A NEW COLLOCATION-BASED METHOD FOR SOLVING PURSUIT/EVASION (DIFFERENTIAL GAMES) PROBLEMS

B. A. Conway* and K. Horie†

A new numerical method for solving zero-sum two-person differential games is developed. The method, which we call semi-direct collocation with nonlinear programming, incorporates necessary conditions for saddle-point trajectories into the direct collocation with nonlinear programming method, and finds saddle-point trajectories and associated control histories. The method is more straightforward and robust than methods usually used to solve problems in differential games, such as shooting methods or differential dynamic programming. An example problem, the well-known dolichobrachistochrone, is solved to verify suitability of the method for a realistic dynamic problem. A second, more complex problem of spacecraft interception of an optimally evasive target is also successfully solved.

INTRODUCTION

In general, a path optimization problem can be categorized as an optimal control problem or a differential game problem. The optimal control problem considers only one player and has been successfully applied to a variety of applications for several decades. On the other hand, a problem such as air combat is most accurately modeled using two competitive players. In that case a path optimization problem often becomes a zero-sum two-person differential game. A zero-sum two-person differential game "mini-maximizes" a cost function, which means that one of two competitive players minimizes a given cost function while another maximizes the cost function. It was originally formulated by Isaacs[1]. Bryson and Ho[2] researched the same problem as an extension of an optimal control problem.

A variety of zero-sum two-person differential game problems have been solved analytically using simplified dynamics. One of the well-developed problems is air combat analysis between a low-speed highly maneuverable evader and high-speed less-

* Professor, Department of Aeronautical and Astronautical Engineering, 306 Talbot Laboratory, University of Illinois, Urbana, Illinois 61801-2935. E-mail: bconway@uiuc.edu.

† Graduate Student, Department of Aeronautical and Astronautical Engineering, 306 Talbot Laboratory, University of Illinois, Urbana, Illinois 61801.

maneuverable pursuer. Breakwell and Merz [3] solved the problem in two-dimensional space on the basis of the "homicidal chauffeur game". Segal and Miloh [4] succeeded in extending the study of Breakwell and Merz to three-dimensional space. One disadvantage of these analytical approaches is the quite simplified dynamics which must be assumed. A problem solved by Guelman, Shinar and Green [5], which used relatively realistic dynamics, is probably the most ambitious problem that may be solved analytically.

An answer to this disadvantage is to formulate a pursuit/evasion problem using realistic dynamics directly and solve it numerically. However, only a modest number of studies have been done using this approach. Hillberg and Järmark [6] solved an air combat maneuvering problem in the horizontal plane with realistic drag and thrust data using differential dynamic programming. A pursuit-evasion problem between a missile and an aircraft has been solved using a multiple shooting method for the two-dimensional case by Breitner, Grimm and Pesch [7-9] and for the three-dimensional case by Lachner, Breitner and Pesch [10].

It is obviously required to apply a numerical method to solve the differential game with realistic dynamics because realistic dynamics are usually complicated and nonlinear. As stated previously, differential dynamic programming and multiple shooting have succeeded in solving several problems. However, differential dynamic programming and multiple shooting methods are often too sensitive to get convergent solutions.

The direct collocation with nonlinear programming method (DCNLP). which was originally proposed by Hargraves and Paris [11], has shown robustness in solving many optimal control problems [12-16]. Unfortunately, DCNLP cannot solve the differential game problem directly because the most suitable nonlinear programming solvers for DCNLP, e.g. NPSOL [17] or similar programs, can only minimize or maximize a single cost function. However, Raivio and Ethamo [18] recently solved a pursuit-evasion problem using DCNLP. They were able to accomplish this by decomposing a pursuit-evasion problem into two optimal control subproblems, i.e. they solve a sequence of subproblems with a pre-specified capture point and pre-specified trajectory of one player and then iteratively adjust the pre-specified capture point to improve the result.

In this research, we develop a new, robust numerical method for zero-sum two-person differential games. The new method is developed employing DCNLP, but in a qualitatively different way from that of Raivio and Ethamo [18], with the hope that it will share the robustness of the DCNLP method. To solve the problem, some optimality conditions are determined analytically and incorporated into the DCNLP problem. This almost invariably requires the explicit appearance of some of the system costate variables, something which is ordinarily avoided in the use of a "direct" solution. Thus, we refer to the method as "semi-direct" collocation with nonlinear programming (semi-DCNLP).

In this research, the method is formulated for a pursuit-evasion problem with inertial coordinates. Then, the semi-DCNLP method is applied to the well-known dolichobrachistochrone problem. The results obtained are compared with the results using a shooting method, to verify its feasibility for solving a zero-sum two-person differential game problem. Finally, the semi-DCNLP method is applied to a realistic dynamic problem of moderate size, the interception by a spacecraft of an optimally evasive target spacecraft, to better learn its suitability for solving practical differential game problems.

METHOD

Problem Statement

In a zero-sum two-person differential game, two competitive sets of control variables, u_p and u_e, drive a dynamic system. In this study, u_p and u_e control state variables for a pursuer and an evader, x_p and x_e, separately. Then, the (independent) equations of motion may be written in the following form:

$$\dot{x}_p = f_p(x_p, u_p, t) \tag{1}$$
$$\dot{x}_e = f_e(x_e, u_e, t) \tag{2}$$

with initial conditions:

$$x_p(t_0) = x_{p0} \tag{3}$$
$$x_e(t_0) = x_{e0} \tag{4}$$

terminal constraints which are functions of the states at the final time and possibly the final time:

$$\psi(x_p(t_f), x_e(t_f), t_f) = 0 \tag{5}$$

where t is time, t_0 is initial time and t_f is the terminal time of a problem.
It is assumed that some control variables are bounded as:

$$u_{p,l} \le u_p \le u_{p,u} \tag{6}$$
$$u_{e,l} \le u_e \le u_{e,u} \tag{7}$$

Path constraints are not considered in this study.

A problem of Mayer type is considered in this research. Then, the cost function for the problem is given as a function of the state variables and terminal time in the following form:

$$J(x_p, x_e, u_p, u_e, t) = \phi[x_p(t_f), x_e(t_f), t_f] \tag{8}$$

The feedback strategies, γ_p and γ_E, are introduced to determine the control variables, u_p and u_e, as a function of state variables, i.e., $u_p(t)=u_P(t,x_p,x_e)$, $u_e(t)=u_E(t,x_p,x_e)$. Using γ_p and γ_E the value of the game, if it exists, is defined as:

$$V = \min_{\gamma_P} \max_{\gamma_E} J = \max_{\gamma_E} \min_{\gamma_P} J \tag{9}$$

The existence of the value of the game is assumed in the following discussion.

The open-loop representation of optimal feedback strategy, i.e., the strategy along the optimal path as a function only of initial states, is defined as $u_p^*(t)= \gamma_p(t, x_p,x_e)$, $u_e^*(t)= \gamma_E(t, x_p,x_e)$, where $u_p^*(t)= \gamma_p(t, x_p,x_e)$ and $u_e^*(t)= \gamma_E(t,x_p,x_e)$ are considered to be satisfied with (9). Then, using initial states from equation (3) and (4), the value of the game is expressed as:

$$V = J(x_p, x_e, u_p^*, u_e^*, t) \tag{10}$$

A feedback saddle-point trajectory is obtained using an open-loop representation of the optimal feedback strategy, u_p^* and u_e^*, under constraints (1)-(7).

Necessary Condition for Saddle-Point Trajectory

Basar and Olsder [19] provide a set of necessary conditions for an open-loop representation of a feedback saddle-point trajectory. Here, the conditions are modified for the specific situation where the governing equations of the pursuer and the evader are independent. At first, a Hamiltonian and a set of functions of terminal conditions are introduced, using the definitions of eqns. (1), (2), (5) and (8) as:

$$H = \lambda_p^T f_p + \lambda_e^T f_e \tag{11}$$

$$\Phi = \phi + \nu^T \psi \tag{12}$$

where λ_p and λ_e are adjoint variables and ν is a set of Lagrange multipliers. As the existence of the value of the game is assumed, the Hamiltonian is separable.

Using the Hamiltonian, adjoint equations are determined as:

$$\dot{\lambda}_p = -\left(\frac{\partial H}{\partial x_p}\right) = -\left(\frac{\partial f_p}{\partial x_p}\right)^T \lambda_p \tag{13}$$

$$\dot{\lambda}_e = -\left(\frac{\partial H}{\partial x_e}\right) = -\left(\frac{\partial f_e}{\partial x_e}\right)^T \lambda_e \tag{14}$$

$$u_p = \arg\min_{u_p} H = \arg\min_{u_p} (\lambda_p^T f_p) \tag{15}$$

$$u_e = \arg\min_{u_e} H = \arg\min_{u_e} (\lambda_e^T f_e) \tag{16}$$

$$\lambda_p(t_f) = \left(\frac{\partial \Phi}{\partial x_p}\right)^T = \left(\frac{\partial \phi}{\partial x_p} + \nu^T \frac{\partial \psi}{\partial x_p}\right)^T \tag{17}$$

$$\lambda_e(t_f) = \left(\frac{\partial \Phi}{\partial x_e}\right)^T = \left(\frac{\partial \phi}{\partial x_e} + \nu^T \frac{\partial \psi}{\partial x_e}\right)^T \tag{18}$$

$$\left[H + \frac{\partial \Phi}{\partial t}\right]_{t=t_f} = 0 \tag{19}$$

Then, (1), (2), (13) and (14) constitute a two-point boundary value problem (TPBVP) with the initial and terminal conditions, (3) - (5) and (17) - (19) and controls satisfying (6), (7), (15) and (16). However, it is normally very difficult to solve this TPBVP if the problem is large, i.e. has many states and/or controls, or has strong nonlinearity, which often obtains for problems including realistic dynamics.

Direct Collocation with Nonlinear Programming

The DCNLP method is a numerical method for solving optimal control problems and may be categorized as a direct method, i.e. one which doesn't use the optimality condition explicitly and which does not employ adjoint variables. It uses the collocation method [11, 12, 14] and nonlinear programming to solve the following generalized optimal control problem:

$$\min_{u} J(x,u,t) \tag{20}$$

subject to

$$\dot{x} = f(x,u,t) \tag{21}$$

$$\chi(x(t_0),t_0) = 0 \tag{22}$$

$$\psi(x(t_f),t_f) = 0 \tag{23}$$

$$g(x,u,t) \leq 0 \tag{24}$$

The DCNLP parameterizes state and control variables by discretizing them in time and finds an optimal set in a parameter space using nonlinear programming. The parameter space consists of the discretized state and control variables and possibly additional parameters such as the total time, T, and is constrained by (21) - (24). However, (21) cannot constrain the parameter space directly; instead (21) is approximated using algebraic equations to relate the state variables at adjacent discretization times.

The collocation method solves the boundary value problem by approximating the solution of the differential equation (21) as some function. In the collocation method in the DCNLP, the time history of the trajectory is divided into "segments". Within each segment the solution of the governing equations (1)-(2) is approximated as a polynomial function in time. At discrete points, which are often called collocation points, the value and slope of the piecewise polynomial (approximating the solution for a state variable) are required to be the same as that of the corresponding state variable in the parameter space and the state variable rate of change, respectively. A cubic polynomial is commonly used for this purpose (Ref. 12); this corresponds to using Simpson's integration rule between discretization times. However, the polynomial can approximate the solution more accurately by selecting optimal collocation points rather than using equally-spaced collocation points. Herman and Conway proposed using the fifth-degree Gauss-Lobatto quadrature rule as the basis for selecting optimal collocation points in each segment [14]. This corresponds to approximating each state variable with a fifth-degree polynomial in each segment. This choice yields greater accuracy of the implicit integration for a given discretization and is used in obtaining the results in the following sections.

The nonlinear programming problem is then to satisfy (20) in given parameter space under constraints of algebraic form of (21) for each segment at each collocation point and satisfying initial and terminal conditions (22), (23).

The reader unfamiliar with the DCNLP method is referred to references (11, 12, & 14) where it is described in much greater detail.

Semi-Direct Collocation with Nonlinear Programming

We noted previously that the TPBVP formed by the necessary conditions for a saddle point is often difficult to solve. Thus, a new solution method based on DCNLP is proposed as an alternative to solving the TPBVP. The method is expected to maintain the robust characteristics of DCNLP. In this method, the optimality condition (15), associated adjoint equations, (13), and the following equations, (25) are incorporated into the DCNLP formulation.

To form what we refer to as the "extended" problem, consider the situation that obtains if we treat the costate λ_p as a "state" variable, i.e. if we add eqn. (13) to the system governing equations (1) and (2). Let

$$\psi_{EXT}(x_p, x_e, \lambda_p, t_f) = 0 \tag{25}$$

be an additional terminal constraint for the extended problem, where (25) are boundary conditions derived from (17) and (18) which are not a function of λ_p or ν.

The control u_p, for one of the players, obtained from the analytical necessary condition (15), minimizes the cost function. Then, the original problem can be converted to:

$$V = \max_{u_e} J \qquad \text{subject to (1) - (7), (13), (15), (17) and (25)} \tag{26}$$

The problem described by equations (26) is now a one-sided optimization with constraints consisting of differential equations and algebraic equations; i.e. it is a problem of the form of eqns. (20) – (24). Thus it is amenable to solution by the DCNLP method. Note that it is still a "direct" method because, while the costate variable λ_p appears, it is now acting as if it is a state variable of the problem.

It is necessary to evaluate the characteristics of the solution of system (26). We apply the calculus of variation and Pontryagin principle. First, a Hamiltonian and a parameter at terminal conditions for (26) are introduced as:

$$\begin{aligned} H_{EXT} &= \lambda_{Ep}{}^T f_p + \lambda_{Ee}{}^T f_e + \lambda_{E\lambda_p}{}^T \left(-\left(\frac{\partial f_p}{\partial x_p} \right)^T \lambda_p \right) \\ &= \lambda_{Ep}{}^T f_p + \lambda_{Ee}{}^T f_e - \lambda_{E\lambda_p}{}^T \left(\frac{\partial f_p}{\partial x_p} \right)^T \lambda_p \end{aligned} \tag{27}$$

$$\Phi_{EXT} = \phi(x_p, x_e, t_f) + \nu_{E_1}{}^T \psi + \nu_{E_2}{}^T \psi_{EXT} \tag{28}$$

Note again that in the Hamiltonian for the extended problem (27) the original costate variable λ_p is treated as a state variable of the problem. The costate (or adjoint) variables for the extended problem are $\lambda_{Ee}, \lambda_{Ep}$, and $\lambda_{E\lambda p}$. The control variables, u_e, and associated adjoint variables must satisfy with following relationships, where u_p is a function of x_p and λ_p:

$$\begin{aligned}
\dot{\lambda}_{Ep} &= -\left(\frac{\partial H_{EXT}}{\partial x_p}\right) = -\left(\frac{\partial f_p}{\partial x_p}\right)^T \lambda_{Ep} + \frac{\partial}{\partial x_p}\left[\lambda_{E\lambda_p}{}^T\left(\frac{\partial f_p}{\partial x_p}\right)^T \lambda_p\right]^T \\
&= -\left(\frac{\partial f_p}{\partial x_p}\right)^T \lambda_{Ep} + \frac{\partial}{\partial x_p}\left(\lambda_p{}^T \frac{\partial f_p}{\partial x_p}\right)\lambda_{E\lambda_p} + \frac{\partial u_p}{\partial x_p}\frac{\partial}{\partial u_p}\left(\lambda_p{}^T \frac{\partial f_p}{\partial x_p}\right)\lambda_{E\lambda_p} \\
&= -\left(\frac{\partial f_p}{\partial x_p}\right)^T \lambda_{Ep} + \frac{\partial}{\partial x_p}\left(\lambda_p{}^T \frac{\partial f_p}{\partial x_p}\right)\lambda_{E\lambda_p}
\end{aligned} \tag{29}$$

where the last term vanishes because of condition (15).

$$\dot{\lambda}_{Ee} = -\left(\frac{\partial H_{EXT}}{\partial x_e}\right)^T = -\left(\frac{\partial f_e}{\partial x_e}\right)^T \lambda_{Ee} \tag{30}$$

$$\begin{aligned}
\dot{\lambda}_{E\lambda_p} &= -\left(\frac{\partial H_{EXT}}{\partial \lambda_p}\right) = \frac{\partial f_p}{\partial x_p}\lambda_{E\lambda_p} + \lambda_p{}^T \frac{\partial}{\partial \lambda_p}\left(\frac{\partial f_p}{\partial x_p}\lambda_{E\lambda_p}\right) - \left(\frac{\partial f_p}{\partial \lambda_p}\right)^T \lambda_{Ep} \\
&= \frac{\partial f_p}{\partial x_p}\lambda_{E\lambda_p} + \lambda_p{}^T \frac{\partial u_p}{\partial \lambda_p}\frac{\partial}{\partial u_p}\left(\frac{\partial f_p}{\partial x_p}\lambda_{E\lambda_p}\right) - \left(\frac{\partial f_p}{\partial u_p}\frac{\partial u_p}{\partial \lambda_p}\right)^T \lambda_{Ep} \\
&= \frac{\partial f_p}{\partial x_p}\lambda_{E\lambda_p} + \frac{\partial u_p}{\partial \lambda_p}\frac{\partial}{\partial x_p}\left(\lambda_p{}^T \frac{\partial f_p}{\partial u_p}\right)^T \lambda_{E\lambda_p} - \left(\frac{\partial f_p}{\partial u_p}\frac{\partial u_p}{\partial \lambda_p}\right)^T \lambda_{Ep} \\
&= \frac{\partial f_p}{\partial x_p}\lambda_{E\lambda_p} - \left(\frac{\partial u_p}{\partial \lambda_p}\right)^T\left(\frac{\partial f_p}{\partial u_p}\right)^T \lambda_{Ep}
\end{aligned} \tag{31}$$

$$u_e = \arg\max\left(\lambda_{Ee}{}^T f_e\right) \tag{32}$$

$$\lambda_{Ep}(t_f) = \frac{\partial \phi}{\partial x_p} + \nu_{E_1}{}^T \frac{\partial \psi}{\partial x_p} + \nu_{E_2}{}^T \frac{\partial \psi_{EXT}}{\partial x_p} \tag{33}$$

$$\lambda_{Ee}(t_f) = \frac{\partial \phi}{\partial x_e} + \nu_{E_1}{}^T \frac{\partial \psi}{\partial x_e} + \nu_{E_2}{}^T \frac{\partial \psi_{EXT}}{\partial x_e} \tag{34}$$

$$\lambda_{E\lambda_p}(t_f) = \nu_{E_2}{}^T \frac{\partial \psi_{EXT}}{\partial \lambda_p} \tag{35}$$

$$\lambda_{E\lambda_p}(t_0) = 0 \tag{36}$$

$$\left[H_{EXT} + \frac{\partial \Phi_{EXT}}{\partial t}\right]_{t=t_f} = \left[H + \frac{\partial \Phi}{\partial t} - \lambda_{E\lambda_p}{}^T \left(\frac{\partial f_p}{\partial x_p}\right)^T \lambda_p + \nu_{E_2}{}^T \frac{\partial \psi_{EXT}}{\partial t_f}\right]_{t=t_f} = 0 \tag{37}$$

Necessary conditions for optimality for the problem described by eqns. (26) are the two-point boundary value problem described by (1) - (7), (13), (15), (25) and (29) - (37). By inspection, the solution of the TPBVP is satisfied with the necessary conditions for an open-loop representation of a feedback saddle-point trajectory when the following conditions are satisfied:

$$\lambda_{Ep} = \lambda_p \tag{38}$$

$$\nu_{E_2} = 0 \tag{39}$$

Variables λ_{Ep} and ν_{E2} can be obtained from the output of the nonlinear programming problem solver NPSOL. Then, it may be determined whether the DCNLP-based method provides the solution satisfying the necessary conditions by checking λ_{Ep} and ν_{E2}.

Before discussing example problem solutions, the characteristics and advantages of the DCNLP-based method should be noted. Equation (26) includes optimality conditions (15) with respect to the control variables of one player, u_p (i.e.(15) will determine optimal control u_p for the pursuer as a function of the "states" x_p and λ_p of the pursuer.) On the other hand, (26) does not include optimality conditions with respect to the control variables, u_e, of the second player (i.e. u_e, will be found numerically, by the NLP problem solver.) Thus the method may be regarded as indirect with respect to control variables u_p, and direct with respect to control variables u_e. Therefore, the method is referred to as semi-direct collocation with nonlinear programming (semi-DCNLP) in this research. In addition to having the robustness of DCNLP, the problem size will be reduced in comparison to the original TPBVP because only adjoint equations (13), i.e. only those required for determining control u_p algebraically, are required.

EXAMPLES

Dolichobrachistochrone

The famous brachistochrone problem proposed by John Bernoulli is the problem of finding a minimum-time descent trajectory of a mass from an initial point to a terminal point under gravity by controlling the shape of the frictionless path. The dolichobrachistochrone problem, which was proposed by Isaacs [20], adds a player who acts delay the progress of the mass toward the terminal surface, i.e., the additional control tries to maximize the descent time. The problem is illustrated in Fig. 1, though with the "descent" direction and the gravitational acceleration assumed upward.

A set of equations of motion and terminal conditions is given as;

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \sqrt{x_2}\cos\alpha + (\beta+1)/2 \\ \sqrt{x_2}\sin\alpha + (\beta-1)/2 \end{bmatrix} \tag{40}$$

$$x_1(t_f) = 0 \tag{41}$$

where (x_1, x_2) is the position of the mass, α (a control) is the instantaneous direction of the motion, chosen to minimize descent time, and β (a control) determines the magnitude of the horizontal and vertical components of the acceleration opposing gravity, and is chosen (by the "second player") to maximize the travel time from a given initial position of the mass, (x_{10}, x_{20}).

Control variable β is bounded as:

$$-1 \le \beta \le 1 \tag{42}$$

The cost function is the terminal time of the problem:

$$J = t_f \tag{43}$$

Then, the Hamiltonian corresponding to (11) becomes:

$$H = -\lambda_{x_1}\left[\sqrt{x_2}\cos\alpha + (\beta+1)/2\right] + \lambda_{x_2}\left[\sqrt{x_2}\sin\alpha + (\beta-1)/2\right] \tag{44}$$

The optimality conditions with respect to α are:

$$H_\alpha = -\lambda_{x_1}\sqrt{x_2}\sin\alpha + \lambda_{x_2}\sqrt{x_2}\cos\alpha = 0 \tag{45}$$

$$H_{\alpha\alpha} = -\lambda_{x_1}\sqrt{x_2}\cos\alpha - \lambda_{x_2}\sqrt{x_2}\sin\alpha \ge 0 \tag{46}$$

The equations for the adjoint variables are:

$$\dot{\lambda}_{x_1} = 0 \qquad \text{with} \qquad \lambda_{x_1}(t_f) = \nu_{x_1} \tag{47}$$

$$\dot{\lambda}_{x_2} = -(\lambda_{x_1}\cos\alpha + \lambda_{x_2}\sin\alpha)/2\sqrt{x_2} \quad \text{with} \quad \lambda_{x_2}(t_f) = 0 \tag{48}$$

From (47), the adjoint variable λ_{x1} is constant. To simplify the analysis, eqn. (48) is divided by λ_{x1} and becomes:

$$\dot{\lambda}_{x_2}/\lambda_{x_1} = -\left[\cos\alpha + (\lambda_{x_2}/\lambda_{x_1})\sin\alpha\right]/2\sqrt{x_2} \quad \text{with}$$

$$\lambda_{x_2}(t_f)/\lambda_{x_1}(t_f) = 0 \tag{49}$$

Finally, the problem is considered as being of the form of system (26); i.e. the objective function (43) is maximized under constraints, (40) - (42), (45), (46) and (49). The control α is determined using the optimality condition (45) with the required adjoint variable satisfying eqn. (49); the control β is determined, at the collocation points, by the NLP problem solver.

The dolichobrachistochrone problem is solved for three different initial conditions; $(x_{10}, x_{20}) = (0.5, 1.5)$, $(1.5, 1.5)$ and $(3.5, 1.5)$. The discretization is constructed using 20 segments. A solution has also been found for this relatively simple problem using a shooting method, for the purpose of comparing the results. Saddle-point trajectories obtained using both the collocation with nonlinear programming and shooting methods are shown in Fig. 2. For $(x_{10}, x_{20}) = (0.5, 1.5)$ and $(1.5, 1.5)$ the trajectory found using the collocation with nonlinear programming corresponds well to the trajectories found using the shooting method. On the other hand, a small difference between the trajectory found using the semi-DCNLP method and that using the shooting method are observed for the starting position $(x_{10},x_{20})=(3.5, 1.5)$. Figures 3 and 4 show the time histories of the control variables α and β, for both solutions, for the initial condition $(x_{10},x_{20})=(3.5, 1.5)$. These time histories are virtually indistinguishable with the exception of the switching time for the discontinuous control, β, as shown in Fig. 4. This difference is quite small and is an artifact of the discretization; the optimal switching time does not coincide exactly with a collocation point. This difference could be reduced by using a finer discretization, i.e. more than 20 segments, or, now that the character of the optimal control is known, by adding another variable, an optimal switching time for β, (which time would then be made a collocation point) to be determined by the NLP problem solver.

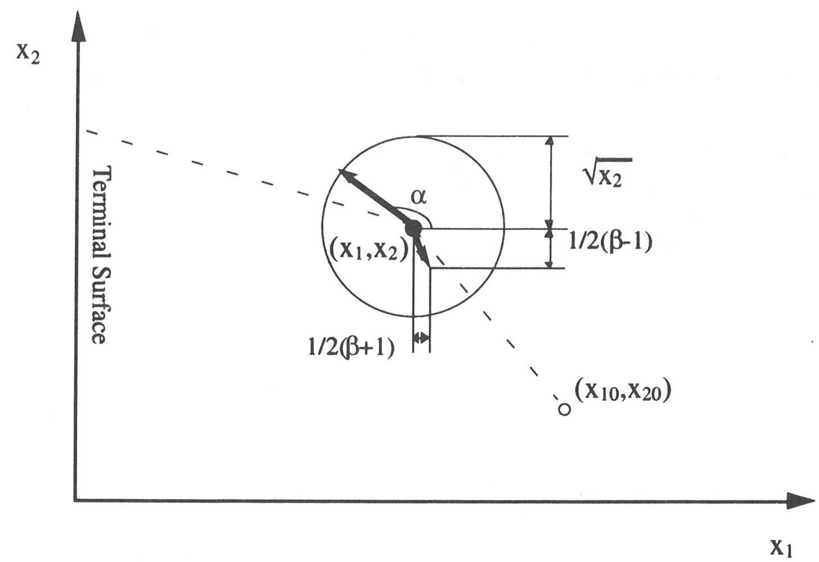

Figure 1. Cartoon of the Dolichobrachistochrone problem.
(Gravity acts upward.)

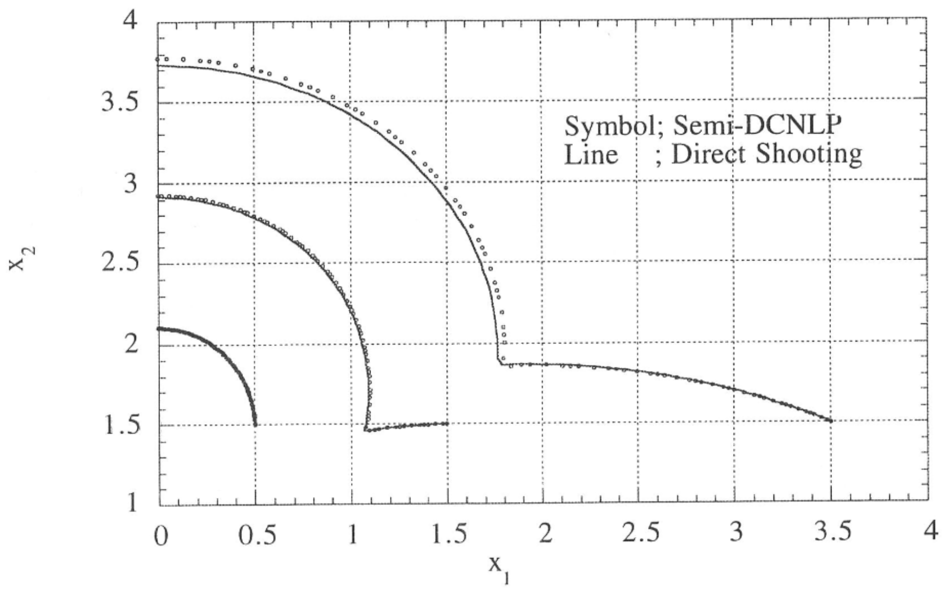

Figure 2. Trajectories for the Dolichobrachistochrone Problem for Various Starting Positions

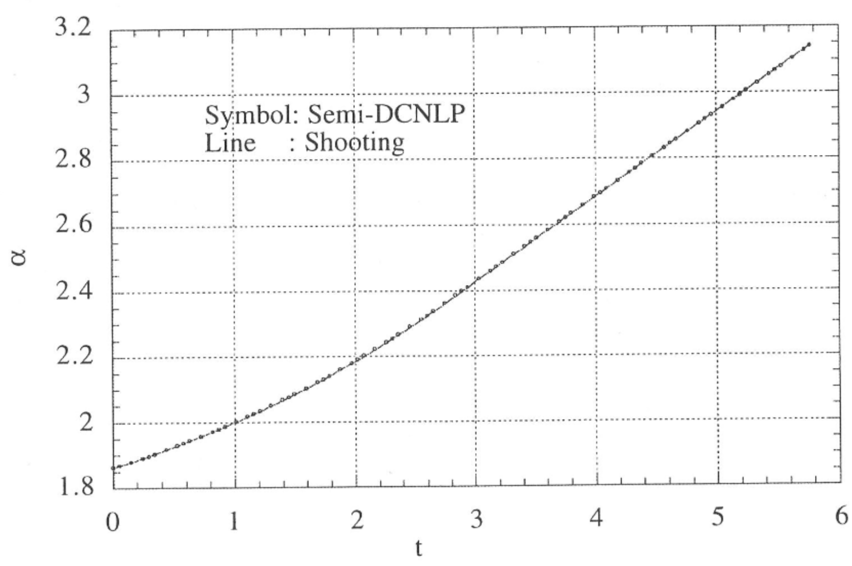

Figure 3. Time History of Control α for the Initial Condition (x_{10},x_{20})=(3.5, 1.5)

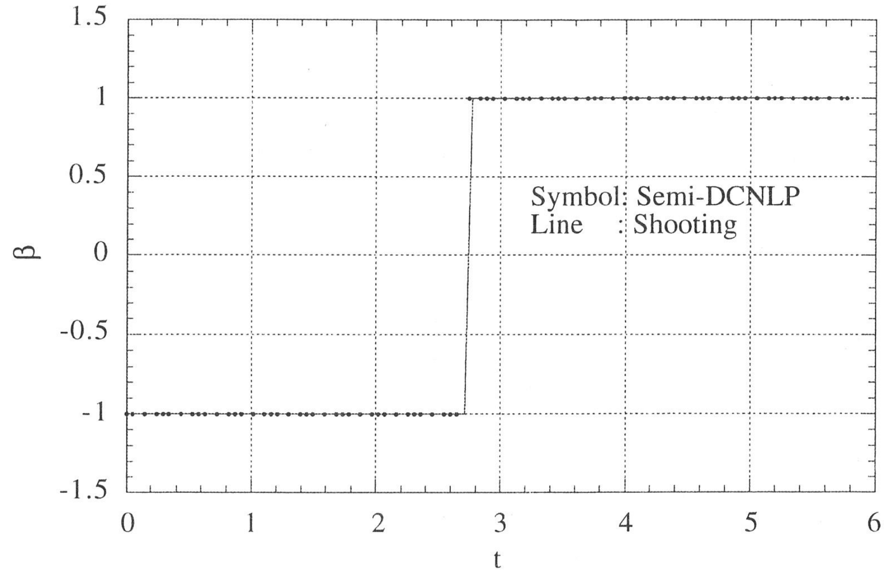

Figure 4. Time History of Control β for the Initial Condition $(x_{10},x_{20})=(3.5,\ 1.5)$

From this example problem, it is seen that the collocation with nonlinear programming method can solve a simple zero-sum two-person differential game problem, with bounded control variables for one-player only. Note that, had the problem been solved as a TPBVP, it would be necessary to integrate the differential equations for 2 states and 2 costates (at least in principle). Using the SDCNLP method only 3 differential equations are required. The saving of effort is small for this simple problem; but much larger for the example to follow.

Minimum-Time Spacecraft Interception for Optimally Evasive Target

As an example of a realistic dynamic problem, saddle-point trajectories are found for the problem of a spacecraft, which is to intercept an optimally evasive target in minimum time. The spacecraft and target vehicles are assumed to be initially in different but coplanar orbits about a planet, and subject to Newtonian gravity. Both vehicles are capable of constant thrust acceleration but the thrust acceleration of the target, T_t, is set smaller than that of the spacecraft, T, so that the spacecraft can intercept the target in finite time. The control variables are the thrust pointing angles of the respective vehicles. Fig. 5 illustrates the problem.

In polar coordinates, the equations of motion may be expressed as:

$$\begin{bmatrix} \dot{v}_r \\ \dot{v}_\theta \\ \dot{r} \\ \dot{\theta} \\ \dot{v}_{rt} \\ \dot{v}_{\theta t} \\ \dot{r}_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} T\sin\delta - (\mu - v_\theta^{\,2} r)/r^2 \\ T\cos\delta - v_r v_\theta / r \\ v_r \\ v_\theta / r \\ T_t \sin\delta_t - (\mu - v_{\theta t}^{\,2} r_t)/r_t^2 \\ T_t \cos\delta_t - v_{rt} v_{\theta t}/r_t \\ v_{rt} \\ v_{\theta t}/r_t \end{bmatrix} \tag{61}$$

where (r, θ) is the position, (v_r, v_θ) the velocity and δ the thrust pointing angle of the spacecraft; (r_t, θ_t) is the position, $(v_{rt}, v_{\theta t})$ the velocity and δ_t the thrust pointing angle of the target.

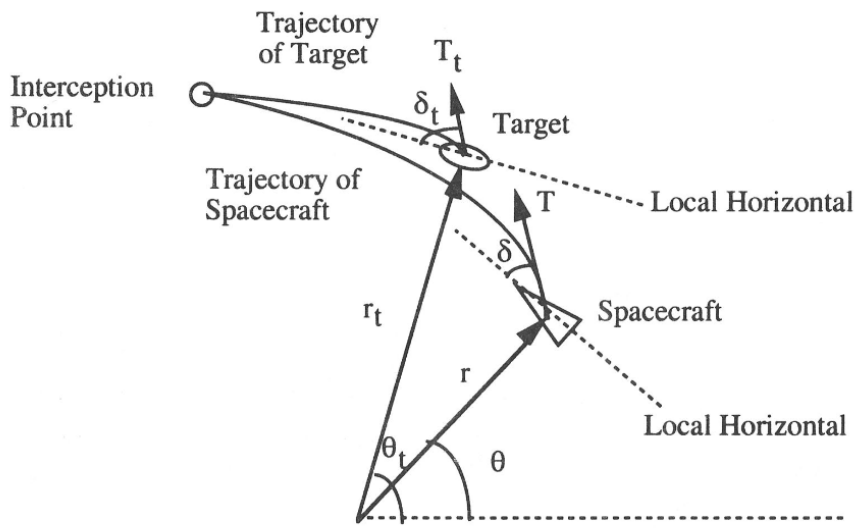

Figure 5. Cartoon of Spacecraft Interception of Optimally Evasive Target

The terminal condition is interception:

$$\begin{bmatrix} r(t_f) - r_t(t_f) \\ \theta(t_f) - \theta_t(t_f) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{62}$$

The cost function is the time to interception:

$$J = t_f \tag{63}$$

Then, the Hamiltonian becomes:

$$H = 1 + \lambda_{v_r}\left[T\sin\delta - (\mu - v_\theta^2 r)/r^2\right] + \lambda_{v_\theta}\left[T\cos\delta - v_r v_\theta/r\right]$$
$$+ \lambda_r v_r + \lambda_\theta v_\theta/r + \lambda_{v_{rt}}\left[T_t\sin\delta_t - (\mu - v_{\theta t}^2 r_t)/r_t^2\right] \quad (64)$$
$$+ \lambda_{v_{\theta t}}\left[T_t\cos\delta_t - v_{rt}v_{\theta t}/r_t\right] + \lambda_{r_t} v_{rt} + \lambda_{\theta_t} v_{\theta_t}/r_t$$

The problem is formulated as a zero-sum two-person differential game. The optimality conditions with respect to δ become:

$$H_\delta = T(\lambda_{v_r}\cos\delta - \lambda_{v_\theta}\sin\delta) = 0 \quad (65)$$

$$H_{\delta\delta} = -T(\lambda_{v_r}\sin\delta + \lambda_{v_\theta}\cos\delta) \geq 0 \quad (66)$$

The equations for adjoint variables associated with the optimality conditions, (65) and (66), are:

$$\dot{\lambda}_{v_r} = \left(\lambda_{v_\theta} v_\theta - \lambda_r r\right)/r \qquad \text{with} \quad \lambda_{v_r}(t_f) = 0 \quad (67)$$

$$\dot{\lambda}_{v_\theta} = \left(-2\lambda_{v_r} v_\theta + \lambda_{v_\theta} v_r - \lambda_\theta\right)/r \qquad \text{with} \quad \lambda_{v_\theta}(t_f) = 0 \quad (68)$$

$$\dot{\lambda}_r = \left[\lambda_{v_r}\left(-2\mu + v_\theta^2 r\right) - \lambda_{v_\theta} v_r v_\theta r + \lambda_\theta v_\theta r\right]/r^3 \qquad \text{with} \quad \lambda_r(t_f) = v_r \quad (69)$$

$$\dot{\lambda}_\theta = 0 \qquad \text{with} \quad \lambda_\theta(t_f) = v_\theta \quad (70)$$

Hence, the problem is to maximize the objective (63) under constraints (61), (62) and (65)-(70) and thus can in principle be solved using the semi-DCNLP method. Note that it is not necessary, when using this method, to solve for the four adjoint variables conjugate to the state variables of the target since the Hamiltonian is not required explicitly and the control for the target, δ_t ,will be found at the collocation points by the NLP problem solver. Had the problem been solved as a TPBVP, it would be necessary to integrate the differential equations for 8 states and 8 costates. Using the semi-DCNLP method only 12 differential equations are required. Since the difficulty of solving such problems, either by shooting (an indirect method) or collocation (a direct method) increases exponentially with the number of variables, this difference in number is significant.

The problem in which the intercepting spacecraft has r(0)=1 with thrust force of T=0.05 and the target spacecraft has $r_t(0)=1.05$ with $T_t=0.0025$, with both vehicles initially in circular orbits, is solved for several initial conditions. Normalized variables are used for convenience i.e. the planet's gravitational constant is set to $\mu=1$ so that the gravitational acceleration at a radial distance r=1, would be 1 distance unit per time unit squared. As an example, saddle-point trajectories and the histories of thrust pointing angles for both spacecraft, for the case $\theta(0)=0$, $\theta_t(0)=0.4$, are shown in Figures 6 and 7 respectively. The semi-DCNLP solution uses 10 segments.

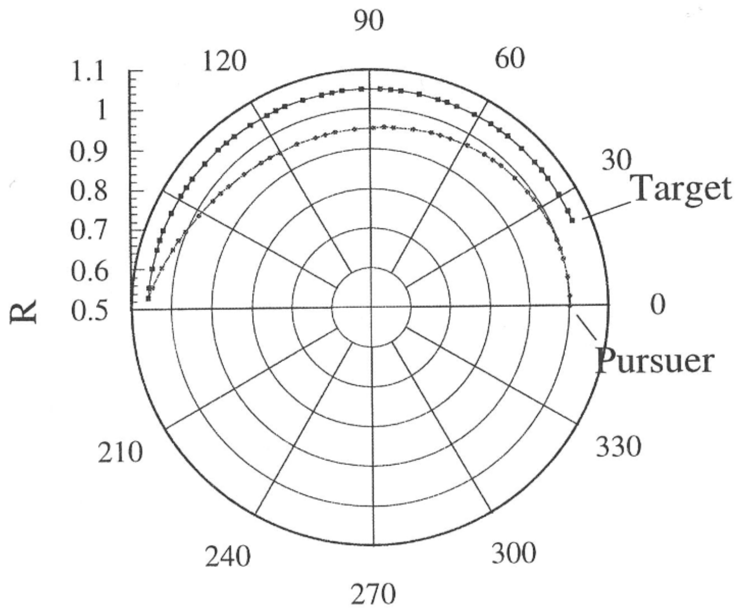

Figure 6. Target and Pursuer Trajectories

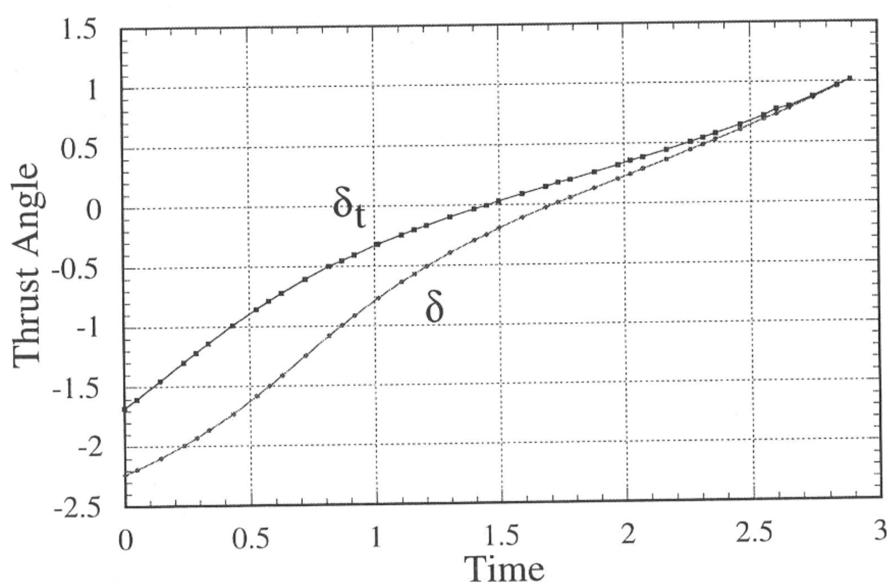

Figure 7. Thrust Pointing Angles of Target and Pursuer

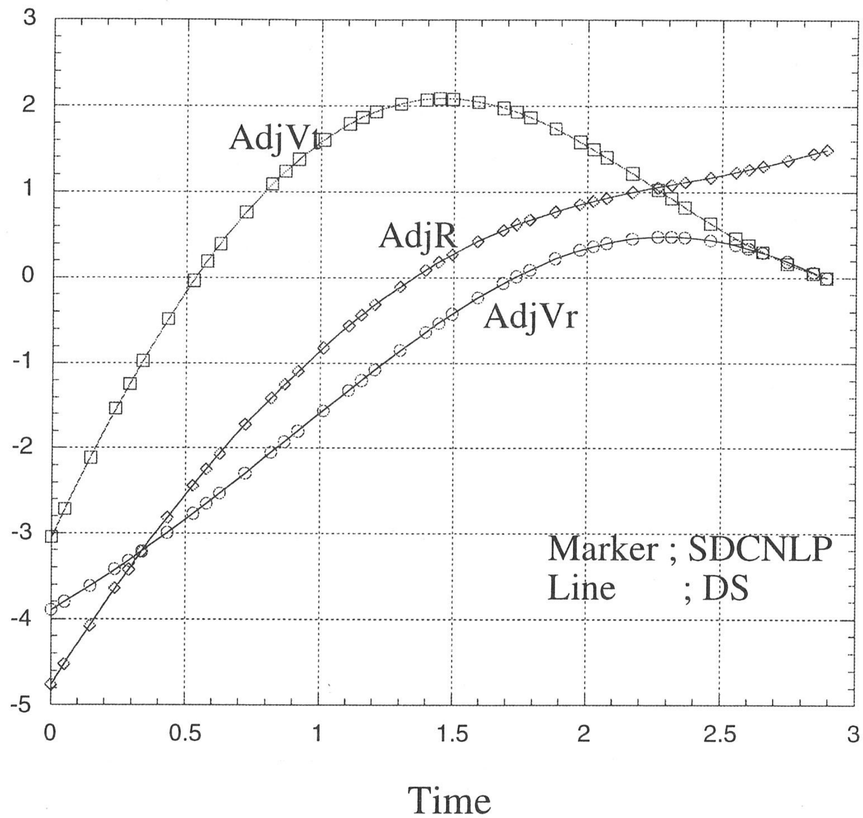

Figure 8. Adjoint Variable Time History for Pursuer Spacecraft

The thrust of both the spacecraft and target have a radially inward component initially which changes to radially outward by the time of interception. Interception occurs after approximately half of a revolution of the planet.

Figure 8 shows the adjoint variable time history for the pursuer spacecraft. The absolute direction of the line-of-sight from the spacecraft to the target for this same example is shown in Fig. 9. Guelman, Shinar and Green [5] suggested "both players turn toward the final line-of-sight direction with an asymptotically decaying rate" in the case of an optimal pursuit-evasion between two variable speed players. It is seen from the figure that the line-of-sight time history for this case has this same characteristic. This observation qualitatively supports the conclusion that the result is a saddle-point trajectory in the pursuit-evasion game.

The spacecraft trajectories and the time histories for the thrust pointing angles for the spacecraft and target, for the same initial and final conditions, are shown in Figures 10 and 11 for the case in which the target thrust pointing angle is bounded; $\pi/6 < \delta_t < \pi/3$. The semi-DCNLP method did not have any greater difficulty solving the problem when the control for one player was bounded.

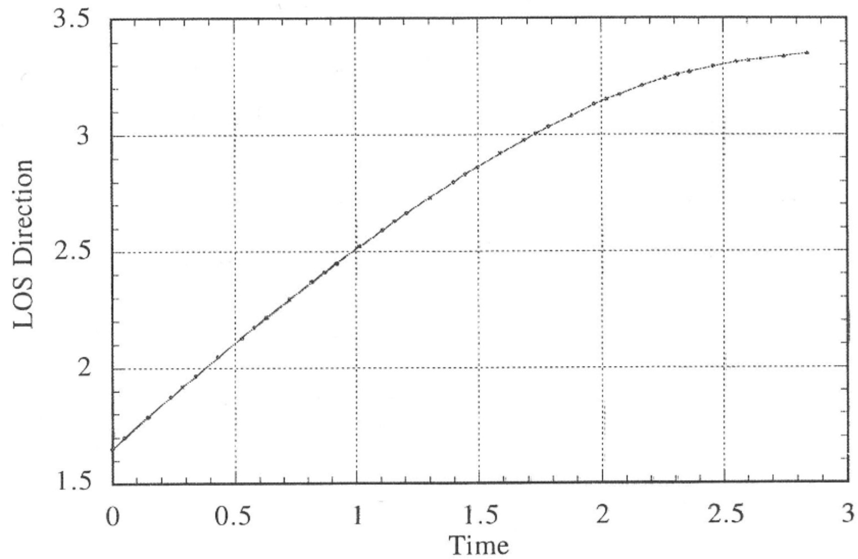

Figure 9. Absolute Line-of-Sight Direction for Spacecraft Interception

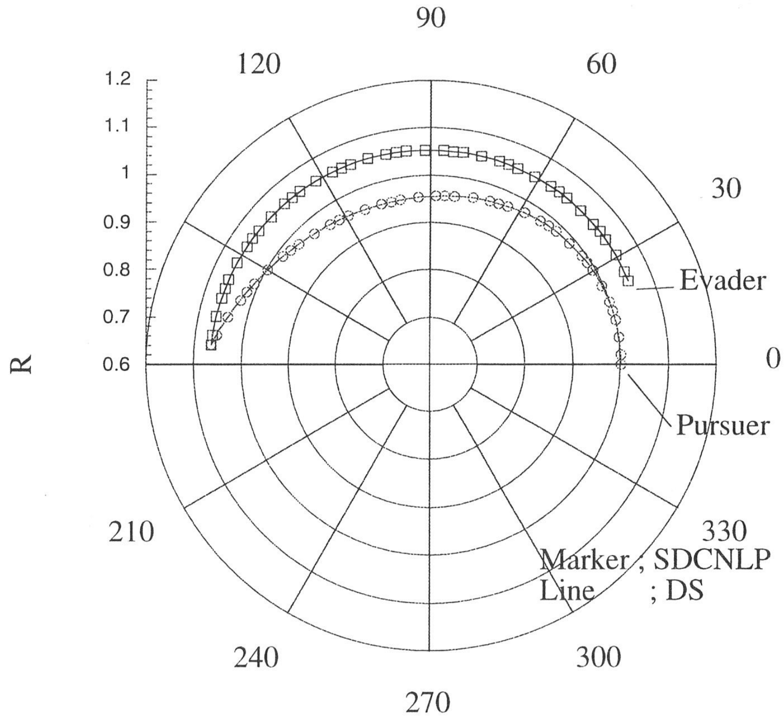

Figure 10. Target and Pursuer Trajectories for Control Constrained Case

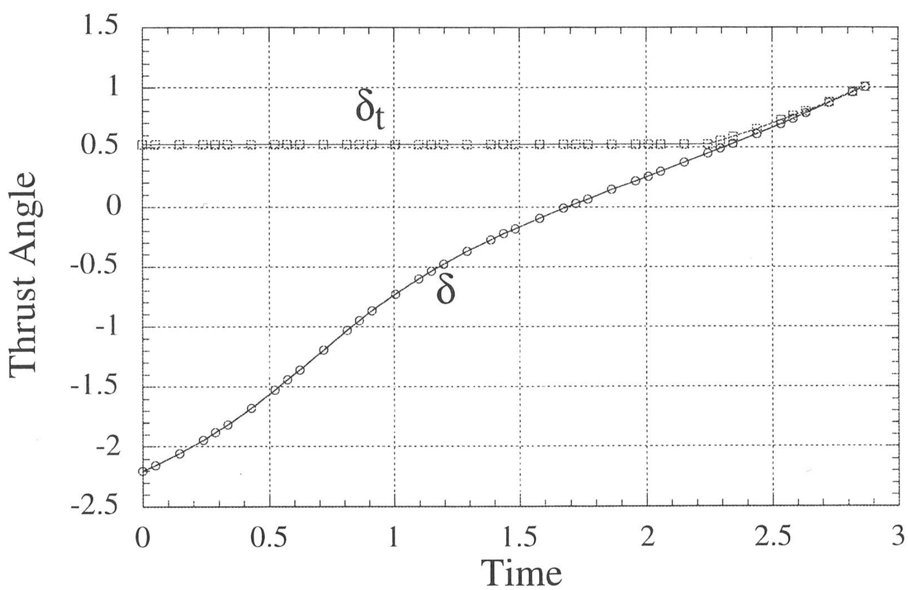

Figure 11. Target and Pursuer Thrust Angles for Control Constrained Case

CONCLUSIONS

A new method for the numerical solution of two-player differential game problems, using collocation, has been developed. Analytical necessary conditions for the problem have been derived. The example of the dolichobrachistochrone shows that the method, which we term semi-direct collocation with nonlinear programming, can solve a zero-sum two-person differential game problem, reproducing the saddle-point trajectories obtained by a shooting method and identifying the principal characteristics of the problem. The spacecraft interception problem verifies that the semi-DCNLP method can solve a practical zero-sum two-person differential game, i.e. one with as many as 8 states and 2 controls.

In both cases the solution shared the robustness which is an attribute of the DCNLP method from which it is derived. Convergence to a solution is undoubtedly aided by the fact that the problem size is significantly reduced when this method is used, in comparison to any method that must solve the original TPBVP, since the costate variables conjugate to the motion of one of the players are not needed.

Some improvements are suggested for future research. Presently, we have not successfully used the semi-DCNLP method to solve a problem whose control variables are bounded for both players. Also, while the original DCNLP method usually solves problems with path constraints easily (e.g. in Ref. 13), the semi-DCNLP method may encounter difficulties because the adjoint equations can change at the entry points of constraint arcs. It is expected that the semi-DCNLP method can be modified to handle path constraints using methods that have been successful for the original DCNLP.

REFERENCES

1. Isaacs, R., *Games of Pursuit*, P-257, the Rand Corporation, Santa Monica, CA., 1951.

2. Bryson, A.E. and Ho, Y.C., *Applied Optimal Control*, Hemisphere, New York, New York, 1975.

3. 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.

4. Segal, A. and Miloh, T., "Novel Three-Dimensional Differential Game and Capture Criteria for a Bank-to-Turn Missile", *Journal of Guidance, Control and Dynamics*, Vol.17, No.5, 1994, pp.1068-1074.

5. 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.

6. Hillberg, C. and Järmark, B., "Pursuit-Evasion Between Two Realistic Aircraft", Paper AIAA-83-2119, *Proceedings of AIAA Atmospheric Flight Mechanics Conference*, Gatlinburg, Tennessee, 1983.

7. Breitner, M., Grimm, W. and Pesch, H.J., "Barrier Trajectories of a Realistic Missile/Target Pursuit-Evasion Game", *Differential Games: Developments in Modeling and Computation*, Edited by R.P.Hamalainen and H.K.Ehtamo, Springer-Verlag, Berlin, Germany, 1991, pp.48-57.

8. 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.

9. 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, pp.443-463, 1993.

10. Lachner, R., Breitner, M.H. and Pesch, H.J., "Three-Dimensional Air Combat: Numerical Solution of Complex Differential Games", *Advances in Dynamic Games and Applications*, Edited by G.J. Olsder, Birkhauser, Boston, Massachusetts, pp.165-190, 1995.

11. 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.

12. Enright, P.J. and Conway, B.A., "Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming", *Journal of Guidance, Control and Dynamics*, Vol.14, No.5, 1991, pp.981-985.

13. Horie, K. and Conway, B. A., "Optimal Aeroassisted Orbital Interception", *Journal of Guidance, Control and Dynamics*, Vol. 22, No. 5, 1999, pp. 625-631.

14. 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.

15. Conway, B. A., "Optimal Low-Thrust Interception of Earth-Crossing Asteroids", *J. of Guidance, Control, and Dynamics*, Vol. 20, No. 5, Sept-Oct 1997, pp. 995-1002.

16. Herman, A. L. and Conway, B. A. "Optimal Low-Thrust, Earth-Moon Orbit Transfer", *J. of Guidance, Control, and Dynamics*, Vol. 21, No. 1, Jan-Feb 1998, pp. 141-147.

17. 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*, SOL 86-2, Stanford University, 1986.

18. Raivio, T. and Ehtamo, H., "On Numerical Solution of a Class of Pursuit-Evasion Games", *Annals of the International Society of Dynamic Games*, Vol. 5, 2000, pp. 177-192.

19. Basar, T. and Olsder, G. J., *Dynamic Noncooperative Game Theory*, SIAM Series in Classics in Applied Mathematics, Philadelphia, 1999.

20. Isaacs, R., *Differential Games*, Wiley, New York, New York, 1965.