Helsinki University of Technology

Systems Analysis Laboratory Research Reports E4, March 2000

APPLYING NONLINEAR PROGRAMMING TO PURSUIT-EVASION GAMES

Harri Ehtamo Tuomas Raivio

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Distribution:
Systems Analysis Laboratory
Helsinki University of Technology
P.O. Box 1100
FIN-02015 HUT, FINLAND
Tel. +358-9-451 3056
Fax. +358-9-451 3096
systems.analysis@hut.fi

This report is available at
www.e-reports.sal.hut.fi/pdf/E4.pdf

Series E - Electronic Reports
www.e-reports.sal.hut.fi

ISBN 951-22-4919-7
ISSN 1456-5218

Title: Applying nonlinear programming to pursuit-evasion games

Authors: Harri Ehtamo
Systems Analysis Laboratory
Helsinki University of Technology
P.O. Box 1100, 02015 HUT, FINLAND
harri.ehtamo@hut.fi
www.sal.hut.fi/Personnel/Homepages/HarriE.html

Tuomas Raivio
Systems Analysis Laboratory
Helsinki University of Technology
P.O. Box 1100, 02015 HUT, FINLAND
tuomas.raivio@hut.fi
www.sal.hut.fi/Personnel/Homepages/TuomasR.html

Date: March, 2000

Status: Systems Analysis Laboratory Research Reports E4 March 2000

Abstract: Motivated by the benefits of discretization in optimal control problems we consider possibilities to discretize pursuit-evasion games. Two approaches are introduced. In the first one, the solution of the necessary conditions of the continuous-time game is decomposed into ordinary optimal control problems that can be solved using discretization and nonlinear programming techniques. In the second approach, the game is discretized and transformed into a bilevel programming problem which is solved using a first order feasible direction method. Although the starting point of the approaches is different, they lead in practice to the same solution algorithm. We demonstrate the usability of the discretization by solving some open-loop representations of feedback solutions for a complex pursuit-evasion game between a realistically modeled aircraft and a missile with terminal time as the payoff. The solutions are compared with solutions obtained by an indirect method.

Keywords: Pursuit-evasion games, optimal control, bilevel programming, aerospace applications.

Applying nonlinear programming to pursuit-evasion games

Harri Ehtamo and Tuomas Raivio

Systems Analysis Laboratory

Helsinki University of Technology

P.O. Box 1100, 02015 HUT, FINLAND

Abstract. Motivated by the benefits of discretization in optimal control problems we consider possibilities to discretize pursuit-evasion games. Two approaches are introduced. In the first one, the solution of the necessary conditions of the continuous-time game is decomposed into ordinary optimal control problems that can be solved using discretization and nonlinear programming techniques. In the second approach, the game is discretized and transformed into a bilevel programming problem which is solved using a first order feasible direction method. Although the starting point of the approaches is different, they lead in practice to the same solution algorithm. We demonstrate the usability of the discretization by solving some open-loop representations of feedback solutions for a complex pursuit-evasion game between a realistically modeled aircraft and a missile with terminal time as the payoff. The solutions are compared with solutions obtained by an indirect method.

Keywords: Pursuit-evasion games, optimal control, bilevel programming, aerospace applications.

1 Introduction

A pursuit-evasion game is an ideal model for many antagonistic aerospace scenarios involving two parties with opposite goals. Games with simple dynamics and low dimensional state space can be solved analytically in the whole state space using the 'tenet of transition' [1]. The solution provides optimal feedback strategies of the players under the presumption that the opponent behaves in the worst possible way. More complex game models are often simplified by linearization or singular perturbation techniques to allow for analytical solutions, see, e.g., Ref. [2].

Without any simplifications, open-loop representations of feedback strategies (see Ref. [3]) can be calculated by setting up and solving the necessary conditions of a saddle point numerically [4, 5, 6]. In practice, a complex, high dimensional multipoint boundary value problem must be solved, although some gradient-based approaches have also been proposed [7], [8]. Approximate feedback strategies can then be synthesized on the basis of a cluster of open-loop representations corresponding to different initial states.

This paper is concerned with the numerical solution of complex pursuit-evasion games without explicitly setting up and solving the problem-specific necessary conditions of a saddle point. In Ref. [9], a decomposition approach was introduced. The method is based on the fact that for certain pursuit-evasion games of degree, the solution of the necessary conditions of a maxmin solution can be decomposed into subproblems that are solved iteratively. The subproblems can be represented as one player optimal control problems that can be solved using any suitable method, including discretization and nonlinear programming techniques. This paper proceeds in a reverse manner. The game dynamics is discretized at the outset and the saddle point problem is analyzed in a discretized framework. The discretized problem turns out to be a special case of bilevel programming problems that recently have been studied extensively, see Refs. [10, 11, 12] and the literature cited therein. For the solution of the bilevel programming problem, a feasible direction method that utilizes the special form of the problem is constructed. Both the feasible direction method and the decomposition approach mentioned above lead to almost the same subproblems and iteration. Since the bilevel approach allows one to address the convergence issues of the computation, it thus can be used to justify also the decomposition approach.

The study is motivated by the benefits of discretization and nonlinear programming in optimal control compared to solving the necessary conditions. Discretization schemes like collocation and Hermite interpolation [13, 14], Runge-Kutta integration formulas [15] or differential inclusions [16], can be used to convert an optimal control problem into a nonlinear programming problem, whose solution approximates the solution of the original problem up to the accuracy that depends on the scheme and discretization interval. For comprehensive reviews, see Refs. [17] and [18]. The necessary conditions are not directly involved in the solution process, but the solution and the corresponding Lagrange multipliers of the problem can be shown to approximately satisfy the necessary optimality conditions. Neither an initial estimate for the adjoints nor a hypothesis of the correct sequence of free, constrained and singular solution arcs, also known as the switching structure, is needed in advance. The convergence domain of the direct methods has turned out to be substantially larger than that of indirect methods, and continuation procedures with long homotopy chains are avoided. Furthermore, the sparsity of the discretized problem allows one to solve huge problems with moderate computational effort [17]. In Ref. [19], discretization and nonlinear programming were successfully applied to automating aircraft trajectory optimization.

The abovementioned benefits apply to specific discretized game models to a great extent as well. For example, when discretization is used in the solution process, the active constraints and singular game solution arcs that concern one player are automatically detected and treated. Nevertheless, contrary to optimal one-player trajectories, a saddle point solution can in principle involve such singular solution arcs and necessary conditions that will not be satisfied by the approaches presented here. To date, there does not exist a systematical method to locate these surfaces but they have to be analyzed separately [3].

In the following, the class of pursuit-evasion games under consideration is defined and the decomposition method introduced in Ref. [9] is reviewed. Then, the saddle point problem is discretized and the feasible direction method to solve it is studied. Finally, the capability of the method is demonstrated by a complex numerical example describing a pursuit-evasion game between a realistically modeled aircraft and a missile in three dimensions. The aircraft has to obey a dynamic pressure limit and a minimum altitude constraint. In optimal control framework the former constraint leads to singular optimal controls. The same problem was treated in Ref. [20], using an indirect solution approach.

2 The game model

We consider pursuit-evasion games where the state equations are given by

$$\dot{x}(t) = \begin{pmatrix} \dot{x}_P(t) \\ \dot{x}_E(t) \end{pmatrix} = f(x(t), u_P(t), u_E(t), t) = \begin{pmatrix} f_P(x_P(t), u_P(t), t) \\ f_E(x_E(t), u_E(t)) \end{pmatrix}, \quad (1)$$

$$x(0) = x_0, \quad t \in [0, \infty). \quad (2)$$

Here $x_P(t) \in R^{N_P}$ and $x_E(t) \in R^{N_E}$ for all t. The subscripts P and E refer to the pursuer and the evader, respectively. The players have perfect information on the state of the game. For all t, the admissible controls $u_i(t)$ of the players belong to the constant sets $A_i \subset R^{p_i}, \;\; i = P, E$, that describe the box constraints of the problem. The admissible control functions u_P and u_E are assumed to be piecewise continuous functions that satisfy the mixed state and control inequality constraints

$$C_i(x_i(t), u_i(t)) \leq \bar{0} \quad (3)$$

and state inequality constraints

$$S_i(x_i(t)) \leq \bar{0}, \;\; i = P, E. \quad (4)$$

Several pursuit-evasion problems, where the players correspond to individual vehicles, belong to this category.

The pursuer's objective is to minimize a common payoff that the evader wants to maximize. The payoff is

$$J[u_P, u_E] = q(x(T), T). \quad (5)$$

The game ends when the vector $(x(t), t)$ enters the target set $\Lambda \subset R^{N_P+N_E} \times R^+$. The unprescribed final time of the game is defined as

$$T = \inf\{t \mid (x(t), t) \in \Lambda\}. \quad (6)$$

The target set Λ is closed. The boundary of Λ is an $N_P + N_E$-dimensional manifold in the space $R^{N_P+N_E} \times R^+$ and is given by the scalar equation

$$l(x(t), t) = 0, \tag{7}$$

often termed as the capture condition. The function l, as well as f, C_i, S_i, $i = P, E$, and q are assumed to be continuously differentiable in x and t.

In games of degree it is tacitly assumed that the initial state of the game belongs to the capture zone, that is, the pursuer can enforce a capture against any action of the evader. Consequently, $T < \infty$ always holds.

Denote by Γ_i, $i = P, E$, the sets of admissible feedback strategies of the players. A strategy pair is called admissible if and only if it drives any state within the capture zone to the terminal manifold (7) and satisfies the constraints (3) and (4). The function describing the upper or minmax value of the cost function $q(x(T), T)$, when the players start from (x, t) is given by

$$\overline{V}(x, t) = \min_{\gamma_P \in \Gamma_P} \max_{\gamma_E \in \Gamma_E} q(x(T), T), \tag{8}$$

where γ_P and γ_E denote the strategies of the players. Similarly, the lower or maxmin value is given by

$$\underline{V}(x, t) = \max_{\gamma_E \in \Gamma_E} \min_{\gamma_P \in \Gamma_P} q(x(T), T). \tag{9}$$

In (8) it is understood that we first solve the max problem for a given γ_P and then optimize with respect to $\gamma_P \in \Gamma_P$. In (9), the opposite is assumed.

If either $\underline{V}$ or $\overline{V}$ exists and is continuously differentiable on the solution trajectory, then the other one also exists, they coincide and the min and max operators commute. This is because the state equation (1) and the constraints (3) and (4) are separable in u_P and u_E, and the payoff is terminal, see Ref. [3], Th. 8.1.

3 Decomposition of the solution of the necessary conditions

In this section we briefly describe a way to decompose the solution of the necessary conditions of the game problem into two separate optimal control problems that are solved iteratively until the necessary conditions are satisfied. The presentation follows Ref. [9]. We will shorten the presentation by assuming that the inequality constraints (3) also include the explicit state inequality constraints (4).

Suppose that the pursuit-evasion game described above admits a saddle point in feedback strategies. An open-loop representation of the feedback saddle point trajectories can be computed, e.g., by solving the necessary conditions of the minmax, or equivalently, the maxmin problem, defined by

$$\begin{aligned} \max_{u_E} \min_{u_P} \quad & q(x(T), T) \\ \dot{x}(t) &= f(x(t), u_P(t), u_E(t), t), \quad x(0) = x_0 \\ C_i(x_i(t), u_i(t)) &\leq \bar{0}, \quad i = P, E, \\ l(x(T), T) &= 0, \end{aligned} \tag{10}$$

where min and max are taken over all admissible controls u_P and u_E, respectively. The step from (9) to (10) in fact changes the information structure into open-loop. Since a feedback solution is difficult to compute, the necessary conditions are usually derived for an open-loop realization of a feedback solution corresponding to a given initial state and solved (for the computation of feedback solutions using the Isaacs equation and the theory of viscosity solutions, see Ref. [21], ch. 8). It is relatively straightforward to check that if the saddle point exists, the necessary conditions of an open-loop representation of the feedback solution of (9), given e.g., in [3], contain the necessary conditions of (10). The solution satisfying these necessary conditions is a candidate saddle point solution. However, in some cases there does exist solutions to (10) that are *not* realizations of feedback saddle point strategies; see Ref. [9] and the discussion in the end of the section.

Both (9) and (10) can also be viewed as feedback and open-loop Stackelberg differential games, respectively, where the evader acts as the leader and the pursuer as the follower. In (8), the roles of the players are reversed. In (9), the evader fixes his strategy optimally, and the pursuer optimizes against this strategy. In (10), the evader fixes his time dependent controls optimally, and the pursuer optimizes against these controls. The former problem is in practice impossible to solve (e.g. [3]) whereas the latter can be solved. Here, the Stackelberg solution naturally equals the maxmin solution.

In Ref. [9] the following method was adopted to solve the maxmin problem. First consider the minimization problem P with a given trajectory $x_E^0(t)$ of the evader:

$$\begin{aligned} \mathrm{P} : \min_{u_P, T} \quad & q(x_P(T), x_E^0(T), T) \\ \dot{x}_P(t) &= f_P(x_P(t), u_P(t), t), \quad x(0) = x_{P0} \\ C_P(x_P(t), u_P(t)) &\leq \bar{0} \\ l(x_P(T), x_E^0(T), T) &= 0. \end{aligned} \tag{11}$$

Let the solution of P be $\bar{u}_P(t)$ with the final time $\bar{T}$. Let $\bar{x}_P(t)$ be the corresponding state trajectory. Now, $\bar{u}_P(t)$ also solves the problem where the evader's trajectory in problem P is replaced by the fixed capture point $\bar{e} = x_E^0(\bar{T}) \in R^{N_E}$ and the final time is fixed to $\bar{T}$. In the neighborhood of $(\bar{e}, \bar{T})$, we define the value function of problem P, corresponding to an initial state x_{P0}, as a function of the capture point (e, T) by

$$\begin{aligned} \tilde{V}(e, T) = \quad & \min_{u_P} \{ q(x_P(T), e, T) \mid \dot{x}_P(t) = f_P(x_P(t), u_P(t), t), \ t \in [0, T], \\ & x_P(0) = x_{P0}; C_P(x_P(t), u_P(t)) \leq \bar{0}, \ l(x_P(T), e, T) = 0 \}. \end{aligned} \tag{12}$$

By definition,

$$\tilde{V}(\bar{e},\bar{T}) = q(\bar{x}_P(\bar{T}),\bar{e},\bar{T}). \tag{13}$$

The maxmin problem now essentially reduces to maximizing $\tilde{V}(x_E(T),T)$ subject to evader's constraints. This problem is difficult to solve since $\tilde{V}(\cdot)$ cannot be expressed analytically. Therefore we use the following iterative procedure. We first fix the final time to $\bar{T}$, linearize $\tilde{V}(e,\bar{T})$ in the neighborhood of $\bar{e}$ and solve the free final state problem E,

$$\begin{aligned} \mathrm{E}:\max_{u_E} \quad & \nabla_e \tilde{V}(\bar{e},\bar{T})'(x_E(\bar{T})-\bar{e}) \\ \dot{x}_E(t) &= f_E(x_E(t),u_E(t)),\ t\in[0,\bar{T}],\ x_E(0)=x_{E0} \\ C_E(x_E(t),u_E(t) &\leq \bar{0}, \end{aligned} \tag{14}$$

where the prime denotes a transpose. Basic sensitivity results applied to (12) (see Refs. [9] and also [22]) imply that the gradient of $\tilde{V}$ with respect to e at $(\bar{e},\bar{T})$ is given by the following analytical expression:

$$\nabla_e \tilde{V}(\bar{e},\bar{T}) = \nabla_e q(\bar{x}_P(\bar{T}),\bar{e},\bar{T}) + \bar{\alpha}\nabla_e l(\bar{x}_P(\bar{T}),\bar{e},\bar{T}), \tag{15}$$

where $\bar{\alpha}$ is the Lagrange multiplier associated with the capture condition in the solution of problem P.

Denote the solution of E by $u_E^1(t)$ and the corresponding state trajectory by $x_E^1(t)$, $t \geq 0$. Also denote $T^1 = \bar{T}$, and use a linear approximation

$$x_E^1(T^1+h) = x_E^1(T^1) + \dot{x}_E^1(T^1)h, \quad h>0, \tag{16}$$

for $t = T^1 + h > T^1$. The solution trajectory $x_E^1(t)$, extended by (16) is inserted into problem P and the procedure is repeated, until $\tilde{V}(\cdot)$ cannot be improved any more. In [9] it is shown that if the iteration converges, the limit solution of the iteration satisfies the necessary conditions of the maxmin problem described above.

The rationale behind the decomposition is that the subproblems in the procedure are ordinary optimal control problems. They can be solved using either indirect methods or discretization and nonlinear programming. If the latter approach is adopted, the complicated, problem dependent necessary conditions are not directly involved in the solution process, but the solution of the discretized problem can be shown to approximately satisfy them, see, e.g., [17]. The discrete optimal state and control variables approximate the optimal state and control trajectories, and the Lagrange multipliers approximate the adjoint trajectories. Consequently, the correct sequence of free, constrained and singular solution parts of the subproblems will be automatically roughly approximated, without the need to guess it in advance.

In most cases, the maxmin solution equals the saddle point solution. Nevertheless, a saddle point solution can, in general, possess also properties that a maxmin solution does not have. For example, a saddle point solution can involve certain singular surfaces, like the equivocal surface, that do not have a counterpart in the calculus of variations [1]. These surfaces require additional necessary conditions to be satisfied that cannot be expressed as the necessary optimality conditions of the subproblems above. A separate analysis would be needed to identify these surfaces. It should be noted that this fact is not related to the solution method used in solving the subproblems.

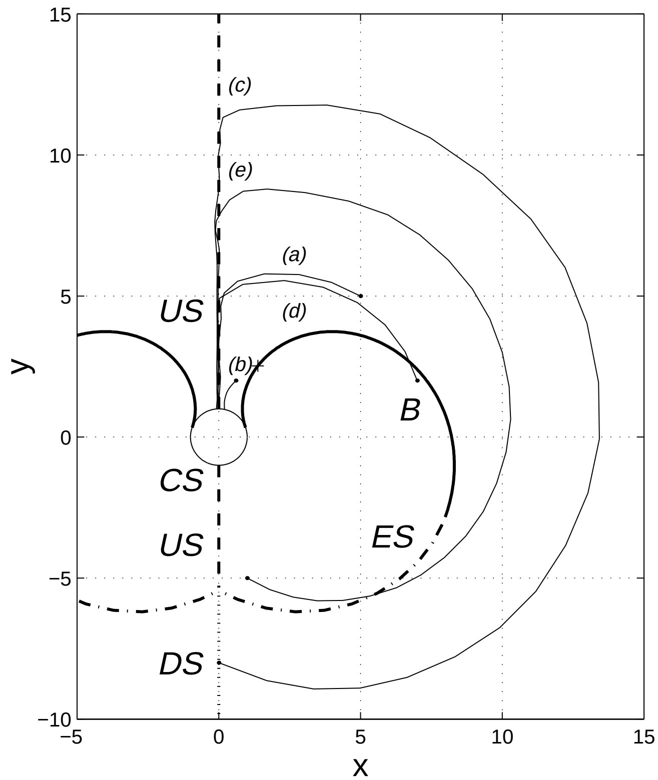

Figure 1: A collection of solutions to the homicidal chauffeur game computed by the decomposition method.

We demonstrate the above with a simple numerical example. Figure 1 presents a collection of solutions obtained by the method for the classical 'homicidal chauffer' pursuit-evasion game [1]. The axes refer to the reduced coordinates that describe the relative position of a car (the pursuer) and a pedestrian (the evader). The pedestrian can change his direction instantaneously, but the car has a nonzero minimal turning radius. The circle denoted CS describes the capture set, where the distance of the players is less than a predetermined capture radius, and the game terminates. For details, see [1] and [9].

The subproblems P and E of the iteration are discretized and solved using the same approach as in Ref. [9], originally given in Ref. [13]. Solution (a) is a saddle point solution that consists of a turn towards the evader and a tail chase along a universal singular surface, denoted by US. Note that this surface is adequately identified. Solution (b) is also a saddle point solution that starts from below the so called 'decision point', marked with a cross, from the region where solution trajectories never reach the US. Solution (c) starts from the dispersal singular surface DS. To which side of the surface the game evolves, depends on the initial estimate of the evader's trajectory. Both sides would produce the same value. Solution (d), however, crosses the barrier B, and solution (e) crosses the equivocal surface ES in an inadequate manner. They both are maxmin solutions. The tributary paths of the equivocal surface ES are not located by the method, since they are not solutions to problems P or E. The appearance and the analysis of these surfaces and related phenomena are problem dependent. The game should be solved separately on both sides of the surfaces, and the solutions should be pieced together with suitable necessary conditions.

4 Discretized game model

We next discretize the dynamics of the players to transform the original continuous time problem defined in Section 2 into a parametric constrained saddle point problem at the outset. In practice, it is possible to discretize the control variables only and explicitly integrate the equations of motion, discretize both state and control variables and carry out the integration implicitly, or eliminate the controls using approximate sets of reachability, see Ref. [18]. We prefer discretizing both state and control variables, which is often called direct transcription.

For simplicity, let $0 = t_0 < t_1 < \ldots < t_M = T$ be an equidistant time partition and denote the state and control variables at time t_j by $x_P(t_j) = x_{Pj}$, $u_P(t_j) = u_{Pj}$, $x_E(t_j) = x_{Ej}$ and $u_E(t_j) = u_{Ej}$, $j = 0, \ldots, M$. Also nonequidistant grids and separate grids for both players can be used. Denote the vectors of decision variables as $X = [u_{E0}, u_{E1}, \ldots, u_{EM}, x_{E1}, x_{E2}, \ldots, x_{EM}]$, and $Y = [u_{P0}, u_{P1}, \ldots, u_{PM}, x_{P1}, x_{P2}, \ldots, x_{PM}]$. The discretized saddle point problem can then be written as

$$\begin{aligned} &\max_{X} \min_{Y,T} q(x_{PM}, x_{EM}, T) \qquad (17)\\ &g_P(Y) \leq \bar{0}, \;\; h_P(Y,T) = \bar{0} \\ &g_E(X) \leq \bar{0}, \;\; h_E(X,T) = \bar{0} \\ &l(x_{PM}, x_{EM}, T) = 0. \end{aligned}$$

Here, the constraints $h_P = \bar{0}$ and $h_E = \bar{0}$ replace the state equations (1) and approximate them on the discretization intervals. For example, if Euler discretization is used, the functions h_P and h_E consist of vectors

$$\begin{aligned} &h_{Pj}(x_{Pj}, x_{P,j-1}, u_{P,j-1}, T) = x_{Pj} - x_{P,j-1} - T/M f_P(x_{P,j-1}, u_{P,j-1}, (j-1)T/M), \\ &h_{Ej}(x_{Ej}, x_{E,j-1}, u_{E,j-1}, T) = x_{Ej} - x_{E,j-1} - T/M f_E(x_{E,j-1}, u_{E,j-1}), \\ &j = 1, \ldots, M. \qquad (18) \end{aligned}$$

For Hermite interpolation with piecewise cubic polynomials and collocation at the middle of each discretization interval, the corresponding constraints are given, e.g., in Ref. [13]. The functions g_P and g_E consist of the state and control inequality constraints (3) and (4), evaluated at (x_{ij}, u_{ij}),

$$g_i(\cdot) = [C_i(x_{i0}, u_{i0})', S_i(x_{i0})', ..., C_i(x_{iM}, u_{iM})', S_i(x_{iM})']', \;\; i = P, E. \qquad (19)$$

The problem above approaches the original continuous time problem, as the discretization intervals approach zero. Using arguments given in Ref. [17], p. 197, it can be shown that also the necessary conditions of a saddle point of the discretized problem approach those of the continuous time problem, given e.g., in Ref. [3].

5 A feasible direction method

We will first consider a problem of the form

$$\begin{aligned} &\max_X \min_{Y,T} q(X,Y,T) \\ &g_P(Y) \le \bar{0}, \;\; h_P(Y,T) = \bar{0} \\ &g_E(X) \le \bar{0}, \;\; h_E(X,T) = \bar{0} \\ &l(X,Y,T) = 0, \end{aligned} \tag{20}$$

where $X \in R^n$, $Y \in R^m$, $T \in R$ and l is a scalar function. In (20) it is understood that for a given X we first choose $Y(X)$ and $T(X)$ to minimize q subject to all constraints containing them. The max problem then takes the form

$$\begin{aligned} &\max_X q(X, Y(X), T(X)) \\ &g_E(X) \le \bar{0}. \end{aligned}$$

Suppose (X^*, Y^*, T^*) is a *local* maxmin solution and suppose that for the min problem, with X considered as a parameter, the appropriate differentiability and regularity conditions (namely, the strong second order sufficiency, linear independency and strict complementary slackness) specified in the basic sensitivity theorems (Theorems 3.2.2 and 3.4.1 in Ref. [22]) hold at (X^*, Y^*, T^*). Then, near X^* it holds that $Y(X)$, $T(X)$ (with $Y(X^*) = Y^*$, $T(X^*) = T^*$) and the corresponding Lagrange multipliers for the min problem are unique and continuously differentiable functions of X. Furthermore, there are Lagrange multipliers $\lambda_P \ge \bar{0}$, $\lambda_E \le \bar{0}$, μ_P, μ_E and $\alpha \in R$, such that the Lagrangian $L(X,Y,T)$,

$$L = q + \alpha l + \lambda'_P g_P + \mu'_P h_P + \lambda'_E g_E + \mu'_E h_E \tag{21}$$

is stationary at X^*, Y^*, T^*, and $\lambda'_P g_P = \lambda'_E g_E = 0$. These are the Karush-Kuhn-Tucker (KKT) conditions for the maxmin problem (20). Similar conditions, under weaker assumptions, can be derived in terms of subgradients, see, e.g., Chapter 9 of Ref. [12].

Note that the place of T in the maxmin problem is irrelevant when deriving the necessary conditions, provided of course, that the appropriate assumptions are made and the constraint equations are handled consistently (we will make use of this fact in Section 5.1). This convenience will no more follow if the inequality constraints $g_P \le 0$ and $g_E \le 0$ depend explicitly on T, since the corresponding terms in the Lagrangian (21) would then have opposite signs for the two cases. Now suppose that X^*, Y^*, T^* satisfies the KKT conditions. If the conditions of the basic sensitivity theorems hold at X^*, Y^*, T^* for the min problem when X is considered as a parameter, and for the max problem when Y is considered as a parameter, then it can be concluded that X^*, Y^*, T^* is a *local* saddle point solution. In this paper we are interested in solving the KKT necessary conditions numerically using the maxmin problem as our starting point.

Problem (20) is a special case of a bilevel programming problem where the minimization and the maximization are identified with the lower level and the upper level problems, respectively (for an introduction to the subject, see Refs. [11] and [23]; for a more advanced treatment see Ref. [12]). In a general case, the upper and the lower level problems have different objective functions.

The solution approaches are divided into three classes [11]. First, one can take the necessary optimality conditions of the lower level problem as constraints to the upper level problem. Nevertheless, the complementary slackness condition usually makes the latter problem complicated and most likely nondifferentiable. The second class of algorithms uses penalty functions to transform the lower and the upper level optimizations into sequences of unconstrained optimizations. The convergence rate of these algorithms is, especially in the nonlinear case, regarded unsatisfactory.

The algorithms of the third class form perhaps the most appealing approach. In these algorithms, all the constraints are included into the lower level problem that is solved for given X, and gradient information on the solution is ackquired using basic sensitivity results. In the upper level, some ascent method together with a line search is applied using this information. In general, the lower level solution can be nondifferentiable, in which case methods of nonsmooth optimization (see Refs. [12, 24]) might be required.

Our approach falls mainly into the third class. Nonetheless, pushing all the constraints into the lower level leads to an uncomfortably large optimization problem, that in addition has to be solved several times in each iteration due to the line search. Here we avoid these problems by utilizing the nature of the maxmin problem and by careful handling of the constraints.

It should be mentioned that so far, the literature on minmax problems has mainly dealt with the nonparametric constrained case, where there are only separate constraints for the upper and lower level problems; see [12] and the references therein. Recently also parametric constrained problems have been considered in the context of subgradient (nondifferentiable) optimization [12].

In our application we have only one nonseparable scalar equation constraint and, based on numerical tests with the kind of problems, we do not expect difficulties with differentiability (see also the discussion in the introduction of [11]). We therefore present our theory by assuming differentiability when needed, e.g., by assuming the regularity conditions mentioned ahead. Nevertheless, it should not be difficult to apply any subgradient based method, like Mifflin's algorithm [12], too.

5.1 The Method

To handle the constraints appropriately, we put T under the maximization (recall the previous discussion) and rewrite problem (20) in the form

$$\begin{aligned} \max_{X,T} \quad & V(X,T) \\ g_E(X) & \leq \bar{0} \\ h_E(X,T) & = \bar{0}, \end{aligned} \tag{22}$$

where

$$V(X,T) = \min_Y \{q(X,Y,T) \mid g_P(Y) \leq \bar{0}, h_P(Y,T) = \bar{0}, l(X,Y,T) = 0\}. \tag{23}$$

Hence, problem (22) contains only the evader's constraints and problem (23) the pursuer's constraints together with the scalar terminal constraint l.

In principle, problem (22) could be solved with any feasible direction method of nonlinear optimization. When computing a feasible ascent direction, the gradient of $V(X,T)$ with respect to X and T is needed. It depends on the current solution $Y(X,T)$ of (23), but not on its derivative. This is a consequence of the KKT conditions of (23) and the basic sensitivity theorems. Hence, at each iteration of the evader's (max) problem, the pursuer's (min) problem must be solved at least once. Additional solutions are required if a line search is applied.

The method outlined here consists of the following steps: i) Given a point (X^k, T^k), generate an improving, but not necessarily feasible direction; ii) perform a line search along this direction; iii) perform a correction move back to the feasible region. We next give a detailed definition of iteration k. Here, the gradients are defined as column vectors and the Jacobians are formed using transposed gradients. Let (X^k, T^k) satisfy $g_E(X^k) \leq \bar{0}$, $h_E(X^k, T^k) = \bar{0}$. We linearize the constraints at (X^k, T^k) and choose a direction $(d^k, \delta^k) \neq (\bar{0}, 0)$ such that

$$\nabla_X g_{Ei}(X^k)' d^k = 0, \quad i \in I_E, \tag{24}$$

$$\nabla_X h_E(X^k, T^k) d^k + \partial_T h_E(X^k, T^k) \delta^k = \bar{0}, \tag{25}$$

where in (24) $I_E := \{i | g_{Ei}(X^k) = 0\}$, that is, only active inequality constraints are present. One should also make sure that $\nabla_X V(X^k, T^k)' d^k + \partial_T V(X^k, T^k) \delta^k > 0$ for (d^k, δ^k) to be an improving direction. A systematic way to do this is to solve a direction finding subproblem such as that in SQP or in Zoutendijk's feasible direction method, see, e.g., Ref. [26], Ch. 10.

We next perform a line search in the direction (d^k, δ^k). The constraints g_{Ei} and h_E can be ignored in the search; the feasibility will be guaranteed by the correction move in the next step. Consequently, one possibility would be to solve $\max V((X^k, T^k) + t(d^k, \delta^k))$ over $t \geq 0$. In principle, the solution can be computed by any suitable line search method. Nevertheless, these methods require that problem (23) be solved numerous times with different values of t.

Instead of doing this we take the minimum of V with respect to $t \geq 0$. It can be obtained with a single solution of (23) which is needed anyway. For the convergence of the method it is required that the line search map is closed, which is the case if it is defined either by maximization or minimization (e.g. [26], Ch. 8).

Problem min $V((X^k, T^k) + t(d^k, \delta^k))$, $t \geq 0$, can be written as

$$\begin{aligned} \text{PD: } \min_{Y, t \geq 0} \quad & q(X^k + td^k, Y, T^k + t\delta^k) \\ g_P(Y) & \leq \bar{0} \\ h_P(Y, T^k + t\delta^k) & = \bar{0} \\ l(X^k + td^k, Y, T^k + t\delta^k) & = 0. \end{aligned} \tag{26}$$

The solution $(\bar{Y}, \bar{t})$, $\bar{t} > 0$, of this problem satisfies

$$\nabla_Y q + \alpha \nabla_Y l + \nabla_Y g_P' \lambda_P + \nabla_Y h_P' \mu_P = \bar{0} \tag{27}$$
$$(\nabla_X q + \alpha \nabla_X l)' d^k + (\partial_T q + \alpha \partial_T l + \mu_P' \partial_T h_P)\delta^k = 0 \tag{28}$$
$$h_P = \bar{0} \tag{29}$$
$$l = 0 \tag{30}$$
$$\lambda_P' g_P = 0 \tag{31}$$
$$\lambda_P \geq \bar{0}, \tag{32}$$

evaluated at $(X^k + \bar{t}d^k, \bar{Y}, T^k + \bar{t}\delta^k)$. Here $\alpha \in R$, and λ_P and μ_P are Lagrange multiplier vectors of appropriate dimensions. Define $\bar{T} = T^k + \bar{t}\delta^k$, $\bar{X} = X^k + \bar{t}d^k$. Then, using basic sensitivity results,

$$V(\bar{X}, \bar{T}) = q(\bar{X}, \bar{Y}, \bar{T}) \tag{33}$$
$$\nabla_X V(\bar{X}, \bar{T}) = \nabla_X q(\bar{X}, \bar{Y}, \bar{T}) + \alpha \nabla_X l(\bar{X}, \bar{Y}, \bar{T}). \tag{34}$$

If the minimum of V on the line is attained at a boundary point $t = 0$, the iteration may stop at a nonoptimal point. In this case the interval should be extended to the left; take, e.g., $t \geq -a$, $a > 0$. In the numerical tests with linear and quadratic costs and constraints typical to the problems of Section 2, there exists an interval $[0, b]$ for t such that the points are infeasible for the pursuer. In this case the minimum of V along the direction (d^k, δ^k) is obtained at a point $t \geq b$. Yet the value of V increases from $V(X^k, T^k)$.

Recall that since (d^k, δ^k) is tangential to the active nonlinear constraints at (X^k, T^k), the new point $(\bar{X}, \bar{T})$ is not necessarily feasible for the evader. Therefore, we finally make a correction move to get back to the feasible region and to further improve the value of V by solving

$$\begin{aligned} \text{ED: } \max_{X} \quad & \nabla_X V(\bar{X}, \bar{T})'(X - \bar{X}) \\ g_E(X) & \leq \bar{0} \\ h_E(X, \bar{T}) & = \bar{0}, \end{aligned} \tag{35}$$

where expression (34) is used. ED is solved in each iteration whether $(\bar{X}, \bar{T})$ is feasible or not. Denote the solution of ED by X^{k+1} and put $T^{k+1} := \bar{T}$. Then

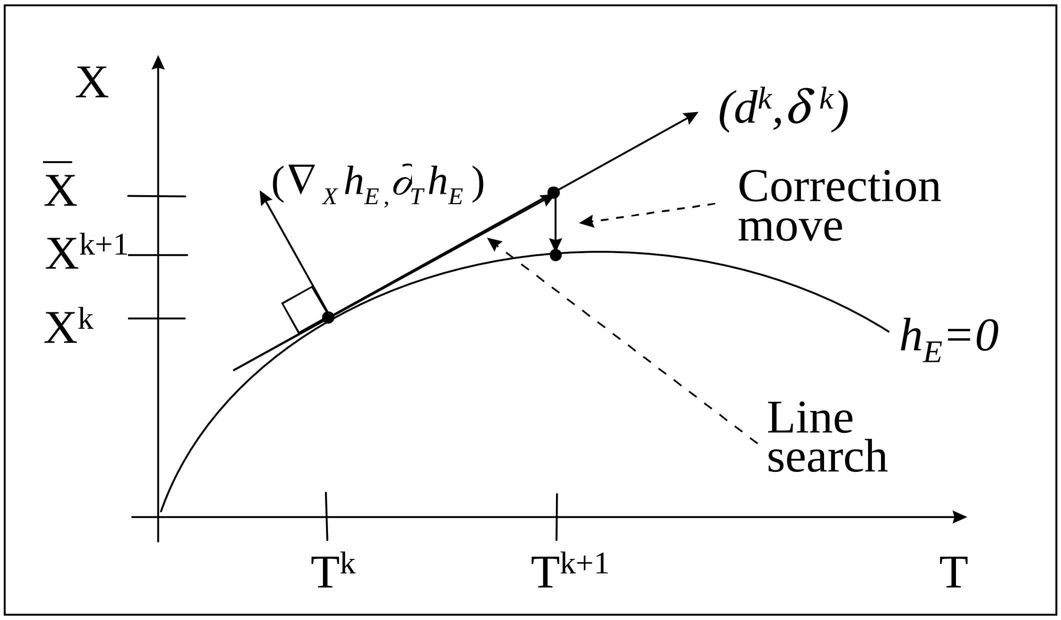

Figure 2: An illustration of the iteration.

(X^{k+1}, T^{k+1}) satisfies the necessary conditions

$$\nabla_X V(\bar{X}, \bar{T}) + \nabla_X g_E' \lambda_E + \nabla_X h_E' \mu_E = \bar{0} \quad (36)$$

$$h_E = \bar{0} \quad (37)$$

$$\lambda_E' g_E = 0, \quad (38)$$

$$\lambda_E \leq \bar{0}, \quad (39)$$

where λ_E and μ_E are Lagrange multiplier vectors. The iteration is continued until (X^{k+1}, T^{k+1}) is close enough to (X^k, T^k). The phases of the iteration are illustrated in Figure 2. To summarize, the iteration proceeds as follows:

1. Choose X^0 and T^0, and a direction (d^0, δ^0) satisfying (24) and (25). Set $k := 0$.
2. Solve problem PD using X^k, T^k and (d^k, δ^k) to obtain $\bar{t}$ and α that is needed in (34).
3. Solve problem ED to obtain X^{k+1} and T^{k+1}. If the relative change of (X^{k+1}, T^{k+1}) from (x^k, T^k) is small enough, terminate. Otherwise, set $k := k + 1$, select a (d^k, δ^k) satisfying (24) and (25) and return to 2.

5.2 Convergence

Suppose the iteration converges. Then the limit solution (X^*, Y^*, T^*) together with the corresponding Lagrange multipliers satisfies the KKT conditions for the maxmin problem (20). Namely, define the Lagrangian as in (21). Then (27), (34) and (36) show that the gradients of the Lagrangian with respect to X and Y vanish at

(X^*, Y^*, T^*). Next, the second term in (36) is $\sum \nabla_X g_{Ei}\lambda_{Ei}$ where the sum is over the active inequality constraints ($i \in I_E$) only, since $\lambda_{Ei} = 0$ otherwise. Multiplying (36) by $d^{k\prime}$, and using (24) multiplied by λ'_{Ei}, $i \in I_E$, (25) multiplied by μ'_E, and (34) we get that the first term of (28) equals $\mu_E \partial_T h_E$. Since $\delta^k \neq 0$ by definition, (28) shows that also the time derivative of the Lagrangian vanishes at (X^*, Y^*, T^*). Eqs. (31), (32), (38) and (39) give the appropriate complementary slackness conditions.

Under what circumstances does the iteration converge? To answer this question we here adopt the general argumentation used for all feasible direction methods. We will discuss these conditions also in section 5.3, when using the discretized model of section 4, and in connection with the numerical example.

Suppose the generated sequence $\{(X^k, T^k)\}$ is contained in a compact set and suppose $V(X^{k+1}, T^{k+1}) > V(X^k, T^k)$ for all k. Suppose further that, in the region of interest, both the correction move C defined by (35), and the direction D defined in (24), and (25) depend continuously on the current point. Letting t to belong to some sufficiently large interval $[a, b]$, possibly with $a < 0$, it can be shown that the line search algorithmic map M defined by the minimization problem (26) is a closed map (see, Ref. [26], Theorem 8.4.1). It then follows that the overall algorithmic map consisting of M, D, and C is also a closed map, and the convergence follows.

5.3 Solving the discretized model

We now return to the discretized pursuit-evasion game model described in Section 4. Instead of using conditions (24) and (25) in determining the direction (d^k, δ^k) we use weaker conditions

$$\sum_{i \in I} \lambda_{Ei} \nabla_X g_{Ei}(X^k)' d^k = 0, \quad (40)$$

$$(\nabla_X h_E(X^k, T^k)'\mu_E)' d^k + (\partial_T h_E(X^k, T^k)'\mu_E)\delta^k = 0, \quad (41)$$

that are actually sufficient in ensuring the KKT-point in the case of convergence, see Section 5.2. In the following we omit the superscript k. Since the functions q and l only depend on the terminal states x_{EM} and x_{PM}, we can select the components of d corresponding to $u_{E0}, \ldots, u_{EM}$ and $x_{E0}, \ldots, x_{E,M-1}$ equal to zero. Without loss of generality, we can also normalize (d, δ) in (40) and (41) such that $\delta = 1$. Thus the equations above reduce to

$$\lambda'_{EM} \nabla_{x_{EM}} g_{EM}(x_{EM}, u_{EM}) d_M = 0 \quad (42)$$

$$\mu'_{EM} \nabla_{x_{EM}} h_{EM}(x_{EM}, x_{E,M-1}, u_{EM}, u_{E,M-1}, T) d_M = -\partial_T h_E(X, T)'\mu_E, \quad (43)$$

where $d_M \in R^{N_E}$ denotes the nonzero component of d, $g_{EM}(x_{EM}, u_{EM}) = [C_E(x_{EM}, u_{EM})', S_E(x_{EM})']'$ and λ_{EM} and μ_{EM} are the corresponding multiplier vectors. We next show that the conditions above are approximately satisfied by selecting $d_M = f_E(x_{EM}, u_{EM})$. Although we show this for Euler discretization (18), extending the result to other discretization schemes should be rather straightforward.

In (42), we first make the usual assumption that the control and state constraints described in $g_{EM}(\cdot)$ are active on a finitely long interval. Constraints that become active only at $t_M = T$ are considered inactive. Further, we assume that the previous discretization point belongs to this interval, that is, $g_{E,M-1}(\cdot) = \bar{0}$. Thus,

$$\bar{0} = g_{EM}(x_{EM}, u_{EM}) - g_{E,M-1}(x_{E,M-1}, u_{E,M-1}) \approx \\ \nabla_{x_{EM}} g_{EM}(x_{EM}, u_{EM}) \frac{x_{EM} - x_{E,M-1}}{T/M} + \nabla_{u_{EM}} g_{EM}(x_{EM}, u_{EM}) \frac{u_{EM} - u_{E,M-1}}{T/M}. \tag{44}$$

The evader's necessary optimality condition (36) states that

$$\lambda'_{EM} \nabla_{u_{EM}} g_{EM}(x_{EM}, u_{EM}) = \bar{0}. \tag{45}$$

Hence, multiplying (44) by λ'_{EM} and using (18) and (45) we obtain that $d_M = f_E(x_{EM}, u_{EM}) \approx f_E(x_{E,M-1}, u_{E,M-1})$ approximately satisfies (42).

Consider next equation (43). For the RHS we get, using (18),

$$-\partial_T \sum_{k=1}^{M} (x_{Ek} - x_{E,k-1} - T/M f_E(x_{E,k-1}, u_{E,k-1}))' \mu_{Ek} = \\ \partial_T \sum_{k=1}^{M} T/M f_E(x_{E,k-1}, u_{E,k-1})' \mu_{Ek} \approx \\ \partial_T \int_0^T f_E(x_E(t), u_E(t))' \mu_E(t) dt \approx f_E(x_{EM}, u_{EM})' \mu_{EM}. \tag{46}$$

Since $\nabla_{x_{EM}} h_{EM}(x_{EM}, x_{E,M-1}, u_{EM}, u_{E,M-1}, T) = I$, the result follows.

Thus $(f_E(x^k_{EM}, u^k_{EM}), 1)$ is a convenient choice for the direction (d^k, δ^k) at iteration k. Whether it is improving or not is a manifestation of the numerical computation for a specific example. It seems to work well at least in problems where the payoff is the terminal time, see below and also Ref. [9], or range minimaximization, see Ref. [25], leading to an increasing sequence $\{V(X^k, T^k)\}$.

Although the approach presented here is conceptually different from the decomposition of the necessary conditions presented in Section 3, The result above means that both approaches lead to almost the same subproblems. Clearly, problem ED is the discretized counterpart of problem E. Stated differently, when E is discretized using the same scheme as in the bilevel programming approach, problem ED results. Problem PD is the discretized counterpart of problem P, except that instead of the extended evader's trajectory (16), direction d_M, that can be specified by $d_M = f_E(x^k_{EM}, u^k_{EM})$ is used. This direction is, up to the discretization accuracy, parallel with the extension in (16). Thus, in the computations we do not have to distinguish between the methods.

6 A Missile-Aircraft Encounter

Consider an aircraft flying in three dimensions. At $t = 0$, a medium-range air-to-air missile is launched towards the aircraft. The objective of the missile is to hit the aircraft in minimum time, whereas the aircraft, noticing this, wishes to maximize the capture time. This encounter can be described as a pursuit-evasion differential game, in which the missile plays the role of the pursuer (P) and the aircraft the role of the evader (E).

Both P and E maneuver in three dimensions. Assuming point mass models, the equations of motion are (e.g. [27])

$$\dot{x}_i = v_i \cos\gamma_i \cos\chi_i \quad (47)$$

$$\dot{y}_i = v_i \cos\gamma_i \sin\chi_i \quad (48)$$

$$\dot{h}_i = v_i \sin\gamma_i \quad (49)$$

$$\dot{\gamma}_i = \frac{g}{v_i}(n_i \cos\mu_i - \cos\gamma_i) \quad (50)$$

$$\dot{\chi}_i = \frac{g}{v_i}\frac{n_i \sin\mu_i}{\cos\gamma_i}, \quad i = P, E; \quad (51)$$

$$\dot{v_P} = \frac{1}{m_P(t)}[T_P(t) - D_P(h_P, v_P, M(h_P, v_P), n_P)] - g\sin\gamma_P \quad (52)$$

$$\dot{v_E} = \frac{1}{m_E}[\eta_E T_{E,max}(h_E, M(h_E, v_E)) - D_E(h_E, v_E, M(h_E, v_E), N_E)] - g\sin\gamma_E \quad (53)$$

The state of the game is described by the state vector

$$[x_P, y_P, h_P, \gamma_P, \chi_P, v_P, x_E, y_E, h_E, \gamma_E, \chi_E, v_E]'. \quad (54)$$

The state variables x, y, h, γ, χ and v are the x and y coordinates, altitudes, flight path angles, heading angles, and velocities of the players. The gravitational acceleration g is assumed constant. $D(\cdot)$ denotes the drag force and $M(\cdot)$ the Mach number.

The direction of the vehicles is controlled by the loadfactor n and the bank angle μ, $\mu_i \in [-\pi, \pi]$, $i = P, E$. The velocity of E is controlled by the throttle setting $\eta_E \in [0, 1]$. The pursuer's thrust force $T_P(t)$ is a fixed function of time that cannot be controlled. For $0 \le t < 3$ [sec.], a powerful boost phase gives the missile an acceleration of several hundred $[m/s^2]$. The sustainer phase at $3 \le t < 15$ [sec.] then keeps up the march speed of the missile. The coasting finally takes place without any thrust. Consequently $m_P(t)$, the mass of the missile decreases piecewise linearly and remains then constant. The mass m_E of the evader is assumed constant. The maximum thrust force $T_{E,max}(\cdot)$ of the evader depends on the altitude and Mach number and is approximated by a two-dimensional polynomial.

The load factors n_P and n_E cannot be chosen freely. At low velocities, a large load factor requires a large angle of attack which results in loss of lift force and stall. At higher velocities, the magnitudes of the load factors are constrained by the largest accelerations that the flight vehicles and the pilot tolerate. Here the bounds are approximated by the box constraints

$$n_P \in [0, 20] \quad (55)$$
$$n_E \in [0, 7]. \quad (56)$$

The drag forces of E and P are assumed to obey a quadratic polar. Eliminating the lift coefficient yields for both vehicles (we suppress the subscripts for brevity)

$$D(h, v, M(h, v), n) = C_{D_0}(M(h, v))SQ(h, v) + n^2 C_{D_I}(M(h, v))\frac{(mg)^2}{SQ(h, v)}, \quad (57)$$

where C_{D_0} and C_{D_I} denote the zero drag and induced drag coefficients, S, not to be mixed with (4), here denotes the reference wing area, and $Q(h, v) = 1/2\varrho(h)v^2$ is the dynamic pressure. The drag coefficients are approximated by rational polynomials on the basis of realistic tabular data. The evader's data describes a high-performance fighter aircraft. The pursuer's data corresponds to a generic air-to-air missile. For details, the reader is urged to the literature cited in Ref. [20]. The air density and the velocity of the sound are taken from the standard ISA atmosphere.

The evader E has to stay in its flight envelope. In the present encounter, the boundaries of interest are the minimum altitude constraint

$$h_E \geq h_{min,E} \quad (58)$$

and the dynamic pressure constraint

$$Q(h_E, v_E) \leq Q_{E,max}. \quad (59)$$

The pursuer must obey the minimum altitude constraint

$$h_P \geq h_{min,P}. \quad (60)$$

This constraint remains likely inactive, provided that $h_{min,E} \geq h_{min,P}$. The initial value of the state vector (54) is specified at $t = 0$, and the game terminates as soon as the capture condition

$$l(\cdot) = (x_P(T) - x_E(T))^2 + (y_P(T) - y_E(T))^2 + (h_P(T) - h_E(T))^2 - d^2 = 0 \quad (61)$$

becomes satisfied. The parameter d denotes the capture radius, i.e., the final distance of the players. The payoff is the final time,

$$J[u_P, u_E] = T, \quad (62)$$

where $u_P = [n_P, \mu_P]'$ and $u_E = [n_E, \mu_E, \eta_E]'$.

7 Numerical Results

In the following the game dynamics described in the previous Section is discretized using direct transcription and the approach presented in Ref. [13] (see also Ref. [14]). In this approach, the solution time interval is divided into discretization segments or intervals using a suitable grid or time partition. The state and the control variables are assumed to be known only at the gridpoints. On each interval, the state trajectories are approximated by piecewisely defined cubic polynomials, and the state trajectories are approximated piecewise linearly. Both approximations are required to be continuous across the gridpoints. In addition, the cubics are required to be smooth in their first derivative. In the middle of each interval, the time derivative of the cubic is required to coincide with the value of the state equation.

Here, the evader's dynamics is discretized using an equidistant grid and 25 discretization points, which is here considered sufficient. The three phases of the pursuing missile are discretized separately by 5, 10, and 15 piecewisely equidistant grids and joined by appropriate continuity conditions on the state and the control variables. On the first two phases the discretization interval is constant, only the interval of the third phase varies. Thus, the nonlinear programming problem PD with (d^k, δ^k) defined by $(f_E(x^k_{EM}, u^k_{EM}), 1)$, contains 241 decision variables. Excluding the linear box constraints, initial conditions and continuity conditions, the problem contains 163 nonlinear constraints. Problem ED contains 225 decision variables and 169 nonlinear constraints. The bilevel programming problem thus has 466 decision variables and 332 nonlinear constraints. Both subproblems are moderate in size. They are solved by the NPSOL software [28] that uses the Sequential Quadratic Programming algorithm. The main iteration is repeated until a point satisfying the KKT necessary conditions is found.

For comparison, we first solve the problem from the same initial constellation as in Ref. [20], that is,

$$\begin{aligned} x_{P,0} &= 0 \text{ [m]} & x_{E,0} &= 12,500 \text{ [m]} \\ y_{P,0} &= 0 \text{ [m]} & y_{E,0} &= 1,000 \text{ [m]} \\ h_{P,0} &= 5,000 \text{ [m]} & h_{E,0} &= 6,000 \text{ [m]} \\ v_{P,0} &= 250 \text{ [m/s]} & v_{E,0} &= 400 \text{ [m/s]} \\ \gamma_{P,0} &= 0 \text{ [deg]} & \gamma_{E,0} &= 0 \text{ [deg]} \\ \chi_{P,0} &= 0 \text{ [deg]} & \chi_{E,0} &= 120 \text{ [deg]}. \end{aligned}$$

The largest allowed dynamic pressure for the evader is 80,000 [Pa]. The minimum altitude is set to 50 [m] for both parties. The capture radius is set to $d = 50$ [m].

We provide a readily feasible initial estimate (X^0, T^0) and the initial direction (d^0, δ^0) by specifying a feasible nominal trajectory for the evader. As such, we use a sharp turn away from the pursuer, towards the assumed direction of evasion. This initial estimate also serves as the initial guess for the first solution of problem ED. Overall, the convergence of the method does not overly seem to depend on the initial guess of the evader's nominal trajectory. Another initial guess is needed to initiate the solution of PD.

The method converges in 6 iterations within a relative accuracy of 10^{-3}. That is, 12 nonlinear programming problems are solved altogether. The sequence $\{T^k\}$ generated by the method increases monotonically as required in the convergence considerations above. The solutions of the previous subproblems are used as the starting points of iteration for the next subproblems. They usually lie in the convergence domain of the next subproblem, which guarantees a satisfactory convergence for the subproblems.

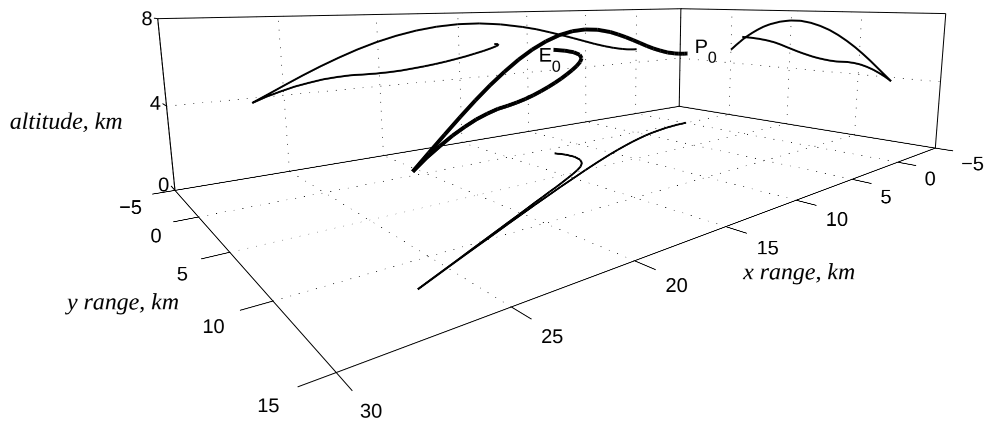

Figure 3: Optimal saddle point trajectories of the pursuer (initial state denoted by P_0) and the evader (E_0) in the first example. The projections of the trajectories are presented on the coordinate planes.

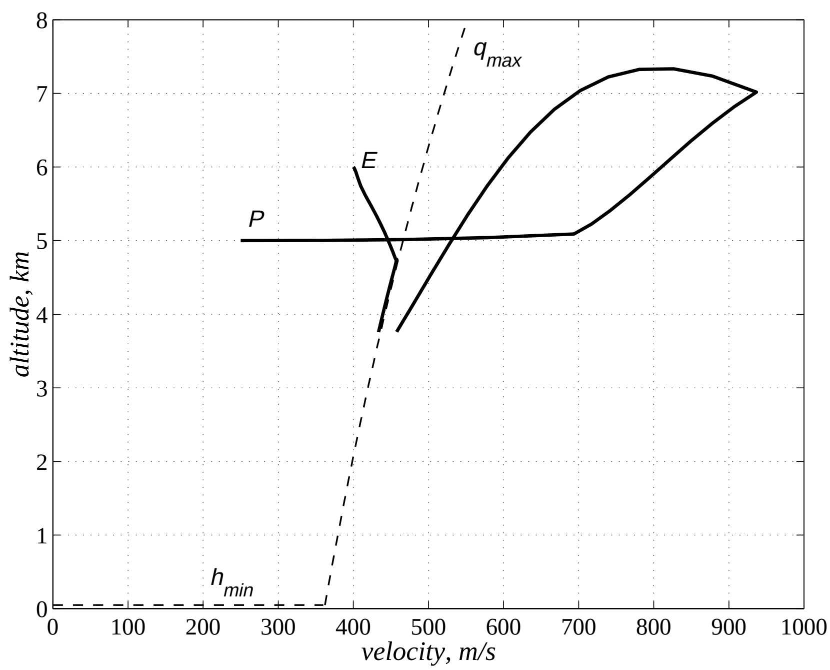

Figure 4: The altitude and velocity histories of the pursuer (P) and the evader(E). The dashed lines indicate the boundaries of E's flight envelope.

The solution trajectories are presented in Figure 3. During the boosting phase the pursuer turns towards the evader and starts to ascend. In the sustaining phase it climbs considerably more to decrease the drag force. The evader turns away from the pursuer and dives until the dynamic pressure constraint becomes active, see the altitude-velocity diagram in Figure 4, and Figure 5 (d). The constraint forces the evader to reduce its descent rate. Both trajectories lead to a common vertical plane. Before the capture takes place, the state of the game enters a singular surface induced by the dynamic pressure constraint of the evader. This can be seen from the evader's throttle setting that receives values between 0 and 1, see figure 5 (c).

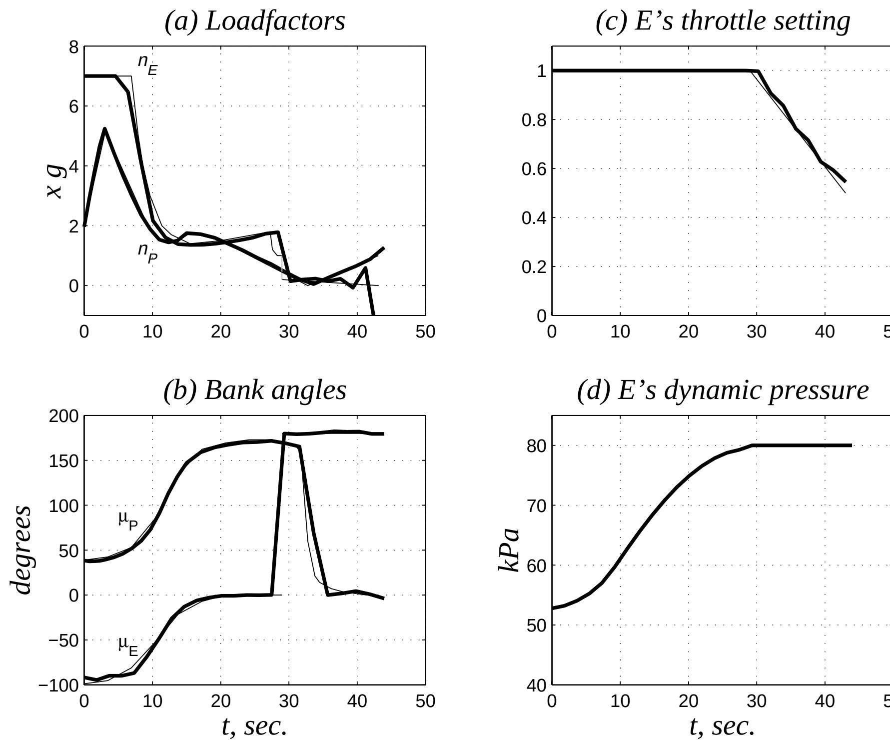

Figure 5: On the left, (a) the loadfactor, (b) the bank angle, histories of the pursuer (subscript P) and the evader (subscript E). On the right, (c) the throttle setting and (d) the dynamic pressure of the evader. The thin lines indicate the control histories obtained with an indirect method.

The computed saddle point controls of the players are shown in Figures 5 (a), (b) and (c), together with the control histories computed with an indirect method, taken from Ref. [20]. The pursuer's loadfactors coincide within the drawing accuracy. The evader's loadfactors deviate slightly, mainly due to the use of an equidistant discretization grid. Also the bank angles coincide well, except for the slight deviation at the pursuer's rapid (continuous) transient around $t \approx 33$ [sec.]. Note that the controls computed with the indirect approach are discontinuous at the beginning of the singular arc at $t \approx 29$ [sec.]. If a more precise solution were needed, this discontinuity could be explicitly taken into account in the later stages of the solution process. This would probably also remove the extra final transient in the evader's loadfactor.

The value of the game corresponding to this initial state is predicted to be 43.8 [sec.], which is approximately 0.8 [sec.] (1.8%) more than the value computed by the indirect method. The difference is most likely due to the deviations in the evader's loadfactor and could possibly be diminished, e.g., by using a larger amount of discretization points, an adaptive grid selection strategy and explicit modeling of the switching structure. Nevertheless, even as such the difference might be considered acceptable for most practical purposes.

The second example is assumed to begin from the following initial conditions:

$$\begin{aligned} x_{P,0} &= 0 \text{ [m]} & x_{E,0} &= 12,500 \text{ [m]} \\ y_{P,0} &= 0 \text{ [m]} & y_{E,0} &= 0 \text{ [m]} \\ h_{P,0} &= 1,000 \text{ [m]} & h_{E,0} &= 450 \text{ [m]} \\ v_{P,0} &= 250 \text{ [m/s]} & v_{E,0} &= 300 \text{ [m/s]} \\ \gamma_{P,0} &= 0 \text{ [deg]} & \gamma_{E,0} &= 0 \text{ [deg]} \\ \chi_{P,0} &= 0 \text{ [deg]} & \chi_{E,0} &= 179 \text{ [deg]}. \end{aligned}$$

This initial state is expected to be computationally more complex than the previous one, as the minimum altitude constraint is likely to become active as well. In addition, the initial state lies almost on a dispersal surface of the evader.

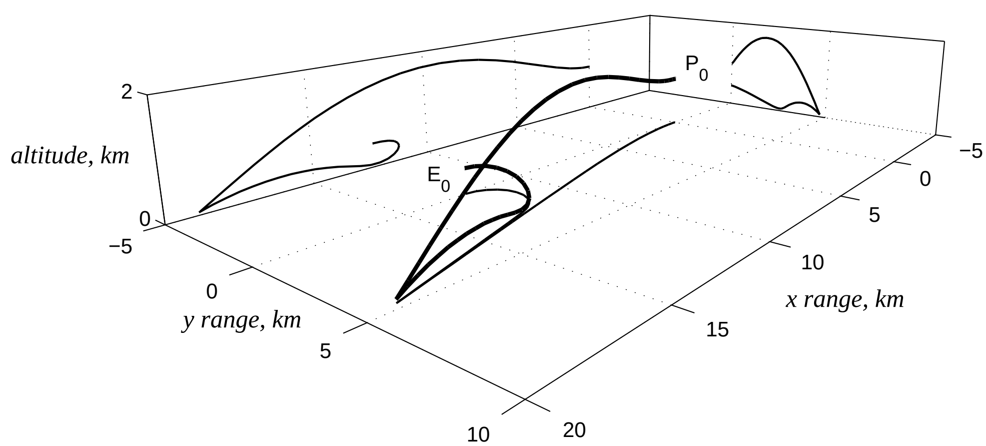

Figure 6: Optimal saddle point trajectories of the pursuer (P) and evader (E) in the second example.

The solution is presented in Figure 6. After an initial guess similar to the previous case, six iterations were again needed to obtain a relative accuracy below 10^{-3}. The value of the game corresponding to this initial setting is 31.2 seconds.

The optimal trajectory of the evader again consists of a sharp turn away from the pursuer, but now the optimal throttle setting is 0 at first, see Figure 8 (c). The turn is followed by a dive, which activates the minimum altitude constraint, see figure 7. Both a touch point and a short constrained arc are possible, since the minimum altitude constraint is of second order. The dynamic pressure constraint (Figures 7 and 8 (d)) then forces the evader to climb momentarily. The evader first ascends and then descends along the constraint. The capture occurs at $h = 50$ m, i.e., both state constraints are active at the moment of termination. For the pursuer, it is again optimal to ascend relatively much. The maximum velocity of the missile is smaller than in the first example due to larger drag force at lower altitudes. The loadfactor and bank angle histories of the players are shown in Figure 8 (a) and 8 (b).

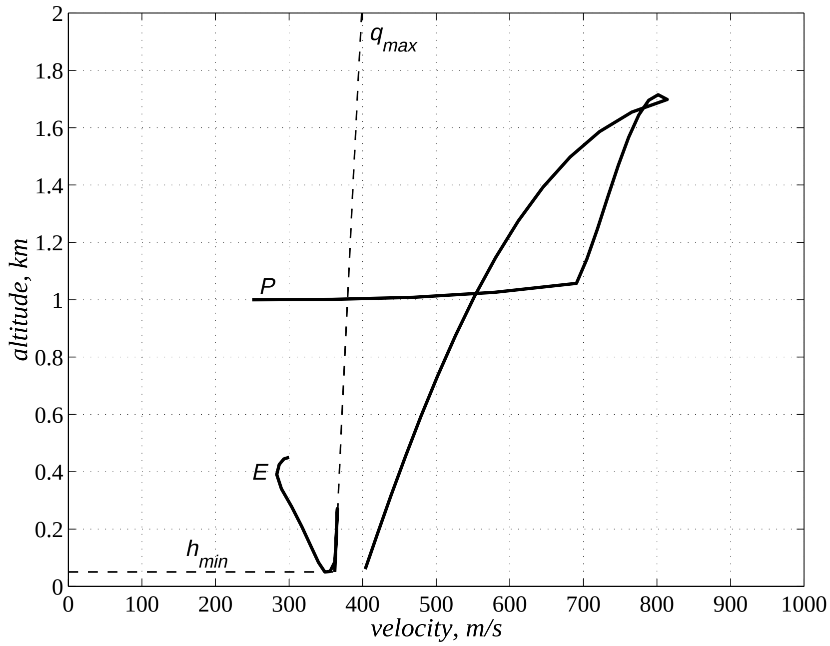

Figure 7: The altitude and velocity histories of the pursuer (P_0) and the evader(E_0).

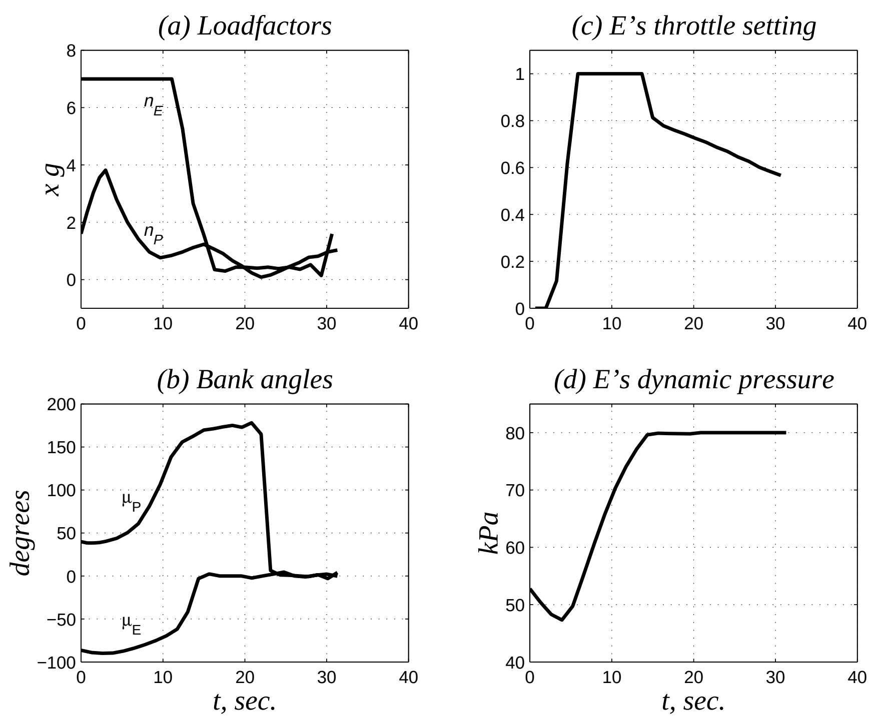

Figure 8: On the left, (a) the loadfactor and (b) the bank angle histories of the players. On the right, (c) the throttle setting and (d) the dynamic pressure of the evader.

The switching structure of the latter solution includes 4-5 switching points at unknown time instants in addition to the thrust discontinuities. The corresponding necessary conditions would form a high-dimensional multipoint boundary value problem with 6-7 interior point conditions and a number of unknown jump parameters at some of these points. Solving the necessary conditions would require that this switching structure is guessed correctly in advance. In the presented solution method, however, the treatment of the switching structure is straightforward due to the discretization approach. If a more detailed solution is needed, the solutions obtained here, together with the corresponding Lagrange multipliers, can be used as an initial estimate of the state and adjoint trajectories in an indirect method. The solutions also provide an estimate of the switching structure of the solution.

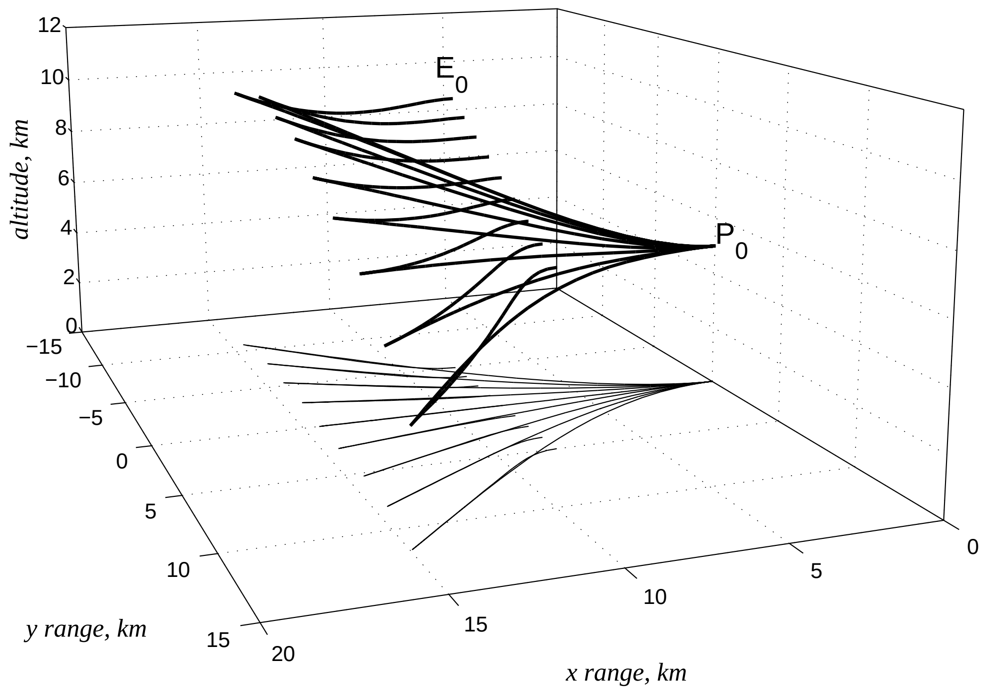

Figure 9: A collection of saddle point trajectories corresponding to different initial states. The pursuer's initial state is denoted by P_0, and one of the evader's initial states by E_0.

Finally, Figure 9 presents nine saddle point trajectory pairs corresponding to the initial conditions

$$\begin{aligned} x_{P,0} &= 0 \text{ [m]} & x_{E,0} &= 8,000 \text{ [m]} \\ y_{P,0} &= 0 \text{ [m]} & & \\ h_{P,0} &= 5,000 \text{ [m]} & & \\ v_{P,0} &= 250 \text{ [m/s]} & v_{E,0} &= 300 \text{ [m/s]} \\ \gamma_{P,0} &= 0 \text{ [deg]} & \gamma_{E,0} &= 0 \text{ [deg]} \\ \chi_{P,0} &= 0 \text{ [deg]} & \chi_{E,0} &= 0 \text{ [deg]}, \end{aligned}$$

$$\begin{aligned} (y_{E,0}, h_{E,0}) \in \; & \{(5000, 6000), (3750, 6500), (2500, 7000), (1250, 7500), (0, 8000), \\ & (-1250, 8500), (-2500, 9000), (-3750, 9500), (-5000, 10000)\} \text{ [m]}. \end{aligned}$$

Every trajectory pair converges to a common vertical plane, but only the solution with the evader's lowest initial altitude activates the dynamic pressure constraint and terminates with a singular subarc.

A well known missile avoidance maneuver consists of a sharp turn away from the missile and straightforward evasion, known as 'missile outrunning' and a final evasive maneuver known as the 'end game', see [29] and references cited therein. All the solutions presented here are in agreement with the first stages of this strategy. The end game stage, however, would require a separate analysis.

8 Summary and Conclusions

In this paper we have explored possibilities to extend the discretization methods of optimal control into pursuit-evasion games. We have analyzed a decomposition method presented earlier in a discretized framework and concluded that applying discretization and nonlinear programming to the subproblems of the decomposition method is equivalent, up to some minor details, to discretizing the saddle point problem at the outset and solving the resulting bilevel programming problem by the feasible direction method outlined above. This means that the original decomposition method can be interpreted as a special kind of bilevel programming algorithm with established convergence conditions.

The performance of the use of discretization and nonlinear programming is illustrated by solving a pursuit-evasion encounter between a missile and an aircraft from some initial states with the final time as the payoff. The flight vehicles are carefully modeled to match the reality. The saddle point is reached robustly with a relatively small number of iterations. The dynamic pressure constraint of the example induces a singular control arc that is treated conveniently by direct transcription. The solution of the first example is compared with a solution obtained by an indirect method. The solutions are close to each other even when a simple equidistant discretization grid is used. The accuracy could further be increased by more sophisticated discretization approaches.

Saddle point problems are, by definition, more difficult than optimization problems. Instead of a minimization or a maximization, both operations have to be performed simultaneously, and as we have seen, to obtain a saddle point solution, a sequence of optimization problems has to be solved. Nevertheless, all the benefits of discretization techniques can be utilized. A rough initial guess leads to convergence even in complex problems like the one considered in this paper, and the necessary conditions are not explicitly involved in the solution process. Hence, the method might offer a possibility to extend the ideas of automated optimal control problem solution presented in [19] to pursuit-evasion game framework, too. This would be especially useful for producing massive amounts of open-loop representations of saddle point trajectories for feedback control synthesis, see Ref. [30].

References

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

[2] Shinar, J., Glizer, V., *Application of the Receding Horizon Strategy to Singularly Perturbed Pursuit-Evasion Problems,* Proceedings of the American Control Conference, pp. 3247-3248, American Automatic Control Council, 1992.

[3] Basar, T., and Olsder, G. J., *Dynamic Noncooperative Game Theory, 2nd Edition,* Academic Press, San Diego, 1995.

[4] Breitner, M., Pesch, H., and Grimm, W., *Complex Differential Games of Pursuit-Evasion Type with State Constraints, Part 1: Necessary Conditions for Optimal Open-Loop Strategies,* Journal of Optimization Theory and Applications, vol. 78, no. 3, pp. 419-441, 1993.

[5] Breitner, M., Pesch, H. J., and Grimm, W., *Complex Differential Games of Pursuit-Evasion Type with State Constraints, Part 2: Multiple shooting and homotopy,* Journal of Optimization Theory and Applications, vol. 78, no. 3, pp. 442-464, 1993.

[6] Lachner, R., Breitner, M. and Pesch, H.J., *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, pp. 564-572, 1996.

[7] Dolezal, J., *A Gradient-Type Algorithm for the Numerical Solution of Two-Player Zero-Sum Differential Game Problems,* Kybernetika, vol. 14, no. 6, pp. 429-446, 1978.

[8] Järmark, B., *Convergence Control in Differential Dynamic Programming Applied to Air-to-Air Combat,* AIAA Journal vol. 14, no. 1, pp. 118-121, 1976.

[9] Raivio, T., and Ehtamo, H., *On Numerical Solution of a Class of Pursuit-Evasion Games,* Annals of the International Society of Dynamic Games, Birkhäuser, to appear. An earlier version: Proceedings of the 7th International Symposium of Dynamic Games and Applications, Kanagawa, Japan, pp. 784-795, 1996.

[10] Anandalingam, G., Editor, *Hierarchical optimization,* Annals of Operations Research, vol. 34, 1992.

[11] Falk, J. and Liu, J., *On bilevel programming, Part I: General nonlinear case,* Mathematical Programming, vol. 70, pp. 47-72, 1995.

[12] Shimitsu, K., Ishizuka, Y. and Bard, J., *Nondifferentiable and Two-Level Mathematical Programming*, Kluwer Academic Publishers, Boston, 1997.

[13] 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, pp. 338-342, 1987.

[14] Raivio, T., Ehtamo, H., and Hämäläinen, R.P., *Aicraft Trajectory Optimization Using Nonlinear Programming,* System Modeling and Optimization, Edited by J. Dolezal and J. Fidler, Chapman & Hall, London, pp. 435-441, 1996.

[15] SCHWARTZ, A., AND POLAK, E., *Consistent Approximations for Optimal Control Problems Based on Runge-Kutta Integration,* SIAM Journal of Control and Optimization, vol. 34, no. 4, pp. 1135-1269, 1996.

[16] SEYWALD, H., *Trajectory Optimization Based on Differential Inclusion,* Journal of Guidance, Control and Dynamics, vol. 17, no. 3, pp. 480-487, 1994.

[17] BETTS, J., *Survey of Numerical Methods for Trajectory Optimization,* Journal of Guidance, Control and Dynamics, vol. 21 no. 2, pp. 193-207, 1998.

[18] HULL, D. *Conversion of Optimal Control Problems into Parameter Optimization Problems,* Journal of Guidance, Control and Dynamics, vol. 20, no. 1, pp. 57-60, 1997.

[19] VIRTANEN, K., EHTAMO, H., RAIVIO, T., AND HÄMÄLÄINEN, R.P., *VIATO – Visual Interactive Aircraft Trajectory Optimization,* IEEE Transactions on Systems, Man and Cybernetics, part C, Vol. 29 No. 3, pp. 409-421, 1999.

[20] LACHNER, R., BREITNER, M., AND PESCH, H.J., *Three-Dimensional Air Combat: Numerical Solution of Complex Differential Games,* New Trends in Dynamic Games and Applications, Edited by G.J. Olsder, Birkhäuser, Boston, pp. 165-190, 1996.

[21] Bardi,M. and Capuzzo-Dolcetta, I., *Optimal Control and Viscosity Solutions of Hamilton-JAcobi-Bellman Equations*, Birkhuser, Boston, 1997.

[22] FIACCO, A., *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, London, 1983.

[23] DE SILVA, A., AND MCCORMICK, G., *Implicitly Defined Optimization Problems,* Annals of Operations Research, Vol. 34, pp. 107-124, 1992.

[24] RUSTEM, B. AND HOWE, M., *Numerical Experiments with Continuous Minimax Algorithms,* Proceedings of the 8th International Symposium on Dynamic Games and Applications, Maastricht, The Netherlands, pp. 546-553, 1998.

[25] RAIVIO, T., AND EHTAMO, H., *Visual Aircraft identification as a Pursuit-Evasion Game*, Unpublished manuscript.

[26] BAZARAA, M., SHERALI, H. AND SHETTY, C., *Nonlinear Programming, Theory and Algorithms, 2nd Edition,* Wiley, 1993.

[27] MIELE, A., *Flight Mechanics I, Theory of Flight Paths,* Addison-Wesley, Reading, 1962.

[28] GILL, P., MURRAY, W., SAUNDERS, M., AND WRIGHT, M., *User's Guide for NPSOL 4.0: A Fortran Package for Nonlinear Programming,* Report SOL 86-4, Stanford University, 1986.

[29] SHINAR, J., AND TABAK, R., *New Results in Optimal Missile Avoidance Analysis,* Journal of Guidance, Control and Dynamics, vol.17, no.5, pp. 897-902, 1994.

[30] BREITNER, M.,*Robust Optimal On-Board Reentry Guidance of a European Space Shuttle: Dynamic Game Approach and Guidance Synthesis with Neural Networks,* Complex dynamical processes with incomplete information, Edited by E. Reithmeier, Birkhäuser, International Series of Numerical Mathematics, Basel, to appear.

Systems Analysis Laboratory
Research Reports, Series A

A70 February 1998	On Arbitrage, Optimal Portfolio and Equilibrium under Frictions and Incomplete Markets Jussi Keppo
A71 February 1998	Optimal Home Currency and the Curved International Equilibrium Jussi Keppo
A72 February 1998	Market Conditions under Frictions and without Dynamic Spanning Jussi Keppo
A73 March 1998	On Attribute Weighting in Value Trees Mari Pöyhönen
A74 March 1998	There is Hope in Attribute Weighting Mari Pöyhönen and Raimo P. Hämäläinen
A75 September 1998	Implementation of the Norwegian Spring-Spawning Herring Stock Dynamics and Risk Analysis Suzanne Touzeau, Veijo Kaitala and Marko Lindroos
A76 November 1998	Management of Regional Fisheries Organisations: An Application of the Shapley Value Marko Lindroos
A77 January 1999	Bayesian Ageing Models and Indicators for Repairable Components Urho Pulkkinen and Kaisa Simola
A78 May 1999	Determinants of Cost Efficiency of Finnish Hospitals: A Comparison of DEA and SFA Miika Linna and Unto Häkkinen
A79 May 1999	The impact of health care financing reform on the productivity change in Finnish hospitals Miika Linna

Systems Analysis Laboratory
Research Reports, Series B

B19 June 1996	Ympäristön arvottaminen - Taloustieteelliset ja monitavoitteiset menetelmät Toim. Pauli Miettinen ja Raimo P. Hämäläinen
B20 September 1996	Sovellettu todennäköisyyslasku Pertti Laininen (uudistettu painos 1997)
B21 June 1997	Monitavoitteisen päätöksenteon mallit vesistön säännöstelyssä Toim. Raimo P. Hämäläinen

Systems Analysis Laboratory
Research Reports, Series E
Electronic Reports: www.e-reports.sal.hut.fi

E1 October 1999	Exploitation and Conservation of Norwegian Spring-Spawning Herring: Open Access Versus Optimal Harvesting Strategies Virpi Junttila, Marko Lindroos and Veijo Kaitala
E2 October 1999	Optimal harvesting of the Norwegian spring-spawning herring stock and a spatial model Mitri Kitti, Marko Lindroos and Veijo Kaitala
E3 March 2000	A continuation method for minimum-time problems Harri Ehtamo, Tuomas Raivio and Raimo P. Hämäläinen

Series A and E reports are downloadable at www.sal.hut.fi/Publications/

Orders for paper copies: Helsinki University of Technology
Systems Analysis Laboratory
P.O. Box 1100, FIN-02015 HUT, Finland
systems.analysis@hut.fi