2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

A Policy Iteration Algorithm for N-player General-Sum Linear Quadratic Dynamic Games

Yuxiang Guan Giulio Salizzoni Maryam Kamgarpour Tyler H. Summers

2024 IEEE 63rd Conference on Decision and Control (CDC) | 979-8-3503-1633-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/CDC56724.2024.10886048

***Abstract*— We present a policy iteration algorithm for the infinite-horizon N-player general-sum deterministic linear quadratic dynamic games and compare it to policy gradient methods. We demonstrate that the proposed policy iteration algorithm is distinct from the Gauss-Newton policy gradient method in the N-player game setting, in contrast to the single-player setting where under suitable choice of step size they are equivalent. We illustrate in numerical experiments that the convergence rate of the proposed policy iteration algorithm significantly surpasses that of the Gauss-Newton policy gradient method and other policy gradient variations. Furthermore, our numerical results indicate that, compared to policy gradient methods, the convergence performance of the proposed policy iteration algorithm is less sensitive to the initial policy and changes in the number of players.**

I. Introduction

In recent years, the field of multi-agent reinforcement learning (MARL) has attracted significant interest within the reinforcement learning (RL) community. This interest has led to a series of successful developments in approximately solving sequential multi-agent decision-making problems, such as multi-robot control [1], autonomous driving [2], and the networking of communication packages [3]. Despite these successes, a comprehensive theoretical understanding of how MARL algorithms perform in environments where cooperation and/or competition among agents exists remains elusive. Recently, there has been a surge of interest in analyzing the performance of policy gradient algorithms within the context of linear quadratic dynamic games (LQDGs). LQDGs present a compelling framework for evaluating the efficacy of MARL algorithms in continuous state and action spaces due to their ability to admit a Nash equilibrium in linear feedback policies. Moreover, this equilibrium can be determined by solving a set of coupled Riccati equations. Properties of this equilibrium have been thoroughly analyzed in [4], [5], [6], [7], [8].

A majority of existing literature [9], [10], [11] has emphasized zero-sum LQDGs with two players, where it has been shown that certain policy gradient methods have global convergence guarantees in such settings. However, some negative results in [12] suggested that the vanilla/standard policy gradient method has no guarantees of even local convergence in infinite-horizon general-sum deterministic LQDGs. This result indicates potential limitations for gradient-type methods in such games. The natural policy gradient and vanilla/standard policy gradient methods demonstrated global convergence in a finite-horizon stochastic LQDG [13], given appropriate step size and the introduction of specific noise levels into the dynamic system. However, the natural policy gradient method may fail to converge to the Nash equilibrium of a deterministic LQDG without careful selection of initial policies and step sizes.

The policy iteration algorithm is well-known in single-player settings for computing optimal policies. It comprises two components: policy evaluation and policy improvement. This algorithm has been extensively studied in dynamic programming and RL and bears a close relationship to the Newton method [14], [15]. In the context of single-player LQDGs, the standard policy iteration algorithm [16] is equivalent to an instance of the Gauss-Newton policy gradient method with a specific step size. Several policy iteration-based RL algorithms have been studied for solving multi-player nonzero-sum differential games [17]. Recently, policy iteration algorithms have been developed for solving N-player nonzero-sum LQDGs [18] which explicitly formulate the policy evaluation and policy update steps. An off-line policy iteration-based RL algorithm was introduced for a two-player nonzero-sum LQDG in [19]. This algorithm is a special two-player case of the one we are presenting here. However, a more general policy iteration algorithm for N-player general-sum LQDGs, along with a comparison of its convergence performance against policy gradient methods, has not yet been undertaken.

In this paper, our main contribution is to present a policy iteration algorithm for the infinite-horizon N-player general-sum deterministic LQDGs and compare it to policy gradient methods. In contrast to the single-player setting, where the proposed policy iteration algorithm and the Gauss-Newton policy gradient method are equivalent under suitable choice of step size, we show that they are not equivalent in the N-player setting. We illustrate in numerical experiments that the convergence rate of the proposed policy iteration algorithm significantly surpasses that of the Gauss-Newton policy gradient method and other policy gradient variations. Furthermore, our numerical results indicate that, compared to policy gradient methods, the convergence performance of the proposed policy iteration algorithm is less sensitive to the initial policy and changes in the number of players.

The rest of the paper is organized as follows. In Section II, we present the formulation of the infinite-horizon N-player general-sum deterministic LQDGs. Section III provides a detailed description of the proposed policy iteration algorithm, which is designed to solve these games. The distinction

Y. Guan and T. Summers are with the Control, Optimization, and Networks Lab, University of Texas at Dallas. M. Kamgarpour and G. Salizzoni are with the Systems Control and Multi-Agent Optimization Research Lab, EPFL. This work was supported by the United States Air Force Office of Scientific Research under Grant FA9550-23-1-0424 and the National Science Foundation under Grant ECCS-2047040.

between the proposed policy iteration algorithm and the Gauss-Newton policy gradient method within the N-player game setting is elucidated in Section IV. Section V showcases the results of our numerical experiments. Finally, we conclude our findings and discussions in Section VI.

II. Problem Formulation: N-player General-Sum Deterministic LQDGs with Infinite-Horizon

We consider a discrete-time N-player general-sum deterministic LQDG over an infinite-horizon with dynamics

$$x_{t+1} = Ax_t + \sum_{i=1}^{N} B^i u_t^i, \tag{1}$$

where $x_t \in \mathbb{R}^n$ is the system state with the initial value x_0 drawn from a Gaussian distribution with $\mathbb{E}[x_0] = 0$ and $\mathbb{E}[x_0 x_0^\top] = X_0$, $u_t^i \in \mathbb{R}^{m_i}$ is the control input of player $i = 1, \ldots, N$, and $A \in \mathbb{R}^{n\times n}$ and $B^i \in \mathbb{R}^{n\times m_i}$ are referred to as system matrices. Each player's objective is to minimize their infinite-horizon cost function

$$\min_{\{u_t^i\}_{t=0}^{\infty}} \mathbf{E}_{x_0}\left[\sum_{t=0}^{\infty} x_t^\top Q^i x_t + (u_t^i)^\top R^i u_t^i\right], \tag{2}$$

where $Q^i \in \mathbb{R}^{n\times n}$ and $R^i \in \mathbb{R}^{m_i\times m_i}$ are symmetric matrices that parameterize the quadratic stage costs.

We consider a memoryless perfect state information structure for all the players. That is, each player i seeks a stationary linear feedback policy of the form $u_t^i = K^i x_t$ that minimizes their cost. The policies of all players can be specified by a set of gain matrices $\mathbf{K} = (K^1, \ldots, K^N)$. Player i's cost induced by the joint policy $\mathbf{K}$ is given by

$$J^i(\mathbf{K}) \coloneqq \mathbf{E}_{x_0}\left[\sum_{t=0}^{\infty} \left(x_t^\top Q^i x_t + (K^i x_t)^\top R^i (K^i x_t)\right)\right]. \tag{3}$$

A standard solution concept for general-sum games is a Nash equilibrium. At a Nash equilibrium, no player can unilaterally improve their cost by deviating from their equilibrium policy, defined as follows:

Definition 1. *A stationary linear feedback Nash equilibrium for an infinite-horizon general-sum deterministic LQDG is a collection of policies* $\mathbf{K}^* = (K^{1*}, \ldots, K^{N*})$ *such that:*

$$J^i(K^{1*}, \ldots, K^{i*}, \ldots, K^{N*}) \leq J^i(K^{1*}, \ldots, K^{i}, \ldots, K^{N*}),$$

for each player $i = 1, \ldots, N$.

Our goal is to study and compare various algorithms for computing Nash equilibria for general-sum LQDGs.

III. Policy Iteration for N-player General-Sum Deterministic LQDGs with Infinite-Horizon

In this section, we first present the well known value iteration algorithm for computing a Nash equilibrium of the infinite-horizon general-sum LQDGs in Section III-A. Then we propose a policy iteration algorithm as an alternative to the value iteration algorithm for solving the general-sum LQDGs in Section III-B.

A. Value Iteration to Compute Nash Equilibrium Policies

Value iteration is a standard algorithm for computing feedback Nash equilibrium policies and cost functions in dynamic games. It utilizes principles from dynamic programming for optimal control and is discussed extensively in [5]. Initializing the cost function parameters for each player with $P_0^i = Q^i$, the value iteration algorithm updates the cost parameters and corresponding policies for each player $i = 1, ..., N$ at iteration $k = 0, 1, ...$ via

$$K_k^i = -\left(R^i + \left(B^i\right)^\top P_k^i B^i\right)^{-1} \left(B^i\right)^\top P_k^i \overline{A}_k^i, \tag{4}$$

$$\begin{aligned} P_{k+1}^i = Q^i + \left(K_k^i\right)^\top R^i K_k^i + \left(\overline{A}_k^i + B^i K_k^i\right)^\top \\ P_k^i \left(\overline{A}_k^i + B^i K_k^i\right), \end{aligned} \tag{5}$$

where $\overline{A}_k^i = A + \sum_{j=1, j\neq i}^{N} B^j K_k^j$. If these iterations converge, then the limiting policies $K^{i*} = \lim_{k\to\infty} K_k^i$ and cost parameters $P^{i*} = \lim_{k\to\infty} P_k^i$ satisfy

$$K^{i*} = -\left(R^i + (B^i)^\top P^{i*} B^i\right)^{-1} (B^i)^\top P^{i*} \overline{A}^{i*}, \tag{6}$$

$$\begin{aligned} P^{i*} = Q^i + \left(K^{i*}\right)^\top R^i K^{i*} + \left(\overline{A}^{i*} + B^i K^{i*}\right)^\top \\ P^{i*} \left(\overline{A}^{i*} + B^i K^{i*}\right), \end{aligned} \tag{7}$$

a set of coupled algebraic Riccati equations for each player $i = 1, ..., N$. Then K^{i*} and P^{i*} are Nash equilibrium policies and cost function parameters for the infinite-horizon general-sum LQDG. The following result provides a sufficient condition for convergence to such a Nash equilibrium.

Proposition 1 (Proposition 6.3 from [5])**.** *Suppose the above value iteration* (4) *and* (5) *converges to* $\{K^{i*}, P^{i*},\ i \in N\}$ *which satisfy* (6) *and* (7)*, and further suppose that for each* $i \in N$ *the pair* $\left(A + \sum_{j=1, j\neq i}^{N} B^j K^{j*},\ B^i\right)$ *is stabilizable and the pair* $\left(A + \sum_{j=1, j\neq i}^{N} B^j K^{j*},\ Q^i + (K^{i*})^\top R^i K^{i*}\right)$ *is detectable. Then stationary feedback policies* $u_t^{i*} = K^{i*} x_t$ *provide a Nash equilibrium solution for the infinite-horizon general-sum LQDGs, leading to the finite infinite-horizon Nash equilibrium cost* $x_0^\top P^{i*} x_0$ *for player* i.

Under an additional assumption that stage cost parameters satisfy $Q^i \succeq 0$ and $R^i \succ 0$ for each player, the value iteration expressions on the right side of (4) and (5) are unique (and correspond to linear feedback Nash equilibrium policies for finite-horizon LQDGs for fixed values of the value iteration index k). If the value iteration algorithm converges, the corresponding unique limiting policies $K^{i*} = \lim_{k\to\infty} K_k^i$ and cost parameters $P^{i*} = \lim_{k\to\infty} P_k^i$ provide a Nash equilibrium solution for the infinite-horizon general-sum LQDGs. However, the coupled algebraic Riccati equations (7) may admit other solutions that are not related to solutions of corresponding finite-horizon dynamic games from the value iteration algorithm [5], [20].

Policy iteration is another dynamic programming-based algorithm that can be utilized to solve general-sum games

$$\begin{bmatrix} R^1+\left(B^1\right)^\top P_k^1 B^1 & \left(B^1\right)^\top P_k^1 B^2 & \left(B^1\right)^\top P_k^1 B^3 & \cdots & \left(B^1\right)^\top P_k^1 B^N \\ \left(B^2\right)^\top P_k^2 B^1 & R^2+\left(B^2\right)^\top P_k^2 B^2 & \left(B^2\right)^\top P_k^2 B^3 & \cdots & \left(B^2\right)^\top P_k^2 B^N \\ \left(B^3\right)^\top P_k^3 B^1 & \left(B^3\right)^\top P_k^3 B^2 & R^3+\left(B^3\right)^\top P_k^3 B^3 & \cdots & \left(B^3\right)^\top P_k^3 B^N \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ \left(B^N\right)^\top P_k^N B^1 & \left(B^N\right)^\top P_k^N B^2 & \left(B^N\right)^\top P_k^N B^3 & \cdots & R^N+\left(B^N\right)^\top P_k^N B^N \end{bmatrix} \begin{bmatrix} K_{k+1}^1 \\ K_{k+1}^2 \\ K_{k+1}^3 \\ \vdots \\ K_{k+1}^N \end{bmatrix} = \begin{bmatrix} \left(B^1\right)^\top P_k^1 \\ \left(B^2\right)^\top P_k^2 \\ \left(B^3\right)^\top P_k^3 \\ \vdots \\ \left(B^N\right)^\top P_k^N \end{bmatrix} A. \tag{10}$$

by iteratively computing a solution to the coupled algebraic Riccati equations (7). We are interested in comparing the convergence properties of policy iteration with variations of policy gradient algorithms for solving general-sum LQDGs. In particular, we aim to study whether policy iteration and policy gradient algorithms converge to the same Nash equilibrium as value iteration or perhaps other Nash equilibria (if they exist), and their respective convergence rates.

B. Proposed Policy Iteration Algorithm

Algorithm 1 presents the proposed policy iteration algorithm for the infinite-horizon general-sum LQDGs. Policy iteration has been extensively studied for computing optimal policies for (single-player) optimal control problems, including linear quadratic problems [21], [22]. Policy iteration begins with an initial stabilizing policy and iterates on two main steps, which are analogous to single-player setting: (1) policy evaluation, which computes the expected costs under the current policy; and (2) policy update, which updates policy under the current cost functions. The policy evaluation step corresponds to solving a set of coupled Lyapunov equations (8), which relates to the Riccati equation in (7) but using a fixed policy. The policy update step corresponds to (9) computing a greedy one-step Nash equilibrium policy with respect to fixed value functions, which relates to the gain expression (6) but using fixed cost matrices.

To the best of our knowledge, the proposed policy iteration algorithm has not been studied and extensively compared with policy gradient methods for general-sum LQDGs. In the single-player case, the standard policy iteration algorithm coincides with the Gauss-Newton policy gradient method under suitable choice of step size [23], [24]. However, there is a key difference in how the policies are updated in the N-player game setting, which we will explain in detail in the next section.

IV. A Comparison of the Policy Iteration and Gauss-Newton Policy Gradient Algorithms

In this section, we first introduce the N-player vanilla/standard policy gradient and natural policy gradient methods, with a brief analysis of the results from [12] and [13] in Section IV-A. Then we extend the Gauss-Newton policy gradient method from the single-player case to the N-player game setting. We show that, unlike the single-player case, the proposed policy iteration algorithm is distinct from the Gauss-Newton policy gradient method in the N-player game setting in Section IV-B.

Algorithm 1 Policy iteration for N-player general-sum LQDGs with infinite-horizon

Input: Stabilizing policies K_0^i, system matrices A and B^i, cost parameters Q^i and R^i, and convergence threshold $\epsilon > 0$.

1: **Initialize:** $K_0^i = (0, \ldots, 0)$, $K_1^i = (\infty, \ldots, \infty)$, $i = 1, \ldots, N$, $k = 0$.

2: **while** $\sum_{i=1}^N \left\| K_{k+1}^i - K_k^i \right\| > \epsilon$ **do**

3: **Policy Evaluation:** Compute the value functions of the current policy set for $i = 1, \ldots, N$ by solving the Lyapunov equations

$$P_k^i = Q^i + \left(K_k^i\right)^\top R^i K_k^i + \left(A + \sum_{j=1}^N B^j K_k^j\right)^\top P_k^i \left(A + \sum_{j=1}^N B^j K_k^j\right). \tag{8}$$

4: **Policy Update:** Update the policy set for $i = 1, \ldots, N$ by solving the coupled equations

$$K_{k+1}^i = -\left(R^i + \left(B^i\right)^\top P_k^i B^i\right)^{-1} \left(B^i\right)^\top P_k^i \left(A + \sum_{j=1, j\neq i}^N B^j K_{k+1}^j\right), \tag{9}$$

which can be jointly computed by solving (10).

5: $k \leftarrow k + 1$

Output: Nash Equilibria K^{i*} $(i = 1, \ldots, N)$

A. The Vanilla/Standard Policy Gradient and Natural Policy Gradient Methods

An extension of the vanilla/standard policy gradient method in the single-player case [25] for the N-player game setting has the following form

$$K_{k+1}^i = K_k^i - \eta^i \nabla_{K_k^i} J^i\left(\mathbf{K}_k\right), \tag{11}$$

where $\mathbf{K}_k = (K_k^1, \ldots, K_k^N)$ is a collection of gain matrices of all players with initial gains $\mathbf{K}_0$ and η^i is the step size. The work in [12] suggests that the vanilla/standard policy gradient method has no guarantees of even local convergence in general-sum infinite-horizon deterministic LQDGs. In contrast, [13] proved the global convergence of the natural policy gradient method to the Nash equilibrium with finite-horizon and stochastic dynamics. The natural policy gradient method presented in [13] is

$$K_{t,k+1}^i = K_{t,k}^i - \eta \nabla_{K_t^i} J^i\left(\mathbf{K}_k\right) \left(\Sigma_{t,\mathbf{K}_k}\right)^{-1}, \tag{12}$$

which is an N-player extension of the natural policy gradient method introduced in [25] and $\Sigma_{t,\mathbf{K}} = \mathbf{E}[x_{t,\mathbf{K}} x_{t,\mathbf{K}}^\top]$ is the state covariance matrix. However, the natural policy gradient method may fail to converge to a Nash equilibrium for deterministic LQDGs without careful selection of the initial policies and step sizes. To our best knowledge, the general convergence properties of policy gradient algorithms for general-sum LQDGs are not fully understood.

B. The Gauss-Newton Policy Gradient Method

Our goal in this section is to compare the Gauss-Newton policy gradient method and our algorithm, to highlight and explain the main difference between the two. For the single-player with stationary feedback policy case, the Gauss-Newton policy gradient method is

$$K_{k+1} = K_k - \eta \left(R + B^\top P_k B\right)^{-1} \nabla J(K_k) \Sigma_k^{-1}, \quad (13)$$

where the gradient $\nabla J(K_k)$ is equal to

$$\nabla J(K_k) = 2 \left(\left(R + B^\top P_k B\right) K_k + B^\top P_k A\right) \Sigma_k.$$

By substituting this in (13) we obtain

$$K_{k+1} = (1 - 2\eta) K_k - 2\eta \left(R + B^\top P_k B\right)^{-1} B^\top P_k A. \quad (14)$$

The natural extension of the Gauss-Newton policy gradient method to the N-player game setting would be as

$$K_{k+1}^i = K_k^i - \eta^i \left(R^i + (B^i)^\top P_k^i B^i\right)^{-1} \nabla_{K_k^i} J^i(\mathbf{K}_k) \Sigma_k^{-1}.$$

Following the same steps, we obtain

$$K_{k+1}^i = (1 - 2\eta^i) K_k^i - 2\eta^i \left(R^i + \left(B^i\right)^\top P_k^i B^i\right)^{-1} \left(B^i\right)^\top P_k^i \left(A + \sum_{j=1, j\neq i}^{N} B^j K_k^j\right). \quad (15)$$

In the single-player case, by setting $\eta = \frac{1}{2}$ in (14), one can easily verify that the Gauss-Newton policy gradient method is equivalent to the standard policy iteration algorithm [16]:

$$K_{k+1} = -\left(R + B^\top P_k B\right)^{-1} B^\top P_k A. \quad (16)$$

For the N-player case, however, this is no longer true. By setting $\eta^i = \frac{1}{2}$ in (15) we have

$$K_{k+1}^i = -\left(R^i + \left(B^i\right)^\top P_k^i B^i\right)^{-1} \left(B^i\right)^\top P_k^i \left(A + \sum_{j=1, j\neq i}^{N} B^j K_k^j\right), \quad (17)$$

which is different from the policy update (9) of the proposed policy iteration algorithm. In particular, in the Gauss-Newton method, player i's policy update at iteration $k+1$ is defined on the premise that all other players' policy gains K_k^j, $j = 1, \ldots, N, j \neq i$ remain fixed at the previous iteration step k. In the proposed policy iteration algorithm, however, all players update their policy gains K_{k+1}^i, $i = 1, \ldots, N$ simultaneously at iteration $k+1$. As a result, the proposed policy iteration algorithm needs to solve a linear system with $\sum_{i=1}^N nm_i$ equations and $\sum_{i=1}^N nm_i$ unknowns as shown in (10) and there has to be a central solver (alternatively, each player can compute their own update independently when the model is assumed to be common knowledge). In the Gauss-Newton policy gradient method, however, each player can compute its own policy gain update.

V. Numerical Experiments

In this section, we first compare the convergence performance of the proposed policy iteration algorithm, natural policy gradient (Algorithm 1 in [13] extended to the infinite-horizon deterministic case), and Gauss-Newton policy gradient methods under the same experimental setup as in [12], [13] in V-A and V-B. Then we compare these algorithms with an additional 1000 random open-loop stable systems with initial policy gain $K_0^i = (0, \ldots, 0)$ that satisfies the conditions in Proposition 1 in V-C.

The model parameters for the numerical results in V-A and V-B are

$$A = \begin{bmatrix} 0.588 & 0.028 \\ 0.570 & 0.056 \end{bmatrix}, \; B^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; B^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$Q^1 = \begin{bmatrix} 0.01 & 0 \\ 0 & 1 \end{bmatrix}, \; Q^2 = \begin{bmatrix} 1 & 0 \\ 0 & 0.147 \end{bmatrix}, \; R^1 = R^2 = 0.01.$$

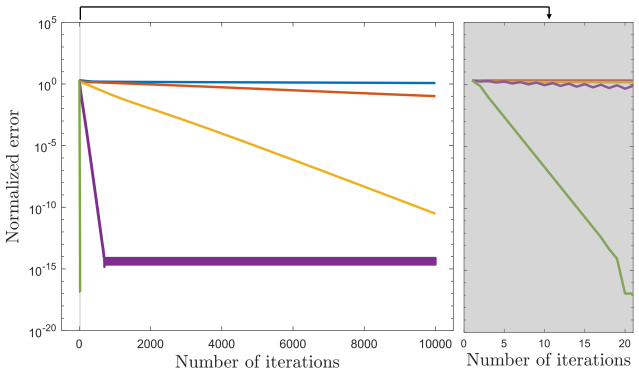
We initialize both players' policy K_0^i such that $(A + \sum_{i=1}^N B^i K_0^i)$ is stable. To analyze the convergence performance of all three algorithms under different initial policies, the initial policy gain K_0^i also satisfies $\|K_0^i - K^{i*}\|_2 \leq r$, where K^{i*} denotes a Nash equilibrium of the system, and r is the radius of the ball centered at K^{i*} in which we initialize the policies. A Nash equilibrium of the above system is $K^{1*} = (-0.5134, -0.0439)$ and $K^{2*} = (-0.0525, -0.0114)$. This Nash equilibrium is computed through value iteration introduced in III-A. Differently from the normalized error definition in [13], we define the normalized error of a given pair of policies (K^1, K^2): for $i = 1, 2$ as

$$e_{norm,k} = \frac{\|K_k^1 - K^{1*}\|_2}{\|K^{1*}\|_2} + \frac{\|K_k^2 - K^{2*}\|_2}{\|K^{2*}\|_2},$$

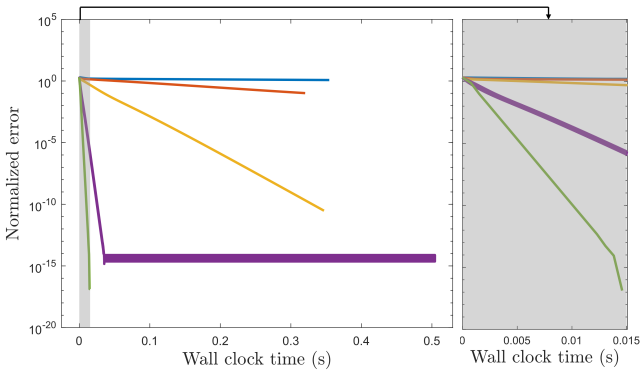
where k is the number of iterations.

A. Faster Convergence Rate of Policy Iteration

Figure 1 reveals that all three methods—the proposed policy iteration algorithm, natural policy gradient ($\eta^i = 0.1$), and Gauss-Newton policy gradient ($\eta^i = 0.5$)—successfully converge to K^{1*} and K^{2*} with the same initial policy gains $K_0^1 = (-0.4266, -0.0938)$ and $K_0^2 = (0.0342, -0.0612)$ ($r = 0.1$). It is evident that the proposed policy iteration algorithm converges to the Nash equilibrium with much fewer iterations and shorter computational time compared with the other two policy gradient methods in this specific example. The natural policy gradient method fails to converge to the Nash equilibrium if the step size is not carefully selected. Although the Gauss-Newton policy gradient method converges to the Nash equilibrium, it requires significantly more iterations and computational time than the proposed



(a) Normalized error vs. number of iterations.



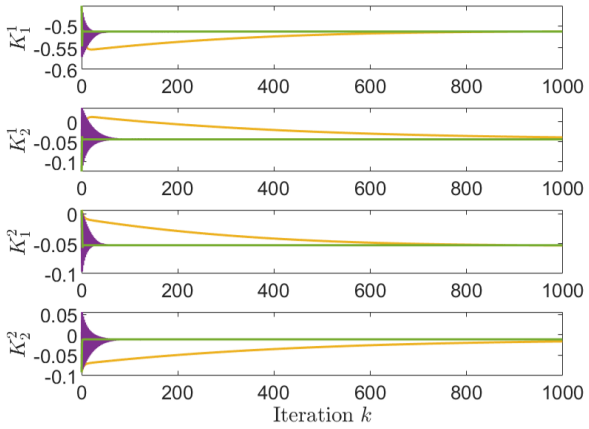
(b) Normalized error vs. elapsed wall clock time.

Fig. 1: Convergence speed of the proposed policy iteration algorithm (green), Gauss-Newton policy gradient (purple, $\eta^i = 0.5$), and natural policy gradient (blue, $\eta^i = 10^{-3}$; red, $\eta^i = 10^{-2}$; yellow, $\eta^i = 10^{-1}$) ($r = 0.1$).

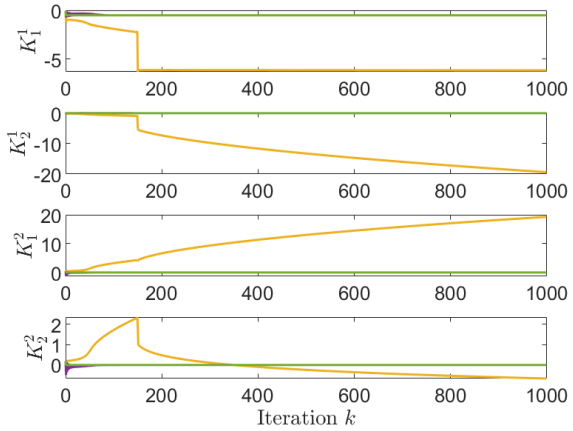
policy iteration algorithm. However, comparing the general theoretical convergence rates of the proposed policy iteration algorithm and policy gradient methods remains open.

B. Convergence Performance of Policy Iteration from a Distant Initial Policy

Figure 2 illustrates the convergence performance of the proposed policy iteration algorithm, natural policy gradient, and Gauss-Newton policy gradient methods as the initial policy gains transition from a smaller to a larger neighborhood around K^{1*} and K^{2*}. Figure 2a presents an instance where all three methods with the same initial policy gains K_0^1 and K_0^2 as in V-A converge to K^{1*} and K^{2*}. Conversely, Figure 2b shows a different scenario where, under a pair of more distant initial policy gains $K_0^1 = (-0.0543, 0.1541)$ and $K_0^2 = (0.4066, 0.1867)$ ($r = 0.5$) from K^{1*} and K^{2*}, the natural policy gradient method does not converge to K^{1*} and K^{2*} as the other two methods. It is evident that the convergence performance of the proposed policy iteration algorithm is less sensitive to changes in initial policy compared to policy gradient methods. It is possible that the convergence performance of the policy gradient methods can be improved by tuning the step size. However, this process for each case can be laborious and time-consuming.



(a) An instance ($r = 0.1$) that all three algorithms converge to K^{1*} and K^{2*}.



(b) An instance ($r = 0.5$) that the natural policy gradient algorithm fails to converge to K^{1*} and K^{2*}.

Fig. 2: Convergence performance of the proposed policy iteration algorithm (green), Gauss-Newton policy gradient (purple, $\eta^i = 0.5$), and natural policy gradient (yellow, $\eta^i = 10^{-1}$) methods under different initial policy gains.

C. Convergence Performance of Policy Iteration for Additional Problem Instances

We now present results for additional problem instances by randomly generating system parameters ($A, B^i, Q^i,$ and R^i). The entries of these parameters were independently drawn from a standard normal distribution. The dynamics matrix is scaled to be open-loop stable, and all cost parameters are made positive definite. In each instance, value iteration was used to compute a Nash equilibrium (if it converges) for that game setting. Then the policy iteration and policy gradient methods with the same (stabilizing) initial policy gain $K_0^i = (0, \ldots, 0)$ were used to compute a solution for the same game setting. We examine whether and how fast the methods converge to the Nash equilibrium computed using value iteration.

Table I shows the convergence performance of the proposed policy iteration algorithm, natural policy gradient, and Gauss-Newton policy gradient methods for 1000 randomly-generated

problem instances for both two and four players. In all experiments, the convergence of the proposed policy iteration algorithm is significantly faster and more reliable than the policy gradient methods. Moreover, the convergence performance of the proposed policy iteration algorithm is less sensitive to a change in the number of players from two to four. These results provide additional evidence that the proposed policy iteration algorithm outperforms policy gradient methods, especially since there is no need to tune the step size for each instance.

TABLE I: Convergence performance of the policy iteration (PI) algorithm, natural policy gradient (NPG, $\eta^i = 0.1$), and gauss-Newton policy gradient (GNPG, $\eta^i = 0.5$) methods for 1000 random systems.

	$n = 4, m^i = 2, N = 2$		$n = 4, m^i = 2, N = 4$	
	convergent cases	average number of iterations	convergent cases	average number of iterations
NPG	5	139	0	N/A
GNPG	880	161	8	1259
PI	960	7	945	7

In all our empirical studies, the proposed policy iteration algorithm converges at a much higher speed than the policy gradient methods. Similar numerical results have also been shown in [18]. This may be due to the fact that the proposed policy iteration algorithm takes into account the policies of the other players at iteration $k + 1$. The update in the policy gradient methods considers only the policy of the other players at iteration k. Thanks to this additional information, the policy iteration algorithm adjusts the update of the policy and therefore, we believe, avoids possible overshooting.

VI. Conclusions

We proposed a policy iteration algorithm for the infinite-horizon N-player general-sum deterministic LQDGs and compare it to policy gradient methods. We demonstrated that the proposed policy iteration algorithm is distinct from the Gauss-Newton policy gradient method in the N-player game setting, in contrast to the single-player case where they are equivalent under suitable choice of step size. In all our numerical experiments, the proposed policy iteration algorithm converges to the same Nash equilibrium as value iteration with fewer iterations and less computational time compared to policy gradient methods. Furthermore, the convergence performance of the proposed policy iteration algorithm is less sensitive to the initial policy and changes in the number of players.

References

[1] L. Matignon, L. Jeanpierre, and A.-I. Mouaddib, "Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, 2012, pp. 2017–2023.

[2] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[3] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[4] J. C. Engwerda, "On scalar feedback nash equilibria in the infinite horizon lq-game," *IFAC Proceedings Volumes*, vol. 31, no. 16, pp. 193–198, 1998.

[5] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory, 2nd Edition*. Society for Industrial and Applied Mathematics, 1998.

[6] C. Possieri and M. Sassano, "An algebraic geometry approach for the computation of all linear feedback Nash equilibria in LQ differential games," in *IEEE Conference on Decision and Control*, Dec. 2015, pp. 5197–5202.

[7] T. Basar, "On the uniqueness of the Nash solution in Linear-Quadratic differential Games," *International Journal of Game Theory*, vol. 5, no. 2-3, pp. 65–90, June 1976.

[8] D. L. Lukes and D. L. Russell, "A global theory for linear-quadratic differential games," *Journal of Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 96–123, 1971.

[9] J. Bu, L. J. Ratliff, and M. Mesbahi, "Global convergence of policy gradient for sequential zero-sum linear quadratic dynamic games," *arXiv preprint arXiv:1911.04672*, 2019.

[10] K. Zhang, Z. Yang, and T. Basar, "Policy optimization provably converges to Nash equilibria in zero-sum linear quadratic games," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[11] K. Zhang, X. Zhang, B. Hu, and T. Basar, "Derivative-free policy optimization for linear risk-sensitive and robust control design: Implicit regularization and sample complexity," *Advances in neural information processing systems*, vol. 34, pp. 2949–2964, 2021.

[12] E. Mazumdar, L. J. Ratliff, M. I. Jordan, and S. S. Sastry, "Policy-gradient algorithms have no guarantees of convergence in linear quadratic games," in *AAMAS Conference proceedings*, 2020.

[13] B. Hambly, R. Xu, and H. Yang, "Policy gradient methods find the nash equilibrium in n-player general-sum linear-quadratic games," *Journal of Machine Learning Research*, vol. 24, no. 139, pp. 1–56, 2023.

[14] M. L. Puterman and S. L. Brumelle, "On the convergence of policy iteration in stationary dynamic programming," *Mathematics of Operations Research*, vol. 4, no. 1, pp. 60–69, 1979.

[15] D. Bertsekas, *Lessons from AlphaZero for optimal, model predictive, and adaptive control*. Athena Scientific, 2022.

[16] R. A. Howard, *Dynamic Programming and Markov Processes*. Cambridge, MA: MIT Press, 1960.

[17] K. G. Vamvoudakis, H. Modares, B. Kiumarsi, and F. L. Lewis, "Game theory-based control system algorithms with real-time reinforcement learning: How to solve multiplayer games online," *IEEE Control Systems Magazine*, vol. 37, no. 1, pp. 33–52, 2017.

[18] B. Nortmann, A. Monti, M. Sassano, and T. Mylvaganam, "Nash Equilibria for Linear Quadratic Discrete-time Dynamic Games via Iterative and Data-driven Algorithms," *IEEE Transactions on Automatic Control*, pp. 1–15, 2024.

[19] Y. Yang, S. Zhang, J. Dong, and Y. Yin, "Data-Driven Nonzero-Sum Game for Discrete-Time Systems Using Off-Policy Reinforcement Learning," *IEEE Access*, vol. 8, pp. 14 074–14 088, 2020.

[20] G. P. Papavassilopoulos and G. J. Olsder, "On the linear-quadratic, closed-loop, no-memory Nash game," *Journal of Optimization Theory and Applications*, vol. 42, no. 4, pp. 551–560, Apr. 1984.

[21] D. Kleinman, "On the stability of linear stochastic systems," *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 429–430, 1969.

[22] G. Hewer, "An iterative technique for the computation of the steady state gains for the discrete optimal regulator," *IEEE Transactions on Automatic Control*, vol. 16, no. 4, pp. 382–384, 1971.

[23] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *Proceedings of the 2005, American Control Conference, 2005.* IEEE, 2005, pp. 300–306.

[24] L.-Z. Liao and C. A. Shoemaker, "Convergence in unconstrained discrete-time differential dynamic programming," *IEEE Transactions on Automatic Control*, vol. 36, no. 6, pp. 692–706, 1991.

[25] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *International conference on machine learning*. PMLR, 2018, pp. 1467–1476.