Multi-agent discrete-time graphical games and reinforcement learning solutions
Abouheaf, M.; Lewis, F.L.; Vamvoudakis, K.G.; Haesaert, S.; Babuska, R.

Published in: Automatica

DOI: 10.1016/j.automatica.2014.10.047

Published: 01/01/2014

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain.
• You may freely distribute the URL identifying the publication in the public portal.

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 07. Dec. 2018
Multi-agent discrete-time graphical games and reinforcement learning solutions

Mohammed I. Abouheaf\textsuperscript{a,1}, Frank L. Lewis\textsuperscript{b}, Kyriakos G. Vamvoudakis\textsuperscript{c}, Sofie Haesaert\textsuperscript{d}, Robert Babuska\textsuperscript{e}

\textsuperscript{a} Systems Engineering Department, King Fahd University of Petroleum & Minerals, P. O. Box: 5067, Dhahran-3126, Saudi Arabia
\textsuperscript{b} University of Texas at Arlington Research Institute, The University of Texas at Arlington, USA
\textsuperscript{c} Center for Control, Dynamical-systems, and Computation (CCDC), University of California, Santa Barbara, USA
\textsuperscript{d} Eindhoven University of Technology, Netherlands
\textsuperscript{e} Delft Center for Systems and Control, Delft University of Technology, Netherlands

\begin{abstract}
This paper introduces a new class of multi-agent discrete-time dynamic games, known in the literature as dynamic graphical games. For that reason a local performance index is defined for each agent that depends only on the local information available to each agent. Nash equilibrium policies and best-response policies are given in terms of the solutions to the discrete-time coupled Hamilton-Jacobi equations. Since in these games the interactions between the agents are prescribed by a communication graph structure we have to introduce a new notion of Nash equilibrium. It is proved that this notion holds if all agents are in Nash equilibrium and the graph is strongly connected. A novel reinforcement learning value iteration algorithm is given to solve the dynamic graphical games in an online manner along with its proof of convergence. The policies of the agents form a Nash equilibrium when all the agents in the neighborhood update their policies, and a best response outcome when the agents in the neighborhood are kept constant. The paper brings together discrete Hamiltonian mechanics, distributed multi-agent control, optimal control theory, and game theory to formulate and solve these multi-agent dynamic graphical games. A simulation example shows the effectiveness of the proposed approach in a leader–synchronization case along with optimality guarantees.

© 2014 Elsevier Ltd. All rights reserved.
\end{abstract}

\section{1. Introduction}

Research on distributed multi-agent cooperative control systems has received extensive attention in the last two decades, mainly due to their applications in computer science, spacecraft, unmanned air vehicles, mobile robots, sensor networks, networked autonomous team, and so on (Beard & Stepanyan, 2003; Mu, Chu, & Wang, 2005). Synchronization (Jadbabaie, Lin, & Morse, 2003; Olfati-Saber, Fax, & Murray, 2007; Olfati-Saber & Murray, 2004; Qu, 2009; Ren & Beard, 2005, 2008; Ren, Beard, & Atkins, 2005; Tsitsiklis, 1984) allows each agent of a cooperative team to reach the same state by the proper selection and design of decision and control protocols.

Consensus has been studied for systems on communication graphs with fixed or varying topologies and communication delays. In the cooperative regulator consensus problem without leader, all agents converge to an uncontrollable common value depending on their initial conditions. In the cooperative tracking consensus problem, all agents synchronize to a leader or control agent state (Hong, Hu, & Gao, 2006; Ren, Moore, & Chen, 2007; Wang & Chen, 2002). In Fox and Murray (2004), the state of each agent is given by identical linear decoupled dynamics. A dynamical system is realized to supply each agent with common reference to be used in the collective behavior. In Ren et al. (2007), the authors generalize the first-order and second-order consensus problems to higher-order...
consensus problems. In Li, Duan, Chen, and Huang (2010), the synchronization among multi-agent systems with identical dynamics is achieved using a distributed observer consensus protocol based on relative output measurements. In Li, Wang, and Chen (2004), a complex dynamical network is controlled via pinning partially to a small number of nodes in the network using local feedback controllers. The placement of the local controllers is affected by the topology of the dynamical network. In the aforementioned consensus algorithm there are no optimality guarantees. For that reason one has to use a game-theoretic framework to overcome this issue.

Game theory provides an environment for formulating multi-player decision control problems for dynamical interacting systems (Başar & Olsder, 1999). Each agent optimizes its performance index independently to determine its optimal policy. The result is the Nash equilibrium solution. In order to find the Nash equilibrium one has to solve coupled Hamilton–Jacobi (HJ) equations, which in the linear quadratic case (LQR) reduce to the coupled game algebraic Riccati equations (Başar & Olsder, 1999; Freiling, Jank, & Abou-Kandil, 2002; Gajic & Li, 1988). Solution methods are generally offline and generate fixed control policies that are used to implement real-time online controllers. These coupled HJ equations are difficult or impossible to solve and depend on global information. As we shall see in the present paper, we will overcome this global information issue by designing dynamic graphical games. Static graphical games have been studied in the computational intelligence community by several researchers (e.g., Kakade, Kearns, Langford, & Ortiz, 2003; Shoham & Leyton-Brown, 2009).

In discrete mechanics, discrete versions of variational principles are used to derive the discrete equivalents of the continuous Euler–Lagrange and Hamiltonian equations (Marsden & West, 2001; Suris, 2003, 2004). The theory of discrete Lagrangian mechanics was introduced in Marsden and West (2001). A formulation for discrete Hamilton mechanics using direct approaches was developed in Gonzalez (1996) and McLachlan, Quispel, and Robidoux (1999). Later, Lall and West (2006), introduced the canonical form of the Hamiltonian theory that corresponds to the theory of discrete Lagrangian mechanics. The importance of discrete-time Hamiltonian relies on the Euclidean relationship between symplectic integrators, discrete time optimal control (Lewis, Vrabie, & Srymos, 2012), and distributed network optimization (Lall & West, 2006). Moreover, it is often more direct to solve problems based on the discrete Hamiltonian rather than discrete Lagrangian (Lall & West, 2006).

Reinforcement Learning (RL) is an area of machine learning concerned with how an agent can pick its actions in a dynamic environment to transit to new states in such a way that optimizes the sum of cumulative reward (Sen & Weiss, 1999; Sutton & Barto, 1998). RL methods allow the development of algorithms to learn online the solutions to optimal control problems for dynamic systems that are described by difference equations (Sutton & Barto, 1998; Werbos, 1974, 1989, 1992). These involve two-step techniques known as Policy Iteration (PI) or Value Iteration (VI) (Bertsekas & Tsitsiklis, 1996). Policy iteration and value iteration algorithms have been developed for continuous time systems in Al-Tamimi, Lewis, and Abu-Khalaf (2008), Vamvoudakis and Lewis (2010), and Vrabie, Pasravanu, Lewis, and Abu-Khalaf (2009). RL algorithms are used to solve multi-player games for finite-state systems in Busoniu, Babuska, and De Schutter (2008), Littman (2001), and Vanx, Verbeek, and Nowe (2008), and to learn online in real-time the solutions for optimal control problems of dynamic systems and differential games in Dierks and Jagannathan (2010), Johnson, Hiramatsu, Fitz-Coy, and Dixon (2010), and Vamvoudakis and Lewis (2010, 2011). Wang, Liu, Wei, Zhao, and Jin (2012) used mathematical induction to prove convergence of an iterative dynamic programming algorithm in order to solve the optimal control problem for unknown non-affine nonlinear discrete-time systems. Actor–critic networks are one type of RL methods. The actor component applies actions or control policies to their environment, while the critic component assesses the values of these actions. Based on this assessment, the actor policy is updated at each learning step (Bertsekas & Tsitsiklis, 1996; Sutton & Barto, 1998).

Game theory is extensively used in multi-agent reinforcement learning. In such scenarios the agents do not have perfect knowledge about the game (e.g. Chang, Hsu, & Fu, 2007; Gopalanrishnan, Marden, & Wierman, 2011; Marden, Arslan, & Shamma, 2009; Young, 1998). In Bowling (2004), different methods are used to learn the optimal policy of every agent through repeated interactions among all agents. Interactions among a large number of players or complicated stochastic large systems are studied by population games and mean field games (Hofbauer & Sigmund, 1998; Sandholm, 2010). In Nourian, Caines, and Malhame (2011), a continuum based mean field control approach is used to solve the initial mean consensus problem. In that game, a set of coupled deterministic Hamilton–Jacobi Bellman and Fokker Planck Kolmogorov equations are used to approximate the stochastic system of agents in the continuum. The dynamics of population and mean field games covers a large class of game dynamics known in evolutionary game theory (Tembine, 2011).

This paper extends the results of continuous time graphical games where introduced in Vamvoudakis, Lewis, and Hudas (2012) to discrete-time systems. It is shown that these dynamic graphical games are a special case of standard dynamic games (Başar & Olsder, 1999) and explicitly capture the structure of the communication graph topology. As such, they allow an analysis that shows the restrictions imposed on local control protocols by the graph topology.

The contributions of the paper are fourfold. The first involves the formulation of a graphical game for dynamic discrete-time multi-agent systems where information flow is restricted by a communication graph structure. A new notion of interactive Nash equilibrium is introduced which holds if the agents are all in Nash and the graph is strongly connected. The second contribution is to use the relation between the discrete-time Bellman equation and the discrete-time Hamilton equation using discrete mechanics for dynamic graphical games. Coupled discrete-time HJ equations are formulated for the dynamic graphical games. The third contribution lies in providing solutions in terms of those coupled HJ equations that converge to Nash equilibrium and best-response. Finally, a value iteration Heuristic Dynamic Programming (HDP) algorithm is given to solve the dynamic graphical games online in real-time by measuring the states along the system trajectories.

The paper is organized as follows. Section 2 provides a brief background on the synchronization problem in multi-agent systems. Section 3 formulates the dynamic graphical game and finds the relation between the discrete-time Hamilton–Jacobi Bellman equation and discrete-time Bellman optimality equation for multi-agent graphical games. Section 4 introduces the new notion of interactive Nash equilibrium. Moreover, this section provides existence solutions for policies forming an interactive Nash equilibrium and best response in terms of the solutions of the coupled HJ equations. Section 5 proposes a value iteration HDP algorithm to solve the coupled HJ equations along with its proof of convergence. Finally, Section 6 develops an online adaptive learning algorithm by using an actor–critic neural network framework to solve the graphical game, along with a simulation example to verify its effectiveness.

2. Graphs and synchronization of multi-agent dynamical systems

2.1. Graphs

The directed graph $G_r$ is defined as the pair $G_r = (V, E)$ with a nonempty finite set of $N$ vertices $V = \{v_1, \ldots, v_N\}$ and a set
of edges $\xi \subseteq V \times V$. The connectivity matrix $E$ is defined such that $E = [e_{ij}]$ with $e_{ij} > 0$ if $(v_i, v_j) \in \xi$ and $e_{ij} = 0$ otherwise. The set of neighbors of every node $v_i$ is $N_i = \{ v_j : (v_i, v_j) \in \xi \}$. Define the in-degree matrix $D$ as a diagonal matrix $D = \text{diag}[d_i]$, with $d_i = \sum_{j \in N_i} e_{ij}$ the weighted in-degree of node $i$. The graph Laplacian matrix $L$ is defined as $L = D - E$.

A directed path from node $v_0$ to node $v_i$ is defined as a sequence of edges $v_0, v_1, \ldots, v_i$ such that $(v_j, v_{j+1}) \in \xi$, $i \in \{0, 1, \ldots, r - 1\}$. A directed graph is strongly connected if there is a directed path from $v_i$ to $v_j$ and vice versa for all distinct nodes $v_i, v_j \in V$.

### 2.2. Synchronization and tracking error dynamics

Consider the communication graph $G_r = (V, \xi)$ having $N$ agents, each with local dynamics given by

$$x_i(k+1) = Ax_i(k) + Bu_i(k)$$  \hspace{1cm} (1)

where $x_i(k) \in \mathbb{R}^n$ is the state vector of node $i$, and $u_i(k) \in \mathbb{R}^m$ is the control input vector for node $i$. Also consider a control or leader node $v_0$ that has command generator dynamics (Lewis, 1992)

$$x_0(k) \in \mathbb{R}^n$$  \hspace{1cm} (2)

$$x_0(k+1) = Ax_0(k).$$

The set of neighbors of every node $v_i$ is $N_i = \{ v_j : (v_i, v_j) \in \xi \}$, $i \in \{0, 1, \ldots, r - 1\}$. A directed path from $v_0$ to $v_i$ and vice versa for all distinct nodes $v_i, v_j \in V$.

The objective is to design the control inputs $u_i(k)$, using information only from neighboring nodes, so that all agent states synchronize to the leader state, that is $\lim_{k \to \infty} \|x_i(k) - x_0(k)\| = 0$, $\forall i$.

To study the synchronization problem on graphs, we define the local neighborhood tracking error (Khoo, Xie, & Man, 2009)

$$e_i(k) \in \mathbb{R}^n$$  \hspace{1cm} (3)

$$e_i(k) = \sum_{j \in N_i} e_{ij} (x_j(k) - x_i(k)) + g_i(x_0(k) - x_i(k))$$

where $g_i \geq 0$ is the pinning gain of node $i$, which is nonzero if node $i$ is coupled to the control node $x_0$ (Li et al., 2004).

The overall tracking error vector for all nodes is given by

$$e(k) = -((L + G) \otimes L_0)x(k) + ((L + G) \otimes L_0)x(0)$$  \hspace{1cm} (4)

where the global node state vector $x = [x_1^T, x_2^T, \ldots, x_n^T]^T$ and

$$e(k) = -((L + G) \otimes L_0)\eta(k)$$

(5)

where the global disagreement vector or the synchronization error vector (Olfati-Saber & Murray, 2004)

$$\eta(k) = (x(k) - x_0(k)) \in \mathbb{R}^{nN}$$

with $x_0 = [x_0, 1] \otimes L_0$ and $1$ the $N$-vector of ones. $G = \text{diag}[g_i] \in \mathbb{R}^{n \times n}$ is a diagonal matrix of pinning gains.

If the graph contains a spanning tree and $g_i \neq 0$ for a root node, then $(L + G)$ is nonsingular (Khoo et al., 2009).

The maximum and minimum singular values of a matrix are denoted respectively as $\sigma^*$, $\sigma^\dagger$.

**Lemma 1.** Let $(L + G)$ be nonsingular. Then the synchronization error is bounded by

$$\|\eta(k)\| \leq \|e(k)\| / \sigma(L + G).$$  \hspace{1cm} (7)

**Proof.** Under the Hypothesis $(L + G)$ is nonsingular. Then $\sigma(L + G) \neq 0$ and (5) implies (7), with $e(k) = 0$ if and only if the nodes synchronize, that is

$$x(k) = L_0 x_0(k).$$  \hspace{1cm} (8)

For ease of notation, we write $x_{ik}$ for $x_i(k)$.

The dynamics of the local neighborhood tracking error for node $i$ are given by

$$e_{i(k+1)} = f_i(e_{ik}, u_{ik}, u_{\ldots ik})$$

$$= A e_{ik} - (d_i + g_i) B u_{ik} + \sum_{j \in N_i} e_{ij} B u_{jk}. \hspace{1cm} (9)$$

These error dynamics are interacting dynamical systems driven by the control actions of agent $i$ and all of its neighbors. Our objective is to minimize the local neighborhood tracking errors $e_i(k)$, which in view of Lemma 1 will guarantee approximate synchronization.

### 3. Dynamic graphical games

In this section we define multi-player dynamic games on graphs. These dynamic graphical games are defined based on the error systems (9), which are locally coupled in the sense that they are driven by the agent’s control actions and those of its neighbors. This structure arises from the nature of the synchronization problem for dynamic systems (1) on communication graphs. Therefore, in similar fashion, we define locally coupled performance indices that depend on the state of an agent, its control action, and the control actions of its neighbors.

Following that, principles of optimal control (Bellman, 1957; Bryson, 1996; Lewis et al., 2012) are used to develop the Hamiltonian functions and the Bellman equations for dynamic graphical games. The Discrete Hamilton–Jacobi theory is used to show the relation between the Hamiltonian equation and the Bellman equation (Lall & West, 2006). These results lay the foundation to solve the dynamic graphical games in subsequent sections.

#### 3.1. Graphical games

Graphical games are based on the responses of each agent $i$ to other players in graph. Define the control actions of the neighbors of agent $i$ as

$$u_{\ldots i} = \{ u_j \mid j \in N_i \}$$

and the actions of all the other agents in the graph excluding $i$ as

$$u_i = \{ u_j \mid j \in N, j \neq i \}.$$  \hspace{1cm} (10)

The structure of the error dynamics (9) arises from the nature of the synchronization problem for dynamic systems on communication graphs. Therefore, in order to define the dynamic graphical game, in similar fashion, we write the local performance indices for each agent $i$ as

$$J_i(e_{ik}, u_{ik}, u_{\ldots ik}) \equiv \sum_{k=0}^{\infty} U_i(e_{ik}, u_{ik}, u_{\ldots ik})$$

$$= \frac{1}{2} \sum_{k=0}^{\infty} \left( \epsilon_{ik}^T Q_0 \epsilon_{ik} + u_{ik}^T R_0 u_{ik} + \sum_{j \in N_i} u_{jk}^T R_0 u_{jk} \right) \hspace{1cm} (12)$$

where $Q_0 > 0 \in \mathbb{R}^{n \times n}$, $R_0 > 0 \in \mathbb{R}^{m \times m}$, $R_0 > 0 \in \mathbb{R}^{m \times m}$, are symmetric time-invariant weighting matrices.

The dynamics (9) and the performance indices (12) depend on the graph topology $G = (V, \xi)$.

Given fixed policies of agent $i$ and its neighbors, the value function for each agent $i$ is given as

$$V_i(e_{ik}) = \sum_{k=0}^{\infty} U_i(e_{ik}, u_{ik}, u_{\ldots ik}).$$  \hspace{1cm} (13)

**Definition 1.** The policies $u_{ik}$, $\forall i$ are said to be admissible if they stabilize (9) and guarantee that $V_i(e_{ik})$, $\forall i$ are finite (Zhang, Luo, & Liu, 2009).

**Definition 2.** The dynamic graphical game with local dynamics (9) and performance indices (12) is well-formed if $B_i \neq 0 = e_i \in E$, $R_0 \neq 0 = e_i \in E$.  \hspace{1cm} \blacksquare
3.2 Comparison of dynamic graphical games with standard dynamic games

Dynamic graphical games as introduced in this paper are a special case of the dynamic games normally discussed in the literature. The standard \( N \)-player dynamic game as defined in Başar and Olsder (1999), has centralized dynamics

\[
\delta(k + 1) = F\delta(k) + \sum_{i=1}^{N} B_i u_i(k)
\]

(14)

where \( \delta(k) \in \mathbb{R}^N \) is the centralized state and \( u_i(k) \in \mathbb{R}^m \) is the control input for each player \( i \). In these standard games, the performance index for each player \( i \) is defined as

\[
\tilde{J}_i(\delta(k), u_i(k)) = \frac{1}{2} \sum_{k=0}^{\infty} \delta(k)^T \hat{Q}_i \delta(k) + \sum_{j \in N} u_j^T(k) \hat{R}_i u_j(k)
\]

(15)

where \( \hat{Q}_i > 0 \in \mathbb{R}^{N \times N}, \hat{R}_i > 0 \in \mathbb{R}^{m \times m} \).

Both the dynamics (14) and the performance indices (15) depend on the control actions of all other players.

To compare these standard games to the dynamic graphical games defined in this paper, write the global error dynamics for the dynamic graphical game as

\[
\epsilon(k + 1) = (I_N \otimes A)\epsilon(k) - ((L + G) \otimes I_N) \tilde{B} u(k)
\]

(16)

with \( \tilde{B} = \text{diag}[B_1, \ldots, B_i, \ldots, B_N] \) and \( u(k) = [u_1^T \ u_2^T \ \cdots \ u_N^T]^T \) is the global vector of control inputs. To find the difference between the standard centralized game (14) and the dynamic graphical game (16), define \( F = (I_N \otimes A) \) and write (14) as

\[
\delta(k + 1) = (I_N \otimes A) \delta(k) + \{B_1 \ \cdots \ B_N\} u(k).
\]

(17)

Now, define \( l_j \) as the \( j \)th element of \( (L + G) \) and write (16) as

\[
\epsilon(k + 1) = (I_N \otimes A)\epsilon(k) - \{l_j B_j\} u(k)
\]

(18)

where \( \{l_j B_j\} \) is a matrix whose \( j \)th block is \( l_j B_j \).

Eqs. (17) and (18) are same if one defines the block column matrix \( B' \) as \( B' = [l_1 B_1 \ l_2 B_2 \ \cdots \ l_N B_N]^T \).

It is easily observed that the graphical game local performance index (12) is a special case of the standard game performance index (15) with appropriate definition of \( \hat{Q}_i, \hat{R}_i \). Therefore, the dynamic graphical game is a special case of the standard game which explicitly displays the graph topology through \( (L + G) \) in (18).

The dynamic graphical game formulation explicitly captures the structure of the communication graph. Therefore, its analysis clearly reveals the interplay of individual node dynamics and the graph topology within a multi-player game. Moreover, as seen in Section 6, it allows the solution of the game in a distributed fashion. Note that the coupled game Riccati equations of each agent presented in Başar and Olsder (1999), depend on the policies of all other agents, and so provide a centralized solution for the game. Moreover, existence of solutions to those coupled game Riccati equations requires reachability conditions that are closely related to requirement (49) in our definition of interactive Nash equilibrium in Section 4.2. See Lemma 3. For these conditions to hold, the graph must be strongly connected.

3.3 Bellman equation for dynamic graphical games

Taking the first difference of (13) yields the graphical game Bellman equations

\[
V_i(\epsilon_{ik}) = U_i(\epsilon_{ik}, u_k, \epsilon_{i(k+1)}) + V_i(\epsilon_{i(k+1)})
\]

(19)

with initial conditions \( V_i(0) = 0 \).

The objective of the graphical games optimization problem is to find the optimal value

\[
V^{*}_i(\epsilon_{ik}) = \min_{u_k} (V_i(\epsilon_{ik})) = \min_{u_k} \left( \sum_{k=0}^{\infty} U_i(\epsilon_{ik}, u_k, \epsilon_{i(k+1)}) \right)
\]

(20)

where \( \bar{U}_i = \{u_k\}_{k=0}^{\infty}, \forall i \in N \). According to the Bellman optimality principle

\[
V^{*}_i(\epsilon_{ik}) = \min_{u_k} (U_i(\epsilon_{ik}, u_k, \epsilon_{i(k+1)})) + V^{*}_i(\epsilon_{i(k+1)})
\]

(21)

Consequently, the optimal control policy for each agent \( i \) is

\[
u^*_k = (d_i + g_i)R_i^{-1}B_i V_i^{*}(\epsilon_{i(k+1)}).
\]

(22)

Substituting (22) into (21) yields the coupled graphical game Bellman optimality equations

\[
V_i^{*}(\epsilon_{ik}) = V_i^{*}(\epsilon_{i(k+1)}) + \frac{1}{2} \left( \epsilon_{ik}^{T}Q_i \epsilon_{ik} + (d_i + g_i)^2 V_i^{*}(\epsilon_{i(k+1)}) + \sum_{j \in N} u_j^{T}R_j u_j \right)
\]

(23)

with initial conditions given by \( V_i^{*}(0) = 0 \).

3.4 Hamiltonian function for dynamic graphical games

Consider the node error dynamics (9) and the performance indices (12). We can define the Hamiltonian function (Lewis et al., 2012) of each agent \( i \) as

\[
H_i(\epsilon_{i(k)}, \lambda_{i(k+1)}, u_k, u_{i-k}) = \epsilon_{i(k+1)}^T (A_{i-k} - (d_i + g_i)B_{i-k}) u_{i-k} + \sum_{j \in N} \epsilon_{j} B_{j} u_{j}
\]

(24)

where \( \lambda_{i(k)} \equiv \lambda_i(k) \) is the costate or adjoint variable of each agent \( i \). The necessary conditions for optimality allow us to find the costate equations as

\[
\partial H_i/\partial \lambda_{i(k)} = \lambda_{i(k)} \Rightarrow \lambda_{i(k)} = A^T \lambda_{i(k+1)} + Q_{i} \epsilon_{i(k)}.
\]

(25)

The optimal control policy based on the Hamiltonian (24) is given by the stationarity condition (Lewis et al., 2012) \( \partial H_i/\partial u_k = 0 \), so that

\[
u_{ik} = \arg \min_{u_k} (H_i(\epsilon_{ik}, \lambda_{i(k+1)}, u_k, u_{i-k}))
\]

(26)

or

\[
u_{ik} = (d_i + g_i)R_i^{-1}B_i \lambda_{i(k+1)}.
\]

(27)

3.5 Discrete Hamilton–Jacobi theory: equivalence of Hamiltonian and Bellman optimality equations

For the following development, we will define the first difference of the value function \( V_i(\epsilon_{ik}) \) as

\[
\Delta V_i(\epsilon_{ik}) = V_i(\epsilon_{i(k+1)}) - V_i(\epsilon_{ik})
\]

(28)

and its gradient as

\[
\nabla V_i(\epsilon_{i(k+1)}) = \partial V_i(\epsilon_{i(k+1)}) / \partial \epsilon_{ik}
\]

(29)

The next result (cf. Lall & West, 2006) relates the Hamiltonian (24) to the value (13). It also introduces the discrete-time HJ equation, which relates (28) and (29).

Theorem 1 (Discrete-Time Hamilton–Jacobi Equation). Consider the Hamiltonian equation (24) and define the value function \( V_i(\epsilon_{ik}) \)
by (13). Then, $V_i(\epsilon_{ik})$ satisfies the following discrete-time Hamilton–Jacobi (DTHJ) equation

$$\Delta V_i(\epsilon_{ik}) - V_i(\epsilon_{ik+1})^T e_{ik+1}$$

$$+ H_i(\epsilon_{ik}, \nabla V_i(\epsilon_{ik+1}), u_{ik}, u_{-ik}) = 0.$$  

(30)

**Proof.** The objective of optimal control is to minimize the performance index for each agent $i (12)$. The optimization problem is subject to the equality constraint

$$\epsilon_{ik+1} = f(\epsilon_{ik}, u_{ik}, u_{-ik}).$$

Then one should define augmented value function as follows

$$V_i(\epsilon_{ik}) = \sum_{\mathcal{I}} \left[ U_i(\epsilon_{ik}, u_{ik}, u_{-ik}) + \lambda_{ik+1}^T f(\epsilon_{ik}, u_{ik}, u_{-ik}) - \epsilon_{ik+1} \right]$$

(32)

with a corresponding Hamiltonian given by,

$$H_i(\epsilon_{ik}, \lambda_{ik+1}, u_{ik}, u_{-ik}) = U_i(\epsilon_{ik}, u_{ik}, u_{-ik}) + \lambda_{ik+1}^T f(\epsilon_{ik}, u_{ik}, u_{-ik}).$$

(33)

Eqs. (32) and (33) yield

$$V_i(\epsilon_{ik+1}) = \sum_{\mathcal{I}} \left[ H_i(\epsilon_{ik}, \lambda_{ik+1}, u_{ik}, u_{-ik}) - \lambda_{ik+1}^T \epsilon_{ik+1} \right].$$

Substituting these equations into (19) yields

$$\Delta V_i(\epsilon_{ik}) + H_i(\epsilon_{ik}, \lambda_{ik+1}, u_{ik}, u_{-ik}) - \lambda_{ik+1}^T \epsilon_{ik+1} = 0.$$  

(34)

Taking the derivative of (34) with respect to $\epsilon_{ik+1}$ yields

$$\nabla V_i(\epsilon_{ik+1}) - \left[ (\partial \lambda_{ik+1}) / \partial \epsilon_{ik+1} \right] \epsilon_{ik+1} + \lambda_{ik+1}$$

$$+ (\partial \lambda_{ik+1}) / \partial \epsilon_{ik+1} \right] \epsilon_{ik+1} + \lambda_{ik+1} = 0.$$  

Rearranging this equation yields

$$\nabla V_i(\epsilon_{ik+1}) = \lambda_{ik+1} + (\partial \lambda_{ik+1}) / \partial \epsilon_{ik+1} \right] \epsilon_{ik+1} + \lambda_{ik+1}.$$

(35)

Eq. (35) yields $\lambda_{ik+1} = \nabla V_i(\epsilon_{ik+1})$. Substituting this into (34) yields (30).

This proof motivates us to define the costate in the value function as

$$\lambda_{ik+1} = \nabla V_i(\epsilon_{ik+1}).$$

(36)

The optimal control policy based on the Bellman optimality equation (23) is given by (22). The next result shows the relation between the policies (22) and (27). It also relates the Hamiltonian (24) and Bellman optimality equation (23).

**Theorem 2 (Discrete-Time Hamilton–Jacobi Bellman Equation).**

a. Let $0 < V^*_i(\epsilon_{ik}) \in C^2$. Vi satisfy the Discrete-Time Hamilton–Jacobi–Bellman (DTHJB) equation

$$H_i(\epsilon_{ik}, \nabla V^*_i(\epsilon_{ik+1}), u^*_ik, u^*_{-ik}) = \nabla V^*_i(\epsilon_{ik+1})^T e_{ik+1}$$

$$+ \frac{1}{2} \left( e_{ik}^T Q_{ik} e_{ik} + u^*_ik^T R_{ik} u^*_ik + \sum_{j \in \mathcal{N}} u^*_jk^T R_{jk} u^*_jk \right) = 0.$$  

(37)

with initial condition given by $V^*_i(\epsilon_{ik}) = 0$, where

$$u^*_ik = (d_{ik} + g_{ik}) R_{ik}^{-1} B_{ik}^T \nabla V^*_i(\epsilon_{ik+1}).$$

(38)

Then $V^*_i(\epsilon_{ik})$ satisfies the Bellman optimality equation (23).

b. Let $(A, B)$ be reachable. Let $0 < V^*_i(\epsilon_{ik}) \in C^2$. Vi satisfy (23). Then $V^*_i(\epsilon_{ik})$ satisfies (37).

*Proof.** a. $V^*_i(\epsilon_{ik})$ satisfies (37) and $u^*_ik$ is given by (38) such that $H_i(\epsilon_{ik}, \nabla V^*_i(\epsilon_{ik+1}), u^*_ik, u^*_{-ik}) = 0$. Then, by applying the results from Theorem 1 we have $\Delta V^*_i(\epsilon_{ik}) = \nabla V^*_i(\epsilon_{ik+1})^T e_{ik+1}$ from which the result follows.

b. Completing the squares on the Hamiltonian (24) for an arbitrary smooth function $V_i(\epsilon_{ik})$ yields

$$H_i(\epsilon_{ik}, \nabla V_i(\epsilon_{ik+1}), u_{ik}, u_{-ik})$$

$$= H_i(\epsilon_{ik}, \nabla V_i(\epsilon_{ik+1}), u^*_ik - u^*_{-ik}) + \frac{1}{2} (u_{ik} - u^*_ik)^T R_{ik} (u_{ik} - u^*_ik)$$

$$+ \frac{1}{2} \sum_{j \in \mathcal{N}} (u_{jk} - u^*_jk)^T R_{jk} (u_{jk} - u^*_jk) + \sum_{j \in \mathcal{N}} u^*_jk^T R_{jk} (u_{jk} - u^*_jk)$$

$$+ \sum_{j \in \mathcal{N}} e_j^T \nabla V_i(\epsilon_{ik+1})^T B_j (u_{jk} - u^*_jk) = 0.$$  

(39)

where $u^*_ik = (g_{ik} + d_{ik}) R_{ik}^{-1} B_{ik}^T \nabla V_i(\epsilon_{ik+1})$. VI. Now, let $V_i(\epsilon_{ik}) \in C^2$, $\forall i$ satisfy the Bellman equation (19). The Hamiltonian with optimal value given by $V^*_i(\epsilon_{ik})$ for arbitrary control policies yields

$$H_i(\epsilon_{ik}, \nabla V^*_i(\epsilon_{ik+1}), u_{ik}, u_{-ik})$$

$$= \nabla V^*_i(\epsilon_{ik+1})^T e_{ik+1} + U_i(\epsilon_{ik}, u_{ik}, u_{-ik})$$

$$= \frac{1}{2} \sum_{j \in \mathcal{N}} (u_{ik} - u^*_ik)^T R_{ik} (u_{ik} - u^*_ik)$$

$$+ \frac{1}{2} (u_{ik} - u^*_ik)^T R_{ik} (u_{ik} - u^*_ik) + \sum_{j \in \mathcal{N}} u^*_jk^T R_{jk} (u_{jk} - u^*_jk)$$

$$+ \sum_{j \in \mathcal{N}} e_j^T \nabla V^*_i(\epsilon_{ik+1})^T B_j (u_{jk} - u^*_jk).$$

(40)

Bellman equation (19) can be written as follows

$$V_i(\epsilon_{ik}) = U_i(\epsilon_{ik}, u_{ik}, u_{-ik}) + \nabla V^*_i(\epsilon_{ik+1})^T \epsilon_{ik+1} + V_i(\epsilon_{ik+1}).$$

Substituting the Hamiltonian (40) into this equation yields

$$V_i(\epsilon_{ik}) = V_i(\epsilon_{ik+1}) - \nabla V^*_i(\epsilon_{ik+1})^T \epsilon_{ik+1}$$

$$+ \frac{1}{2} (u_{ik} - u^*_ik)^T R_{ik} (u_{ik} - u^*_ik)$$

$$+ \frac{1}{2} \sum_{j \in \mathcal{N}} (u_{jk} - u^*_jk)^T R_{jk} (u_{jk} - u^*_jk)$$

$$+ \sum_{j \in \mathcal{N}} u^*_jk^T R_{jk} (u_{jk} - u^*_jk) + \sum_{j \in \mathcal{N}} e_j^T \nabla V^*_i(\epsilon_{ik+1})^T B_j (u_{jk} - u^*_jk).$$

(41)

Bellman’s optimality principle, yields that $V^*_i(\epsilon_{ik})$ has to satisfy the following equation

$$V^*_i(\epsilon_{ik}) = \min_{u_{ik}} \left\{ V_i(\epsilon_{ik+1}) - \nabla V^*_i(\epsilon_{ik+1})^T \epsilon_{ik+1} \right\}$$

$$+ \frac{1}{2} (u_{ik} - u^*_ik)^T R_{ik} (u_{ik} - u^*_ik) + \sum_{j \in \mathcal{N}} u^*_jk^T R_{jk} (u_{jk} - u^*_jk)$$

$$+ \frac{1}{2} \sum_{j \in \mathcal{N}} (u_{jk} - u^*_jk)^T R_{jk} (u_{jk} - u^*_jk)$$

$$+ \sum_{j \in \mathcal{N}} u^*_jk^T R_{jk} (u_{jk} - u^*_jk) + \sum_{j \in \mathcal{N}} e_j^T \nabla V^*_i(\epsilon_{ik+1})^T B_j (u_{jk} - u^*_jk).$$

(41)

Applying the stationarity conditions $\partial V^*_i(\epsilon_{ik}) / \partial u_{ik} = 0$, the control policy $u^*_ik$ is given by solving

$$- (g_{ik} + d_{ik}) R_{ik}^{-1} B_{ik}^T \nabla V^*_i(\epsilon_{ik+1}) + R_{ik} (u^*_ik - u^*_{-ik}) = 0.$$
The hessians of the Hamiltonian (24) and Bellman equations (19) with respect to all control policies are positive definite values since 
\[ V_{i}^{2} (H_{i}) = R_{i} \text{ and } V_{i}^{2} (V_{i}) = R_{i} \]. Therefore the optimal control policy is unique \( u_{i}^{*} = u_{0}^{i} \) \( \forall k \).

Now, (42) and the costate (25) show that \( (g_{i} + d_{i})R_{i}^{-1}B_{i}^{T}(A^{T})^{2}((V_{i}^{2})(\epsilon_{i(k+1)}) - V_{i}^{0}(\epsilon_{i(k+1)}))) = 0 \), \( \forall k \), \( p = 0, 1, \ldots, n - 1 \).

The reachability matrix 
\[ \bar{U}_{k} = [B_{i}, AB_{i}, A^{2}B_{i}, \ldots, A^{n-1}B_{i}] \] (44)
under the hypothesis has full rank. Therefore, since \( V_{i}^{0} (0) = 0 \) and \( V_{i}^{0} (0) = 0 \) then,
\[ V_{i}^{*}(\epsilon_{i(k+1)}) = V_{i}^{0}(\epsilon_{i(k)}) \], \( \forall k \) (45)
from which the result follows.

The next lemma relates the first difference of the optimal value function and its gradient.

**Lemma 2.** Let \( V_{i}^{*}(\epsilon_{i}) \) satisfy the Bellman optimality equation (23) or equivalently the coupled HJ (37) then,
\[ \nabla V_{i}^{*}(\epsilon_{i(k+1)})^T \epsilon_{i(k+1)} = \Delta V_{i}^{*}(\epsilon_{i}) \] (46)
or
\[ (\partial V_{i}^{*}(\epsilon_{i(k+1)})/\partial \epsilon_{i(k+1)})^T \epsilon_{i(k+1)} = V_{i}^{*}(\epsilon_{i(k+1)}) - V_{i}^{*}(\epsilon_{i}) \].

**Proof.** The proof follows from the results of Theorems 1 and 2.

## 4. Nash solution for dynamic graphical games

The objective of a dynamic graphical game is to solve the non-cooperative minimization problem given by the coupled equations (20), which lead to the Bellman optimality equations (23). It is shown here that the concept of Nash equilibrium is incomplete for dynamic graphical games, since it does not take into account the graph connectivity properties.

### 4.1. Nash equilibrium and idiosyncrasies of dynamic graphical games

The next definition is given with respect to the actions of all the other players \( u_{i} = \{ u_{j} \mid j \in N, j \neq i \} \), namely a complete graph.

**Definition 3** (Başar & Olsder, 1999). The N-player dynamic graphical game with N-tuple of optimal control policies \( u_{i}, u_{j}, \ldots, u_{n} \) is said to have a global Nash equilibrium solution if for all \( i \in N \)
\[ J_{i}^{*} \triangleq J_{i}(u_{i}, u_{j}^{*}) \leq J_{i}(u_{i}, u_{j}). \] (47)
The N-tuple \( \{ J_{i}^{*}, J_{j}^{*}, \ldots, J_{n}^{*} \} \) is called the Nash equilibrium outcome of the N-player game.

### 4.2. Interactive Nash equilibrium

In the case of a disconnected graph, the agents can be in Nash equilibrium, yet have no influence on each other. In such situations, the definition of coalition-proof Nash equilibrium (Shimohara, 2010) may also hold, that is, no set of agents has an incentive to break away from the Nash equilibrium and seek a new Nash solution among them. To guarantee that all agents in a graph are involved in the same game, the stronger definition of interactive Nash equilibrium is introduced here (Vamvoudakis et al., 2012).

**Definition 4** (Shoham & Leyton-Brown, 2009). Agent i’s best response to fixed control actions of his neighbors is the policy \( u_{i}^{*} \) such that
\[ J_{i}(u_{i}^{*}, u_{-i}) \leq J_{i}(u_{i}, u_{-i}) \] (48)

It is noted that, as in the standard case of multi-player games with a centralized state (14), all agents are in best response to their neighbors (e.g. Definition 4) if and only if they are in global Nash equilibrium (e.g. Definition 3).

The next definition provides a strengthened notion of Nash equilibrium that is suitable for graphical games.

**Definition 5** (Interactive Global Nash Equilibrium). An N-tuple of policies \( u_{1}^{*}, u_{2}^{*}, \ldots, u_{n}^{*} \) is said to constitute an interactive global Nash equilibrium solution for an N-player game if, for all \( i \in N \), the Nash condition (47) holds and in addition there exists a policy \( u_{i}^{*} \) such that
\[ J_{i}(u_{i}^{*}, u_{i}^{*}) \neq J_{i}(u_{i}^{*}, u_{-i}). \] (49)
For all \( i, z \in N \), that is, at the equilibrium point there exists a policy of every player z that influences the performance of all other players i.

Condition (49) means that the reaction curve (Başar & Olsder, 1999) of any player i is not constant with respect to all variations in the policy of any other player z. For that reason one has to find conditions under which the local best responses in Definition 4 imply the interactive global Nash of Definition 5.

Consider the systems given by (9) in closed-loop with admissible feedbacks given by (25) and (27). Denote a suboptimal policy \( u_{i} = K_{i}z_{i} + K_{i}z_{i} + u_{i} \) for a single node z and optimal policies given by \( u_{i} = K_{i}z_{i} + K_{i}z_{i}, \forall j \neq z \). That is, node z has an extra control input \( u_{z} \). Then we have
\[ e_{ik(k+1)} = Ae_{ik} - (d_{i} + g_{i})B_{i}u_{ik} + \sum_{j \in N_{i}} e_{jB}B_{j}u_{jk} + e_{zB}u_{zk}. \] (50)
and
\[ K_{i} = -(g_{i} + d_{i})B_{i}B_{i}A^{-1}Q_{i}. \] (51)

According to (25) and (50) the global closed-loop dynamics are given by the Hamiltonian system
\[ e_{k+1} = A_{z}e_{k} - A_{z}e_{k}B_{z} + B_{z}. \] (52)
where \( B = [(l + G) \otimes I]diag(\beta_{i}) \), \( \beta_{z} = [0, \ldots, 0, 0, 0] \), and \( A = [(l + G) \otimes I]diag(\beta_{i}) - diag(A^{-1}Q_{i}) \).

Consider node i and let \( M > 0 \) be the first integer such that \( M (L + G) \beta_{z} \neq 0 \), where \( \beta_{z} \) denotes the element (i, z) of a matrix. That is, \( M \) is the length of the shortest directed path from node z to node i. Denote the nodes along this path by \( z = z_{0}, z_{1}, \ldots, z_{M-1}, z_{M} = i \). Denote element (i, z) of (L + G) by \( \beta_{z} \). Then the \( n \times m \) block element in block row i and block column z of matrix \( \tilde{A}^{MT-1}\tilde{B} \in R^{nMxM} \) is equal to
\[ \tilde{A}^{MT-1}\tilde{B}^{i} = \sum_{j \in M^{z-1}} \beta_{z}^{i}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j}B_{z}^{j} \] (53)
where \( B_{z}^{i} \in R^{MxM} \) and \( \beta_{z}^{i} \) denotes the position of the block element in the block matrix.

**Assumption 1.** a. The matrix \( (B_{z}^{i}A^{-1}Q_{i}B_{z}^{i}) \) has full row rank for \( M > j \geq 0 \). b. There is unique shortest path between every two nodes.
This assumption holds for a large class of systems and graphs. Note that condition (a) holds if \( m_i = m_j \), \( \forall i, j, B_i = B_j \), \( \forall i, j \), with \( B_i \) of full column rank, \( Q_i > 0 \), \( \forall i \), and \( A \) is nonsingular.

Lemma 3. Let Assumption 1 hold and let the control policies are given by (27). Then the ith performance index of closed-loop system (52) depends on the input \( v_i \) if and only if there exists a directed path from node z to node i.

Proof. Sufficiency. If \( z = i \) the result is obvious. Otherwise, the reachability matrix from node z to node i has the following composition

\[
(\mathbf{A}^{[\overrightarrow{G}]} - \mathbf{B}^T)^z [(\mathbf{A}^{[\overrightarrow{G}]} - \mathbf{B}^T)^{z+1}]^z \ldots
\]

with

\[
(\mathbf{A}^{[\overrightarrow{G}]} - \mathbf{B}^T)^z = \sum_{z \in \overrightarrow{G}_{i} \cup \overrightarrow{G}^{-1}_{i}} \mathbf{A}^{-1} \mathbf{B} \mathbf{A}^{-1} \mathbf{B} \mathbf{A}^{-1} \mathbf{B} \ldots
\]

If there exists a path from node z to i, then the first block matrix has rank \( m_{z^{-1}} \), therefore at least \( m_{z^{-1}} \) nodes of the states of node i are reachable. Thus the state will be a function of \( u_{i_k} \), because \( Q_i \) is positive definite. Therefore, the ith performance index will depend on \( u_{i_k} \).

Necessity. If there is no path from node z to node i, then the control input of node z cannot influence the state or value of node i. ■

Theorem 3. Let every node i be in best response to all its neighbors \( j \in N_i \). Let Assumption 1 hold. Then all nodes in the graph are in interactive Nash equilibrium if and only if the graph is strongly connected.

Proof. Let every node i be in best response to all its neighbors \( j \in N_i \). Then \( f_i(u_i^*, u_{-i}) \leq f_i(u_i, u_{-i}) \), \( \forall i \). Hence \( u_i = u_i^* \), \( \forall u_i \leq u_{i_k}^* \) and \( f_i(u_i^*, u_{-i}) \leq f_i(u_i, u_{i_k}^*) \), \( \forall i \).

However, according to (12) \( f_i(u_i^*, u_{-i}^*) = f_i(u_i, u_i^*) \), \( \forall u_i \leq u_{i_k}^* \), so that when \( u_i^* \) is reached, \( u_i = u_i^* \) and the nodes are in Nash equilibrium.

Necessity. If the graph is not strongly connected, then there exist nodes z and i such that there is no path from node z to node i. Then, the control input of node z cannot influence the state or value of node i. Therefore, the Nash equilibrium is not interactive.

Sufficiency. If there is a path from node z to node i, the performance index depends on \( u_z \). Strong connectivity means there is a path from every node k to every node i and condition (49) holds for all i, z \( \in N \).

According to the results just established, the following assumption is made.

Assumption 2. The graph is strongly connected and \( g_i \) is non-zero for at least one root node i. ■

Note that existence of a spanning tree is necessary and sufficient for synchronization of all states with dynamics given in (1). However, interactive Nash equilibrium requires the graph to be strongly connected.

4.3. Stability and Nash solution of the graphical games

We will now prove that the policies given in terms of the solutions to the coupled Bellman optimality equations (23) provide Nash equilibrium solution for the dynamic graphical game.

Theorem 4 (Stability and Nash Equilibrium Solution). Let \( 0 < V_i^*(\epsilon_{ik}) \in C^2 \) satisfy DTHJB (37), or equivalently the Bellman optimality equation (23). Let all agents use the control policies given by (38).

Let the graph contain a spanning tree with at least one nonzero pinning gain. Then:

a. The error dynamics (9) are asymptotically stable and all agents synchronize to the target node dynamics (2).

b. The optimal performance index for each agent i is given by \( J_i^*(\epsilon_{il}) \).

c. All agents are in Nash equilibrium.

Proof. a. \( V_i^*(\epsilon_{ik}) \) satisfies the Bellman optimality equation such that

\[
V_i^*(\epsilon_{ik+1}) - V_i^*(\epsilon_{ik}) = -U_i^*(\epsilon_{ik}, u_{ik}, u_{i_k}^*) < 0.
\]

Therefore, \( V_i^*(\epsilon_{ik}) \) serves as Lyapunov function for (9), and the error system (9) is asymptotically stable. If there is a spanning tree, then according to Lemma 1, all agents synchronize to the targets node dynamics.

b. Using Theorem 2 and DTHJB (37), then the Hamiltonian (39) for arbitrary control policies is given by

\[
H_i(\epsilon_{ik}, \nabla V_i^*(\epsilon_{ik+1}), u_{ik}, u_{-ik}) = \nabla V_i^*(\epsilon_{ik+1})^T(\epsilon_{ik+1}) + U_i(\epsilon_{ik}, u_{ik}, u_{-ik})
\]

\[
= \frac{1}{2} \sum_{j \in N_i} (u_{jk} - u_{ik})^T R_j(u_{jk} - u_{ik}) + \frac{1}{2} \sum_{j \in N_i} (u_{jk} - u_{ik})^T R_j^*(u_{jk} - u_{ik})
\]

\[
+ \sum_{j \in N_i} u_{jk}^T R_j(u_{jk} - u_{ik}) + \sum_{j \in N_i} u_{jk}^T R_j^*(u_{jk} - u_{ik}).
\]

Using the result in part a, \( \epsilon_{i}(\infty) \to 0 \). Therefore \( V_i^*(\epsilon_{i}(\infty)) = 0 \) and

\[
J_i(\epsilon_{i}, u_{i}, u_{-i}) = V_i^*(\epsilon_{i}(\infty)) + \sum_{k=0}^{\infty} U_i(\epsilon_{ik}, u_{ik}, u_{-ik}).
\]

Rearranging this equation yields,

\[
J_i(\epsilon_{ik}, u_{ik}, u_{-ik}) = V_i^*(\epsilon_{ik}) + \sum_{k=1}^{\infty} (U_i(\epsilon_{ik}, u_{ik}, u_{-ik})
\]

\[
- U_i^*(\epsilon_{ik}, u_{ik}^*, u_{i_k}^*).
\]

The Hamiltonian for arbitrary control inputs is given by

\[
H_i(\epsilon_{ik}, \nabla V_i^*(\epsilon_{ik+1}), u_{ik}, u_{-ik}) = \nabla V_i^*(\epsilon_{ik+1})^T(\epsilon_{ik+1})|_{u_{ik}^*, u_{i_k}^*} + U_i(\epsilon_{ik}, u_{ik}, u_{-ik}).
\]

The Hamiltonian for optimal control inputs is given by

\[
H_i(\epsilon_{ik}, \nabla V_i^*(\epsilon_{ik+1}), u_{ik}^*, u_{i_k}^*) = \nabla V_i^*(\epsilon_{ik+1})^T(\epsilon_{ik+1})|_{u_{ik}^*, u_{i_k}^*} + U_i(\epsilon_{ik}, u_{ik}^*, u_{i_k}^*) = 0.
\]

It is further noted that,

\[
- \nabla V_i^*(\epsilon_{ik+1})^T(\epsilon_{ik+1})|_{u_{ik}, u_{i_k}} + \nabla V_i^*(\epsilon_{ik+1})^T(\epsilon_{ik+1})|_{u_{ik}^*, u_{i_k}^*} = u_{ik}^* R_j(u_{ik} - u_{ik}^*) - \sum_{j \in N_i} u_{ik}^* R_j^*(u_{ik} - u_{ik}^*).
\]

Eqs. (59), (60), and (61) yield

\[
U_i(\epsilon_{ik}, u_{ik}, u_{-ik}) = U_i^*(\epsilon_{ik}, u_{ik}^*, u_{i_k}^*) - \frac{1}{2} (u_{ik} - u_{ik}^*)^T R_j(u_{ik} - u_{ik}^*) + \frac{1}{2} \sum_{j \in N_i} (u_{ik} - u_{ik}^*)^T R_j(u_{ik} - u_{ik}^*) + \sum_{j \in N_i} (u_{i_k} - u_{i_k}^*)^T R_j(u_{i_k} - u_{i_k}^*).
\]
Using (62) into (58) yields
\[
J_i(\varepsilon_i, u_i, u_{-i}) = V_i^*(\varepsilon_i) + \sum_{k \neq i} \left( \sum_{j \neq k} \frac{1}{2} (u_{ik} - u_{jk})^T R_{ik} (u_{ik} - u_{jk}) + u_{ik}^T R_k (u_{ik} - u_{ik}^*) + \frac{1}{2} \sum_{j \neq k} (u_{jk} - u_{jk}^*) R_{jk} (u_{jk} - u_{jk}^*) \right). 
\]
Using (63) with the optimal policies (38) yields the optimal performance index $J_i^*$ such that
\[
J_i^*(\varepsilon_i, u_i^*, u_{-i}^*) = V_i^*(\varepsilon_i). 
\]
c. Given that the summation of the performance index (58) is positive for arbitrary control policies such that
\[
\sum_{k \neq i} U_i(\varepsilon_i, u_i, u_{-i}) - U_i^*(\varepsilon_i, u_i, u_{-i}) > 0. 
\]
Eqs. (63), (64), and (65) yield
\[
J_i^*(\varepsilon_i, u_i^*, u_{-i}^*) \leq J_i(\varepsilon_i, u_i, u_i^*) \quad \text{(66)}
\]
from which the result follows, according to Definition 3. ■

The next result shows that the interactive Nash equilibrium requires the graph to be strongly connected.

**Lemma 4.** Let the hypotheses of Theorem 4 and Assumptions 1 and 2 hold. Then $[u_1^*, u_2^*, \ldots, u_n^*]$ are in interactive Nash equilibrium and hence all agents synchronize to the target node dynamics.

**Proof.** The proof is a consequence of Theorems 1–4. ■

4.4. Best response solution of dynamic graphical games

Theorem 4 and Lemma 4 reveal that the agents are in interactive Nash equilibrium if, for all $i \in N$, agent $i$ selects its best response to its neighbors policies and the graph is strongly connected.

Now by considering fixed neighborhood policies $u_{-i} = \{u_j : j \in N\}$, we can define the best response Bellman equation for each agent $i$ as
\[
V_i^*(\varepsilon_i) = V_i^*(\varepsilon_i(\varepsilon_i) + 1 \left( e_i^T R_{ii} e_i + \sum_{j \neq i} u_{ij}^T R_{ij} u_{ij} + (d_i + g_i)^2 \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1) B_{ii} R_{ii}^{-1} B_{ii}^T \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1) \right) \quad \text{(67)}
\]
with initial condition given by $V_i^0(0) = 0$ and $u_{ik} = u_{ik}^*$ where $u_{ik}^*$ is given by (22) in terms of the solution (67).

Define the best response Hamilton-Jacobi (HJ) equation as
\[
H_i(\varepsilon_i, \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1), u_i^*, u_{-i}) = \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1) + \frac{1}{2} \left( e_i^T R_{ii} e_i + u_{ij}^T R_{ij} u_{ij} + \sum_{j \neq i} u_{ij}^T R_{ij} u_{ij} \right) = 0 \quad \text{(68)}
\]
with $V_i^0(0) = 0$ and $u_{ik} = u_{ik}^*$ where $u_{ik}^*$ is given by (38) in terms of the solution to (68).

The next lemma shows the relation between the best response Bellman equation (67) and the best response Hamilton-Jacobi equation (68).

**Lemma 5.** a. Let $0 < V_i^*(\varepsilon_i) \in C^2$, $\forall i$ satisfy the best response (DHJ) equation (68) with initial condition given by $V_i^0(0) = 0$, and optimal control policy $u_{ik}^*$ given by (38). Then, $V_i^*(\varepsilon_i)$ satisfies the best response Bellman equation (67).

b. Let $(A, B_i)$ be reachable. Let $0 < V_i^*(\varepsilon_i) \in C^2$, $\forall i$ satisfy (67). Then $V_i^*(\varepsilon_i)$ satisfies (68).

**Proof.** a. Let $V_i^*(\varepsilon_i)$ satisfy (68) and $u_{ik}^*$ be given by (38), then $H_i(\varepsilon_i, \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1), u_{ik}^*, u_{-i}) = 0$. Then by using Theorem 1, we have \( \Delta V_i^*(\varepsilon_i) = \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1) u_{ik}^* \). Therefore $V_i^*(\varepsilon_i)$ satisfies (67).

b. The best response Hamiltonian with arbitrary smooth value $V_i^*(\varepsilon_i)$ for arbitrary control policy $u_{ik}$ is given by:
\[
H_i(\varepsilon_i, \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1), u_{ik}, u_{-i}) = \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1) u_{ik}^* + \frac{1}{2} (u_{ik} - u_{ik}^*) R_i (u_{ik} - u_{ik}^*). \quad \text{(69)}
\]

Now, let $V_i(\varepsilon_i) \in C^2$, $\forall i$ satisfy (19) such that
\[
V_i(\varepsilon_i) = U_i(\varepsilon_i, u_i, u_{-i}) + \left( \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1) u_{ik}^* \right) + V_i(\varepsilon_i(\varepsilon_i) + 1). \quad \text{(70)}
\]

Substituting (69) into (70) yields
\[
V_i(\varepsilon_i) = \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1) u_{ik}^* + \frac{1}{2} (u_{ik} - u_{ik}^*) R_i (u_{ik} - u_{ik}^*). \quad \text{(71)}
\]

By applying Bellman’s optimality principle, yields that $V_i^*(\varepsilon_i)$ has to satisfy the following equation
\[
V_i^*(\varepsilon_i) = \min_{u_{ik}} \left[ V_i^0(0) - \left( \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1) u_{ik}^* \right) + \frac{1}{2} (u_{ik} - u_{ik}^*) R_i (u_{ik} - u_{ik}^*) \right]. \quad \text{(72)}
\]

By applying the stationarity condition $\partial V_i(\varepsilon_i)/\partial u_{ik} = 0$ one has to solve the following equation for the control policy $u_{ik}^*$:
\[
u_{ik}^* - u_{ik}^* = (g_i + d_i) R_{ii}^{-1} B_i^T (\nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1) - \nabla V_i^*(\varepsilon_i(\varepsilon_i) + 1)). \quad \text{(73)}
\]

The remainder of the proof involves the reachability result from Theorem 2. ■

**Theorem 5 (Best Response Solution).** Given fixed neighboring policies $u_{-i} = \{u_j : j \in N\}$, assume there exists an admissible policy $u_i$. Let $0 < V_i^*(\varepsilon_i) \in C^2$ satisfy the best response (HJ) equation (68), or equivalently the best response Bellman equation (67). Let each agent $i$ use control policy (38). Let the graph contain a spanning tree with at least one pinning gain nonzero. Then:

a. The error dynamics (9) are asymptotically stable and all the agents synchronize to the target node dynamics (2).

b. The optimal performance index for each agent $i$ is given by $J_i^*(\varepsilon_i, u_i, u_{-i}) = V_i^*(\varepsilon_i)$.

c. All agents are in Nash equilibrium.

**Proof.** a. Suppose that $V_i^*(\varepsilon_i)$ satisfies (67) such that,
\[
V_i^*(\varepsilon_i(\varepsilon_i) + 1) - V_i^*(\varepsilon_i) = -u_i^T (e_i^T R_{ii} e_i + u_{ij}^T R_{ij} u_{ij} + \sum_{j \neq i} u_{ij}^T R_{ij} u_{ij}) < 0. \quad \text{(72)}
\]

Therefore, $V_i^*(\varepsilon_i)$ serves as Lyapunov function for the error system (9), and the error system (9) is asymptotically stable. Hence, according to Lemma 1, all agents synchronize to the targets node dynamics (2).

b. Using Lemma 5 and the best response (HJ) equation (68), then the best response Hamiltonian for arbitrary control policies is given by (69).
Using the result in part a, \(e_i(\infty) \to 0\). Therefore \(V_i^*(e_i(\infty)) = 0\) and the best response performance index for each agent \(i\) is given by
\[
J_i(e_i, u^*_i, u_{-i}) = V_i^*(e_i, u^*_i, u_{-i}) + \sum_{k=1}^\infty \left( U_i(e_i, u_i, u_{-i}) - U_i^*(e_i, u^*_i, u_{-i}) \right). \tag{73}
\]
The best response Hamiltonian with arbitrary control input \(u_i\) is given by
\[
H_i(e_i, \nabla V_i^*(e_i(k+1)), u_i, u_{-i}) = \nabla V_i^*(e_i(k+1))^T e_i(k+1) u_{-i} + U_i^*(e_i, u_i, u_{-i}). \tag{74}
\]
The best response (HJ) equation for agent \(i\) is given by
\[
H_i(e_i, \nabla V_i^*(e_i(k+1)), u_i, u_{-i}) = \nabla V_i^*(e_i(k+1))^T e_i(k+1) u_{-i} + U_i^*(e_i, u_i, u_{-i}) = 0. \tag{75}
\]

If it is straightforward to see that,
\[
U_i(e_i, u_i, u_{-i}) - U_i^*(e_i, u_i, u_{-i}) = \frac{1}{2}(u_i - u_i^*)^T R_i (u_i - u_i^*) + u_i^T R_i (u_i - u_i^*). \tag{76}
\]

Using (76) into (73) yields
\[
J_i(e_i, u_i, u_{-i}) = V_i^*(e_i, u_i, u_{-i}) + \sum_{k=1}^\infty \left( \frac{1}{2}(u_i - u_i^*)^T R_i (u_i - u_i^*) + u_i^T R_i (u_i - u_i^*) \right). \tag{77}
\]

The best response performance index (77) with the optimal control policy (38) is given by the unique value \(V_i^*(e_i)\).
\[
J_i^*(e_i, u_i, u_{-i}) = V_i^*(e_i). \tag{78}
\]

The summation of the best response performance index (77) is positive for arbitrary control policies \(u_i\) such that
\[
\sum_{k=1}^\infty U_i(e_i, u_i, u_{-i}) - U_i^*(e_i, u_i, u_{-i}) > 0. \tag{79}
\]

Eqs. (77), (78), and (79) yield
\[
J_i^*(e_i, u_i, u_{-i}) \leq J_i(e_i, u_i, u_{-i}) \tag{80}
\]

which proves that the policies \(u_i^*\) form a Nash equilibrium according to Definitions 3 and 4.

5. Value iteration algorithm for graphical games

In this section, a value iteration HDP algorithm is proposed for solving the discrete-time dynamic graphical games. This is a cooperative version of adaptive dynamic programming (Werbos, 1974, 1992). Specifically, the single-agent HDP algorithm is extended to the multi-player graphical game.

**Algorithm 1 (HDP Algorithm for Graphical Games)**. Step 1: Start with arbitrary initial policies \(u_i^0\) and values \(V_i^0(e_i)\).

Step 2: Solve for \(V_i^{l+1}\) using Bellman equations
\[
V_i^{l+1}(e_i) = U_i(e_i, u_i^l, u_{-i}^l) + V_i^l(e_i(k+1)) \tag{81}
\]

where \(l\) is the iteration index.

Step 3: Update the control policies using
\[
u_i^{l+1} = (d_i + g_i)R_i^{-1}B_i^l \nabla V_i(e_i(k+1)+1) \tag{82}
\]

Step 4: On convergence of \(V_i(e_i(k)+1) - V_i(e_i(k))\) End. ■

Theorems 6 and 7 that follow provide the convergence proof for Value Iteration Algorithm 1 for two separate cases. In the first case, every agent \(i\) performs the Algorithm 1 while its neighboring agents hold their policies fixed. In second case, all agents update their policies simultaneously using Algorithm 1.

5.1. Best response solution using HDP algorithm

For the first case, all the neighbors of agent \(i\) retain fixed policies \(u_j, j \in N_i\). Then the optimal control policy sequence \(\{u_i^l\}_{l=0}^\infty \in R^{ni}\) for each agent at each iteration step in Algorithm 1 is given by
\[
l_i^l = \arg \min_{u_i} \left( \frac{1}{2} (e_i^T Q_i e_i + u_i^T R_i u_i + \sum_{j \in N_i} u_j^T R_j u_j) + V_i^l(e_i(k+1)) \right). \tag{83}
\]

The notation has been streamlined to simplify the presentation of the proofs, and \(l\) is the iteration index. The associated value function sequence
\[
V_i^{l+1}(e_i) = F_i(V_i^l, l_i) \tag{84}
\]

where \(F_i(V_i^l, l_i) = \frac{1}{2} (e_i^T Q_i e_i + l_i^T R_i l_i + \overline{V}_i^l(R_i, u_{-i})) + V_i^l(e_i(k+1))\)

Now consider arbitrary stabilizing control policy sequence \(\{M_i^l\}_{l=0}^\infty \in R^{ni}\). The associated value function sequence is
\[
Z_i^{l+1}(e_i) = F_i(Z_i^l, M_i^l) = \frac{1}{2} (e_i^T Q_i e_i + M_i^T R_i M_i^l + \overline{Z}_i^l(R_i, u_{-i}) + Z_i^{l+1}(e_i(k+1))) \tag{85}
\]

where \(\overline{Z}_i^l(R_i, u_{-i}) = \sum_{j \in N_i} u_j^T R_j u_j\) has a fixed value. Note that \(\overline{V}_i^l(R_i, u_{-i}) = \overline{Z}_i^l(R_i, u_{-i})\).

The following theorem proves convergence of the HDP algorithm with fixed neighboring policies by using an induction proof following the results of Wang et al. (2012) and Zhang et al. (2009).

**Theorem 6 (Convergence of HDP Algorithm)**. Let the neighbors of each agent \(i\) have fixed policies \(u_i\). Assume there exists an admissible policy \(u_i\). Let agent \(i\) perform Algorithm 1. Then the solution sequence \(\{V_i^l\}_{l=0}^\infty \in R^l\) converges to the best response solution \(V_i^*(e_i) \forall i\) of (67).

**Proof.** According to Lemmas 6, 7 in the Appendix, one has
\[
0 \leq V_i^l \leq Z_i^l \leq U_i. \tag{86}
\]

Using the value function sequences (84) and (85) we can write, \(V_i^{l+1}(e_i) = \frac{1}{2} (e_i^T Q_i e_i + l_i^T R_i l_i + \overline{V}_i^l(R_i, u_{-i})) + V_i^l(e_i(k+1))\) (87)
\[
Z_i^l(e_i) = \frac{1}{2} (e_i^T Q_i e_i + l_i^T R_i l_i + \overline{V}_i^{l-1}(R_i, u_{-i}^{-1}) + Z_i^{l-1}(e_i(k+1))). \tag{88}
\]

Starting with \(l = 0\), we will show that the hypothesis \(Z_i^0(e_i) \leq V_i^{l+1}(e_i)\) holds. By setting \(V_i^l = Z_i^l = 0\) and \(V_i^l(e_i) = \frac{1}{2} (e_i^T Q_i e_i + l_i^T R_i l_i + \overline{V}_i^{l-1}(R_i, u_{-i}^{-1})) \geq 0\), then (87) and (88) for \(V_i^l\) yield
\[
V_i^l(e_i) - Z_i^0(e_i) = \frac{1}{2} (e_i^T Q_i e_i + l_i^T R_i l_i + \overline{V}_i^{l-1}(R_i, u_{-i}^{-1})) \geq 0. \tag{89}
\]
or
\[ Z_i^0(\epsilon_{ik}) \leq V_i^1(\epsilon_{ik}). \tag{90} \]

Now we need to assume that \( Z_i^l(\epsilon_{ik}) \leq V_i^{l+1}(\epsilon_{ik}) \) holds for \( l = 1 \) such that \( Z_i^{l-1}(\epsilon_{ik}) \leq V_i^l(\epsilon_{ik}), \forall l \). Then at (step l) (87) and (88) yield
\[ V_i^{l+1}(\epsilon_{ik}) - Z_i^l(\epsilon_{ik}) = V_i^l(\epsilon_{ik}) - Z_i^{l-1}(\epsilon_{ik}) \geq 0. \tag{91} \]

Therefore the mathematical induction for \( \forall l \) gives
\[ Z_i^l(\epsilon_{ik}) \leq V_i^{l+1}(\epsilon_{ik}), \quad \forall l. \tag{92} \]

Consequently, for all policies \( l_i^l, \forall l \) the following monotonic sequence holds,
\[ V_i^{l+1} > Z_i^l > V_i^l > \cdots > 0. \tag{93} \]

Since \( Z_i^l(\epsilon_{ik}) \) is a lower bound on \( V_i^{l+1}(\epsilon_{ik}) \) and Lemma 7 sets \( \overline{U} \) as an upper bound on \( V_i^{l+1} \), then (86) and (93) yield
\[ 0 \leq V_i^l \leq Z_i^l \leq V_i^{l+1} \leq \overline{U}. \tag{94} \]

From (94) the sequence \( V_i^l \) is increasing and it has an upper bound \( \overline{U} \) which means that \( V_i^l \) converges to the best response solution \( V_i^\ast \) that satisfies the following equation
\[ V_i^\ast(\epsilon_{ik}) = \frac{1}{2} \left( \epsilon_i^T Q_i \epsilon_{ik} + L_i^T R_i L_i^T + \nabla_i (R_{ij}, u_{-i}) \right) + V_i^l(\epsilon_{ik+1}) \tag{95} \]

where
\[ L_i^l = (d_i + g_i) R_i^{-1} B_i \nabla_i V_i^l(\epsilon_{ik+1}). \tag{96} \]

**Remark 1.** In this theorem, the existence of an admissible best response policy is required. Admissibility is a standard assumption and is required in order for an agent to have a best response towards its neighborhood. If this assumption does not hold, there is no solution to the game for the selected policies in the neighborhood. ■

### 5.2. Nash solution using HDP algorithm

For the second case, all the agents \( i \) perform Algorithm 1 simultaneously at each iteration step. Then the control policy sequences \( \{l_i^l\}_{l=0}^\infty \in \mathbb{R}^{m_i} \) for every agent \( i \) are given by (83) and the associated value sequences are given by
\[ V_i^{l+1}(\epsilon_{ik}) = \frac{1}{2} \left( \epsilon_i^T Q_i \epsilon_{ik} + L_i^T R_i L_i^T + \sum_{j \in N_i} L_j^T R_j L_j^T \right) + V_i^l(\epsilon_{ik+1}). \tag{97} \]

For arbitrary admissible policies for the agents \( \{M_i^l\}_{l=0}^\infty \in \mathbb{R}^{m_i} \), their associated value sequences are given by
\[ Z_i^{l+1}(\epsilon_{ik}) = \frac{1}{2} \left( \epsilon_i^T Q_i \epsilon_{ik} + M_i^T R_i M_i^T + \sum_{j \in N_i} M_j^T R_j M_j^T \right) + Z_i^l(\epsilon_{ik+1}). \tag{98} \]

The following theorem proves convergence of the HDP algorithm when all agents update their policies simultaneously. The proof follows the induction proof of Theorem 6.

**Theorem 7 (Convergence of HDP Algorithm).** Assume there exist admissible policies \( u_i, \forall i \). Let all agents update their policies simultaneously using Algorithm 1. Suppose that \( \sigma (R_i^{-1} R_j) \) is small. Then the solution sequences \( \{V_i^l\}_{l=0}^\infty \in \mathbb{R}^{l} \) converge monotonically to the optimal solution \( V_i^\ast(\epsilon_{ik}) \) for \( \forall i \) of (23).

**Proof.** According to Lemmas 8, 9 in the Appendix, one has the following inequality for every agent
\[ 0 \leq V_i^l \leq Z_i^l \leq \overline{U}. \tag{99} \]

The value function sequences (97) and (98), with control policies given by (83) yield
\[ V_i^{l+1}(\epsilon_{ik}) = \frac{1}{2} \left( \epsilon_i^T Q_i \epsilon_{ik} + L_i^T R_i L_i^T + \sum_{j \in N_i} L_j^T R_j L_j^T \right) + V_i^l(\epsilon_{ik+1}) \tag{100} \]

and
\[ Z_i^l(\epsilon_{ik}) = \frac{1}{2} \left( \epsilon_i^T Q_i \epsilon_{ik} + L_i^T R_i L_i^T + \sum_{j \in N_i} L_j^T R_j L_j^T \right) + Z_i^{l-1}(\epsilon_{ik+1}). \tag{101} \]

Starting with \( l = 0 \), we will show that the hypothesis \( Z_i^0(\epsilon_{ik}) \leq V_i^{l+1}(\epsilon_{ik}) \) holds. By setting \( V_i^0 = Z_i^0 = 0 \) and \( V_i^l(\epsilon_{ik}) = \frac{1}{2} \left( \epsilon_i^T Q_i \epsilon_{ik} + L_i^T R_i L_i^T + \sum_{j \in N_i} L_j^T R_j L_j^T \right) \geq 0 \), then (100) and (101) for \( V_i^l, l \) yield
\[ V_i^1(\epsilon_{ik}) - Z_i^0(\epsilon_{ik}) = \frac{1}{2} \left( \epsilon_i^T Q_i \epsilon_{ik} + L_i^T R_i L_i^T + \sum_{j \in N_i} L_j^T R_j L_j^T \right) \geq 0 \tag{102} \]

or
\[ Z_i^0(\epsilon_{ik}) \leq V_i^1(\epsilon_{ik}). \tag{103} \]

Now we need to assume that \( Z_i^l(\epsilon_{ik}) \leq V_i^{l+1}(\epsilon_{ik}) \) holds for \( l = 1 \) such that \( Z_i^{l-1}(\epsilon_{ik}) \leq V_i^l(\epsilon_{ik}), \forall l \). Then at (step l) (100) and (101) yield
\[ V_i^{l+1}(\epsilon_{ik}) - Z_i^l(\epsilon_{ik}) = V_i^l(\epsilon_{ik}) - Z_i^{l-1}(\epsilon_{ik}) \geq 0. \tag{104} \]

Therefore the mathematical induction for \( \forall l \) yields
\[ Z_i^l(\epsilon_{ik}) \leq V_i^{l+1}(\epsilon_{ik}), \quad \forall l. \tag{105} \]

Consequently, for all the policies \( l_i^l, l_j^l, \forall l \) the following monotonic sequence holds
\[ V_i^{l+1} > Z_i^l > V_i^l > \cdots > 0. \tag{106} \]

Since \( Z_i^l(\epsilon_{ik}) \) is the lower bound of \( V_i^{l+1}(\epsilon_{ik}) \) and Lemma 9 sets \( \overline{U} \) as an upper value of \( V_i^{l+1} \), then (99) and (106) yield
\[ 0 \leq V_i^l \leq Z_i^l \leq V_i^{l+1} \leq \overline{U}. \tag{107} \]

From (107) the sequence of \( V_i^l \) is increasing and it has an upper bound \( \overline{U} \), which means that \( V_i^l \) will converge to the optimal solution \( V_i^\ast \) monotonically by satisfying
\[ \tilde{V}_i^\ast(\epsilon_{ik}) = \frac{1}{2} \left( \epsilon_i^T Q_i \epsilon_{ik} + \tilde{L}_i^T R_i \tilde{L}_i^T + \sum_{j \in N_i} \tilde{L}_j^T R_j \tilde{L}_j^T \right) + \tilde{V}_i^\ast(\epsilon_{ik+1}) \tag{108} \]

where
\[ \tilde{L}_i^l = (d_i + g_i) R_i^{-1} B_i \nabla_i V_i^\ast(\epsilon_{ik+1}). \tag{109} \]

**Remark 2.** The condition \( \sigma (R_i^{-1} R_j) \) small means that agent \( j \) weights his own control effort in his value function higher than his out-neighbors weight his control effort in their value functions. This can be guaranteed for any choice of \( R_j \) for agent \( i \) by selecting \( R_j \) large enough. This is related to the weakly coupling condition defined in Chapter 6 of Başar and Olsder (1999). ■
6. Graphical game solutions by actor–critic learning

This section develops an actor–critic framework based on value function approximation which will be used to solve the dynamic graphical game online. This framework is motivated by Algorithm 1. Each agent $i$ has its own critic network to perform the value update (81) and an actor network to perform the policy improvement (82). The actor–critic network structures depend only on local information.

6.1. Actor–critic networks and tuning

The value function $V_i(\epsilon_{i(k+1)})$ for each agent $i$ is approximated by a critic network $\tilde{V}_i(\cdot | \tilde{W}_i)$, and the control policy is approximated by an actor network $\tilde{\mu}_i(\cdot | \tilde{W}_i)$ so that

\[
\tilde{V}_i(\tilde{W}_i) = \tilde{Z}_i^T \tilde{W}_i^T Z_i \tag{110}
\]

\[
\tilde{\mu}_i(\tilde{W}_i) = \tilde{W}_i^T Z_i \tag{111}
\]

where $\tilde{W}_i \in \mathbb{R}^{n_i \times n_i}$ and $\tilde{W}_i \in \mathbb{R}^{n_i \times m_i}$ are the critic and actor weights respectively. $N_{ij}$ is the total number of each agent $i$ and its neighbors. $Z_i \in \mathbb{R}^{N_{ij}}$ is a vector of the state $x_i$ of agent $i$ and the states of its neighbors.

Let $\epsilon_{i(k+1)}$ be the approximation error of the actor network so that

\[
\epsilon_{i(k+1)} = \tilde{u}_i(\tilde{W}_i) - \hat{u}_i = \tilde{W}_i^T Z_i - \hat{u}_i \tag{112}
\]

where, based on (82), the target control policy $\tilde{u}_i$ is given in terms of the critic network such that

\[
\tilde{u}_i = (g_i + d_i) \tilde{u}_i^T \nabla \tilde{V}_i(\epsilon_{i(k+1)}) \tag{113}
\]

This target control policy $\tilde{u}_i$ can be expressed in terms of the critic network weights $\tilde{W}_i$ such that

\[
\tilde{u}_i = (g_i + d_i) \tilde{u}_i^T \nabla \tilde{V}_i(\epsilon_{i(k+1)}) \tag{114}
\]

where $O_i = 2 \times [0 \ldots [1]_i \ldots 0] \in \mathbb{R}^{n \times n_i}$.

The squared approximation error is

\[
\text{err}_{\text{actor}} = \frac{1}{2} (\epsilon_{i(k+1)}^i)^T (\epsilon_{i(k+1)}^i) \tag{115}
\]

The change in the actor weights is given by the gradient descent. The update rule for the actor weights is therefore given by

\[
\tilde{W}_i^{(l+1)} = \tilde{W}_i^l - \mu_\text{c} \epsilon_{i(k+1)}^i \tag{116}
\]

where $0 < \mu_\text{c} < 1$ is the actor network learning rate.

The value update equation is given by (81). Let $\gamma_i^{(l+1)}$ be the target value of the critic network at step $l$ such that

\[
\gamma_i^{(l+1)} = \frac{1}{2} \left( \epsilon_i^T \tilde{Q} \epsilon_i + \tilde{u}_i^T R_i \tilde{u}_i + \sum_{j \neq i} \tilde{u}_j^T R_{ij} \tilde{u}_j \right) + \tilde{V}_i^{(l+1)} \tag{117}
\]

The critic network approximation error at step $l$ is given by

\[
\epsilon_i^{(l+1)} = \gamma_i^{(l+1)} - \tilde{V}_i(\tilde{W}_i) \tag{118}
\]

Similarly, we define squared approximation error for the critic as

\[
\text{err}_{\text{critic}} = \frac{1}{2} (\epsilon_i^{(l+1)})^T (\epsilon_i^{(l+1)}) = \frac{1}{2} \left\| \epsilon_i^{(l+1)} - Z_i \tilde{W}_i^T Z_i \right\|_2 \tag{119}
\]

By employing gradient descent, the update rule for the critic weights is given by

\[
\tilde{W}_i^{(l+1)} = \tilde{W}_i^l - \mu_\text{c} \epsilon_i^{(l+1)} \tag{120}
\]

where $0 < \mu_\text{c} < 1$ is the critic network learning rate.

6.2. Actor–critic online tuning in real-time

The following algorithm is used for online tuning of the actor–critic networks in real-time using data measured along the system trajectories.

Algorithm 2 (Actor–Critic Network Online Tuning).

1. Initialize the actor weights $\tilde{W}_a$ randomly and initialize the critic weights $\tilde{W}_c$ with zero values.
2. Do Loop (l iterations) 
   2.1 Start with initial state $x_0$ on the system trajectory.
   2.2 Calculate $\tilde{u}_i$ using (111).
   2.3 Calculate the dynamics $\dot{x}_i$ using (9).
   2.4 Calculate the value $\tilde{V}_i^{(l+1)}$ using (110).
   2.5 Critic update rule
   \[
   \tilde{W}_c^{(l+1)} = \tilde{W}_c^l - \mu_\text{c} \epsilon_i^{(l+1)} (\tilde{V}_i^{(l+1)} - Z_i \tilde{W}_c^l Z_i)^T \tag{117}
   \]
   where $\gamma_i^{(l+1)}$ is given by (117).
   2.6 Actor update rule
   \[
   \tilde{W}_a^{(l+1)} = \tilde{W}_a^l + \mu_\text{c} (\tilde{W}_c^l Z_i - \tilde{u}_i^T Z_i)^T \tag{117}
   \]
   where $\epsilon_i^{(l+1)}$ is given by (117).
   2.7 On convergence of $\| \gamma_i^{(l+1)} - \epsilon_i^{(l+1)} \|$ End Loop.

Remark 3. Algorithm 2 uses gradient descent to tune the weights of the critic and actor networks at each iteration. Assuming that the gradient descent algorithms converge exactly at each iteration, then Algorithm 2 at each step solves the Bellman equation (81). Then, Theorem 7 proves convergence of Algorithm 1. Unfortunately, gradient descent cannot always be guaranteed to converge to the exact solutions in approximation structures. However, simulations have shown the effectiveness of this algorithm.

6.3. Graphical game example and simulation results

The graphical game can be solved online in real-time by using Algorithm 2. In this section we perform simulations to verify the theoretical developments. Consider the directed graph with four agents shown in Fig. 1.

The plant and input matrices for every agent are given as,

\[
A = \begin{bmatrix} 0.995 & 0.09983 \\ -0.09983 & 0.995 \end{bmatrix},
\]

\[
B_1 = \begin{bmatrix} 0.2047 \\ 0.08984 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0.2147 \\ 0.2895 \end{bmatrix},
\]

\[
B_3 = \begin{bmatrix} 0.2097 \\ 0.1897 \end{bmatrix}, \quad B_4 = \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix}
\]

with pinning gains given by $g_1 = g_2 = g_3 = 0, g_4 = 1$, edge weights given by $e_{12} = 0.8, e_{14} = 0.7, e_{23} = 0.6, e_{31} = 0.8$ and the learning rates are selected as $(\mu_\text{c} = 0.1, \mu_\text{c} = 0.1, \mu_i = 0.1, \forall i)$.

The user defined matrices in the performance indices are selected to be,

\[
Q_{11} = Q_{22} = Q_{33} = Q_{44} = I_{2 \times 2}, \quad R_{11} = R_{22} = R_{33} = R_{44} = 1, \quad R_{13} = R_{21} = R_{32} = R_{41} = 0,
\]

\[
R_{12} = R_{14} = R_{23} = R_{31} = 1.
\]
This paper introduces a new class of discrete-time dynamic games known as dynamic graphical games. It brings together discrete Hamiltonian mechanics, distributed multi-agent control, optimal control theory, and game theory to formulate and solve these dynamic graphical games. The relation between the Bellman optimality equation and the discrete-time HJB equation for the graphical game is shown. It is shown that standard notions of Nash equilibrium are insufficient to guarantee that all agents are involved in the same game and for that reason a new notion of interactive Nash equilibrium is introduced which holds if the agents are all in Nash and the graph is strongly connected. A value iteration algorithm is proposed to solve the dynamic graphical games. Based on this algorithm, a real adaptive learning structure is developed to solve the dynamic graphical game in real-time. Simulation results show the effectiveness of the proposed structure.

Appendix

The next two technical lemmas are required to prove Theorem 6. They are motivated by Al-Tamimi et al. (2008), and Lancaster and Rodman (1995).

Fig. 2. Critic–actor weights update of Agent (1) versus iteration steps.

Fig. 3. Tracking error and agents’ states versus iteration steps.

Fig. 4. Phase plane plots of the agents that show that the agents are being synchronized.

**Lemma 6.** Let the neighbors of agent \( i \) have fixed policies \( u_j \). Given arbitrary stabilizing control policies \( \{ M_i^{l} \}_{l=0}^{\infty} \) for every agent \( i \), let the associated value sequence be \( \{ V_i^{l} \}_{l=0}^{\infty} \). Define the sequence of control policies generated by Algorithm 1 as \( \{ l_i^{l} \}_{l=0}^{\infty} \) with associated value sequence \( \{ V_i^{l} \}_{l=0}^{\infty} \). Then starting with \( 0 \leq V_i^{l_{0}} \leq Z_i^{l_{0}} \) one has

\[
0 \leq V_i^{l_{1}} \leq Z_i^{l_{1}} \quad \text{(121)}
\]

**Proof.** The value function sequences for each agent \( i \) are given by (84) and (85), the arbitrary stabilizing control sequence for each agent \( i \) can be written as \( M_i^{l_{1}} = (l_i^{l_{1}} + (M_i^{l_{1}} - l_i^{l_{1}})) \). The arbitrary value sequence \( \{ Z_i^{l_{1}} \}_{l=0}^{\infty} \) for each agent \( i \) is given as

\[
Z_i^{l_{1}+1}(\epsilon_{ik}) \equiv F_i(Z_i^{l_{1}}, M_i^{l_{1}})
\]

\[
= Z_i^{l_{1}+1}|_{M_i^{l_{1}}} + \frac{1}{2}(e_{ik}^{T}Q_{ik}e_{ik} + Z_i^{l_{1}}(R_{ik}, u_{ik}))
\]

\[
+ \frac{1}{2}(l_i^{l_{1}} + (M_i^{l_{1}} - l_i^{l_{1}}))^{T}R_{ik}(l_i^{l_{1}} + (M_i^{l_{1}} - l_i^{l_{1}}))
\]

where \( V_i^{l_{1}}(R_{ik}, u_{ik}) = Z_i^{l_{1}}(R_{ik}, u_{ik}) = \sum_{j \in N_{i}} u_{ij}^{T} R_{ij} u_{ij} \).

Rearranging this equation yields

\[
Z_i^{l_{1}+1}(\epsilon_{ik}) \equiv F_i(Z_i^{l_{1}}, l_i^{l_{1}}) + \frac{1}{2}(M_i^{l_{1}} - l_i^{l_{1}})^{T}R_{ik}(M_i^{l_{1}} - l_i^{l_{1}})
\]

\[
+ (Z_i^{l_{1}}|_{M_i^{l_{1}}} + M_i^{l_{1}} R_{ik} l_i^{l_{1}}) - (Z_i^{l_{1}}(\epsilon_{ik}^{l_{1}+1}) + l_i^{l_{1}} R_{ik} l_i^{l_{1}}) \quad \text{(122)}
\]
where,
\[ F_i(Z_i^1, L_i^1) = \frac{1}{2} (\epsilon_{ik}^T Q_i \epsilon_{ik} + L_i^1 R_i u_{i-1} + Z_i^1 (R_{ij}, u_{i-1}^j)) + Z_i^1 (\epsilon_{i(i+1)} | l_i^1) . \]

By induction,
\[ \frac{1}{2} (M_i^1 - L_i^1)^T R_i (M_i^1 - L_i^1) + (Z_i^1 (\epsilon_{i(i+1)} | m_i^1) + M_i^{1T} R_i L_i) \]
\[ = (Z_i^1 (\epsilon_{i(i+1)} | l_i^1) + L_i^1 R_i L_i^1) > 0. \] (123)

Then (122) and the induction result (123) yield,
\[ \tilde{Z}_i^{i+1} (\epsilon_{ik}) = F_i(Z_i^1, M_i^1) > \tilde{Z}_i^{i+1} = F_i(Z_i^1, L_i^1). \] (124)

Similarly,
\[ V_i^{i+1}(\epsilon_{ik}) = F_i(V_i^1, M_i^1) \geq V_i^{i+1}(\epsilon_{ik}) = F_i(V_i^1, L_i^1). \] (125)

Using the initial sequence 0 ≤ \( V_i^0 \) ≤ \( Z_i^0 \), which by applying induction becomes 0 ≤ \( V_i^t \) ≤ \( Z_i^t \). Inequality (124) gives the lower bound on the average value sequence \( Z_i^{i+1}(\epsilon_{ik}) \) such that
\[ 0 \leq V_i^t \leq Z_i^t. \] (127)

Proof. \( M_i \) is stabilizing control policy for each agent \( i \). Using (98) and the sequence \( V_i^0 = Z_i^0 = 0 \) yields
\[ Z_i^{i+1}(\epsilon_{ik}) - Z_i^i(\epsilon_{ik}) = F_i(Z_i^1, M_i^1) - F_i(Z_i^1, M_i^1) \]
\[ = Z_i^1(\epsilon_{i(i+1)} | m_i^1) \frac{1}{2} (\epsilon_{ik}^T Q_i \epsilon_{ik} + M_i^{1T} R_i M_i^1)
+ Z_i^1 (R_{ij}, u_{i-1}^j) - Z_i^{i+1}(\epsilon_{i(i+1)} | m_i^1)
- \frac{1}{2} (\epsilon_{ik}^T Q_i \epsilon_{ik} + M_i^{1T} R_i M_i^1)
+ Z_i^{i+1}(\epsilon_{i(i+1)} | m_i^1)
= Z_i^1(\epsilon_{i(i+1)} | m_i^1) - Z_i^1(\epsilon_{i(i+1)} | m_i^1). \] (129)

Rearranging (129) yields,
\[ Z_i^{i+1}(\epsilon_{ik}) - Z_i^i(\epsilon_{ik}) = Z_i^1(\epsilon_{i(i+1)} | m_i^1) - Z_i^{i+1}(\epsilon_{i(i+1)} | m_i^1) 
= Z_i^{i+1}(\epsilon_{i(i+1)} | m_i^1) - Z_i^{i+1}(\epsilon_{i(i+1)} | m_i^1) \]
\[ = Z_i^1(\epsilon_{i(i+1)} | m_i^1) - Z_i^1(\epsilon_{i(i+1)} | m_i^1). \] (130)

with \( Z_i^0(\epsilon_{i(i+1)} | m_i^1) = 0 \). Eq. (130) yields,
\[ Z_i^{i+1}(\epsilon_{ik}) = Z_i^1(\epsilon_{i(i+1)} | m_i^1) + Z_i^1(\epsilon_{ik}) = Z_i^1(\epsilon_{i(i+1)} | m_i^1) + Z_i^1(\epsilon_{i(i+1)} | m_i^1) + Z_i^1(\epsilon_{ik}) \] (131)

Then (131) is rearranged so that
\[ Z_i^{i+1}(\epsilon_{ik}) = \sum_{n=0}^{l_i^1} Z_i^{i+1}(\epsilon_{i(i+n)} | m_i^1) + \frac{1}{2} \sum_{n=0}^{l_i^1} (\epsilon_{i(i+n)}^T Q_i \epsilon_{i(i+n)} | m_i^1) \]
\[ + \sum_{n=0}^{l_i^1} (\epsilon_{i(i+n)}^T Q_i \epsilon_{i(i+n)} | m_i^1) R_i u_{i-1}^j) + Z_i^1 (R_{ij}, u_{i-1}^j). \] (132)

Since the used policies are stabilizable policies, then
\[ Z_i^{i+1}(\epsilon_{ik}) \geq \sum_{n=0}^{\infty} Z_i^{i+1}(\epsilon_{i(i+n)} | m_i^1) = \bar{U}. \] (133)

This inequality satisfies (128). ■

Lemma 7. Let the neighbors of agent \( i \) have fixed policies \( u_r \). Define the sequence of control policies generated by Algorithm 1 for each agent \( i \) as \( [L_i^1]_{i=0}^{\infty} \) with associated value sequence \( [V_i^1]_{i=0}^{\infty} \). Then, there exists an upper bound \( U \) such that
\[ 0 < V_i^t \leq U. \] (128)

Proof. \( M_i \) is stabilizing control policy for each agent \( i \). Using (98) and the sequence \( V_i^0 = Z_i^0 = 0 \) yields
\[ Z_i^{i+1}(\epsilon_{ik}) - Z_i^i(\epsilon_{ik}) = F_i(Z_i^1, M_i^1) - F_i(Z_i^1, M_i^1) \]
\[ = Z_i^1(\epsilon_{i(i+1)} | m_i^1) \frac{1}{2} (\epsilon_{ik}^T Q_i \epsilon_{ik} + M_i^{1T} R_i M_i^1)
+ Z_i^1 (R_{ij}, u_{i-1}^j) - Z_i^{i+1}(\epsilon_{i(i+1)} | m_i^1)
- \frac{1}{2} (\epsilon_{ik}^T Q_i \epsilon_{ik} + M_i^{1T} R_i M_i^1)
+ Z_i^{i+1}(\epsilon_{i(i+1)} | m_i^1)
= Z_i^1(\epsilon_{i(i+1)} | m_i^1) - Z_i^1(\epsilon_{i(i+1)} | m_i^1). \] (129)

where \( Z_i^1 (R_{ij}, u_{i-1}^j) = Z_i^{i}(R_{ij}, u_{i-1}^j) \). Rearranging (129) yields,
\[ Z_i^{i+1}(\epsilon_{ik}) - Z_i^i(\epsilon_{ik}) = Z_i^1(\epsilon_{i(i+1)} | m_i^1) - Z_i^{i+1}(\epsilon_{i(i+1)} | m_i^1) 
= Z_i^{i+1}(\epsilon_{i(i+1)} | m_i^1) - Z_i^{i+1}(\epsilon_{i(i+1)} | m_i^1) \]
\[ = Z_i^1(\epsilon_{i(i+1)} | m_i^1) - Z_i^1(\epsilon_{i(i+1)} | m_i^1). \] (130)

with \( Z_i^0(\epsilon_{i(i+1)} | m_i^1) = 0 \). Eq. (130) yields,
\[ Z_i^{i+1}(\epsilon_{ik}) = Z_i^1(\epsilon_{i(i+1)} | m_i^1) + Z_i^1(\epsilon_{ik}) \]
\[ = Z_i^1(\epsilon_{i(i+1)} | m_i^1) + Z_i^1(\epsilon_{i(i+1)} | m_i^1) + Z_i^{i+1}(\epsilon_{ik}) \]
\[ = Z_i^1(\epsilon_{i(i+1)} | m_i^1) + Z_i^1(\epsilon_{i(i+1)} | m_i^1) + + Z_i^1(\epsilon_{ik}). \] (131)

Then (131) is rearranged so that
\[ Z_i^{i+1}(\epsilon_{ik}) = \sum_{n=0}^{l_i^1} Z_i^{i+1}(\epsilon_{i(i+n)} | m_i^1) \]
\[ + \frac{1}{2} (\epsilon_{i(i+n)}^T Q_i \epsilon_{i(i+n)} | m_i^1) \]
\[ + \sum_{n=0}^{l_i^1} (\epsilon_{i(i+n)}^T Q_i \epsilon_{i(i+n)} | m_i^1) R_i u_{i-1}^j) + Z_i^1 (R_{ij}, u_{i-1}^j). \] (132)

Since the used policies are stabilizable policies, then
\[ Z_i^{i+1}(\epsilon_{ik}) \leq \sum_{n=0}^{\infty} Z_i^{i+1}(\epsilon_{i(i+n)} | m_i^1) = \bar{U}. \] (133)

This inequality satisfies (128). ■
Since the following inequality is true,
\[
Z_i^l(\epsilon_{ik+1} | \theta_i | M_i^l) + M_i^l R_i L_i^l = Z_i^l(\epsilon_{ik+1} | \theta_i | L_i^l) + L_i^l R_i L_i^l
\]
and using the condition
\[
\frac{1}{2} \sum_{n \in N} (M_i^l - L_i^l)^T R_n (M_i^l - L_i^l) > 0
\]
(137) becomes
\[
\sum_{n \in N} \left\| R_n \right\| \left\| \Delta E_{ik} \right\| \geq \sum_{n \in N} \left( g_i + d_i \right) \sigma_i (R_{ij}^{-1} R_n) \left\| \nabla V_i^l(\epsilon_{ik+1}) \right\| \left\| B_k \right\|
\]
where
\[
\left\| \Delta E_{ik} \right\| = \left\| L_i^l - M_i^l \right\|.
\]
By considering (137) and under the assumption that \( \sigma_i (R_{ij}^{-1} R_n) \) is small then (139) yields
\[
Z_i^{l+1}(\epsilon_{ik}) = F_i (Z_i^l, M_i^l, M_i^l) \geq Z_i^{l+1}(\epsilon_{ik}) = F_i (Z_i^l, L_i^l, L_i^l).
\]
Similarly we have,
\[
V_i^{l+1}(\epsilon_{ik}) = F_i (V_i^l, M_i^l, M_i^l) \geq Z_i^{l+1}(\epsilon_{ik}) = F_i (Z_i^l, L_i^l, L_i^l).
\]
(140)
By setting the initial sequence \( 0 \leq V_i^0 \leq Z_i^2 \), and after applying induction we have \( 0 \leq V_i^l \leq Z_i^2 \). The inequality (139) gives a lower bound on the arbitrary value sequence \( Z_i^{l+1}(\epsilon_{ik}) \) such that
\[
0 \leq V_i^{l+1}(\epsilon_{ik}) = F_i (V_i^l, M_i^l, M_i^l) \leq Z_i^{l+1}(\epsilon_{ik}) = F_i (Z_i^l, L_i^l, L_i^l).
\]
(141)
Then finally we have that,
\[
0 \leq V_i^l \leq Z_i^l \leq Z_i^2.
\]
(142)
Eq. (142) yields (134).

**Lemma 9.** Define the sequences of control policies generated by Algorithm 1 for agent \( i \) and its neighbor \( j \) as \( \{V_i^l\}_{l=0}^{\infty} \) and \( \{Z_i^l\}_{l=0}^{\infty} \) respectively, with associated value sequences \( \{V_i^l\}_{l=0}^{\infty} \). Suppose that \( \sigma (R_{ij}^{-1} R_n) \) is small. Then there exists a finite upper bound \( \overline{V} \) such that
\[
0 \leq V_i^l \leq \overline{V}.
\]
(143)

**Proof.** The policies \( M_i \) and \( M_j \) are stabilizing control policies for each agent \( i \) and its neighbor \( j \). Using (98) and the sequence \( V_i^0 = Z_i^0 = 0 \) we have
\[
Z_i^{l+1}(\epsilon_{ik}) - Z_i^l(\epsilon_{ik}) = Z_i^l(\epsilon_{ik+1} | \theta_i | M_i^l) - Z_i^l(\epsilon_{ik+1} | \theta_i | M_i^l)
\]
\[
= Z_i^{l+1}(\epsilon_{ik+1} | \theta_i | M_i^l) - Z_i^{l+2}(\epsilon_{ik+2} | \theta_i | M_i^l)
\]
\[
= Z_i^{l+1}(\epsilon_{ik+1} | \theta_i | M_i^l) - Z_i^0(\epsilon_{ik+1} | \theta_i | M_i^l)
\]
(144)
with \( Z_i^0(\epsilon_{ik+1} | \theta_i | M_i^l) = 0 \).

By doing some manipulations in (144) we have
\[
Z_i^{l+1}(\epsilon_{ik}) = Z_i^l(\epsilon_{ik+1} | \theta_i | M_i^l) + Z_i^l(\epsilon_{ik})
\]
\[
= Z_i^l(\epsilon_{ik+1} | \theta_i | M_i^l) + Z_i^l(\epsilon_{ik+1} | \theta_i | M_i^l) + Z_i^{l+1}(\epsilon_{ik})
\]
\[
= Z_i^l(\epsilon_{ik+1} | \theta_i | M_i^l) + Z_i^l(\epsilon_{ik+1} | \theta_i | M_i^l) + Z_i^l(\epsilon_{ik+1} | \theta_i | M_i^l) + \ldots + Z_i^l(\epsilon_{ik}).
\]
(145)
Finally (145) can be written as
\[
Z_i^{l+1}(\epsilon_{ik}) = \sum_{n=0}^{l} Z_i^l(\epsilon_{ik+n} | \theta_i | M_i^l)
\]
\[
= \sum_{n=0}^{l} \left( \sum_{j \in N} \left( g_i + d_i \right) \sigma_i (R_{ij}^{-1} R_n) \left\| \nabla V_i^l(\epsilon_{ik+n}) \right\| \left\| B_k \right\| \right)
\]
(146)
and since the used policies are stabilizable we have
\[
Z_i^{l+1}(\epsilon_{ik}) \leq \sum_{n=0}^{\infty} Z_i^l(\epsilon_{ik+n} | \theta_i | M_i^l) = \overline{V}.
\]
(147)
This inequality satisfies (143) and hence the result follows.

**References**


Learning of the IEEE Computational Intelligence Society (ADPRLTC), an Associate Editor on the IEEE Control Systems Society Conference Editorial Board, a registered Electrical/Computer engineer (PE) and a member of the Technical Chamber of Greece.

**Sofie Haesaert** was born in Leuven, Belgium. She obtained the Bachelor’s Degree in Mechanical Engineering in 2010 with distinction and the Master of Science Degree in Systems and Control in 2012 with distinction in combination with the Masters Honors Program at Delft University of Technology. She is a recipient of a Ph.D. grant from the Dutch Institute of Systems and Control (DISC) in the scope of the Graduate Program of The Netherlands Organization for Scientific Research (NOW). Currently, she is working towards the Ph.D. degree in the Control Systems group of the Electrical Engineering Department at Eindhoven University of Technology. Her current research interests include identification for control, Bayesian identification, machine learning, formal verification and correct-by-design controller synthesis.

**Robert Babuska** received his M.Sc. degree (with honors) in Electrical Engineering from the Czech Technical University Prague (1990), and his Ph.D. degree (cum laude) in Control from the Delft University of Technology (1997). He has had faculty appointments at the Technical Cybernetics Department of the Czech Technical University Prague (1991–1993) and at the Faculty of Information Technology and Systems of the Delft University of Technology (1993–2003). Currently, he is professor of Intelligent Control and Robotics at the Delft Center for Systems and Control, Delft University of Technology. He is the founder and scientific director of the TU Delft Robotics Institute. His research interests include adaptive and learning control, nonlinear identification and state-estimation, dynamic multi-agent systems, predictive control, neural networks, fuzzy systems and machine learning. Robert Babuska has co-authored over 390 publications, including three research monographs, three edited books and 85 journal papers. He has been serving as Associate Editor of several archived journals including Automatica, Engineering Applications of Artificial Intelligence, and IEEE Transactions on Fuzzy Systems. He served the chairman of IFAC (International Federation of Automatic Control) technical committee on Cognition and Control.