JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 31, No. 4, July–August 2008

Optimal Interception of Evasive Missile Warheads: Numerical Solution of the Differential Game

Mauro Pontani* and Bruce A. Conway†
University of Illinois at Urbana–Champaign, Urbana, Illinois 61801

DOI: 10.2514/1.30893

The problem of interception of a ballistic missile warhead by a defending missile is formulated as a differential game. Each missile is given a modest postlaunch capability to maneuver. Interception concludes the game and occurs if the interceptor reduces the distance between it and the warhead to a specified value. The objective of the warhead is to minimize the final distance to the target, which lies on the Earth's surface but is not necessarily collocated with the interceptor launch site. The objective of the interceptor is to maximize this same distance at the capture time. Saddle-point equilibrium solutions are found using a recently developed, direct numerical method that nevertheless uses the analytical necessary conditions to find the optimal control for one of the players. The method requires an initial guess of the solution, and this is provided by generating an approximate solution using genetic algorithms. For initial conditions yielding eventual capture, we derive the necessary condition that the saddle-point trajectories terminate on the usable part of the terminal hypersurface; this condition is shown to have an interesting physical interpretation. The numerical method successfully finds the saddle-point trajectories and is used to determine the sensitivity of the value of the game to the capability of the interceptor, for design purposes, and also to gauge the robustness of the numerical method for the solution of this dynamic game.

Nomenclature

c_i	=	constraint in the genetic algorithm preprocessing ($i = 1, \ldots, n_c$)
d_{capt}	=	capture radius
d_f	=	distance between the intercontinental ballistic missile warheads and the target at t_f
H	=	Hamiltonian
H_f	=	Hamiltonian at t_f
H_0	=	Hamiltonian at t_0
J	=	objective function
$\bar{J}$	=	objective function used by the genetic algorithm
m_E	=	mass of the intercontinental ballistic missile warhead
m_P	=	mass of the defending missile
N	=	number of subarcs
R_E	=	Earth radius
$\mathbf{r}_E$	=	position vector of the evader
$\mathbf{r}_{Ef}$	=	position vector of the evader at t_f
$\mathbf{r}_P$	=	position vector of the pursuer
$\mathbf{r}_{Pf}$	=	position vector of the pursuer at t_f
T_E	=	thrust of the intercontinental ballistic missile warhead
T_P	=	thrust of the defending missile
t	=	time
t_L	=	time at the intercontinental ballistic missile launch
t_f	=	terminal time
t_0	=	initial time
$\tilde{\mathbf{u}}$	=	extended control ($\tilde{m}$-dimensional column vector)
$\mathbf{u}_E$	=	evader control ($\hat{m}_E$-dimensional column vector)
$\mathbf{u}_{Ef}$	=	evader control at t_f ($\hat{m}_E$-dimensional column vector) at t_f
$\mathbf{u}_P$	=	pursuer control ($\hat{m}_P$-dimensional column vector)
$\mathbf{u}_{Pf}$	=	pursuer control at t_f ($\hat{m}_P$-dimensional column vector)
V	=	value of the game
V_{P0}	=	magnitude of the pursuer velocity at t_0
$\mathbf{v}_E$	=	velocity vector of the evader
$\mathbf{v}_{Ef}$	=	velocity vector of the evader at t_f
$\mathbf{v}_P$	=	velocity vector of the pursuer
$\mathbf{v}_{Pf}$	=	velocity vector of the pursuer at t_f
$\tilde{\mathbf{x}}$	=	extended state ($\tilde{n}$-dimensional column vector)
$\mathbf{x}_E$	=	evader state (n_E-dimensional column vector)
$\mathbf{x}_{Ef}$	=	evader state at t_f (n_E-dimensional column vector)
x_{Ek}	=	kth component of the evader state
$\mathbf{x}_{E0}$	=	evader state at t_0 (n_E-dimensional column vector)
$\mathbf{x}_P$	=	pursuer state (n_P-dimensional column vector)
$\mathbf{x}_{Pf}$	=	pursuer state at t_f (n_P-dimensional column vector)
x_{Pk}	=	kth component of the pursuer state
$\mathbf{x}_{P0}$	=	pursuer state at t_0 (n_P-dimensional column vector)
$\boldsymbol{\gamma}_E$	=	evader strategy ($\hat{m}_E$-dimensional column vector)
$\boldsymbol{\gamma}_P$	=	pursuer strategy ($\hat{m}_P$-dimensional column vector)
$\Delta\phi$	=	angular displacement between the target and the defending missile launch site at t_0
δ_E	=	thrust pointing angle of the intercontinental ballistic missile warhead
δ_P	=	thrust pointing angle of the defending missile
θ_{g0}	=	Greenwich absolute longitude at t_0
θ_{gL}	=	Greenwich absolute longitude at t_L
$\boldsymbol{\lambda}_E$	=	adjoint variable conjugate to the evader equations of motion (n_E-dimensional column vector)
$\boldsymbol{\lambda}_E$	=	adjoint variable conjugate to the pursuer equations of motion (n_P-dimensional column vector)
μ_E	=	Earth planetary constant
$\boldsymbol{\nu}$	=	adjoint variable conjugate to the boundary conditions (q-dimensional column vector)
ϕ	=	Mayer term of the objective function
$\boldsymbol{\psi}$	=	function of boundary conditions (q-dimensional column vector)
ω_E	=	Earth rotation rate

I. Introduction

IN GENERAL, a flight path optimization problem can be treated as a one-sided optimization problem (optimal control problem) or a two-sided optimization problem. The one-sided optimization problem considers only one player and has been successfully applied

Received 8 March 2007; revision received 16 November 2007; accepted for publication 21 November 2007. Copyright © 2008 by Mauro Pontani and Bruce A. Conway. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0731-5090/08 $10.00 in correspondence with the CCC.

*Currently Post-Doctoral Fellow, Scuola di Ingegneria Aerospaziale, University of Rome "La Sapienza," 00184 Rome, Italy; mauro.pontani@uniroma1.it.

†Professor, Department of Aerospace Engineering; bconway@uiuc.edu. Associate Fellow AIAA.

to a variety of aerospace problems, for example, optimal trajectories for aircraft or spacecraft, for decades. However, the problem considered in this work, the optimal "combat" of a ballistic missile warhead and a defending interceptor missile, is best modeled using two competing players, that is, it becomes a two-sided optimization problem, a zero-sum two-player differential game. Zero-sum games belong to a special class of games in which two players with opposite aims are involved. This ballistic missile interception game is furthermore of a type referred to as a pursuit–evasion game, because it involves a pursuer P trying to catch an evader E. These games were first introduced by Isaacs [1,2].

It is relatively straightforward to derive the necessary conditions for optimality of the solution of the game. Bryson and Ho [3] do this as an extension of the analytical necessary conditions for the one-player game. Basar and Olsder [4] provide the necessary conditions for the existence of an open-loop representation of a saddle-point (SP) trajectory. Few problems are amenable to an analytic solution. Exceptions relevant here include the "homicidal chauffeur" problem [2], which may be considered an analog to air combat between a low-speed, highly maneuverable evader and high-speed, less-maneuverable pursuer, and a small number of air combat problems that have been solved analytically but only by using simplified dynamics [5–7].

For a problem with realistic dynamics (and perhaps realistic atmosphere and propulsion models), the only choice is numerical solution. However, only a limited number of studies have been done using this approach because the optimization of a realistic combat in which both vehicles maneuver optimally is very challenging to solve even numerically. Hillberg and Järmark [8] solved an air combat maneuvering problem in the horizontal plane with steady turn and realistic drag and thrust data. Järmark et al. [9] solved a qualitatively similar "tail-chase" air combat problem but used a very different method, differential dynamic programming, and considered only coplanar cases. A pursuit–evasion problem between missile and aircraft has been solved using an indirect, multiple shooting method by Breitner et al. [10,11]; Raivio and Ehtamo [12] solved a pursuit–evasion problem for a visual identification of the target by iterating a direct method. With regard to the ballistic missile defense problem formulated as a differential game problem, the work most qualitatively similar to this work is that of Shima and Shinar [13]. However, their interest is in finding a guidance law for the "end-game" stage, and for this, a linearized dynamics model is sufficient.

Horie and Conway [14] introduced a new numerical solution method for pursuit–evasion dynamic games and applied it to three-dimensional air combat between two F-16 fighters. Their method is very similar to recent direct methods for solution of one-sided optimal control problems that convert the continuous optimization problem into a nonlinear programming (NLP) problem [15–17]. However, these methods normally make no use of the analytical necessary conditions of the problem (the distinguishing feature of "direct" methods). To solve the pursuit–evasion problem, the analytical necessary conditions for one player are considered and are used to find the optimal control for that player, whereas the optimal control for the second player is found numerically. This requires of course that the costate variables for one player be included in the problem. Herman and Conway [17] called this method "semidirect." This is the method of solution chosen for the current problem, and it will be discussed at greater length in a subsequent section.

Ballistic missile defense (BMD) is a problem of great national interest. A system with limited capability is now being fielded and tested. This work addresses one of the mechanisms, that is, evasion, that might be employed by the incoming warhead or missile to defeat the defense system. If the evader maneuvers to delay or avoid interception, the defending missile must respond. The problem is easily formulated as a pursuit–evasion game. The BMD problem is thus to determine what is required on the part of the defending player, for example in terms of initial velocity and maneuvering capability, and so that intercept occurs at a significant distance from the defended target point even if the incoming warhead maneuvers (or "plays") optimally. This work makes a number of approximations; for example, it considers only coplanar motion of the warhead and interceptor trajectories, albeit over a rotating Earth, and it does not consider the effect of the atmosphere, though that is likely small considering how rapidly both missiles climb. In addition, as a dynamic game problem, it is assumed that the evader has continuous knowledge of the state of the pursuer. In real life, the evader is unlikely to possess this information, which it needs to execute the optimal evasion. However, the solution from game theory for the optimal strategy of the warhead can provide the "worst-case scenario" faced by the defender, which is very useful to know.

The objective of this work is thus to demonstrate the solution of the pursuit–evasion game form of the BMD problem via the semidirect method, derive the analytical necessary conditions that must be satisfied by the saddle-point trajectories as a partial verification of optimality, determine in a preliminary form the sensitivity of the game to the capability of the interceptor for design purposes, and finally, to gauge the robustness of the numerical method for the solution of this dynamic game.

II. Zero-Sum Differential Games

The problem of the optimal interception of an optimally evasive missile can be formulated as a zero-sum differential game in which the pursuer (the defending missile) tries to catch the evader (the attacking missile).

Each player drives the dynamic system through its own set of control variables (denoted by $\mathbf{u}_P$ for player P and $\mathbf{u}_E$ for player E). In this study, the dynamic system is described by an uncoupled pair of state equations, each related to a single player:

$$\dot{\mathbf{x}}_P = \mathbf{f}_P(\mathbf{x}_P, \mathbf{u}_P, t) \tag{1}$$

$$\dot{\mathbf{x}}_E = \mathbf{f}_E(\mathbf{x}_E, \mathbf{u}_E, t) \tag{2}$$

with $t_0 \le t \le t_f$. The initial time t_0 and the terminal time t_f can be either specified or free. In general, some components of the states at t_0 can be unspecified.

Usually, several boundary conditions hold for the problem at hand. In the most general formulation, these conditions can be collected in $\boldsymbol{\psi}$, which is a vector function of the starting and final values of the states and of the initial and terminal times:

$$\boldsymbol{\psi}\,(\mathbf{x}_{P0}\mathbf{x}_{E0}, \mathbf{x}_{Pf}, \mathbf{x}_{Ef}, t_0, t_f) = \mathbf{0} \tag{3}$$

A single terminal condition corresponding to the last component $\Psi_q[=\Psi_q(\mathbf{x}_{Pf}, \mathbf{x}_{Ef}, t_f)]$ of $\boldsymbol{\psi}$ is assumed. If the termination is the capture, this condition is expressed by requiring that the terminal distance between the two players is equal to a given value, namely, the *capture radius*.

Moreover, a problem of Mayer type is considered. Then, in the most general case of such a problem, the objective cost is given as a function of the initial and terminal values of the states and of the initial and terminal times:

$$J = \phi(\mathbf{x}_{P0}, \mathbf{x}_{E0}, \mathbf{x}_{Pf}, \mathbf{x}_{Ef}, t_0, t_f) \tag{4}$$

In this zero-sum game context, the pursuer plays to minimize J, whereas the evader plays to maximize J.

Two feedback strategies, $\boldsymbol{\gamma}_P$ and $\boldsymbol{\gamma}_E$, can be introduced for the two players so that the control variables may be expressed as functions of the state variables:

$$\mathbf{u}_P(t) = \boldsymbol{\gamma}_P(\mathbf{x}_P, \mathbf{x}_E, t) \quad \text{and} \quad \mathbf{u}_E(t) = \boldsymbol{\gamma}_E(\mathbf{x}_P, \mathbf{x}_E, t) \tag{5}$$

The *value* V of the game is defined as the outcome of the objective function when both players employ their optimal strategies along the optimal path:

$$V = \min_{\mathbf{u}_P(t)} \max_{\mathbf{u}_E(t)} J = \max_{\mathbf{u}_E(t)} \min_{\mathbf{u}_P(t)} J \quad \text{yielding}$$
$$\begin{cases} \mathbf{u}_P^*(t) = \boldsymbol{\gamma}_P^*[\mathbf{x}_P^*(t), \mathbf{x}_E^*(t), t] \\ \mathbf{u}_E^*(t) = \boldsymbol{\gamma}_E^*[\mathbf{x}_P^*(t), \mathbf{x}_E^*(t), t] \end{cases} \qquad (t_0^* \le t \le t_f^*) \tag{6}$$

The superscript $*$ denotes the optimal solution for the corresponding variables.

The existence of the value is assumed because the dynamic system is separable [2], and this is a sufficient condition for the interchangeability of the minimum and maximum operators in Eq. (6). In zero-sum differential games, a pair of optimal strategies corresponds to a saddle-point equilibrium solution and produces a saddle-point trajectory in the state space.

By definition, an open-loop representation of an optimal feedback strategy is the strategy along the optimal trajectory as a function of the initial states only. Necessary conditions for the existence of an open-loop representation of a saddle-point solution are provided by Basar and Olsder [4] and reported by Horie and Conway [14].

A Hamiltonian H and a function of terminal conditions Φ are introduced:

$$H = \boldsymbol{\lambda}_P^T \mathbf{f}_P + \boldsymbol{\lambda}_E^T \mathbf{f}_E \tag{7}$$

$$\Phi = \phi + \boldsymbol{\nu}^T \boldsymbol{\psi} \tag{8}$$

where $\boldsymbol{\lambda}_P(t)$, $\boldsymbol{\lambda}_E(t)$, and $\boldsymbol{\nu}$ are the adjoint variables conjugate to the dynamic equations of the pursuer, the evader, and with the boundary conditions, respectively. Because of the uncoupled form of the state equations, the Hamiltonian is separable, thus ensuring the existence of the value of the game as stated previously.

The following variational equations must hold for the adjoints $\boldsymbol{\lambda}_P$ and $\boldsymbol{\lambda}_E$:

$$\dot{\boldsymbol{\lambda}}_P = -\frac{\partial H}{\partial \mathbf{x}_P} = -\left[\frac{\partial \mathbf{f}_P}{\partial \mathbf{x}_P}\right]^T \boldsymbol{\lambda}_P \tag{9}$$

$$\dot{\boldsymbol{\lambda}}_E = -\frac{\partial H}{\partial \mathbf{x}_E} = -\left[\frac{\partial \mathbf{f}_E}{\partial \mathbf{x}_E}\right]^T \boldsymbol{\lambda}_E \tag{10}$$

in conjunction with the boundary conditions:

$$\lambda_{Pk}(t_0) + \frac{\partial \Phi}{\partial x_{Pk}(t_0)} = 0 \quad \text{if } x_{Pk}(t_0) \text{ is unspecified } (k = 1, \ldots, n_P) \tag{11}$$

$$\lambda_{Pk}(t_f) - \frac{\partial \Phi}{\partial x_{Pk}(t_f)} = 0 \quad \text{if } x_{Pk}(t_f) \text{ is unspecified } (k = 1, \ldots, n_P) \tag{12}$$

$$\lambda_{Ek}(t_0) + \frac{\partial \Phi}{\partial x_{Ek}(t_0)} = 0 \quad \text{if } x_{Ek}(t_0) \text{ is unspecified } (k = 1, \ldots, n_E) \tag{13}$$

$$\lambda_{Ek}(t_f) - \frac{\partial \Phi}{\partial x_{Ek}(t_f)} = 0 \quad \text{if } x_{Ek}(t_f) \text{ is unspecified } (k = 1, \ldots, n_E) \tag{14}$$

Equations (11) and (13) (involving the initial values of $\boldsymbol{\lambda}_P$ and $\boldsymbol{\lambda}_E$) are the counterparts of Eqs. (12) and (14), which refer to the terminal time.

In addition, a saddle-point solution must satisfy the following pair of equations:

$$\mathbf{u}_P = \arg\min_{\mathbf{u}_P} H = \arg\min_{\mathbf{u}_P} (\boldsymbol{\lambda}_P^T \mathbf{f}_P) \tag{15}$$

$$\mathbf{u}_E = \arg\max_{\mathbf{u}_E} H = \arg\max_{\mathbf{u}_E} (\boldsymbol{\lambda}_E^T \mathbf{f}_E) \tag{16}$$

Unbounded control variables are assumed for the problem at hand, and so the necessary conditions in Eqs. (15) and (16) can be replaced by the following equations:

$$\left[\frac{\partial H}{\partial \mathbf{u}_P}\right]^T = \left[\frac{\partial \mathbf{f}_P}{\partial \mathbf{u}_P}\right]^T \boldsymbol{\lambda}_P = \mathbf{0} \tag{17}$$

$$\left[\frac{\partial H}{\partial \mathbf{u}_E}\right]^T = \left[\frac{\partial \mathbf{f}_E}{\partial \mathbf{u}_E}\right]^T \boldsymbol{\lambda}_E = \mathbf{0} \tag{18}$$

in conjunction with the following second-order conditions:

$$H_{\mathbf{u}_P \mathbf{u}_P} = \frac{\partial^2 H}{\partial \mathbf{u}_P^2} \geq 0 \tag{19}$$

$$H_{\mathbf{u}_E \mathbf{u}_E} = \frac{\partial^2 H}{\partial \mathbf{u}_E^2} \leq 0 \tag{20}$$

that is, the Hessian matrix $H_{\mathbf{u}_P \mathbf{u}_P}$ must be positive semidefinite, whereas the Hessian $H_{\mathbf{u}_E \mathbf{u}_E}$ must be negative semidefinite. Equations (17) and (18) ensure that the solution is stationary with respect to the control variables $\mathbf{u}_P$ and $\mathbf{u}_E$. Because P wants to minimize J, whereas E wants to maximize J, Eqs. (19) and (20) constitute second-order necessary conditions, which enforce the respective first-order conditions in Eqs. (17) and (18).

Finally, the transversality conditions are written as follows:

$$\frac{\partial \Phi}{\partial t_0} - H_0 = 0 \quad \text{if } t_0 \text{ is unspecified} \tag{21}$$

$$\frac{\partial \Phi}{\partial t_f} + H_f = 0 \quad \text{if } t_f \text{ is unspecified} \tag{22}$$

The last two relationships are needed only when the terminal time or the initial time is free: Eq. (21) becomes unnecessary if t_0 is specified, and Eq. (22) becomes unnecessary if t_f is specified.

Equations (1–3), (9–14), and (17–22) constitute a two-point boundary-value problem (TPBVP), in which the unknowns are the state vectors $\mathbf{x}_P(t)$ and $\mathbf{x}_E(t)$, the control variables $\mathbf{u}_P(t)$ and $\mathbf{u}_E(t)$, the Lagrange multipliers $\boldsymbol{\lambda}_P(t)$, $\boldsymbol{\lambda}_E(t)$, and $\boldsymbol{\nu}$, and possibly the times t_0 and t_f.

During the evolution of the differential game, the pursuer P tries to force the state toward a terminal manifold, henceforward referred to as the *terminal surface*. This manifold corresponds to those terminal states that satisfy the equation $\boldsymbol{\psi}_q(\mathbf{x}_{Pf}, \mathbf{x}_{Ef}, t_f) = 0$; hence, the terminal surface represents the capture in the state space. The evader E tries to avoid the capture. In differential games of pursuit–evasion, there exists a singular surface, termed the *barrier* by Isaacs [2], that divides the state space into a capture zone and an evasion zone. In the capture zone, if P plays optimally, then he can catch E regardless of E's strategy; in the evasion zone, if E plays optimally, then he can avoid the capture altogether. Hence, the barrier is said to be semipermeable, which means that a player is not able to cross it without cooperation, that is, if the opponent player plays optimally. The barrier terminates on the terminal surface, and its intersection with the terminal surface constitutes the boundary of the *usable part* (UP) of the terminal surface (Fig. 1). Hence, due to the semipermeability of the barrier, a saddle-point trajectory is expected to terminate on the usable part of the terminal surface, provided that the initial state was inside the capture zone. An interesting property holds for the UP, which can be used to identify those points belonging to it [2]:

$$\left[\frac{\partial \boldsymbol{\psi}_q}{\partial \mathbf{x}_{Pf}} \quad \frac{\partial \boldsymbol{\psi}_q}{\partial \mathbf{x}_{Ef}}\right] \left[\mathbf{f}_P^T(\mathbf{x}_{Pf}, \mathbf{u}_{Pf}, t_f) \quad \mathbf{f}_E^T(\mathbf{x}_{Ef}, \mathbf{u}_{Ef}, t_f)\right]^T \leq 0 \tag{23}$$

As the game is assumed to terminate with capture, formally expressed by $\boldsymbol{\psi}_q(\mathbf{x}_{Pf}, \mathbf{x}_{Ef}, t_f) = 0$, the necessary conditions described previously can yield saddle-point solutions only when the game has starting conditions inside the capture zone. In the evasion zone, the game must be reformulated because a saddle-point trajectory will not lead to capture. Hence, a different termination

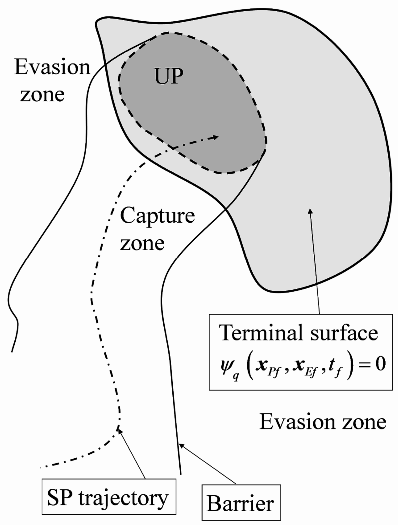

Evasion zone
UP
Capture zone
Terminal surface $\psi_q\left(x_{Pf}, x_{Ef}, t_f\right) = 0$
Evasion zone
SP trajectory
Barrier

Fig. 1 Schematic representation of a SP trajectory in a three-dimensional state space.

condition is to be used, and different necessary conditions will be derived. As a consequence, in each of the two zones separated by the barrier, the optimal feedback strategies must be computed separately. These considerations are also consistent with the observations made by Breitner et al. [10,11].

In this paper, starting conditions leading to the capture are assumed. The resulting optimal trajectories are then checked by employing the UP condition in Eq. (23). The latter condition is strictly related to the geometry of the capture; this property will be discussed in Sec. VI.

III. Two-Sided Optimization via Semidirect Collocation with Nonlinear Programming Algorithm

In the previous section, the necessary conditions for an optimal open-loop representation of the feedback strategies were derived. These conditions, in conjunction with the dynamic equations and the boundary conditions, form a two-point boundary-value problem (TPBVP).

Realistic problems are usually characterized by strong nonlinearities and can involve a large number of state components. As a consequence, even some TPBVPs related to optimal control problems can be very challenging to solve through indirect algorithms because they explicitly employ the adjoint variables. This circumstance has two unfavorable consequences. First, the problem is increased in size due to the use of the adjoint variables. In addition, indirect algorithms need the initial guesses for these usually nonintuitive variables. Hence, a much more robust approach is based on the use of direct algorithms. A direct method, the direct collocation with nonlinear programming (DCNLP) algorithm, has been successfully employed to solve a wide variety of aerospace optimal control problems [18–20]. Its basic features are the following:

1) The continuous problem is discretized in time, and so only the values of the state and of the control at discrete times are employed by the algorithm.

2) The differential equations governing the system dynamics are translated into nonlinear algebraic equations by means of high-order quadrature rules (such as the Gauss–Lobatto quadrature rules [17]).

3) The resulting nonlinear programming problem is solved by software for solving constrained nonlinear programs, for instance NPSOL [21], which does not need the adjoint variables because they are not explicitly used to yield the solution.

Unfortunately, such an approach cannot be directly employed to solve the differential game at hand because the numerical NLP solver is able to generate an optimal (minimizing) solution only, that is, it can consider only a single objective function. However, Horie and Conway [22] have recently developed a method based on the DCNLP algorithm; this method is able to transform a differential game into a single objective problem in such a form that the numerical optimizer can treat it as an optimal control problem. This method is based on the following points:

1) The adjoint variables for one player, for instance, the evader, are included in the DCNLP. The related optimality conditions are used and the control for E is expressed as a function of E's adjoint variables.

2) The optimal control for the other player, for example, the pursuer, is found numerically by the optimizer.

The method was termed "semidirect collocation with nonlinear programming" (semi-DCNLP) by Horie and Conway [22] because it makes use of the adjoint variables for one player, whereas typically direct methods do not explicitly employ any Lagrange multipliers. This approach was successfully applied by Horie and Conway to the challenging problem of the two-sided optimization of the maneuvers of two opposing F-16 fighter aircraft [14,23].

To use the semi-DCNLP algorithm, first, through the analytical necessary conditions in Eqs. (18) and (20), the control for the evader is written as a function of $\mathbf{x}_E$, $\boldsymbol{\lambda}_E$, and t: $\mathbf{u}_E = \mathbf{u}_E(\mathbf{x}_E, \boldsymbol{\lambda}_E, t)$. The extended control variable $\tilde{\mathbf{u}}$ now includes $\mathbf{u}_P$ only: $\tilde{\mathbf{u}} = \mathbf{u}_P$. Then, an extended "state" vector $\tilde{\mathbf{x}}$ is defined after the inclusion of the adjoint variable $\boldsymbol{\lambda}_E$:

$$\tilde{\mathbf{x}}(t) = \left[\, \mathbf{x}_P^T(t) \quad \mathbf{x}_E^T(t) \quad \boldsymbol{\lambda}_E^T(t) \,\right]^T \tag{24}$$

The corresponding dynamic equation is

$$\dot{\tilde{\mathbf{x}}}(t) = \left[\mathbf{f}_P^T \;\; \mathbf{f}_E^T \quad -\boldsymbol{\lambda}_E^T \left[\frac{\partial \mathbf{f}_E}{\partial \mathbf{x}_E}\right] \right]^T = \tilde{\mathbf{f}}$$
$$\text{where } \mathbf{f}_E^T = \mathbf{f}_E^T[\mathbf{x}_E, \mathbf{u}_E(\mathbf{x}_E, \boldsymbol{\lambda}_E, t), t] = \mathbf{f}_E^T(\mathbf{x}_E, \boldsymbol{\lambda}_E, t) \tag{25}$$

Equations (13) and (14) constitute the boundary conditions for the adjoint variable $\boldsymbol{\lambda}_E$. Each of these equations can possibly depend on $\boldsymbol{\nu}$. The left-hand sides of the equations, which are independent of $\boldsymbol{\nu}$, are collected in the vector $\boldsymbol{\psi}_{\text{EXT}}$. The remaining equations are opportunely handled to eliminate the components of $\boldsymbol{\nu}$; this procedure will be detailed in Sec. VI, where it is directly applied. Then, also the left-hand sides of the resulting equations, which do not yet include any component of $\boldsymbol{\nu}$, are inserted in $\boldsymbol{\psi}_{\text{EXT}}$. The extended terminal function $\tilde{\Phi}$ now includes $\tilde{\boldsymbol{\psi}} = [\, \boldsymbol{\psi}^T \quad \boldsymbol{\psi}_{\text{EXT}}^T \,]^T$, whereas the extended Hamiltonian includes an additional term corresponding to the differential equation for $\boldsymbol{\lambda}_E$:

$$\tilde{\Phi} = \phi + \tilde{\boldsymbol{\nu}}^T \tilde{\boldsymbol{\psi}} = \phi \boldsymbol{\nu}_1^T \boldsymbol{\psi} + \boldsymbol{\nu}_{\text{EXT}}^T \boldsymbol{\psi}_{\text{EXT}} \quad \text{where } \tilde{\boldsymbol{\nu}} = \left[\, \boldsymbol{\nu}_1^T \quad \boldsymbol{\nu}_{\text{EXT}}^T \,\right]^T \tag{26}$$

$$\tilde{H} = \tilde{\boldsymbol{\lambda}}^T \tilde{\mathbf{f}} = \boldsymbol{\lambda}_{P(e)}^T \mathbf{f}_P + \boldsymbol{\lambda}_{E(e)}^T \mathbf{f}_E - \boldsymbol{\lambda}_{\lambda(e)}^T \left[\frac{\partial \mathbf{f}_E}{\partial \mathbf{x}_E}\right]^T \boldsymbol{\lambda}_E$$
$$\text{where } \tilde{\boldsymbol{\lambda}} = \left[\, \boldsymbol{\lambda}_{P(e)}^T \quad \boldsymbol{\lambda}_{E(e)}^T \quad \boldsymbol{\lambda}_{\lambda(e)}^T \,\right]^T \tag{27}$$

Hence, the zero-sum game has been converted into a typical optimal control problem. The numerical optimizer has to solve the problem:

$$\min_{\tilde{\mathbf{u}}(t)} J \quad \text{subject to the constraints} \quad \dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{f}} \quad \text{and} \quad \tilde{\boldsymbol{\psi}} = \mathbf{0} \tag{28}$$

For the transcription of the continuous problem into a discrete problem, the time interval is partitioned into N subarcs: $[t_{i-1}, t_i]_{i=1,\ldots,N}$, with $t_N \equiv t_f$. Each subinterval has a length equal to $\Delta T = t_i - t_{i-1} = (t_f - t_0)/N$. The method of solution requires the discretization of the continuous variables, that is, the state components, which are to be represented by a set of parameters. Their number corresponds to the degree of the polynomial, which approximates each variable in each subarc. In this research, the fifth-degree Gauss–Lobatto quadrature rule is employed because it is characterized by an improved accuracy when compared with lower degree rules such as Simpson's rule [17]. Therefore, in each subinterval, six coefficients for each state component are required.

These coefficients are the values of the state component at the initial, at the central, and at the terminal point of each subarc and the values of the corresponding time derivatives at the same points. These derivatives are simply given by the right-hand sides of the state equations, whereas the discrete values at the initial, central, and final points are properly the parameters that represent the state component under consideration. In brief, if the state $\tilde{\mathbf{x}}$ has $\tilde{n}$ components, it is represented by $(2\tilde{n}N + \tilde{n})$ parameters, $\{\tilde{\mathbf{x}}_{i-1}, \tilde{\mathbf{x}}_{Ci}, \tilde{\mathbf{x}}_i\}_{i=1,\ldots,N}$, and the corresponding time derivatives are $\{\tilde{\mathbf{f}}_{i-1}, \tilde{\mathbf{f}}_{Ci}, \tilde{\mathbf{f}}_i\}_{i=1,\ldots,N}$.

The fifth-order integration rules require the evaluation of the approximate values of the state vector at two collocation points, corresponding to

$$t = t_i^{(A)} = t_{Ci} - 0.5\Delta T\sqrt{3/7}$$

and

$$t = t_i^{(B)} = t_{Ci} + 0.5\Delta T\sqrt{3/7}$$

where $t_{Ci} = 0.5(t_i + t_{i-1})$ [16]:

$$\begin{aligned}\tilde{\mathbf{x}}_i^{(A)} = {} & \frac{1}{686}[(39\sqrt{21} + 231)\tilde{\mathbf{x}}_{i-1} + 224\tilde{\mathbf{x}}_{Ci} \\ & + (-39\sqrt{21} + 231)\tilde{\mathbf{x}}_i] + \frac{\Delta T}{686}[(3\sqrt{21} + 21)\tilde{\mathbf{f}}_{i-1} \\ & - 16\sqrt{21}\tilde{\mathbf{f}}_{Ci} + (3\sqrt{21} - 21)\tilde{\mathbf{f}}_i]\end{aligned} \tag{29}$$

$$\begin{aligned}\tilde{\mathbf{x}}_i^{(B)} = {} & \frac{1}{686}[(-39\sqrt{21} + 231)\tilde{\mathbf{x}}_{i-1} + 224\tilde{\mathbf{x}}_{Ci} \\ & + (39\sqrt{21} + 231)\tilde{\mathbf{x}}_i] + \frac{\Delta T}{686}[(-3\sqrt{21} + 21)\tilde{\mathbf{f}}_{i-1} \\ & + 16\sqrt{21}\tilde{\mathbf{f}}_{Ci} + (-3\sqrt{21} - 21)\tilde{\mathbf{f}}_i]\end{aligned} \tag{30}$$

Then these values, in conjunction with the discrete values of the control variable $\tilde{\mathbf{u}}$ at $t = t_i^{(A)}$ and at $t = t_i^{(B)}$, are used to evaluate

$$\mathbf{f}_i^{(A)} = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}_i^{(A)}, \tilde{\mathbf{u}}_i^{(A)}, t_i^{(A)})$$

and

$$\mathbf{f}_i^{(B)} = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}_i^{(B)}, \tilde{\mathbf{u}}_i^{(B)}, t_i^{(B)})$$

$(i = 1, \ldots, N)$. The system constraint includes the following pair of equations [16]:

$$\begin{aligned}\mathbf{C}_i^{(A)} = {} & \frac{1}{360}[(32\sqrt{21} + 180)\tilde{\mathbf{x}}_{i-1} - 64\sqrt{21}\tilde{\mathbf{x}}_{Ci} \\ & + (32\sqrt{21} - 180)\tilde{\mathbf{x}}_i] + \frac{\Delta T}{360}[(9 + \sqrt{21})\tilde{\mathbf{f}}_{i-1} + 98\mathbf{f}_i^{(A)} \\ & + 64\tilde{\mathbf{f}}_{Ci} + (9 - \sqrt{21})\tilde{\mathbf{f}}_i] = \mathbf{0}\end{aligned} \tag{31}$$

$$\begin{aligned}\mathbf{C}_i^{(B)} = {} & \frac{1}{360}[(-32\sqrt{21} + 180)\tilde{\mathbf{x}}_{i-1} + 64\sqrt{21}\tilde{\mathbf{x}}_{Ci} \\ & + (-32\sqrt{21} - 180)\tilde{\mathbf{x}}_i] + \frac{\Delta T}{360}[(9 - \sqrt{21})\tilde{\mathbf{f}}_{i-1} \\ & + 98\mathbf{f}_i^{(B)} + 64\tilde{\mathbf{f}}_{Ci} + (9 + \sqrt{21})\tilde{\mathbf{f}}_i] = \mathbf{0}\end{aligned} \tag{32}$$

Equations (31) and (32) are equivalent to $2\tilde{n}N$ scalar constraints.

The control variable $\tilde{\mathbf{u}}$ is represented through the discrete values $\tilde{\mathbf{u}}_{i-1}$, $\tilde{\mathbf{u}}_{Ci}$, and $\tilde{\mathbf{u}}_i$ (at $t = t_{i-1}$, t_{Ci}, and t_i) and, in addition, through the discrete values at the collocation points $\tilde{\mathbf{u}}_i^{(A)}$ and $\tilde{\mathbf{u}}_i^{(B)}$, which are needed to evaluate $\tilde{\mathbf{f}}_i^{(A)}$ and $\tilde{\mathbf{f}}_i^{(B)}$. Hence, if $\tilde{\mathbf{u}}$ has $\tilde{m}$ components, then it is represented by $(4\tilde{m}N + \tilde{m})$ parameters.

Once the NLP solver has produced the result for the optimal control problem of interest, the parameters $\{\tilde{\mathbf{x}}_{i-1}, \tilde{\mathbf{x}}_{Ci}, \tilde{\mathbf{x}}_i\}_{i=1,\ldots,N}$ and $\{\tilde{\mathbf{u}}_{i-1}, \tilde{\mathbf{u}}_i^{(A)}, \tilde{\mathbf{u}}_{Ci}, \tilde{\mathbf{u}}_i^{(B)}, \tilde{\mathbf{u}}_i\}_{i=1,\ldots,N}$ obtained are employed to evaluate $\{\tilde{\mathbf{f}}_{i-1}, \tilde{\mathbf{f}}_{Ci}, \tilde{\mathbf{f}}_i\}_{i=1,\ldots,N}$. Finally, the state components are interpolated through fifth-order polynomials, which represent the continuous approximations of their optimal time histories.

The solution of the system in Eq. (28) also satisfies the necessary conditions for an open-loop representation of the feedback saddle-point trajectory when the following conditions hold:

$$\boldsymbol{\lambda}_{E(e)} = \boldsymbol{\lambda}_E \quad \text{and} \quad \boldsymbol{\nu}_{\text{EXT}} = \mathbf{0} \tag{33}$$

This property was demonstrated by Horie [23] and is consistent with the result in Papavassilopoulos and Cruz [24]. The multipliers $\boldsymbol{\lambda}_{E(e)}$ and $\boldsymbol{\nu}_{\text{EXT}}$ are generated by the numerical optimizer, once the solution to the optimal control problem in Eq. (28) is achieved. The conditions in Eq. (33) can be used to check whether this solution actually corresponds to a saddle-point solution, thus providing a possible final verification on the numerical trajectory produced by the method.

IV. Genetic Algorithm Preprocessing

The inclusion of the adjoint equations for $\boldsymbol{\lambda}_E$ in the method of solution based on the DCNLP algorithm has the unfavorable consequence that a starting guess for these adjoint variables is needed. The initial values for $\boldsymbol{\lambda}_E$ can be provided by using a trial-and-error approach, especially when the size of the problem is small. However, the adjoint variables usually have a nonintuitive significance, and the trial-and-error selection of first attempt values for them is thus completely arbitrary, and so this approach is likely to be unsuccessful (and has often been unsuccessful in our experience), especially for large problems.

This difficulty can be resolved by using a completely different, and, above all, systematic approach. Genetic algorithms (GA) do not require any guess for the variables involved in the problem. The parameters are coded as binary arrays, and a complete set of them forms an individual. Suitable operators, such as crossover and mutation, govern the reproduction mechanism, and the population is improved (with reference to the objective function) generation after generation. Elitism is an additional operator, which is usually employed to preserve the best individual in the generation to come. As a result, at the end of the process, the best individual includes the optimal (coded) values of the parameters involved. Basically, genetic algorithms constitute an effective statistical search technique for selecting the best parameters, that is, the parameters that minimize the objective function. Genetic algorithms do not need any guess because the starting population is randomly generated. However, they require the definition of the search space for all the parameters, that is, the ranges in which the values of the parameters are to be searched.

An intrinsic limit of the GA is its poor numerical accuracy, which is mainly related to the representation of the parameters as binary arrays with a finite number of digits. However, when used as a preprocessing technique, the GA is only required to provide a reasonable guess for the NLP solver by selecting parameters in a suitable feasible region of the search space.

All the equations that form the TPBVP, that is, Eqs. (1–3), (9–14), and (17–22), are to be employed by the GA.

The boundary conditions in Eqs. (3) and (11–14) and the transversality conditions in Eqs. (21) and (22) are assimilated to constraints $\{c_i\}_{i=1,\ldots,n_C}$ and used as penalty terms in the objective function $\bar{J}$ to be minimized by the GA:

$$\bar{J} = \sum_{i=1}^{n_C} k_i c_i^2 \tag{34}$$

where the weights k_i are given positive numbers, and n_C is the total number of constraints. The relationships in Eqs. (17–20) are used to express the control variables for both players as functions of $\mathbf{x}_P$, $\mathbf{x}_E$, $\boldsymbol{\lambda}_P$, $\boldsymbol{\lambda}_E$, and t:

$$\mathbf{u}_P = \mathbf{u}_P(\mathbf{x}_P, \boldsymbol{\lambda}_P, t) \quad \text{and} \quad \mathbf{u}_E = \mathbf{u}_E(\mathbf{x}_E, \boldsymbol{\lambda}_E, t) \tag{35}$$

The parameter set is composed of all the unspecified initial and terminal values of the states and of the adjoint variables, and of the possible components of $\boldsymbol{\nu}$ included in Eqs. (11–14).

The GA is employed to solve this well-posed TPBVP in which the number of the unknown parameters is equal to the number of constraints. The selection of the objective function in Eq. (34) is related to this circumstance because the GA thus has to minimize the constraint violations only.

Some basic settings must be chosen by the user, such as the number of bits and an appropriate range for each parameter, the population size, and the number of generations. Additional options concern the mutation probability and the specific type of crossover which the GA has to adopt. However, with the proper settings, the GA is expected to generate a reasonable guess for the NLP solver, as previously demonstrated for a dynamic game by Horie and Conway [22].

V. Problem Definition

In this research, the problem of the optimal interception of an optimally evasive missile is considered. The performance of the attacking missile can be assimilated to that of an intercontinental ballistic missile (ICBM). Its purpose is to impact a specified target point on the Earth's surface. However, it is assumed that such a missile has some kind of maneuvering capability to be able to perform suitable escape actions whenever an intercepting missile should be used to defend that target. Similarly, the interceptor missile is able to perform maneuvers for the capture of the opponent missile. In this conflicting situation, the ICBM desires to get a position as close as possible to the target before being intercepted. On the contrary, the interceptor desires to keep the opponent as far as possible from the target.

Interception terminates the combat game, and some self-evident conditions will be stated in the next section to ensure that the capture can occur before the ICBM impact. Should that not be the case, the problem should be completely reformulated by adopting a different terminal condition.

In realistic scenarios, both missiles will employ all their maneuvering capabilities, that is, both missiles will use their maximum thrust during the combat. In this research, for both missiles, a constant thrust over mass ratio is assumed along the whole trajectory, that is, the control thrust direction is the only control.

This study employs a point-mass model to describe the trajectories of the two missiles in the context of a two-degrees-of-freedom problem of optimal interception. The motion of the two missiles is confined to the equatorial plane, and the target is located on the equator of the rotating Earth. The initial time corresponds to the end of the interceptor vertical ascent phase, just before it chooses its initial flight path angle. At the same time, the ICBM has specified position and velocity, which are such that a possible completely ballistic trajectory would lead it to impact on the target.

The state vector is composed of four variables for each missile: the radial component of the velocity v_r, the horizontal component of the velocity v_θ, the radius r, and the absolute longitude ξ:

$$\mathbf{x} = \begin{bmatrix} v_{rP} & v_{\theta P} & r_P & \xi_P & v_{rE} & v_{\theta E} & r_E & \xi_E \end{bmatrix}^T \tag{36}$$

In this study, atmospheric drag is not considered because most of the two trajectories correspond to altitudes at which the atmospheric density can be assumed negligible. The equations of motion may be written in the rotating frames of the two missiles, shown in Fig. 2:

$$\begin{aligned} &P\begin{cases} \dot{v}_{rP} = \frac{T_P}{m_P}\sin\delta_P - \frac{\mu_E - v_{rP}^2 r_P}{r_P^2} \\ \dot{v}_{\theta P} = \frac{T_P}{m_P}\cos\delta_P - \frac{v_{rP}v_{\theta P}}{r_P} \\ \dot{r}_P = v_{rP} \\ \dot{\xi}_P = \frac{v_{\theta P}}{r_P} \end{cases} \\ &E\begin{cases} \dot{v}_{rE} = \frac{T_E}{m_E}\sin\delta_E - \frac{\mu_E - v_{rE}^2 r_E}{r_E^2} \\ \dot{v}_{\theta E} = \frac{T_E}{m_E}\cos\delta_E - \frac{v_{rE}v_{\theta E}}{r_E} \\ \dot{r}_E = v_{rE} \\ \dot{\xi}_E = -\frac{v_{\theta E}}{r_E} \end{cases} \end{aligned} \tag{37}$$

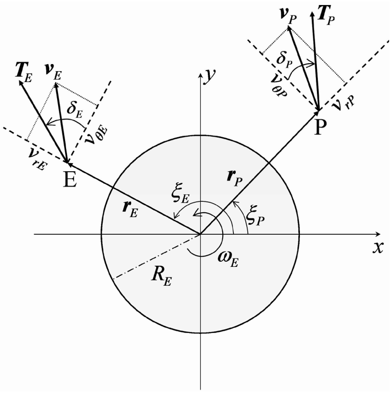


Fig. 2 Rotating frames of the two missiles.

with some initial conditions specified for the pursuer, and all the initial conditions specified for the evader:

$$r_P(t_0) = x_3(t_0) = x_{30}, \qquad \xi_P(t_0) = x_4(t_0) = x_{40} \tag{38}$$

$$\begin{aligned} v_{rE}(t_0) = x_5(t_0) = x_{50}, \qquad v_{\theta E}(t_0) = x_6(t_0) = x_{60} \\ r_E(t_0) = x_7(t_0) = x_{70}, \qquad \xi_E(t_0) = x_8(t_0) = x_{80} \end{aligned} \tag{39}$$

The initial position of the target is specified through its absolute longitude at t_0, ξ_{T0}. The velocity magnitude of the pursuer at t_0, V_{P0} is specified, and so the following constraint holds:

$$v_{rP}^2(t_0) + v_{\theta P}^2(t_0) = V_{P0}^2 \to x_{10}^2 + x_{20}^2 - V_{P0}^2 = 0 \tag{40}$$

The angles δ_P and δ_E are the control variables that define the thrust direction with respect to the local horizontal:

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_P^T & \mathbf{u}_E^T \end{bmatrix}^T = \begin{bmatrix} \delta_P & \delta_E \end{bmatrix}^T \tag{41}$$

The interception concludes the combat game and occurs when the distance between the two missiles is equal to a given capture radius d_{capt}:

$$\begin{aligned} r_E^2(t_f) + r_P^2(t_f) - 2r_E(t_f)r_P(t_f)\cos[\xi_E(t_f) - \xi_P(t_f)] = d_{\text{capt}}^2 \\ \to x_{7f}^2 + x_{3f}^2 - 2x_{7f}x_{3f}\cos(x_{8f} - x_{4f}) - d_{\text{capt}}^2 = 0 \end{aligned} \tag{42}$$

The boundary conditions consist of Eqs. (40) and (42), whose left-hand sides are included in the vector $\boldsymbol{\psi}$:

$$\boldsymbol{\psi} = \begin{bmatrix} x_{10}^2 + x_{20}^2 - V_{P0}^2 = 0 \\ x_{7f}^2 + x_{3f}^2 - 2x_{7f}x_{3f}\cos(x_{8f} - x_{4f}) - d_{\text{capt}}^2 \end{bmatrix} \tag{43}$$

Finally, the objective function is the opposite of the distance (squared) between the target and the ICBM at the interception:

$$\begin{aligned} J = \phi = -d_f^2 = -\Big\{ r_E^2(t_f) + R_E^2 - 2r_E(t_f)R_E\cos[\xi_E(t_f) \\ - \xi_{T0} - \omega_E(t_f - t_0)] \Big\} = -x_{7f}^2 - R_E^2 \\ + 2x_{7f}R_E\cos[x_{8f} - \xi_{T0} - \omega_E(t_f - t_0)] \end{aligned} \tag{44}$$

Hence, the value of the game is given by

$$V = \min_{\delta_P(t)} \max_{\delta_E(t)} J = \max_{\delta_E(t)} \min_{\delta_P(t)} J \tag{45}$$

After introducing the Hamiltonian H and the function Φ, defined by Eqs. (7) and (8), the following adjoint equations are derived:

$$\dot{\lambda}_1 = -\lambda_3 + \frac{\lambda_2 v_{\theta P}}{r_P} \tag{46}$$

$$\dot{\lambda}_2 = \frac{-2v_{\theta P}\lambda_1 + v_{rP}\lambda_2 - \lambda_4}{r_P} \tag{47}$$

$$\dot{\lambda}_3 = -\frac{2\mu_E\lambda_1}{r_P^3} + \frac{v_{\theta P}^2\lambda_1 - v_{rP}v_{\theta P}\lambda_2 + v_{\theta P}\lambda_4}{r_P^2} \tag{48}$$

$$\dot{\lambda}_4 = 0 \Rightarrow \lambda_4(t) = \lambda_4 = \text{const} \tag{49}$$

$$\dot{\lambda}_5 = -\lambda_7 + \frac{\lambda_6 v_{\theta E}}{r_E} \tag{50}$$

$$\dot{\lambda}_6 = \frac{-2v_{\theta E}\lambda_5 + v_{rE}\lambda_6 + \lambda_8}{r_E} \tag{51}$$

$$\dot{\lambda}_7 = -\frac{2\mu_E\lambda_5}{r_E^3} + \frac{v_{\theta E}^2\lambda_5 - v_{rE}v_{\theta E}\lambda_6 - v_{\theta E}\lambda_8}{r_E^2} \tag{52}$$

$$\dot{\lambda}_8 = 0 \Rightarrow \lambda_8(t) = \lambda_8 = \text{const} \tag{53}$$

The corresponding boundary conditions in Eqs. (11–14) are

$$\lambda_1(t_0) = 2\upsilon_1 v_{rP}(t_0) \tag{54}$$

$$\lambda_2(t_0) = 2\upsilon_1 v_{\theta P}(t_0) \tag{55}$$

$$\lambda_1(t_f) = \lambda_2(t_f) = 0 \tag{56}$$

$$\lambda_3(t_f) = 2\upsilon_2\{r_P(t_f) - r_E(t_f)\cos[\xi_E(t_f) - \xi_P(t_f)]\} \tag{57}$$

$$\lambda_4(t_f) = -2\upsilon_2 r_P(t_f) r_E(t_f)\sin[\xi_E(t_f) - \xi_P(t_f)] \tag{58}$$

$$\lambda_5(t_f) = \lambda_6(t_f) = 0 \tag{59}$$

$$\begin{aligned}\lambda_7(t_f) = {} & -2r_E(t_f) + 2R_E\cos[\xi_E(t_f) - \xi_{T0} - \omega_E(t_f - t_0)] \\ & + 2\upsilon_2\{r_E(t_f) - r_P(t_f)\cos[\xi_E(t_f) - \xi_P(t_f)]\}\end{aligned} \tag{60}$$

$$\begin{aligned}\lambda_8(t_f) = {} & -2r_E(t_f)R_E\sin[\xi_E(t_f) - \xi_{T0} - \omega_E(t_f - t_0)] \\ & + 2\upsilon_2 r_E(t_f) r_P(t_f)\sin[\xi_E(t_f) - \xi_P(t_f)]\end{aligned} \tag{61}$$

The control variables δ_P and δ_E are expressed as functions of λ_1, λ_2, λ_5, and λ_6 through Eqs. (17) and (18):

$$\begin{aligned}\frac{\partial H}{\partial \mathbf{u}_P} & = \frac{T_P}{m_P}(\lambda_1\cos\delta_P - \lambda_2\sin\delta_P) = 0 \to \delta_P^{(1)} \\ & = \arctan\left(\frac{\lambda_1}{\lambda_2}\right) \quad \text{and} \quad \delta_P^{(2)} = \arctan\left(\frac{\lambda_1}{\lambda_2}\right) + \pi\end{aligned} \tag{62}$$

$$\begin{aligned}\frac{\partial H}{\partial \mathbf{u}_E} & = \frac{T_E}{m_E}(\lambda_5\cos\delta_E - \lambda_6\sin\delta_E) = 0 \to \delta_E^{(1)} \\ & = \arctan\left(\frac{\lambda_5}{\lambda_6}\right) \quad \text{and} \quad \delta_E^{(2)} = \arctan\left(\frac{\lambda_5}{\lambda_6}\right) + \pi\end{aligned} \tag{63}$$

The selection of one of the two solutions, denoted with the superscripts (1) and (2), is made taking into consideration the respective second-order conditions in Eqs. (19) and (20):

$$\frac{\partial^2 H}{\partial \mathbf{u}_P^2} = \frac{T_P}{m_P}(-\lambda_1\cos\delta_P - \lambda_2\sin\delta_P) \geq 0 \tag{64}$$

$$\frac{\partial^2 H}{\partial \mathbf{u}_E^2} = \frac{T_E}{m_E}(-\lambda_5\cos\delta_E - \lambda_6\sin\delta_E) \leq 0 \tag{65}$$

Finally, as t_f is free, the transversality condition in Eq. (22) must hold, where

$$\frac{\partial \Phi}{\partial t_f} = 2r_P(t_f)R_E\omega_E\sin[\xi_E(t_f) - \xi_{T0} - \omega_E(t_f - t_0)] \tag{66}$$

Equations (37), (46–53), and (62–65), in conjunction with the boundary conditions in Eqs. (38–40), (42), and (54–61), and with the transversality condition in Eq. (22), define the TPBVP for the combat game at hand.

The following inequality, derived from the UP condition in Eq. (23), may be used as a final check on the numerical trajectories generated by the NLP solver:

$$\begin{aligned}& v_{rP}(t_f)r_P(t_f) + v_{rE}(t_f)r_E(t_f) - r_E(t_f)[v_{\theta P}(t_f)\sin\vartheta \\ & \quad + v_{rP}(t_f)\cos\vartheta] - r_P(t_f)[v_{\theta E}(t_f)\sin\vartheta + v_{rE}(t_f)\cos\vartheta] \leq 0\end{aligned} \tag{67}$$

where $\vartheta = \xi_E(t_f) - \xi_P(t_f)$. This inequality can be shown to yield a straightforward physical interpretation of the UP condition. First, with reference to Fig. 3, the two terms included in the square brackets may be rewritten as dot products:

$$r_E(t_f)[v_{\theta P}(t_f)\sin\vartheta + v_{rP}(t_f)\cos\vartheta] = \mathbf{v}_P^T\mathbf{r}_{Ef} \tag{68}$$

$$r_P(t_f)[v_{\theta E}(t_f)\sin\vartheta + v_{rE}(t_f)\cos\vartheta] = \mathbf{v}_P^T\mathbf{r}_{Ef} \tag{69}$$

Also, the first two terms of Eq. (67) can be handled as follows:

$$r_P(t_f)v_{rP}(t_f) = \mathbf{v}_P^T\mathbf{r}_{Ef} \quad \text{and} \quad r_E(t_f)v_{rE}(t_f) = \mathbf{v}_P^T\mathbf{r}_{Ef} \tag{70}$$

Thus, the UP condition in Eq. (67) becomes

$$\begin{aligned}& (\mathbf{v}_{Ef} - \mathbf{v}_{Pf})^T(\mathbf{r}_{Ef} - \mathbf{r}_{Pf}) \leq 0 \to \left(\frac{\mathrm{d}\boldsymbol{\rho}}{\mathrm{d}t}\right)_{t=t_f}^T \boldsymbol{\rho}(t_f) \leq 0 \\ & \quad \text{where } \boldsymbol{\rho}(t) = \mathbf{r}_E(t) - \mathbf{r}_P(t)\end{aligned} \tag{71}$$

That is, the UP condition is equivalent to the requirement that the two missiles are not separating at $t = t_f$, that is, $\dot{\rho}(t_f) = (d|\boldsymbol{\rho}|/\mathrm{d}t)_{t=t_f} \leq 0$.

A. Genetic Algorithm Preprocessing Settings

The GA preprocessing is aimed at providing a reasonable guess for the semi-DCNLP algorithm.

First of all, some boundary conditions can be handled to introduce the fewest number of parameters in the preprocessing. The relationship in Eq. (40) is used to express $v_{\theta P}(t_0)$ as a function of $v_{rP}(t_0)$:

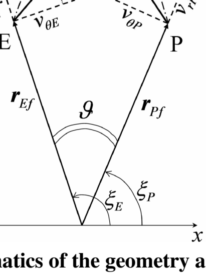


Fig. 3 Schematics of the geometry at the capture.

$$v_{\theta P}(t_0) = \sqrt{V_{P0}^2 - v_{rP}^2(t_0)} \tag{72}$$

Similarly, the two conditions in Eqs. (54) and (55) are combined by eliminating υ_1 to write $\lambda_2(t_0)$ as a function of $v_{rP}(t_0)$ and $\lambda_1(t_0)$:

$$\lambda_2(t_0) = v_{\theta P}(t_0)\frac{\lambda_1(t_0)}{v_{rP}(t_0)} = \sqrt{V_{P0}^2 - v_{rP}^2(t_0)}\,\frac{\lambda_1(t_0)}{v_{rP}(t_0)} \tag{73}$$

Then the three relationships in Eqs. (57), (58), and (61) are combined by eliminating υ_2:

$$\lambda_3(t_f)r_P(t_f)r_E(t_f)\sin[\xi_E(t_f) - \xi_P(t_f)] + \lambda_4\{r_E(t_f) - r_P(t_f)\cos[\xi_E(t_f) - \xi_P(t_f)]\} = 0 \tag{74}$$

$$\lambda_4 + \lambda_8 + 2R_E r_E(t_f)\sin[\xi_E(t_f) - \xi_{T0} - \omega_E(t_f - t_0)] = 0 \tag{75}$$

and, analogously, after combining Eqs. (60) and (61):

$$\begin{aligned}&\{\lambda_7(t_f) + 2r_E(t_f) - 2R_E\cos[\xi_E(t_f) - \xi_{T0}\\ &- \omega_E(t_f - t_0)]\}r_E(t_f)r_P(t_f)\sin[\xi_E(t_f) - \xi_P(t_f)]\\ &- \{\lambda_8 + 2r_E(t_f)R_E\sin[\xi_E(t_f) - \xi_{T0} - \omega_E(t_f - t_0)]\}\{r_E(t_f)\\ &- r_P(t_f)\cos[\xi_E(t_f) - \xi_P(t_f)]\} = 0\end{aligned} \tag{76}$$

Because of Eqs. (49) and (53), the two constant Lagrange multipliers λ_4 and λ_8 are considered as parameters, and Eqs. (49) and (53) become unnecessary. Hence, each individual is constituted by the following set of parameters Λ:

$$\Lambda = \{v_{rP}(t_0), \lambda_1(t_0), \lambda_3(t_0), \lambda_4, \lambda_5(t_0), \lambda_6(t_0), \lambda_7(t_0), \lambda_8, t_f\} \tag{77}$$

For each individual, the GA program integrates the differential equations (37), (46–48), and (50–52) assuming for the control variables δ_P and δ_E the values given by (62–65). Then, the constraints in Eqs. (22), (42), (56), (59), and (74–76) are evaluated. As stated in the preceding section, the number of parameters is exactly equal to the number of constraints, and the GA selects the best individual, that is, the set of parameters that minimizes Eq. (34), which is a nonnegative function of the constraint violations only.

For the problem at hand, the GA code by Deb et al. [25,26] was employed because it has proved to be quite robust in solving problems with a relevant number of constraints. We used the following basic settings: a population composed of 2000 individuals and 500 generations. The results of the GA preprocessing are reported in Sec. VI.

B. Semidirect Collocation with Nonlinear Programming Settings

The method of solution based on the semi-DCNLP algorithm was described in Sec. IV. For the problem at hand, the control of the evader (i.e., the ICBM warhead) is expressed as a function of its states and adjoint variables through Eqs. (63) and (65). In contrast, the control of the pursuer (i.e., the interceptor) is found numerically by the NLP problem solver, that is, the Fortran package NPSOL [21].

Hence, the time-varying adjoint variables of the evader are included as components of the extended state vector $\tilde{\mathbf{x}}$:

$$\tilde{\mathbf{x}} = \begin{bmatrix} v_{rP} & v_{\theta P} & r_P & \xi_P & v_{rE} & v_{\theta E} & r_E & \xi_E & \lambda_5 & \lambda_6 & \lambda_7 \end{bmatrix}^T \tag{78}$$

whereas the multiplier λ_8 is treated as a parameter because it is constant. In this context, the control vector $\tilde{\mathbf{u}}$ includes δ_P only. Moreover, the right-hand sides of Eqs. (37) and (50–52) form $\tilde{\mathbf{f}}\ (=\dot{\tilde{\mathbf{x}}})$.

In accordance with the approach described in Sec. IV, the state is represented by $(2\tilde{n}N + \tilde{n})$ parameters, whereas the control is approximated by $(4\tilde{m}N + \tilde{m})$ parameters. For the problem at hand, there are two additional parameters, the adjoint variable λ_8 and the terminal time t_f. In this study, the time interval is partitioned into ten subarcs $(N = 10)$. Moreover, $\tilde{n} = 11$ and $\tilde{m} = 1$, and so the total number of parameters involved is $(2\tilde{n}N + \tilde{n} + 4\tilde{m}N + \tilde{m} + 2) = 274$.

In addition, the NLP solver has to satisfy the constraints deriving from the $2\tilde{n}N$ collocation conditions in Eqs. (31) and (32), and the boundary conditions in Eqs. (38–40), (42), (59), and (76). [Equations (22) and (73–75) are not considered because the adjoint variables of the pursuer are not involved in the semi-DCNLP algorithm.] Hence, the total number of scalar constraints is $(2\tilde{n}N + 11) = 231$.

Two fundamental settings concern the numerical accuracy specified for the NLP solver: both the optimality tolerance and the constraint violation tolerance were set to 10^{-8}.

VI. Numerical Results

In this research, for the numerical solution canonical units were used. These are a normalized set of units. The Earth radius R_E is assumed as the distance unit (DU) and the time unit (TU) is such that the Earth planetary constant μ_E is equal to 1 $\mathrm{DU}^3/\mathrm{TU}^2$. Hence, 1 DU = 6378.165 km and 1 TU = 806.8 sec. In canonical units, the Earth rotation rate is $\omega_E = 5.8834 \times 10^{-2}\ \mathrm{TU}^{-1}$.

The inertial frame shown in Fig. 2 is defined by two axes included in the equatorial plane. By definition, the x axis is oriented towards the perigee of the hypothetical (nominal) evader trajectory, which would lead it to impact on the target without performing any corrective maneuver. With reference to Fig. 4a, the g axis intersects the Greenwich meridian at $t = t_L (< t_0)$, and θ_{gL} is the Greenwich absolute longitude at t_L. The position of the target is identified by its geographic longitude μ_T.

The nominal (ballistic) trajectory of the ICBM is such that a completely ballistic trajectory would lead it to impact on the target in a given time interval Δt after launch. With reference to Fig. 4a, the angular separation φ between the starting point L and the terminal (impact) point I is given by

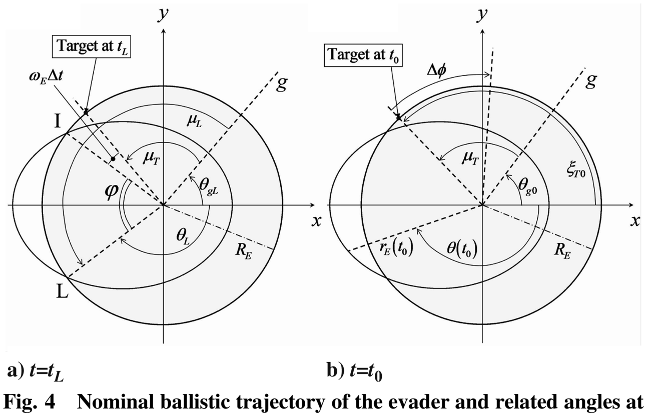
$$\varphi = \mu_L - (\mu_T + \omega_E\Delta t) \tag{79}$$

where μ_L and μ_T are the geographical longitudes of the ICBM launch site and of the target, respectively. As the (nominal) ballistic arc is symmetrical with respect to the apogee, the true anomalies at launch and at impact are

$$\theta_L = \pi - \frac{\varphi}{2} \quad \text{and} \quad \theta_I = \pi + \frac{\varphi}{2} \tag{80}$$

The following two relationships yield the semimajor axis a and the eccentricity e of the ICBM nominal trajectory:

$$\Delta t = \sqrt{\frac{a^3}{\mu_E}}[E_I - E_L - e(\sin E_I - \sin E_L)] \tag{81}$$



a) t=t_L b) t=t_0

Fig. 4 Nominal ballistic trajectory of the evader and related angles at a) t_L and at b) t_0.

$$R_E = \frac{a(1-e^2)}{1+e\cos\theta_I} \tag{82}$$

where E_L and E_I are the eccentric anomalies related to θ_L and θ_I.

The initial time t_0 corresponds to the detection of the attacking missile, which is assumed to occur when the ICBM is at an altitude of 0.01 DU (63.8 km). Hence, its true anomaly θ at t_0 is given by

$$\theta(t_0) = \arccos\left[\frac{a(1-e^2) - r_E(t_0)}{e r_E(t_0)}\right] \quad \text{where } r_E(t_0) = 1.01 \text{ DU} \tag{83}$$

After inspecting Fig. 4b, the initial absolute longitude can be derived from $\theta(t_0)$: $\xi_E(t_0) = 2\pi - \theta(t_0)$. Finally, the remaining evader state components at t_0, that is, the two components of velocity, are calculated as follows:

$$v_{rE}(t_0) = \frac{\sqrt{\mu_E}}{\sqrt{a(1-e^2)}} e\sin[\theta(t_0)] \quad \text{and}$$
$$v_{\theta E}(t_0) = \frac{\sqrt{\mu_E}}{\sqrt{a(1-e^2)}}\{1 + e\cos[\theta(t_0)]\} \tag{84}$$

The initial position of the target is identified by its absolute longitude at t_0: $\xi_{T0} = \theta_{g0} + \mu_T$, where $\theta_{g0} = \theta_{gL} + \omega_E(t_0 - t_L)$.

The following values are assumed for μ_L, μ_T, θ_{g0}, Δt, and for (T_E/m_E), which represents the thrust-to-mass ratio of the evader:

$$\mu_L = 229.1 \text{ deg}, \qquad \mu_T = 132.4 \text{ deg}, \qquad \theta_{g0} = 0 \text{ deg}$$
$$\Delta t = 2 \text{ TU} \qquad \frac{T_E}{m_E} = 0.15 \text{ DU/TU}^2 \tag{85}$$

The first four values yield the following initial conditions:

$$v_{rE}(t_0) = 2.025 \cdot 10^{-1} \text{ DU/TU}$$
$$v_{\theta E}(t_0) = 8.908 \cdot 10^{-1} \text{ DU/TU}, \qquad r_E(t_0) = 1.01 \text{ DU} \tag{86}$$
$$\xi_E(t_0) = 222.5 \text{ deg}$$

In this paper, optimal intercepting trajectories are found for several different positions of the pursuer launch site; also, different values of the pursuer thrust-to-mass ratio (T_P/m_P) are considered. The interception, which was assumed as the termination of the game, occurs in every simulation that was made if the pursuer has the following:

1) a thrust-over-mass ratio greater than (or at least equal to) that of the evader, or

2) an angular separation from the impact point less than that of the evader.

We suspect that this is unequivocally true. These two circumstances constitute evident sufficient conditions for the interception because the pursuer can also select its starting velocity direction, that is, its launch elevation, which represents an additional degree of freedom available to it. The conditions 1) and 2) are assumed for all the cases presented in this section.

The position of the pursuer at t_0 is identified by its angular displacement $\Delta\phi$ from the target location, as shown in Fig. 4b. The ICBM was assumed to be able to pick its velocity direction at the end of its vertical ascent phase, which occurs at a very low altitude (usually a few miles in real situations). For the sake of simplicity, it is now supposed that the pursuer chooses its velocity direction immediately after launch, that is, just above the Earth's surface. Hence, its initial radius is set equal to 1 DU: $r_P(t_0) = 1$ DU. In addition, the pursuer launch capabilities are assumed to be analogous to those of the evader, that is, after launch, the osculating trajectory of the pursuer is supposed to have the same semimajor axis a as that of the evader nominal trajectory. Hence, the magnitude of the pursuer velocity at t_0, V_{P0} is simply given by

$$V_{P0} = \sqrt{\frac{2\mu_E}{R_E} - \frac{\mu_E}{a}} \tag{87}$$

The interceptor absolute longitude at t_0 is related to ξ_{T0} and to $\Delta\phi$ as follows:

$$\xi_P(t_0) = \xi_{T0} - \Delta\phi \tag{88}$$

where $\xi_{T0} = \mu_T$ as $\theta_{g0} = 0$ deg was assumed. The evaluation of Eq. (87) yields $V_{P0} = 9.243 \times 10^{-1}$ DU/TU. As the initial time t_0 is arbitrary, it is set to 0. Finally, the capture radius is set equal to 1.568×10^{-5} DU (100 m).

As an example, the GA preprocessing is applied to generate the numerical guess for the case $\Delta\phi = 0$ deg and $(T_P/m_P) = 0.15$ DU/TU^2. The search space is defined by the parameter ranges, which are set as follows:

1) The initial radial velocity of the pursuer must be included in the range $[0, V_{P0}]$.

2) All the Lagrange multipliers are searched in the range $[-5, 5]$.

3) The terminal time is searched in the range $[0, 2]$.

The following parameters have been obtained by the use of the GA:

$$\Lambda = \{1.300 \times 10^{-1}, -2.250 \times 10^{-1},$$
$$-4.891 \times 10^{-1}, -1.358, 5.764 \times 10^{-1}, -2.401 \times 10^{-1},$$
$$8.446 \times 10^{-1}, -2.443 \times 10^{-1}, 9.813 \times 10^{-1}\} \tag{89}$$

(When omitted, the variables are numerically written with reference to their usual units.) These parameters yield the control time histories illustrated in Fig. 5.

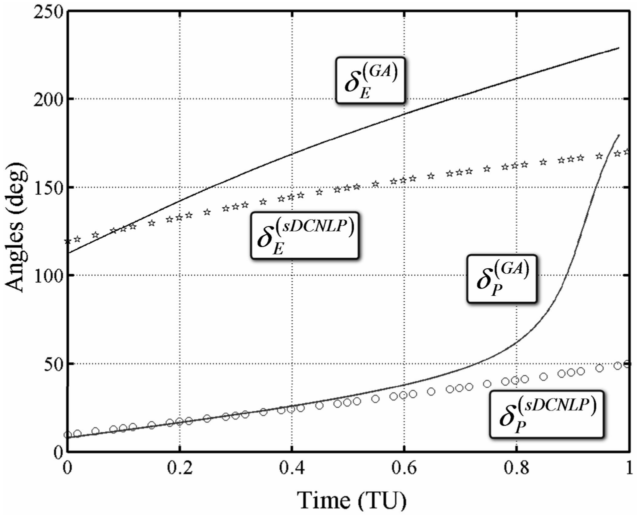


Fig. 5 Control time histories (guesses produced by the GA and optimal solutions generated by the semi-DCNLP algorithm). Case $\Delta\phi = 0$ deg and $(T_P/m_P) = 0.15$ DU/TU^2.

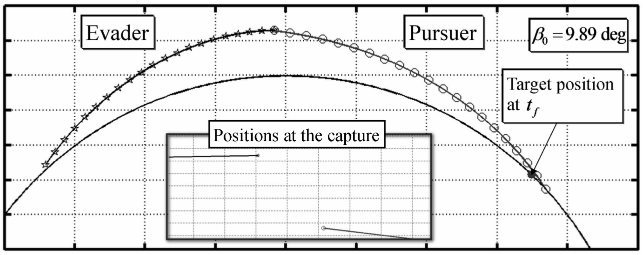


Fig. 6 Saddle-point trajectories [$\Delta\phi = 0$ deg and $(T_P/m_P) = 0.15$ DU/TU^2].

Table 1 Distance d_f (DU) between the target and the ICBM at the interception, for different values of $\Delta\phi$ and (T_p/m_p). $((T_E/m_E) = 0.15\ \mathrm{DU/TU^2}$ for all cases)

T_p/m_p, DU/TU^2		0.15	0.20	0.30	0.40	0.50
$\Delta\phi$, deg	0	0.8404	0.8535	0.8764	0.8958	0.9127
	5	0.8051	0.8199	0.8457	0.8675	0.8862
	10	0.7701	0.7867	0.8156	0.8397	0.8603
	15	0.7356	0.7541	0.7860	0.8126	0.8347

Using the approximate solution achieved through the GA preprocessing as an initial guess, the saddle-point solution for the problem at hand was obtained through the semi-DCNLP algorithm. The corresponding control time histories and saddle-point trajectories are shown in Figs. 5 and 6, which, in addition, illustrates the interception geometry in the inset and the optimal initial flight path angle of the pursuer $\beta_0\{= \arctan[v_{rP}(t_0)/v_{\theta P}(t_0)]\}$.

Other numerical solutions were derived for different values of $\Delta\phi$ and (T_P/m_P). The numerical solution for the case $\Delta\phi = 0$ deg and $(T_P/m_P) = 0.2\ \mathrm{DU/TU^2}$ was obtained by using the optimal solution of the first case as a guess. This method is commonly

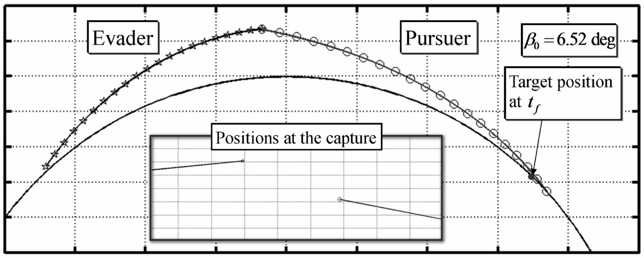


Fig. 7 Saddle-point trajectories [$\Delta\phi = 0$ deg and (T_P/m_P)= 0.3 DU/TU^2]

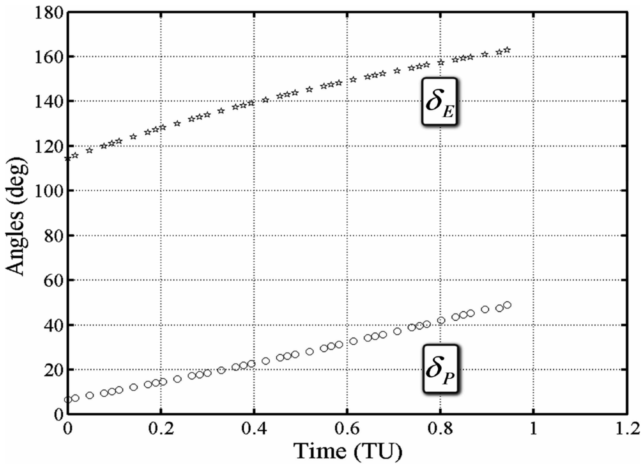


Fig. 8 Optimal control time histories [$\Delta\phi = 0$ deg and (T_P/m_P)= 0.3 DU/TU^2].

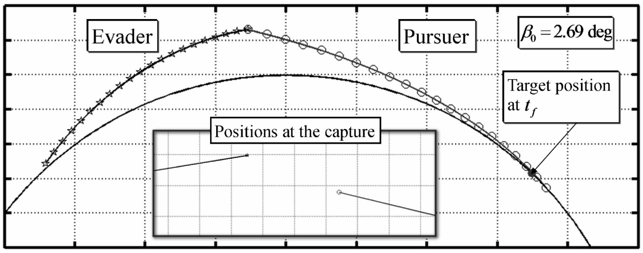


Fig. 9 Saddle-point trajectories [$\Delta\phi = 0$ deg and (T_P/m_P)= 0.5 DU/TU^2].

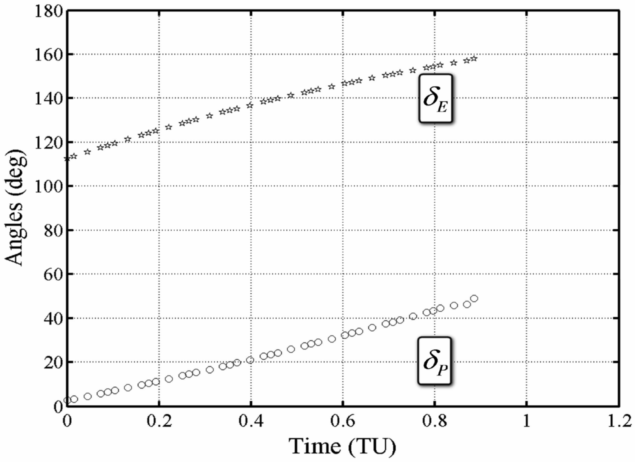


Fig. 10 Optimal control time histories [$\Delta\phi = 0$ deg and $(T_P/m_P) = 0.5$ DU/TU^2].

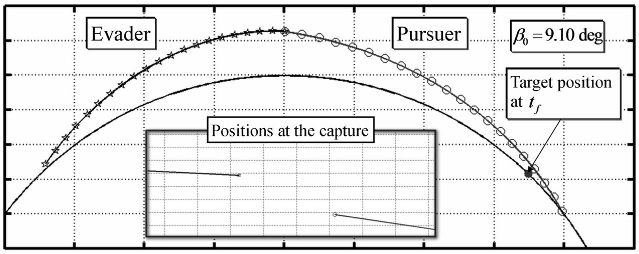


Fig. 11 Saddle-point trajectories [$\Delta\phi = 5$ deg and $(T_P/m_P) = 0.15$ DU/TU^2].

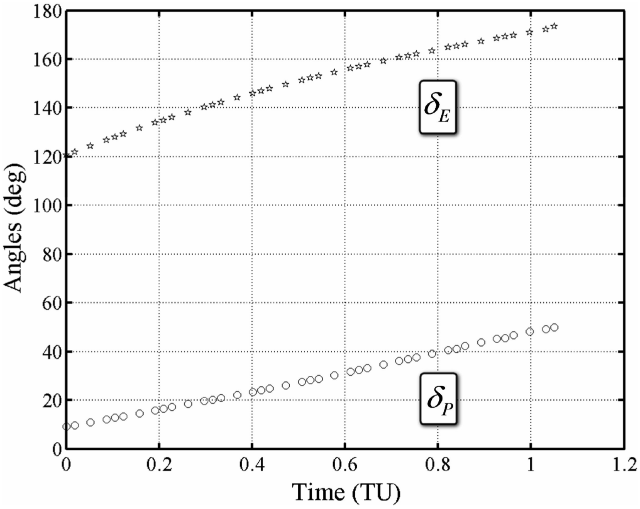


Fig. 12 Optimal control time histories [$\Delta\phi = 5$ deg and $(T_P/m_P) = 0.15$ DU/TU^2].

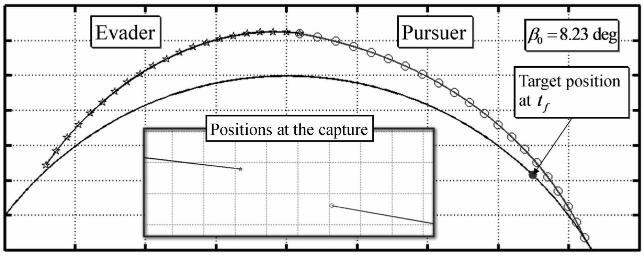


Fig. 13 Saddle-point trajectories [$\Delta\phi = 10$ deg and $(T_P/m_P) = 0.15$ DU/TU^2].

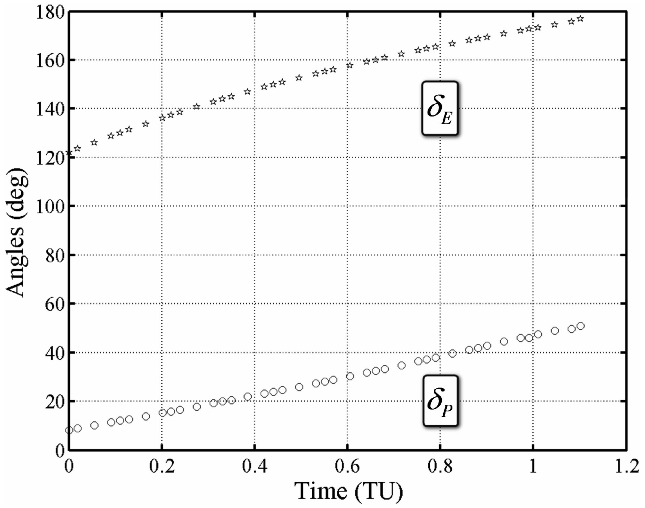


Fig. 14 Optimal control time histories [$\Delta\phi = 10$ deg and $(T_P/m_P) = 0.15$ DU/TU^2].

referred as the homotopy approach and was employed to derive all the remaining solutions once the first one was found.

For all the cases, the distance d_f between the ICBM and the target at the interception is reported in Table 1. As expected, this distance decreases as $\Delta\phi$ increases and increases as (T_P/m_P) increases.

Some of the saddle-point trajectories we obtained and the related control variables are illustrated in Figs. 7–14. After inspection, it emerges that all the control time histories are near-linear functions of the time. Another common detail of all the solutions concerns the geometry of the capture, which correctly occurs while the two missiles are still approaching. Hence, the UP condition is satisfied by all the solutions reported in Table 1.

Note that solutions were obtained that satisfied the necessary conditions but did not terminate on the usable part of the terminal surface. These solutions were found with poor initial guesses obtained by intuition and not via the GA preprocessing and must be rejected as saddle-point trajectories.

The homotopy method can also be employed to generate saddle-point trajectories when the defending missile has a lower thrust-to-mass ratio than that of the ICBM. The semi-DCNLP algorithm is able to find solutions leading to the interception even in the case when the pursuer has no thrust. This kind of solution corresponds to a sort of pathological saddle-point trajectory, in which the pursuer is able to intercept the evader by simply selecting a suitable (optimal) direction of its initial velocity. This result is achieved by the defending missile under the assumption that the evader will play optimally. Such an outcome of the semi-DCNLP algorithm is not in contradiction with the mathematical formulation of the problem because the method is aimed at generating open-loop representations of feedback saddle-point strategies, which are functions of the initial states only. So, if the ICBM should adopt a preprogrammed strategy, that is, a strategy not modifiable after launch, the optimal choice for this strategy would correspond to that which yields the pathological saddle-point trajectory. This means that the defending missile would be able to intercept the ICBM warhead by simply picking an optimal velocity direction, provided that the ICBM will not deviate from its optimal trajectory. Such pathological saddle-point solutions yield the paradox that the ICBM could avoid the interception by simply employing a mixed strategy. An attacking missile that employs a prefixed and not modifiable thrust direction programming does not present any challenge to the interceptor, and so pathological saddle-point solutions have no usefulness, even though they suggest likely interesting considerations on the semi-DCNLP method capabilities and limits.

With reference to the theoretical foundations of differential games, these limits are basically related to the difficulty encountered by numerical methods in the characterization of the barrier, which separates the capture zone from the evasion zone. For the same differential game, these two zones correspond to different terminal conditions, which yield different sets of equations describing the game in each zone. However, in this paper, the capture was assumed as the termination, and the sufficient conditions 1) and 2) ensuring the interception were stated.

VII. Conclusions

This paper presents some numerical optimal trajectories for the interception of evasive intercontinental ballistic missile warheads. The problem is formulated as a zero-sum differential game, in which the pursuer P (the defending missile) tries to catch the evader E (the ICBM warhead). The latter must hit a location on the Earth's surface and tries to get as close as possible to the target. In contrast, the defending missile tries to keep the ICBM as far as possible from the target. Each missile is given a modest postlaunch capability to maneuver.

Saddle-point equilibrium solutions are found using a recently developed, direct numerical method that nevertheless uses the analytical necessary conditions to find the optimal control for one of the players (the evader). The inclusion of the adjoint equations for the evader in the method of solution has the unfavorable consequence that the starting guesses for the nonintuitive adjoint variables of the evader are needed. This difficulty was resolved by employing a systematic approach based on the use of genetic algorithms, which do not require any a priori information about the solution. It was found that the relative inaccuracy of the solution obtained with a GA is not a limitation when the GA is used as a preprocessing method, that is, when the GA is only required to provide a reasonable guess for the nonlinear programming problem solver used in the direct method.

In differential games of pursuit evasion, a singular surface called the barrier divides the state space into a capture zone and an evasion zone. In the capture zone, if P plays optimally, then he can catch E regardless of E's strategy; in the evasion zone, if E plays optimally, then he can avoid the capture altogether. In this study, sufficient conditions ensuring the interception are provided. Moreover, due to the semipermeability of the barrier, a saddle-point trajectory is expected to terminate on the usable part of the terminal surface, provided that the initial state is inside the capture zone. For the problem at hand, the usable part condition, formerly introduced by Isaacs, is considered and is shown to have an interesting physical interpretation, which can be simply stated: saddle-point trajectories lead to capture while the two missiles are still approaching each other.

Once one solution was obtained using the GA-derived guess, saddle-point trajectories for cases with different starting positions and different values of the pursuer thrust-to-mass ratio could be obtained by using a homotopy approach. In all the cases, the control time histories are near-linear functions of the time. Another common detail of the saddle-point trajectories concerns the geometry of the capture, which correctly occurs while the two missiles are still approaching. Hence, the UP condition is satisfied by all the solutions.

A number of simplifying approximations are made in this work; for example, the analysis is two dimensional and both missile trajectories lie in the equatorial plane, atmospheric forces are not modeled, etc. However, we believe the work proves the concept of formulating and solving the BMD problem as a dynamic game problem. The extension of the analysis to 3-D and the inclusion of a more realistic dynamic model should not be very challenging now that the framework, including a successful numerical solution algorithm, exists.

With reference to the theoretical foundations of differential games, a common limit of all the numerical methods is basically related to the difficulty encountered in the characterization of the barrier, which separates the capture zone from the evasion zone. For the same differential game, these two zones correspond to different terminal conditions, which yield different sets of equations describing the game in each zone. Additional investigations on the relationships between these two distinct two-point boundary-value problems (related to the same differential game) seem extremely interesting.

As a matter of fact, such investigations could disclose relevant information on the nature of the barrier, even in complex differential game contexts.

References

[1] Isaacs, R., *Games of Pursuit*, The Rand Corporation, Santa Monica, CA, 1951, p. 257.

[2] Isaacs, R., *Differential Games*, Wiley, New York, 1965, pp. 278–280.

[3] Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Hemisphere, New York, 1975, pp. 271–293.

[4] Basar, T., and Olsder, G. J., *Dynamic Noncooperative Game Theory*, SIAM, Philadelphia, PA, 1999, pp. 424–441.

[5] Breakwell, J. V., and Merz, A. W., "Minimum Required Capture Radius in a Coplanar Model of the Aerial Combat Problem," *AIAA Journal*, Vol. 15, No. 8, 1977, pp. 1089–1094.

[6] Guelman, M., Shinar, J., and Green, A., "Qualitative Study of a Planar Pursuit Evasion Game in the Atmosphere," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 6, 1990, pp. 1136–1142.

[7] Roberts, D. A., and Montgomery, R. C., "Development and Application of a Gradient Method for Solving Differential Games," *Langley Research Center*, NASA TN D-6502, Washington, D. C., Nov. 1971.

[8] Hillberg, C., and Järmark, B., "Pursuit-Evasion Between Two Realistic Aircraft," AIAA Paper 83-2119, 1983.

[9] Järmark, B. S. A., Merz, A. W., and Breakwell, J. V., "The Variable Speed Tail-Chase Aerial Combat Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 4, No. 3, 1981, pp. 323–328.

[10] Breitner, M. H., Pesch, H. J., and Grimm, W., "Complex Differential Games of Pursuit-Evasion Type with State Constraints, Part 1: Necessary Conditions for Open-Loop Strategies," *Journal of Optimization Theory and Applications*, Vol. 78, No. 3, 1993, pp. 419–441.
doi:10.1007/BF00939876

[11] Breitner, M. H., Pesch, H. J., and Grimm, W., "Complex Differential Games of Pursuit-Evasion Type with State Constraints, Part 2: Numerical Computation of Open-Loop Strategies," *Journal of Optimization Theory and Applications*, Vol. 78, No. 3, 1993, pp. 443–463.
doi:10.1007/BF00939877

[12] Raivio, T., and Ehtamo, H., "Visual Aircraft Identification as a Pursuit-Evasion Game," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 4, 2000, pp. 701–708.

[13] Shima, T., and Shinar, J., "Time-Varying Linear Pursuit-Evasion Game Models with Bounded Controls," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 3, 2002, pp. 425–432.

[14] Horie, K., and Conway, B. A., "Optimal Fighter Pursuit-Evasion Maneuvers Found via Two-Sided Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 1, 2006, pp. 105–112.
doi:10.2514/1.3960

[15] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342.

[16] Enright, P. J., and Conway, B. A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994–1002.

[17] Herman, A. L., and Conway, B. A., "Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599.

[18] Herman, A. L., and Conway, B. A., "Optimal Low-Thrust, Earth-Moon Orbit Transfer," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 1, 1998, pp. 141–147.

[19] Conway, B. A., "Optimal Low-Thrust Interception of Earth-Crossing Asteroids," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, 1997, pp. 995–1002.

[20] Tang, S., and Conway, B. A., "Optimization of Low-Thrust Interplanetary Trajectories Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 3, 1995, pp. 599–604.

[21] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., "User's Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming," Stanford Optimization Laboratory Publ. 2, Stanford University, Palo Alto, CA, 1986.

[22] Horie, K., and Conway, B. A., "Genetic Algorithm Pre-Processing for Numerical Solution of Differential Games Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, 2004, pp. 1075–1078.
doi:10.2514/1.3361

[23] Horie, K., "Collocation with Nonlinear Programming for Two-Sided Flight Path Optimization," Ph.D. Thesis, University of Illinois at Urbana–Champaign, Champaign, IL, 2002.

[24] Papavassilopoulos, G. P., and Cruz, J. B., Jr., "Nonclassical Control Problems and Stackelberg Games," *IEEE Transactions on Automatic Control*, Vol. AC-24, No. 2, 1979, pp. 155–166.
doi:10.1109/TAC.1979.1101986

[25] Deb, K., Mohan, M., and Mishra, S., "A Fast Multi-Objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions," Kanpur Genetic Algorithms Laboratory, Rept. 2003002, 2003.

[26] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T., "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," Kanpur Genetic Algorithms Laboratory, Rept. 2000001, 2000.