On the Numerical Solution of a Class of Pursuit-Evasion Games

Tuomas Raivio and Harri Ehtamo
Systems Analysis Laboratory
Helsinki University of Technology
Espoo, Finland

Abstract

This paper presents a new computational approach for a class of pursuit-evasion games of degree. The saddle-point problem is decomposed into two subproblems that are solved by turns iteratively. The subproblems are ordinary optimal control problems that can be solved efficiently using discretization and nonlinear programming techniques. Hence it is not necessary to supply an initial guess of the adjoint variables or a hypothesis on the switching structure of the solution. Furthermore, in the presented approach the numerical differentiation of the payoff is avoided. We test the algorithm with different numerical examples and compare the results with solutions obtained by an indirect method. In the test examples the method converges rapidly from a rough initial guess, and the results coincide well with the reference solutions.

1 Introduction

Optimal feedback strategies of a pursuit-evasion game of degree are, in principle, obtained by solving the Isaacs partial differential equation in the state space split by singular surfaces. For games with simple dynamics and low-dimensional state vector, the strategies can be determined using the theory of viscosity solutions, see [1] and references cited therein, or the "Tenet of Transition" [8] and integration in retrograde time, see also [17].

For more complex problems, indirect methods can be used to solve the multipoint boundary value problem arising from the necessary conditions of a saddle point. The methods include finite differences, quasilinearization, and the well-known multiple shooting [4], [10]. The convergence domain of indirect methods is small, and continuation and homotopy techniques are frequently required. A drawback of these techniques is the unpredictable changes in the switching structure of the solution in the course of continuation.

In this paper we provide an alternative way to solve the necessary conditions for games in which the players control their own state equations. In such games, the necessary conditions and the optimal saddle-point control histories are coupled only via the terminal payoff and the capture set. Consequently, the saddle-point

J. A. Filar et al. (eds.), *Advances in Dynamic Games and Applications*

problem can be decomposed into two optimal control problems that are solved by turns iteratively using either indirect methods or, as is done here, discretization and nonlinear programming [7], [16]. In particular, it is shown that if the solutions of the subproblems converge, the limit solution satisfies the necessary conditions of an open-loop representation of a feedback saddle-point trajectory. In all the numerical examples the iteration converges well; see Section 4, where the solutions of the numerical examples are compared with reference trajectories obtained by solving the necessary conditions of a saddle point using an indirect method. The examples also suggest that the method works in the presence of certain singular surfaces and separable state-dependent control constraints.

The subproblems of the method are the min and the linearized max problems for the pursuer and the evader, respectively. In [9], Krasovskii and Subbotin give sufficient conditions for the lower (maxmin) value function to coincide with the upper (minmax) value function of the game and suggest that the maxmin problem be solved as a Stackelberg game. Nevertheless, the numerical solution of this rather involved problem is not considered.

Moritz et al. [12] have suggested direct discretization of a pursuit-evasion game by control parametrization. The solution method is based on alternating minimization and maximization steps against a fixed trajectory of the opponent. The convergence of the method essentially depends on the number of optimization steps that the method is allowed to take at each iteration. A large fraction of total computation time is needed to evaluate the gradient of the payoff numerically, whereas in this paper basic sensitivity results are used to express the gradient analytically. Furthermore, it is not clear whether the solutions obtained in [12] satisfy the necessary conditions of a saddle point.

2 Pursuit-Evasion Game

We consider two-player, perfect information zero-sum differential games with unspecified terminal time. We assume that the dynamics of the individual players can be divided into two distinct sets of state equations independent of the states and controls of the other player,

$$\dot{x}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = f(x(t), u_1(t), u_2(t), t) = \begin{pmatrix} f_1(x_1(t), u_1(t), t) \\ f_2(x_2(t), u_2(t), t) \end{pmatrix}, \quad (1)$$
$$x(0) = x_0, \qquad t \in [0, \infty).$$

The state vector $x(t)$ consists of the vectors $x_1(t) \in R^{n_1}$ and $x_2(t) \in R^{n_2}$ that describe the states of the players. Subscripts 1 and 2 refer to the pursuer and the evader, respectively. The admissible controls $u_i(t)$ of the players belong to sets $S_i \subset R^{p_i}$, $i = 1, 2$, for all t. The admissible control functions u_1 and u_2 are assumed to be piecewise continuous functions on the interval $[0, \tilde{T}]$ with $\tilde{T}$

sufficiently large. The terminal payoff of the game is

$$J[u_1, u_2] = q(x(T), T).$$

The game ends when the vector $(x(t), t)$ enters the target set $\Lambda \subset R^{n_1+n_2} \times R^+$, which, together with the terminal cost, are the only connecting factors between the players. The unprescribed final time of the game is defined as

$$T = \inf\{t \mid (x(t), t) \in \Lambda\}.$$

The target set Λ is closed. The boundary $\partial\Lambda$ of Λ is an $(n_1 + n_2)$-dimensional manifold in the space $R^{n_1+n_2} \times R^+$ and is characterized by the scalar equation

$$l(x(t), t) = 0.$$

The function l as well as the functions f and q are assumed to be continuously differentiable in x and t. We assume that the initial state x_0 of the game is selected so that the pursuer can enforce a capture against any action of the evader. Consequently, $T < \infty$ always holds.

Suppose that a pair $(\gamma_1^*, \gamma_2^*) \in \Gamma_1 \times \Gamma_2$ is a saddle-point solution in feedback strategies for the game and let $x^*(t)$ be the corresponding trajectory. The value function of the game when the players start from (x, t) and apply their feedback saddle-point strategies is defined by

$$V(x, t) = \min_{\gamma_1 \in \Gamma_1} \max_{\gamma_2 \in \Gamma_2} q(x(T), T) = \max_{\gamma_2 \in \Gamma_2} \min_{\gamma_1 \in \Gamma_1} q(x(T), T).$$

In this paper we are interested in the open-loop representations $u_i^*(t) := \gamma_i^*(x^*(t), t)$, $i = 1, 2$, of the feedback strategies. The following necessary conditions hold (see [2, Theorem 8.2, p. 433]): There is an adjoint vector $p(t) \in R^{n_1+n_2}$, $t \in [0, T]$, and a Lagrange multiplier $\alpha \in R$ such that

$$\dot{x}^*(t) = f(x^*(t), u_1^*(t), u_2^*(t), t), \tag{2}$$

$$x^*(0) = x_0,$$

$$H(x^*(t), u_1^*(t), u_2(t), p(t), t) \le H(x^*(t), u_1^*(t), u_2^*(t), p(t), t) \le$$
$$H(x^*(t), u_1(t), u_2^*(t), p(t), t) \quad \forall t \in [0, T],\ u_i(t) \in S_i,\ i = 1, 2, \tag{3}$$

$$\dot{p}(t) = -\frac{\partial}{\partial x} H(x^*(t), u_1^*(t), u_2^*(t), p(t), t), \tag{4}$$

$$p(T) = \frac{\partial}{\partial x} q(x^*(T), T) + \alpha \frac{\partial}{\partial x} l(x^*(T), T), \tag{5}$$

$$l(x^*(T), T) = 0, \tag{6}$$

$$H(x^*(T), u_1^*(T), u_2^*(T), p(T), T) = -\frac{\partial}{\partial t} q(x^*(T), T) - \alpha \frac{\partial}{\partial t} l(x^*(T), T), \tag{7}$$

where

$$H(x(t), u_1(t), u_2(t), p(t), t) = p(t)^T f(x(t), u_1(t), u_2(t), t).$$

3 A Feasible Direction Method for the Game

In this section we provide a numerical method to solve the necessary conditions (2)–(7). The solution method is based on the min and the linearized max subproblems that are solved by turns iteratively, either by using indirect methods or discretization and nonlinear programming. If the iteration converges, the solution satisfies the necessary conditions; see Theorem 3.1 below.

The maxmin problem is defined by

$$\begin{aligned} \max_{u_2} \min_{u_1} \quad & q(x(T), T) \\ \dot{x}(t) &= f(x(t), u_1(t), u_2(t), t), \qquad x(0) = x_0, \\ l(x(T), T) &= 0. \end{aligned}$$

To solve it we first consider the minimization problem

$$\begin{aligned} \text{P:} \min_{u_1, T} \quad & q(x_1(T), x_2^0(T), T) \\ \dot{x}_1(t) &= f_1(x_1(t), u_1(t), t), \qquad x_1(0) = x_{10}, \\ l(x_1(T), x_2^0(T), T) &= 0, \end{aligned}$$

where $x_2^0(t)$ is some fixed function of time that represents an arbitrary trajectory of the evader. Let the solution of P be $\bar{x}_1(\cdot)$ with the capture time T^0, and let $e^0 := x_2^0(T^0)$ denote the corresponding capture point. Now, $\bar{x}_1(\cdot)$ also solves the problem where the given trajectory $x_2^0(\cdot)$ in problem P is replaced by the fixed point $e^0 \in R^{n_2}$, and the final time is fixed to T^0.

Next consider all the points $(e, T) \in R^{n_2+1}$ in the neighborhood of (e^0, T^0), and define the value function of the pursuer's problem, corresponding to the initial state x_0, as a function of the capture point (e, T) by

$$\begin{aligned} \tilde{V}(e, T) = \min_{u_1} \{ q(x_1(T), e, T) \mid \dot{x}_1(t) &= f_1(x_1(t), u_1(t), t),\ t \in [0, T], \\ x_1(0) &= x_{10};\ l(x_1(T), e, T) = 0 \}. \end{aligned} \tag{8}$$

It then holds that

$$\tilde{V}(e^0, T^0) = q(\bar{x}_1(T^0), e^0, T^0).$$

The evader's problem is to maximize the pursuer's value function (8). Therefore the original maxmin problem can equivalently be written as

$$\begin{aligned} \text{E}' : \max_{u_2, T} \tilde{V}(x_2(T), T) & \\ \dot{x}_2(t) = f_2(x_2(t), u_2(t), t), & \qquad x_2(0) = x_{20}. \end{aligned}$$

Problem E′ is difficult to solve directly since $\tilde{V}(e, T)$ cannot be expressed analytically. Therefore we proceed as follows: we first approximate the solution of E′ by the solution of problem E (see below), where the final time is fixed to T^0 and $\tilde{V}(e, T^0)$ is linearized in the neighborhood of e^0. To approximate the solution of E′ for $t > T^0$ we extend the solution of E into the interval $[T^0, T^0 + \Delta T]$, $\Delta T > 0$, using a linear approximation.

The linear approximation of $\tilde{V}(e, T^0)$ in the neighborhood of e^0 is given by

$$\tilde{V}(e^0, T^0) + \frac{\partial}{\partial e}\tilde{V}^T(e^0, T^0)(e - e^0).$$

Basic sensitivity results (see [6, Chap. 3.4]; see also [5], [15] for the derivation of the result) imply that the gradient of the value function is given by

$$\frac{\partial}{\partial e}\tilde{V}(e^0, T^0) = \frac{\partial}{\partial e}q(\bar{x}_1(T^0), e^0, T^0) + \alpha^0\frac{\partial}{\partial e}l(\bar{x}_1(T^0), e^0, T^0), \tag{9}$$

where α^0 is the Lagrange multiplier associated with the capture condition in the solution of P. Note that this is an analytical expression. Numerical differentiation of the payoff is avoided.

Neglecting the constant $\tilde{V}(e^0, T^0)$, the fixed final time, free final state problem E can be written as

$$\begin{aligned} \text{E:} \max_{u_2} \quad & c^T(x_2(T^0) - e^0) \\ \dot{x}_2(t) = f_2(x_2(t), u_2(t), t), \quad & \quad x_2(0) = x_{20}, \end{aligned}$$

where

$$c := \frac{\partial}{\partial e}q(\bar{x}_1(T^0), e^0, T^0) + \alpha^0\frac{\partial}{\partial e}l(\bar{x}_1(T^0), e^0, T^0).$$

Let the solution of E be $x_2^1(t)$, $t \in [0, T^0]$. The extension to the interval $[T^0, T^0 + \Delta T]$, $\Delta T > 0$, is done using the linear approximation

$$x_2^1(T^0 + h) = x_2^1(T^0) + \dot{x}_2^1(T^0)h, \qquad h \in [0, \Delta T]. \tag{10}$$

The extended solution of E is now inserted back into P, which is solved anew to locate the new capture point and to evaluate the linear approximation of the value function. Problems P and E are solved and updated until a constrained maximum of $\tilde{V}(e, T)$, and thus the solution of problem E′, is achieved. To summarize, the iteration proceeds as follows:

1. Fix an initial trajectory of the evader and solve P. Obtain e^0, T^0, and α^0. Set $k := 0$.
2. Solve E using e^k, T^k and α^k.
3. Insert the extended solution x_2^{k+1} of E into P and solve P to obtain e^{k+1}, T^{k+1} and α^{k+1}. If $||e^{k+1} - e^k|| < \epsilon$, where ϵ is the desired accuracy, terminate. Otherwise, set $k := k + 1$ and go to step 2.

The following theorem shows that the limit solution satisfies the necessary conditions of a saddle point; the proof is given in the Appendix:

Theorem 3.1. *Suppose that the iteration converges. Denote the limit solution of P and E by $u_1^*(t)$ and $u_2^*(t)$, respectively, the corresponding trajectories by $x_1^*(t)$ and $x_2^*(t)$, the capture time by T^*, and the Lagrange multiplier by α^*. Then the* *solution satisfies the necessary conditions* (2)–(7) *for an open-loop saddle point or for an open-loop representation of a feedback saddle-point strategy.*

Thus the solution of the necessary conditions can be decomposed to the solution of the necessary conditions for subproblems P and E involving one player. The subproblems can be solved, e.g., using multiple shooting. Nevertheless, as the subproblems are optimal control problems, discretization and nonlinear programming can also be applied. The solutions produced by these methods approximate the optimal state and control trajectories up to an accuracy that depends on the discretization interval and the scheme being used. In addition, the Lagrange multipliers corresponding to the constraints that enforce pointwise satisfaction of the state equations approximate the adjoint trajectories (for Euler discretization and direct collocation, see [18]).

Discretizing P results in a nonlinear programming problem PD of the form

$$\begin{aligned} \text{PD: min} \quad & q(x_1^{N_1}, x_2^0(T), T) \\ g_1(\bar{x}_1, \bar{u}_1, T) &\le \bar{0}, \\ h_1(\bar{x}_1, \bar{u}_1, T) &= \bar{0}, \\ l(x_1^{N_1}, x_2^0(T), T) &\le 0, \\ T &> 0. \end{aligned}$$

Here $\bar{x}_1 \in R^{N_1 \times n_1}$ and $\bar{u}_1 \in R^{N_1 \times p_1}$ refer to the discretized trajectory and controls of the pursuer. The number of discretization points is N_1, the superscript referring to the last discretization point. Note that x_2^0 is a given function of T and need not be discretized.

The expression for the multiplier α^0 to be used in (9) can be obtained analytically using the necessary conditions (15) and (17) of problem P, and the definition of H_1; see the Appendix:

$$\begin{aligned} \alpha^0 = & -[\partial q/\partial x_1^{N_1}(x_1^{N_1}, x_2^0(T^0), T^0) f_1(x_1^{N_1}, u_1^{N_1}, T^0) \\ & + \partial q/\partial x_2(x_1^{N_1}, x_2^0(T^0), T^0) f_2(x_2^0(T^0), u_2^0(T^0), T^0) \\ & + \partial q/\partial T(x_1^{N_1}, x_2^0(T^0), T^0)]/ \\ & [\partial l/\partial x_1^{N_1}(x_1, x_2^0(T^0), T^0) f_1(x_1^{N_1}, u_1^{N_1}, T^0) \\ & + \partial l/\partial x_2(x_1^{N_1}, x_2^0(T^0), T^0) f_2(x_2^0(T^0), u_2^0(T^0), T^0) \\ & + \partial l/\partial T(x_1^{N_1}, x_2^0(T^0), T^0)]. \end{aligned}$$

Problem ED, the finite-dimensional version of problem E, is of the form

$$\begin{aligned} \text{ED: max} \quad & \eta^T(x_2^{N_2} - e^0) \\ g_2(\bar{x}_2, \bar{u}_2, T^0) &\le \bar{0}, \\ h_2(\bar{x}_2, \bar{u}_2, T^0) &= \bar{0}, \end{aligned}$$

where

$$\eta = \frac{\partial}{\partial e} q(x_1^{N_1}, e^0, T^0) + \alpha^0 \frac{\partial}{\partial e} l(x_1^{N_1}, e^0, T^0).$$

4 Numerical Examples

In this section we assess the approach by solving some numerical examples that can be solved also analytically or indirectly. First the method is applied to the "Homicidal Chauffeur" game [8]. To check that the necessary conditions indeed become satisfied we solve a simple aerial dogfight in the horizontal plane. Finally, we investigate the method in a head-on encounter by solving a collision avoidance problem proposed by Lachner et al. [11]. All the subproblems are discretized using direct collocation and solved by Sequential Quadratic Programming; see [7].

Example 4.1. In the first example the pursuer has a positive minimum turning radius whereas the evader may redirect his velocity vector instantaneously. Both players move with constant velocities v_1 and v_2, $v_1 > v_2$, in the horizontal plane. The objective of the pursuer is to catch the evader in minimum time and the evader tries to avoid capture as long as possible. The payoff q equals the terminal time T. The players' equations of motion are

$$\begin{aligned} \dot{x}_1 &= v_1 \cos\phi_1, & \dot{x}_2 &= v_2 \cos u_2, \\ \dot{y}_1 &= v_1 \sin\phi_1, & \dot{y}_2 &= v_2 \sin u_2, \\ \dot{\phi}_1 &= u_1. \end{aligned}$$

Here and in the other examples x_i and y_i, $i = 1, 2$, refer to the x- and y-coordinates of the players and ϕ_1 refers to the heading of the pursuer. The absolute angular velocity $|u_1|$ of the pursuer is restricted to be less than some given maximum

$$|u_1| \le u_{\max}.$$

The target set Λ can be expressed by the inequality

$$(x_1(T) - x_2(T))^2 + (y_1(T) - y_2(T))^2 \le d^2,$$

where d is the capture radius. The boundary of Λ is the circle

$$l(x(T)) = (x_1(T) - x_2(T))^2 + (y_1(T) - y_2(T))^2 - d^2 = 0.$$

The vector $x = [x_1, y_1, \phi_1, x_2, y_2]^T$ denotes here the state of the game. Despite its simplicity the game captures salient features of pursuit-evasion games. For example, most saddle-point solutions involve singular surfaces.

The parameters of the game are $v_1 = 3$, $v_2 = 1$, $d = 1$, and $u_{1,\max} = 1$. Four cases are considered. The evader's initial position is varied according to Table 1, and the initial conditions of the pursuer are

$$x_{10} = y_{10} = \phi_{10} = 0.$$

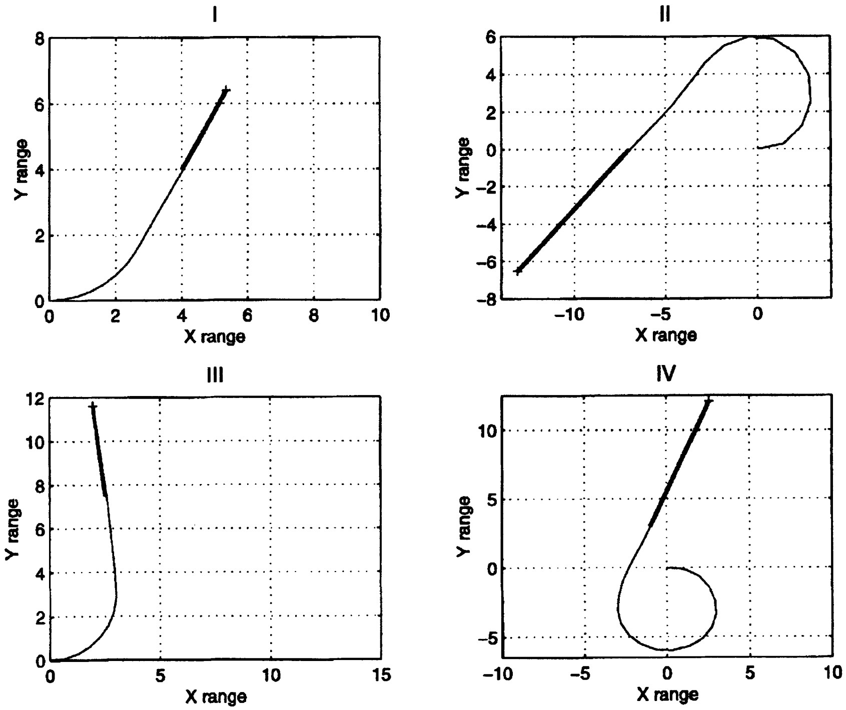

Figure 1: The solutions of the homicidal chauffeur game corresponding to the initial conditions given in Table 1. The thick line represents the evader's trajectory.

In the first case the evader starts in front of the pursuer. A dispersal surface is encountered in the second case, as the initial state of the evader is directly behind the pursuer. The third initial state lies near the barrier, and the fourth one is on the other side of it. The solution trajectories are presented in Figure 1, and information on the results and the computational procedure in Table 1.

All the solutions end up in a tail chase, corresponding to a universal surface. Only games where the evader is initially close to the pursuer in front of him or on the barrier would end otherwise. The treatment of a dispersal surface in case II would require instantaneous mixed strategies for the pursuer [8]. From the computational point of view, the initial guess of the pursuer's trajectory determines the side to which the play evolves. The vicinity of the barrier does not seem to affect the convergence of case III.

The last case converges to a maxmin solution, i.e., the evader plays nonoptimally. The evader should first pursue the pursuer to gain time. The method cannot identify the equivocal surface. Its tributaries are not solutions of the maxmin problem.

Example 4.2. We consider two aircraft moving in a horizontal plane. The pursuing aircraft attempts to capture the evader in minimum time. The evading aircraft tries

Table 1: Initial conditions of the evader and summary of the results of Example 1. For cases I and III the initial guess of the evader's trajectory is a straight path along the positive x-axis and for cases II and IV along the negative x-axis. The iteration was executed until the relative change of the capture point was less than 5×10^{-3}. The reference value is computed analytically. Case IV is a maxmin solution.

Case	I	II	III	IV
(x_{20}, y_{20})	(4,4)	(−7,0)	(2.5,7.5)	(−1,3)
No. of iterations	8	10	11	11
Computed value, sec.	2.506	8.956	4.124	9.763
Ref. value, sec.	2.504	8.927	4.120	*
Relative error, %	0.08	0.3	0.08	*

to avoid this situation as long as possible. Assuming point mass dynamics, the players' equations of motion are

$$\begin{aligned}
\dot{x}_i &= v_i \cos\phi_i, \\
\dot{y}_i &= v_i \sin\phi_i, \\
\dot{\phi}_i &= \omega_i, \\
\dot{v}_i &= g(A_i(v_i) - B_i(v_i)v_i^2 - C_i(v_i)((\omega_i/g)^2 + 1/v_i^2)), \qquad i = 1, 2,
\end{aligned}$$

where x_i, y_i, ϕ_i, and v_i stand for the x- and y-coordinates, heading angle, and velocity of aircraft i. The gravitational acceleration is denoted by g. The coefficients A_i, B_i, and C_i, $i = 1, 2$, describe the thrust and the drag forces of the aircraft, see [14]. The angular velocities $\omega_{1,2}$ are constrained by

$$|\omega_i| \leq \omega_{\max,i}, \qquad i = 1, 2.$$

The pursuing aircraft is assumed slightly faster and more agile than the evading one. The capture condition is similar to the one in previous examples, the capture radius being now 100 [m].

The scenario takes place at the altitude of $h_0 = 3000$ m from the initial conditions

$$\begin{aligned}
x_{10} &= 0 \text{ m}, & x_{20} &= 3000 \text{ m}, \\
y_{10} &= 0 \text{ m}, & y_{20} &= 5000 \text{ m}, \\
\phi_{10} &= 0 \text{ rad}, & \phi_{20} &= -1 \text{ rad}, \\
v_{10} &= 200 \text{ m/s}, & v_{20} &= 100 \text{ m/s}.
\end{aligned}$$

The saddle-point trajectories produced by this method are shown in Figure 2 together with the reference solution that was obtained via solving the necessary conditions of the saddle point for this problem. The boundary value problem was solved using the BNDSCO software package [13]. The solution trajectory, as well as the value, are almost identical with the reference solution. It is easy to numerically check that the solution satisfies the necessary conditions (2)–(7) for

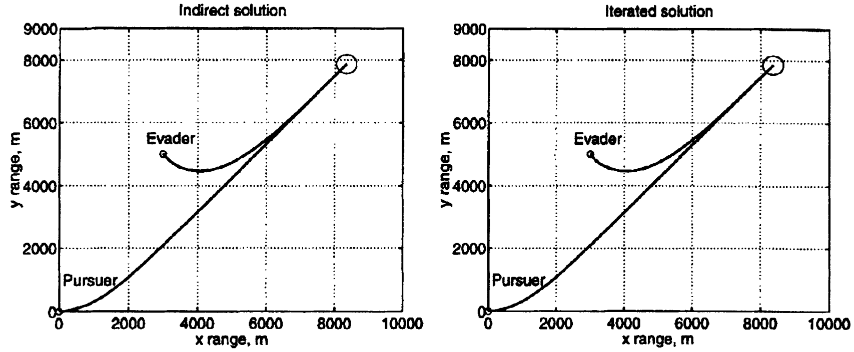

Figure 2: On the left, the reference solution of the aerial dogfight problem. On the right, the solution obtained by the proposed method. The circle indicates the projection of the terminal manifold (drawn larger than actual). The computed value of the game is 34.647, whereas the reference value is 34.649 [sec].

the saddle point when the Lagrange multipliers of the collocation constraints are used in approximating the adjoint variables.

Example 4.3. The last example, a collision avoidance problem, is a variant of the game of two cars [8]. The pursuer is a car driving in the wrong direction on a freeway defined by

$$-7.5 < x < 7.5, \qquad y \text{ free.}$$

The evader, driving in the correct direction, tries to avoid collision with the pursuer, who is assumed to aim at colliding with the evader. The players obey the state equations

$$\begin{aligned} \dot{x}_1 &= v_1 \sin\phi_1, & \dot{x}_2 &= v_2 \sin\phi_2, \\ \dot{y}_1 &= v_1 \cos\phi_1, & \dot{y}_2 &= v_2 \cos\phi_2, \\ \dot{\phi}_1 &= u_1, & \dot{\phi}_2 &= u_2, \\ & & \dot{v}_2 &= b\eta_2, \end{aligned}$$

where ϕ_i, $i = 1, 2$, are measured clockwise from the positive y-axis. The velocity of the pursuer, v_1, is assumed constant, but the evader can accelerate and decelerate by selecting $\eta_2 \in [-0.1, 1]$ appropriately. The maximum deceleration rate is described by the constant b. The angular velocities of both players are constrained by

$$|u_i| \le u_{i,\max} = \frac{\mu_0 g}{v_i},$$

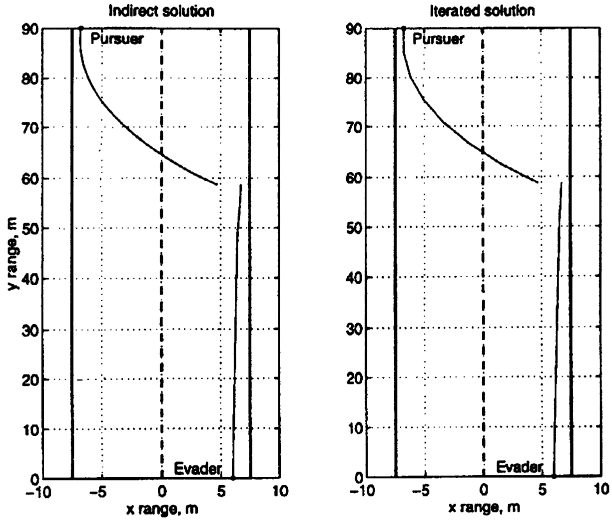

Figure 3: The solution of the collision avoidance problem. On the left, the solution obtained by an indirect method and on the right, the solution obtained by the proposed approach. The value is computed to be 1.991, whereas the value obtained by the indirect approach is 2.012 [m]. The final times are computed to be 2.2876 and 2.2870 seconds, respectively.

where μ_0 is the friction coefficient between tires and asphalt and g is the acceleration due to gravity. To stay on the road the evader has to satisfy the following additional constraint, which is here given for the right edge of the freeway:

$$\begin{aligned} &\frac{x_2 - a}{v_2^2}(K_1^2 + K_2^2) + K_2(\cos\phi_2 - e^{-\phi_2(K_1/K_2)}) \\ &-K_1 \sin\phi_2 + \frac{u_2/u_{2,\max} + 1}{K_3} - \frac{\eta_2 - \eta_{2,\max}}{K_4} \leq 0. \end{aligned} \tag{11}$$

Here $a = 6.75$, $K_1 = 2b\eta_{2,\max}$, $K_2 = \mu_0 g u_{2,\min}$, and $K_3 = K_4 = 120$. The constraint assures that the evader will stay on the freeway even after the avoidance maneuver. For details, see [11]. Note that both constraints above are mixed state and control constraints. The target set Λ is the set of states where

$$y_1(T) \leq y_2(T).$$

The boundary of the target set is characterized by

$$l(x(T)) = y_1(T) - y_2(T) = 0,$$

that is, the game ends when the cars pass each other. The payoff is the distance at the moment of termination,

$$q(x(T)) = |x_1(T) - x_2(T)|.$$

Table 2: Summary of the last two examples. The reference values are computed by an indirect method. For the collision avoidance problem the number of discretization points was smaller for the pursuer.

Example	Aerial Dogfight	Wrong Driver
No. of nodes	15	8,15
No. of iterations	5	2
Computed value	34.647 s	1.991 m
Ref. value	34.649 s	2.012 m
Relative error, %	0.005	1.0

The values for the parameters are $\mu_0 = 0.55$, $b = -6$ [m/s^2], and $v_1 = 15$ [m/s]. The initial conditions are

$$\begin{aligned} x_{10} &= -6.75 \text{ m}, & x_{20} &= 6.0 \text{ m}, \\ y_{10} &= 90.0 \text{ m}, & y_{20} &= 0 \text{ m}, \\ \phi_{10} &= 3.202 \text{ rad}, & \phi_{20} &= 0 \text{ rad}, \\ v_{20} &= 25 \text{ m/s}, \end{aligned}$$

i.e., the game is a head-on encounter. The solution, together with the reference solution obtained by an indirect method, is given in Figure 3. The evader's discretized problem cannot fully capture the resulting rapid changes of controls. The value of the game is computed rather accurately. The results of the last two examples are summarized in Table 2.

5 Conclusion

In this paper, the computation of a solution for certain pursuit-evasion games of degree is decomposed into a solution of a sequence of optimal control problems involving a single player. These optimal control problems can be solved either by indirect methods or by discretization and nonlinear programming. The converged solution is shown to satisfy the regular necessary conditions of a saddle point.

In the numerical examples the iteration converges well. The method as such is a feasible direction method (see [3, Chap. 10]) where a new direction is produced by linearizing the objective function, see problem E$'$ and the linearized problem E. In order to ensure convergence for such methods a line search should be conducted along the produced direction. In our method the line search could, in principle, be performed by computing the value of $\tilde{V}(e, T)$ in discrete points along the line from (e^k, T^k) to (e^{k+1}, T^{k+1}).

Numerical examples suggest that in addition to regular solution arcs the method is able to locate at least a universal surface and a switching line and can handle a dispersal surface. In Example 4.3, the switching structure of the solution, induced by state-dependent control constraints, is approximately identified. Case

IV of Example 4.1 demonstrates that an equivocal surface cannot be detected by the method. Nevertheless, the method still provides a maxmin solution, which also might be useful in practice.

The method can possibly offer a robust way to extend the recent automated solution framework of optimal control problems [19] into pursuit-evasion games. Automatically computed open-loop representations of feedback strategies would allow designers, engineers, and decision makers to easily study the worst-case performance of different systems. On the other hand, synthesis of optimal feedback strategies for complex models requires that massive amounts of open-loop representations corresponding to different initial states are computed. The flexibility of the method makes it easy to automate the computational procedure, since the convergence domain is large and the switching structure of the solution need not be specified in advance.

In the present approach we first decompose the continuous-time saddle-point problem into subproblems and then discretize them. Another possibility, to be addressed in the future, would be to first discretize the saddle-point problem and then solve this finite-dimensional setting using a suitable decomposition method. For a preliminary work along this direction, see [5].

Appendix: Proof of Theorem 1

First, the optimal solutions u_1^*, x_1^*, p_1, and α^* satisfy the necessary conditions of optimality for P, that is,

$$\dot{x}_1^*(t) = f_1(x_1^*(t), u_1^*(t), t), \tag{12}$$

$$\forall t \in [0, T^*] : u_1^*(t) = \arg \min_{u_1(t) \in S_1} H_1(x_1^*(t), u_1(t), p_1(t), t), \tag{13}$$

$$\dot{p}_1(t) = -\frac{\partial}{\partial x_1} H_1(x_1^*(t), u_1^*(t), p_1(t), t), \tag{14}$$

$$\begin{aligned} p_1(T^*) = {} & \frac{\partial}{\partial x_1} q(x_1^*(T^*), x_2^*(T^*), T^*) \\ & + \alpha^* \frac{\partial}{\partial x_1} l(x_1^*(T^*), x_2^*(T^*), T^*), \end{aligned} \tag{15}$$

$$l(x_1^*(T^*), x_2^*(T^*), T^*) = 0, \tag{16}$$

$$\begin{aligned} H_1(x_1^*(T^*), u_1^*(T^*), p_1(T^*), T^*) = {} & -\frac{\partial}{\partial t} q(x_1^*(T^*), x_2^*(T^*), T^*) \\ & - \alpha^* \frac{\partial}{\partial t} l(x_1^*(T^*), x_2^*(T^*), T^*), \end{aligned} \tag{17}$$

where

$$H_1(x_1(t), u_1(t), p_1(t), t) = p_1(t)^T f_1(x_1(t), u_1(t), t).$$

Note that in (17), the functions $q(\cdot)$ and $l(\cdot)$ explicitly depend on time also through $x_2^*(t)$, since in problem P it is assumed that $x_2^0(t)$, which now equals $x_2^*(t)$, is a given, time-dependent function.

Second, the optimal control u_2^* and the corresponding state and adjoint trajectories x_2^* and p_2 satisfy the necessary conditions of optimality for E:

$$\dot{x}_2^*(t) = f_2(x_2^*(t), u_2^*(t), t), \tag{18}$$

$$\forall t \in [0, T^*]: \; u_2^*(t) = \arg \max_{u_2(t) \in S_2} H_2(x_2^*(t), u_2(t), p_2(t), t), \tag{19}$$

$$\dot{p}_2(t) = -\frac{\partial}{\partial x_2} H_2(x_2^*(t), u_2^*(t), p_2(t), t), \tag{20}$$

$$p_2(T^*) = \frac{\partial}{\partial x_2} c^T (x_2^*(T^*) - e^*) = c, \tag{21}$$

where

$$H_2(x_2(t), u_2(t), p_2(t), t) = p_2(t)^T f_2(x_2(t), u_2(t), t),$$

and

$$c = \frac{\partial q}{\partial x_2}(x_1^*(T^*), x_2^*(T^*), T^*) + \alpha^* \frac{\partial l}{\partial x_2}(x_1^*(T^*), x_2^*(T^*), T^*).$$

Combining (12) and (18) gives (2); combining (14) and (20) gives (4). The terminal conditions of the adjoint variables, (15) and (21), coincide with (5) and the capture condition (16) with (6). Summing H_1 and H_2 yields the Hamiltonian of the game and conditions (13) and (19) correspond to the saddle-point condition (3). Furthermore, at time T^*,

$$\begin{aligned} &H_1(x_1^*(T^*), u_1^*(T^*), p_1(T^*), T^*) + H_2(x_2^*(T^*), u_2^*(T^*), p_2(T^*), T^*) \\ &\quad = -\frac{\partial}{\partial t}\{q(x_1^*(T^*), x_2^*(T^*), T^*) + \alpha^* l(x_1^*(T^*), x_2^*(T^*), T^*)\} \\ &\qquad + p_2(T^*)^T f_2(x_2^*(T^*), u_2^*(T^*), T^*). \end{aligned} \tag{22}$$

The time derivative in the right-hand side of (22) can be written as (recall that $q(\cdot)$ and $l(\cdot)$ here explicitly depend on time also through x_2^*):

$$\begin{aligned} &-\frac{\partial}{\partial x_2}\{q(x_1^*(T^*), x_2^*(T^*), T^*) + \alpha^* l(x_1^*(T^*), x_2^*(T^*), T^*)\}\dot{x}_2^*(T^*) \\ &\quad -\frac{\partial}{\partial t}\{q(x_1^*(T^*), x_2^*(T^*), T^*) + \alpha^* l(x_1^*(T^*), x_2^*(T^*), T^*)\} \\ &= -p_2(T^*)^T f_2(x_2^*(T^*), u_2^*(T^*), T^*) \\ &\quad -\frac{\partial}{\partial t}\{q(x_1^*(T^*), x_2^*(T^*), T^*) + \alpha^* l(x_1^*(T^*), x_2^*(T^*), T^*)\}; \end{aligned} \tag{23}$$

above the time derivative $\partial/\partial t$ is taken only with respect to the explicit t-dependence.

Substituting (23) into (22) yields

$$H_1(x_1^*(T^*), u_1^*(T^*), p_1(T^*), T^*) + H_2(x_2^*(T^*), u_2^*(T^*), p_2(T^*), T^*)$$
$$= -\frac{\partial}{\partial t} q(x_1^*(T^*), x_2^*(T^*), T^*) - \alpha^* \frac{\partial}{\partial t} l(x_1^*(T^*), x_2(T^*), T^*), \quad (24)$$

which gives (7). □

Remark. The proof can be easily extended to include separable mixed state and control constraints, too. Suppose constraints of the form $g_1(x_1, u_1) \leq 0$, $g_2(x_2, u_2) \leq 0$ arise for the pursuer and the evader, respectively. Let $\mu_1(t)$ and $\mu_2(t)$ be the Lagrange multipliers corresponding to these constraints at optimum. The only effect the additional constraints have on the proof above is to add the terms $\mu_1^T g_1$, $\mu_2^T g_2$ to the definitions of H_1 and H_2 above. The terminal condition (22) and hence the derived equation (24) remain unaffected, since $\mu_2^T(T^*) g_2(x_2^*(T^*), u_2(T^*)) = 0$.

Acknowledgments

The authors wish to thank Dr. Rainer Lachner, TU-Clausthal, Germany, for providing the collision avoidance problem and the reference solution.

REFERENCES

[1] Bardi, M., S. Bottacin, and M. Falcone. "Convergence of Discrete Schemes for Discontinuous Value Functions of Pursuit-Evasion Games", In: G. J. Olsder, ed. *New Trends in Dynamic Games and Applications*, pp. 273–304. Birkhäuser, Boston, 1996.

[2] Basar, T. and G. Olsder. *Dynamic Noncooperative Game Theory*, 2nd ed. Academic Press, London, 1995.

[3] Bazaraa, M., H. Sherali, and C. Shetty. *Nonlinear Programming, Theory and Algorithms*. 2nd ed., Wiley, New York, 1993.

[4] Breitner, M., H. Pesch, and W. Grimm. "Complex Differential Games of Pursuit-Evasion Type with State Constraints, Part 2: Multiple Shooting and Homotopy", *Journal of Optimization Theory and Applications*, **78**, No. 3, 442–464, 1993.

[5] Ehtamo, H. and T. Raivio. A Feasible Direction Method for Saddle Point Problems. *Proceedings of the 8th International Symposium on Dynamic Games and Applications*, Maastricht, The Netherlands, pp. 184–188, 1998.

[6] Fiacco, A. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press, New York, 1983.

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

[8] Isaacs, R. *Differential Games*. Reprint. Krieger, New York, 1975.

[9] Krasovskii, N. and A. Subbotin. *Game Theoretical Control Problems*. Springer-Verlag, New York, 1988.

[10] Lachner, R., M. Breitner, and H. Pesch. Three-Dimensional Air Combat: Numerical Solution of Complex Differential Games. In: G. Olsder, ed. *New Trends in Dynamic Games and Applications*, pp. 165–190. Birkhäuser, Boston, 1996.

[11] Lachner, R., M. Breitner, and H. Pesch. Real-Time Collision Avoidance Against Wrong Drivers: Differential Game Approach, Numerical Solution and Synthesis of Strategies with Neural Networks. *Proceedings of the 7th International Symposium on Dynamic Games and Applications*, Kanagawa, Japan, 564–572, 1996.

[12] Moritz, K., R. Polis, and K. Well. Pursuit-Evasion in Medium-Range Air-Combat Scenarios. *Computers and Mathematics with Applications*, **13**, No. 1–3, 167–180, 1987.

[13] Oberle, H. and W. Grimm. BNDSCO—A Program for the Numerical Solution of Optimal Control Problems. Program manual, 1989.

[14] Raivio, T., H. Ehtamo, and R. Hämäläinen. Simulating the Worst Possible Prediction Error of System Trajectories. *Proceedings of the European Simulation Multiconference*, Budapest, Hungary, pp. 299–303, 1996.

[15] Raivio, T., H. Ehtamo, and R. Hämäläinen. A Decomposition Method for a Class of Pursuit-Evasion Games. *Proceedings of the 7th International Symposium on Dynamic Games and Applications*, Kanagawa, Japan, pp. 784–795, 1996.

[16] Seywald, H. Trajectory Optimization Based on Differential Inclusion. *Journal of Guidance, Control and Dynamics*, **17**, No. 3, 480–487, 1994.

[17] Shinar, J., M. Guelman, and A. Green. An Optimal Guidance Law for a Planar Pursuit-Evasion Game of Kind, *Computers and Mathematics with Applications*, **18**, No. 1–3, 35–44, 1989.

[18] Stryk von, O. Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variabeln. Doctoral Dissertation, Munich University of Technology, 1994 (In German).

[19] Virtanen, K., H. Ehtamo, T. Raivio, and R. Hämäläinen. VIATO—Visual interactive Aircraft Trajectory Optimization. *IEEE Transactions on Systems, Man and Cybernetics*, Part C, *Applications and Reviews*, **29**, 3, 409–421, 1999.