Inverse reinforcement learning for identification of linear–quadratic zero-sum differential games

Martirosyan, E.; Cao, M.

Published in:
Systems & Control Letters

DOI:
10.1016/j.sysconle.2022.105438

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

Inverse reinforcement learning for identification of linear–quadratic zero-sum differential games

E. Martirosyan [*], M. Cao

Engineering and Technology Institute Groningen, University of Groningen, Nijenborgh 4, Groningen, 9712CP, Netherlands

ARTICLE INFO

Article history:
Received 10 April 2022
Received in revised form 28 September 2022
Accepted 10 December 2022
Available online xxxx

Keywords:
Inverse differential game
Identification
Optimal control
Optimization
Reinforcement learning

ABSTRACT

In this paper, we address the inverse problem in the case of linear–quadratic zero-sum differential games. The problem is to evaluate an unknown cost function given the observed trajectories that are known to be generated by a stationary linear feedback Nash equilibrium pair. Using the observed data, we construct a game that is equivalent to the game that leads to the observed trajectories in the sense that the equilibrium feedback law of any of the two player is the same for that player in the original and constructed games. Towards this end, we introduce a model-based algorithm that uses the given trajectories to accomplish this task. The algorithm combines both inverse optimal control and reinforcement learning methods making extensive use of gradient descent optimization for the latter. The analysis of the algorithm focuses on the proof of its convergence and stability. Simulation results validate the effectiveness of the proposed algorithm.

1. Introduction

Dynamic game theory brings together four components that are key to many situations in economics, ecology, and other related disciplines: optimizing behavior, the presence of multiple agents/ players, enduring consequences of decisions, and robustness with respect to the changing environment [1]. Non-cooperative differential games were first introduced in [2] within the framework of zero-sum games. This type of game attracted considerable attention from the control community due to the fact that quadratic differential games provide new angles to examine the performances of control laws. The application of differential games is far-reaching [3–5]. Although most of the literature has focused on determining the outcome of a game given the players' objective function, recently, an increasing interest appeared in the inverse problem, where, given the players' game-playing behavior, one wants to reverse engineer the objective of a player.

Inverse problems have attracted considerable attention due to, in part, their application in guiding the system to desired behavior outcomes. Significant research has been done in the area of inverse optimal control (IOC) [6,7]. Another closely related research area is inverse reinforcement learning (IRL) [8]. Although these two areas are concerned with similar problems, they are different in structure — the IOC aims to reconstruct an objective function given the state/action samples assuming dealing with a stable control system, while the IRL recovers an objective function using expert demonstration assuming that the expert behavior is optimal [9]. There is a close relationship between IOC and the inverse problem for linear quadratic differential games. There are various works dedicated to the inverse problem for non-cooperative linear–quadratic differential games. Some of them use purely IRL approaches [10,11], while others are based on IOC [12]. However, not much attention was paid to the linear–quadratic zero-sum differential games despite the fact that they might be used to solve the L_2-gain problem [13]. Some results for the inverse problem in such games were achieved via inverse Q-learning in the context of the imitation learning problem [14].

For linear–quadratic zero-sum differential games, finding the Nash equilibrium is done via solving the so-called Generalized Algebraic Riccati Equation (GARE) [15,16]. In this work, we use reinforcement learning methods and inverse optimal control methods to solve GARE. The developed algorithm is model-based, i.e., in addition to knowing equilibrium trajectories, we also know the weight matrices in the dynamics. Instead of seeking the cost function that, together with the dynamics, generated the observed behavior, we are looking for an equivalent cost function that, together with the given dynamics, constitutes a game that shares the same feedback law with the original game.

The paper is structured as follows. Section 2 provides preliminary results on linear–quadratic zero-sum differential games and formulates the problem addressed in the paper. In Section 3, we describe each step of the algorithm. Section 4 is dedicated

[*] Corresponding author.

E-mail addresses: e.n.martirosyan@rug.nl (E. Martirosyan), m.cao@rug.nl (M. Cao).

URLs: https://www.rug.nl/staff/e.n.martirosyan (E. Martirosyan), https://www.rug.nl/staff/m.cao (M. Cao).

https://doi.org/10.1016/j.sysconle.2022.105438

to the analysis of the algorithm; we show its convergence and stability and characterize possible solutions. Sections 5 and 6 provide simulation results and conclusion, respectively.

Notations: For a matrix $P \in \mathbb{R}^{m\times n}$, P^k, $P^{(k)}$ denote P to the power of k, and matrix P at the kth iteration, respectively. In addition, $P > 0$, $P \geq 0$, $P \leq 0$, and $P < 0$, denote positive (semi-)definiteness, and (semi-)negative definiteness of matrix P, respectively. $\mathrm{Tr}\, P$ denotes the trace of matrix P. I_k is the $k \times k$ identity matrix. $\mathbb{R}^+$ denotes the set of positive real numbers. $\mathbb{Z}^+$ denotes the set of positive integers.

2. Problem formulation

This section introduces linear–quadratic (LQ) zero-sum differential games and defines stationary linear feedback Nash equilibrium (referred to as NE). We clarify what an optimal behavior for the game is and introduce the inverse problem.

2.1. LQ zero-sum differential game

Consider a differential game with continuous time dynamics

$$\dot{x}(t) = Ax(t) + Bu(t) + Dd(t), \tag{1}$$

$$x(0) = x_0 \tag{2}$$

where $x \in \mathbb{R}^n$ is the state and $u \in \mathbb{R}^m$ and $d \in \mathbb{R}^p$ are control inputs of players 1 and 2, respectively; plant matrix A, control input matrices B and D have appropriate dimensions.

We consider that the players select their control to be linear time-invariant feedback laws of the form

$$u(t) = Fx(t), \tag{3}$$

$$d(t) = Lx(t), \tag{4}$$

where F and L are linear time invariant feedback matrices of players 1 and 2, respectively. Further, to ease notations, we use $x(t) = x$, $u(t) = u$ and $d(t) = d$.

Within the game, player 1 aims to find a controller that minimizes a cost function, and player 2, on the opposite, looks for a controller that maximizes it. The cost function is quadratic and given as follows

$$J(x_0, u, d) = \int_0^\infty \left(x^\top Qx + u^\top Ru - d^\top Md \right) dt, \tag{5}$$

where $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$, $M \in \mathbb{R}^{p\times p}$ are symmetric and $R, M > 0$.

In the game, we are interested in finding a Nash equilibrium (u^*, d^*) in the sense that

$$J(x(0), u^*, d) \leq J(x(0), u^*, d^*) \leq J(x(0), u, d^*), \tag{6}$$

that is $J(x(0), u^*, d^*) = \min_u \max_d J(x(0), u, d)$.

The optimal value function in the game is defined by

$$\begin{aligned} V^*(x) &:= \min_u \max_d \int_0^\infty \left(x^\top Qx + u^\top Ru - d^\top Md \right) dt \\ &= x^\top Kx \end{aligned} \tag{7}$$

where K is a symmetric matrix, sometimes referred to as the value matrix. Define $\nabla V^* := \left(\frac{\partial V^*}{\partial x}\right)$. The Hamiltonian function is

$$\begin{aligned} H(V^*, u, d) :=& x^\top Qx + u^\top Ru - d^\top Md \\ &+ \nabla V^{*\top}(Ax + Bu + Dd). \end{aligned} \tag{8}$$

Using the stationarity conditions

$$\frac{\partial H(V^*, u, d)}{\partial u} = 0, \quad \frac{\partial H(V^*, u, d)}{\partial d} = 0 \tag{9}$$

we obtain

$$u^* = -R^{-1}B^\top K^* x := F^* x, \tag{10}$$

$$d^* = M^{-1}D^\top K^* x := L^* x \tag{11}$$

where K^* satisfies the following *Generalized Algebraic Riccati Equation* (GARE) [1]

$$-A^\top K^* - K^* A + K^*(BR^{-1}B^\top - DM^{-1}D^\top)K^* - Q = 0. \tag{12}$$

In this game, we restrict the set of admissible controllers (F, L) to belong to the following set

$$\mathcal{F} = \{(F, L) | A + BF + DL \text{ is stable}\}, \tag{13}$$

since (u^*, d^*) need to stabilize trajectories to qualify as the unique NE equilibrium in this game [1]. This restriction is essential because, as shown in [17], without this restriction it is possible to provide an example where a non-stabilizing feedback yields lower cost for one of the player while another player sticks to the stabilizing feedback law. Thus, beside satisfying (12), K should also be stabilizing to qualify (F, L) as a unique NE [1]. The following assumption guarantees the non-emptiness of the set.

Assumption 1. $(A, [B, D])$ in (1) is stabilizable.

2.2. Inverse problem

We formulate the inverse problem for LQ zero-sum differential games in this subsection.

Consider an LQ differential game (referred to as the observed LQ game) with continuous-time system dynamics

$$\dot{x}_o = Ax_o + Bu_o + Dd_o, \tag{14}$$

$$x_o(0) = x_{0,o} \tag{15}$$

where $x_o \in \mathbb{R}^n$, $u_o \in \mathbb{R}^m$ and $d_o \in \mathbb{R}^p$ are NE trajectories of the observed LQ game with u and d being trajectories of players 1 and 2, respectively; A, B, D have appropriate dimensions and satisfy Assumption 1. The cost function of the game has the following known quadratic structure

$$J_o(x_0, u, d) = \int_0^\infty \left(x^\top Q_o x + u^\top R_o u - d^\top M_o d \right) dt, \tag{16}$$

with the *unknown* matrices $Q_o = Q_o^\top$, $R_o = R_o^\top > 0$ and $M_o = M_o^\top > 0$. Considering that (x_o, u_o, d_o) are NE trajectories, we have

$$u_o = F_o x = -R_o^{-1}B^\top K_o x, \tag{17}$$

$$d_o = L_o x = M_o^{-1}D^\top K_o x, \tag{18}$$

where K_o is the unique stabilizing symmetric solution of the following GARE

$$A^\top K_o + K_o A - K_o(BR_o^{-1}B^\top - DM_o^{-1}D^\top)K_o + Q_o = 0. \tag{19}$$

Assumption 2. $A + BF_o$ is stable.

The above assumption is the only restriction we have on the game that lead to the observed equilibrium trajectories. In fact, assumption that the unique stabilizing solution of (19) is positive definite, i.e., $K_o > 0$, leads to $A + BF_o$ being stable. Although the result is known [16], it lacks the proof which is presented below.

Lemma 1. *Consider the observed LQ game (A, B, C, Q_o, R_o, M_o) described in Section 2.2. Then if (F_o, L_o) is the feedback NE equilibrium pair and $K_o > 0$, then*

$$A + BF_o \quad \text{is a stable matrix.} \tag{20}$$

To see this, we use the fact that $F_o = R^{-1}B^\top K_o$ and $L_o = M_o^{-1}D^\top K_o$ constitute the equilibrium pair and $K_o > 0$. Hence,

$$\begin{aligned}&(A+BF_o+DL_o)^\top K_o + K_o(A+BF_o+DL_o) = \\ &\qquad - Q_o - F_o^\top R_o F_o + L_o^\top M_o L_o < 0.\end{aligned} \tag{21}$$

Moving the DL_o terms to the right-hand side, we get

$$(A+BF_o)^\top K_o + K_o(A+BF_o) = -Q_o - F_o^\top R_o F_o - L_o^\top M_o L_o. \tag{22}$$

From the inequality in (21), one can conclude that

$$-Q_o - F_o^\top R_o F_o - L_o^\top M_o L_o < 0 \tag{23}$$

and, as a result of $K_o > 0$, $A+BF_o$ is a stable matrix. ■

Positive definiteness of the value matrix is a common assumption for differential games [18]. Note that whether $A+BF_o$ is true or not can be checked using the estimation of F_o, which can be computed via the procedure described in 3.2.

Notation: we use the (A, B, D, Q, R, M) tuple to describe an LQ differential game with the dynamics' matrices A, B, D and the cost function parameters Q, R, M.

Definition 1 (*Equivalent Game*)**.** The (A, B, D, Q, R, M) game is called equivalent to the observed game (A, B, D, Q_o, R_o, M_o) with the value matrix K_o if for the selected Q, R and M, GARE (12) has a unique stabilizing solution K such that $R^{-1}B^\top K = R_o^{-1}B^\top K_o$, i.e., $F := -R^{-1}B^\top K = F_o$.

In other words, the games are equivalent if they share the same equilibrium feedback law of the player that minimizes the cost function (16), i.e. player 1.

Now, we are ready to formulate the inverse problem to be addressed in this paper.

Inverse Problem: Given the dynamics' matrices A, B, D and the observed trajectories (x_o, u_o), we want to derive a game equivalent to the (A, B, D, Q_o, R_o, M_o) game.

Remark 1. Since no assumptions on definiteness of Q_o are made, the problem can be reformulated for player 2, and the solution proposed further is still valid.

The goal is to be accomplished via a model-based inverse reinforcement learning algorithm described in the following section.

3. Model-based inverse learning

In this section, we describe the algorithm that uses trajectories (x_o, u_o) generated by *known* dynamics in (14) for learning a cost function equivalent to the one parametrized by (Q_o, R_o, M_o).

The procedure is as follows — firstly, we initialize an LQ differential game with dynamics (A, B, D). We generate an initial $Q^{(0)}$ updated in each iteration, $M^{(0)} > 0$ updated when necessary and, the control input weights $R > 0$ remaining the same in each iteration. The next step is to provide an estimation of F_o using the observed trajectories. Then, to solve the resulting LQ game, we solve GARE (12) to derive the unique stabilizing solution $K^{(0)}$. After that, we start the iterative update of $K^{(0)}$ using the gradient descent method [19] and update of $Q^{(0)}$ using the inverse optimal control method [20].

3.1. Optimal control on given cost function parameters

Following the first step, we need to initialize $Q^{(0)} = Q^{(0)\top}$, $R = R^\top > 0$ and $M^{(0)} = M^{(0)\top} > 0$. Moreover, we initialize M as $M^{(0)} = M^{(0)\top} = (\gamma^{(0)})^2 I_p$ where $\gamma^{(0)} \in \mathbb{R}^+$ (which allows us to write $D\big(M^{(0)}\big)^{-1}D^\top = (\gamma^{(0)})^{-2}DD^\top$. With the known dynamics A, B, D, one needs to solve the following GARE

$$\begin{aligned}&A^\top K^{(0)} + K^{(0)}A- \\ &K^{(0)}(BR^{-1}B^\top - (\gamma^{(0)})^{-2}DD^\top)K^{(0)} + Q^{(0)} = 0\end{aligned} \tag{24}$$

with respect to the symmetric $K^{(0)}$, which is the *unique stabilizing solution*. To solve (24) might not be straightforward because GARE is not guaranteed to have the desired solution due to the following term

$$BR^{-1}B^\top - (\gamma^{(0)})^{-2}DD^\top \tag{25}$$

which might be indefinite. However, since we have the freedom to choose $Q^{(0)}$, R and $\gamma^{(0)}$, referring again to [18], we initialize $Q^{(0)} \geq 0$ such that $(A, \sqrt{Q^{(0)}})$ is observable. Note that if the desired solution exists, it is unique [21,22].

Then, using the algorithm presented in [18] (Algorithm 3), with the initialized parameters $Q^{(0)}$, R, $\gamma^{(0)}$, through the iterative procedure, the process is guaranteed to converge to the unique stabilizing positive definite solution $K^{(0)} > 0$. Using the derived solution, we calculate the state feedback law of player 1 as follows

$$F^{(0)} = -R^{-1}B^\top K^{(0)}. \tag{26}$$

Together with $L^{(0)} = (\gamma^{(0)})^{-2}DD^\top K^{(0)}$, $F^{(0)}$ forms the NE pair for the initialized game $(A, B, C, Q^{(0)}, R, M^{(0)})$.

3.2. Gradient descent update

We aim at tracking the difference between the feedback law F that is the NE feedback law for the current iteration game and the desired feedback law F_o. Towards this end, we need to derive an estimation $\hat{F}_o$ of F_o. Given the (x_o, u_o) trajectories, we use the batch least-square (LS) method [23]. To estimate that matrix pair, we need $k \geq n$, $k \in \mathbb{Z}^+$ data samples from the trajectories, i.e.

$$\hat{x}_o = [x_o(t_1), \ldots, x_o(t_k)] \in \mathbb{R}^{n\times k}, \tag{27a}$$

$$\hat{u}_o = [u_o(t_1), \ldots, u_o(t_k)] \in \mathbb{R}^{m\times k}. \tag{27b}$$

These data samples are used to estimate F_o via (17), i.e.,

$$\hat{F}_o = -\hat{u}_o\hat{x}_o^\top(\hat{x}_o\hat{x}_o^\top)^{-1}. \tag{28}$$

When tracking the difference between $\hat{F}_o$ and $F^{(i)}$ at the ith iteration of the algorithm, we denote the difference function by

$$s^{(i)}\big(K\big) := F^{(i)} - \hat{F}_o = -R^{-1}B^\top K^{(i)} - \hat{F}_o. \tag{29}$$

Next, we define an error function as

$$E^{(i)}\big(K\big) := \mathrm{Tr}\big(s^{(i)\top}s^{(i)}\big), \tag{30}$$

which is a function of K that we aim to minimize. Employing the gradient descent method [19], we introduce the following update rule

$$\bar{K}^{(i)} = K^{(i)} - \alpha\frac{\partial E^{(i)}}{\partial K} = K^{(i)} - \alpha\frac{\partial\, \mathrm{Tr}\big(s^{(i)\top}s^{(i)}\big)}{\partial K}, \tag{31}$$

where $\alpha > 0$ is the learning rate and the partial derivative is

$$\begin{aligned}\frac{\partial E^{(i)}}{\partial K^{(i)}} &= K^{(i)}BR^{-1}R^{-1}B^\top + BR^{-1}R^{-1}B^\top K^{(i)} \\ &+ \hat{F}_o^\top R^{-1}B^\top + BR^{-1}\hat{F}_o \\ &= -(F^{(i)} - \hat{F}_o)^\top R^{-1}B^\top - BR^{-1}(F^{(i)} - \hat{F}_o) \\ &= -s^{(i)\top}R^{-1}B^\top - BR^{-1}s^{(i)}.\end{aligned} \tag{32}$$

Note that $\|s^{(i)}\|_{2,1}$ is bounded for each i if $K^{(0)}$ in $F^{(0)} = -R^{-1}B^\top K^{(0)}$ is a solution of the initialized GARE (24). In that case, using the fact that $\|s^{(i)}\|_{2,1} > \|s^{(i+1)}\|_{2,1}$ for $i = 0, 1, \ldots$ due to gradient descent update, we have

$$C = \|s^{(0)}\|_{2,1} > \|s^{(1)}\|_{2,1} > \cdots \geq 0. \tag{33}$$

Also, as explained in Section 2.1, we want to guarantee the stability of the resulting solution, i.e.,

$$A + BF^* + (\gamma^*)^{-2}DD^\top K^* \quad \text{is a stable matrix,} \tag{34}$$

where K^* is the goal of the optimization procedure described before, i.e., $-R^{-1}B^\top K^* = F^* = F_o$; and $\gamma^* > 0$ is a parameter that we might need to update starting from $\gamma^{(0)}$ to guarantee the stability of the resulting dynamics. Thus, to update $K^{(i)}$, we need to always check whether

$$A - BR^{-1}B^\top K^* + (\gamma^{(i)})^{-2}DD^\top K^{(i)} \quad \text{is a stable matrix,} \tag{35}$$

and if it is not the case, $\gamma^{(i)}$ needs to be increased. This update can be performed, for example, linearly, i.e., $\gamma^{(i+1)} = c\gamma^{(i)}$. As it is shown in Section 4.2 dedicated to the stability analysis, such a c always exists.

3.3. Inverse optimal control update

The last step is to update $Q^{(i)}$ using $\bar{K}^{(i)}$ received via the gradient descent update. We simply substitute the update value matrix into GARE (12)

$$\begin{aligned} Q^{(i+1)} &= \\ &- A^\top \bar{K}^{(i)} - \bar{K}^{(i)}A + \bar{K}^{(i)}(BR^{-1}B^\top - (\gamma^{(i+1)})^{-2}DD^\top)\bar{K}^{(i)}. \end{aligned} \tag{36}$$

We repeat the presented steps till $0 \le E^{(i)} < \epsilon$ where $\epsilon \in \mathbb{R}^+$ is a desired precision. The resulting Q^*, $M^* = (\gamma^*)^2 I_p$ together with some initialized R and the given dynamics (A, B, D), constitute an LQ game that is equivalent to the observed LQ game (A, B, D, Q_o, R_o, M_o) in the sense described in Definition 1. Hence, we get a new GARE and the feedback NE pair

$$A^\top K^* + K^*A - K^*(BR^{-1}B^\top - D(M^*)^{-1}D^\top)K^* + Q^* = 0, \tag{37}$$

$$F^* = -R^{-1}B^\top K^* = -R_o^{-1}B^\top K_o = F_o, \tag{38}$$

$$L^* = (\gamma^*)^{-2}D^\top K^* \neq \gamma_o^{-2}D^\top K_o = L_o, \tag{39}$$

where $M^* = (\gamma^*)^2 I_p$. To summarize, we present the whole procedure in **Algorithm 1**. The section thereafter provides the analysis of the proposed algorithm.

Remark 2. From the complexity point of view, the demanding parts of algorithm are finding solution of the game with initialized parameters $(Q^{(0)}, R, M^{(0)})$ and matrix multiplication done in the following steps. The algorithm proposed in [18], is used in our work to solve the initialized GARE. This algorithm is based on so-called Lyapunov Iterations. Methods to solve the Lyapunov Equations with respect to $K \in \mathbb{R}^{n\times n}$ usually have complexity $\mathcal{O}(n^3)$ [24]. The steps of the algorithm that require performing matrix multiplication via standard methods have complexity $\mathcal{O}(n^3 + n^2\ m + nm^2 + np^2)$. Hence, the overall computational complexity is $\mathcal{O}(n^3 + n^2\ m + nm^2 + np^2)$.

4. Analysis of the algorithm

In this section, we derive a few analytical results for the presented algorithm. Firstly, we show the convergence of the algorithm. Next, we show that $F^* = F_o$ and L^* constitute the equilibrium for the synthesized game. Finally, we provide some results on the characterization of possible solutions in the addressed inverse problem.

We introduce the following notations

$$\Gamma(\gamma^{(i)}) = \Gamma^{(i)} := BR^{-1}B^\top - (\gamma^{(i)})^{-2}DD^\top, \tag{46}$$

and

$$\phi^{(i)}(s) := \frac{\partial E^i}{\partial K^{(i)}} = -\left(s^{(i)\top}R^{-1}B^\top + BR^{-1}s^{(i)}\right). \tag{47}$$

Note that $\phi^{(i)}(s)$ for $i = 0, 1, \ldots$ is a symmetric matrix.

Algorithm 1 Model-based Inverse Learning Algorithm

1. Initialize $R = R^\top > 0$ and $\gamma^{(0)} > 0$. Initialize $Q^{(0)} = Q^{(0)\top} > 0$ such that $(A, \sqrt{Q^{(0)}})$ is observable for the *known* A and set $i = 0$. Solve GARE (24) with respect to $K^{(0)}$.
2. Estimate F_o using the observed trajectories as
$$\hat{F}_o = -\hat{u}_o\hat{x}_o^\top(\hat{x}_o\hat{x}_o^\top)^{-1}. \tag{40}$$
3. Compute
$$F^{(i)} = -R^{-1}B^\top K^{(i)}, \tag{41}$$
and evaluate the difference
$$s^{(i)} = F^{(i)} - F_o. \tag{42}$$
4. Update $K^{(i)}$ to $\bar{K}^{(i)}$ as
$$\bar{K}^{(i)} = K^{(i)} + \alpha\left(s^{(i)\top}R^{-1}B^\top + BR^{-1}s^{(i)}\right), \tag{43}$$
$$K^{(i+1)} = \bar{K}^{(i)}. \tag{44}$$
5. **If** $A+B\hat{F}_o+(\gamma^{(i)})^{-2}DD^\top K^{(i+1)}$ is not stable, then $(\gamma^{(i+1)})^{-2} = c^{(i+1)})(\gamma^{(i)})^{-2}$ where $c^{(i+1)} > \bar{c}^{(i+1)} \ge 1$. **Otherwise**, $\gamma^{(i+1)} = \gamma^{(i)}$, i.e. $c^{(i+1)} = 1$.
6. Perform evaluation of $Q^{(i+1)}$ as
$$\begin{aligned} Q^{(i+1)} = &- A^\top \bar{K}^{(i)} - \bar{K}^{(i)}A + \\ &\bar{K}^{(i)}(BR^{-1}B^\top - (\gamma^{(i+1)})^{-2}DD^\top)\bar{K}^{(i)}. \end{aligned} \tag{45}$$
7. Set $i = i + 1$. Perform steps 3-5 till $E^{(i)} = \mathrm{Tr}(s^{(i)\top}s^{(i)}) < \epsilon$ where $\epsilon > 0$ is a small constant.

4.1. Convergence analysis

The first result claims the convergence of the proposed algorithm.

Theorem 1. *In Algorithm 1, the reward weight $Q^{(i)}$ converges to Q^* such that GARE (12), associated with a game (A, B, D, Q^*, R, M^*), has solution K^* such that*

$$R^{-1}B^\top K^* = R_o^{-1}B^\top K_o. \tag{48}$$

Proof. Let us consider (43). Using the gradient descent method to update $K^{(i)}$, we drive $F^{(i)}$ to $\hat{F}_o$. Hence, the error decreases with each iteration, i.e.,

$$E^{(i)} > E^{(i+1)} \ge 0, \quad \text{for all} \quad i = 0, 1, 2, \ldots, \tag{49}$$

and we have

$$\lim_{i\to\infty} E^{(i)} = 0, \quad \lim_{i\to\infty} s^{(i)} = 0 \quad \text{and} \quad \lim_{i\to\infty} \phi^{(i)}(s) = 0. \tag{50}$$

Then,

$$\lim_{i\to\infty} K^{(i+1)} = \lim_{i\to\infty} \bar{K}^{(i)} = \lim_{i\to\infty}(K^{(i)} - \alpha\phi^{(i)}(s)) = \lim_{i\to\infty} K^{(i)}. \tag{51}$$

Considering the effectiveness of LS estimation $\hat{F}_o = F_o$, we have

$$\lim_{i\to\infty} R^{-1}BK^{(i)} = \lim_{i\to\infty} F^{(i)} = F_o = R_o^{-1}BK_o. \tag{52}$$

We denote $\lim_{i\to\infty} K^{(i)} = K^*$. It is clear that when $K^{(i)}$ converges to K^*, $\gamma^{(i)}$ also converges to some $\gamma^* \ge \gamma^{(i)}$ for $i = 1, 2, \ldots$, i.e., $\lim_{i\to\infty} \gamma^{(i)} = \gamma^*$ or $\lim_{i\to\infty} c^{(i)} = 1$. Next, using the gradient update rule

$$\bar{K}^{(i)} = K^{(i)} - \alpha\phi^{(i)}(s), \tag{53}$$

we expand $\bar{K}^{(i)}$ in (36) and get

$$\begin{aligned} Q^{(i+1)} = &-(A^\top K^{(i)} + K^{(i)}A - K^{(i)}\Gamma^{(i)}K^{(i)}) \\ &+ \alpha(A^\top\phi^{(i)}(s) + \phi^{(i)}(s)A) \\ &- \alpha(K^{(i)}\Gamma^{(i+1)}\phi^{(i)}(s) + \phi^{(i)}(s)\Gamma^{(i+1)}K^{(i)}) \\ &+ \alpha^2\phi^{(i)}\Gamma^{(i+1)}\phi^{(i)}(s) + \\ &(1 - c^{(i+1)})(\gamma^{(i)})^{-2}K^{(i)}DD^\top K^{(i)}. \end{aligned} \tag{54}$$

Taking the limit of both sides and using (50), we get

$$\begin{aligned} \lim_{i\to\infty} Q^{(i+1)} &= -\lim_{i\to\infty}(A^\top K^{(i)} + K^{(i)}A - K^{(i)}\Gamma^{(i)}K^{(i)}) \\ &= \lim_{i\to\infty} Q^{(i)}. \end{aligned} \tag{55}$$

Then, we denote $\lim_{i\to\infty} Q^{(i)} = Q^*$. Thus, one can conclude the following

$$\begin{aligned} Q^* = \lim_{i\to\infty} Q^{(i)} &= -\lim_{i\to\infty}(A^\top K^{(i)} + K^{(i)}A - K^{(i)}\Gamma^{(i)}K^{(i)}) \\ &= -(A^\top K^* + K^*A - K^*\Gamma^*K^*), \end{aligned} \tag{56}$$

which shows that K^* satisfying $R^{-1}BK^* = F_o$ is a solution of GARE associated with (A, B, D, Q^*, R, M^*). ■

4.2. Stability analysis

In this section, we show the stability of the proposed algorithm.

Since $A + BF_o$ is a stable matrix, for any $K^{(i)}$ it will always be possible to find $\gamma^{(i)} > 0$ such that $A + BF_o$ dominates $(\gamma^{(i)})^{-2}DD^\top K^{(i)}$.

We give some more details on the initial choice of $\gamma^{(0)}$. Notice that before implementing check in step 5 and iterative update of $Q^{(i)}$ in step 6, steps $3 - 4$ in Algorithm 1 only require initialized $K^{(0)}$. Thus, we can always evaluate K^* and see whether for the initialized $\gamma^{(0)}$ the resulting solution K^* is a stabilizing one. Note that if GARE (12) for some γ_1, γ_2 such that $0 < \gamma_2 < \gamma_1$ and some fixed A, B, D, R, Q has solution, then $K(\gamma_1) < K(\gamma_2)$ [21].

Before presenting the result on the stability of the proposed algorithm, we use the following result from [1].

Theorem 2. *Consider an LQ zero-sum differential game described by (1) with the cost function given by (5). The game has for every initial state a feedback NE if and only if the following Riccati equation*

$$-A^\top K - KA + K(BR^{-1}B^\top - DM^{-1}D^\top)K - Q = 0 \tag{57}$$

has a symmetric solution such that the matrix $A - BR^{-1}B^\top K + \gamma^{-2} DD^\top K$ is stable. Moreover, the pair of $F = -R^{-1}B^\top K$ and $L = M^{-1}D^\top K$ constitutes the unique equilibrium.

Finally, the following can be concluded for the algorithm.

Theorem 3. *The output of Algorithm* 1*, given the observed trajectories (x_o, u_o) generated by a game (A, B, D, Q_o, R_o, M_o) described in Section* 2*, is the tuple (Q^*, R, M^*) such that, combined with the known dynamics (A, B, D), it forms a game with the unique NE feedback law for player* 1 *identical to (A, B, D, Q_o, R_o, M_o) game, i.e., $F^* = F_o$.*

Proof. In view of Theorem 1 and the validity of (34) via step 5 in Algorithm 1, one concludes that K^* is both a solution of GARE (57) and stabilizing. ■

Corollary 1. *There exist $\bar{\alpha} > 0$ and N such that for $i = N, N+1, \ldots$ Algorithm* 1 *produces $Q^{(i)}$ that together with $(A, B, D, R, M^{(i)})$, where $M^{(i)} = \gamma^{(i)}I_p$, forms GARE where $K^{(i)}$ is a stabilizing solution, i.e.,*

$$(A + BF^{(i)} + DL^{(i)}) < 0. \tag{58}$$

Proof. Note that

$$A + BF^{(i)} + DL^{(i)}, \tag{59}$$

using (42), can be rewritten as

$$A + B(F_o + s^{(i)}) + DL^{(i)}. \tag{60}$$

As shown in Lemma 1, $A + BF_o$ is stable; and $\gamma^{(i)}$ in $L^{(i)}$ is updated if needed in a way to guarantee the stability of $A + BF_o + DL^{(i)}$. Thus, the term $Bs^{(i)}$ is the one that might violate the stability of (60). However, $s^{(i)}$ is decreasing with each i and, starting from $i = N, N+1, \ldots, Bs^{(i)}$ is small enough so (60) is satisfied.

Now, we have that $(F^{(N)}, L^{(N)})$ and (F^*, L^*), where $F^* = F_o$ is the terminal feedback law for player 1, are both stabilizing pairs, i.e.,

$$\begin{aligned} &A + BF^{(N)} + DL^{(N)} = \\ &A - BR^{-1}B^\top K^{(N)} + (\gamma^{(N)})^{-2}DD^\top K^{(N)} < 0, \end{aligned} \tag{61}$$

$$\begin{aligned} &A + BF^* + DL^* = \\ &A - BR^{-1}B^\top K^* + (\gamma^*)^{-2}DD^\top K^* < 0. \end{aligned} \tag{62}$$

Since $K^{(i)}$ linearly affects $(F^{(i)}, L^{(i)})$ for $i = N, N+1, \ldots$ and $(F^{(i)}, L^{(i)})$ is a result of the gradient descent update from $(F^{(N)}, L^{(N)})$ in the direction of (F^*, L^*), there exists $\alpha = \bar{\alpha}$ in (43) that guarantees the stability of $(F^{(i)}, L^{(i)})$ [19]. Hence, $Q^{(i)}$, updated via (36) using $K^{(i)}$, is stabilizing. This completes the proof. ■

Remark 3. *N might be reduced by increasing $\gamma^{(i)}$ since bigger $\gamma^{(i)}$* changes the eigenvalues of $A + BF^{(i)} + DL^{(i)}$ (that are all negative) in a non-increasing way. In fact, picking $\gamma^{(0)}$ such that $K^{(0)}$ and resulting K^* for $\gamma^* = \gamma^{(0)}$ are both stabilizing, which guarantees that every $K^{(i)}$ is stabilizing, i.e., $N = 0$. Practical advice for implementing the algorithm would be to choose "high" $\gamma^{(0)}$ from the beginning.

4.3. Characterization of the solutions

In this section, we provide a discussion and results on the characterization of the possible output of the algorithm.

Note that we are looking for (Q^*, R, M^*) such that with the *known* (A, B, C) that form GARE (12) that has a stabilizing solution K^* satisfying $R_o^{-1}B^\top K_o = R^{-1}B^\top K^*$. Since $R > 0$, $B^\top K^* = RR_o^{-1}B^\top K_o$. If B has no full rank, there might be an infinite number of possible K^* [14].

Remark 4. All possible outputs of Algorithm 1, i.e., Q^*, γ^* and K^*, satisfy the following equality

$$\begin{aligned} &A^\top(K_o - K^*) + (K_o - K^*)A + F_o^\top(R - R_o)F_o - \\ &\gamma^*L^{*\top}L^* + L_o^\top M_oL_o = Q_o - Q. \end{aligned} \tag{63}$$

(63) is received via subtracting (37) from (19).

Let us denote

$$\begin{aligned} &Q_d = Q^* - Q_o, \quad K_d = K^* - K_o, \quad R_d = R - R_o \\ &\beta^* = (\gamma^*)^{-2}, \quad \beta_o = \gamma_o^{-2}, \quad \beta_d = \beta^* - \beta_o. \end{aligned} \tag{64}$$

Corollary 2. *K_d and Q_d satisfy the following equality*

$$\begin{aligned} &Q_d + A^\top K_d + K_dA - F_oR_dF_o + \\ &\beta_d(K_d + K_o)DD^\top(K_d + K_o) + \\ &\beta_o(K_dDD^\top K_d + K_dDD^\top K_o + K_oDD^\top K_d) = 0. \end{aligned} \tag{65}$$

Proof. Considering (37) and (64), we receive

$$Q^* + A^\top K^* + K^*A - \tag{66}$$

$$K^*(BR^{-1}B^\top - (\gamma^*)-2DD^\top)K^* =$$
$$Q_d + Q_o + A^\top K_d + A^\top K_o + K_d A + K_o A -$$
$$K^*BR^{-1}B^\top K^* + \beta^* K^* DD^\top K^*. \tag{67}$$

Expanding the two last terms and using $RR_o^{-1}B^\top K_o = B^\top K^*$, we get

$$K^*BR^{-1}B^\top K^* = K_o BR_o^{-1}RR_o^{-1}B^\top K_o =$$
$$K_o BR_o^{-1}(R_d + R_o)R^{-1}B^\top K_o \tag{68}$$

and

$$\beta^* K^* DD^\top K^* = \beta_o K_o DD^\top K_o +$$
$$\beta_d(K_d + K_o)DD^\top(K_d + K_o) +$$
$$\beta_o(K_d DD^\top K_d + K_d DD^\top K_o + K_o DD^\top K_d). \tag{69}$$

Substituting (68) and (69) into (66) and using (19), (37) with

$$K_o BR_o^{-1}R_d R_o^{-1}B^\top K_o = F_o^\top R_d F_o,$$

we arrive (65). ■

5. Simulations

In this section, we present simulation results of the model-based algorithm developed in this paper.

5.1. Simulation results 1

Consider the following continuous time system dynamics

$$\dot{x} = Ax + Bu + Dd, \tag{70}$$

where

$$A = \begin{pmatrix} 3 & -2 \\ 2 & -4 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad D = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{71}$$

The observed NE trajectories are generated for the game with the following weight matrices

$$Q_o = \begin{pmatrix} 7 & 2 \\ 2 & 5 \end{pmatrix}, \quad R_o = 2, \quad M_o = \begin{pmatrix} 5 & 0 \\ 0 & 3 \end{pmatrix}. \tag{72}$$

Given this game, F_o and K_o are

$$F_o = \begin{pmatrix} -5.8515 & 1.4358 \end{pmatrix} \quad K_o = \begin{pmatrix} 11.7030 & -2.8716 \\ -2.8716 & 1.6603 \end{pmatrix}.$$

The initialized parameters are the following

$$Q^{(0)} = 3I_{2\times 2}, \quad R = 3, \quad M^{(0)} = (\gamma^{(0)})^2 I_{2\times 2}, \tag{73}$$

with $\gamma^{(0)} = 2$. The learning rate is set to $\alpha = 0.1$.

The solution generated by the algorithm is

$$Q^* = \begin{pmatrix} 9.9875 & 3.4593 \\ 3.4593 & 3.4247 \end{pmatrix}, \quad \gamma^* = 2, \tag{74}$$

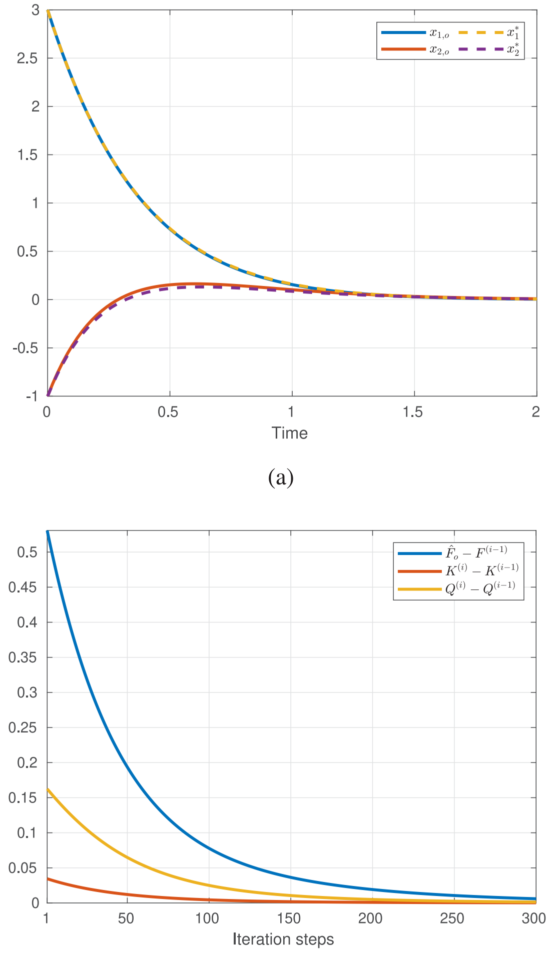
with

$$F^* = \begin{pmatrix} -5.8509 & 1.4418 \end{pmatrix}, \; K^* = \begin{pmatrix} 17.5528 & -4.3251 \\ -4.3251 & 1.9330 \end{pmatrix}. \tag{75}$$

In addition,

$$L^* = \gamma^{-2}DD^\top K^* = \begin{pmatrix} 0 & 0 \\ -1.0813 & 0.4832 \end{pmatrix}. \tag{76}$$

The resulting dynamics $A + BF + DL < 0$ is stable as shown in Fig. 1(a). The convergence of the iterative procedure is shown in Fig. 1(b).



(a)

(b)

Fig. 1. (a) The stability of the observed and resulting dynamics. (b) Convergence of the norm for iterations of $F^{(i)}$, $K^{(i)}$ and $Q^{(i)}$.

5.2. Simulation results 2

In this example, we use the dynamics and the cost function provided in [18]. Consider the following continuous time system dynamics

$$\dot{x} = Ax + Bu + Dd, \quad \text{where} \quad A = \begin{pmatrix} 1 & 0 & 3 & 0 \\ 0 & -2 & 3 & 0 \\ 0 & 1 & -3 & 0 \\ 1 & 0 & 0 & 4 \end{pmatrix}, \tag{77}$$

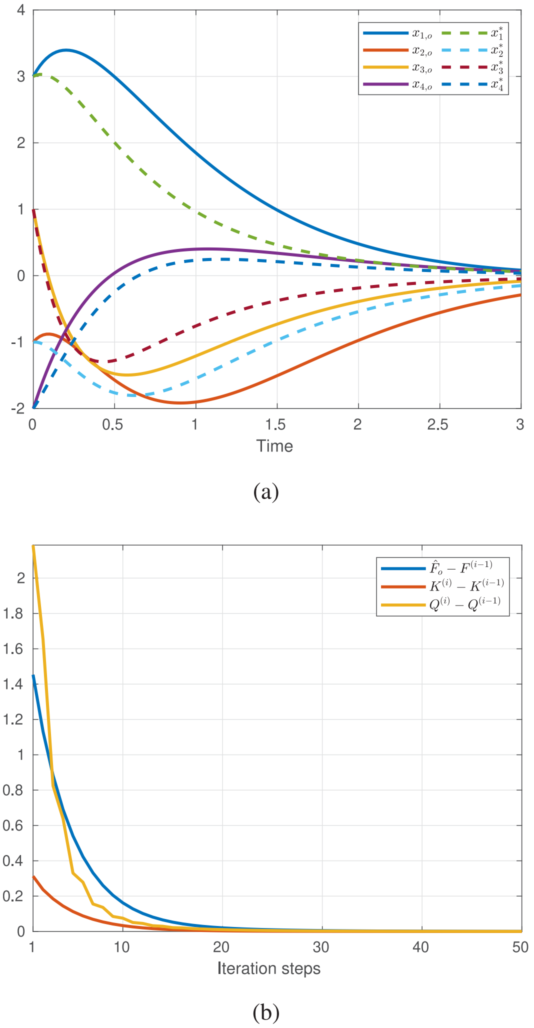
$$B = \begin{pmatrix} 0.0116 & 0.6020 \\ 0.9215 & 0.5565 \\ 0.5450 & 0.0730 \\ 0.5565 & 0.3834 \end{pmatrix}, \quad D = \begin{pmatrix} 0.4814 \\ 0.3909 \\ 0.4087 \\ 0.5591 \end{pmatrix}. \tag{78}$$

The observed NE trajectories are generated for the game with the following weight matrices

$$Q_o = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad R_o = I_{2\times 2}, \quad M_o = 1. \tag{79}$$

Given this game, F_o and K_o are

$$F_o = \begin{pmatrix} -2.2299 & -0.7173 & -2.0091 & 0 \\ -2.8548 & -0.5269 & -2.2088 & 0 \end{pmatrix}, \tag{80}$$



(a)

(b)

Fig. 2. (a) The stability of the observed and resulting dynamics. (b) Convergence of the norm for iterations of $F^{(i)}$, $K^{(i)}$ and $Q^{(i)}$.

$$K_o = \begin{pmatrix} 3.0103 & 0.3834 & 2.1315 & 0 \\ 0.3834 & 0.3875 & 0.5205 & 0 \\ 2.1315 & 0.5205 & 1.8134 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{81}$$

The initialized parameters are the following

$$Q^{(0)} = 3I_{4\times 4}, \quad R = 0.5I_{2\times 2}, \quad M^{(0)} = (\gamma^{(0)})^2, \tag{82}$$

with $\gamma^{(0)} = \sqrt{2}$. The learning rate is set to $\alpha = 0.1$.

The solution generated by the algorithm is

$$Q^* = \begin{pmatrix} 2.7575 & 0.7219 & 1.3039 & -0.5685 \\ 0.7219 & 1.6441 & -1.2203 & -0.4297 \\ 1.3039 & -1.2203 & 3.0806 & -0.3972 \\ -0.5685 & -0.4297 & -0.3972 & 2.4634 \end{pmatrix}, \tag{83}$$

with $\gamma^* = \sqrt{2}$ and

$$F^* = \begin{pmatrix} -2.2298 & -0.7171 & -2.0091 & -0.0001 \\ -2.8550 & -0.5271 & -2.2088 & 0.0002 \end{pmatrix}, \tag{84}$$

$$K^* = \begin{pmatrix} 1.5801 & 0.3175 & 1.0288 & -0.0766 \\ 0.3175 & 0.4113 & 0.1715 & -0.0912 \\ 1.0288 & 0.1715 & 1.0409 & -0.1255 \\ -0.0766 & -0.0912 & -0.1255 & 0.3081 \end{pmatrix}. \tag{85}$$

In addition,

$$L^* = \gamma^{-2} D D^\top K^* = \begin{pmatrix} 0.6312 & 0.1664 & 0.4588 & 0.0242 \end{pmatrix}. \tag{86}$$

The resulting dynamics $A + BF + DL < 0$ is stable as shown in Fig. 2(a). The convergence of the iterative procedure is shown in Fig. 2(b).

6. Conclusion

In this paper, we provided the algorithm that solves the inverse problem for linear–quadratic zero-sum differential games. We showed that the algorithm's output is the set of weight matrices that together with the known dynamics form an equivalent game for one of the players. After proving the convergence of the algorithm to a desired output, we provided simulations to demonstrate the effectiveness of the proposed method.

The presented algorithm has the potential for extension to model-free (neither plant matrix A nor control input matrices B, D are unknown) or to partially model-free (plant matrix A is unknown) settings. The steps of the algorithm that require plant matrix A are related to solving the initialized GARE and the inverse update of matrix Q. These steps might be implemented in different ways if methods to find the optimal controller for ARE without knowledge of dynamics are exploited [25,26]. Note that in the case of unknown control input matrices, the gradient update step might require changes in order to avoid using matrix B (or D in the case for player 2).

For future work, the case of a general-sum game will be considered, where instead of GARE, described in this work, the coupled algebraic Riccati equation arise [18].

CRediT authorship contribution statement

E. Martirosyan: Methodology, Writing – original draft. **M. Cao:** Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ming Cao reports financial support was provided by European Research Council. Co-author prof. Ming Cao serves as a senior editor for the Systems and Control Letters journal.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported in part by the European Research Council (ERC-CoG-771687).

References

[1] J. Engwerda, LQ Dynamic Optimization and Differential Games, John Wiley & Sons, 2005.
[2] R. Isaacs, Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization, SIAM Series in Applied Mathematics, Wiley, 1965.
[3] M. Flad, L. Fröhlich, S. Hohmann, Cooperative shared control driver assistance systems based on motion primitives and differential games, IEEE Trans. Hum.-Mach. Syst. 47 (5) (2017) 711–722.
[4] T. Mylvaganam, M. Sassano, A. Astolfi, A differential game approach to multi-agent collision avoidance, IEEE Trans. Automat. Control 62 (8) (2017) 4229–4235.
[5] D. Gu, A differential game approach to formation control, IEEE Trans. Control Syst. Technol. 16 (1) (2008) 85–93.

[6] M. Menner, M.N. Zeilinger, Convex formulations and algebraic solutions for linear quadratic inverse optimal control problems, in: 2018 European Control Conference, ECC, 2018, pp. 2107–2112.

[7] F. Jean, S. Maslovskaya, Inverse optimal control problem: the linear–quadratic case, in: 2018 IEEE Conference on Decision and Control, CDC, IEEE, Miami Beach, FL, 2018, pp. 888–893.

[8] A.Y. Ng, S. Russell, Algorithms for inverse reinforcement learning, in: Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, 2000, pp. 663–670.

[9] N. Ab Azar, A. Shahmansoorian, M. Davoudi, From inverse optimal control to inverse reinforcement learning: A historical review, Annu. Rev. Control 50 (2020) 119–138.

[10] J. Inga, E. Bischoff, T.L. Molloy, M. Flad, S. Hohmann, Solution sets for inverse non-cooperative linear–quadratic differential games, IEEE Control Syst. Lett. 3 (4) (2019) 871–876, Conference Name: IEEE Control Systems Letters.

[11] F. Köpf, J. Inga, S. Rothfuß, M. Flad, S. Hohmann, Inverse reinforcement learning for identification in linear–quadratic dynamic games, IFAC-PapersOnLine 50 (1) (2017) 14902–14908.

[12] T.L. Molloy, J. Inga, M. Flad, J.J. Ford, T. Perez, S. Hohmann, Inverse open-loop noncooperative differential games and inverse optimal control, IEEE Trans. Automat. Control 65 (2) (2020) 897–904.

[13] F.L. Lewis, D.L. Vrabie, V.L. Syrmos, Optimal Control, third ed., Wiley, Hoboken, 2012.

[14] B. Lian, W. Xue, F.L. Lewis, T. Chai, Robust inverse Q-learning for continuous-time linear systems in adversarial environments, IEEE Trans. Cybern. (2021) 1–13.

[15] K.G. Vamvoudakis, D. Vrabie, F.L. Lewis, Online learning algorithm for zero-sum games with integral reinforcement learning, J. Artif. Intell. Soft Comput. Res. (2011) 18.

[16] T. Basar, G. Olsder, Dynamic Noncooperative Game Theory: Second Edition, in: Classics in Applied Mathematics, Society for Industrial and Applied Mathematics, 1999.

[17] E. Mageirou, Values and strategies for infinite time linear quadratic games, IEEE Trans. Automat. Control 21 (4) (1976) 547–550.

[18] T.-Y. Li, Z. Gajic, Lyapunov iterations for solving coupled algebraic Riccati equations of Nash differential games and algebraic Riccati equations of zero-sum games, in: G.J. Olsder (Ed.), New Trends in Dynamic Games and Applications, Birkhäuser Boston, Boston, MA, 1995, pp. 333–351.

[19] D. Bertsekas, Nonlinear Programming, Athena Scientific, 1999.

[20] W. Haddad, V. Chellaboina, Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach, Princeton University Press, 2011.

[21] G. Hewer, Existence theorems for positive semidefinite and sign indefinite stabilizing solutions of H_∞ Riccati equations, SIAM J. Control Optim. 31 (1) (1993) 16–29.

[22] I.R. Petersen, Some new results on algebraic Riccati equations arising in linear quadratic differential games and the stabilization of uncertain linear systems, Systems Control Lett. 10 (5) (1988) 341–348.

[23] J.L. Devore, Probability and Statistics for Engineering and the Sciences, eighth ed., Brooks/Cole, ISBN: 978-0-538-73352-6, 2011.

[24] G. Golub, C. Van Loan, Matrix Computations, in: Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 2013.

[25] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, F. Lewis, Adaptive optimal control for continuous-time linear systems based on policy iteration, Automatica 45 (2) (2009) 477–484.

[26] Y. Jiang, Z.-P. Jiang, Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics, Automatica 48 (10) (2012) 2699–2704.