**ESAIM: PROCEEDINGS AND SURVEYS, August 2021, Vol. 71, p. 33-42**
Didier Auroux, Jean-Baptiste Caillau, Régis Duvigneau, Abderrahmane Habbal, Olivier Pantz, Luc Pronzato, Ludovic Rifford

# ON THE NUMERICAL SOLUTION OF A FREE END-TIME HOMICIDAL CHAUFFEUR GAME*

Francesca Calà Campana[1], Alberto De Marchi[2], Alfio Borzì[1] and Matthias Gerdts[2]

**Abstract.** A functional formulation of the classical homicidal chauffeur Nash game is presented and a numerical framework for its solution is discussed. This methodology combines a Hamiltonian based scheme with proximal penalty to determine the time horizon where the game takes place with a Lagrangian optimal control approach and relaxation to solve the Nash game at a fixed end-time.

## Introduction

The Homicidal Chauffeur (HC) game is a classic problem in the field of differential (dynamical) games that was introduced by Rufus Philip Isaacs in [11] and further elaborated in his seminal book [12]. The statement of the problem is that of a car with a limited radius of turn and constant velocity that pursues a pedestrian, whose velocity is bounded by a given value, that tries to prevent collision. We refer to the car as the pursuer and to the pedestrian as the evader; both are players of the HC game.

This is a continuous pursuit-evasion game that can be considered the archetypal of problems of this class and has motivated much research work with early fundamental contribution as in [17]. We refer to [18] for a review of results and a survey of the literature on this topic. We remark that these works focus on a geometrical setting of the HC game and construct solutions based on optimal trajectories and singular lines that disperse, join or refract. In particular, Isaacs investigated the homicidal chauffeur game using a particular method for solving partial differential games based on backward computation of characteristics.

On the other hand, in forthcoming works, different functional settings of pursuit-evasion differential games were considered where the actions of the players is modelled by time-dependent functions, and the purpose of each player and the cost of its action is formulated in terms a cost functional; the player's objective. In this framework, zero-sum versions of the HC problem were proposed in [2,10]; see [3] for a review. Furthermore, also based on the solution concept of Nash equilibrium (NE), nonzero-sum pursuit-evasion games were introduced in [19]; see also [16] for a recent contribution and further references in this field. However, we notice that in many of the latter references the pursuit-evasion game is considered in a fixed end-time horizon and only few works address the case of a free end time; see, e.g., [6]. Furthermore, we remark that, in many research works

* *The authors are grateful to the Bayerisch-Französisches Hochschulzentrum (BFHZ) for partial support (Project Nr. FK32_2017).*

[1] Institut für Mathematik, Universität Würzburg, Emil-Fischer-Strasse 30, 97074 Würzburg, Germany;
e-mail: francesca.cala-campana@mathematik.uni-wuerzburg.de
alfio.borzi@mathematik.uni-wuerzburg.de

[2] Institute of Applied Mathematics and Scientific Computing, Department of Aerospace Engineering, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany;
e-mail: alberto.demarchi@unibw.de, matthias.gerdts@unibw.de

Article published online by EDP Sciences and available at https://www.esaim-proc.org or https://doi.org/10.1051/proc/202171104

on the HC game, the focus is on theoretical results as the existence of NE, whereas less effort has been put in the construction of algorithms that accommodate the functional framework for pursuit-evasion games.

It is the purpose of this work to develop a numerical scheme for solving a non-zero sum HC game in a time-optimal formulation. Our scheme has a bilevel structure where the outer procedure determines the optimal time horizon in a Hamiltonian framework including proximal penalty, and the inner procedure solves a HC game in a fixed time interval by using an optimal control strategy. Then a relaxation step is performed to get a common end-time for the players and a new control couple. At convergence these controls together with the end-time will give the equilibrium solution of the HC game.

In the HC game formulation, the state of our pursuit-evasion system consists of the planar space coordinates of the position of the players, and the controls/strategies are the velocity of the evader and the radial velocity of turn of the pursuer. As illustrated in [18], we discuss this HC system in an inertial reference frame and in a frame moving with the pursuer. Further, by choosing appropriate values for the weights of the cost functionals, we focus on the case where collision may occur and, correspondingly, determine the time for this event.

In the next section, we illustrate the dynamical system modelling the motion of the evader and pursuer and including the control mechanisms (an Appendix is included to facilitate the modelling procedure). Further, we introduce the functional objectives of these two players and define the corresponding NE problem. Also in this section, we draw a connection with optimal control problems and discuss the (partial) characterization of the NE solution in terms of optimality systems.

In Section 2, we present our numerical framework that combines an optimal control strategy and relaxation with a method for determining the time horizon of the game. This latter scheme represents an extension of previous work in [7–9].

Section 3 is devoted to the numerical validation of our NE game formulation and of our solution procedure. With our method, we are able to find different solutions of the HC problem starting from different initialisations. On the other hand, small changes of the optimization weights result in similar solutions.

A section of conclusion completes our work.

## 1. A homicidal-chauffeur game

We consider a planar system of a pursuer and an evader whose positions, with respect to an inertial Cartesian reference frame, are subject to the following dynamics [18]

$$\begin{aligned} \dot{x}_p &= \sin\theta, & \dot{x}_e &= v_1, \\ \dot{y}_p &= \cos\theta, & \dot{y}_e &= v_2, \\ \dot{\theta} &= u, |u| \leq \nu_u, & |v_i| &\leq \bar{\nu}_v,\ i = 1, 2, \end{aligned} \tag{1}$$

with given initial conditions. In this system, $(x_p, y_p)$ represents the position of the pursuer $(P)$ that has a unit-vector velocity while the orientation of this vector with respect to the $y$-axis (clockwise) is given by $\theta$. On the other hand, the position of the evader $(E)$ is given by $(x_e, y_e)$ and its velocity is denoted with $(v_1, v_2)$.

In the original work of Isaacs in [12], it appears convenient to reformulate this model in a reference frame moving with the pursuer. In this setting, the origin of the reference system corresponds to the position of $P$, and the direction of the (new) $y$-axis coincides with the velocity vector of the pursuer. By coordinate transformation (see the Appendix), we obtain the following dynamical system for the position of the evader in the reference frame of the pursuer

$$\begin{aligned} \dot{x} &= -u\, y + v_x, & x(0) &= x_0, \\ \dot{y} &= u\, x - 1 + v_y, & y(0) &= y_0, \end{aligned} \tag{2}$$

where $(x_0, y_0)$ represents the initial position of the evader in the moving reference frame at time $t = 0$.

Notice that $u = u(t)$ denotes the scalar control (strategy) function of $P$, and $\mathbf{v} = \mathbf{v}(t)$ represents the vector control (strategy) function of the evader $E$. As in (1), these functions are required to lay in given admissible sets: $u \in U_{ad}$, and $\mathbf{v} \in V_{ad}$. Specifically, we have $U_{ad} = \{u \in L^2(0,T) : |u(t)| \leq \nu_u \text{ a.e. in } (0,T)\}$, $V_{ad} = \{\mathbf{v} \in L^2(0,T;\mathbb{R}^2) : |v_s(t)| \leq \nu_v \text{ a.e. in } (0,T),\ s \in \{x,y\}\}$. The numbers $\nu_u$ and $\nu_v$ are given control constraints, and $T > 0$ is the final time that will be determined by our algorithm. Notice that the components of $\mathbf{v}$ refer to the new reference frame and are different than $(v_1, v_2)$ in (1) and the value of $\nu_v$ possibly differs from $\bar{\nu}_v$ in (1) because the absolute value is considered, and not the 2-norm.

The HC model (2) can be put in a compact form as follows. Denote with $\mathbf{z} := (x,y)'$, $\mathbf{z}_0 := (x_0, y_0)'$, $B := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ and $\mathbf{b} := (0,-1)'$ (the prime $'$ means transpose) then we can write (2) in the following form

$$\dot{\mathbf{z}} = u\,B\,\mathbf{z} + \mathbf{v} + \mathbf{b}, \qquad \mathbf{z}(0) = z_0. \tag{3}$$

The objective of player $P$ is to come as close as possible to player $E$, while minimizing the $L^2$ cost of its action. The player $E$ wants to prevent its capture, while also minimizing the cost of its action. These objectives are modelled by the following functionals

$$J_p(u,\mathbf{v},\mathbf{z},T) := \frac{\nu_p}{2}\|u\|_{L^2}^2 + \frac{r_p}{2}\|\mathbf{z}(T)\|_2^2 + \omega_p T, \qquad J_e(u,\mathbf{v},\mathbf{z},T) := \frac{\nu_e}{2}\|\mathbf{v}\|_{L^2}^2 - \frac{r_e}{2}\|\mathbf{z}(T)\|_2^2 - \omega_e T, \tag{4}$$

where $\nu_p, \nu_e > 0$, $r_p, r_e$ represent the relative strength of the interaction with $r_p, r_e \geq 0$ and $\omega_p > 0, \omega_e \geq 0$.

One can prove that, for given $u^* \in U_{ad}$ and $\mathbf{v}^* \in V_{ad}$, the control-to-state maps $u \mapsto \mathbf{z}(u, \mathbf{v}^*)$ and $\mathbf{v} \mapsto \mathbf{z}(u^*, \mathbf{v})$ are well defined; where $\mathbf{z}(u,\mathbf{v})$ is the unique solution to (3), with the given initial condition $\mathbf{z}(0) = z_0$, and $u \in U_{ad}$, and $\mathbf{v} \in V_{ad}$. With these maps, we can introduce the reduced cost functionals

$$\widetilde{J}_p(u,\mathbf{v},T) := J_p(u,\mathbf{v},\mathbf{z}(u,\mathbf{v}),T), \qquad \widetilde{J}_e(u,\mathbf{v},T) := J_e(u,\mathbf{v},\mathbf{z}(u,\mathbf{v}),T). \tag{5}$$

Notice that both functionals depend on both strategies, therefore the solution to the differential game is sought as a Nash equilibrium that is defined as follows

**Definition 1.** Let $\nu_u,\, \nu_v \in (0,\infty]$ be fixed. The functions $u^* \in U_{ad}$ and $\mathbf{v}^* \in V_{ad}$ are said to form a Nash equilibrium strategy for the game $G = (J_e, J_p; \nu_u,\, \nu_v)$ for a $T = T^*$ if it holds

$$\widetilde{J}_p(u^*,\mathbf{v}^*,T^*) \leq \widetilde{J}_p(u,\mathbf{v}^*,T^*) \tag{6}$$

$$\widetilde{J}_e(u^*,\mathbf{v}^*,T^*) \leq \widetilde{J}_e(u^*,\mathbf{v},T^*) \tag{7}$$

for all $u \in U_{ad}$ and $\mathbf{v} \in V_{ad}$.

We see that (6) can be written as $u^* = \arg\min_u \widetilde{J}_p(u,\mathbf{v}^*,T^*)$, and (7) can be written as $\mathbf{v}^* = \arg\min_{\mathbf{v}} \widetilde{J}_e(u^*,\mathbf{v},T^*)$. This definition allows to interpret our Nash game as two coupled optimal control problems. If $(u^*,\mathbf{v}^*,T^*)$ is the Nash equilibrium, then $u^*$ is optimal for player $P$, in the sense that it solves the following optimal control problem

$$\begin{aligned} \min \quad & J_p(u,\mathbf{v}^*,\mathbf{z},T^*) \\ \text{s.t.} \quad & \dot{\mathbf{z}} = u\,B\,\mathbf{z} + \mathbf{v}^* + \mathbf{b}, \quad \mathbf{z}(0) = z_0. \end{aligned} \tag{8}$$

On the other hand, the function $\mathbf{v}^*$ is optimal for player $E$, that is, it is a solution of the following optimal control problem

$$\begin{aligned} \min \quad & J_e(u^*,\mathbf{v},\mathbf{z},T^*) \\ \text{s.t.} \quad & \dot{\mathbf{z}} = u^*\,B\,\mathbf{z} + \mathbf{v} + \mathbf{b}, \quad \mathbf{z}(0) = z_0. \end{aligned} \tag{9}$$

Therefore $(u^*, \mathbf{v}^*)$ solves simultaneously two optimal control problems whose solutions are characterized by the following optimality system

$$\begin{aligned} \dot{\mathbf{z}} &= u\,B\,\mathbf{z} + \mathbf{v} + \mathbf{b}, & \mathbf{z}(0) &= z_0, \\ -\dot{\mathbf{p}}_p &= u\,B\,\mathbf{p}_p, & \mathbf{p}_p(T^*) &= -r_p\,\mathbf{z}(T^*), \\ -\dot{\mathbf{p}}_e &= u\,B\,\mathbf{p}_e, & \mathbf{p}_e(T^*) &= r_e\,\mathbf{z}(T^*), \\ u &= \mathcal{P}_{U_{ad}}\left(\nu_p^{-1}\langle B\mathbf{z}, \mathbf{p}_p\rangle\right), & \mathbf{v} &= \mathcal{P}_{V_{ad}}\left(\nu_e^{-1}\mathbf{p}_e\right), \end{aligned} \tag{10}$$

where $\mathcal{P}_{U_{ad}}$ and $\mathcal{P}_{V_{ad}}$ represent the $L^2$ projections onto $U_{ad}$ and $V_{ad}$, respectively, and $\langle\cdot,\cdot\rangle$ denotes the Euclidean scalar product in $\mathbb{R}^2$. We refer to the functions $\mathbf{p}_p$ and $\mathbf{p}_e$ as the adjoint variables (Lagrange multipliers), and the corresponding differential equations are called adjoint equations.

## 2. A Numerical Method to Solve the HC Nash Game

In this section, we illustrate all aspects of our numerical method to compute a NE strategy for the HC game with free end-time. For clarity of presentation, we first discuss some subproblems with fixed end-time.

### 2.1. Fixed End-Time Optimal Control Problems

At the core of our iterative numerical scheme, where a tentative end-time $T$ is available and iterates $u^k$ and $\mathbf{v}^k$ have been computed, we consider the solutions to the two optimal control problems (8) and (9). Specifically, given $\mathbf{v}^k$, we aim at computing $\bar{u}$ by solving the optimality system given by

$$\begin{aligned} \dot{\mathbf{z}} &= u\,B\,\mathbf{z} + \mathbf{v}^k + \mathbf{b}, & \mathbf{z}(0) &= z_0, \\ -\dot{\mathbf{p}}_p &= u\,B\,\mathbf{p}_p, & \mathbf{p}_p(T) &= -r_p\mathbf{z}(T), \\ u &= \mathcal{P}_{U_{ad}}\left(\nu_p^{-1}\langle B\mathbf{z}, \mathbf{p}_p\rangle\right). \end{aligned} \tag{11}$$

In a similar way, given $u^k$, we can compute $\bar{\mathbf{v}}$ by solving the following optimality system

$$\begin{aligned} \dot{\mathbf{z}} &= u^k\,B\,\mathbf{z} + \mathbf{v} + \mathbf{b}, & \mathbf{z}(0) &= z_0, \\ -\dot{\mathbf{p}}_e &= u^k\,B\,\mathbf{p}_e, & \mathbf{p}_e(T) &= r_e\mathbf{z}(T), \\ \mathbf{v} &= \mathcal{P}_{V_{ad}}\left(\nu_e^{-1}\mathbf{p}_e\right). \end{aligned} \tag{12}$$

In practice, to compute $\bar{u}$ and $\bar{\mathbf{v}}$, we use these optimality systems to determine reduced gradients, $\nabla_u \widetilde{J}_p$ and $\nabla_{\mathbf{v}} \widetilde{J}_e$, that are required to implement the well-known nonlinear conjugate gradient (NCG) method. We employ the Polak-Ribiére variant of NCG to determine $\bar{u}$ and $\bar{\mathbf{v}}$; see [5] for more details.

This approach requires the numerical solution of the HC model and its adjoint. For this purpose, we use the so-called modified Crank-Nicolson (MCN) scheme [4], which requires a time grid that results by subdividing the interval $[0, T]$ in uniform intervals of size $h$ and $N_t$ points, such that $t_j = (j-1)h$ and $0 = t_1 < \cdots < t_{N_t} = T$. In this setting, the MCN approximation to our HC model is given by

$$\frac{\mathbf{z}_{j+1} - \mathbf{z}_j}{h} = \frac{u_{j+1} + u_j}{2}\,B\,\frac{\mathbf{z}_{j+1} + \mathbf{z}_j}{2} + \frac{\mathbf{v}_{j+1} + \mathbf{v}_j}{2} + \mathbf{b}, \tag{13}$$

where $j = 1, \ldots, N_t - 1$, $h = T/(N_t - 1)$, and $\mathbf{z}_j$, $\mathbf{v}_j$, etc., represent the numerical approximation to $\mathbf{z}(t_j)$, $\mathbf{v}(t_j)$, etc.. The initial point $z_1 = z(0)$ is given. A similar scheme is used to solve the adjoint equations. As in [4] one can prove that this scheme is stable and second-order accurate.

For ease of presentation, we summarize the numerical procedure discussed above with the following algorithm.

**Algorithm 1** Solution of fixed end-time optimal control problems

```
input $u^k$, $\mathbf{v}^k$, $T$;
Compute $\bar{u} = \arg\min_u \widetilde{J}_p(u, \mathbf{v}^k, T)$
Compute $\bar{\mathbf{v}} = \arg\min_{\mathbf{v}} \widetilde{J}_e(u^k, \mathbf{v}, T)$
```

## 2.2. Fixed End-Time Nash Games

A classical iterative method for solving NE problems is the relaxation scheme discussed in [13] and implemented in the following algorithm, where $\tau$ is a relaxation factor that we specify in our numerical experiments. In general, there is no a priori choice of $\tau$ available, and for a discussion on the convergence of Algorithms 2 and 4 some strong convexity criteria are required; see [1,14]. However, in our numerical experiments we always attain convergence of this scheme by a moderate choice of the relaxation factor. Notice that the steps 4 and 5 of Algorithm 2 compute $\bar{u}$ and $\bar{\mathbf{v}}$ in parallel as illustrated with Algorithm 1.

**Algorithm 2** Relaxation scheme with fixed end-time control problems

```
1: input $u^0$, $\mathbf{v}^0$, $T$; parameters $\tau \in (0,1)$, $\epsilon > 0$
2: Set $k = 0$
3: repeat
4:     Compute $\bar{u} = \arg\min_u \widetilde{J}_p(u, \mathbf{v}^k, T)$
5:     Compute $\bar{\mathbf{v}} = \arg\min_{\mathbf{v}} \widetilde{J}_e(u^k, \mathbf{v}, T)$
6:     Set $u^{k+1} := \tau\, u^k + (1-\tau)\, \bar{u}$
7:     Set $\mathbf{v}^{k+1} := \tau\, \mathbf{v}^k + (1-\tau)\, \bar{\mathbf{v}}$
8:     Set $k := k + 1$
9: until $\max\left(\|u^{k+1} - u^k\|, \|\mathbf{v}^{k+1} - \mathbf{v}^k\|\right) < \epsilon$
```

## 2.3. Free End-Time Optimal Control Problems

The free end-time version of the optimal control problems (8) and (9) are solved through a bilevel approach, which aims at decoupling the determination of the end-time from the solution of the optimal control problems. This allows avoiding time transformation techniques, which usually requires solving a problem very sensitive to the initial guess, see [8].

On the other hand, our iterative solution procedure leads to a two-nested-loops algorithm. The outer loop is devoted to computing the end-time, while the inner loop solves the optimal control problems for a given, tentative time horizon. The key result supporting this approach, in particular how the two levels interact, is drawn in [7, Thm. 10] for linear-quadratic problems and extended in [8] to nonlinear problems.

In order to illustrate our outer loop, we assume that at the $k$-th iterate the estimates $u^k$, $\mathbf{v}^k$ and $T^k$ are available. Further, let us denote with $u[T] \in U_{ad}$ and $\mathbf{v}[T] \in V_{ad}$ the optimal controls associated with final time $T$ as follows

$$u[T] = \arg\min_u \widetilde{J}_p(u, \mathbf{v}^k, T), \qquad \mathbf{v}[T] = \arg\min_{\mathbf{v}} \widetilde{J}_e(u^k, \mathbf{v}, T). \tag{14}$$

Clearly, this construction requires to extend or restrict the functions $u^k$ and $\mathbf{v}^k$ defined on $[0, T^k]$ to the interval $[0, T]$. With this preparation, we can consider the following optimization problems, which formally involve only one decision variable: the final time.

$$T_u = \arg\min_T \widetilde{J}_p(u[T], \mathbf{v}^k, T) + \frac{\mu_k}{2}\left(T - T^k\right)^2, \qquad T_v = \arg\min_T \widetilde{J}_e(u^k, \mathbf{v}[T], T) + \frac{\mu_k}{2}\left(T - T^k\right)^2. \tag{15}$$

For the purpose of defining a robust iteration procedure, a (quadratic) proximal regularization term is introduced, with parameters $\mu_k > 0$. Corresponding to (8) and (9), we have the Pontryagin-Hamiltonian (HP) functions given by

$$H_p(\mathbf{z}, u, \mathbf{v}^k, \mathbf{p}_p) = \mathbf{p}_p \left(uB\mathbf{z} + \mathbf{v}^k + \mathbf{b}\right) - \frac{\nu_p}{2}u^2 - \omega_p, \quad H_e(\mathbf{z}, u^k, \mathbf{v}, \mathbf{p}_e) = \mathbf{p}_e \left(u^k B\mathbf{z} + \mathbf{v} + \mathbf{b}\right) - \frac{\nu_e}{2}\mathbf{v}^2 + \omega_e. \tag{16}$$

Then, considering (15) and (16), we have the following augmented HP functions

$$\widetilde{H}_p(u[T], \mathbf{v}^k, T) = H_p(\mathbf{z}, u, \mathbf{v}^k, \mathbf{p}_p) + \mu_k \left(T - T^k\right), \quad \widetilde{H}_e(u^k, \mathbf{v}[T], T) = H_e(\mathbf{z}, u^k, \mathbf{v}, \mathbf{p}_e) + \mu_k \left(T - T^k\right), \tag{17}$$

where the term $\mu_k(T - T^k)$ is associated to the quadratic proximal penalty in (15). In order to implement a gradient-based scheme for determining $T$, we consider the sensitivity of the cost functional with respect to this variable. This can be evaluated using the HP function of the underlying problem along a fixed-time solution [7, Thm. 10], namely

$$\frac{\mathrm{d}\widetilde{J}_p}{\mathrm{d}T}(u[T], \mathbf{v}^k, T) = H_p(u[T], \mathbf{v}^k, T), \qquad \frac{\mathrm{d}\widetilde{J}_e}{\mathrm{d}T}(u^k, \mathbf{v}[T], T) = H_e(u^k, \mathbf{v}[T], T). \tag{18}$$

By exploiting the fact that the (non-augmented) system is autonomous and thus the Hamiltonian is constant along a solution, these sensitivities can be evaluated at any $t \in [0, T]$ at the outer loop of our solution procedure. For more details see [8, Algorithm 3].

Summarizing, the method for solving the following free end-time optimal control problem

$$(u, T_u) = \arg\min_{u,T} \widetilde{J}_p(u, \bar{\mathbf{v}}, T),$$

is given in Algorithm 3 below. A similar procedure applies for the optimal control problem corresponding to the evader. If required in Step 8 of Algorithm 3, an extrapolation step is performed to extend the control function to the given time interval.

**Algorithm 3** Free end-time optimal control problem and proximal penalty

```
1:  input $u^0$,$\bar{\mathbf{v}}$, $T^0$; parameters $\epsilon > 0$, $\{\mu_k\} \to \infty$
2:  Set $k = 0$, $T^{-1} = T^0$
3:  repeat
4:      Compute $\bar{u} = \arg\min_u \widetilde{J}_p(u, \bar{\mathbf{v}}, T^k)$
5:      Compute $\mathbf{z}^k, \mathbf{p}_p^k$
6:      Evaluate $\tilde{H}_p^k = H_p(\mathbf{z}^k, \bar{u}, \bar{\mathbf{v}}, \mathbf{p}_p^k)\big|_{T^k} + \mu_k\,(T^k - T^{k-1})$
7:      Set $T^{k+1} = T^k - s\,\tilde{H}_p^k$ ($s > 0$ suff. small step size)
8:      If needed, extend functions $\bar{u}, \bar{\mathbf{v}}$ to $[0, T^{k+1}]$
9:      Set $k := k + 1$
10: until $|H_p\big|_{T^k}| < \epsilon$
```

### 2.4. Free End-Time Nash Games

In this section, we assemble the numerical optimization schemes discussed above to define our algorithm for solving the HC Nash game. We can say that it is a relaxation method with free end-time sub-problems and proximal penalty. A numerical issue that arises when considering the different free end-time optimal control problems for $P$ and $E$ is that the corresponding optimal final times may be different during iterates. However, the relaxation step requires to combine different approximations of the control functions that are given on different intervals. Thus, before the relaxation step is performed, an extrapolation/restriction procedure is applied to define all functions involved on the same time horizon. Our free end-time HC Nash game solver is given by Algorithm 4.

**Algorithm 4** Relaxation scheme with free end-time sub-problems and proximal penalty

```
1: input $u^0$, $\mathbf{v}^0$, $T^0$; parameters $\tau \in (0,1)$, $\alpha \in (0,1)$, $\epsilon > 0$, $\{\mu_k\} \to \infty$
2: Set $k = 0$
3: repeat
4:     Compute $(\bar{u}, T_u) = \arg\min_{u,T} \widetilde{J}_p(u, \mathbf{v}^k, T) + \frac{\mu_k}{2} \left(T - T^k\right)^2$
5:     Compute $(\bar{\mathbf{v}}, T_v) = \arg\min_{\mathbf{v},T} \widetilde{J}_e(u^k, \mathbf{v}, T) + \frac{\mu_k}{2} \left(T - T^k\right)^2$
6:     Compute $\bar{T} := (T_u + T_v)/2$
7:     Set $T^{k+1} := \alpha T^k + (1-\alpha)\bar{T}$
8:     If needed, extend functions $u^k$, $\mathbf{v}^k$, $\bar{u}$, and $\bar{\mathbf{v}}$ to $[0, T^{k+1}]$
9:     Set $(u^{k+1}, \mathbf{v}^{k+1}) := \tau(u^k, \mathbf{v}^k) + (1-\tau)(\bar{u}, \bar{\mathbf{v}})$
10:    Set $k := k + 1$
11: until $\max(|H_e|, |H_p|) < \epsilon$
```

In this algorithm, the parameters $\tau$ and $\alpha$ are relaxation factors that we specify in our numerical experiments. The main advantage of Algorithm 4 is that we can compute $(\bar{u}, T_u)$ and $(\bar{\mathbf{v}}, T_v)$ separately (in parallel) using an efficient optimization scheme. In this respect, Algorithm 4 is of Jacobi, and not of Gauß-Seidel type.

Notice that Algorithm 4 stops when the two Hamilton functions are smaller, in absolute value, than a given tolerance $\epsilon$. In fact, since our problem is autonomous, at optimality these Hamiltonians should have zero value.

A different approach, not further investigated in this paper, may consider replacing the free end-time optimal control problems at Steps 4–5 of Algorithm 4 with fixed end-time problems, with final time $T^k$. Successively, lacking of estimates of $T_u$ and $T_v$, Step 6 could find a $\bar{T}$ based on $\bar{u}$, $\bar{\mathbf{v}}$, and $T^k$. This approach would avoid solving free-time optimization problems, which may be a valuable property. On the other hand, this still needs extrapolation or truncation to match the time domains. Also, further decoupling the optimization problem may lead to a slower, less robust convergence. This is a subject for future research.

To conclude this section, we remark that we prefer avoiding time scaling techniques in our game framework and rather use a bilevel approach. However, time scaling provides useful tools for studying free end time optimal control problems. The core idea is to transform the original problem, with time variable $t \in [0, T]$, $T > 0$, into a fixed-time one by treating the final time $T$ as a parameter (or a constant state). By introducing the mapping $s \mapsto t(s) := Ts$, the new time variable is $s \in [0, 1]$. The system dynamics have to be transformed as well, scaled by $T$. Then, the problem is of finding the optimal control over the scaled time domain $[0, 1]$ and the optimal final time $T$. Given a final time $T > 0$, the (original) control $u \in U_{ad}$ can be recovered as $u(t) = u_s(t/T)$ for any $t \in [0, T]$. See [9,15] for more details.

## 3. Numerical experiments

In this section, we present results of numerical experiments to validate our HC Nash game formulation and the ability of our numerical framework to solve the resulting problems.

In our numerical experiments, we choose the initial state of the system $\mathbf{z}_0 = (3, 2)^T$ and initialize $T^0 = 10$, $\nu_u = 1$ and $\nu_v = 0.3$. Moreover, let $u^{00}(t) = 0$ and $\mathbf{v}^{00}(t) = (0.1, 0.1)'$, $t \in [0, T^0]$. With these parameters, we solve the HC game for fixed end-time $T^0$, in order to get a better initial guess for the controls, namely $u^0$ and $\mathbf{v}^0$, in our algorithm for a faster convergence. The extrapolation step is achieved by using the MATLAB function `interp1` with the `nearest` method [20]. Similar results are obtained with the `linear` method.

Next, assume that player $P$ has an advantage on player $E$. This situation results from an appropriate choice of the parameters of the game. For example, let $\nu_e = 10^{-6}$, $\nu_p = 10^{-6}$, $r_e = 10^{-4}$, $r_p = 1$ and $\omega_e = 0$, $\omega_p = 10^{-6}$.

Notice that this choice of parameters aims at having the pursuer catch the evader. Otherwise, the game will stop only by prescribing a maximun number of iterations, at which P and E will have different positions.

To solve this NE problem, we use Algorithm 4 with $\tau = 0.5$, $\alpha = 0.5$ and $\epsilon = 10^{-5}$. The regularization parameters are $\mu_k = \mu = 10^{-10}$. In this implementation, the differential HC model and its adjoint are approximated by the MCN scheme with a grid of $N_t = 500$ points. Notice that further experiments have been performed to verify that the results reported in this section are not mesh dependent.

With this setting, we obtain the $P$ and $E$ strategies depicted in Figure 1, and the time when the pursuer catches the evader results to be $T^* = 8.88$. This result is obtained after 15 iterations in Algorithm 4.

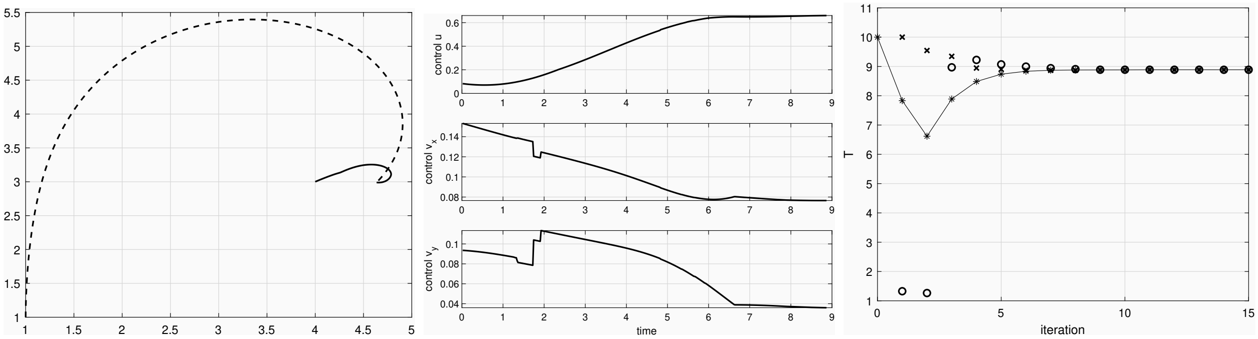

FIGURE 1. Test 1: (left) players' trajectories: pursuer in dashed line and evader in solid line; (middle) controls corresponding to the two players; and (right) final time during iterations: 'o' for the evader and 'x' for the pursuer.

Starting the algorithm with a different end-time initialization, e.g., $T^0 = 9$, we obtain $T^* = 8.81$ and the control functions are very similar to those obtained above, hence omitted for brevity.

We would like to comment on the fact that, even using the tolerance $\epsilon = 10^{-5}$ in the stopping condition, we get relatively different values for $T^*$ that means the end-time changes much more than the two Hamilton functions. Hence, we do not need a very strict tolerance.

However, it is well-known that Nash games admit many solutions, and these may result by different initializations. In fact, choosing $T^0 = 7$, we obtain $T^* = 10.14$ and the corresponding strategies and trajectories are shown in Figure 2.

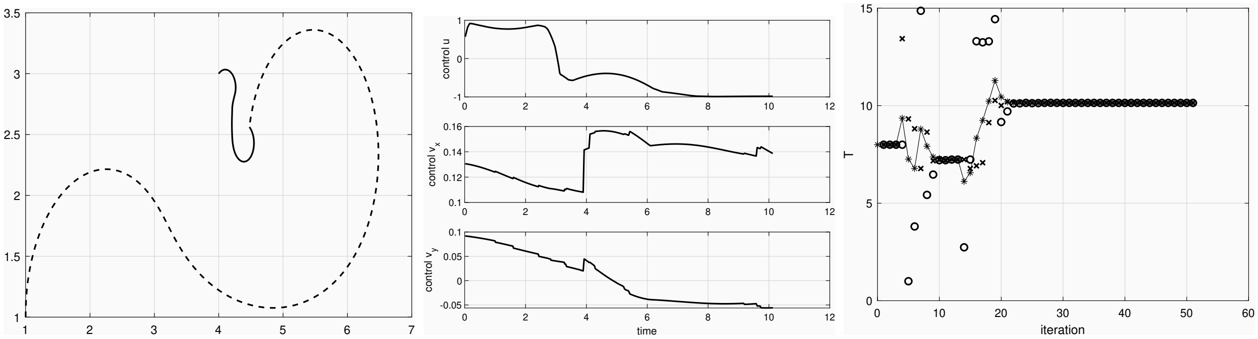

FIGURE 2. Test 2: (left) players' trajectories: pursuer in dashed line and evader in solid line; (middle) controls corresponding to the two players; and (right) final time during iterations: 'o' for the evader and 'x' for the pursuer.

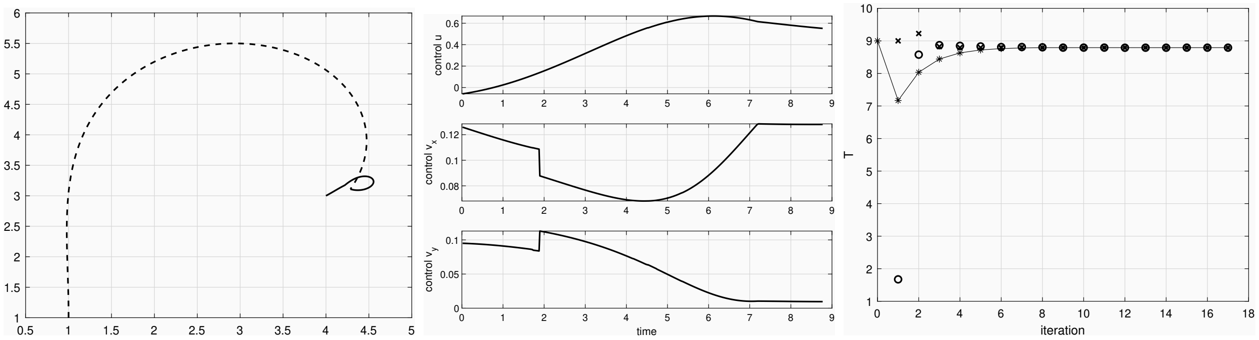

Figure 3. Test 3: (left) players' trajectories: pursuer in dashed line and evader in solid line; (middle) controls corresponding to the two players; and (right) final time during iterations: 'o' for the evader and 'x' for the pursuer.

We conclude this section showing that if we change the regularization terms $\nu_e = 10^{-6}$, $\nu_p = 10^{-4}$ and start with $T^0 = 9$ ( all the others parameters are as mentioned above), then we obtain (again) the optimal final time $T^* = 8.79$ and similar trajectories, as depicted in Figure 3.

The results of these experiments can be interpreted as follows. At the beginning of the optimization procedure, the evader tries to take the time to the lower bound of the time interval to avoid the chance to be captured. At the same time, the pursuer tries to get enough time to do it. After few iterations it holds the opposite, i.e., the evader aims to get more time to escape, while the pursuer tries to catch it earlier.

Moreover, as explained in [11], if the pursuer turns straight in the direction of the evader, then he can frustrate the pursuer by entering in the circle of maximal curvature. Hence, the pursuer should minimize their distance by moving around the evader until the capture.

## 4. Conclusion

In this paper, a numerical framework for solving a pursuit-evader homicidal chauffeur Nash game was presented. The Nash game was formulated in a functional setting involving cost functionals for the two players and the classical homicidal chauffeur differential models. In this model, the strategies of the pursuer and evader are represented by control functions that are subject to constraints. The numerical solution procedure was obtained combining a Hamiltonian based scheme with proximal penalty to determine the time horizon where the game takes place with a Lagrangian optimal control approach and relaxation to solve the Nash game at a fixed end-time.

## 5. Acknowledgement

We would like to thank G. Ciaramella, A. Habbal and S. Roy for many insightful discussions and helpful notes on differential games.

## 6. Appendix

In this appendix, we derive (2) from (1). For this purpose, we consider the following equations of the pursuer's coordinates in the evader's reference system with an angle of $\theta$ measured clockwise from the $y$-axis.

$$\begin{aligned} x &= (x_e - x_p)\cos\theta - (y_e - y_p)\sin\theta, \\ y &= (y_e - y_p)\cos\theta + (x_e - x_p)\sin\theta. \end{aligned} \tag{19}$$

By differentiation of (19) one gets

$$\begin{aligned}
\dot{x} &= (\dot{x}_e - \dot{x}_p)\cos\theta - (x_e - x_p)\sin\theta\,\dot{\theta} - (\dot{y}_e - \dot{y}_p)\sin\theta - (y_e - y_p)\cos\theta\,\dot{\theta},\\
\dot{y} &= (\dot{y}_e - \dot{y}_p)\cos\theta - (y_e - y_p)\sin\theta\,\dot{\theta} + (\dot{x}_e - \dot{x}_p)\sin\theta + (x_e - x_p)\cos\theta\,\dot{\theta}.
\end{aligned} \tag{20}$$

Hence, equations (2) are obtained by replacing in (20) the values of $\dot{x}_i, \dot{y}_i, \dot{\theta}$ given in (1) and by using the following geometric relations

$$x_e - x_p = y\sin\theta + x\cos\theta, \qquad y_e - y_p = y\cos\theta - x\sin\theta.$$

## References

[1] T. Başar. Relaxation techniques and asynchronous algorithms for on-line computation of non-cooperative equilibria. *Journal of Economic Dynamics and Control*, 11(4):531–549, 1987.

[2] L. D. Berkovitz. A variational approach to differential games. *Annals of Math Study*, 52:127–174, 1964.

[3] P. Bernhard. Pursuit-evasion games and zero-sum two-person differential games. In J. Baillieul and T. Samad, editors, *Encyclopedia of Systems and Control*, pages 1–7. Springer London, 2013.

[4] A. Borzì, G. Ciaramella, and M. Sprengel. *Formulation and Numerical Solution of Quantum Control Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

[5] A. Borzì and V. Schulz. *Computational Optimization of Systems Governed by Partial Differential Equations*. Society for Industrial and Applied Mathematics, 2011.

[6] M. H. Breitner, H. J. Pesch, and W. Grimm. 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*, 78(3):419–441, 1993.

[7] A. De Marchi and M. Gerdts. Free finite horizon LQR: a bilevel perspective and its application to model predictive control. *Automatica*, 100:299–311, Feb 2019.

[8] A. De Marchi and M. Gerdts. *An Iterative Method for Final Time Optimization in Nonlinear Optimal Control*, pages 60–66. 2019 Proceedings of the Conference on Control and its Applications, SIAM, 2019.

[9] M. Gerdts. A variable time transformation method for mixed-integer optimal control problems. *Optimal Control Applications and Methods*, 27(3):169–182, 2006.

[10] Y. Ho, A. Bryson, and S. Baron. Differential games and optimal pursuit-evasion strategies. *IEEE Transactions on Automatic Control*, 10(4):385–389, October 1965.

[11] R. Isaacs. Games of pursuit. *RAND Corporation*, 6:257, 1951.

[12] R. Isaacs. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization.* John Wiley, 1965.

[13] J. B. Krawczyk and S. Uryasev. Relaxation algorithms to find Nash equilibria with economic applications. *Environmental Modeling & Assessment*, 5(1):63–73, Jan 2000.

[14] S. Li and T. Başar. Distributed algorithms for the computation of noncooperative equilibria. *Automatica*, 23(4):523–533, 1987.

[15] Q. Lin, R. Loxton, and K. L. Teo. The control parameterization method for nonlinear optimal control: A survey. *Journal of Industrial & Management Optimization*, 10(1):275–309, 2014.

[16] W. Lin, Z. Qu, and M. A. Simaan. Nash strategies for pursuit-evasion differential games involving limited observations. *IEEE Transactions on Aerospace and Electronic Systems*, 51(2):1347–1356, April 2015.

[17] A. W. Merz. *The Homicidal Chauffeur- a differential game.* PhD thesis, Stanford University, 1971.

[18] V. S. Patsko and V. L. Turova. Homicidal chauffeur game: History and modern studies. In M. Breton and K. Szajowski, editors, *Advances in Dynamic Games: Theory, Applications, and Numerical Methods for Differential and Stochastic Games*, pages 227–251. Birkhäuser Boston, Boston, 2011.

[19] A. W. Starr and Y. C. Ho. Nonzero-sum differential games. *Journal of Optimization Theory and Applications*, 3(3):184–206, Mar 1969.

[20] The MathWorks, Inc. MATLAB Release 2018b, 2018. Natick, Massachusetts, United States.