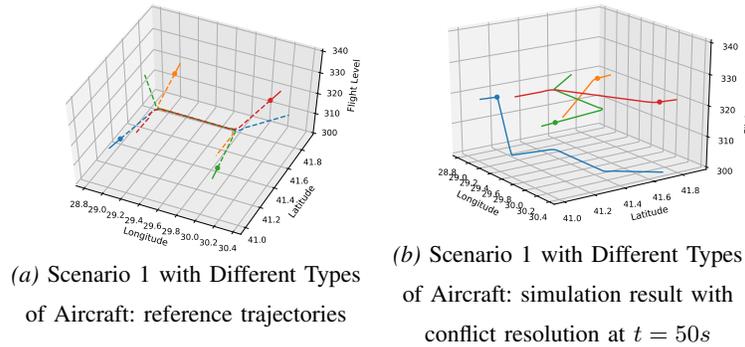


Fig. 10: Example Scenario with Different Types of Aircraft (Blue: Boeing 737-800 / Medium , Red and Orange: Boeing 747-8F / Heavy, Green: Cessna Citation CJ3 / Light).

of the medium aircraft (i.e., blue). The ATCo tries to minimize the total fuel burn in the airspace, so it gives direct routing for the heavy aircraft because of their high fuel consumption, and the light one follows its planned trajectory without intervention because of its low fuel consumption. The altitude of the medium aircraft is decreased to ensure the separation while minimizing the fuel consumption. The priority of the ATCo is determined according to the fuel burn of the aircraft. Therefore, its first action is to give direct routing for the heavy aircraft if there is a violation on the separation requirement. Then, it focuses on the medium aircraft. It is shown in this scenario that the proposed method can efficiently manage a traffic that contains different types of aircraft. The proposed method is decoupled from the performance model, so it can be used with different aircraft models.

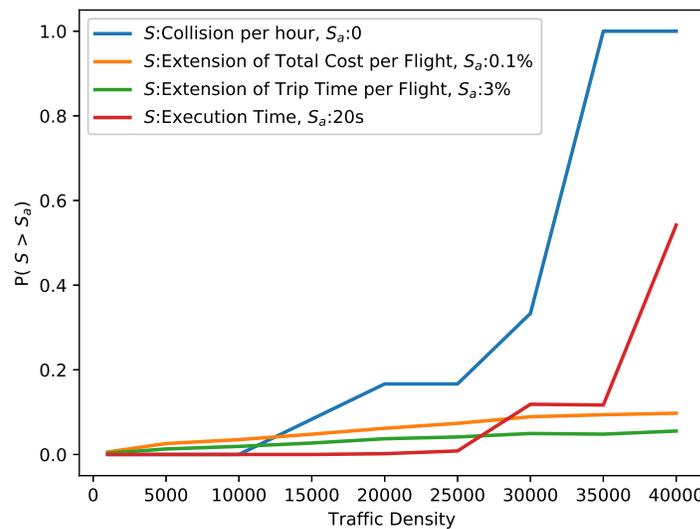


Fig. 11: Airspace Capacity Estimation.

B. Simulations in Stochastic Traffic Environment

We perform simulations for ISTANBUL ACC in the presented stochastic traffic environment with traffic densities from 1000 to 40000 flights per day. During simulations, the optimization-based autonomous ATCo intervenes the traffic every 20 seconds.

The probabilities of the presented metrics with respect to traffic density are obtained from the results of the simulations to redefine the airspace capacity as illustrated in Fig. 11. As shown in this figure, second and third metrics, which are extension of total cost per flight and extension of trip time per flight, do not have a breaking point in the interval between 1000 and 40000 flights per day. Thus, they do not cause a bottleneck to the airspace capacity. The main metric that defines the critical capacity is about safety. The collision per hour has a breaking point around 30000 flights per day. It has a sharp transition in probability for the traffic density range [30000, 35000], whereas the time performance metric has this transition for the range [35000, 40000]. Therefore, the critical capacity of the airspace can be defined as 30000 that is limited by the safety.

The distributions of the number of aircraft that the ATCo operates in the airspace (ISTANBUL ACC) are presented in Fig. 12. The real throughput of the airspace during a standard day is presented in Fig. 12a, where the mean of the distribution is 10.7. In Fig. 12b, the distribution of the throughput during simulations for the breaking point 30000 flights per day is illustrated and the mean of the simulated throughput is 143 aircraft. As a result of the analysis of the distributions, the optimization-based autonomous ATCo can manage traffic approximately 10 times denser than current traffic.

To analyze the efficiency of the chosen maneuvers, the distribution of the given maneuvers during simulations is illustrated in Fig. 13. The traffic density almost has no impact on this distribution. The priority of the ATCo is the

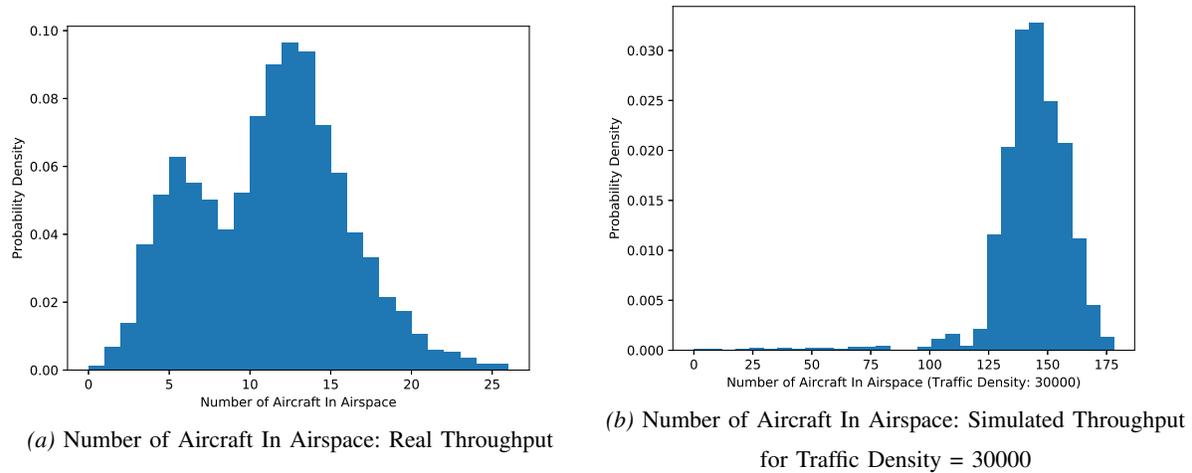


Fig. 12: Distributions for Number of Aircraft in Airspace.

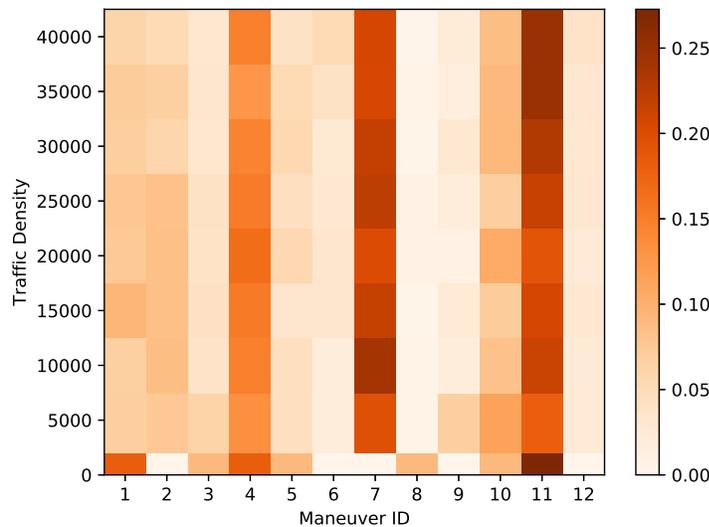


Fig. 13: Maneuver Distribution.

maneuver 11, which is a course change. As mentioned before, a course change can be used to decrease the length of the trajectory while preventing collision. The ATCo usually prefers this maneuver to minimize the cost function. The maneuver 10 is also a course change. However, the sign of the course change angle and the direction of the reference trajectory define the characteristic of the maneuver. It can act like a direct routing or a delaying motion. In analyzed airspace, maneuver 11 is preferred because of the directions of the reference trajectories, however, the ATCo can prefer maneuver 10 to maneuver 11 in another airspace. Therefore, it is important to keep two different course change actions that have opposite signs in the set of maneuvers. The second most frequent maneuver is

maneuver 7, which corresponds to decrease of altitude. The ATCo always prefers descent to the climb because of less fuel consumption and sometimes it is preferred because of performance limitations. The third most preferred maneuver is decrease of speed as maneuver 4. The autonomous ATCo usually prefers decrease of speed to holding pattern or vector for spacing as a delaying motion because of the less impact on the cost. As seen in Fig. 13, the least preferred actions are maneuver 8, 9 and 6. Both maneuver 8 and 9 are vector for spacing, and speed change dominates them. The maneuver 6 is increase of the altitude as $H_c = 4000$ ft, and this action is dominated by other altitude change maneuvers. The least used maneuvers, especially maneuver 8, can be removed from the set of maneuvers or changed with more efficient maneuvers to improve the performance of the optimization process and increase the airspace capacity.

VII. CONCLUSIONS AND FUTURE WORK

This paper presented an optimization-based autonomous air traffic control system to improve airspace capacity. Although the proposed system is presented as a fully autonomous system, it can also be used as a semi-autonomous system for providing decision support to human air traffic operators. It is not easy to say which automation level is better. But enhanced automation support is inevitable to accommodate the increase in the air traffic volume. Therefore, we designed a system that can be used in different automation levels. It is up to the authorities to choose the automation level according to the preferences of the stakeholders and other factors. The aviation authorities can choose the automation level, and then the proposed system can be used in the specified automation level. The proposed system is based on an integer linear programming formulation, constructed via a mapping process that discretizes the airspace, with predicted trajectories to ensure the safety. A trajectory prediction method was also introduced. We showed that the proposed approach is scalable for large-scale ATM scenarios. The sub-components of the proposed system can also be used individually in different ATM applications. For example, the trajectory prediction model can be used when dealing with trajectory management, or the mapping process can be utilized when dealing with mitigation of airspace congestion. Moreover, we defined an airspace capacity estimation procedure to determine the capacity of an airspace with the proposed ATC system. We showed that the proposed approach can manage traffic approximately 10 times denser than current traffic in ISTANBUL ACC. In a different airspace, the capacity improvement could be different because of the airways and the airspace structure. However, we clearly showed that the capacity of a dense airspace can be improved approximately 10 times via the proposed system. Note that when the proposed system is used in a semi-autonomous manner, a set of parameters can also be considered with regard to the human controller's state such as fatigue and workload to specify the system's limit.

One of the main advantages of the proposed system is its high scalability that leads to improvement of airspace capacity. The other one is that it can be used easily in the existing ATM system. Another good characteristic of the proposed system is the utilization of realistic models that prevent to endanger the operation as a result of inaccurate predictions. Besides, the optimization process improves the efficiency of the operation. However, there are some drawbacks of the proposed system. In the current form, it could not handle different multi-agent systems, which contain multiple vehicles with assigned tasks, without adapting the maneuvers. It is also necessary to present a new set of maneuvers to manage the traffic in an approach and terminal control area.

Future research will extend the method to handle different multi-agent systems. By adapting the maneuvers, the method can be used for managing approach traffic. The method can also be utilized with multi-quadrotor systems when the aircraft dynamics and maneuvers are modified. Furthermore, benefiting from the proposed ATC system, we expect to investigate alternative air traffic complexity metrics. Besides, we also expect to construct a detailed wind model by evaluating the wind as a function of altitude and environmental variables and generating a wind map to improve the precision of trajectory prediction under impact of wind. Another future study could be the improvement of the aircraft model for more precise trajectory prediction. Benefiting from real flight records, the BADA model could be enhanced to improve the performance of trajectory prediction.

ACKNOWLEDGMENT

The authors disclosed that Barış Başpınar was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) with International Doctoral Research Fellowship.

REFERENCES

- [1] Airbus, "Global market forecast: Mapping demand 2016 - 2035," Airbus, Tech. Rep. D14029463, August 2016.
- [2] W. E. Kelly, "Conflict detection and alerting for separation assurance systems," in *Gateway to the New Millennium. 18th Digital Avionics Systems Conference. Proceedings (Cat. No. 99CH37033)*, vol. 2, St. Louis, MO, USA, 1999, pp. 6–D.
- [3] J. M. Shewchun, J.-H. Oh, and E. Feron, "Linear matrix inequalities for free flight conflict problems," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 3, San Diego, CA, USA, 1997, pp. 2417–2422.
- [4] R. A. Paielli and H. Erzberger, "Conflict probability for free flight," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 588–596, 1997.
- [5] B. Sridhar and G. Chatterji, "Computationally efficient conflict detection methods for air traffic management," in *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*, vol. 2, Albuquerque, NM, USA, 1997, pp. 1126–1130.
- [6] A. Vink, S. Kauppinen, J. Beers, and K. de Jong, "Medium term conflict detection in eatchip phase iii," in *16th DASC. AIAA/IEEE Digital Avionics Systems Conference. Reflections to the Future. Proceedings*, vol. 2, Irvine, CA, USA, 1997, pp. 9–3.
- [7] G. Bakker and H. A. Blom, "Air traffic collision risk modelling," in *Proceedings of the 32nd IEEE Conference on Decision and Control*, San Antonio, TX, USA, 1993, pp. 1464–1469.
- [8] W. Liu and I. Hwang, "Probabilistic trajectory prediction and conflict detection for air traffic control," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 6, pp. 1779–1789, 2011.
- [9] Y. Matsuno and T. Tsuchiya, "Probabilistic conflict detection in the presence of uncertainty," in *Air traffic management and systems*. Springer, 2014, pp. 17–33.
- [10] M. S. Eby, "A self-organizational approach for resolving air traffic conflicts." *Lincoln Laboratory Journal*, vol. 7, no. 2, pp. 239–254, 1994.
- [11] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA, 1991, pp. 1398–1404.
- [12] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–521, 1998.
- [13] C. Tomlin, I. Mitchell, and R. Ghosh, "Safety verification of conflict resolution manoeuvres," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, no. 2, pp. 110–120, 2001.
- [14] A. Bayen, P. Grieder, and C. Tomlin, "A control theoretic predictive model for sector-based air traffic flow," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, USA, 2002, p. 5011.
- [15] N. Durand, J.-M. Alliot, and G. Granger, "Optimal resolution of en route conflicts," in *Proceedings of the 1st USA/Europe Seminar*, Saclay, France, 1997.
- [16] E. Frazzoli, Z.-H. Mao, J.-H. Oh, and E. Feron, "Resolution of conflicts involving many aircraft via semidefinite programming," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 1, pp. 79–86, 2001.

- [17] L. Pallottino, E. M. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 3–11, 2002.
- [18] Y. Yang, J. Zhang, K.-Q. Cai, and M. Prandini, "Multi-aircraft conflict detection and resolution based on probabilistic reach sets," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 1, pp. 309–316, 2016.
- [19] A. Alonso-Ayuso, L. F. Escudero, P. Olaso, and C. Pizarro, "Conflict avoidance: 0-1 linear models for conflict detection & resolution," *Top*, vol. 21, no. 3, pp. 485–504, 2013.
- [20] R. Conde, D. Alejo, J. A. Cobano, A. Viguria, and A. Ollero, "Conflict detection and resolution method for cooperating unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 495–505, 2012.
- [21] B. Baspinar, C. Pasaoglu, N. K. Ure, and G. Inalhan, "Infrastructure development for ground-based separation assurance with optional automation," in *Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems*, Toulouse, France, 2015, pp. 65–74.
- [22] C. Pasaoglu, B. Baspinar, N. K. Ure, and G. Inalhan, "Hybrid systems modeling and automated air traffic control for three-dimensional separation assurance," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace and Astronautics*, vol. 230, no. 9, pp. 1788–1809, 2016.
- [23] L. C. Yang and J. K. Kuchar, "Prototype conflict alerting system for free flight," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 4, pp. 768–773, 1997.
- [24] H. Erzberger, T. J. Davis, and S. Green, "Design of center-tracon automation system," 1993.
- [25] D. Brudnicki and A. McFarland, "User request evaluation tool (uret) conflict probe performance and benefits assessment," in *Proc. USA/Europe ATM Seminar*, Saclay, France, 1997.
- [26] B. Hilburn, "Cognitive complexity in air traffic control: A literature review," *EEC note*, vol. 4, no. 04, 2004.
- [27] J. N. Warfield and A. R. Cárdenas, *A handbook of interactive management*. Iowa State University Press Ames, 1994.
- [28] J. M. Histon and R. J. Hansman, "Mitigating complexity in air traffic control: the role of structure-based abstractions," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2008.
- [29] I. V. Laudeman, S. Shelden, R. Branstrom, and C. Brasil, "Dynamic density: An air traffic management metric (nasa/tm-1998-112226)," *Moffett Field, CA: NASA Ames*, 1998.
- [30] P. Kopardekar, A. Schwartz, S. Magyarits, and J. Rhodes, "Airspace complexity measurement: An air traffic control simulation analysis," in *7th USA/Europe Air Traffic Management R&D Seminar*, Barcelona, Spain, 2007.
- [31] G. Chatterji and B. Sridhar, "Measures for air traffic controller workload prediction," in *1st AIAA, Aircraft, Technology Integration, and Operations Forum*, Los Angeles, CA, USA, 2001, p. 5242.
- [32] S. Mondoloni, "Airspace fractal dimensions and applications," in *The 4th ATM Seminar*, Santa Fe, NM, USA, 2001.
- [33] K. Lee, E. Feron, and A. Pritchett, "Air traffic complexity: an input-output approach," in *American Control Conference*, New York, NY, USA, 2007, pp. 474–479.
- [34] M. Uzun, B. Başpınar, E. Koyuncu, and G. İnalanhan, "Takeoff weight error recovery for tactical trajectory prediction automaton of air traffic control operator," in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, St. Petersburg, FL, USA, 2017, pp. 1–7.
- [35] B. Başpınar, H. Balakrishnan, and E. Koyuncu, "Mission planning and control of multi-aircraft systems with signal temporal logic specifications," *IEEE Access*, vol. 7, pp. 155 941–155 950, 2019.
- [36] B. Baspinar, M. Uzun, A. F. Guven, T. Basturk, I. Tasdelen, E. Koyuncu, and G. Inalhan, "A 4d trajectory generation infrastructure tool for controller working position," in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, St. Petersburg, FL, USA, 2017, pp. 1–10.
- [37] "User manual for the base of aircraft data (bada) revision 3.11," Eurocontrol Experimental Centre, Tech. Rep., 2013.
- [38] J. Lygeros, C. Tomlin, and S. Sastry, "Hybrid systems: modeling, analysis and control," *preprint*, 1999.
- [39] "User manual for the base of aircraft data (bada) family 4," Eurocontrol Experimental Centre, Tech. Rep., 2014.
- [40] P. Sujit, S. Saripalli, and J. Sousa, "An evaluation of uav path following algorithms," in *2013 European Control Conference (ECC)*, Zurich, Switzerland, 2013, pp. 3332–3337.
- [41] V. Bulusu and V. Polishchuk, "A threshold based airspace capacity estimation method for uas traffic management system," in *IEEE Systems Conference*, Montreal, QC, Canada, 2017.



Baris Baspinar received the B.S. degrees in aeronautical engineering and control and automation engineering and the M.S. degree in aerospace engineering from Istanbul Technical University (ITU), Istanbul, Turkey, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree in aerospace engineering.

He was a Visiting Researcher with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology (MIT), from 2018 to 2019. He has been a Research Assistant with the Controls and Avionics Research Group with the ITU Aerospace Research Center (ITU ARC), since 2013, and a Teaching Assistant with the Department of Aeronautical Engineering, ITU, since 2017. His current research interests include data-driven ATM network modelling, high-level autonomy in air traffic control systems, resiliency problem in large scale complex systems, conflict detection and separation assurance algorithms, air traffic complexity, epidemic spreading and control of infections on network, survivability analysis for complex air combat missions, air combat engagement modelling, guidance navigation and control problems in aerial vehicles.



Hamsa Balakrishnan received the B.Tech. degree in aerospace engineering from the IIT Madras, in 2000, and the Ph.D. degree in aeronautics and astronautics from Stanford University, Stanford, CA, in 2006.

She was a Principal Development Engineer at the University Affiliated Research Center (UC Santa Cruz) and the NASA Ames Research Center's Terminal Air Traffic Management Concepts Branch. She is currently a Professor and the Associate Department Head of Aeronautics and Astronautics at the Massachusetts Institute of Technology (MIT), Cambridge. She is also the Director of transportation with MIT. Her current research interests include the design, analysis, and implementation of control and optimization algorithms for large-scale cyber-physical infrastructures, with an emphasis on air transportation systems. These include airport congestion control algorithms, air traffic routing and airspace resource allocation methods, machine learning for weather forecasts and flight delay prediction, and methods to mitigate environmental impacts.



Emre Koyuncu received the B.S. degree in electrical engineering, the M.S. degree in mechatronics engineering, and the Ph.D. degree in aerospace engineering from Istanbul Technical University (ITU), Istanbul, Turkey, in 2005, 2008, and 2015, respectively.

He was a Visiting Researcher at the Boeing Research and Technology of Europe from 2013 to 2014, and the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology (MIT), from 2014 to 2015. He is currently an Assistant Professor with the Department of Aeronautical Engineering, ITU. He is also the director of the ITU Aerospace Research Center (ITU ARC), Controls and Avionics Research Group. His current research interests include aeronautics, robotics, and navigation and control theory with the application areas of agile unmanned aerial vehicles, flight trajectory optimization and planning, airborne conflict avoidance and resolution, high-level autonomy in air traffic control systems, flight management and decision support systems. In particular, he focuses on the implementation of optimization-based control and probability theory for the design and analysis of high-performance cyber-physical systems