This paper focuses on modeling of air-to-air combat via optimization-based control and game-theoretic approach. The flatness property of the dynamical system is used to present aircraft dynamics in terms of specific variables and their derivatives that can be described via a set of parameterized curves. With the help of game theory, the combat between two aircraft is translated into an optimization problem in terms of these parameterized curves. The optimization problem is solved with a moving time horizon scheme to generate optimal strategies for aircraft in aerial combat. In this way, the algorithm produces optimal feasible strategies that meet all the given and dynamical constraints. The method is demonstrated with aerial combat between two UAVs. Two different case studies are analyzed to show and validate the method.

**INDEX TERMS** Aerial combat, optimization, game theory, receding horizon control, differential flatness.

**I. INTRODUCTION**

Unmanned systems are becoming parts of civil and military aviation. They are replacing the manned systems in various aerial missions. However, it is a challenging issue to replace a manned platform by an unmanned system in aerial combat because of the dynamic nature of combat. An unmanned aerial vehicle (UAV) can be controlled remotely in an aerial combat. However, the UAV will be in a disadvantageous position against a manned platform because of the limited situation awareness of the UAV pilot, which originates from current pilot-vehicle interface technology. This limitation can be overcome via the automated combat maneuvering. Besides, an operator could improve vehicle performance and manage multiple UAVs in aerial combat by the help of automated combat maneuvering.

In spite of the advances made in missile technology and long range radar, modern fighter aircraft (e.g., F/A-22, F-15, and F-35) are still designed for close combat and military pilots are trained to effectively use in one-on-one aerial combat. The aerial combat has a vital role in military aviation. In air-to-air combat, the primary role of a combatant is destroying the enemy aircraft that is achieved by establishing air superiority over a battlefield. The key element that brings victory is always considered as achieving and maintaining air superiority that are affected by several factors such as performance of the aircraft, skill of pilot and combat tactics [1], [2]. The decision making, which contains evaluation of these factors, is a challenging subject due to the multiple decision makers that have conflicted interests. Besides, analysis of aerial combat tactics and technologies are expensive tasks in practice. Hence, we benefit from mathematical models, which are based on optimization, game theory and control theory, to analyze combat tactics and technologies. After creating a strategy generation platform for aerial combat, this platform can be used for decision support of the UAV operators, autonomous aerial combat, training of pilots and analyzing potential combat scenarios.

The modeling of aerial combat game has been studied from different aspects. The pursuit-evasion game problem is well-known version of these modeling perspectives, where the combatants have fixed roles. In the papers [3], [4], the authors generated reachable sets via Hamilton-Jacobi-Isaacs (HJI) partial differential equation to find solution for the pursuit-evasion game. The study [4] showed that the level set method can be used to calculate the backward reachable set for a two player differential game. However, this kind of methods suffers from curse of dimensionality. They are computationally inefficient. There are also alternative approaches to construct reachable sets. The zonotope-based reachability algorithms
such as [5] and the methods based on approximations that use oriented rectangular hulls such as [6] have been proposed. These methods have not been implemented for the pursuit-evasion game. However, if they are implemented, the inherited problems will still continue with these methods. The role of the combatant will be fixed, and the generated results will be a set instead of an optimal trajectory that can be used in an autonomous aerial combat. The pursuit-evasion game has been also evaluated using model predictive control. The studies [7], [8] presented a nonlinear model predictive tracking control algorithm that provided evasive maneuvers to a fixed-wing aircraft when confronted by an airborne adversary of a priori type. In these studies, it was needed to encode proven aircraft maneuvering tactics into the cost functions, because the required behaviours cannot be produced by the algorithm. Besides, the method was not capable of switching between pursuit and evasion roles. The rule-based structure is an alternative to model the air combat as in study [9].

In this study, a rule-based logic was used to generate strategies. The influence diagram is another approach to solve the problem as presented in [10] that contained the modeling of the decision-making process of the pilot for one-on-one aerial combat using an influence diagram. In the works [11], [12], the combat between helicopters was simulated via a game theoretic approach that tried to select optimal maneuver from a set of elemental maneuver such as steady flight, max load factor turn, etc. The authors of [13], [14] used approximate dynamic programming approach to produce maneuver inputs such as roll left, roll right and maintain the current bank for aerial combat on horizontal plane. In the studies [15], [16], the authors focused on a game theoretic hybrid system modeling approach that used real aerial combat maneuvers to obtain strategies for the engagement phase of the aerial combat. All of these methods (i.e., rule-based approaches, influence diagram, game theoretic formulation, approximate dynamic programming, maneuver-based approach) used a subset of the control actions as action space in spite of the fact that they proposed different approaches for strategy generation. This action space can contain elemental maneuvers such as turn, go along, roll left with a predefined angle or advanced combat maneuvers such as immelmann, split S, spiral dive.

In these cases, the method cannot search all of the movement space of the fighter. All of these methods share the same drawback originated from using the subset of the control inputs. When the input space is not discretized via maneuvers such as elemental or real air combat, the computation time is a challenging issue. The study [17] focused on generating combat strategies in continuous control space. However, it used a simplified aircraft model in 2D and an inefficient trajectory prediction strategy when evaluating the possible choices of the opponent.

In this study, we overcome these limitations. We present a method to model one-on-one aerial combat that is based on differential-flatness, curve parametrization, game theoretic evaluation and receding horizon control. The flat description of aircraft dynamics, which contains movements on horizontal and vertical planes, is presented that allows to represent the aircraft motion in terms of a set of specific variables and their derivatives. This property provides an opportunity to evaluate aircraft motion without integration of a set of differential equations. This leads to parametrization of aircraft trajectory in terms of these specific variables and their derivatives. By the help of game theory, the aerial combat is modeled as an optimization problem in terms of parametrized trajectories and performance constraints are also presented in this optimization scheme to generate feasible strategies for combatants. This method allows the presentation of the problem in a lower dimensional space with all given and dynamical constraints that speed up the strategy generation process. The created optimization problem is solved over a moving time horizon till the end to generate the combat strategies. For simulation purpose, we have implemented the algorithm to aerial combat between two UAVs. We demonstrated the success of the algorithm through two different scenarios.

The rest of the paper is organized as follows. The section II explains the smooth trajectory generation process. The section III contains the aircraft model, which can maneuver on horizontal and vertical planes, and flat description of this model. In the section IV, the one-on-one aerial combat is formulated as an optimization problem by the help of game theory and presented trajectory generation process. The details about optimization problem, generating reachable set of the enemy, initialization scheme, solution method and also receding horizon approach are given in this section. Section V contains the demonstrations. The aerial combats between two UAVs are analyzed to show and validate the methodology. Finally, concluding remarks are given in Section VI.

II. SMOOTH TRAJECTORY GENERATION

Let us focus on the continuous-time dynamical systems of the form:

\[
\dot{\chi}(t) = f(\chi(t), u(t)) \quad \chi(0) = \chi_0 \quad (1)
\]

where \(u(t) \in \mathbb{R}^m\) consists of the control inputs, \(\chi(t) \in \mathbb{R}^n\) contains the system states, and \(\chi_0 \in \mathbb{R}^n\) is the initial state of the system.

In the trajectory generation problem, a feasible trajectory \(t \mapsto (\chi(t), u(t))\) that starts from an assigned initial condition and ends in a specific final state while satisfying the system dynamics (1) should be generated. It is necessary to generate appropriate control inputs that are used to obtain the feasible trajectory by driving the system equation. In general, because of the integration of the system equation, this problem can be quite difficult. For nonlinear systems, it may pose some additional problems [18].

For differentially flat systems, the trajectory generation process can be more easily achieved. The dynamical system (1) is differentially flat if there are relations such that ([19]–[21])

\[
z = \kappa(x, u, \dot{u}, \ldots, u^{(r)}),
\]

\[
(\chi, u) = v(z, \dot{z}, \ldots, z^{(r-1)}, z^{(r)}).
\]
where \( \kappa, \nu \) are smooth functions, and \( z \) contains the flat outputs. In this case, all system dynamics can be expressed in terms of the flat outputs and their derivatives. This model and the system dynamics (1) are identical. Hence, it can be used to efficiently generate trajectories. The set of the equations (3) yields that for every given flat trajectory \( t \mapsto z(t) \), the evolution of the system variables \( t \mapsto \chi(t) \) and \( t \mapsto u(t) \) is also determined without integration of the differential equations. Furthermore, given a sufficiently smooth flat trajectory \( t \mapsto z^*(t) \), the corresponding feedforward \( u^* \) can be obtained directly via the equation (3).

A set of basis functions can be used to present each element of the flat output \( z \) such that

\[
z_i = \sum_{j=1}^{n_i} w_{ji} \phi_{ji}(t) = W_i \Phi_i(t)
\]

where, \( w_{ji} \) corresponds to a control point, \( \phi_{ji}(t) \) is a basis function and \( z = [z_1, z_2, \ldots, z_J]^T \).

There are many types of basis functions. In this study, we use the B-spline basis functions to present the flat outputs. Let \( \rho \) be a nonnegative integer and let \( \mathbb{T} = \{t_0, t_1, \ldots, t_m\} \) be a nondecreasing sequence of real numbers. The \( \rho \)th B-spline basis function of \( p \)-degree, denoted by \( N_{q,p}(t) \), is defined as [22]:

\[
N_{q,0}(t) = \begin{cases} 1 & \text{if } t_q \leq t < t_{q+1} \\ 0 & \text{otherwise} \end{cases}
\]

\[
N_{q,p}(t) = \frac{t - t_q}{t_{q+p} - t_q} N_{q,p-1}(t) + \frac{t_{q+p+1} - t}{t_{q+p+1} - t_{q+1}} N_{q+1,p-1}(t)
\]

When dealing with the dynamical systems, some of the geometric properties of the B-spline functions [22] have positive impact on the performance of the trajectory generation process. For example, a B-spline curve such as \( z_i \) is \( C^{p-1} \) in any \( t \in \mathbb{T} \), and \( C^{\infty} \) otherwise. It is a continuous and differentiable function. The combination of B-splines of lower order can be used to calculate the higher order derivatives of B-spline basis functions. Besides, the control points have linear impact on the higher order derivatives of the curve such that \( z_i^{(k)} = \sum_{j=1}^{N_i} w_{ji} \phi_{ji}^{(k)}(t) \), where \( \phi_{ji}^{(k)}(t) \) is \( k \)-order derivative of basis function. Furthermore, the B-spline functions satisfy that \( N_{q,p}(t) \geq 0 \) and \( \sum_{q=0}^{\infty} N_{q,p}(t) = 1, \forall t \in [t_0, t_m] \), and they have local support such that \( N_{q,p}(t) \neq 0 \) iff \( t \in [t_q, t_{q+p}] \). Therefore, the change in a specific control point only affects a specific region of the curve without influencing the rest of it.

Assume that there is a flat trajectory that is configured by a set of control points \( W \) for the dynamical system (1). This trajectory can be shaped via the control points in \( W \), and each control point has an impact on a specific region of this trajectory. However, some configurations of the control points can cause infeasible trajectories because of the violation of the dynamical constraints. To prevent infeasible trajectory generation, the trajectories of the system variables and control inputs can be expressed in terms of the control points \( W \) via the expressions in (3) and any violation in the performance constraints can be checked via these system trajectories. The violations can be prevented by finding the proper control inputs. Because of the local support property, it is not necessary to modify all of the control inputs to prevent a specific violation in a specific region of the trajectory. This property makes it easy to find the proper control inputs. In this study, we formulate and solve an optimization problem to generate appropriate control inputs that lead to feasible trajectories.

### III. CHARACTERIZATION OF AIRCRAFT DYNAMICS FOR MILITARY APPLICATIONS

This section contains the aircraft model for military applications, and the presentation of this model as a differentially flat system. Differential flatness is a structural property of a class of nonlinear dynamical systems, denoting that all system variables can be written in terms of a set of specific variables and their derivatives [19]–[21]. By using this property, aircraft dynamics can be evaluated via a set of algebraic equations without numerical integration.

The motion of aircraft can be expressed by the following set of differential equations:

\[
\dot{x} = V \cos(\psi) \cos(\gamma)
\]

\[
\dot{y} = V \sin(\psi) \cos(\gamma)
\]

\[
\dot{z} = V \sin(\gamma)
\]

\[
\dot{\psi} = \frac{L + T \sin(\alpha) \sin(\mu)}{mV} \cos(\gamma)
\]

\[
\dot{\gamma} = \frac{L + T \sin(\alpha) \cos(\mu) - g \cos(\gamma)}{mV}
\]

where \( x, y \) and \( h \) define the position of aircraft. The rest of the state variables are the speed \( V \), the heading angle \( \psi \) and the flight path angle \( \gamma \). The control variables are defined as the bank angle \( \mu \), the angle of attack \( \alpha \) and the thrust \( T \). \( L \) and \( D \) refer to lift and drag force, respectively. The mass of aircraft is symbolized as \( m \). The gravity is presented as \( g \).

The longitudinal load factor \( n_x \) and normal load factor \( n_z \) can be described as follows:

\[
n_x = \frac{T \cos(\alpha) - D}{mg}
\]

\[
n_z = \frac{L + T \sin(\alpha)}{mg}
\]

The equations of aircraft’s motion can be reorganized by using load factors as follows:

\[
\dot{x} = V \cos(\psi) \cos(\gamma)
\]

\[
\dot{y} = V \sin(\psi) \cos(\gamma)
\]

\[
\dot{h} = V \sin(\gamma)
\]

\[
\dot{\psi} = \frac{g n_x - \sin(\gamma)}{V \cos(\gamma)}
\]

\[
\dot{\gamma} = \frac{g}{V} (n_z \cos(\mu) - \cos(\gamma))
\]
where the longitudinal load factor \( n_x \), the normal load factor \( n_z \), and the bank angle \( \mu \) are the control inputs. We will use the set of differential equations (14)-(19) as aircraft model to represent the motion of a combatant during aerial combat.

This system is differentially flat, if flat outputs are chosen as \( x, y \) and \( h \):

\[
z = [z_1 \ z_2 \ z_3]^T = [x \ y \ h]^T
\] (20)

The other state variables, \( V, \psi \) and \( \gamma \) can be described in terms of the flat outputs. The equation (14), (15) and (16) can be used to calculate the speed \( V \) in terms of the flat outputs. By squaring the both sides of these equations and summing the squared equations, the square of the speed can be obtained. Then the speed \( V \) can be presented as follows:

\[
V = \sqrt{z_1^2 + z_2^2 + z_3^2}
\] (21)

The heading angle \( \psi \) can be expressed using the equation (14) and (15). Dividing eq. (15) by eq.(14), the tangent of the heading angle can be obtained. Then, the heading angle \( \psi \) is represented as follows:

\[
\psi = \arctan \left( \frac{z_2}{z_1} \right)
\] (22)

By using the equation (16), the flight path angle \( \gamma \) is described in terms of flat outputs with the equation below.

\[
\gamma = \arcsin \left( \frac{z_3}{V} \right)
\] (23)

where the speed \( V \) was already represented in terms of flat outputs in the eq. (21).

The derivatives of the speed \( V \), the heading angle \( \psi \) and the flight path angle \( \gamma \) can be calculated by taking the derivative of the equation (21), (22) and (23), respectively, that correspond to the following equations:

\[
\dot{V} = \frac{\dot{z}_1 z_3 + z_2 \dot{z}_3 + z_3 \dot{z}_2}{\sqrt{z_1^2 + z_2^2 + z_3^2}}
\] (24)

\[
\dot{\psi} = \frac{\dot{z}_2 z_1 - z_2 \dot{z}_1}{z_1^2 + z_2^2}
\] (25)

\[
\dot{\gamma} = \frac{\dot{z}_3 V - \dot{V} z_3}{V \sqrt{z_1^2 + z_2^2}}
\] (26)

The control inputs can also be formulated in terms of flat variables via the equation (17), (18) and (19). The bank angle \( \mu \) can be obtained using the equation (18) and (19). In eq. (18), \( \sin(\mu) = \psi V \cos(\gamma)/g n_c \) and, in eq. (19), \( \cos(\mu) = \dot{\gamma} V + g \cos(\gamma)/g n_c \). Then \( \mu \) can be found by dividing the first equation by the second equation as follows:

\[
\mu = \arctan \left( \frac{V \dot{\psi} \cos(\gamma)}{\dot{V} \psi + g \cos(\gamma)} \right)
\] (27)

where state variables \( V, \psi, \gamma \) and their derivatives were already presented in terms of flat variables in the equations (21)-(26). The longitudinal load factor \( n_x \) can be described in terms of flat outputs by reorganizing the eq. (17) as follows:

\[
n_x = \frac{\dot{V}}{g} + \sin(\gamma)
\] (28)

The last control variable \( n_z \) can be expressed by reorganizing the eq. (19) as follows:

\[
n_z = \frac{\dot{V} V}{g \cos(\mu)} + \frac{\cos(\gamma)}{\cos(\mu)}
\] (29)

The set of equations (20)-(29) presents the flat description of the aircraft model. The differential flatness relies on the representation of system variables in terms of the flat outputs and their derivatives. This characteristic provides an advantage when dealing with trajectories. Because of this property, the system dynamics can be described as analytic expressions in the flat output space without integrating any differential equation. Then, the objective of trajectory planning, which is to find a trajectory going from point \( a \) to \( b \), can be reduced to finding smooth curves that satisfy a set of defined conditions in the flat output space. In this methodology, firstly, all dynamical and environmental constraints are expressed in terms of flat outputs and their derivatives, then the trajectory planning problem is solved in the flat output space, and in the end, the variables are converted back to the original state and input space. In this way, the number of variables in the optimization problem is reduced to speed up the process.

IV. FORMULATION OF AERIAL COMBAT AS AN OPTIMIZATION PROBLEM

This section contains the construction of the optimization problem that models the decision process of a fighter in the presence of another decision maker. The decision makers are intelligent opponents who make decisions according to their own self-interest. This is the topic of game theory that focuses on decision process of players with conflicted interests [23].

A. AERIAL COMBAT BETWEEN AIRCRAFT

Let us consider that two combatants simultaneously make a decision without knowing how to other will act, where the fighters have completely opposing interests. This corresponds to a zero-sum game in which a gain for one decision maker leads to a loss for the other, and vice versa.

Suppose that the cost functions of the players are symbolized as \( L_i : U_b \times U_r \rightarrow \mathbb{R} \) for \( i = 1, 2 \), where \( U_b \) and \( U_r \) correspond to action spaces of the blue and red combatants, respectively. An important constraint for zero-sum games is:

\[
L_1(u_b, u_r) = -L_2(u_b, u_r)
\] (30)

which means that the opponents have completely conflicted interests and a loss for one player corresponds to the reward for other. The name of zero-sum originates from this property, which means that the sum of opponents’ costs is zero. Instead of defining two different cost functions, it is common to define a cost function \( L \) for first player. In this case, the goal of the blue fighter is to minimize \( L \), whereas the red aircraft
tries to maximize $L$. Thus $L$ can be considered as a reward for red, but a cost for blue.

There are some assumptions on the game setting. The players have knowledge of each other’s cost functions. This implies that each player completely understands the opponent’s motivation. The other assumption is that the players are rational, which corresponds to trying to get best reward whenever possible. The first player will not prefer a higher cost action to a lower cost action when it is available. Likewise, the second player will not choose an action that leads to lower cost. The final assumption is that both players make their decisions simultaneously without knowledge of opponents’ decisions. For any player, there is no information about the decision of opponent that can be exploited by the player.

Let us consider the game from the standpoint of blue. Applying worst-case analysis is rational that contains the evaluation of possible actions of the opponent to make a decision. This corresponds to selecting an action with minimum cost that the first player can guarantee itself no matter what the second player does. Let us define this selection as a security strategy for blue. A security strategy, $u^*_b$, for the blue aircraft can be formalized as:

$$u^*_b = \arg \min_{u_b} \left\{ \max_{u_r} \left\{ L(u_b, u_r) \right\} \right\}$$ (31)

Let us focus on the game from the perspective of red, which aims to maximize $L$. It can also use worst-case analysis, which means that it chooses an action that guarantees a high cost, in spite of the action of the first player. A security strategy, $u^*_r$, for red is described as:

$$u^*_r = \arg \max_{u_r} \left\{ \min_{u_b} \left\{ L(u_b, u_r) \right\} \right\}$$ (32)

The upper value, $\overline{L}$, and lower value, $\underline{L}$, of the game are defined as follows:

$$\overline{L} = \max_{u_r} \left\{ L(u^*_b, u_r) \right\}, \quad \underline{L} = \min_{u_b} \left\{ L(u_b, u^*_r) \right\}$$ (33)

An interesting relationship between the upper and lower values is that $\underline{L} \leq \overline{L}$ for a zero-sum game. This is shown by observing that

$$\underline{L} \leq L(u^*_b, u^*_r) \leq \overline{L}$$ (34)

in which $L(u^*_b, u^*_r)$ is the cost as a result of players’ security strategies. If the players are rational decision makers, then the resulting cost always lies between $\underline{L}$ and $\overline{L}$. If $\underline{L} = \overline{L}$, the security strategies are called a saddle point that refers to an equilibrium. For some games, the values are equal, but for many $\underline{L} < \overline{L}$ [24]. It is the property of the game.

Let us consider the cost function in an aerial combat. The cost can be presented as combination of angular relationship and range score. In air combat, a combatant tries to get an advantageous position to shoot the enemy. The most advantageous position for a fighter is the rear of the opponent. Therefore, combatants try to locate behind of their enemies. Aerial combat relative geometry showing Aspect Angle (AA), Bearing Angle (BA) and Heading Crossing Angle (HCA) are illustrated for blue in the Fig. 1. The aspect angle (AA) and bearing angle (BA) can be used to quantify this objective [25]. Let us focus on the blue combatant. The blue aircraft would like to minimize both angles to get air superiority. When $\text{AA}_b$ and $\text{BA}_b$ equal to zero, the blue aircraft’s orientation is perfect for shooting. The orientation goal can be presented via $\text{AA}_b$ and $\text{BA}_b$ as follows:

$$L_o = \frac{\text{AA}_b + \text{BA}_b}{\pi} - 1$$ (35)

The blue aircraft tries to minimize $L_o \in [-1, 1]$, whereas the red aircraft would like to maximize this score. The second objective is about range between combatants. The distance between aircraft must be smaller than a threshold, which corresponds to missile or gun range, for shooting. The range objective is described as a Gaussian distribution that is given by:

$$L_r = ae^{-(R - R_d)^2}$$ (36)

where $R \in \mathbb{R}_{>0}$ denotes the range between combatants. $R_d$ and $\sigma$ correspond to the desired range and standard deviation, respectively. In this study, the shooting range is taken as 300m. $R_d = 300$ and $\sigma = 100$ are chosen to construct the range score. The objective of combat can be represented as combination of these two functions as follows:

$$L = L_oL_r$$ (37)

The blue combatant desires to minimize the $L$, while the red fighter aspires to maximize it. As presented before, the blue aircraft applies the security strategy such that $\min_{u_b} \max_{u_r} L$. The blue combatant tries to minimize his maximum loss no matter what the red combatant does. This strategy gives the minimum amount of score that blue can guarantee himself.

Note that the formulation of the aerial combat as a zero-sum game is not an assumption. It is the consequence of the fighters’ objectives. The goal of air combat is to maneuver your aircraft into a position of advantage over the other aircraft, while minimizing risk to your own aircraft. As presented in the handbook of F-16 [25], the positional advantage can be presented in terms of aspect angle and bearing angle.
When this goal is used, the fighters have conflicted interests. A fighter maximizes its advantageous position, while minimizing the opponent’s advantageous position. Therefore, the problem is a zero-sum game. And the chosen objective function is practical as described in the handbook of F-16 [25]. Besides, there are several studies that use similar objective functions based on aspect angle and bearing angle such as [10]–[13].

Let us consider the case that the dynamical system for blue is presented in terms of flat outputs, and these outputs are described via B-spline curves. In this case, the objective can be formulated in terms of B-splines’ parameters:

$$\min_{P_b} \max_{u_r} F_L(t)$$

where $P_b$ consists of control points of B-spines that present the trajectory of the blue aircraft. $F_L(t)$ refers to the objective $L$ in terms of B-splines. Let $c$ be a new variable that is used to represent the objective function in standard optimization form. The new variable is restricted to be greater than objective function, $c \geq \max F_L(t)$, and it is tried to make $c$ as small as possible. The problem is represented as:

$$\min_{P_b} c$$

subject to: $F_L(\tilde{x}^r_{0}, \tilde{u}(t)) \leq c \quad \forall \tilde{u} \in \tilde{U}$

$$p(z) \leq r$$

$$q(z) = s$$

where $\tilde{u}$ corresponds to an action trajectory and $\tilde{U}$ contains samples for action trajectories. The state trajectory for red is symbolized as $\tilde{x}(x^r_0, \tilde{u})$, which is derived from $\tilde{u}$. $F_L(\tilde{x}^r_{0}, \tilde{u}(t))$ denotes the value of score for sampled state trajectory of red $\tilde{x}(x^r_0, \tilde{u})$ with respect to the set of decision variables $P_b$ at time $t$. The set of all state trajectories derived from action trajectories in $\tilde{U}$ creates the reachable set of the red aircraft. By evaluating each state in the sampled reachable set of the red aircraft, blue tries to find its security strategy. The vector $p(z)$ consists of the inequality constraints in output space such as speed constrains, orientation constrains etc. $r$ consists of the bounds for associated variables. The vector $q(z)$ contains the equality constraints in output space such as initial conditions, way-points etc.. The solution of this optimization problem gives the best strategy for the blue fighter. By constructing and solving the same problem from perspective of the red combatant, the optimal strategy for the red aircraft can also be generated.

**B. REACHABLE SET OF ENEMY AIRCRAFT**

The set of constraints in Eq. (40) contains the reachable set of the enemy aircraft. The reachable set of an aircraft can be generated via a sampling-based method. A state trajectory can be derived from an action trajectory $\tilde{u}$ as follows:

$$\chi(T) = \chi(0) + \int_0^T f(\chi(t), u(t))dt$$

where $\chi(0)$ is the initial state at time $t = 0$. Let $\tilde{x}(\chi_0, \tilde{u})$ denote the state trajectory over all time, obtained via integration (43). Let $U$ be the set of all permissible action trajectories on the time interval $[0, \infty)$. A state trajectory $\tilde{x}(\chi_0, \tilde{u})$ can be generated from each $\tilde{u} \in U$ via equation (43). The main question is which states in $X$ are visited by these trajectories? In general, it may not be possible to visit some states because of dynamical constraints. The set of states that can be reached from an initial condition is described as reachable set of the dynamical system.

Let $R(\chi_0, U) \subseteq X$ denote the reachable set from $\chi_0$, which is the set of all states that are visited by any trajectories that start at $\chi_0$ and are obtained from some $\tilde{u} \in U$ by integration. This can be expressed as

$$R(\chi_0, U) = \{x_1 \in X : \exists \tilde{u} \in U$$

and $\exists t \in [0, \infty)$ such that $\chi(t) = \chi_1\}$$

Under differential constraints, a sampling-based approach can be used by sampling the space of action trajectories to generate the time-limited reachable set. This results in a reduced set of possible action trajectories. The main difference in the current setting is that the approach here work with a sample sequence over $U$ to generate the reachable set.

**C. PERFORMANCE CONSTRAINTS**

In the optimization problem, the set of Eq. (41) symbolizes the inequality constraints in the flat output space. For example, $V$ is one of these constraints due to the limitation of system dynamics. Using Eq. (21), the speed can be bounded in terms of the parametrized curves as follows:

$$V_{min} \leq \sqrt{(W_1 \dot{\Phi}_1)^2 + (W_2 \dot{\Phi}_2)^2 + (W_3 \dot{\Phi}_3)^2} \leq V_{max}$$

The other performance parameters can also be bounded. For example, using the presented flat descriptions, the bank angle $\mu$ and the total lift factor $n$ can be limited as follows:

$$\mu_{min} \leq \arctan \left(\frac{\psi V \cos(\gamma)}{V \dot{\gamma} + g \cos(\gamma)}\right) \leq \mu_{max}$$

$$n^2 \leq n_{max}^2$$

where $n = \sqrt{n_x^2 + n_z^2}$. In the optimization problem, this kind of constraints can be enforced at the predefined points that are chosen uniformly over the time interval $[t_1, t_f]$. The equality constraints were also presented in Eq. (42). For example, the initial position and initial heading of aircraft are known and used as inputs. These variables can be described as equality constraints in the optimization scheme.
Let \([wp_x, wp_y]\) be the way-point that must be visited by aircraft at time \(t_w\), then this constraint can be represented as follows:

\[
W_1 \Phi_1(t_w) = wp_x \text{ and } W_2 \Phi_2(t_w) = wp_y
\]

(49)

In the same way, the heading constraint can be also formulated. Let \(\psi_d\) be the desired heading at time \(t_d\), then it is given as follows:

\[
\arctan(2(W_2 \Phi_2(t_d), W_1 \Phi_1(t_d))) = \psi_d
\]

(50)

D. Initialization Scheme

An optimization problem is said to be convex if the objective is a convex function, as well as the constraint set is a convex set. An optimization problem that violates either one of these conditions, i.e., one that has a non-convex objective, or a non-convex constraint set, or both, is called a non-convex optimization problem. A function \(f : S \rightarrow \mathbb{R}^n\) is convex if its domain \(S\) is a convex set and

\[
\forall s_1, s_2 \in S, \quad 0 \leq \alpha \leq 1 \Rightarrow \quad f(\alpha s_1 + (1 - \alpha)s_2) \leq \alpha f(s_1) + (1 - \alpha)f(s_2)
\]

(51)

In our problem, objective function and some of the constraints are non-convex. When dealing with a non-convex optimization problem, initial guess is important that can improve the performance of algorithm. The bounds of the decision variables should also be specified. In our case, the decision variables are control points that parametrize the location of the aircraft in \(x, y\) and \(h\). The bounds of control points can be specified via maximum speed and climb/descent ratio of the aircraft. Let \(\Gamma\) be the planning horizon, then the bounds for control points on \(x\)-axis and \(y\)-axis can be expressed as follows:

\[
(x_{\min}^b, x_{\max}^b) = (x_0 - 2V_{\max}\Gamma, x_0 + 2V_{\max}\Gamma)
\]

\[
(y_{\min}^b, y_{\max}^b) = (y_0 - 2V_{\max}\Gamma, y_0 + 2V_{\max}\Gamma)
\]

(52)

The control points that parametrize the \(z_1\) should be on the interval \([z_{\min}^b, z_{\max}^b]\) while the control points for \(z_2\) should be between \(y_{\min}^b\) and \(y_{\max}^b\). By setting bounds of each control point to these values, all of the feasible trajectories can be evaluated during optimization process without focusing on control points that violate the aircraft dynamics. As in the horizontal plane, the control points that specify the altitude can also be bounded via maximum rate of climb/descent:

\[
(h_{\min}^b, h_{\max}^b) = (h_0 - 2h_{\max}\Gamma, h_0 + 2h_{\max}\Gamma)
\]

(53)

The initial guess of the control points can be obtained via convex hull of the enemy’s reachable set. Let us define the convex set and convex hull. A set \(S \subseteq \mathbb{R}^{3n}\) is a convex set if the line segment connecting any pair of points of \(S\) lies entirely in \(S\), i.e.

\[
s_1, s_2 \in S, \quad 0 \leq \alpha \leq 1 \Rightarrow \alpha s_1 + (1 - \alpha)s_2 \in S
\]

(54)

The convex hull of a set \(S \subseteq \mathbb{R}^{3n}\) is the smallest convex set containing \(S\) [26], i.e.

\[
\text{conv}(S) := \bigcap \{ \tilde{S} \subseteq \mathbb{R}^{3n} : S \subseteq \tilde{S}, \tilde{S} \text{ is convex} \}
\]

(55)

For any finite convex set \(S = \{s_1, \ldots, s_k\}\) with \(k \in \mathbb{N}\) it follows that

\[
\text{conv}(S) = \bigg\{ \sum_{i=1}^{k} \alpha_is_i : \forall s_i \in S, \alpha_i \geq 0, i = 1, \ldots, k, \quad \sum_{i=1}^{k} \alpha_i = 1, k \in \mathbb{N} \bigg\}
\]

(56)

The control points can be initialized using the elements of the convex hull of enemy’s reachable set. The first control points are taken as current position of the fighter. The coordinates of the closest point on the convex hull to the current position of the aircraft are defined as second control points. The farthest point on the convex hull to the current location is used to specify the last control points. The other control vertices are taken as some of the points on the convex hull between closest and farthest points. Let us consider an illustrative example that the positions of the blue and red fighters are \([-250, 0, 3000]\) and \([250, 0, 3000]\) with 0 heading angle, respectively, as shown in Fig. 2. Assume that the performance constraints are as follows: \(V_{\min} = 30\ m/s, V_{\max} = 40\ m/s, \mu_{\min} = -30^\circ, \mu_{\max} = 30^\circ\), the number of control points that construct the curve for a flat output is 5 and the planning horizon is 20s. In this case, the reachable set of the red combatants consists of the grey points in the Fig. 2. The convex hull that contains the reachable set of red is illustrated via the black line in the Fig. 2. And, the initial control points on the horizontal plane for the blue aircraft are indicated via dashed blue circles that consist of the current position of the blue aircraft and the vertices on the convex hull of the red’s reachable set. These initial control points construct a curve that lies partly in the convex hull of the enemy’s reachable set. From perspective of aerial combat, this initialization strategy is logical that contains staying close to the enemy and intercepting enemy’s trajectory with an orientation advantage. If the aircraft has a disadvantage at the beginning of the combat, the strategy will also be logical that contains an initialized trajectory that closes to the dynamic limitations of the enemy. Therefore, the disadvantageous situation can also be resolved around this initialization.
E. SEQUENTIAL QUADRATIC PROGRAMMING

In this study, we use a Sequential Quadratic Programming (SQP)-based method that is presented in the paper [27] to solve the optimization problem in (39)-(42). Let us present the general nonlinear programming problem:

$$\min_{\chi \in \mathbb{R}^n} f(\chi)$$
subject to:
$$g_i(\chi) = 0, \quad i = 1, \ldots, m_e$$
$$g_i(\chi) \geq 0, \quad i = m_e + 1, \ldots, m$$
$$\chi_l \leq \chi \leq \chi_u$$

In the SQP-based methods, this problem is solved iteratively by updating the value of the vector $\chi$, which is initialized as $\chi^0$, via the following equation:

$$\chi^{k+1} := \chi^k + \alpha^k d^k$$

where superscript $k$ indicates the number of step. $d^k$ and $\alpha^k$ correspond to the search direction and the step length at the $k^{th}$ step, respectively. A quadratic programming sub-problem is solved to determine the search direction. This sub-problem is formulated by a quadratic approximation of the Lagrange function of the optimization problem and a linear approximation of the constraints $g_i$, where the Lagrange function is presented as follows:

$$L(\chi, \lambda) = f(\chi) - \sum_{i=1}^{m} \lambda_i g_i(\chi)$$

This sub-problem is of the following standard form of quadratic programming:

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} d^T B^k d + \nabla f(\chi^k) d$$
subject to:
$$\nabla g_i(\chi^k) d + g_i(\chi^k) = 0, \quad i = 1, \ldots, m_e$$
$$\nabla g_i(\chi^k) d + g_i(\chi^k) \geq 0, \quad i = m_e + 1, \ldots, m$$

in a general SQP scheme, $B$ is taken as $\nabla^2 L(\chi, \lambda)$. In the paper [27], which is the used algorithm in this study, the objective function is replaced by a linear least squares term that utilizes the stable factorization of the matrix $B$. The details about the update and approximation strategies can be found in [27].

F. RECEDING HORIZON CONTROL

Receding horizon control (RHC) corresponds to solving an optimization problem in a receding time horizon to generate control actions for a system. Using system model and current state information, RHC generates the control strategy over a planning horizon via solving constrained optimization problem, but only implements the control actions for the control horizon. The optimization problem is solved at the time $t$ over the interval $[t, t + \Gamma]$, where $\Gamma$ denotes the planning horizon. Then, the optimal trajectory is used to drive the system from time $t$ to time $t + \delta$, where the $\delta$ corresponds to control horizon. Every $\delta$ seconds, this process is repeated to find a new optimal trajectory. In this study, we use the receding horizon approach to produce the strategies of the combatants. At every $\delta$ seconds, using initial conditions at the current time $t$, the optimization problem in (39)-(42) is solved over the interval $[t, t + \Gamma]$ for blue and also red, separately, and then the strategies for both fighters are implemented during $\delta$ seconds.
V. IMPLEMENTATIONS AND RESULTS

We analyze two different scenarios to show the validity of the method. The first scenario corresponds to equivalent initial conditions without any air superiority, while the second case examines the effect of a fighter’s initial orientation advantage during aerial combat. The performance limitations of the combatants are defined as $V_{\text{min}} = 30\, \text{m/s}$, $V_{\text{max}} = 40\, \text{m/s}$, $\mu_{\text{min}} = -30^\circ$, $\mu_{\text{max}} = 30^\circ$ and $n_{\text{max}}^2 = 7$. The degree of the B-spline curves is chosen as 4. The shooting clearance is described via combat angles and the range between the combatants. The range, aspect angle and bearing angle thresholds are taken as 300 m, 30° and 30°, respectively. It is assumed that the attacker has a shooting clearance to win the combat, when the range, attacker’s aspect angle and attacker’s bearing angle are smaller than the defined threshold values. During the implementations, the planning horizon $\Gamma$ is taken as 20 seconds and the control horizon $\delta$ is defined as 10 seconds. Hence, at every 10 seconds, optimal strategies for both blue and red are generated by solving the optimization problem over the 20 seconds time horizon and the strategies are used over the interval $[t, t+10]$ for each solution.

The first scenario begins with tail-to-tail position of the combatants at $(x_0, y_0, h_0) = (0, 0, 3000)$. Both combatants have $35\, \text{m/s}$ speed at the beginning of the combat. None of aircraft has air superiority with these initial conditions. The results of the aerial combat are illustrated in Fig. 3 and Fig. 4.

As shown in Fig. 4a and 4b, the constrained variables are within limits during aerial combat. The speed, load factor and also bank angle do not violate the bounds. The algorithm generates feasible strategies. The positions of the combatants are presented in Fig. 3a, Fig. 3b and Fig. 3c. At the beginning of...
the combat, the fighters are in symmetric conditions without any air superiority. This equivalent situation continues until the end of the implementation and the combatant comes symmetric orientation again around the end of implementation as shown in Fig. 3b. The combatants make circular motions to get a clearance for shooting and also try to stay close during combat due to the range score. They select optimal strategies to prevent the opponent’s victory and get an advantageous position to win the combat. However, none of them gets air superiority during combat and they end the combat at equivalent conditions. These results show that the algorithm is capable of generating rational outcomes for aerial combat between two equivalent aircraft in the symmetric initial conditions without any air superiority.

In the second scenario, the blue aircraft has an orientation advantage at the beginning of the combat. The initial conditions are specified as $(x_0^b, y_0^b, h_0^b) = (-250, 0, 3000)$, $\psi_0^b = 0$, $V_0^b = 35$ and $(x_0^r, y_0^r, h_0^r) = (250, 0, 3000)$, $\psi_0^r = 0$, $V_0^r = 35$. The security strategies are generated as solutions of the optimization problems with presented receding horizon scheme and the results are illustrated in Fig. 5 and Fig. 6. At the beginning of the combat, the blue fighter has air superiority without sufficient range for a shooting.

This orientation advantage continues for 3 seconds with lack of range feasibility for shooting as shown in Fig. 6a and Fig. 6b. The position information of the fighters are represented in Fig. 5a, Fig. 5b and Fig. 5c. During combat, the blue aircraft pursues the red combatant to obtain an advantageous position for shooting, while red tries to avoid the disadvantaged orientation. Towards the end of the simulation, the range between aircraft is close to the threshold of the feasible range and blue has sufficient bearing angle, but it loses the necessary aspect angle for shooting as shown in Fig. 6a. As presented in the results, the red combatant manages to escape from blue’s shooting via optimal strategy that causes either orientation infeasibility or range infeasibility for a shooting.

VI. DISCUSSION AND CONCLUSION
In this paper, flatness-based game theoretic approach has been used to create a platform to generate optimal combat strategies during one-on-one aerial combat. The approach used flatness property of the dynamical system to generate control and state sequences in terms of flat outputs that were formulated with B-spline curves. Then, the aerial combat has been presented as an optimization problem in terms of parameterized curves by the help of security strategy approach in game theory. In the optimization scheme, all initial conditions and dynamical constraints have been presented to generate feasible trajectories. For the simulation and demonstration purposes, the method has been implemented to aerial combat between two UAVs and the results have been presented. The results show that the proposed methodology is capable of providing solutions to the air combat of autonomous aerial vehicles.

The proposed method does not use a subset of control inputs. All of the control actions within the performance envelope of the aircraft can be used during the evaluation process. As presented in the implementations, the best one within these continuous control actions is chosen as optimal strategy, and this control strategy can take any value within the performance envelope. However, the vast majority of the previous studies made discretization assumption and used a subset of control inputs. As mentioned before, there are several approaches for autonomous aerial combat such as approximate dynamic programming, rule-based approaches, game theoretic formulation, maneuver-based approach, influence diagram. They use different strategies to generate trajectories. But all of them use a subset of the control actions as action space. This action space can contain elemental maneuvers such as turn, go along, roll left with a predefined angle or advanced combat maneuvers such as immelman, split S, spiral dive. In these cases, the method cannot search all of the movement space of the fighter. However, the proposed method does not discretize the control inputs, so there is no drawback originated from using the subset of the control inputs. In the comparison of a method based on continuous input space and the other one that uses a subset of control inputs, the first one has an obvious advantage.
In the literature, there are some studies based on Hamilton-Jacobi equations that use continuous input space. These approaches have some inherited drawbacks such as curse of dimensionality and preassigned fighter roles. The proposed method overcomes these drawbacks. Another drawback of these approaches is the utilization of the simplified aircraft model. Benefiting from flatness property, the proposed method makes the use of more complex nonlinear aircraft dynamics possible. As presented in the implementations, in this case, complex dynamic parameters such as load factor can be restricted as a performance index and more realistic aircraft trajectories can be generated.

REFERENCES

[1] R. L. Shaw, *Fighter Combat*. Annapolis, MD, USA: Naval Institute Press, 1985.

[2] R. G. Sheffield, *The Official F-15 Strike Eagle Handbook*. New York, NY, USA: Compute Pu, 1990.

[3] P. Mitchell and C. J. Tomlin, “Overapproximating reachable sets by hamilton-jacobi projections,” *J. Sci. Comput.*, vol. 19, no. 1, pp. 323–346, 2003.

[4] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, “A time-dependent AC formulation of reachable sets for continuous dynamic games,” *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.

[5] A. Girard, C. Le Guernic, and O. Maler, “Efficient computation of reachable sets of linear time-invariant systems with inputs,” in *Proc. Int. Workshop Hybrid Syst.*, Comput. Control. Berlin, Germany: Springer, 2006, pp. 257–271.

[6] S. Stursberg and B. H. Krogh, “Efficient representation and computation of reachable sets for hybrid systems,” in *Proc. Int. Workshop Hybrid Syst.*, Comput. Control. Berlin, Germany: Springer, 2003, pp. 482–497.

[7] J. Sprinkle, J. M. Ekholn, H. J. Kim, and S. Sastry, “Encoding aerial pursuit/evasion games with fixed wing aircraft into a nonlinear model predictive tracking controller,” in *Proc. 43rd IEEE Conf. Decis. Control (CDC)*, vol. 3, 2004, pp. 2609–2614.

[8] J. M. Ekholm, J. Sprinkle, and S. Sastry, “Implementing and testing a nonlinear model predictive tracking controller for aerial pursuit/evasion games on a fixed wing aircraft,” in *Proc. Amer. Control Conf.*, Aug. 2005, pp. 1509–1514.

[9] J. H. Burgin and L. Sidor, “Rule-based air combat simulation,” *Titaun Syst. Inc.*, La Jolla, CA, USA: NASA Contractor Tech. Rep. 4160, 1988.

[10] V. Vafar, J. Karlakhi, and T. Raivio, “Modeling air combat by a moving horizon profile diagram game,” *J. Guid., Control, Dyn.*, vol. 29, no. 5, pp. 1080–1091, Sep. 2006.

[11] F. Austin, G. Carbone, M. Falco, H. Hinz, and M. Lewis, “Automated maneuvering decisions for air-to-air combat,” in *Proc. Guid., Navigat. Control Conf.*, Aug. 1987, p. 2393.

[12] F. Austin, G. Carbone, M. Falco, H. Hinz, and M. Lewis, “Game theory for automated maneuvering during air-to-air combat,” *J. Guid., Control, Dyn.*, vol. 13, no. 6, pp. 1143–1149, Nov. 1990.

[13] J. S. McGrew, “Real-time maneuvering decisions for autonomous air combat,” Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2008.

[14] J. S. McGrew, J. P. How, B. Williams, and N. Roy, “Air-combat strategy using approximate dynamic programming,” *J. Guid., Control, Dyn.*, vol. 33, no. 5, pp. 1641–1654, Sep. 2010.

[15] B. Baspinar and E. Koyuncu, “Aerial combat simulation environment for one-on-one engagement,” in *Proc. AIAA Model. Simul. technol. Conf.*, Jan. 2018, p. 0432.

[16] B. Baspinar and E. Koyuncu, “Evaluation of two-vs-one air combat using hybrid maneuver-based framework and security strategy approach,” *J. Aerounaut. Space Technol.*, vol. 12, no. 1, pp. 95–107, 2019.

[17] B. Baspinar and E. Koyuncu, “Differential flatness-based optimal air combat maneuver strategy generation,” in *Proc. AIAA Scitech Forum*, Jan. 2019, p. 1985.

[18] J. Levine, *Analysis and Control of Nonlinear Systems: A Flatness-Based Approach*. Berlin, Germany: Springer, 2009.

[19] 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, Jun. 1995.

[20] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, “A AC approach to equivalence and flatness of nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 44, no. 5, pp. 922–937, May 1999.

[21] B. Baspinar, H. Balakrishnan, and E. Koyuncu, “Mission planning and control of multi-aircraft systems with signal temporal logic specifications,” *IEEE Access*, vol. 7, pp. 155941–155950, 2019.

[22] L. Piegl and W. Tiller, *The NURBS book*. New York, NY, USA: Springer, 1992.

[23] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. Philadelphia, PA, USA: SIAM, 1998.

[24] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[25] D. L. Johnson, *F-16 Combat Aircraft Fundamentals*. Washington, DC, USA: United States Air Forces, 1996.

[26] F. Christophersen, *Optimal Control of Piecewise Affine Systems*, vol. 359. Berlin, Germany: Springer, 2007.

[27] D. Kraft, “A software package for sequential quadratic programming,” DLR German Aerosp. Center-Inst. Flight Mech., Cologne, Germany, Tech. Rep. DFVLR-FB 88-28, 1988.

BARİŞ BAŞPINAR received the B.S. degree 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. He was a Research Assistant with the Department of Aeronautical Engineering, ITU, since 2017. His current research interests include data-driven ATN 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 networks, survival analysis for complex air combat missions, air combat engagement modeling, guidance navigation, and control problems in aerial vehicles.

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 Boeing Research and Technology Europe, from 2013 to 2014, and the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology (MIT), from 2018 to 2019. He has been a Research Assistant with the ITU Aerospace Research Center (ITU ARC), Controls and Avionics Research Group, since 2013, and a Teaching Assistant with the Department of Aeronautical Engineering, ITU, since 2017. 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 networks, survival analysis for complex air combat missions, air combat engagement modeling, guidance navigation, and control problems in aerial vehicles.