

Received September 28, 2019, accepted October 16, 2019, date of publication October 24, 2019, date of current version November 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949426

Mission Planning and Control of Multi-Aircraft Systems With Signal Temporal Logic Specifications

BARIŞ BAŞPINAR^{1,2}, **HAMSA BALAKRISHNAN**², AND **EMRE KOYUNCU**¹

¹Department of Aeronautical Engineering, Istanbul Technical University, 34469 Istanbul, Turkey

²Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Corresponding author: Barış Başpınar (baspinarb@itu.edu.tr)

The work of B. Başpınar was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) with the International Doctoral Research Fellowship.

ABSTRACT This paper focuses on optimization-based control of multi-aircraft systems that have several mission objectives. Signal Temporal Logic (STL) is used to express the mission specifications that combine temporal and logical constraints. A methodology is presented to construct an optimization problem in the form of Mixed-Integer Linear Programming (MILP) by using the differential flatness property of a nonlinear dynamical system and STL specifications to generate feasible trajectories. Contrary to general implementations of Temporal Logic to discrete-time systems, the proposed method deals with continuous-time systems. It can be used to find optimal control strategies to achieve the assigned tasks for nonlinear dynamical systems without discretizing the system dynamics. As an illustration, we present an air traffic control example. The nonlinear dynamical model for the aircraft is represented as a partially differentially flat system, and the presented method is applied to manage approach control and to solve the arrival sequencing problem. The method is also applied with a quadrotor fleet to show that the method can be used with different multi-agent systems.

INDEX TERMS Signal temporal logic, multi-aircraft systems, air traffic control, optimization-based control, differential flatness.

I. INTRODUCTION

Mission planning and control of multi-aircraft systems involve several temporal and logical constraints. These constraints can naturally be specified using Temporal Logic (TL), a system of rules and symbolism. STL is an extension of TL in which temporal operators also contain timing constraints for specifying properties of real-valued signals [1]. When dealing with continuous systems, STL is convenient to specify these constraints.

Temporal logic has been studied as a formal language for specifying system behaviors and complex tasks. Linear Temporal Logic (LTL) has been employed as a tool for specifying the restrictions in discrete-time dynamical systems [2]. The LTL specifications are represented as mixed-integer linear constraints to generate optimal control strategy for a discrete-time system via Mixed-Integer Programming (MIP). STL

specifications have been encoded as mixed-integer linear constraints to specify system's properties such as safety and response for model predictive control of discrete-time systems [3]. In both the above studies, it is required to add decision variables at every time step for a specific temporal logic constraint. Therefore, these approaches do not scale well. There are also studies that present heuristics in order to reduce MIP complexity in these approaches. The authors of [4] propose a heuristic to add constraints when necessary instead of adding auxiliary decision variables at every time step. There are also studies such as [5] that evaluate the problem as a non-convex optimization problem to generate trajectories with Metric Temporal Logic (MTL) specifications. Consequently, the system can have nonlinear dynamics but it should be discretized. Stochastic heuristics have also been proposed for finding system behaviors that falsify a temporal logic property. For example, the study [6] presents a Monte-Carlo technique for finding counterexamples to MTL properties. The algorithm can be applied to nonlinear dynamical systems

The associate editor coordinating the review of this manuscript and approving it for publication was Xiwang Dong.

in continuous-time. However, it is a sampling-based method without guarantees and it is hard to apply this approach to multi-agent systems. Temporal logic specifications have also been studied with multi-agent systems. The desired behavior of a group of agents is specified with variants of TL in the studies [7]–[9] that evaluate the problem in a grid-based environment using discrete abstractions of the system dynamics. It is common practice in many studies to use discrete abstractions when dealing with temporal logic for multi-agent systems. These studies provide correctness guarantees for the discrete behavior. However, it is a simplified version of the real system. In the study [10], the authors focus on mission planning of multi-quadrotor systems. Using a trajectory generator, they construct a non-convex optimization problem to obtain trajectories that satisfy the STL specifications in continuous-time. However, the study mainly focuses on quadrotors, and it decouples the equations of motion along three orthogonal axes. Because of this decoupling, the presented method cannot be implemented to fixed-wing aircraft. Moreover, the study mainly focuses on the position information, and STL specifications cannot be given in terms of the angles. However, sometimes it may be necessary to specify orientations. For example, in arrival sequencing, it may be necessary to align the aircraft with respect to the runway's direction.

In this paper, we propose a methodology to overcome these limitations. We encode the missions of the multiple aircraft as STL specifications. Then, using differential flatness theory, we construct an optimization process to generate optimal strategies for multiple aircraft to satisfy the STL specifications, which corresponds to completing the assigned tasks. The proposed method generates control inputs as continuous real valued functions, and it generates feasible trajectories that satisfy the missions and performance limitations. We focus on air traffic control tasks using a realistic nonlinear aircraft model to illustrate our approach. We also simulate a case study with a quadrotor fleet to show the generalizability of the proposed method to other multi-agent systems.

As mentioned before, the majority of the existing studies such as [2]–[5] discretize the system dynamics to ensure the STL specifications in discrete-time. One of the contributions of this study is that the proposed method generates feasible trajectories in continuous-time that satisfy the tasks described via STL, without discretizing the system dynamics. Although some of the constraints in the optimization problem are enforced only at the sampled times, this sampling does not jeopardize continuous-time satisfiability. Compared to the existing MILP-based approaches such as [2], [3], the developed method has better scalability to deal with nonlinear system dynamics, because it is not necessary to add excessive auxiliary decision variables. The method can be used with different multi-agent systems such as a fleet of fixed-wing aircraft or multi-quadrotor systems. Because of this generalizability, it overcomes the restriction of the studies such as [10] that are developed for specific systems. The proposed method fills a gap in the literature by showing that a MILP based

approach works well for the STL satisfaction of differentially flat nonlinear systems. The study presents a convenient way to use the flatness property of a nonlinear dynamical system to satisfy STL specifications. To the best of our knowledge, no other study in the literature presents a convenient way to form a MILP that is used to guarantee the continuous-time satisfiability of the STL specifications for nonlinear continuous systems. Moreover, the method enables us to use realistic nonlinear dynamical models when evaluating the complex missions of multi-agent systems, contrary to many existing studies such as [7]–[9] that use discrete abstractions of the system dynamics and grid-based environments for mission planning of multi-agent systems. The presentation of the partially-flat aircraft model that can be used in Air Traffic Management (ATM) applications is also a contribution.

The paper is organized as follows. Section II presents the system behavior and STL specification formalism. Section III introduces (partially) differentially flat systems and explains motion planning for flat systems to generate feasible trajectories that satisfy STL specifications. The aircraft dynamics is expressed as a partially differentially flat system in Section IV. Section V explains the details of the optimization process. Finally, examples are given in Section VI.

II. SYSTEM BEHAVIOR AND SIGNAL TEMPORAL LOGIC

We consider the continuous-time dynamical systems of the form:

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

where $x(t) \in \mathbb{R}^n$ is the vector of system states, $u(t) \in \mathbb{R}^m$ is the vector of control inputs and $x_0 \in \mathbb{R}^n$ is the initial state of the system. A state trajectory x is a vector of continuous-time signals, and this trajectory is derived from an action trajectory by running the system model (1). An action trajectory contains the control inputs for a specific time period $[0, T]$, and the state trajectory is generated for this finite time period.

A. SIGNAL TEMPORAL LOGIC

The desired system behaviors can be specified using Signal Temporal Logic (STL) [1]. In this study, we use the future fragment of STL, which does not contain the since operator. The set of formulas of STL can be recursively defined by:

$$\psi ::= \top \mid \mu \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \mathcal{U}_{[a,b]} \psi_2$$

where ψ is an STL formula, and μ is an atomic predicate whose value depends on the sign of a function of x . \top is the Boolean True. \neg , \wedge , and \mathcal{U} are the negation, conjunction, and until operators, respectively. The other connectives can be defined with regard to these operators. The following identity allows to define the disjunction (\vee) in terms of the negation and the conjunction, $\psi_1 \vee \psi_2 = \neg(\neg\psi_1 \wedge \neg\psi_2)$. The operators eventually ($\diamond_{[a,b]}$) and always ($\square_{[a,b]}$) can be defined as $\diamond_{[a,b]}\psi = \top \mathcal{U}_{[a,b]}\psi$ and $\square_{[a,b]}\psi = \neg\diamond_{[a,b]}\neg\psi$, respectively. Additionally, the operators implication (\Rightarrow) and equivalency (\Leftrightarrow) can be presented as $\psi_1 \Rightarrow \psi_2 = \neg\psi_1 \vee \psi_2$ and $\psi_1 \Leftrightarrow \psi_2 = (\psi_1 \Rightarrow \psi_2) \wedge (\psi_2 \Rightarrow \psi_1)$, respectively.

The validity of a formula ψ with respect to signal x at time t is defined as follows:

$$(x, t) \models \top \text{ iff } \top$$

$$(x, t) \models \mu \text{ iff } \mu(x(t)) \geq 0$$

$$(x, t) \models \neg\psi \text{ iff } (x, t) \not\models \psi$$

$$(x, t) \models \psi_1 \wedge \psi_2 \text{ iff } (x, t) \models \psi_1 \text{ and } (x, t) \models \psi_2$$

$$(x, t) \models \psi_1 \mathcal{U}_{[a,b]} \psi_2 \text{ iff } \exists s \in [t+a, t+b], (x, s) \models \psi_2 \\ \text{and } \forall s' \in [t, s], (x, s') \models \psi_1$$

The trajectory x satisfies the formula ψ if and only if $(x, t) \models \psi$. Additionally, the semantics of the operators eventually and always can be given as follows:

$$(x, t) \models \diamond_{[a,b]} \psi \text{ iff } \exists s \in [t+a, t+b], (x, s) \models \psi$$

$$(x, t) \models \square_{[a,b]} \psi \text{ iff } \forall s \in [t+a, t+b], (x, s) \models \psi$$

B. ROBUST STL SPECIFICATIONS

The robust semantics of STL ([11], [12]) can be used to give the system the ability of tolerating perturbations. The robustness of STL formula ψ can be specified via a function $\rho^\psi(x, t)$ that is defined recursively as follows:

$$\rho^\mu(x, t) = \mu(x(t))$$

$$\rho^{\neg\psi}(x, t) = -\rho^\psi(x, t)$$

$$\rho^{\psi_1 \wedge \psi_2}(x, t) = \min(\rho^{\psi_1}(x, t), \rho^{\psi_2}(x, t))$$

$$\rho^{\psi_1 \mathcal{U}_{[a,b]} \psi_2}(x, t) = \max_{s \in [t+a, t+b]} \left(\min(\rho^{\psi_2}(x, s), \right. \\ \left. \min_{s' \in [t, s]} (\rho^{\psi_1}(x, s')) \right)$$

For any signal x and STL formula ψ , x satisfies ψ at time t if $\rho^\psi(x, t) > 0$ such that $\rho^\psi(x, t) > 0 \Rightarrow (x, t) \models \psi$. The magnitude of the $\rho^\psi(x, t)$ quantifies the robustness for the formula ψ .

III. MOTION PLANNING WITH STL SPECIFICATIONS FOR FLAT SYSTEMS

The motion planning problem corresponds to finding a trajectory $t \mapsto (x(t), u(t))$ from a set of specific initial conditions to a defined final state while satisfying the system dynamics $\dot{x} = f(x, u)$. If some STL specifications are added as constraints on the trajectory, the problem is transformed into a motion planning with constraints. In the general case, this problem can be quite difficult because it requires the integration of the system equations to find the sequence of control inputs that satisfies the initial conditions, final conditions and constraints. For nonlinear systems, it may pose some additional problems [13].

The trajectory generation is particularly easy for the differentially flat systems. The dynamical system (1) is differentially flat if there exist relations ([14]–[16])

$$\begin{aligned} \zeta &: \mathbb{R}^n \times (\mathbb{R}^m)^{r+1} \rightarrow \mathbb{R}^m, \\ \eta &: (\mathbb{R}^m)^r \rightarrow \mathbb{R}^n, \text{ and} \\ \kappa &: (\mathbb{R}^m)^{r+1} \rightarrow \mathbb{R}^m. \end{aligned} \quad (2)$$

such that

$$z = \zeta(x, u, \dot{u}, \dots, u^{(r)}), \quad (3)$$

$$x = \eta(z, \dot{z}, \dots, z^{(r-1)}), \text{ and} \quad (4)$$

$$u = \kappa(z, \dot{z}, \dots, z^{(r-1)}, z^{(r)}). \quad (5)$$

where ζ, η, κ are smooth functions, and z is the flat output vector. This means that all system dynamics can be expressed as a function of the flat outputs and their derivatives. This model is equivalent to (1) and can be used to efficiently generate trajectories. The equations (4) and (5) yield that for every given trajectory of the flat output $t \mapsto z(t)$, the evolution of all other variables of the system $t \mapsto x(t)$ and $t \mapsto u(t)$ is also determined without integration of the system of differential equations. Moreover, given a sufficiently smooth trajectory for the flat output $t \mapsto z^*(t)$, equation (5) can be used to generate the corresponding feedforward u^* directly.

Let us suppose that all system variables cannot be expressed as a function of the flat outputs and their derivatives. The dynamical system (1) is partially differentially flat if a partition of the system variables (x_d, u_d) can be expressed as in equation (4) and (5) via the set of smooth functions η_d, κ_d , while the rest of the system variables (x_n, u_n) are presented in the following form:

$$\dot{x}_n = \alpha(x_n, u_n, z, \dot{z}, \dots, z^{(r-1)}), \text{ and} \quad (6)$$

$$u_n = \beta(x_n, z, \dot{z}, \dots, z^{(r-1)}, z^{(r)}). \quad (7)$$

In this case, it is also possible to generate trajectories for the system variables (x_d, u_d) such that $t \mapsto x_d(t)$ and $t \mapsto u_d(t)$ from a given trajectory $t \mapsto z(t)$ without numerical integration. However, it is necessary to integrate the equation (6) to generate the trajectories for the system variables (x_n, u_n) such that $t \mapsto x_n(t)$ and $t \mapsto u_n(t)$.

Let us define each element of the flat output z as a linear combination of certain basis functions of the time, i.e.,

$$z_i = \sum_{j=1}^{N_i} c_{ji} \phi_{ji}(t) \quad (8)$$

where, c_{ji} is a weighting coefficient or control point, $\phi_{ji}(t)$ is a basis function and $z = [z_1, z_2, \dots, z_s]^T$.

There are several candidates that can be used as basis functions. In this study, B-spline basis functions are used to present the flat outputs. Let p be a nonnegative integer and let $\mathbb{T} = \{\tau_0, \tau_1, \dots, \tau_m\}$, the knot vector, be a nondecreasing sequence of real numbers. The q^{th} B-spline basis function of p -degree, denoted by $N_{q,p}(t)$, is defined as [17]:

$$N_{q,0}(t) = \begin{cases} 1 & \text{if } \tau_q \leq t < \tau_{q+1} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$N_{q,p}(t) = \frac{t - \tau_q}{\tau_{q+p} - \tau_q} N_{q,p-1}(t) + \frac{\tau_{q+p+1} - t}{\tau_{q+p+1} - \tau_{q+1}} N_{q+1,p-1}(t) \quad (10)$$

The B-spline functions yield some geometric properties [17] that affect the performance of the trajectory generation positively when dealing with the dynamical systems.

For example, a B-spline curve such as z_i in (8) is C^{p-1} in any $t \in \mathbb{T}$, and C^∞ otherwise. It is a continuous and differentiable function. The higher order derivatives of B-spline basis functions can be presented as linear combinations of B-splines of lower order, and the weighting coefficients (or control points) have linear impact on the higher order derivatives of the curve such that $z_i^{(k)} = \sum_{j=1}^{N_i} c_{ji} \phi_{ji}^{(k)}(t)$, where $\phi_{ji}^{(k)}(t)$ is k -order derivative of basis function. Moreover, the B-spline functions satisfy that $N_{q,p}(t) \geq 0$ and $\sum_{q=0}^n N_{q,p}(t) = 1, \forall t \in [\tau_0, \tau_m]$, and they have local support such that $N_{q,p}(t) \neq 0$ iff $t \in [\tau_q, \tau_{q+p}]$. Hence, the adjustment of a specific weighting coefficient leads to change shape of a specific region of the curve without affecting the rest of it.

Let C be the set of the control points that configure the flat output for the dynamical system (1). Equation (8) and a given C construct the trajectory of the flat output $t \mapsto z^C(t)$. The trajectory $t \mapsto z^C(t)$ can be modified by changing the coefficients in C , where each coefficient is a control point that shapes a specific region of the trajectory. However, some coefficient sets can generate infeasible trajectories because of the violation of the dynamical constraints or STL formulas. Using the analytical expressions in (4) and (5), the trajectories of the system variables $t \mapsto x^C(t)$ and $t \mapsto u^C(t)$ can be expressed in terms of C . Then, the violation of any constraint can be checked via these system trajectories. By tuning the coefficients, these violations can be removed. Because of the local support property, each coefficient has impact on a specific region of the trajectory. Therefore, the violations can be removed by changing only some specific coefficients without modifying all of them. The tuning process can be achieved via an algorithm or an optimization problem. In this study, we construct an optimization problem to find the proper control points that generate feasible trajectories with STL specifications. When dealing with partially differentially flat systems, we avoid presenting the dynamical constraints and STL specifications in terms of the system variables (x_n, u_n) during the trajectory planning to prevent the costs that originate from integrating the differential equations. All constraints and STL specifications are expressed with respect to the system variables (x_d, u_d) that form the differentially flat part of the system. However, the system variables (x_n, u_n) are generated from the flat output's trajectory $t \mapsto z^C(t)$ after the set C is determined.

IV. AIRCRAFT DYNAMICS AS A PARTIALLY DIFFERENTIALLY FLAT SYSTEM

In this study, we use the following set of equations as the dynamical system (1) to model the aircraft dynamics that was used in ([18]–[20]) for air traffic control applications.

$$\dot{x}_1 = x_4 \cos(x_5) \cos(u_3) \tag{11}$$

$$\dot{x}_2 = x_4 \sin(x_5) \cos(u_3) \tag{12}$$

$$\dot{x}_3 = x_4 \sin(u_3) \tag{13}$$

$$\dot{x}_4 = -\frac{C_D S \rho x_4^2}{2x_6} - g \sin(u_3) + \frac{u_1}{x_6} \tag{14}$$

$$\dot{x}_5 = \frac{C_L S \rho x_4}{2x_6} \frac{\sin(u_2)}{\cos(u_3)} \tag{15}$$

$$\dot{x}_6 = -F \tag{16}$$

The model is a nonlinear dynamical system, where the control inputs are the engine thrust (u_1), bank angle (u_2), and flight path angle (u_3), and the state variables are the horizontal position (x_1 and x_2), altitude (x_3), true airspeed (x_4), heading angle (x_5), and the mass of the aircraft (x_6). In the equation set above, aerodynamic lift and drag coefficients are denoted by C_L and C_D , gravitational acceleration is g , 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, flight path angle and mass are obtained from the Base of Aircraft Data (BADA) [21].

The lift coefficient and drag coefficient are expressed as follows:

$$C_L = \frac{2mg \cos(\gamma)}{\rho V^2 S \cos(\phi)} \tag{17}$$

$$C_D = C_{D0} + C_{D2} C_L^2 \tag{18}$$

where C_{D0} and C_{D2} are constants specified in the database. γ, ϕ and V are the flight path angle, bank angle and speed, respectively.

The dynamical system (11)–(16) is partially differentially flat, when the flat output z is expressed as follows:

$$z = [x_1, x_2, x_3]^T \tag{19}$$

The differentially flat part of the system consists of the system variables $x_1, x_2, x_3, x_4, x_5, u_2$ and u_3 . Let us give the flat descriptions of these variables. The first three variables directly correspond to the flat outputs. Using Equation (11), (12) and (13), the fourth variable x_4 can be expressed as the Euclidean norm of the derivatives of the flat outputs as follows:

$$x_4 = \sqrt{\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2} \tag{20}$$

The variable x_5 can be described in terms of the derivatives of the first two flat outputs by dividing Equation (12) to Equation (11), as:

$$x_5 = \arctan\left(\frac{\dot{z}_2}{\dot{z}_1}\right) \tag{21}$$

By modifying Equation (13), the variable u_3 can be given by:

$$u_3 = \arcsin\left(\frac{\dot{z}_3}{\sqrt{\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2}}\right) \tag{22}$$

The variable \dot{x}_5 can be obtained from Equation (21) and the lift coefficient C_L is presented in Equation (17). By writing the expressions for \dot{x}_5 and C_L into Equation (15), the variable u_2 can be formulated as follows:

$$u_2 = \arctan\left(\frac{(\ddot{z}_2 \dot{z}_1 - \dot{z}_2 \ddot{z}_1) \sqrt{\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2}}{g(\dot{z}_1^2 + \dot{z}_2^2)}\right) \tag{23}$$

The rest of the system variables u_1, x_6 cannot be expressed only in terms of the flat outputs and their derivatives. These variables can be presented in the form of (6) and (7). The acceleration \dot{x}_4 can be derived from Equation (20) and the variables x_4, u_3 are presented before. By putting these expressions into Equation (14), the engine thrust u_1 can be given as follows:

$$u_1 = x_6 \left(\frac{\dot{z}_1 \ddot{z}_1 + \dot{z}_2 \ddot{z}_2 + \dot{z}_3 (g + \ddot{z}_3)}{\sqrt{\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2}} \right) + 0.5 C_{D5} (\dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2). \quad (24)$$

The value of u_1 depends on the variable x_6 . To calculate x_6 , the fuel consumption F must be expressed. In BADA, there are different functions that are used to determine the fuel consumption, and these functions are presented with regard to the aircraft type and flight phase. For a jet aircraft in the descent phase, the fuel consumption is as follows:

$$F = C_{f3} \left(1 - \frac{H_p}{C_{f4}} \right) \quad (25)$$

where C_{f3} and C_{f4} are constants, and H_p is geopotential pressure altitude. Using this expression, the derivative of the mass can be presented as a function of the altitude:

$$\dot{x}_6 = \alpha_{11}(z_3) \quad (26)$$

A function of the altitude z_3 , speed x_4 and thrust u_1 can be used to generalize the calculation of the fuel consumption. Then, the following equation covers the operation in any flight phase:

$$\dot{x}_6 = \alpha_{12}(z_3, \dot{z}_1, \dot{z}_2, \dot{z}_3, u_1) \quad (27)$$

Both Equation (26) and Equation (27) are in the form of (6). Therefore, the dynamical system (11)-(16) is a partially differentially flat system, where $x_d = [x_1, x_2, x_3, x_4, x_5]$, $u_d = [u_2, u_3]$, $x_n = x_6$, $u_n = u_1$ with the flat descriptions (19)-(27).

V. OPTIMIZATION-BASED CONTROL OF MULTIPLE AIRCRAFT WITH STL SPECIFICATIONS

This section presents the optimization problem that is used to generate the feasible trajectories with STL specifications to control multiple aircraft.

Let A be the set of aircraft, whose dynamics are presented with the flat descriptions (19)-(27), and each flat output is expressed as in Equation (8). Let C^A be the set that contains all control points for all aircraft in A . Then, by specifying the control points in C^A as decision variables, the following optimization problem can be formulated to generate the optimum trajectories and control inputs that ensure the given STL formulas:

$$\max_{C^A} \rho^\psi(z) \quad (28)$$

$$\text{subject to } d(z) \geq 0 \quad (29)$$

$$b(z) = 0 \quad (30)$$

$$\mu_\psi(z) \geq 0 \quad (31)$$

where $\rho^\psi(z)$ symbolizes the robust specification of the STL formula, $\mu_\psi(z)$ symbolizes the vector of the boolean specifications that construct the formula ψ , the vector $d(z)$ denotes the dynamical constraints, and the vector $b(z)$ denotes the initial conditions or other equality constraints that are presented in the flat space. The solution of this optimization problem corresponds to the values of the control points for all aircraft. After obtaining the control points, the state trajectories and control inputs are generated in continuous-time by using the flat descriptions (19)-(27).

This optimization problem can be expressed as a non-convex optimization problem or a mixed-integer linear programming (MILP). In this study, we focus on the second case and present the objective and all constraints as linear expressions.

A. PERFORMANCE LIMITS AND INITIAL CONDITIONS

Any performance limit can be presented in the constraint set (29). Let us consider the restriction of the aircraft speed. By using the expression (20), the aircraft speed can be bounded with the following constraints:

$$V_{min}^2 \leq \dot{z}_1^2 + \dot{z}_2^2 + \dot{z}_3^2 \leq V_{max}^2 \quad (32)$$

where V_{min} and V_{max} are the bounds for the speed. However, this expression is nonlinear. In a non-convex optimization problem, this expression can be directly used, whereas it should be presented as linear constraints in MILP formulation. Let us evaluate the vertical and horizontal speeds separately. The rate of climb/descent \dot{z}_3 can be directly bounded such that $V_{min}^v \leq \dot{z}_3 \leq V_{max}^v$. To bound the horizontal speed, we approximate the Euclidean norm $\|[\dot{z}_1 \ \dot{z}_2]^T\|$ by the edges of an N-sided polygon that can be captured by the following inequalities [22]:

$$\begin{aligned} \dot{z}_1 \sin\left(\frac{2\pi n}{N}\right) + \dot{z}_2 \cos\left(\frac{2\pi n}{N}\right) &\leq V_{max}^h, \quad n = 1, \dots, N \\ \dot{z}_1 \sin\left(\frac{2\pi n}{N}\right) + \dot{z}_2 \cos\left(\frac{2\pi n}{N}\right) &\geq V_{min}^h - M a_n \\ \sum_{n=1}^N a_n &\leq N - 1 \\ a_n &\in \{0, 1\} \end{aligned} \quad (33)$$

where V_{min}^h and V_{max}^h are the bounds for the horizontal speed, and M is a large enough number. The other performance variables can also be restricted. To bound the acceleration, the N-sided polygon approach can also be used. In the optimization problem, this kind of constraint is enforced at the predefined points that are chosen uniformly over the time interval $[t_0, t_f]$.

The constraint set (30) consists of the initial conditions. Let $[x_0, y_0, h_0]^T$ be the initial position of the aircraft at t_0 . Then, the initial position is defined in the constraints as follows:

$$z_1(t_0) = x_0, \text{ and } z_2(t_0) = y_0, \text{ and } z_3(t_0) = h_0 \quad (34)$$

The other system variables can also be assigned to a certain value at a specific time. Let us focus on the heading angle

that is presented in the equation (21). Let θ be the assigned value for the heading angle x_5 at time t_s such that $x_5(t_s) = \theta$. By using the expression (21), this condition can be presented as follows:

$$\begin{aligned} \dot{z}_2(t_s) - \dot{z}_1(t_s) \tan(\theta) &= 0, \quad \text{and} \\ \dot{z}_1(t_s) &\leq 0 \text{ if } \pi/2 \leq \theta < 3\pi/2 \\ \dot{z}_1(t_s) &\geq 0 \text{ otherwise} \end{aligned} \quad (35)$$

When it is necessary to specify the values of the several system variables, the combination of them can be used to form linear constraints. Let us consider that the initial speed, initial heading angle and initial path angle are known, and we want to set x_4, x_5, u_2 to these specific values at time t_0 . Instead of setting them separately via equations (20), (21), (23), these initial values are used to calculate $\dot{x}_0, \dot{y}_0, \dot{h}_0$ via equations (11), (12), (13), and these derivatives are assigned as linear constraints:

$$\dot{z}_1(t_0) = \dot{x}_0, \text{ and } \dot{z}_2(t_0) = \dot{y}_0, \text{ and } \dot{z}_3(t_0) = \dot{h}_0 \quad (36)$$

B. STL OPERATORS

The tasks of the aircraft such as reaching particular regions, avoiding obstacles, ensuring appropriate separations can be described by the help of the convex hulls, where the faces of a convex hull are defined in terms of the affine expressions. These tasks can be evaluated as lying inside or outside the convex hulls. For example, obstacle avoidance refers to staying outside of the corresponding convex hull or lying in at least one of the outer halfspaces determined by the faces of the convex hull that can be formulated via disjunction operator. In like manner, the conjunction operator can be used to enforce the arriving a particular region. These missions can be specified with STL formalism. The STL operators should be described as MILP constraints to present these missions for the optimization problem.

Let μ be a predicate such that it holds at time t if and only if $\mu(x, t) \geq 0$. When the predicate μ is an affine expression, this condition of the STL formulas can be directly described as a MILP constraint:

$$\mu(x, t) \geq 0 \quad (37)$$

The other connectives in the STL formulas can also be described as MILP constraints. The negation of the predicate μ at time t can be presented as $-\mu(x, t) \geq 0$. The conjunction $\bigwedge_{i=1}^k \mu^i(x, t_i)$ is enforced with the following constraints:

$$\mu^i(x, t_i) \geq 0, \quad i = 1, \dots, k \quad (38)$$

The disjunction $\bigvee_{i=1}^k \mu^i(x, t_i)$ can also be ensured as follows:

$$\mu^i(x, t_i) \geq -Mb_i, \quad i = 1, \dots, k \quad (39)$$

$$\sum_{i=1}^k b_i \leq k - 1 \quad (40)$$

where M is a large enough number and b_i 's are binary variables.

The operators eventually $\diamond_{[\alpha, \beta]} \mu$ and always $\square_{[\alpha, \beta]} \mu$ can be described in terms of the conjunction and disjunction operators. The operator eventually can be presented as $\bigvee_{\tau=\alpha}^{\beta} \mu(x, \tau)$, and MILP formulation of the disjunction operator is given in (39)-(40). In a similar manner, the operator always can be described as $\bigwedge_{\tau=\alpha}^{\beta} \mu(x, \tau)$.

The robust definition of STL contains the min and max operators. Let us focus on the MILP formulation of conjunction operator in the robust setting such that $\rho^\psi = \bigwedge_{i=1}^k \rho^{\psi_i}(x, t_i)$. The following set of constraints can be used to obtain the robust conjunction:

$$\rho^\psi \leq \rho^{\psi_i}(x, t_i), \quad i = 1, \dots, k \quad (41)$$

$$M(b_{t_i}^{\psi_i} - 1) \leq \rho^\psi - \rho^{\psi_i}(x, t_i) \leq M(1 - b_{t_i}^{\psi_i}) \quad (42)$$

$$\sum_{i=1}^k b_{t_i}^{\psi_i} = 1 \quad (43)$$

$$b_{t_i}^{\psi_i} \in \{0, 1\}, \quad i = 1, \dots, k \quad (44)$$

where the variables $b_{t_i}^{\psi_i}$ are auxiliary binary variables that are used to enforce that $\rho^\psi = \min_i(\rho^{\psi_i}(x, t_i))$. The robust disjunction can also be presented in a similar manner. By replacing the first inequality with $\rho^\psi \geq \rho^{\psi_i}(x, t_i)$, the set of constraints enforces that $\rho^\psi = \max_i(\rho^{\psi_i}(x, t_i))$. The negation of $\rho^\psi(x, t)$ was also presented as $-\rho^\psi(x, t)$. Then, the rest of the operators can be obtained from these three operators.

C. CONTINUOUS-TIME SATISFIABILITY

It is stated in [11] that satisfying an STL formula for a sampled trajectory does not imply continuous-time satisfiability unless the formula is strictified. In order to guarantee the continuous-time satisfiability, a given formula ϕ is strengthened via a function such that $str : \phi \rightarrow \phi_s$. Firstly, an appropriate sampling period Δt must be chosen that satisfy a set of predefined conditions and guarantee the existence of at least one sampling point within each timing interval of the temporal operators. Secondly, the trajectory must have conservative bounds between two consecutive samples which can be satisfied as $\|x(t) - x(t + \Delta t)\| \leq \mathcal{E} \Delta t$, where $\mathcal{E} \geq 0$. Note that B-spline curves ensure this property. Then, the following relation holds [11, Theorem 5.3.1]: $\rho^{\phi_s}(x^{samp}, t) > \mathcal{E} \Rightarrow \rho^\phi(x, t) \models \top$, where x^{samp} is the sampled trajectory and x is the continuous-time trajectory. More detailed information can be found in Ch. 5 of the study [11]. For an appropriate sampling period, the value of \mathcal{E} can be calculated when the performance limits are given, and this value can be used in optimization problem as a buffer. In this way, the continuous-time satisfiability of an STL formula can be guaranteed, although the conditions are enforced for the sampled times. This strategy is also valid for the performance constraints in the optimization problem such as speed or acceleration.

VI. ILLUSTRATIVE EXAMPLES

In this section, we present the experimental results to evaluate the performance of the proposed method and show the validity of the method for air traffic control and UAV applications. In the applications, the air vehicles are controlled by

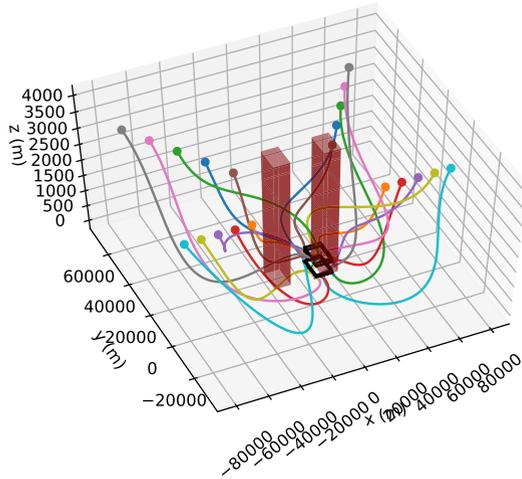


FIGURE 1. Optimal trajectories for a reach and avoid problem with 20 aircraft.

TABLE 1. Performance evaluation.

# aircraft	Boolean Mode		Robust Mode	
	Construction Time (s)	Solution Time (s)	Construction Time (s)	Solution Time (s)
2	0.33 ± 0.12	0.18 ± 0.06	0.35 ± 0.14	0.21 ± 0.08
5	0.97 ± 0.11	0.65 ± 0.03	1.01 ± 0.37	0.84 ± 0.27
10	2.41 ± 0.27	1.17 ± 0.34	3.34 ± 0.61	3.57 ± 0.52
20	7.36 ± 0.91	3.40 ± 0.63	7.96 ± 1.12	-

a centralized mechanism. In the ATC examples, we use the performance parameters of Boeing 737-800 for all aircraft, and we specify the minimum horizontal separation as 3nm, which is used in the real approach control operations. In the last case study, we focus on the control of a UAV fleet and specify the minimum separation as 5m. We construct the MILP problems in Python programming language using the PuLP library [23] and generate the solutions using the solver Gurobi.¹ All optimization problems are solved on a laptop with an Intel i7 processor and 16GB RAM.

A. REACH AND AVOID PROBLEM

Firstly, we analyse the simulation results for the reach-avoid problem. In this problem, all of the aircraft try to reach a target region, while avoiding conflicts with the other aircraft and obstacles. We consider this problem as an ATC application and use the performance parameters of Boeing 737-800. An example scenario for the control of 20 aircraft is presented in Fig. 1. The computation times for the solutions of the reach-avoid problems with different number of aircraft are presented in Table 1. The solutions are generated through two different modes. The Boolean mode corresponds to solving the optimization problem to generate trajectories that satisfy STL specifications without maximizing robustness, whereas the Robust mode refers to the maximization of the robustness of the STL specifications. It is observed that the optimization problem is always solved faster in the Boolean mode than the Robust mode. As presented in Table 1, the performance of the

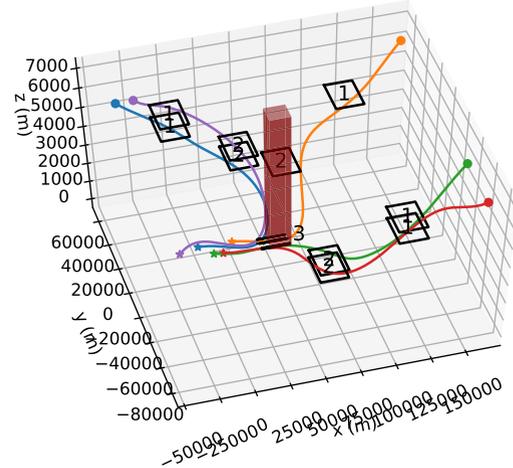


FIGURE 2. Optimal trajectories for approach control and arrival sequencing.

designed method is computationally tractable for the control of multiple aircraft. Although the problem cannot be solved in the Robust mode for 20 aircraft within feasible time limit, the solution is generated efficiently in the Boolean mode. The method can be used for the real-time applications.

B. APPROACH CONTROL AND ARRIVAL SEQUENCING

The second case is the application of the method to a realistic approach control scenario. This scenario contains the control of multiple aircraft in arrival traffic and sequencing them at 1000m. In this case study, all aircraft have the same performance limits except second aircraft (orange). The speed limits and initial speed of the second aircraft are 10m/s lower than the others to show the capability of the proposed method dealing with heterogeneous aircraft performance. In the scenario, each aircraft should visit three predefined regions at the specified times that refer to a standard terminal arrival (STAR) procedure in the real operation. After reaching the last region, they should arrange their headings according to the runway’s direction. In this phase, the aircraft are sequenced while the minimum separation requirements are satisfied and their headings are arranged according to the runway’s direction. The simulation results for the control of five aircraft are presented in Fig. 2. The observation is that the aircraft visit the predefined regions while avoiding obstacles and each other, and obtain the necessary heading angles according to the runway’s direction after reaching the last region. Furthermore, they are sequenced with minimum horizontal separation requirement. The results show that all aircraft always ensure the minimum separation requirement (3nm/5556meters) as illustrated in Fig. 3. The generated trajectories are also feasible in terms of performance limitations. For example, the speeds are always within limits during operation as shown in Fig. 4. The speed of the second aircraft is also always within its limits, and its speed is often lower than the other aircraft. This case study also shows that the proposed method can handle heterogeneous

¹http://www.gurobi.com/

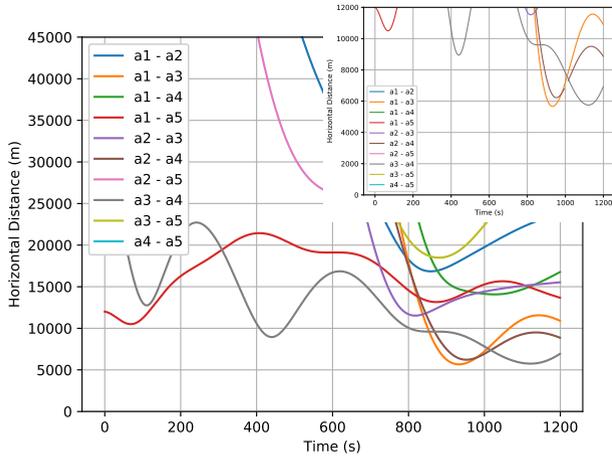


FIGURE 3. Solution to approach control and arrival sequencing: horizontal distances between aircraft pairs.

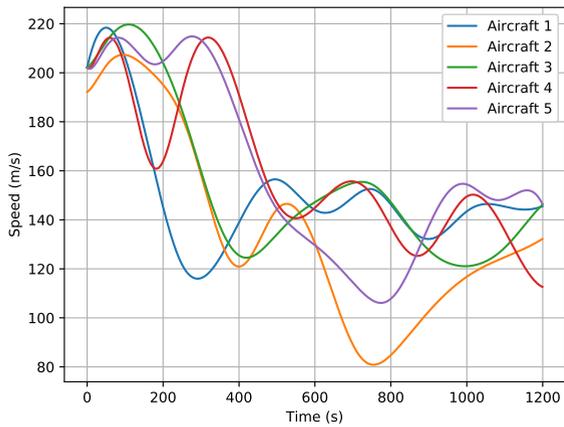


FIGURE 4. Solution to approach control and arrival sequencing: speeds of each aircraft stay within speed bounds.

aircraft performance during approach control. As mentioned before, the control inputs are also generated at the end of the optimization. The bank angles are illustrated as an example in Fig. 5. It is shown that the bank angles are also within limits, which are $[-25^\circ, 25^\circ]$ during take-off and landing and $[-45^\circ, 45^\circ]$ during other flight phases for civil flight [24], and they are tractable by a Boeing 737-800 because of the soft angle changes. Moreover, the time performance of the method is also practicable. For this case study, the construction time of the MILP is approximately 1.8s, whereas the solution time is around 3.9s.

C. UAV FLEET NARROW-PASSAGE PROBLEM

In the last example, we solve a narrow-passage problem for a quadrotor fleet, where the quadrotors aim to reach a delivery point by passing a narrow-passage. We aim to show that the method can also be used in UAV applications. The flat descriptions of a quadrotor are presented in [25]. The problem is solved for a fleet that has 7 quadrotors as presented in Fig. 6, which also contains the positions of the quadrotors while passing the passage at time $t = 30s$. As presented in Fig. 7, the quadrotor pairs always ensure the minimum

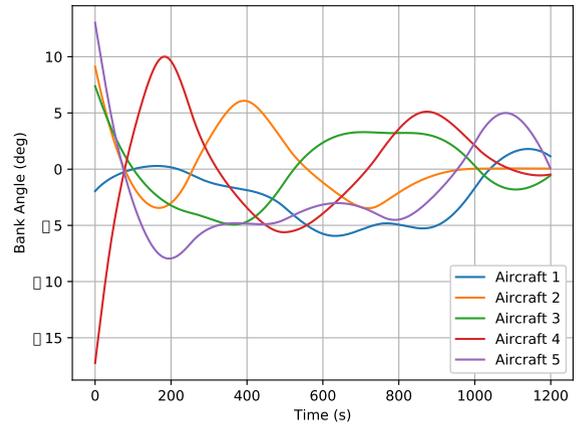


FIGURE 5. Solution to approach control and arrival sequencing: bank angles of each aircraft stay within bank angle limits.

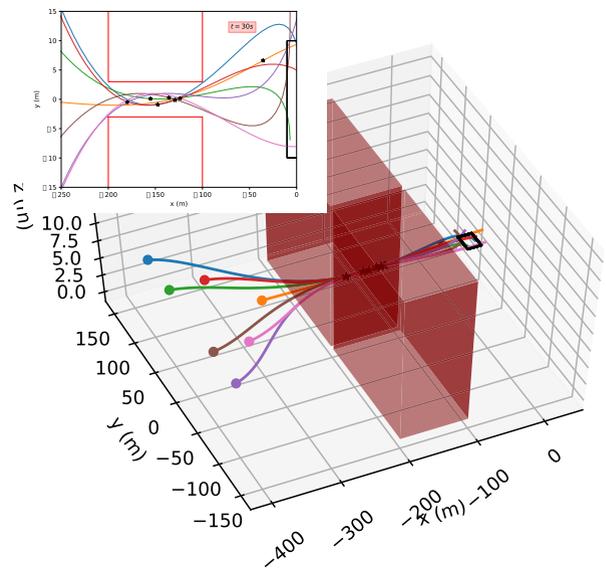


FIGURE 6. Optimal trajectories for the quadrotor fleet for the narrow-passage problem.

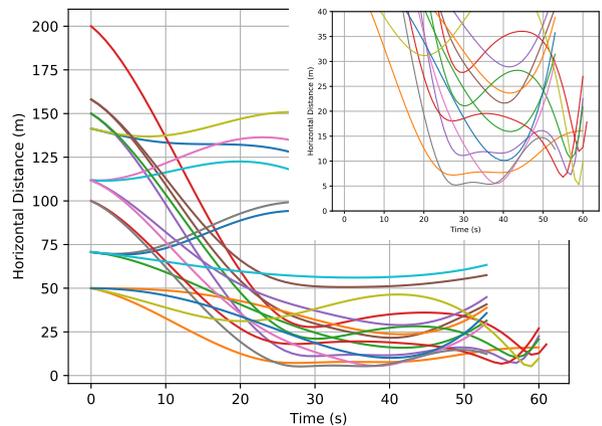


FIGURE 7. Solution to UAV fleet narrow-passage problem: horizontal distances between quadrotor pairs.

separation requirement, which is 5m, during operation. These figures show that the fleet satisfies the mission specifications. The system variables are also within limits during

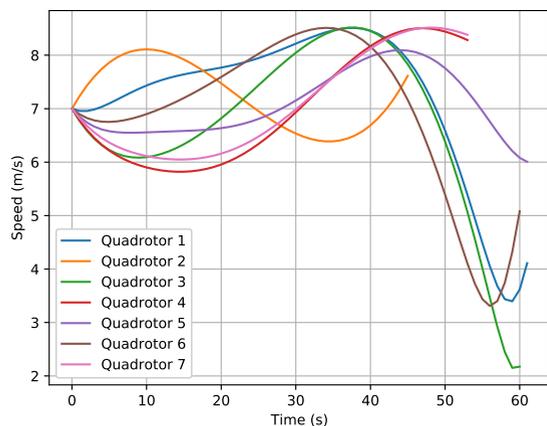


FIGURE 8. Solution to UAV fleet narrow-passage problem: speeds of each quadrotor stay within speed bounds.

operation. For example, the speeds of the UAVs are smaller than the maximum speed 8.5m/s as presented in Fig. 8.

VII. CONCLUSION

In this paper, we developed an optimization-based method for the mission planning and control of multi-aircraft systems with STL specifications. We represented the missions of aircraft via an STL formalism and described the system dynamics as a (partially) differentially flat system. We then constructed a MILP-based formulation to generate optimal trajectories that satisfy STL specifications. In our examples, we used a realistic aircraft model with performance parameters of Boeing 737-800 and realistic conditions to simulate arrival traffic, and evaluated the performance of the proposed method. Moreover, we showed that the method can be applied to other multi-agent systems, such as mission planning and control of multiple unmanned aircraft.

REFERENCES

- [1] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Berlin, Germany: Springer, 2004, pp. 152–166.
- [2] S. Karaman, R. G. Sanfelice, and E. Frazzoli, "Optimal control of mixed logical dynamical systems with linear temporal logic specifications," in *Proc. 47th IEEE Conf. Decis. Control*, Dec. 2008, pp. 2117–2122.
- [3] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 81–87.
- [4] S. Saha and A. A. Julius, "An MILP approach for real-time optimal controller synthesis with metric temporal logic specifications," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 1105–1110.
- [5] Y. V. Pant, H. Abbas, and R. Mangharam, "Smooth operator: Control using the smooth robustness of temporal logic," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2017, pp. 1235–1240.
- [6] H. Abbas, G. Fainekos, S. Sankaranarayanan, and F. Ivančić, and A. Gupta, "Probabilistic temporal logic falsification of cyber-physical systems," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 2s, p. 95, 2013.
- [7] I. Saha, R. Ramaiithima, V. Kumar, G. J. Pappas, and S. A. Seshia, "Automated composition of motion primitives for multi-robot systems from safe LTL specifications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 1525–1532.
- [8] J. A. DeCastro, J. Alonso-Mora, V. Raman, D. Rus, and H. Kress-Gazit, "Collision-free reactive mission and motion planning for multi-robot systems," in *Robotics Research*. Cham, Switzerland: Springer, 2018, pp. 459–476.
- [9] A. Nikou, D. Boskos, J. Tumova, and D. V. Dimarogonas, "On the timed temporal logic planning of coupled multi-agent systems," *Automatica*, vol. 97, pp. 339–345, Nov. 2018.
- [10] Y. V. Pant, H. Abbas, R. A. Quaye, and R. Mangharam, "Fly-by-logic: Control of multi-drone fleets with temporal logic objectives," in *Proc. 9th ACM/IEEE Int. Conf. Cyber-Phys. Syst.*, Apr. 2018, pp. 186–197.
- [11] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theor. Comput. Sci.*, vol. 410, no. 42, pp. 4262–4291, 2009.
- [12] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *Proc. Int. Conf. Formal Modeling Anal. Timed Syst.* Berlin, Germany: Springer, 2010, pp. 92–106.
- [13] J. Levine, *Analysis and Control of Nonlinear Systems: A Flatness-Based Approach*. Berlin, Germany: Springer, 2009.
- [14] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *Int. J. Control*, vol. 61, no. 6, pp. 1327–1361, 1995.
- [15] M. Fliess, J. Levine, P. Martin, and P. Rouchon, "A lie-backlund approach to equivalence and flatness of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 44, no. 5, pp. 922–937, May 1999.
- [16] B. Başpınar and E. Koyuncu, "Differential flatness-based optimal air combat maneuver strategy generation," in *Proc. AIAA Scitech Forum*, 2019, p. 1985.
- [17] L. Piegl and W. Tiller, *The NURBS Book*. New York, NY, USA: Springer, 2012.
- [18] B. Başpınar, C. Pasaoglu, N. K. Ure, and G. Inalhan, "Infrastructure development for ground-based separation assurance with optional automation," in *Proc. 5th Int. Conf. Appl. Theory Automat. Command Control Syst.*, 2015, pp. 65–74.
- [19] C. Pasaoglu, B. Başpınar, N. K. Ure, and G. Inalhan, "Hybrid systems modeling and automated air traffic control for three-dimensional separation assurance," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 230, no. 9, pp. 1788–1809, 2016.
- [20] M. Uzun, B. Başpınar, E. Koyuncu, and G. I. Inalhan, "Takeoff weight error recovery for tactical trajectory prediction automaton of air traffic control operator," in *Proc. IEEE/AIAA 36th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2017, pp. 1–7.
- [21] B. Başpınar, 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 *Proc. IEEE/AIAA 36th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2017, pp. 1–10.
- [22] T. Schouwenaars, "Safe trajectory planning of autonomous vehicles," Ph.D. dissertation, Dept. Aeronaut. Astronaut., Massachusetts Inst. Technol., Cambridge, MA, USA, 2006.
- [23] S. Mitchell, M. O'Sullivan, and I. Dunning, "PuLP: A linear programming toolkit for Python," 2011, unpublished. [Online]. Available: <https://code.google.com/p/pulp-or/>
- [24] *User Manual for the Base of Aircraft Data (BADA) Revision 3.11*, Eurocontrol Exp. Centre, Les Bordes, France, 2013.
- [25] S. Formentin and M. Lovera, "Flatness-based control of a quadrotor helicopter via feedforward linearization," in *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, Dec. 2011, pp. 6171–6176.



BARIŞ BAŞPINAR 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 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. He is currently a Visiting Student Researcher with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology (MIT). His current research interests include data-driven ATM network modeling, 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 modeling, 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, USA, in 2006.

She was a Principal Development Engineer with 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 Head of the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. 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 with Boeing Research and Technology of Europe, from 2013 to 2014, and 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.

...