Journal of Guidance, Control, and Dynamics
Vol. 32, No. 2, March–April 2009

Numerical Solution of the Three-Dimensional Orbital Pursuit–Evasion Game

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

DOI: 10.2514/1.37962

The problem of interception of an evasive spacecraft by a pursuing spacecraft is formulated as a differential game. Each spacecraft is given a modest capability to maneuver in the three-dimensional space. Interception concludes the game and occurs if the pursuing spacecraft reaches the instantaneous position of the evading spacecraft. The objective of the pursuer is to minimize the time for interception, whereas the evader tries to delay it indefinitely. Saddle-point equilibrium solutions are found using a recently developed direct numerical method that uses the analytical necessary conditions (unlike ordinary direct methods) 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. The evolutionary algorithm is employed as a preprocessing technique and is very useful in this context because the trial-and-error selection of first-attempt values for the variables involved is very challenging for the problem at hand. The numerical method is tested on a variety of different starting conditions related to the distinct initial orbits of the two spacecraft. It successfully finds the saddle-point trajectories of the two spacecraft, thus proving its effectiveness and robustness in solving a quite complicated problem such as the three-dimensional orbital game at hand.

Nomenclature

a_i = semimajor axis of spacecraft i, $i = P$ or E
c_j = constraint in the genetic algorithm preprocessing; $j = 1, \ldots, n_C$
e_i = eccentricity of spacecraft i, $i = P$ or E
f_i = true anomaly of spacecraft i, $i = P$ or E
H = Hamiltonian
H_f = Hamiltonian at t_f
H_0 = Hamiltonian at t_0
h = altitude
$\tilde{h}_i$ = angular momentum magnitude of spacecraft i, $i = P$ or E
$\hat{h}_i$ = unit vector associated to the angular momentum $\boldsymbol{h}_i = \boldsymbol{r}_i \times \boldsymbol{v}_i$ of spacecraft i, $i = P$ or E
$h_{\min}$ = minimum allowed altitude
i = spacecraft index, $i = P$ or E
i_i = inclination of spacecraft i, $i = P$ or E
J = objective function
$\bar{J}$ = objective function used by the genetic algorithm
M_i = mean anomaly of spacecraft i, $i = P$ or E
m_E = mass of the evading spacecraft
m_P = mass of the pursuing spacecraft
N = number of subarcs
R_E = Earth radius
r_i = radius of spacecraft i, $i = P$ or E
$\boldsymbol{r}_i$ = position vector of spacecraft i, $i = P$ or E
$\hat{r}_i$ = unit vector associated to the position vector $\boldsymbol{r}_i$ of spacecraft i, $i = P$ or E
T_E = thrust of the evading spacecraft
T_P = thrust of the pursuing spacecraft
t = time
t_f = terminal time
t_0 = initial time
$\tilde{\boldsymbol{u}}$ = extended control ($\tilde{m}$-dimensional column vector)
$\boldsymbol{u}_E$ = evader control ($\hat{m}_E$-dimensional column vector)
$\boldsymbol{u}_P$ = pursuer control ($\hat{m}_P$-dimensional column vector)
v_i = velocity magnitude of spacecraft i, $i = P$ or E
$\boldsymbol{v}_i$ = velocity vector of spacecraft i, $i = P$ or E
$\tilde{\boldsymbol{x}}$ = extended state ($\tilde{n}$-dimensional column vector)
$\boldsymbol{x}_E$ = evader state (n_E-dimensional column vector)
$\boldsymbol{x}_{Ef}$ = evader state at t_f (n_E-dimensional column vector)
$\boldsymbol{x}_{E0}$ = evader state at t_0 (n_E-dimensional column vector)
x_{Ek} = kth component of the evader state
$\boldsymbol{x}_P$ = pursuer state (n_P-dimensional column vector)
$\boldsymbol{x}_{Pf}$ = pursuer state at t_f (n_P-dimensional column vector)
x_{Pk} = kth component of the pursuer state
$\boldsymbol{x}_{P0}$ = pursuer state at t_0 (n_P-dimensional column vector)
α_i = in-plane control angle of spacecraft i, $i = P$ or E
β_i = out-of-plane control angle of spacecraft i, $i = P$ or E
γ_i = flight path angle of spacecraft i, $i = P$ or E
θ_i = argument of latitude of spacecraft i, $i = P$ or E
$\hat{\theta}_i$ = unit vector associated to the projection of $\boldsymbol{v}_i$ into the local horizontal plane, $i = P$ or E
$\boldsymbol{\lambda}_E$ = adjoint variable conjugate to the evader equations of motion (n_E-dimensional column vector)
$\boldsymbol{\lambda}_P$ = adjoint variable conjugate to the pursuer equations of motion (n_P-dimensional column vector)
μ_E = Earth planetary constant
ξ_i = absolute longitude of spacecraft i, $i = P$ or E
ζ_i = coazimuth angle of spacecraft i $i = P$ or E
$\boldsymbol{\upsilon}$ = adjoint variable conjugate to the boundary conditions (q-dimensional column vector)
ϕ = Mayer term of the objective function
ϕ_i = latitude of spacecraft i, $i = P$ or E
$\boldsymbol{\Psi}$ = function of boundary conditions (q-dimensional column vector)
Ω_i = right ascension of the ascending node of spacecraft i, $i = P$ or E
ω_i = argument of perigee of spacecraft i, $i = P$ or E

I. Introduction

IN GENERAL, a flight path optimization problem can be treated as a one-sided optimization problem (optimal control problem) or

Received 8 April 2008; revision received 9 October 2008; accepted for publication 12 October 2008.

*Ph.D.; currently Research Assistant, 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.

a two-sided optimization problem. The one-sided optimization problem considers only one player and has been successfully applied to a variety of aerospace problems for decades, for example, the optimal trajectories for aircraft or spacecraft. However, the problem considered in this work, the combat of an optimally evasive spacecraft against an optimally pursuing spacecraft, is best modeled using two competing players, that is, it becomes a two-sided optimization problem or a zero-sum two-player differential game. This kind of game was 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] did this as an extension of the analytical necessary conditions for the one-player game. Basar and Olsder [4] provided the necessary conditions for the existence of an open-loop representation of a saddle-point trajectory. A small number of air combat problems has 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 a numerical solution. However, only a limited number of studies have been done using this approach because the optimization of 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 a missile and an 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 orbital pursuit–evasion games, past studies are often based on simplified dynamical models. Anderson and Grazier [13] described the construction of a closed-form solution for the barrier in a planar pursuit–evasion game between two spacecraft by linearizing the problem about a reference circular orbit. Kelley et al. [14] derived the impulsive maneuvers available to two spacecraft involved in an orbital combat. They argued that optimal evasion only consists of in-plane maneuvers. More recently, Zhang et al. [15] examined the pursuit–evasion game between two coplanar, near-Earth orbiters under the assumption of radial, very low, continuous, and variable thrust for both spacecraft. Vasile and Bernelli Zazzera [16] proposed a multiphase distributed approach for the solution of dynamic games and applied it to a planar orbital pursuit–evasion game. Menon and Calise [17,18] developed an effective guidance strategy for two spacecraft involved in a three-dimensional pursuit–evasion game. By using feedback linearization and a quadratic objective function, established methods for the solution of linear quadratic problems are available and the optimal feedback laws may be found in closed form.

Horie and Conway [19] and Horie [20] introduced a new numerical solution method for pursuit–evasion dynamic games called semidirect. They applied this method to a planar orbital pursuit–evasion game [21] and to the challenging problem of the three-dimensional air combat between 2 F-16 fighters [19]. Their method is very similar to recent direct methods for the solution of one-sided optimal control problems that convert the continuous optimization problem into a nonlinear programming (NLP) problem [22–24]. This is the method of solution chosen for the current problem; it will be discussed at greater length in a subsequent section. An initial guess of the solution is needed by the method and can be provided by generating an approximate solution using genetic algorithms.

This research is concerned with a three-dimensional orbital pursuit–evasion game in which the time for interception is to be minimized by the pursuing spacecraft and maximized by the evading spacecraft. The three-dimensional motion is described by six nonlinear differential equations for each spacecraft, and only a few simplifying hypotheses are made. First of all, atmosphere is neglected, because the minimum allowed altitude is set to 100 km. In addition, a low, constant, thrust-to-mass ratio is assumed for both spacecraft. Lastly, the two spacecraft start maneuvering simultaneously, and each of them is supposed to possess complete and instantaneous information on the state of the opponent player. 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 evading spacecraft can provide the worst-case-scenario faced by the pursuer, which is very useful to know.

The objective of this work is thus to formulate the three-dimensional orbital combat as a dynamic game, derive the necessary analytical conditions that must be satisfied by the saddle-point trajectories as a partial verification of optimality, demonstrate the solution of the pursuit–evasion game via the joint use of a genetic algorithm and of the semidirect method, find the saddle-point solutions (and the related saddle-point trajectories) for a variety of cases, and to gauge the robustness of the numerical method for solving the quite complex dynamic game at hand.

II. Zero-Sum Differential Games

When two competitive actors (usually referred to as players) are involved in a flight-path optimization, the problem can be modeled as a zero-sum (or pursuit–evasion) differential game.

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

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

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

with $t_0 \le t \le t_f$. The initial and the terminal times, t_0 and t_f, can be either specified or free. The boundary conditions are collected in $\boldsymbol{\Psi}$, which is a vector function of the unknown starting and final values of the states and of the initial and terminal times:

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

Without any loss of generality, a Mayer-type objective function is assumed:

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

depending on the unknown initial and terminal values of the state and, possibly, on the times t_0 and t_f. The performance index J is to be minimized by P and maximized by E.

In zero-sum differential games, a pair of optimal strategies corresponds to a saddle-point (SP) 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 [19]. To state these conditions, a Hamiltonian H and a function of terminal conditions Φ are introduced as

$$H \triangleq \boldsymbol{\lambda}_P^T \boldsymbol{f}_P + \boldsymbol{\lambda}_E^T \boldsymbol{f}_E \tag{5}$$

$$\Phi \triangleq \phi + \boldsymbol{\upsilon}^T \boldsymbol{\Psi} \tag{6}$$

where $\boldsymbol{\lambda}_P(t)$, $\boldsymbol{\lambda}_E(t)$, and $\boldsymbol{\upsilon}$ are the adjoint variables conjugate to the state equations of the pursuer, to those of the evader, and to the boundary conditions (3), respectively.

The necessary conditions for an open-loop representation of an optimal saddle-point solution include the adjoint equations for the Lagrange multipliers $\boldsymbol{\lambda}_P$ and $\boldsymbol{\lambda}_E$:

$$\dot{\boldsymbol{\lambda}}_P = -\left[\frac{\partial H}{\partial \boldsymbol{x}_P}\right]^T = -\left[\frac{\partial \boldsymbol{f}_P}{\partial \boldsymbol{x}_P}\right]^T \boldsymbol{\lambda}_P \tag{7}$$

$$\dot{\boldsymbol{\lambda}}_E = -\left[\frac{\partial H}{\partial \boldsymbol{x}_E}\right]^T = -\left[\frac{\partial \boldsymbol{f}_E}{\partial \boldsymbol{x}_E}\right]^T \boldsymbol{\lambda}_E \tag{8}$$

in conjunction with the respective boundary conditions:

$$\lambda_{Pk}(t_0) + \frac{\partial \Phi}{\partial x_{Pk}(t_0)} = 0 \quad \text{if } x_{Pk}(t_0) \text{ is unspecified } (k = 1, \ldots, n_P) \tag{9}$$

$$\lambda_{Pk}(t_f) - \frac{\partial \Phi}{\partial x_{Pk}(t_f)} = 0 \quad \text{if } x_{Pk}(t_f) \text{ is unspecified } (k = 1, \ldots, n_P) \tag{10}$$

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

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

The transversality conditions are referred to the initial and terminal times and are written as follows:

$$\frac{\partial \Phi}{\partial t_0} - H_0 = 0 \qquad \text{if } t_0 \text{ is unspecified} \tag{13}$$

$$\frac{\partial \Phi}{\partial t_f} + H_f = 0 \qquad \text{if } t_f \text{ is unspecified} \tag{14}$$

The two relationships (13) and (14) are needed only when the terminal time or the initial time is free; Eq. (13) becomes unnecessary if t_0 is specified, and Eq. (14) becomes unnecessary if t_f is specified. In addition, a saddle-point solution must satisfy the following pair of equations:

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

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

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

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

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

in conjunction with the following second-order conditions:

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

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

that is, the Hessian matrix $H_{\boldsymbol{u}_P \boldsymbol{u}_P}$ must be positive semidefinite, whereas the Hessian $H_{\boldsymbol{u}_E \boldsymbol{u}_E}$ must be negative semidefinite. Equations (17) and (18) ensure that the solution is stationary with respect to the control variables $\boldsymbol{u}_P$ and $\boldsymbol{u}_E$. 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 (17) and (18).

Equations (1–3), (7–14), and (17–20) constitute a two-point boundary-value problem (TPBVP), in which the unknowns are the state vectors $\boldsymbol{x}_P(t)$ and $\boldsymbol{x}_E(t)$; the control variables $\boldsymbol{u}_P(t)$ and $\boldsymbol{u}_E(t)$; the Lagrange multipliers $\boldsymbol{\lambda}_P(t)$, $\boldsymbol{\lambda}_E(t)$, and $\boldsymbol{\upsilon}$; and possibly the times t_0 and t_f.

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

The necessary conditions (7–14) and (17–20), in conjunction with the constraints (1–3), allow the translation of the problem into a TPBVP. In general, this kind of problem is not amenable to an analytic solution, and so numerical algorithms must be employed.

The semidirect collocation with nonlinear programming algorithm (semi-DCNLP algorithm), which is being employed in the present research, is based on the formal conversion of the two-sided optimization problem into a single-objective optimal control problem. Then the optimal control problem is transcribed into a nonlinear programming problem (by employing the highly accurate fifth-degree Gauss–Lobatto quadrature rules [24]), and solved through the use of an NLP solver, for example, NPSOL [25].

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

$$\tilde{\boldsymbol{x}}(t) = [\boldsymbol{x}_P^T(t) \quad \boldsymbol{x}_E^T(t) \quad \boldsymbol{\lambda}_E^T(t)]^T \tag{21}$$

The corresponding dynamic equation now includes Eq. (8) and is given by

$$\dot{\tilde{\boldsymbol{x}}}(t) = \left[\boldsymbol{f}_P^T \quad \boldsymbol{f}_E^T \quad -\boldsymbol{\lambda}_E^T(\partial \boldsymbol{f}_E / \partial \boldsymbol{x}_E)\right]^T \triangleq \tilde{\boldsymbol{f}} \tag{22}$$

where $\boldsymbol{f}_E^T = \boldsymbol{f}_E^T(\boldsymbol{x}_E, \boldsymbol{u}_E(\boldsymbol{x}_E, \boldsymbol{\lambda}_E, t), t) = \boldsymbol{f}_E^T(\boldsymbol{x}_E, \boldsymbol{\lambda}_E, t)$. Equations (11) and (12) constitute the boundary conditions for the adjoint variable $\boldsymbol{\lambda}_E$. Each of these equations can possibly depend on $\boldsymbol{\upsilon}$. The left-hand sides of the equations, which are independent of $\boldsymbol{\upsilon}$, are collected in the vector $\boldsymbol{\Psi}_{\text{EXT}}$. The remaining equations are opportunely handled to eliminate the components of $\boldsymbol{\upsilon}$. Then, the left-hand sides of the resulting equations, which do not yet include any component of $\boldsymbol{\upsilon}$, 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 $\tilde{H}$ includes an additional term corresponding to the differential equation for $\boldsymbol{\lambda}_E$:

$$\tilde{\Phi} = \phi + \tilde{\boldsymbol{\upsilon}}^T \boldsymbol{\Psi} = \phi + \boldsymbol{\upsilon}_1^T \boldsymbol{\Psi} + \boldsymbol{\upsilon}_{\text{EXT}}^T \boldsymbol{\Psi}_{\text{EXT}} \tag{23}$$

where $\tilde{\boldsymbol{\upsilon}} = \left[\boldsymbol{\upsilon}_1^T \quad \boldsymbol{\upsilon}_{\text{EXT}}^T\right]^T$.

$$\tilde{H} = \tilde{\boldsymbol{\lambda}}^T \tilde{\boldsymbol{f}} = \boldsymbol{\lambda}_{P(e)}^T \boldsymbol{f}_P + \boldsymbol{\lambda}_{E(e)}^T \boldsymbol{f}_E - \boldsymbol{\lambda}_{\lambda(e)}^T \left[\frac{\partial \boldsymbol{f}_E}{\partial \boldsymbol{x}_E}\right]^T \boldsymbol{\lambda}_E \tag{24}$$

where $\tilde{\boldsymbol{\lambda}} = \left[\boldsymbol{\lambda}_{P(e)}^T \quad \boldsymbol{\lambda}_{E(e)}^T \quad \boldsymbol{\lambda}_{\lambda(e)}^T\right]^T$. Hence, the zero-sum game has been converted into a typical optimal control problem. The numerical optimizer has to solve the problem

$$\min_{\tilde{\boldsymbol{u}}(t)} J \quad \text{subject to constraints} \quad \dot{\tilde{\boldsymbol{x}}}(t) = \tilde{\boldsymbol{f}} \quad \text{and} \quad \tilde{\boldsymbol{\Psi}} = \boldsymbol{0} \tag{25}$$

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 $t_N \equiv t_f$. The method of solution requires the discretization of the continuous variables; the extended state $\tilde{\boldsymbol{x}}$ ($\tilde{n}$-dimensional vector) is represented by $(2\tilde{n}N + \tilde{n})$ parameters, whereas the extended control $\tilde{\boldsymbol{u}}$ ($\tilde{m}$-dimensional vector) is represented by $(4\tilde{m}N + \tilde{m})$ parameters. In this research, for the implicit integration of the dynamic equations (22), the highly accurate fifth-degree Gauss–Lobatto quadrature rule [24] is employed. This rule allows the conversion of the $\tilde{n}$ scalar differential equations (22) into $2\tilde{n}N$ nonlinear collocation constraints.

The NLP solver is expected to produce the optimal values of the parameters involved, in particular those related to the state and control variables. Finally, the state components are interpolated through fifth-order polynomials, which represent the continuous approximations of their optimal time histories.

Additional details about the semi-DCNLP algorithm are reported in [19–21,26], which describe several applications modeled as zero-sum games and solved with the method outlined in this section.

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, so that this approach is likely to be unsuccessful especially for large, challenging problems, such as that at hand.

This difficulty can be resolved by using a systematic approach based on genetic algorithms, also referred to as evolutionary methods. As a matter of fact, they do not require any guess for the variables of the problem; the parameters involved in the process are coded as binary arrays and a complete set of them forms an individual. 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. 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 that is usually employed to preserve the best individual in the generation. Basically, evolutionary methods constitute an effective statistical search technique for selecting the best parameters, that is, the parameters that minimize the objective function. At the end of the process, genetic algorithms are expected to generate the best individual, which includes the optimal values of all the parameters.

An intrinsic limitation of genetic algorithms is their poor numerical accuracy, which is mainly related to the representation of the parameters as binary arrays with a finite number of digits. This can be ameliorated in part by using real GAs. However, when used here as a preprocessing technique, a genetic algorithm (GA) is only required to provide a reasonable guess for the NLP solver by selecting the parameters in a suitable feasible region of the search space.

All the equations that form the TPBVP, that is, Eqs. (1–3), (7–14), and (17–20), are to be employed by the GA. In more detail, the GA preprocessing is based on the following points:

1) Each individual is composed of all the unknown values of the states ($\boldsymbol{x}_P$ and $\boldsymbol{x}_E$) and costates ($\boldsymbol{\lambda}_P$ and $\boldsymbol{\lambda}_E$) at t_0 and the possible components of $\boldsymbol{v}$ included in Eqs. (9–12), as well as the unknown time-independent variables (e.g., the time of flight).

2) For both players, the control variables are expressed as functions of $\boldsymbol{x}_P, \boldsymbol{x}_E, \boldsymbol{\lambda}_P$, and $\boldsymbol{\lambda}_E, t$ through the relationships (17–20):

$$\boldsymbol{u}_P = \boldsymbol{u}_P(\boldsymbol{x}_P, \boldsymbol{\lambda}_P, t) \quad \text{and} \quad \boldsymbol{u}_E = \boldsymbol{u}_E(\boldsymbol{x}_E, \boldsymbol{\lambda}_E, t) \tag{26}$$

3) The equations of motion (1) and (2) and the adjoint equations (7) and (8) are numerically integrated for each individual.

4) The necessary conditions (3) and (9–14) are assimilated to constraints $\{c_j\}_{j=1,\ldots,n_C}$ and used as penalty terms in the objective function $\bar{J}$ to be minimized by the GA:

$$\bar{J} = \sum_{j=1}^{n_C} k_j c_j^2 \tag{27}$$

where the weights k_j are given positive numbers and n_C is the total number of constraints.

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 (27) 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 that the GA has to adopt. However, with the proper settings, the GA is expected to generate a reasonable guess for the NLP solver, as demonstrated previously for a dynamic game by Horie and Conway [27] and subsequently by Pontani and Conway [26].

V. Problem Definition

In this research, the problem of the optimal interception of an optimally evasive spacecraft by a pursuing spacecraft is considered. Both spacecraft are given modest propulsive capabilities to perform their respective maneuvers, which start simultaneously (at $t = t_0 = 0$). The problem is formulated as a zero-sum differential game, in which the pursuer P tries to catch the evader E. The time for interception, t_f, which is assumed as the termination of the game, is to be minimized by P and maximized by E. Hence, the objective function is simply

$$J = t_f \tag{28}$$

In realistic scenarios, both spacecraft will employ all their maneuvering capabilities, that is, both spacecraft will use their maximum thrust during the game. In this study, a low, constant thrust-to-mass ratio is assumed along the entire trajectory for both spacecraft, that is, the thrust pointing direction is the only control. Interception terminates the combat game and occurs when the pursuer achieves the instantaneous position of the evader. A sufficient condition that ensures that capture actually ends the game is simply

$$(T_p/m_p) > (T_E/m_E) \tag{29}$$

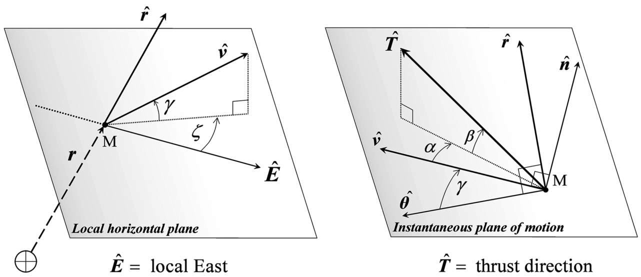
that is, the thrust-to-mass ratio of P, (T_P/m_P), is assumed to be greater than that of E, (T_E/m_E). The condition (29) states that the pursuer has superior capabilities with respect to the evader. As unbounded controls are assumed for both spacecraft, the condition (29) implies that interception can occur in a finite time.

This study employs a point-mass model to describe the trajectories of the two spacecraft in the context of a 3-degree-of-freedom problem of optimal interception. The motion of the two spacecraft is supposed to take place at an altitude greater than 100 km ($h \geq h_{\min} = 100$ km), so that atmospheric forces can be considered negligible.

A. Equations of Motion

The trajectories of the two players can be described with a convenient set of variables related to the orbital references $(\hat{r}_i, \hat{\theta}_i, \hat{h}_i)$ and $(\hat{n}_i, \hat{v}_i, \hat{h}_i)$ $(i = P, E)$, where $\hat{h}_i$ is the vector associated to the angular momentum, and $\hat{r}_i$ and $\hat{v}_i$ correspond to the instantaneous position and velocity vectors $\boldsymbol{r}_i$ and $\boldsymbol{v}_i$, respectively. This set of variables defines the state of each spacecraft and includes the instantaneous radius r_i, the velocity magnitude v_i, the flight-path angle γ_i, the absolute longitude ξ_i, the latitude ϕ_i, and the velocity coazimuth angle ζ_i (which is portrayed in Fig. 1a). The control is performed with the thrust direction, described by the two angles, α_i and β_i, shown in Fig. 1b. For each player, denoted with $i = P$ or E, the equations of motion are written as follows:

$$\dot{r}_i = v_i \sin\gamma_i \tag{30}$$



a) b)

Fig. 1 Axes and angles related to the following: a) local horizontal plane, and b) instantaneous plane of flight.

$$\dot{v}_i = \frac{T_i}{m_i}\cos\alpha_i\cos\beta_i - \frac{\mu_E\sin\gamma_i}{r_i^2} \tag{31}$$

$$\dot{\gamma}_i = \frac{v_i\cos\gamma_i}{r_i} + \frac{T_i}{m_i}\frac{\sin\alpha_i\cos\beta_i}{v_i} - \frac{\mu_E\cos\gamma_i}{r_i^2 v_i} \tag{32}$$

$$\dot{\xi}_i = \frac{v_i\cos\gamma_i\cos\zeta_i}{r_i\cos\phi_i} \tag{33}$$

$$\dot{\phi}_i = \frac{v_i\cos\gamma_i\sin\zeta_i}{r_i} \tag{34}$$

$$\dot{\zeta}_i = \frac{T_i}{m_i}\frac{\sin\beta_i}{v_i\cos\gamma_i} - \frac{v_i\cos\gamma_i\sin\phi_i\cos\zeta_i}{r_i\cos\phi_i} \tag{35}$$

where (T_i/m_i) represents the (constant) thrust-to-mass ratio of each spacecraft.

The initial conditions are specified and are related to the orbital elements of the two spacecraft at $t_0(=0)$. The game ends when the pursuer reaches the instantaneous position of the evader, that is, when

$$r_P(t_f) = r_E(t_f), \qquad \xi_P(t_f) = \xi_E(t_f), \qquad \phi_P(t_f) = \phi_E(t_f) \tag{36}$$

The state of each spacecraft is written as follows:

$$\begin{aligned}\boldsymbol{x}_P &= \begin{bmatrix} r_P & v_P & \gamma_P & \xi_P & \phi_P & \zeta_P\end{bmatrix}^T \triangleq \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6\end{bmatrix}^T\\ \boldsymbol{x}_E &= \begin{bmatrix} r_E & v_E & \gamma_E & \xi_E & \phi_E & \zeta_E\end{bmatrix}^T \triangleq \begin{bmatrix} x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12}\end{bmatrix}^T\end{aligned} \tag{37}$$

whereas the control variables are represented by the two angles, α_i and β_i, collected in $\boldsymbol{u}_P$ and $\boldsymbol{u}_E$:

$$\boldsymbol{u}_P = \begin{bmatrix}\alpha_P & \beta_P\end{bmatrix}^T \triangleq \begin{bmatrix}u_1 & u_2\end{bmatrix}^T \quad\text{and}\quad \boldsymbol{u}_E = \begin{bmatrix}\alpha_E & \beta_E\end{bmatrix}^T \triangleq \begin{bmatrix}u_3 & u_4\end{bmatrix}^T \tag{38}$$

The boundary conditions are inserted in $\boldsymbol{\Psi}$:

$$\boldsymbol{\Psi} = \begin{bmatrix} x_{1f} - x_{7f}\\ x_{4f} - x_{10f}\\ x_{5f} - x_{11f}\end{bmatrix} = \boldsymbol{0} \tag{39}$$

Then, the adjoint variables conjugate to the state equations are introduced as

$$\boldsymbol{\lambda}_P \triangleq \begin{bmatrix}\lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6\end{bmatrix}^T \quad\text{and}\quad \boldsymbol{\lambda}_E \triangleq \begin{bmatrix}\lambda_7 & \lambda_8 & \lambda_9 & \lambda_{10} & \lambda_{11} & \lambda_{12}\end{bmatrix}^T \tag{40}$$

whereas the multiplier $\boldsymbol{\upsilon}$, conjugate to $\boldsymbol{\Psi}$, is simply given by $\boldsymbol{\upsilon} = [\upsilon_1 \quad \upsilon_2 \quad \upsilon_3]^T$. After introducing the Hamiltonian H and the function Φ, defined by Eqs. (5) and (6), the necessary conditions (7) and (8) yield

$$\begin{aligned}\dot{\lambda}_1 = {}& \left(x_2\lambda_3\cos x_3 + \lambda_4 x_2\cos x_3\frac{\cos x_6}{\cos x_5}\right.\\ & \left. + x_2\lambda_5\cos x_3\sin x_6 - x_2\lambda_6\cos x_3\sin x_5\frac{\cos x_6}{\cos x_5}\right)\frac{1}{x_1^2}\\ & - \left(2\mu_E\lambda_2\sin x_3 + 2\mu_E\lambda_3\frac{\cos x_3}{x_2}\right)\frac{1}{x_1^3}\end{aligned} \tag{41}$$

$$\begin{aligned}\dot{\lambda}_2 = {}& -\lambda_1\sin x_3 - \lambda_3\left[\cos x_3\left(\frac{1}{x_1} + \frac{\mu_E}{x_1^2 x_2^2}\right) - \frac{T_P}{m_P x_2^2}\sin u_1\cos u_2\right]\\ & - \lambda_4\cos x_3\frac{\cos x_6}{x_1\cos x_5} - \lambda_5\cos x_3\frac{\sin x_6}{x_1}\\ & + \lambda_6\left(\frac{T_P\sin u_2}{m_P x_2^2\cos x_3} + \frac{\cos x_3\sin x_5\cos x_6}{x_1\cos x_5}\right)\end{aligned} \tag{42}$$

$$\begin{aligned}\dot{\lambda}_3 = {}& -x_2\lambda_1\cos x_3 + \mu_E\lambda_2\frac{\cos x_3}{x_1^2} + \lambda_3\sin x_3\left(\frac{x_2}{x_1} - \frac{\mu_E}{x_1^2 x_2}\right)\\ & + x_2\lambda_4\sin x_3\frac{\cos x_6}{x_1\cos x_5} + x_2\lambda_5\sin x_3\frac{\sin x_6}{x_1}\\ & - \lambda_6\left[\frac{T_P}{m_P}\frac{\sin u_2\sin x_3}{x_2(\cos x_3)^2} + x_2\frac{\sin x_3\sin x_5\cos x_6}{x_1\cos x_5}\right]\end{aligned} \tag{43}$$

$$\dot{\lambda}_4 = 0 \tag{44}$$

$$\dot{\lambda}_5 = \frac{x_2\cos x_3\cos x_6}{x_1(\cos x_5)^2}(-\lambda_4\sin x_5 + \lambda_6) \tag{45}$$

$$\dot{\lambda}_6 = \frac{x_2\cos x_3}{x_1\cos x_5}(\lambda_4\sin x_6 - \lambda_5\cos x_5\cos x_6 - \lambda_6\sin x_5\sin x_6) \tag{46}$$

with the respective boundary conditions (9–12):

$$\begin{gathered}\begin{cases}\lambda_{1f} = \upsilon_1\\ \lambda_{4f} = \lambda_4 = \upsilon_2\\ \lambda_{5f} = \upsilon_3\\ \lambda_{2f} = \lambda_{3f} = \lambda_{6f} = 0\end{cases} \qquad \begin{cases}\lambda_{7f} = -\upsilon_1\\ \lambda_{10f} = \lambda_{10} = -\upsilon_2\\ \lambda_{11f} = -\upsilon_3\\ \lambda_{8f} = \lambda_{9f} = \lambda_{12f} = 0\end{cases}\\ \Rightarrow \begin{cases}\lambda_{1f} + \lambda_{7f} = 0\\ \lambda_4 + \lambda_{10} = 0\\ \lambda_{5f} + \lambda_{11f} = 0\\ \lambda_{2f} = \lambda_{3f} = \lambda_{6f} = \lambda_{8f} = \lambda_{9f} = \lambda_{12f} = 0\end{cases}\end{gathered} \tag{47}$$

where the subscript f denotes the value of the corresponding variable at t_f. The adjoint equations for the evader are formally identical to those of the pursuer and can be written by simply shifting by 6 all the indexes of the state components included in the relationships (41–46) (after replacing (T_P/m_P) with (T_E/m_E), u_1 with u_3, and u_2 with u_4). For the control variables, the first-order conditions (17) and (18) hold:

$$\frac{\partial H}{\partial u_1} = \frac{T_P}{m_P x_2}\cos u_2(\lambda_3\cos u_1 - x_2\lambda_2\sin u_1) = 0 \tag{48}$$

$$\begin{aligned}\frac{\partial H}{\partial u_2} = {}& -\frac{T_P}{m_P}\lambda_2\sin u_2\cos u_1\\ & + \frac{T_P}{m_P x_2}\left(\lambda_6\frac{\cos u_2}{\cos x_3} - \lambda_3\sin u_1\sin u_2\right) = 0\end{aligned} \tag{49}$$

$$\frac{\partial H}{\partial u_3} = \frac{T_E}{m_E x_8} \cos u_4 (\lambda_9 \cos u_3 - x_8 \lambda_8 \sin u_3) = 0 \qquad (50)$$

$$\frac{\partial H}{\partial u_4} = -\frac{T_E}{m_E} \lambda_8 \sin u_4 \cos u_3 + \frac{T_E}{m_E x_8}\left(\lambda_{12}\frac{\cos u_4}{\cos x_9} - \lambda_9 \sin u_3 \sin u_4\right) = 0 \qquad (51)$$

Under the assumption that $\cos u_2 \neq 0$ and $\cos u_4 \neq 0$, Eqs. (48) and (50) yield two pairs of possible values for u_1 and u_3:

$$u_1^{(A)} = \arctan\left(\frac{\lambda_3}{x_2\lambda_2}\right) \quad \text{and} \quad u_1^{(B)} = \arctan\left(\frac{\lambda_3}{x_2\lambda_2}\right) + \pi \qquad (52)$$

$$u_3^{(A)} = \arctan\left(\frac{\lambda_9}{x_8\lambda_8}\right) \quad \text{and} \quad u_3^{(B)} = \arctan\left(\frac{\lambda_9}{x_8\lambda_8}\right) + \pi \qquad (53)$$

Moreover, Eqs. (49) and (51) yield two pairs of possible values for u_2 and u_4:

$$u_2^{(A)} = \arctan\left(\frac{\lambda_6 - \lambda_3 \cos x_3 \sin u_1}{x_2\lambda_2 \cos x_3 \cos u_1}\right) \quad \text{and} \quad u_2^{(B)} = \arctan\left(\frac{\lambda_6 - \lambda_3 \cos x_3 \sin u_1}{x_2\lambda_2 \cos x_3 \cos u_1}\right) + \pi \qquad (54)$$

$$u_4^{(A)} = \arctan\left(\frac{\lambda_{12} - \lambda_9 \cos x_9 \sin u_3}{x_8\lambda_8 \cos x_9 \cos u_3}\right) \quad \text{and} \quad u_4^{(B)} = \arctan\left(\frac{\lambda_{12} - \lambda_9 \cos x_9 \sin u_3}{x_8\lambda_8 \cos x_9 \cos u_3}\right) + \pi \qquad (55)$$

Hence, for each player, four possible solutions for $\{\boldsymbol{u}_P, \boldsymbol{u}_E\}$ fulfilling the first-order conditions exist:

$$\{u_1^{(A)}, u_2^{(A)}\}, \quad \{u_1^{(A)}, u_2^{(B)}\}, \quad \{u_1^{(B)}, u_2^{(A)}\}, \quad \{u_1^{(B)}, u_2^{(B)}\} \quad \text{for the pursuer } P \qquad (56)$$

$$\{u_3^{(A)}, u_4^{(A)}\}, \quad \{u_3^{(A)}, u_4^{(B)}\}, \quad \{u_3^{(B)}, u_4^{(A)}\}, \quad \{u_3^{(B)}, u_4^{(B)}\} \quad \text{for the evader } E \qquad (57)$$

The selection of the correct solution can be made by taking into account the second-order conditions (19) and (20), which lead to the following relationships:

$$\frac{\partial^2 H}{\partial \boldsymbol{u}_P^2} = \begin{bmatrix} H_{11}^{(P)} & H_{12}^{(P)} \\ H_{21}^{(P)} & H_{22}^{(P)} \end{bmatrix} \geq 0$$

$$\text{where} \begin{cases} H_{11}^{(P)} = -\frac{T_P \cos u_2}{m_P x_2}(x_2\lambda_2 \cos u_1 + \lambda_3 \sin u_1) \\ H_{12}^{(P)} = H_{21}^{(P)} = \frac{T_P \sin u_2}{m_P x_2}(x_2\lambda_2 \sin u_1 - \lambda_3 \cos u_1) \\ H_{22}^{(P)} = -\frac{T_P}{m_P}\left[\frac{1}{x_2}\left(\lambda_3 \sin u_1 \cos u_2 + \lambda_6 \frac{\sin u_2}{\cos x_3}\right) + \lambda_2 \cos u_1 \cos u_2\right] \end{cases} \qquad (58)$$

$$\frac{\partial^2 H}{\partial \boldsymbol{u}_E^2} = \begin{bmatrix} H_{11}^{(E)} & H_{12}^{(E)} \\ H_{21}^{(E)} & H_{22}^{(E)} \end{bmatrix} \leq 0$$

$$\text{where} \begin{cases} H_{11}^{(E)} = -\frac{T_E \cos u_4}{m_E x_8}(x_8\lambda_8 \cos u_3 + \lambda_9 \sin u_3) \\ H_{12}^{(E)} = H_{21}^{(E)} = \frac{T_E \sin u_4}{m_E x_8}(x_8\lambda_8 \sin u_3 - \lambda_9 \cos u_3) \\ H_{22}^{(E)} = -\frac{T_E}{m_E}\left[\frac{1}{x_8}\left(\lambda_9 \sin u_3 \cos u_4 + \lambda_{12} \frac{\sin u_4}{\cos x_9}\right) + \lambda_8 \cos u_3 \cos u_4\right] \end{cases} \qquad (59)$$

Each of these conditions is equivalent to two inequalities:

$$H_{11}^{(P)} + H_{22}^{(P)} \geq 0 \quad \text{and} \quad H_{11}^{(P)}H_{22}^{(P)} - H_{12}^{(P)}H_{12}^{(P)} \geq 0 \quad (\text{holding for } P) \qquad (60)$$

$$H_{11}^{(E)} + H_{22}^{(E)} \leq 0 \quad \text{and} \quad H_{11}^{(E)}H_{22}^{(E)} - H_{12}^{(E)}H_{12}^{(E)} \geq 0 \quad (\text{holding for } E) \qquad (61)$$

Equations (60) and (61) can be used to choose the correct control law (at any time t) from the four options reported in Eqs. (56) and (57). Finally, as t_f is unspecified, the transversality condition (14) must hold:

$$H_f + 1 = 0 \qquad (62)$$

For the sake of brevity, the full expression of H_f is not reported. It is straightforward to derive from Eq. (5) employing Eqs. (30–35). It is worth mentioning that the Hamiltonian is time independent; therefore, its value must be constant ($H = -1\ \forall t$), which is consistent with the general property of autonomous systems.

Equations (30–36), (41–51), and (60–62) form the TPBVP associated to the zero-sum game at hand, where the unknowns are represented by the state vectors $\boldsymbol{x}_P(t)$ and $\boldsymbol{x}_E(t)$; the control vectors $\boldsymbol{u}_P(t)$ and $\boldsymbol{u}_E(t)$; the adjoint variables $\boldsymbol{\lambda}_P(t)$, $\boldsymbol{\lambda}_E(t)$, and $\boldsymbol{\upsilon}$; and the final time t_f.

VI. Method of Solution

The numerical method of solution of the orbital pursuit–evasion game of interest is based on the joint application of a genetic algorithm in the preprocessing (aimed at providing a starting, approximate guess) and a local method (the semi-DCNLP) tailored to solving differential games with a high degree of accuracy [24].

A. Genetic Algorithm Preprocessing Settings

During the GA preprocessing, the flight time t_f and the unknown values of the costate at t_0 (denoted with the subscript 0) are assumed as parameters (and coded in each individual):

$$\{\lambda_{1,0}, \lambda_{2,0}, \lambda_{3,0}, \lambda_4, \lambda_{5,0}, \lambda_{6,0}, \lambda_{7,0}, \lambda_{8,0}, \lambda_{9,0}, \lambda_{10}, \lambda_{11,0}, \lambda_{12,0}, t_f\} \qquad (63)$$

The relationships (52–55) [with the respective second-order conditions (60) and (61)] are employed to express the control variables as functions of the state and costate. Then, the state and costate equations are integrated for each individual, with the exception of that for λ_{10}, because λ_{10} turns out to be constant (as λ_4). Finally, the boundary conditions (39) and the necessary conditions not yet considered [i.e., (47) and (62)] are assimilated to constraints and evaluated. As stated in the previous section, the number of parameters is exactly equal to the number of constraints (i.e., 13), and the GA selects the best individual, that is, the set of parameters that minimizes Eq. (27), which is a nonnegative function of the constraint violations only.

For the problem at hand, the GA code by Deb et al. [28,29] 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 500 individuals and 100 generations.

B. Semi-DCNLP Settings

The semi-DCNLP algorithm is based on the inclusion of the adjoint variable $\boldsymbol{\lambda}_i$ ($i = P$ or E) of one player as an additional component of the extended state $\tilde{\boldsymbol{x}}$, as described in Sec. III. For the problem at hand, the control of the evader is expressed as a function of its states and adjoint variables, through Eqs. (53), (55), and (61). In contrast, the control of the pursuer is found numerically by the NLP problem solver, that is, the Fortran package NPSOL [25]. Hence, the time-varying adjoint variables of the evader are included as additional components of the extended state vector $\tilde{\boldsymbol{x}}$:

$$\tilde{\boldsymbol{x}} = [x_P^T \quad x_E^T \quad \lambda_7 \quad \lambda_8 \quad \lambda_9 \quad \lambda_{11} \quad \lambda_{12}]^T \tag{64}$$

whereas the multiplier λ_{10} is treated as a parameter, because it is constant. In this context, the control vector $\tilde{\boldsymbol{u}}$ is given by $\tilde{\boldsymbol{u}} = \boldsymbol{u}_P$, that is, it includes α_P and β_P only.

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_{10} and the terminal time t_f. In this study, the time interval is partitioned into 14 subarcs ($N = 14$). Moreover, $\tilde{n} = 17$ and $\tilde{m} = 2$, so the total number of parameters involved is $(2\tilde{n}N + \tilde{n} + 4\tilde{m}N + \tilde{m} + 2) = 609$. The NLP solver has to satisfy $2\tilde{n}N$ collocation constraints and the boundary conditions (3), and $\lambda_{8f} = \lambda_{9f} = \lambda_{12f} = 0$. [The remaining equations included in Eq. (47) 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 $(2\tilde{n}N + 6) = 582$.

Two fundamental settings concern the numerical accuracy required for the NLP solver: the optimality tolerance, which was set to 10^{-6}, and the constraint violation tolerance, which was set to 10^{-8}.

VII. Numerical Results

In this research, canonical units were used for the numerical solution. 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 s. In canonical units, $1\ \mathrm{DU/TU}^2 \equiv 1\ \mathrm{g} = 9.798 \cdot 10^{-3}\ \mathrm{km/s}^2$. Moreover, the minimum value for the altitude is set to 100 km ($=1.568 \cdot 10^{-2}$ DU), whereas the following values for the thrust-to-mass ratios are assumed for all the test cases:

$$T_P/m_P = 0.1(\mathrm{DU/TU}^2) = 0.1\ \mathrm{g} \quad \text{and}$$
$$T_E/m_E = 0.05(\mathrm{DU/TU}^2) = 0.05\ \mathrm{g} \tag{65}$$

Five test cases have been considered. They differ in the starting conditions of each spacecraft, related to the respective (initial) orbital elements, which are the semimajor axis a, the eccentricity e, the inclination i, the right ascension of the ascending node Ω, the argument of perigee ω, and the mean anomaly, M. The respective values at t_0 are specified and denoted by the subscript 0. For circular orbits, the perigee is not defined and only the argument of latitude $\theta(\equiv \omega + M)$ is meaningful. For each case, the saddle-point solutions and the corresponding saddle-point trajectories were obtained.

Test case 1 (circular orbits, initial altitude of 200 km for both spacecraft): The orbital elements at t_0 and the related initial values of the state components are

$$P\begin{cases} a_{P,0} = 6578.165\ \mathrm{km} \\ e_{P,0} = 0 \\ i_{P,0} = 30\ \mathrm{deg} \\ \Omega_{P,0} = 10\ \mathrm{deg} \\ \theta_{P,0} = 0\ \mathrm{deg} \end{cases} \Rightarrow \begin{cases} r_{P,0} = 6578.165\ \mathrm{km} \\ v_{P,0} = 7.784\ \mathrm{km/s} \\ \gamma_{P,0} = 0\ \mathrm{deg} \\ \xi_{P,0} = 10\ \mathrm{deg} \\ \phi_{P,0} = 0\ \mathrm{deg} \\ \zeta_{P,0} = 30\ \mathrm{deg} \end{cases}$$
$$E\begin{cases} a_{E,0} = 6578.165\ \mathrm{km} \\ e_{E,0} = 0 \\ i_{E,0} = 50\ \mathrm{deg} \\ \Omega_{E,0} = 30\ \mathrm{deg} \\ \theta_{E,0} = 10\ \mathrm{deg} \end{cases} \Rightarrow \begin{cases} r_{E,0} = 6578.165\ \mathrm{km} \\ v_{E,0} = 7.784\ \mathrm{km/s} \\ \gamma_{E,0} = 0\ \mathrm{deg} \\ \xi_{E,0} = 36.5\ \mathrm{deg} \\ \phi_{E,0} = 7.6\ \mathrm{deg} \\ \zeta_{E,0} = 49.6\ \mathrm{deg} \end{cases} \tag{66}$$

The GA preprocessing was employed to provide the first guess for the subsequent use of the semi-DCNLP method. The method outlined in Sec. III is now described with reference to the application at hand. First of all, for each player the control can be written as a function of the state and costate variables with the relationships (48–51), which yield the four solutions (56) and (57). It is relatively straightforward to prove that at any instant only one of these four options fulfills the second-order conditions (60) and (61), which, in fact, are used to select the correct control among the four available options. Then the state and costate Eqs. (30–35) and Eqs. (41–46), respectively, are numerically integrated for each individual. Finally, the constraint violation associated to each individual is evaluated with Eq. (27), and all the individuals are employed for generating a new population, according to the GA selection mechanisms. With reference to this first test case, after 100 generations, the GA produced the best individual, which is representative of the optimal values (in canonical units) for the parameter set (63):

$$\begin{aligned} &\{\lambda_{1,0}, \lambda_{2,0}, \lambda_{3,0}, \lambda_4, \lambda_{5,0}, \lambda_{6,0}, \lambda_{7,0}, \lambda_{8,0}, \lambda_{9,0}, \lambda_{10}, \lambda_{11,0}, \lambda_{12,0}, t_f\} = \\ &= \{1.665, 1.515, 1.845, -2.531, 3.092, \\ &-1.477, 1.625, 2.914, 0.147, -0.162, 0.362, -0.032, 3.582\} \end{aligned} \tag{67}$$

With reference to this parameter set, Fig. 2 portrays the time history of the optimal selection of the control as a function of state and costate among the four available options. For the evader, two switches occurred, whereas, for the pursuer, only one. The result provided by the GA was then employed by the semi-DCNLP, which produced the actual (accurate) saddle-point solution. It is worth mentioning that in the execution of the semi-DCNLP algorithm this process of selection of the correct option for the control regards the evader only. This is due to the fact that the analytical conditions holding for the control are employed only for a single player in the semi-DCNLP method.

The control from the GA preprocessing and the optimal control are portrayed in Fig. 3. In this case, for the pursuer, the similarities between the preprocessed and optimal control angles are weak. In contrast, the preprocessed solution for the evader control angles partially resembles the optimal solution, in particular, $\beta_E^{(\mathrm{GA})}$ is discontinuous, as well as the respective optimal control law $\beta_E^{(\mathrm{sDCNLP})}$; in addition, $\alpha_E^{(\mathrm{GA})}$ and $\alpha_E^{(\mathrm{sDCNLP})}$ are qualitatively similar. Despite the dissimilarity, the guess solution generated by the GA preprocessing is sufficient to yield convergence to a saddle-point solution by the semi-DCNLP algorithm. Figure 4 shows the saddle-point trajectories leading to interception, whereas some state components corresponding to the saddle-point solution are illustrated in Fig. 5: altitude, velocity, flight-path angle, and coazimuth angle.

Test case 2 (circular orbits, initial altitude of 200 km for both spacecraft): The orbital elements at t_0 and the related initial values of the state components (denoted by the subscript 0) are

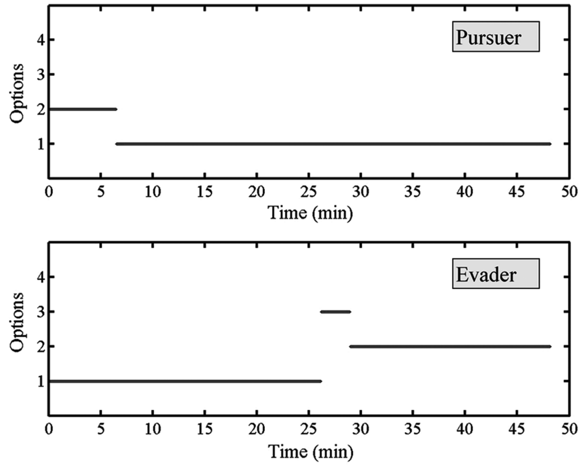


Fig. 2 Correct control selection among four options in the GA preprocessing.

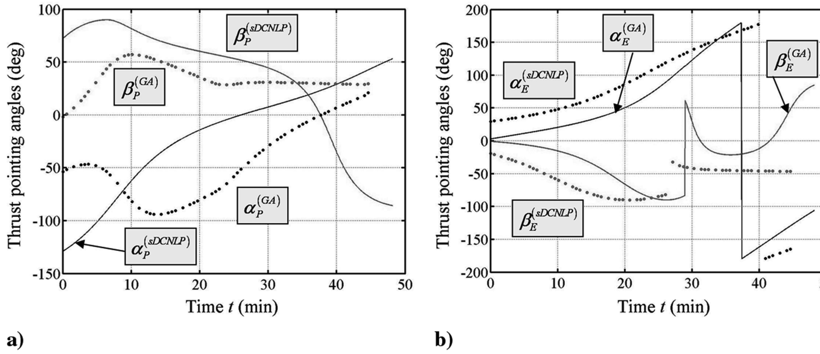


Fig. 3 Test case 1: preprocessed and optimal control time histories.

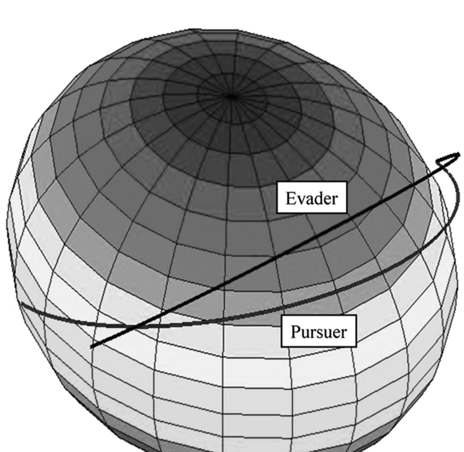


Fig. 4 Test case 1: SP trajectories of the two spacecraft.

$$P\begin{cases} a_{P,0} = 6578.165 \text{ km} \\ e_{P,0} = 0 \\ i_{P,0} = 30 \text{ deg} \\ \Omega_{P,0} = 10 \text{ deg} \\ \theta_{P,0} = 0 \text{ deg} \end{cases} \Rightarrow \begin{cases} r_{P,0} = 6578.165 \text{ km} \\ v_{P,0} = 7.784 \text{ km/s} \\ \gamma_{P,0} = 0 \text{ deg} \\ \xi_{P,0} = 10 \text{ deg} \\ \phi_{P,0} = 0 \text{ deg} \\ \zeta_{P,0} = 30 \text{ deg} \end{cases}$$

$$E\begin{cases} a_{E,0} = 6578.165 \text{ km} \\ e_{E,0} = 0 \\ i_{E,0} = 30 \text{ deg} \\ \Omega_{E,0} = 20 \text{ deg} \\ \theta_{E,0} = 0 \text{ deg} \end{cases} \Rightarrow \begin{cases} r_{E,0} = 6578.165 \text{ km} \\ v_{E,0} = 7.784 \text{ km/s} \\ \gamma_{E,0} = 0 \text{ deg} \\ \xi_{E,0} = 20 \text{ deg} \\ \phi_{E,0} = 0 \text{ deg} \\ \zeta_{E,0} = 30 \text{ deg} \end{cases} \qquad (68)$$

For this second case (as well as for all the remaining cases), the preprocessing proceeds just as described for test case 1. Figure 6 illustrates the preprocessed and the optimal control laws for both spacecraft. It emerges that the optimal and the preprocessed control angles are not far away from each other, that is, the GA preprocessing produced a reasonable guess solution, qualitatively similar to the optimal one. Figure 7 shows the saddle-point trajectories leading to interception. Some state components corresponding to the saddle-point solution are portrayed in Fig. 8: altitude (directly related to the radius r), velocity, flight-path angle, and coazimuth angle.

Test case 3 (elliptic orbits; initial perigee and apogee altitudes of 200 and 800 km for both spacecraft): The orbital elements at t_0 and the related initial values of the state components are

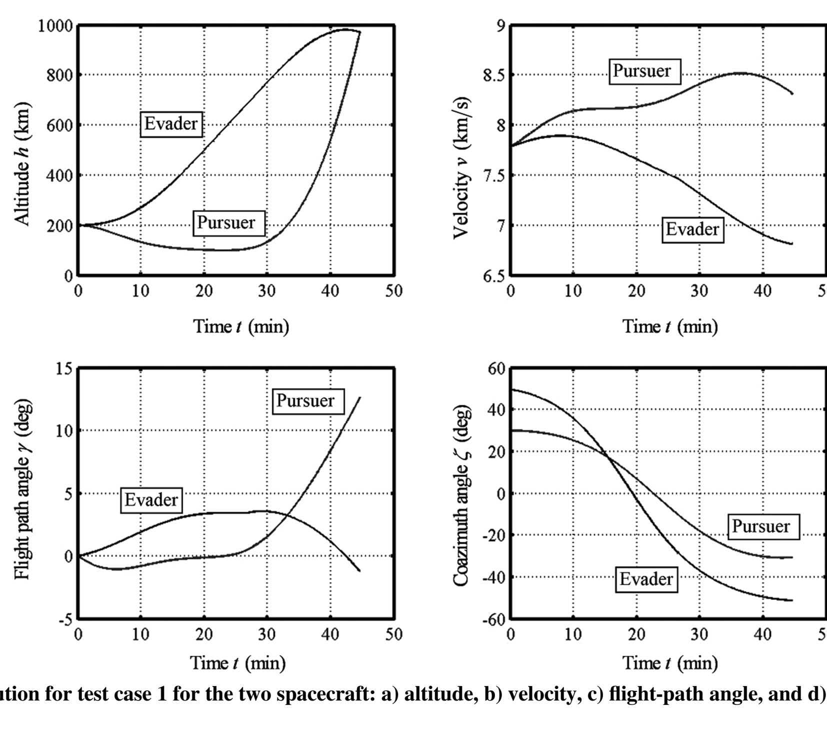


Fig. 5 SP solution for test case 1 for the two spacecraft: a) altitude, b) velocity, c) flight-path angle, and d) coazimuth angle.

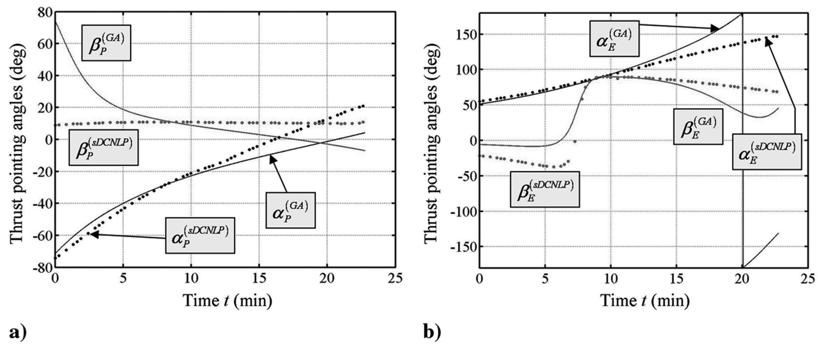


Fig. 6 Test case 2: preprocessed and optimal control time histories.

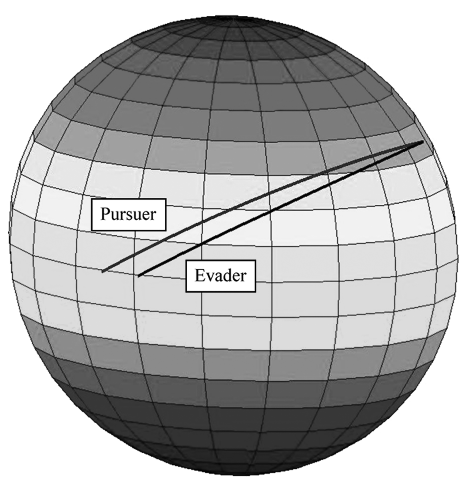


Fig. 7 Test case 2: SP trajectories of the two spacecraft.

$$
P\begin{cases} a_{P,0} = 6878.165 \text{ km} \\ e_{P,0} = 0.044 \\ i_{P,0} = 30 \text{ deg} \\ \Omega_{P,0} = 10 \text{ deg} \\ \omega_{P,0} = 30 \text{ deg} \\ M_{P,0} = -120 \text{ deg} \end{cases} \Rightarrow \begin{cases} r_{P,0} = 7037.658 \text{ km} \\ v_{P,0} = 7.438 \text{ km/s} \\ \gamma_{P,0} = -2.1 \text{ deg} \\ \xi_{P,0} = -84.9 \text{ deg} \\ \phi_{P,0} = -29.9 \text{ deg} \\ \zeta_{P,0} = -2.4 \text{ deg} \end{cases}
$$

$$
E\begin{cases} a_{E,0} = 6878.165 \text{ km} \\ e_{E,0} = 0.044 \\ i_{E,0} = 150 \text{ deg} \\ \Omega_{E,0} = 70 \text{ deg} \\ \omega_{E,0} = 30 \text{ deg} \\ M_{E,0} = -120 \text{ deg} \end{cases} \Rightarrow \begin{cases} r_{E,0} = 7037.658 \text{ km} \\ v_{E,0} = 7.438 \text{ km/s} \\ \gamma_{E,0} = -2.1 \text{ deg} \\ \xi_{E,0} = 164.9 \text{ deg} \\ \phi_{E,0} = -29.9 \text{ deg} \\ \zeta_{E,0} = -177.6 \text{ deg} \end{cases} \tag{69}
$$

Figure 9 illustrates the preprocessed and the optimal control laws for both spacecraft. It emerges that there is no apparent similarity between the preprocessed and optimal saddle-point control angles. Nevertheless, starting from the preprocessed solution, the semi-DCNLP is able to find an optimal saddle-point solution, thus indicating again that the local algorithm needs only a feasible guess solution, at least in this specific case. Figure 10 shows the saddle-point trajectories leading to interception. Some state components corresponding to the saddle-point solution are portrayed in Fig. 11: altitude, velocity, flight–path angle, and coazimuth angle.

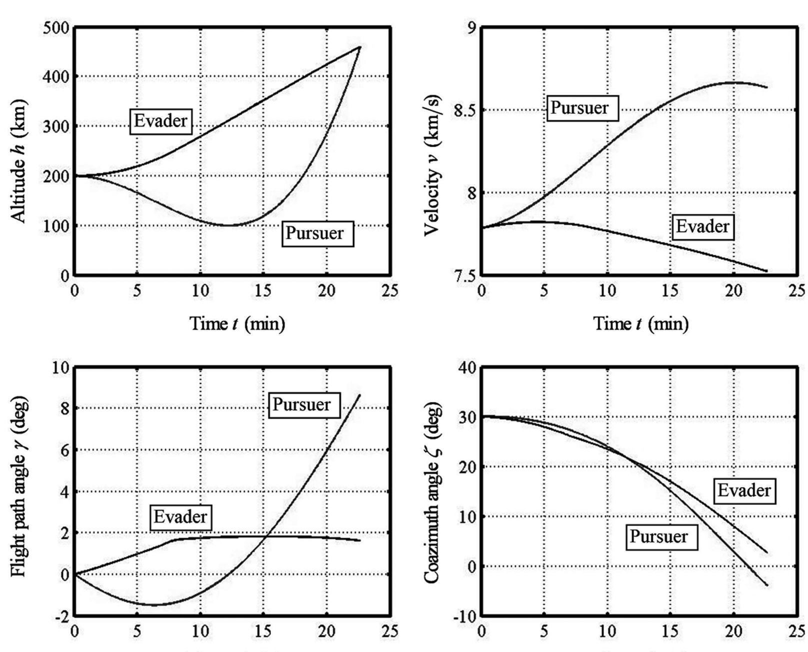


Fig. 8 SP solution for test case 2 for the two spacecraft: a) altitude, b) velocity, c) flight-path angle, and d) coazimuth angle.

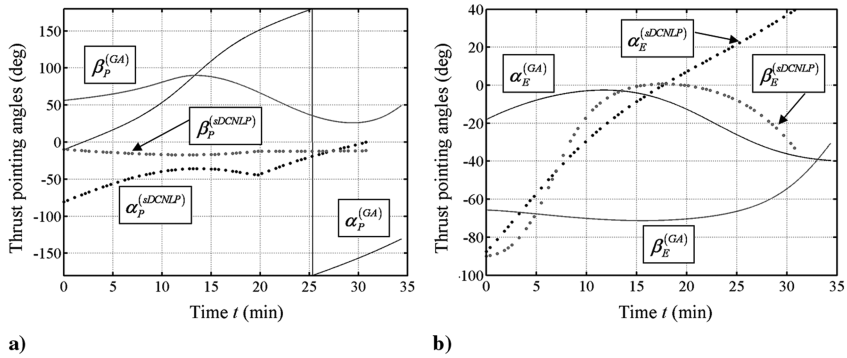


Fig. 9 Test case 3: preprocessed and optimal control time histories.

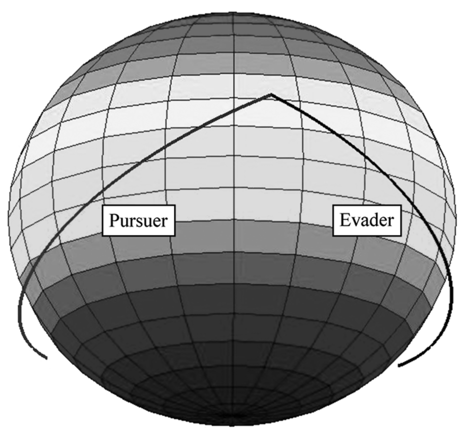


Fig. 10 Test case 3: SP trajectories of the two spacecraft.

Test case 4 (circular orbits; initial altitude of 400 km for the pursuer, initial altitude of 1500 km for the evader): The orbital elements at t_0 and the related initial values of the state components are

$$P\begin{cases} a_{P,0}=6778.165\ \text{km} \\ e_{P,0}=0 \\ i_{P,0}=150\ \text{deg} \\ \Omega_{P,0}=170\ \text{deg} \\ \theta_{P,0}=0\ \text{deg} \end{cases} \Rightarrow \begin{cases} r_{P,0}=6778.165\ \text{km} \\ v_{P,0}=7.669\ \text{km/s} \\ \gamma_{P,0}=0\ \text{deg} \\ \xi_{P,0}=170\ \text{deg} \\ \phi_{P,0}=0\ \text{deg} \\ \zeta_{P,0}=150\ \text{deg} \end{cases}$$

$$E\begin{cases} a_{E,0}=7878.165\ \text{km} \\ e_{E,0}=0 \\ i_{E,0}=30\ \text{deg} \\ \Omega_{E,0}=30\ \text{deg} \\ \theta_{E,0}=10\ \text{deg} \end{cases} \Rightarrow \begin{cases} r_{E,0}=7878.165\ \text{km} \\ v_{E,0}=7.113\ \text{km/s} \\ \gamma_{E,0}=0\ \text{deg} \\ \xi_{E,0}=38.7\ \text{deg} \\ \phi_{E,0}=5.0\ \text{deg} \\ \zeta_{E,0}=29.6\ \text{deg} \end{cases} \tag{70}$$

The preprocessed and the optimal control laws are portrayed in Fig. 12. It emerges that the optimal and the preprocessed control angles are not far away from each other, that is, the GA preprocessing produced a reasonable guess solution, qualitatively similar to the optimal one. Figure 13 shows the saddle-point trajectories leading to interception, whereas some state components corresponding to the saddle-point solution are illustrated in Fig. 14: altitude, velocity, flight-path angle, and coazimuth angle.

Test case 5 (circular orbits; initial altitude of 400 km for the pursuer, initial altitude of 1500 km for the evader): The orbital

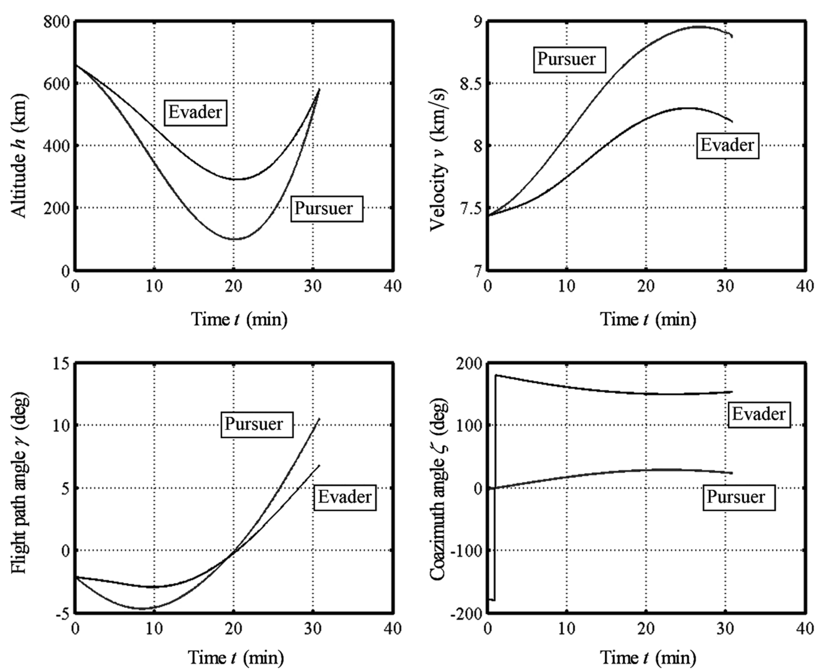
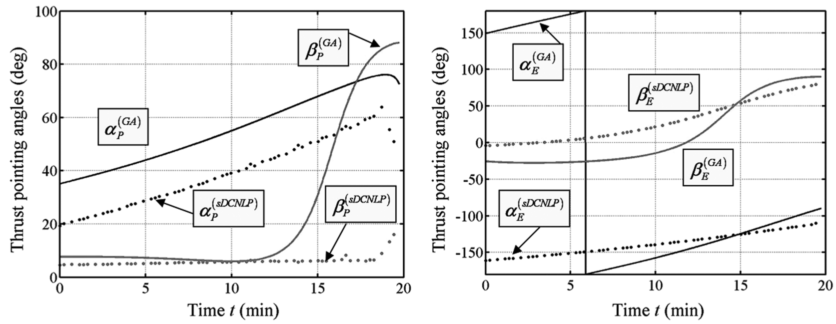


Fig. 11 SP solution for test case 3 for the two spacecraft: a) altitude, b) velocity, c) flight-path angle, and d) coazimuth angle.



a) b)

Fig. 12 Test case 4: preprocessed and optimal control time histories.

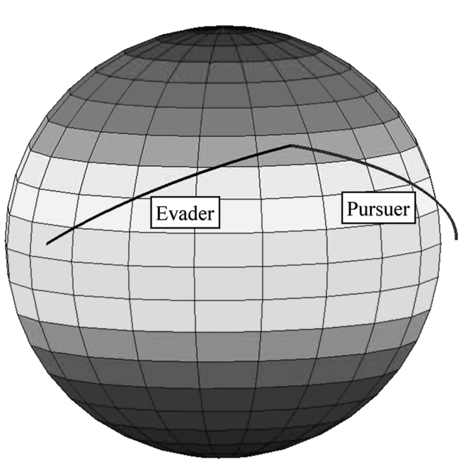


Fig. 13 Test case 4: SP trajectories of the two spacecraft.

elements at t_0 and the related initial values of the state components are

$$P\begin{cases} a_{P,0} = 6778.165 \text{ km} \\ e_{P,0} = 0 \\ i_{P,0} = 70 \text{ deg} \\ \Omega_{P,0} = 170 \text{ deg} \\ \theta_{P,0} = 0 \text{ deg} \end{cases} \Rightarrow \begin{cases} r_{P,0} = 6778.165 \text{ km} \\ v_{P,0} = 7.669 \text{ km/s} \\ \gamma_{P,0} = 0 \text{ deg} \\ \xi_{P,0} = 170 \text{ deg} \\ \phi_{P,0} = 0 \text{ deg} \\ \zeta_{P,0} = 70 \text{ deg} \end{cases}$$
$$E\begin{cases} a_{E,0} = 7878.165 \text{ km} \\ e_{E,0} = 0 \\ i_{E,0} = 10 \text{ deg} \\ \Omega_{E,0} = -150 \text{ deg} \\ \theta_{E,0} = 10 \text{ deg} \end{cases} \Rightarrow \begin{cases} r_{E,0} = 7878.165 \text{ km} \\ v_{E,0} = 7.113 \text{ km/s} \\ \gamma_{E,0} = 0 \text{ deg} \\ \xi_{E,0} = 219.9 \text{ deg} \\ \phi_{E,0} = 1.7 \text{ deg} \\ \zeta_{E,0} = 9.9 \text{ deg} \end{cases} \tag{71}$$

Figure 15 illustrates the preprocessed and the optimal control laws for both spacecraft. It emerges that there is no apparent similarity between the preprocessed and optimal saddle-point control angles. Nevertheless, starting from the preprocessed solution, the semi-DCNLP is able again to find an optimal saddle-point solution. A special feature distinguishes this fifth case: the evader optimal thrust direction is perpendicular to the orbital plane ($\beta_E \simeq 90$ deg) for the entire duration of the game. Figure 16 shows the saddle-point

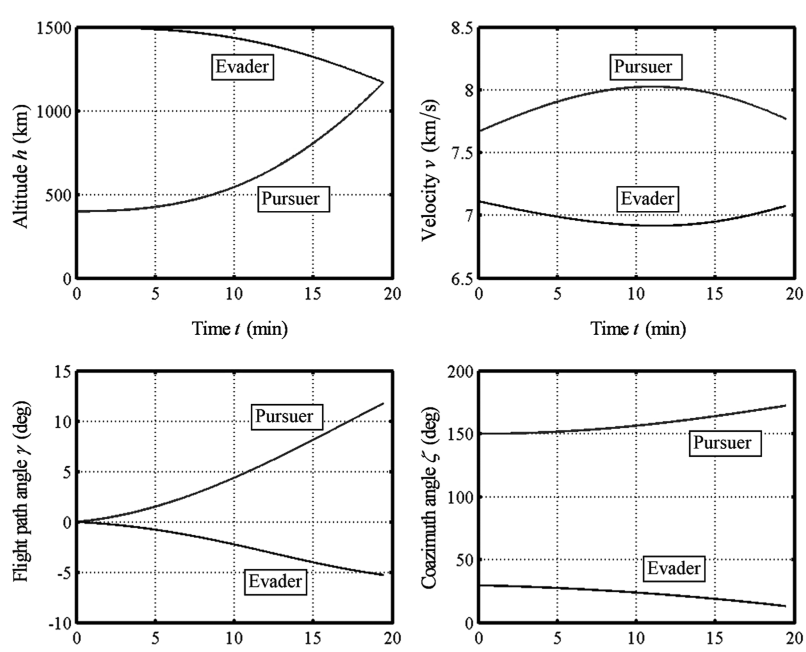
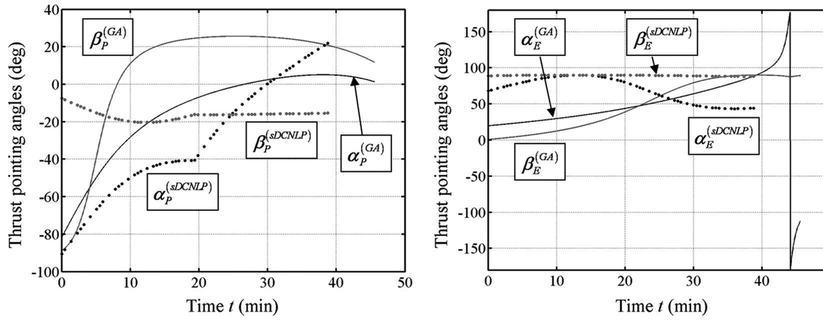


Fig. 14 SP solution for test case 4 for the two spacecraft: a) altitude, b) velocity, c) flight-path angle, and d) coazimuth angle.



a) b)

Fig. 15 Test case 5: preprocessed and optimal control time histories.

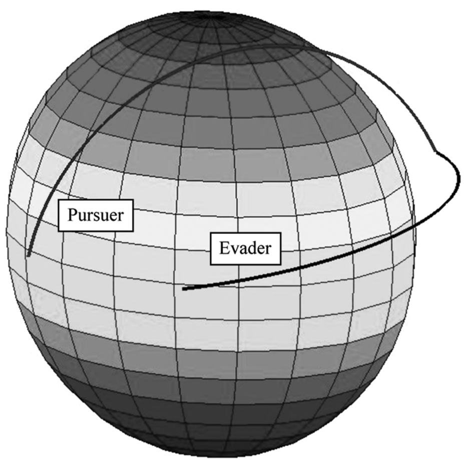


Fig. 16 Test case 5: SP trajectories of the two spacecraft.

trajectories leading to interception. Some state components corresponding to the saddle-point solution are portrayed in Fig. 17: altitude, velocity, flight-path angle, and coazimuth angle.

VIII. Conclusions

A challenging pursuit–evasion game, corresponding to the exo-atmospheric missile warhead interception problem, in which the warhead is able to maneuver modestly to avoid or delay interception, has been solved with a relatively new numerical method in which the continuous problem is converted into a nonlinear programming problem. The semidirect solution algorithm employs the analytical necessary conditions for one player but finds the strategy for the other player numerically. Because the nonlinear programming problem solver requires an initial guess of the solution vector of parameters, an approximate solution for the saddle-point equilibrium is found using genetic algorithms. In the five qualitatively different cases solved, corresponding to different initial conditions for the two spacecraft, the guess found by the genetic algorithm is occasionally found to have only a poor correspondence to the final, converged solution. The fact that a solution is found testifies to the robustness of this solution method.

Appendix: Relations Between State Components and Orbital Elements

State components $(r_i, v_i, \gamma_i, \xi_i, \phi_i, \zeta_i)$ and orbital elements $(a_i, e_i, i_i, \Omega_i, \omega_i, M_i)$ represent a set of six variables, which describe the dynamic state of each spacecraft. Once the orbital elements are known, the state components are unequivocally determined and vice versa. This appendix deals with the formal derivation of all the relationships needed to calculate the state components from the orbital elements.

First of all, the ranges for which angular variables are defined are the following:

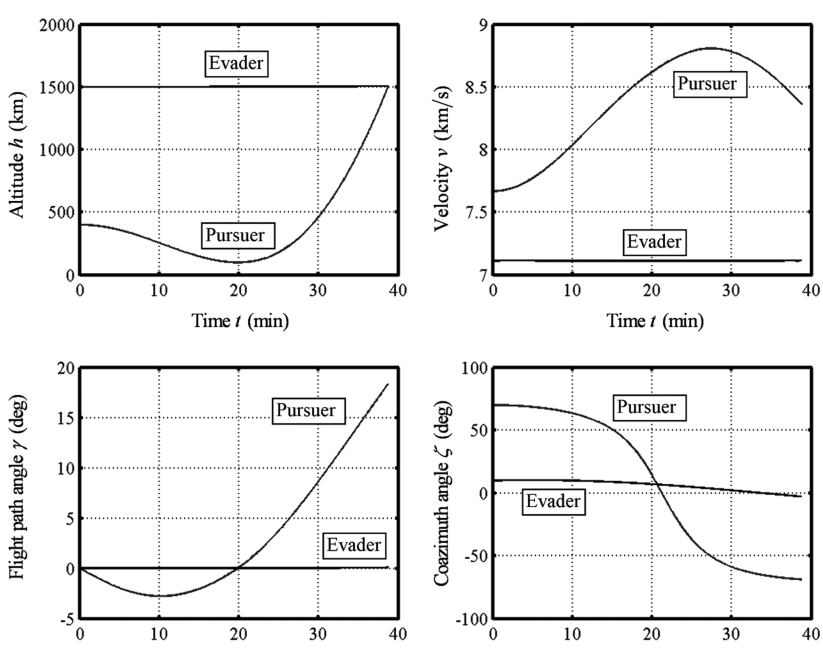


Fig. 17 SP solution for test case 5 for the two spacecraft: a) altitude, b) velocity, c) flight-path angle, and d) coazimuth angle.

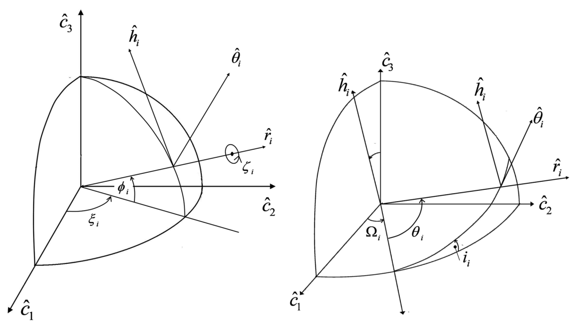


Fig. A1 Rotation angles relating the inertial frame with the orbital frame.

$$
\begin{aligned}
-\pi \le \xi_i < \pi, \qquad -(\pi/2) \le \phi_i \le \pi/2 \\
-\pi \le \zeta_i < \pi, \qquad -(\pi/2) \le \gamma_i \le \pi/2
\end{aligned} \tag{A1}
$$

$$
\begin{aligned}
-\pi \le \Omega_i < \pi, \qquad 0 \le i_i \le \pi \\
-\pi \le \omega_i < \pi, \qquad -\pi \le \theta_i < \pi
\end{aligned} \tag{A2}
$$

With reference to Fig. A1a, the Earth-centered inertial frame is identified by $(\hat{c}_1, \hat{c}_2, \hat{c}_3)$, where$\hat{c}_1$ is the vernal axis and $(\hat{c}_1, \hat{c}_2)$ belong to the Earth equatorial plane [30]. This frame and the orbital frame $(\hat{r}, \hat{\theta}, \hat{h}_i)$ (portrayed in Fig. A1a) are related through the rotation matrix $\mathbf{R}_A$, defined as follows:

$$
\begin{bmatrix} \hat{r}_i \\ \hat{\theta}_i \\ \hat{h}_i \end{bmatrix} = \begin{bmatrix} \cos\phi_i \cos\xi_i & \cos\phi_i \sin\xi_i & \sin\phi_i \\ -\sin\zeta_i \sin\phi_i \cos\xi_i - \cos\zeta_i \sin\xi_i & -\sin\zeta_i \sin\phi_i \sin\xi_i - \cos\zeta_i \cos\xi_i & \sin\zeta_i \cos\phi_i \\ -\cos\zeta_i \sin\phi_i \cos\xi_i + \sin\zeta_i \sin\xi_i & -\cos\zeta_i \sin\phi_i \sin\xi_i - \sin\zeta_i \cos\xi_i & \cos\zeta_i \cos\phi_i \end{bmatrix} \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \end{bmatrix} \triangleq \mathbf{R}_A \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \end{bmatrix} \tag{A3}
$$

The rotation $\mathbf{R}_A$ is written in terms of the angles ξ_i, ϕ_i, and ζ_i, and can be described as the result of three successive rotations performed in the following sequence: the first (counterclockwise by an angle ξ_i) about axis 3, the second (clockwise by an angle ϕ_i) about axis 2, and the third (counterclockwise by an angle ζ_i) about axis 1. Similarly, with reference to Fig. A1b, the orbital frame $(\hat{r}, \hat{\theta}, \hat{h}_i)$ can be obtained from the inertial frame $(\hat{c}_1, \hat{c}_2, \hat{c}_3)$ through an alternative rotation $\mathbf{R}_B$, written in terms of the angles Ω_i, i_i, and θ_i, where $\theta_i (\triangleq \omega_i + f_i)$ is the argument of latitude:

$$
\begin{bmatrix} \hat{r}_i \\ \hat{\theta}_i \\ \hat{h}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i \cos\Omega_i - \sin\theta_i \cos i_i \sin\Omega_i & \cos\theta_i \sin\Omega_i + \sin\theta_i \cos i_i \cos\Omega_i & \sin\theta_i \sin i_i \\ -\sin\theta_i \cos\Omega_i - \cos\theta_i \cos i_i \sin\Omega_i & -\sin\theta_i \sin\Omega_i + \cos\theta_i \cos i_i \cos\Omega_i & \cos\theta_i \sin i_i \\ \sin i_i \sin\Omega_i & -\sin i_i \cos\Omega_i & \cos i_i \end{bmatrix} \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \end{bmatrix} \triangleq \mathbf{R}_B \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \hat{c}_3 \end{bmatrix} \tag{A4}
$$

This rotation can be described as the result of three successive rotations performed in the following sequence: the first (counterclockwise by an angle Ω_i) about axis 3, the second (counterclockwise by an angle i_i) about axis 1, the third (counterclockwise by an angle θ_i) about axis 3. The two matrices $\mathbf{R}_A$ and $\mathbf{R}_B$ must coincide, and this fact implies that the corresponding elements must be identical. As a result,

$$
\begin{aligned}
\cos\phi_i \cos\xi_i = \cos\theta_i \cos\Omega_i - \sin\theta_i \cos i_i \sin\Omega_i \quad \text{and} \\
\cos\phi_i \sin\xi_i = \cos\theta_i \sin\Omega_i + \sin\theta_i \cos i_i \cos\Omega_i
\end{aligned} \tag{A5}
$$

$$
\sin\phi_i = \sin\theta_i \sin i_i \tag{A6}
$$

$$
\cos\theta_i \sin i_i = \sin\zeta_i \cos\phi_i \tag{A7}
$$

$$
\cos i_i = \cos\zeta_i \cos\phi_i \tag{A8}
$$

If the orbital elements $(a_i, e_i, i_i, \Omega_i, \omega_i, M_i)$ are specified, the state components $(r_i, v_i, \gamma_i, \xi_i, \phi_i, \zeta_i)$ can be easily deduced. First of all, the numerical solution of Kepler's equation [30] yields the eccentric anomaly E_i from the mean anomaly $M_i (= E_i - e_i \sin E_i)$ and the true anomaly f_i, which is directly related to E_i.

Then, the polar equation [30] of elliptic orbits yields the radius r_i, whereas the vis-viva equation [30] allows the velocity v_i to be obtained. The flight-path angle γ_i can be deduced from the radial component of velocity, v_{ri}:

$$
v_{ri} = \sqrt{\frac{\mu_E}{a_i(1 - e_i^2)}} e_i \sin f_i = v_i \sin\gamma_i \to \gamma_i \tag{A9}
$$

Lastly, the three angles (ξ_i, ϕ_i, ζ_i) can be calculated from $(\Omega_i, i_i, \theta_i)$ with the relationships (A5–A8).

References

[1] Isaacs, R., *Games of Pursuit*, Rand Corp., Santa Monica, CA, 1951.

[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*, Society for Industrial and Applied Mathematics, Philadelphia, 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. doi:10.2514/3.7399

[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. doi:10.2514/3.20589

[7] Roberts, D. A., and Montgomery, R. C., "Development and Application of a Gradient Method for Solving Differential Games," NASA Langley Research Center 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., 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. doi:10.2514/3.19738

[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.
doi:10.2514/2.4586

[13] Anderson, G. M., and Grazier, V. W., "A Closed-Form Solution for the Barrier in Pursuit-Evasion Problems Between Two Low Thrust Orbital Spacecraft and Its Application," *Aerospace Sciences Meeting*, AIAA, New York, Jan. 1975.

[14] Kelley, H. J., Cliff, E. M., and Lutze, F. H., "Pursuit–Evasion in Orbit," *Journal of the Astronautical Sciences*, Vol. 29, No. 3, 1981, pp. 277–288.

[15] Zhang, Q.-H., Sun, Y., Huang, M.-M., and Duan, G.-R., "Pursuit–Evasion Barrier of Two Spacecraft Under Minute Continuous Radial Thrust in Coplanar Orbit," *Control and Decision*, Vol. 22, No. 5, 2007, pp. 530–534.

[16] Vasile, M., and Bernelli Zazzera, F., "Direct Multiphase Optimisation of Multiobjective Trajectory Design Problems," American Astronautical Society Paper 02-205, Jan. 2002.

[17] Menon, P. K. A., and Calise, A. J., "Interception, Evasion, Rendezvous and Velocity-to-Be-Gained Guidance for Spacecraft," *AIAA Guidance, Navigation and Control Conference*, AIAA, New York, Aug. 1987; AIAA, Vol. 1, A87-50401 22-08, 1987, pp. 334–341.

[18] Menon, P. K. A., and Calise, A. J., "Guidance Laws for Spacecraft Pursuit-Evasion and Rendezvous," *AIAA Guidance, Navigation and Control Conference*, AIAA, Washington, D.C., Aug. 1988; AIAA, Vol. 2, A88-50160 21-08, 1988, pp. 688–697.

[19] 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

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

[21] Conway, B. A., and Horie, K., "A New Collocation-Based Method for Solving Pursuit–Evasion (Differential Games) Problems," American Astronautical Society Paper 01-445, Aug. 2001.

[22] 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.
doi:10.2514/3.20223

[23] 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.
doi:10.2514/3.20934

[24] 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.
doi:10.2514/3.21662

[25] Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., User's Guide for NPSOL (Ver. 4.0): A Fortran Package for Nonlinear Programming, SOL 86-2, Stanford University, Palo Alto, CA, 1986.

[26] Pontani, M., and Conway, B. A., "Optimal Interception of Evasive ICBM Warheads: Numerical Solution of the Differential Game," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 4, 2008, pp. 1111–1122.
doi:10.2514/1.30893

[27] 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

[28] 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, March 2003.

[29] 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.

[30] Prussing, J. E., and Conway, B. A., *Orbital Mechanics*, Oxford Univ. Press, New York, 1993, pp. 15, 20, 27, 46–49.