2

Online Adaptive Learning Solution of Multi-Agent Differential Graphical Games

Kyriakos G. Vamvoudakis[1] and Frank L. Lewis[2]
[1]*Center for Control, Dynamical-Systems, and Computation (CCDC), University of California, Santa Barbara,*
[2]*Automation and Robotics Research Institute, The University of Texas at Arlington, USA*

1. Introduction

Distributed networks have received much attention in the last year because of their flexibility and computational performance. The ability to coordinate agents is important in many real-world tasks where it is necessary for agents to exchange information with each other. Synchronization behavior among agents is found in flocking of birds, schooling of fish, and other natural systems. Work has been done to develop cooperative control methods for consensus and synchronization (Fax and Murray, 2004; Jadbabaie, Lin and Morse, 2003; Olfati-Saber, and Murray, 2004; Qu, 2009; Ren, Beard, and Atkins, 2005; Ren, and beard, 2005; Ren, and Beard, 2008; Tsitsiklis, 1984). See (Olfati-Saber, Fax, and Murray, 2007; Ren, Beard, and Atkins, 2005) for surveys. Leaderless consensus results in all nodes converging to common value that cannot generally be controlled. We call this the cooperative regulator problem. On the other hand the problem of cooperative tracking requires that all nodes synchronize to a leader or control node (Hong, Hu, and Gao, 2006; Li, Wang, and Chen, 2004; Ren, Moore, and Chen, 2007; Wang, and Chen, 2002). This has been called pinning control or control with a virtual leader. Consensus has been studied for systems on communication graphs with fixed or varying topologies and communication delays.

Game theory provides an ideal environment in which to study multi-player decision and control problems, and offers a wide range of challenging and engaging problems. Game theory (Tijs, 2003) has been successful in modeling strategic behavior, where the outcome for each player depends on the actions of himself and all the other players. Every player chooses a control to minimize independently from the others his own performance objective. Multi player cooperative games rely on solving coupled Hamilton-Jacobi (HJ) equations, which in the linear quadratic case reduce to the coupled algebraic Riccati equations (Basar, and Olsder, 1999; Freiling, Jank, and Abou-Kandil, 2002; Gajic, and Li, 1988). Solution methods are generally offline and generate fixed control policies that are then implemented in online controllers in real time. These coupled equations are difficult to solve.

Reinforcement learning (RL) is a sub-area of machine learning concerned with how to methodically modify the actions of an agent (player) based on observed responses from its environment (Sutton, and Barto, 1998). RL methods have allowed control systems researchers to develop algorithms to learn online in real time the solutions to optimal control problems for dynamic systems that are described by difference or ordinary differential equations. These involve a computational intelligence technique known as Policy Iteration (PI) (Bertsekas, and Tsitsiklis, 1996), which refers to a class of algorithms with two steps, *policy evaluation* and *policy improvement*. PI has primarily been developed for discrete-time systems, and online implementation for control systems has been developed through approximation of the value function (Bertsekas, and Tsitsiklis, 1996; Werbos, 1974; Werbos, 1992). PI provides effective means of learning solutions to HJ equations online. In control theoretic terms, the PI algorithm amounts to learning the solution to a nonlinear Lyapunov equation, and then updating the policy through minimizing a Hamiltonian function. Policy Iteration techniques have been developed for continuous-time systems in (Vrabie, Pastravanu, Lewis, and Abu-Khalaf, 2009).

RL methods have been used to solve multiplayer games for finite-state systems in (Busoniu, Babuska, and De Schutter, 2008; Littman, 2001). RL methods have been applied to learn online in real-time the solutions for optimal control problems for dynamic systems and differential games in (Dierks, and Jagannathan, 2010; Johnson, Hiramatsu, Fitz-Coy, and Dixon, 2010; Vamvoudakis 2010; Vamvoudakis 2011).

This book chapter brings together cooperative control, reinforcement learning, and game theory to solve multi-player differential games on communication graph topologies. There are four main contributions in this chapter. The first involves the formulation of a *graphical game* for dynamical systems networked by a communication graph. The dynamics and value function of each node depend only on the actions of that node and its neighbors. This graphical game allows for synchronization as well as Nash equilibrium solutions among neighbors. It is shown that standard definitions for Nash equilibrium are not sufficient for graphical games and a new definition of "Interactive Nash Equilibrium" is given. The second contribution is the derivation of coupled Riccati equations for solution of graphical games. The third contribution is a Policy Iteration algorithm for solution of graphical games that relies only on local information from neighbor nodes. It is shown that this algorithm converges to the best response policy of a node if its neighbors have fixed policies, and to the Nash solution if all nodes update their policies. The last contribution is the development of an online adaptive learning algorithm for computing the Nash equilibrium solutions of graphical games.

The book chapter is organized as follows. Section 2 reviews synchronization in graphs and derives an error dynamics for each node that is influenced by its own actions and those of its neighbors. Section 3 introduces differential graphical games cooperative Nash equilibrium. Coupled Riccati equations are developed and stability and solution for Nash equilibrium are proven. Section 4 proposes a policy iteration algorithm for the solution of graphical games and gives proofs of convergence. Section 5 presents an online adaptive learning solution based on the structure of the policy iteration algorithm of Section 4. Finally Section 6 presents a simulation example that shows the effectiveness of the proposed algorithms in learning in real-time the solutions of graphical games.

2. Synchronization and node error dynamics

2.1 Graphs

Consider a graph $G=(V,\mathrm{E})$ with a nonempty finite set of N nodes $V=\{v_1,\cdots,v_N\}$ and a set of edges or arcs $\mathrm{E}\subseteq V\times V$. We assume the graph is simple, e.g. no repeated edges and $(v_i,v_i)\notin E,\forall i$ no self loops. Denote the connectivity matrix as $E=[e_{ij}]$ with $e_{ij}>0$ *if* $(v_j,v_i)\in\mathrm{E}$ and $e_{ij}=0$ otherwise. Note $e_{ii}=0$. The set of neighbors of a node v_i is $N_i=\{v_j:(v_j,v_i)\in\mathrm{E}\}$, i.e. the set of nodes with arcs incoming to v_i. Define the in-degree matrix as a diagonal matrix $D=diag(d_i)$ with $d_i=\sum_{j\in N_i}e_{ij}$ the weighted in-degree of node i (i.e. i *-th* row sum of E). Define the graph Laplacian matrix as $L=D-E$, which has all row sums equal to zero.

A directed path is a sequence of nodes $v_0,v_1,\cdots,v_r$ such that $(v_i,v_{i+1})\in E, i\in\{0,1,\cdots,r-1\}$. A directed graph is strongly connected if there is a directed path from v_i to v_j for all distinct nodes $v_i,v_j\in V$. A (directed) tree is a connected digraph where every node except one, called the root, has in-degree equal to one. A graph is said to have a spanning tree if a subset of the edges forms a directed tree. A strongly connected digraph contains a spanning tree.

General directed graphs with fixed topology are considered in this chapter.

2.2 Synchronization and node error dynamics

Consider the N systems or agents distributed on communication graph G with node dynamics

$$\dot{x}_i=Ax_i+B_iu_i \tag{1}$$

where $x_i(t)\in\mathbb{R}^n$ is the state of node i, $u_i(t)\in\mathbb{R}^{m_i}$ its control input. Cooperative team objectives may be prescribed in terms of the *local neighborhood tracking error* $\delta_i\in\mathbb{R}^n$ (Khoo, Xie, and Man, 2009) as

$$\delta_i=\sum_{j\in N_i}e_{ij}(x_i-x_j)\quad+g_i(x_i-x_0) \tag{2}$$

The pinning gain $g_i\geq 0$ is nonzero for a small number of nodes i that are coupled directly to the leader or control node x_0, and $g_i>0$ for at least one i (Li, Wang, and Chen, 2004). We refer to the nodes i for which $g_i\neq 0$ as the pinned or controlled nodes. Note that δ_i represents the information available to node i for state feedback purposes as dictated by the graph structure.

The state of the control or target node is $x_0(t)\in\mathbb{R}^n$ which satisfies the dynamics

$$\dot{x}_0 = Ax_0 \tag{3}$$

Note that this is in fact a *command generator* (Lewis, 1992) and we seek to design a cooperative control command generator tracker. Note that the trajectory generator A may not be stable.

The Synchronization control design problem is to design local control protocols for all the nodes in G to synchronize to the state of the control node, i.e. one requires $x_i(t) \to x_0(t), \forall i$.

From (2), the overall error vector for network Gr is given by

$$\delta = \big((L+G)\otimes I_n\big)(x - \underline{x}_0) = \big((L+G)\otimes I_n\big)\zeta \tag{4}$$

where the global vectors are

$x = \left[x_1^T \quad x_2^T \quad \cdots \quad x_N^T\right]^T \in \mathbb{R}^{nN}$ $\delta = \left[\delta_1^T \quad \delta_2^T \quad \cdots \quad \delta_N^T\right]^T \in \mathbb{R}^{nN}$ and $\underline{x}_0 = \underline{I}x_0 \in \mathbb{R}^{nN}$, with $\underline{I} = \underline{1} \otimes I_n \in R^{nN\times n}$ and $\underline{1}$ the N-vector of ones. The Kronecker product is $\otimes$ (Brewer, 1978). $G \in R^{N\times N}$ is a diagonal matrix with diagonal entries equal to the pinning gains g_i . The (global) consensus or synchronization error (e.g. the disagreement vector in (Olfati-Saber, and Murray, 2004)) is

$$\zeta = \big(x - \underline{x}_0\big) \in \mathbb{R}^{nN} \tag{5}$$

The communication digraph is assumed to be strongly connected. Then, if $g_i \neq 0$ for at least one i , $(L+G)$ is nonsingular with all eigenvalues having positive real parts (Khoo, Xie, and Man, 2009). The next result therefore follows from (4) and the Cauchy Schwartz inequality and the properties of the Kronecker product (Brewer, 1978).

Lemma 1. Let the graph be strongly connected and $G \neq 0$. Then the synchronization error is bounded by

$$\|\zeta\| \leq \|\delta\| / \underline{\sigma}(L+G) \tag{6}$$

with $\underline{\sigma}(L+G)$ the minimum singular value of $(L+G)$, and $\delta(t) \equiv 0$ if and only if the nodes synchronize, that is

$$x(t) = \underline{I}x_0(t) \tag{7}$$

■

Our objective now shall be to make small the local neighborhood tracking errors $\delta_i(t)$, which in view of Lemma 1 will guarantee synchronization.

To find the dynamics of the local neighborhood tracking error, write

$$\dot{\delta}_i = A\delta_i + (d_i + g_i)B_i u_i - \sum_{j\in N_i} e_{ij}B_j u_j \tag{8}$$

with $\delta_i \in \mathbb{R}^n$, $u_i \in \mathbb{R}^{m_i}$, $\forall i$.

This is a dynamical system with multiple control inputs, from node *i* and all of its neighbors.

3. Cooperative multi-player games on graphs

We wish to achieve synchronization while simultaneously optimizing some performance specifications on the agents. To capture this, we intend to use the machinery of multi-player games (Basar, Olsder, 1999). Define $u_{G-i} = \{u_j : j \in N, j \neq i\}$ as the set of policies of all other nodes in the graph other than node *i*. Define $u_{-i}(t)$ as the vector of the control inputs $\{u_j : j \in N_i\}$ of the neighbors of node *i*.

3.1 Cooperative performance index

Define the local performance indices

$$J_i(\delta_i(0), u_i, u_{-i}) = \frac{1}{2}\int_0^\infty (\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii} u_i + \sum_{j\in N_i} u_j^T R_{ij} u_j)\, dt \equiv \frac{1}{2}\int_0^\infty L_i(\delta_i(t), u_i(t), u_{-i}(t))\, dt \qquad (9)$$

where all weighting matrices are constant and symmetric with $Q_{ii} > 0, R_{ii} > 0, R_{ij} \geq 0$. Note that the *i*-th performance index includes only information about the inputs of node *i* and its neighbors.

For dynamics (8) with performance objectives (9), introduce the associated Hamiltonians

$$H_i(\delta_i, p_i, u_i, u_{-i}) \equiv p_i^T \left(A\delta_i + (d_i + g_i)B_i u_i - \sum_{j\in N_i} e_{ij} B_j u_j \right) + \frac{1}{2}\delta_i^T Q_{ii}\delta_i + \frac{1}{2}u_i^T R_{ii} u_i + \frac{1}{2}\sum_{j\in N_i} u_j^T R_{ij} u_j = 0 \qquad (10)$$

where p_i is the costate variable. Necessary conditions (Lewis, and Syrmos, 1995) for a minimum of (9) are (1) and

$$-\dot{p}_i = \frac{\partial H_i}{\partial \delta_i} \equiv A^T p_i + Q_{ii}\delta_i \qquad (11)$$

$$0 = \frac{\partial H_i}{\partial u_i} \Rightarrow u_i = -(d_i + g_i)R_{ii}^{-1} B_i^T p_i \qquad (12)$$

3.2 Graphical games

Interpreting the control inputs u_i, u_j as state dependent policies or strategies, the value function for node *i* corresponding to those policies is

$$V_i(\delta_i(t)) = \frac{1}{2}\int_t^\infty (\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii} u_i + \sum_{j\in N_i} u_j^T R_{ij} u_j)\, dt \qquad (13)$$

Definition 1. Control policies $u_i,\ \forall i$ are defined as admissible if u_i are continuous, $u_i(0)=0$, u_i stabilize systems (8) locally, and values (13) are finite.

When V_i is finite, using Leibniz' formula, a differential equivalent to (13) is given in terms of the Hamiltonian function by the Bellman equation

$$H_i(\delta_i,\frac{\partial V_i}{\partial \delta_i},u_i,u_{-i})\equiv \frac{\partial V_i}{\partial \delta_i}^T\left(A\delta_i+(d_i+g_i)B_iu_i-\sum_{j\in N_i}e_{ij}B_ju_j\right)+\tfrac{1}{2}\delta_i^TQ_{ii}\delta_i+\tfrac{1}{2}u_i^TR_{ii}u_i+\tfrac{1}{2}\sum_{j\in N_i}u_j^TR_{ij}u_j=0 \quad (14)$$

with boundary condition $V_i(0)=0$. (The gradient is disabused here as a column vector.) That is, solution of equation (14) serves as an alternative to evaluating the infinite integral (13) for finding the value associated to the current feedback policies. It is shown in the Proof of Theorem 2 that (14) is a Lyapunov equation. According to (13) and (10) one equates $p_i=\partial V_i/\partial\delta_i$.

The local dynamics (8) and performance indices (9) only depend for each node *i* on its own control actions and those of its neighbors. We call this a *graphical game*. It depends on the topology of the communication graph $G=(V,\mathrm{E})$. We assume throughout the chapter that the game is well-formed in the following sense.

Definition 2. The graphical game with local dynamics (8) and performance indices (9) is well-formed if $B_j\neq 0\rightleftharpoons e_{ij}\in E$, $R_{ij}\neq 0\rightleftharpoons e_{ij}\in E$.

The control objective of agent *i* in the graphical game is to determine

$$V_i^*(\delta_i(t))=\min_{u_i}\int_t^\infty \tfrac{1}{2}(\delta_i^TQ_{ii}\delta_i+u_i^TR_{ii}u_i+\sum_{j\in N_i}u_j^TR_{ij}u_j)\,dt \quad (15)$$

Employing the stationarity condition (12) (Lewis, and Syrmos, 1995) one obtains the control policies

$$u_i=u_i(V_i)\equiv-(d_i+g_i)R_{ii}^{-1}B_i^T\frac{\partial V_i}{\partial\delta_i}\equiv-h_i(p_i) \quad (16)$$

The game defined in (15) corresponds to Nash equilibrium.

Definition 3. (Basar, and Olsder, 1999) **(**Global Nash equilibrium) An *N-tuple* of policies $\{u_1^*,u_2^*,...,u_{\mathrm{N}}^*\}$ is said to constitute a global Nash equilibrium solution for an *N* player game if for all $i\in N$

$$J_i^*\triangleq J_i(u_i^*,u_{G-i}^*)\leq J_i(u_i,u_{G-i}^*) \quad (17)$$

The *N-tuple* of game values $\{J_1^*,J_2^*,...,J_{\mathrm{N}}^*\}$ is known as a Nash equilibrium outcome of the *N*-player game.

The distributed multiplayer graphical game with local dynamics (8) and local performance indices (9) should be contrasted with standard multiplayer games (Abou-Kandil, Freiling, Ionescu, and Jank, 2003; Basar, and Olsder 1999) which have centralized dynamics

$$\dot{z} = Az + \sum_{i=1}^{N} B_i u_i \tag{18}$$

where $z \in \mathbb{R}^n$ is the state, $u_i(t) \in \mathbb{R}^{m_i}$ is the control input for every player, and where the performance index of each player depends on the control inputs of all other players. In the graphical games, by contrast, each node's dynamics and performance index only depends on its own state, its control, and the controls of its immediate neighbors.

It is desired to study the distributed game on a graph defined by (15) with distributed dynamics (8). It is not clear in this scenario how global Nash equilibrium is to be achieved.

Graphical games have been studied in the computational intelligence community (Kakade, Kearns, Langford, and Ortitz, 2003; Kearns, Littman, and Singh, 2001; Shoham, and Leyton-Brown, 2009). A (nondynamic) graphical game has been defined there as a tuple (G, U, v) with $G = (V, E)$ a graph with N nodes, action set $U = U_1 \times \cdots \times U_N$ with U_i the set of actions available to node i, and $v = \begin{bmatrix} v_1 & \cdots & v_N \end{bmatrix}^T$ a payoff vector, with $v_i(U_i, \{U_j : j \in N_i\}) \in R$ the payoff function of node i. It is important to note that *the payoff of node i only depends on its own action and those of its immediate neighbors*. The work on graphical games has focused on developing algorithms to find standard Nash equilibria for payoffs generally given in terms of matrices. Such algorithms are simplified in that they only have complexity on the order of the maximum node degree in the graph, not on the order of the number of players N. Undirected graphs are studied, and it is assumed that the graph is connected.

The intention in this chapter is to provide online real-time adaptive methods for solving differential graphical games that are distributed in nature. That is, the control protocols and adaptive algorithms of each node are allowed to depend only information about itself and its neighbors. Moreover, as the game solution is being learned, all node dynamics are required to be stable, until finally all the nodes synchronize to the state of the control node. These online methods are discussed in Section V.

The following notions are needed in the study of differential graphical games.

Definition 4. (Shoham, and Leyton-Brown, 2009) Agent i's *best response* to fixed policies u_{-i} of his neighbors is the policy u_i^* such that

$$J_i(u_i^*, u_{-i}) \leq J_i(u_i, u_{-i}) \tag{19}$$

for all policies u_i of agent i.

For centralized multi-agent games, where the dynamics is given by (18) and the performance of each agent depends on the actions of all other agents, an equivalent definition of Nash equilibrium is that each agent is in best response to all other agents. In

graphical games, if all agents are in best response to their neighbors, then all agents are in Nash equilibrium, as seen in the proof of Theorem 1.

However, a counterexample shows the problems with the definition of Nash equilibrium in graphical games. Consider the completely disconnected graph with empty edge set where each node has no neighbors. Then Definition 4 holds if each agent simply chooses his single-player optimal control solution $J_i^* = J_i(u_i^*)$, since, for the disconnected graph case one has

$$J_i(u_i) = J_i(u_i, u_{G-i}) = J_i(u_i, u'_{G-i}), \ \forall i \tag{20}$$

for any choices of the two sets u_{G-i}, u'_{G-i} of the policies of all the other nodes. That is, the value function of each node does not depend on the policies of any other nodes.

Note, however, that Definition 3 also holds, that is, the nodes are in a global Nash equilibrium. Pathological cases such as this counterexample cannot occur in the standard games with centralized dynamics (18), particularly because stabilizability conditions are usually assumed.

3.3 Interactive Nash equilibrium

The counterexample in the previous section shows that in pathological cases when the graph is disconnected, agents can be in Nash equilibrium, yet have no influence on each others' games. In such situations, the definition of coalition-proof Nash equilibrium (Shinohara, 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 themselves.

To rule out such undesirable situations and guarantee that all agents in a graph are involved in the same game, we make the following stronger definition of global Nash equilibrium.

Definition 5. (Interactive Global Nash equilibrium) An *N-tuple* of policies $\{u_1^*, u_2^*, ..., u_N^*\}$ is said to constitute an interactive global Nash equilibrium solution for an *N* player game if, for all $i \in N$, the Nash condition (17) holds and in addition there exists a policy u'_k such that

$$J_i(u_k^*, u_{G-k}^*) \neq J_i(u'_k, u_{G-k}^*) \tag{21}$$

for all $i, k \in N$. That is, at equilibrium there exists a policy of every player *k* that influences the performance of all other players *i*.

If the systems are in Interactive Nash equilibrium, the graphical game is well-defined in the sense that all players are in a single Nash equilibrium with each player affecting the decisions of all other players. Condition (21) means that the reaction curve (Basar, and Olsder, 1999) of any player *i* is not constant with respect to all variations in the policy of any other player *k*.

The next results give conditions under which the local best responses in Definition 4 imply the interactive global Nash of Definition 5.

Consider the systems (8) in closed-loop with admissible feedbacks (12), (16) denoted by $u_k = K_k p_k - v_k$ for a single node *k* and $u_j = K_j p_j, \forall j \neq k$. Then

$$\dot{\delta}_i = A\delta_i + (d_i + g_i)B_i K_i p_i - \sum_{j \in N_i} e_{ij} B_j K_j p_j + e_{ik} B_k v_k, \quad k \neq i \tag{22}$$

The global closed-loop dynamics are

$$\begin{bmatrix} \dot{\delta} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} (I_N \otimes A) & ((L+G) \otimes I_n) diag(B_i K_i) \\ -diag(Q_{ii}) & -(I_N \otimes A^T) \end{bmatrix} \begin{bmatrix} \delta \\ p \end{bmatrix} + \begin{bmatrix} ((L+G) \otimes I_n) \underline{B}_k \\ 0 \end{bmatrix} \overline{v}_k \equiv \overline{A} \begin{bmatrix} \delta \\ p \end{bmatrix} + \overline{B} \overline{v}_k \tag{23}$$

with $\underline{B}_k = diag(B_i)$ and $\overline{v}_k = \begin{bmatrix} 0 & \cdots & v_k^T & \cdots & 0 \end{bmatrix}^T$ has all block entries zero with v_k in block *k*. Consider node *i* and let $M > 0$ be the first integer such that $[(L+G)^M]_{ik} \neq 0$, where $[.]_{ik}$ denotes the element *(i,k)* of a matrix. That is, *M* is the length of the shortest directed path from *k* to *i*. Denote the nodes along this path by $k = k_0, k_1, \cdots, k_{M-1}, k_M = i$. Denote element *(i,k)* of $L+G$ by ℓ_{ik}. Then the $n \times m$ block element in block row *i* and block column *k* of matrix $\overline{A}^{2(M-1)}\overline{B}$ is equal to

$$\left[\overline{A}^{2(M-1)} \overline{B} \right]^{ik} = \sum_{k_{M-1}, \cdots, k_1} \ell_{i,k_{M-1}} \cdots \ell_{k_1,k} B_{k_{M-1}} K_{k_{M-1}} Q_{k_{M-1}} B_{k_{M-2}} \cdots B_{k_1} K_{k_1} Q_{k_1} B_k \equiv \sum_{k_{M-1}} B_{k_{M-1}} \overline{B}_{k_{M-1},k} \tag{24}$$

where $\overline{B}_{k_{M-1},k} \in R^{m_{k_{M-1}} \times m_k}$ and $[\ ]^{ik}$ denotes the position of the block element in the block matrix.

Assumption 1.

a. $\overline{B}_{k_{M-1},k} \in R^{m_{k_{M-1}} \times m_k}$ has rank $m_{k_M - 1}$.

All shortest paths to node *i* from node *k* pass through a single neighbor $k_M - 1$ of *i*.

An example case where Assumption 1a holds is when there is a single shortest path from *k* to *i*, $m_i = m, \forall i$, $rank(B_i) = m, \forall i$.

Lemma 2. Let (A, B_j) be reachable for all $j \in N$ and let Assumption 1 hold. Then the *i*-th closed-loop system (22) is reachable from input v_k if and only if there exists a directed path from node *k* to node *i*.

Proof:

Sufficiency. If $k = i$ the result is obvious. Otherwise, the reachability matrix from node *k* to node *i* has the $n \times m$ block element in block row *i* and block column *k* given as

$$\left[\bar{A}^{2(M-1)}\bar{B} \quad \bar{A}^{2(M-1)+1}\bar{B} \quad \bar{A}^{2(M-1)+2}\bar{B} \quad \cdots\right]^{ik} = \left[\sum_{k_{M-1}} B_{k_{M-1}} \quad \sum_{k_{M-1}} AB_{k_{M-1}} \quad \sum_{k_{M-1}} A^2 B_{k_{M-1}} \quad \cdots\right]$$

$$\times \begin{bmatrix} \bar{B}_{k_{M-1},k} & * & * & \\ 0 & \bar{B}_{k_{M-1},k} & * & \\ \vdots & 0 & \bar{B}_{k_{M-1},k} & \\ 0 & \cdots & 0 & \ddots \end{bmatrix}$$

where * denotes nonzero entries. Under the assumptions, the matrix on the right has full row rank and the matrix on the left is written as $\left[B_{k_{M-1}} \quad AB_{k_{M-1}} \quad A^2 B_{k_{M-1}} \quad \cdots\right]$.

However, $(A, B_{k_{M-1}})$ is reachable.

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

■

Theorem 1. Let (A, B_i) be reachable for all $i \in N$. Let every node *i* be in best response to all his neighbors $j \in N_i$. Let Assumption 1 hold. Then all nodes in the graph are in interactive global Nash equilibrium if and only if the graph is strongly connected.

Proof:

Let every node i be in best response to all his neighbors $j \in N_i$. Then $J_i(u_i^*, u_{-i}) \le J_i(u_i, u_{-i}), \forall i$. Hence $u_j = u_j^*, \forall u_j \in u_{-i}$ and $J_i(u_i^*, u_{-i}^*) \le J_i(u_i, u_{-i}^*), \forall i$. However, according to (9) $J_i(u_i^*, u_{-i}^*, u_k) = J_i(u_i^*, u_{-i}^*, u_k), \forall k \notin \{i\} \cup N_i$ so that $J_i(u_i^*, u_{G-i}^*) \le J_i(u_i, u_{G-i}^*), \forall i$ and the nodes are in Nash equilibrium.

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

Sufficiency. Let (A, B_i) be reachable for all $i \in N$. Then if there is a path from node *k* to node *i*, the state δ_i is reachable from u_k, and from (9) input u_k can change the value J_i. Strong connectivity means there is a path from every node *k* to every node *i* and condition (21) holds for all $i, k \in N$.

■

The reachability condition is sufficient but not necessary for Interactive Nash equilibrium.

According to the results just established, the following assumptions are made.

Assumptions 2.

a. (A, B_i) is reachable for all $i \in N$.
b. The graph is strongly connected and at least one pinning gain g_i is nonzero. Then $(L+G)$ is nonsingular.

3.4 Stability and solution of graphical games

Substituting control policies (16) into (14) yields the coupled cooperative game Hamilton-Jacobi (HJ) equations

$$\frac{\partial V_i}{\partial \delta_i}^T A_i^c + \tfrac{1}{2}\delta_i^T Q_{ii}\delta_i + \tfrac{1}{2}(d_i+g_i)^2 \frac{\partial V_i}{\partial \delta_i}^T B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} + \tfrac{1}{2}\sum_{j\in N_i}(d_j+g_j)^2 \frac{\partial V_j}{\partial \delta_j}^T B_j R_{jj}^{-1} R_{ij} R_{jj}^{-1} B_j^T \frac{\partial V_j}{\partial \delta_j} = 0, i \in N \quad (25)$$

where the closed-loop matrix is

$$A_i^c = A\delta_i - (d_i+g_i)^2 B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} + \sum_{j\in N_i} e_{ij}(d_j+g_j) B_j R_{jj}^{-1} B_j^T \frac{\partial V_j}{\partial \delta_j}, i \in N \quad (26)$$

For a given V_i, define $u_i^* = u_i(V_i)$ as (16) given in terms of V_i. Then HJ equations (25) can be written as

$$H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}, u_i^*, u_{-i}^*) = 0 \quad (27)$$

There is one coupled HJ equation corresponding to each node, so solution of this *N*-player game problem is blocked by requiring a solution to *N* coupled partial differential equations. In the next sections we show how to solve this *N*-player cooperative game online in a distributed fashion at each node, requiring only measurements from neighbor nodes, by using techniques from reinforcement learning.

It is now shown that the coupled HJ equations (25) can be written as coupled Riccati equations. For the global state δ given in (4) we can write the dynamics as

$$\dot{\delta} = (I_N \otimes A)\delta + (L+G) \otimes I_n diag(B_i) u \quad (28)$$

where u is the control given by

$$u = -diag(R_{ii}^{-1} B_i^T)\big((D+G) \otimes I_n p\big) \quad (29)$$

where $diag(.)$ denotes diagonal matrix of appropriate dimensions. Furthermore the global costate dynamics are

$$-\dot{p} = \frac{\partial H}{\partial \delta} \equiv (I_N \otimes A)^T p + diag(Q_{ii})\delta \quad (30)$$

This is a set of coupled dynamic equations reminiscent of standard multi-player games (Basar, and Olsder, 1999) or single agent optimal control (Lewis, and Syrmos, 1995). Therefore the solution can be written without any loss of generality as

$$p = \bar{P}\delta \quad (31)$$

for some matrix $\overline{P} > 0 \in \mathbb{R}^{nNxnN}$.

Lemma 3. HJ **e**quations (25) are equivalent to the coupled Riccati equations

$$\delta^T \overline{P}^T \overline{A}_i \delta - \delta^T \overline{P}^T \overline{B}_i \overline{P} \delta + \tfrac{1}{2} \delta^T \overline{Q}_i \delta + \tfrac{1}{2} \delta^T \overline{P}^T \overline{R}_i \overline{P} \delta = 0 \tag{32}$$

or equivalently, in closed-loop form,

$$(\overline{P}^T \overline{A}_{ic} + \overline{A}_{ic}{}^T \overline{P} + \overline{Q}_i + \overline{P}^T \overline{R}_i \overline{P}) = 0 \tag{33}$$

where $\overline{P}$ is defined by (31), and

$$\overline{A}_i = \begin{bmatrix} 0 & & \\ & 0 & \\ & & [A]^{ii} \\ & & & 0 \end{bmatrix}, \overline{B}_i = \begin{bmatrix} 0 & & & \\ & \left[(d_i + g_i) I_n\right]^{ii} & \left[-a_{ij} I_n\right]^{ij} & \\ & & 0 & \end{bmatrix} diag((d_i + g_i) B_i R_{ii}^{-1} B_i{}^T)$$

$$\overline{A}_{ic} = \overline{A}_i - \overline{B}_i \overline{P}$$

$$\overline{Q}_i = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & [Q_{ii}]^{ii} & \\ & & & 0 \end{bmatrix}, \overline{R}_i = diag((d_i + g_i) B_i R_{ii}^{-1}) \begin{bmatrix} R_{i1} & & & & & \\ & \ddots & & & & \\ & & R_{ij} & & & \\ & & & \ddots & & \\ & & & & R_{ii} & \\ & & & & & R_{iN} \end{bmatrix} diag((d_i + g_i) R_{ii}^{-1} B_i{}^T)$$

Proof:

Take (14) and write it with respect to the global state and costate as

$$H_i \equiv \begin{bmatrix} \frac{\partial V_1}{\partial \delta_1} \\ \vdots \\ \vdots \\ \frac{\partial V_N}{\partial \delta_N} \end{bmatrix}^T \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & [A]^{ii} & \\ & & & 0 \end{bmatrix} \delta$$

$$+ \begin{bmatrix} \frac{\partial V_1}{\partial \delta_1} \\ \vdots \\ \vdots \\ \frac{\partial V_N}{\partial \delta_N} \end{bmatrix}^T \begin{bmatrix} 0 & \cdots & 0 & 0 \\ \vdots & 0 & \vdots & \vdots \\ \vdots & \vdots & \left[(d_i + g_i) I_n\right]^{ii} & \left[-a_{ij} I_n\right]^{ij} \\ 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} B_1 & & & \\ & \ddots & & \\ & & B_i & \\ & & & B_N \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_i \\ u_N \end{bmatrix} \tag{34}$$

$$+\tfrac{1}{2}\delta^T\begin{bmatrix}0 & & & \\ & 0 & & \\ & & [Q_{ii}]^{ii} & \\ & & & 0\end{bmatrix}\delta+\tfrac{1}{2}\begin{bmatrix}u_1\\ \vdots\\ u_i\\ u_N\end{bmatrix}^T\begin{bmatrix}R_{i1} & & & \\ & R_{ij} & & \\ & & R_{ii} & \\ & & & R_{iN}\end{bmatrix}\begin{bmatrix}u_1\\ \vdots\\ u_i\\ u_N\end{bmatrix}=0$$

By definition of the costate one has

$$p\equiv\begin{bmatrix}\frac{\partial V_1}{\partial\delta_1} & \dots & \dots & \frac{\partial V_N}{\partial\delta_N}\end{bmatrix}^T=\overline{P}\delta \tag{35}$$

■

From the control policies (16), (34) becomes (32).

It is now shown that if solutions can be found for the coupled design equations (25), they provide the solution to the graphical game problem.

Theorem 2. Stability and Solution for Cooperative Nash Equilibrium.

Let Assumptions 1 and 2a hold. Let $V_i>0\in C^1$, $i\in N$ be smooth solutions to HJ equations (25) and control policies u_i^*, $i\in N$ be given by (16) in terms of these solutions V_i. Then

a. Systems (8) are asymptotically stable so all agents synchronize.

$\left\{u_1^*,u_2^*,...,u_N^*\right\}$ are in global Nash equilibrium and the corresponding game values are

$$J_i^*(\delta_i(0))=V_i\ ,\ i\in N \tag{36}$$

Proof:

If $V_i>0$ satisfies (25) then it also satisfies (14). Take the time derivative to obtain

$$\dot{V}_i=\frac{\partial V_i}{\partial\delta_i}^T\dot{\delta}_i=\frac{\partial V_i}{\partial\delta_i}^T\left(A\delta_i+(d_i+g_i)B_iu_i-\sum_{j\in N_i}e_{ij}B_ju_j\right)=-\tfrac{1}{2}\left(\delta_i^TQ_{ii}\delta_i+u_i^TR_{ii}u_i+\sum_{j\in N_i}u_j^TR_{ij}u_j\right) \tag{37}$$

which is negative definite since $Q_{ii}>0$. Therefore V_i is a Lyapunov function for δ_i and systems (8) are asymptotically stable.

According to part *a*, $\delta_i(t)\to 0$ for the selected control policies. For any smooth functions $V_i(\delta_i)$, $i\in N$, such that $V_i(0)=0$, setting $V_i(\delta_i(\infty))=0$ one can write (9) as

$$J_i(\delta_i(0),u_i,u_{-i})=\tfrac{1}{2}\int_0^\infty(\delta_i^TQ_{ii}\delta_i+u_i^TR_{ii}u_i+\sum_{j\in N_i}u_j^TR_{ij}u_j)\,dt+V_i(\delta_i(0))$$

$$+\int_0^\infty\frac{\partial V_i}{\partial\delta_i}^T(A\delta_i+(d_i+g_i)B_iu_i-\sum_{j\in N_i}e_{ij}B_ju_j)dt$$

Now let V_i satisfy (25) and u_i^*, u_{-i}^* be the optimal controls given by (16). By completing the squares one has

$$J_i(\delta_i(0), u_i, u_{-i}) = V_i(\delta_i(0)) + \int_0^\infty (\tfrac{1}{2}\sum_{j\in N_i}(u_j - u_j^*)^T R_{ij}(u_j - u_j^*) + \tfrac{1}{2}(u_i - u_i^*)^T R_{ii}(u_i - u_i^*)$$
$$-\frac{\partial V_i}{\partial \delta_i}^T \sum_{j\in N_i} e_{ij}B_j(u_j - u_j^*) + \sum_{j\in N_i} u_j^{*T} R_{ij}(u_j - u_j^*))dt$$

At the equilibrium point $u_i = u_i^*$ and $u_j = u_j^*$ so

$$J_i^*(\delta_i(0), u_i^*, u_{-i}^*) = V_i(\delta_i(0))$$

Define

$$J_i(u_i, u_{-i}^*) = V_i(\delta_i(0)) + \tfrac{1}{2}\int_0^\infty (u_i - u_i^*)^T R_{ii}(u_i - u_i^*)dt$$

and $J_i^* = V_i(\delta_i(0))$. Then clearly J_i^* and $J_i(u_i, u_{-i}^*)$ satisfy (19). Since this is true for all *i*, Nash condition (17) is satisfied.

■

The next result shows when the systems are in Interactive Nash equilibrium. This means that the graphical game is well defined in the sense that all players are in a single Nash equilibrium with each player affecting the decisions of all other players.

Corollary 1. Let the hypotheses of Theorem 2 hold. Let Assumptions 1 and 2 hold so that the graph is strongly connected. Then $\{u_1^*, u_2^*, ..., u_N^*\}$ are in interactive Nash equilibrium and all agents synchronize.

Proof:

From Theorems 1 and 2.

■

3.5 Global and local performance objectives: Cooperation and competition

The overall objective of all the nodes is to ensure synchronization of all the states $x_i(t)$ to $x_0(t)$. The multi player game formulation allows for considerable freedom of each agent while achieving this objective. Each agent has a performance objective that can embody team objectives as well as individual node objectives.

The performance objective of each node can be written as

$$J_i = \tfrac{1}{N_i}\sum_{j\in N_i} J_j + \tfrac{1}{N_i}\sum_{j\in N_i}(J_i - J_j) \equiv J_{team} + J_i^{conflict}$$

where J_{team} is the overall ('center of gravity') performance objective of the networked team and $J_i^{conflict}$ is the conflict of interest or competitive objective. J_{team} measures how much the players are vested in common goals, and $J_i^{conflict}$ expresses to what extent their objectives differ. The objective functions can be chosen by the individual players, or they may be assigned to yield some desired team behavior.

4. Policy iteration algorithms for cooperative multi-player games

Reinforcement learning (RL) techniques have been used to solve the single-player optimal control problem online using adaptive learning techniques to determine the optimal value function. Especially effective are the approximate dynamic programming (ADP) methods (Werbos, 1974; Werbos, 1992). RL techniques have also been applied for multiplayer games with centralized dynamics (18). See for example (Busoniu, Babuska, and De Schutter, 2008; Vrancx, Verbeeck, and Nowe, 2008). Most applications of RL for solving optimal control problems or games online have been to finite-state systems or discrete-time dynamical systems. In this section is given a policy iteration algorithm for solving continuous-time differential games on graphs. The structure of this algorithm is used in the next section to provide online adaptive solutions for graphical games.

4.1 Best response

Theorem 2 and Corollary 1 reveal that, under assumptions 1 and 2, the systems are in interactive Nash equilibrium if, for all $i \in N$ node i selects his best response policy to his neighbors policies and the graph is strongly connected. Define the best response HJ equation as the Bellman equation (14) with control $u_i = u_i^*$ given by (16) and arbitrary policies $u_{-i} = \{u_j : j \in N_i\}$

$$0 = H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}, u_i^*, u_{-i}) \equiv \frac{\partial V_i}{\partial \delta_i}^T A_i^c + \tfrac{1}{2}\delta_i^T Q_{ii}\delta_i + \tfrac{1}{2}(d_i + g_i)^2 \frac{\partial V_i}{\partial \delta_i}^T B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} + \tfrac{1}{2}\sum_{j \in N_i} u_j^T R_{ij} u_j \quad (38)$$

where the closed-loop matrix is

$$A_i^c = A\delta_i - (d_i + g_i)^2 B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} - \sum_{j \in N_i} e_{ij} B_j u_j \quad (39)$$

Theorem 3. Solution for Best Response Policy

Given fixed neighbor policies $u_{-i} = \{u_j : j \in N_i\}$, assume there is an admissible policy u_i. Let $V_i > 0 \in C^1$ be a smooth solution to the best response HJ equation (38) and let control policy u_i^* be given by (16) in terms of this solution V_i. Then

a. Systems (8) are asymptotically stable so that all agents synchronize.
b. u_i^* is the best response to the fixed policies u_{-i} of its neighbors.

Proof:

a. $V_i > 0$ satisfies (38). Proof follows Theorem 2, part a.

b. According to part *a*, $\delta_i(t) \to 0$ for the selected control policies. For any smooth functions $V_i(\delta_i), i \in N$, such that $V_i(0) = 0$, setting $V_i(\delta_i(\infty)) = 0$ one can write (9) as

$$J_i(\delta_i(0), u_i, u_{-i}) = \tfrac{1}{2}\int_0^\infty (\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii} u_i + \sum_{j\in N_i} u_j^T R_{ij} u_j)\, dt + V_i(\delta_i(0))$$

$$+\int_0^\infty \frac{\partial V_i}{\partial \delta_i}^T (A\delta_i + (d_i + g_i)B_i u_i - \sum_{j\in N_i} e_{ij} B_j u_j) dt$$

Now let V_i satisfy (38), u_i^* be the optimal controls given by (16), and u_{-i} be arbitrary policies. By completing the squares one has

$$J_i(\delta_i(0), u_i, u_{-i}) = V_i\ (\delta_i(0)) + \int_0^\infty \tfrac{1}{2}(u_i - u_i^*)^T R_{ii}(u_i - u_i^*) dt$$

The agents are in best response to fixed policies u_{-i} when $u_i = u_i^*$ so

$$J_i(\delta_i(0), u_i^*, u_{-i}) = V_i\ (\delta_i(0))$$

Then clearly $J_i(\delta_i(0), u_i, u_{-i})$ and $J_i(\delta_i(0), u_i^*, u_{-i})$ satisfy (19).

■

4.2 Policy iteration solution for graphical games

The following algorithm for the *N*-player distributed games is motivated by the structure of policy iteration algorithms in reinforcement learning (Bertsekas, and Tsitsiklis, 1996; Sutton, and Barto, 1998) which rely on repeated policy evaluation (e.g. solution of (14)) and policy improvement (solution of (16)). These two steps are repeated until the policy improvement step no longer changes the present policy. If the algorithm converges for every i, then it converges to the solution to HJ equations (25), and hence provides the distributed Nash equilibrium. One must note that the costs can be evaluated only in the case of admissible control policies, admissibility being a condition for the control policy which initializes the algorithm.

Algorithm 1. Policy Iteration (PI) Solution for N**-player distributed games.**

Step 0: Start with admissible initial policies $u_i^0, \forall i$.

Step 1: (Policy Evaluation) Solve for V_i^k using (14)

$$H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_i^k, u_{-i}^k) = 0, \forall i = 1, \ldots, N \tag{40}$$

Step 2: (Policy Improvement) Update the *N*-tuple of control policies using

$$u_i^{k+1} = \arg\min_{u_i} H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_i, u_{-i}^k), \forall i = 1,\ldots,N$$

which explicitly is

$$u_i^{k+1} = -(d_i + g_i) R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i}^k, \forall i = 1,\ldots,N . \tag{41}$$

Go to step 1.

On convergence- End

■

The following two theorems prove convergence of the policy iteration algorithm for distributed games for two different cases. The two cases considered are the following, i) *only* agent *i* updates its policy and ii) all the agents update their policies.

Theorem 4. Convergence of Policy Iteration algorithm when only ith agent updates its policy and all players u_{-i} in its neighborhood do not change. Given fixed neighbors policies u_{-i}, assume there exists an admissible policy u_i. Assume that agent *i* performs Algorithm 1 and the its neighbors do not update their control policies. Then the algorithm converges to the best response u_i to policies u_{-i} of the neighbors and to the solution V_i to the best response HJ equation (38).

Proof:

It is clear that

$$H_i^o(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_{-i}^k) \equiv \min_{u_i} H_i(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_i^k, u_{-i}^k) = H_i(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_i^{k+1}, u_{-i}^k) \tag{42}$$

Let $H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_i^k, u_{-i}^k) = 0$ from (40) then according to (42) it is clear that

$$H_i^o(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_{-i}^k) \le 0 \tag{43}$$

Using the next control policy u_i^{k+1} and the current policies u_{-i}^k one has the orbital derivative (Leake, Wen Liu, 1967)

$$\dot{V}_i^k = H_i(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_i^{k+1}, u_{-i}^k) - L_i(\delta_i, u_i^{k+1}, u_{-i}^k)$$

From (42) and (43) one has

$$\dot{V}_i^k = H_i^0(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_{-i}^k) - L_i(\delta_i, u_i^{k+1}, u_{-i}^k) \le -L_i(\delta_i, u_i^{k+1}, u_{-i}^k) \tag{44}$$

Because only agent i update its control it is true that $u_{-i}^{k+1} = u_{-i}^k$ and

$$H_i(\delta_i, \frac{\partial V_i^{k+1}}{\partial \delta_i}, u_i^{k+1}, u_{-i}^k) = 0 \cdot$$

But since $\dot{V}_i^{k+1} = -L_i(\delta_i, u_i^{k+1}, u_{-i}^{k+1})$, from (44) one has

$$\dot{V}_i^k = H_i^0(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_{-i}^k) - L_i(\delta_i, u_i^{k+1}, u_{-i}^k) \le -L_i(\delta_i, u_i^{k+1}, u_{-i}^k) = \dot{V}_i^{k+1} \tag{45}$$

So that $\dot{V}_i^k \le \dot{V}_i^{k+1}$ and by integration it follows that

$$V_i^{k+1} \le V_i^k \tag{46}$$

Since $V_i^* \le V_i^k$, the algorithm converges, to V_i^*, to the best response HJ equation (38).

■

The next result concerns the case where all nodes update their policies at each step of the algorithm. Define the relative control weighting as $\rho_{ij} = \bar{\sigma}(R_{jj}^{-1} R_{ij})$, where $\bar{\sigma}(R_{jj}^{-1} R_{ij})$ is the maximum singular value of $R_{jj}^{-1} R_{ij}$.

Theorem 5. Convergence of Policy Iteration algorithm when all agents update their policies. Assume all nodes i update their policies at each iteration of PI. Then for small enough edge weights e_{ij} and ρ_{ij}, u_i converges to the global Nash equilibrium and for all i, and the values converge to the optimal game values $V_i^k \to V_i^*$.

Proof:

It is clear that

$$H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}^{k+1}, u_i^{k+1}, u_{-i}^{k+1}) \equiv H_i^0(\delta_i, \frac{\partial V_i}{\partial \delta_i}^{k+1}, u_{-i}^k) + \tfrac{1}{2} \sum_{j \in N_i} (u_j^{k+1} - u_j^k)^T R_{ij} (u_j^{k+1} - u_j^k)$$
$$+ \sum_{j \in N_i} u_j^{kT} R_{ij} (u_j^{k+1} - u_j^k) + \frac{\partial V_i}{\partial \delta_i}^{k+1T} \sum_{j \in N_i} e_{ij} B_j (u_j^k - u_j^{k+1})$$

and so

$$\dot{V}_i^{k+1} = -L_i(\delta_i, u_i^{k+1}, u_{-i}^{k+1}) = -L_i(\delta_i, u_i^{k+1}, u_{-i}^k) + \tfrac{1}{2} \sum_{j \in N_i} (u_j^{k+1} - u_j^k)^T R_{ij} (u_j^{k+1} - u_j^k)$$
$$+ \frac{\partial V_i}{\partial \delta_i}^{k+1T} \sum_{j \in N_i} e_{ij} B_j (u_j^k - u_j^{k+1}) + \sum_{j \in N_i} u_j^{kT} R_{ij} (u_j^{k+1} - u_j^k)$$

Therefore,

$$\dot{V}_i^k \le \dot{V}_i^{k+1} - \tfrac{1}{2}\sum_{j\in N_i}(u_j{}^{k+1} - u_j{}^k)^T R_{ij}(u_j{}^{k+1} - u_j{}^k)$$
$$+\frac{\partial V_i}{\partial \delta_i}^{k+1T}\sum_{j\in N_i} e_{ij}B_j(u_j{}^{k+1} - u_j{}^k) - \sum_{j\in N_i} u_j{}^{kT}R_{ij}(u_j{}^{k+1} - u_j{}^k)$$

A sufficient condition for $\dot{V}_i^k \le \dot{V}_i^{k+1}$ is

$$\tfrac{1}{2}\Delta u_j{}^T R_{ij}\Delta u_j - e_{ij}(p_i^{k+1})^T B_j \Delta u_j - (d_j + g_j)(p_j^{k-1})B_j{}^T R_{jj}^{-1}R_{ij}\Delta u_j > 0$$

$\tfrac{1}{2}\underline{\sigma}(R_{ij})\left\|\Delta u_j\right\| > e_{ij}\left\|p_i^{k+1}\right\|\cdot\left\|B_j\right\| + (d_j + g_j)\rho_{ij}\left\|p_j^{k-1}\right\|\cdot\left\|B_j\right\|$ where $\Delta u_j = (u_j{}^{k+1} - u_j{}^k)$, p_i the costate and $\underline{\sigma}(R_{ij})$ is the minimum singular value of R_{ij}.

This holds if $e_{ij} = 0, \rho_{ij} = 0$. By continuity, it holds for small values of e_{ij}, ρ_{ij}.

■

This proof indicates that for the PI algorithm to converge, the neighbors' controls should not unduly influence the *i*-th node dynamics (8), and the *j*-th node should weight its own control u_j in its performance index J_j relatively more than node *i* weights u_j in J_i. These requirements are consistent with selecting the weighting matrices to obtain proper performance in the simulation examples. An alternative condition for convergence in Theorem 5 is that the norm $\left\|B_j\right\|$ should be small. This is similar to the case of weakly coupled dynamics in multi-player games in (Basar, and Olsder, 1999).

5. Online solution of multi-agent cooperative games using neural networks

In this section an online algorithm for solving cooperative Hamilton-Jacobi equations (25) based on (Vamvoudakis, Lewis 2011) is presented. This algorithm uses the structure in the PI Algorithm 1 to develop an actor/critic adaptive control architecture for approximate online solution of (25). Approximate solutions of (40), (41) are obtained using value function approximation (VFA). The algorithm uses two approximator structures at each node, which are taken here as neural networks (NN) (Abu-Khalaf, and Lewis, 2005; Bertsekas, and Tsitsiklis, 1996; Vamvoudakis, Lewis 2010; Werbos, 1974; Werbos, 1992). One critic NN is used at each node for value function approximation, and one actor NN at each node to approximate the control policy (41). The critic NN seeks to solve Bellman equation (40). We give tuning laws for the actor NN and the critic NN such that equations (40) and (41) are solved simultaneously online for each node. Then, the solutions to the coupled HJ equations (25) are determined. Though these coupled HJ equations are difficult to solve, and may not even have analytic solutions, we show how to tune the NN so that the approximate solutions are learned online. The next assumption is made.

Assumption 2. For each admissible control policy the nonlinear Bellman equations (14), (40) have smooth solutions $V_i \ge 0$.

In fact, only local smooth solutions are needed. To solve the Bellman equations (40), approximation is required of both the value functions V_i and their gradients $\partial V_i / \partial \delta_i$. This requires approximation in Sobolev space (Abu-Khalaf, and Lewis, 2005).

5.1 Critic neural network

According to the Weierstrass higher-order approximation Theorem (Abou-Khalaf, and Lewis, 2005) there are NN weights W_i such that the smooth value functions V_i are approximated using a critic NN as

$$V_i(\delta_i) = W_i^T \phi_i(z_i) + \varepsilon_i \tag{47}$$

where $z_i(t)$ is an information vector constructed at node i using locally available measurements, e.g. $\delta_i(t), \{\delta_j(t) : j \in N_i\}$. Vectors $\phi_i(z_i) \in \mathbb{R}^h$ are the critic NN activation function vectors, with h the number of neurons in the critic NN hidden layer. According to the Weierstrass Theorem, the NN approximation error ε_i converges to zero uniformly as $h \to \infty$. Assuming current weight estimates $\hat{W}_i$, the outputs of the critic NN are given by

$$\hat{V}_i = \hat{W}_i^T \phi_i \tag{48}$$

Then, the Bellman equation (40) can be approximated at each step k as

$$H_i(\delta_i, \hat{W}_i, u_i, u_{-i}) = \delta_i^T Q_{ii} \delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j + \hat{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} (A\delta_i + (d_i + g_i) B_i u_i - \sum_{j \in N_i} e_{ij} B_j u_j) = e_{H_i} \tag{49}$$

It is desired to select $\hat{W}_i$ to minimize the square residual error

$$E_1 = \tfrac{1}{2} e_{H_i}^T e_{H_i} \tag{50}$$

Then $\hat{W}_i \to W_i$ which solves (49) in a least-squares sense and e_{H_i} becomes small. Theorem 6 gives a tuning law for the critic weights that achieves this.

5.2 Action neural network and online learning

Define the control policy in the form of an action neural network which computes the control input (41) in the structured form

$$\hat{u}_i \equiv \hat{u}_{i+N} = -\tfrac{1}{2} (d_i + g_i) R_{ii}^{-1} B_i^T \left(\frac{\partial \phi_i}{\partial \delta_i} \right)^T \hat{W}_{i+N} \tag{51}$$

where $\hat{W}_{i+N}$ denotes the current estimated values of the ideal actor NN weights W_i. The notation $\hat{u}_{i+N}$ is used to keep indices straight in the proof. Define the critic and actor NN estimation errors as $\tilde{W}_i = W_i - \hat{W}_i$ and $\tilde{W}_{i+N} = W_i - \hat{W}_{i+N}$.

The next results show how to tune the critic NN and actor NN in real time at each node so that equations (40) and (41) are simultaneously solved, while closed-loop system stability is

also guaranteed. Simultaneous solution of (40) and (41) guarantees that the coupled HJ equations (25) are solved for each node *i*. System (8) is said to be uniformly ultimately bounded (UUB) if there exists a compact set $S \subset \mathbb{R}^n$ so that for all $\delta_i(0) \in S$ there exists a bound *B* and a time $T(B, \delta_i(0))$ such that $\|\delta_i(t)\| \le B$ for all $t \ge t_0 + T$.

Select the tuning law for the *i*th critic NN as

$$\dot{\hat{W}}_i = -a_i \frac{\partial E_1}{\partial \hat{W}_i} = -a_i \frac{\sigma_{i+N}}{(1+\sigma_{i+N}{}^{\mathrm{T}}\sigma_{i+N})^2}[\sigma_{i+N}{}^{\mathrm{T}}\hat{W}_i + \delta_i^T Q_{ii}\delta_i + \tfrac{1}{4}\hat{W}_{i+N}{}^T \bar{D}_i \hat{W}_{i+N}$$
$$+\tfrac{1}{4}\sum_{j\in N_i}(d_j+g_j)^2 \hat{W}_{j+N}{}^T \frac{\partial \phi_j}{\partial \delta_j} B_j R_{jj}^{-T} R_{ij} R_{jj}^{-1} B_j{}^T \frac{\partial \phi_j^T}{\partial \delta_j}\hat{W}_{j+N}] \tag{52}$$

where $\sigma_{i+N} = \frac{\partial \phi_i}{\partial \delta_i}(A\delta_i + (d_i+g_i)B_i\hat{u}_{i+N} - \sum_{j\in N_i} e_{ij}B_j\hat{u}_{j+N})$, and the tuning law for the *i*th actor NN as

$$\dot{\hat{W}}_{i+N} = -a_{i+N}\{(S_i\hat{W}_{i+N} - F_i\bar{\sigma}_{i+N}^T\hat{W}_i) - \frac{1}{4}\bar{D}_i\hat{W}_{i+N}\frac{\bar{\sigma}_{i+N}}{m_{si}}{}^T\hat{W}_i$$
$$-\frac{1}{4}\sum_{j\in N_i}(d_j+g_j)^2\frac{\partial\phi_j}{\partial\delta_j}B_jR_{jj}^{-T}R_{ij}R_{jj}^{-1}B_j{}^T\frac{\partial\phi_j}{\partial\delta_j}{}^T\hat{W}_j\frac{\bar{\sigma}_{i+N}}{m_{s_i}}{}^T\hat{W}_{i+N}\} \tag{53}$$

where

$\bar{D}_i(x) \equiv \frac{\partial\phi_i}{\partial\delta_i}B_iR_{ii}^{-1}B_i{}^T\frac{\partial\phi_i}{\partial\delta_i}{}^T$, $m_{s_i} \equiv (\sigma_{i+N}{}^T\sigma_{i+N}+1)$, $\bar{\sigma}_{i+N} = \sigma_{i+N}/(\sigma_{i+N}{}^T\sigma_{i+N}+1)$, and $a_i > 0, \ldots a_{i+N} > 0$ and $F_i > 0, G_i > 0, \quad i \in N$ are tuning parameters.

Theorem 6. Online Cooperative Games.

Let the error dynamics be given by (8), and consider the cooperative game formulation in (15). Let the critic NN at each node be given by (48) and the control input be given for each node by actor NN (51). Let the tuning law for the *i*th critic NN be provided by (52) and the tuning law for the *i*th actor NN be provided by (53). Assume $\bar{\sigma}_{i+N} = \sigma_{i+N}/(\sigma_{i+N}{}^T\sigma_{i+N}+1)$ is persistently exciting. Then the closed-loop system states $\delta_i(t)$, the critic NN errors $\tilde{W}_i$, and the actor NN errors $\tilde{W}_{i+N}$ are uniformly ultimately bounded.

Proof:

The proof is similar to (Vamvoudakis, 2011).

■

Remark 1. Theorem 6 provides algorithms for tuning the actor/critic networks of the *N* agents at the same time to guarantee stability and make the system errors $\delta_i(t)$ small and

the NN approximation errors bounded. Small errors guarantee synchronization of all the node trajectories.

Remark 2. Persistence of excitation is needed for proper identification of the value functions by the critic NNs, and nonstandard tuning algorithms are required for the actor NNs to guarantee stability. It is important to notice that the actor NN tuning law of every agent needs information of the critic weights of all his neighbors, while the critic NN tuning law of every agent needs information of the actor weights of all his neighbors,

Remark 3. NN usage suggests starting with random, nonzero control NN weights in (51) in order to converge to the coupled HJ equation solutions. However, extensive simulations show that convergence is more sensitive to the persistence of excitation in the control inputs than to the NN weight initialization. If the proper persistence of excitation is not selected, the control weights may not converge to the correct values.

Remark 4. The issue of which inputs $z_i(t)$ to use for the critic and actor NNs needs to be addressed. According to the dynamics (8), the value functions (13), and the control inputs (16), the NN inputs at node *i* should consist of its own state, the states of its neighbors, and the costates of its neighbors. However, in view of (31) the costates are functions of the states. In view of the approximation capabilities of NN, it is found in simulations that it is suitable to take as the NN inputs at node *i* its own state and the states of its neighbors.

The next result shows that the tuning laws given in Theorem 6 guarantee approximate solution to the coupled HJ equations (25) and convergence to the Nash equilibrium.

Theorem 7. Convergence to Cooperative Nash Equilibrium.

Suppose the hypotheses of Theorem 6 hold. Then:

a. $H_i(\delta_i, \hat{W}_i, \hat{u}_i, \hat{u}_{-i}), \quad \forall i \in N$ are uniformly ultimately bounded, where $\hat{u}_i = -\frac{1}{2}(d_i + g_i) R_{ii}^{-1} B_i^T \left(\frac{\partial \phi_i}{\partial \delta_i}\right)^T \hat{W}_i$. That is, $\hat{W}_i$ converge to the approximate cooperative coupled HJ-solution.

b. $\hat{u}_{i+N}$ converge to the approximate cooperative Nash equilibrium (Definition 2) for every i.

Proof:

The proof is similar to (Vamvoudakis, 2011) but is done only with respect to the neighbors (local information) of each agent and not with respect to all agents.

Consider the weights $\hat{W}_i, \hat{W}_{i+N}$ to be UUB as proved in Theorem 6.

a. The approximate coupled HJ equations are $H_i(\delta_i, \hat{W}_i, \hat{u}_i, \hat{u}_{-i}), \quad \forall i \in N$.

$$H_i(\delta_i, \hat{W}_i, \hat{u}_i, \hat{u}_{-i}) \equiv H_i(\delta_i, \hat{W}_i \hat{W}_{-i}) = \delta_i^T Q_{ii} \delta_i + \hat{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} A \delta_i - \frac{1}{4}(d_i + g_i)^2 \hat{W}_i^T \frac{\partial \phi_i}{\partial \delta_i} B_i R_{ii}^{-1} B_i^T \left(\frac{\partial \phi_i}{\partial \delta_i}\right)^T \hat{W}_i$$

$$+\tfrac{1}{4}\sum_{j\in N_i}(d_j+g_j)^2\hat{W}_j^T\frac{\partial\phi_j}{\partial\delta_j}B_jR_{jj}^{-1}R_{ij}R_{jj}^{-1}B_j^T\frac{\partial\phi_j}{\partial\delta_j}^T\hat{W}_j+\tfrac{1}{2}\hat{W}_i^T\frac{\partial\phi_i}{\partial\delta_i}\sum_{j\in N_i}e_{ij}B_jR_{jj}^{-1}B_j^T\frac{\partial\phi_j}{\partial\delta_j}^T\hat{W}_j-\varepsilon_{HJ_i}$$

where ε_{HJ_i}, $\forall i$ are the residual errors due to approximation.

After adding zero we have

$$H_i(\delta_i,\hat{W}_i\,\hat{W}_{-i})=-\tilde{W}_i^T\frac{\partial\phi_i}{\partial\delta_i}A\delta_i-\tfrac{1}{4}(d_i+g_i)^2\tilde{W}_i^T\frac{\partial\phi_i}{\partial\delta_i}B_iR_{ii}^{-1}B_i^T\frac{\partial\phi_i}{\partial\delta_i}^T\tilde{W}_i$$

$$+\tfrac{1}{2}(d_i+g_i)^2W_i^T\frac{\partial\phi_i}{\partial\delta_i}B_iR_{ii}^{-1}B_i^T\frac{\partial\phi_i}{\partial\delta_i}^TW_i+\tfrac{1}{2}(d_i+g_i)^2W_i^T\frac{\partial\phi_i}{\partial\delta_i}B_iR_{ii}^{-1}B_i^T\frac{\partial\phi_i}{\partial\delta_i}^T\hat{W}_i$$

$$-\tfrac{1}{4}\sum_{j\in N_i}(d_j+g_j)^2\tilde{W}_j^T\frac{\partial\phi_j}{\partial\delta_j}B_jR_{jj}^{-1}R_{ij}R_{jj}^{-1}B_j^T\frac{\partial\phi_j}{\partial\delta_j}^T\tilde{W}_j+\tfrac{1}{2}\sum_{j\in N_i}(d_j+g_j)^2W_j^T\frac{\partial\phi_j}{\partial\delta_j}B_jR_{jj}^{-1}R_{ij}R_{jj}^{-1}B_j^T\frac{\partial\phi_j}{\partial\delta_j}^TW_j$$
$$+\tfrac{1}{2}\sum_{j\in N_i}(d_j+g_j)^2W_j^T\frac{\partial\phi_j}{\partial\delta_j}B_jR_{jj}^{-1}R_{ij}R_{jj}^{-1}B_j^T\frac{\partial\phi_j}{\partial\delta_j}^T\hat{W}_j+\hat{W}_i^T\frac{\partial\phi_i}{\partial\delta_i}\sum_{j\in N_i}e_{ij}B_jR_{jj}^{-1}B_j^T\frac{\partial\phi_j}{\partial\delta_j}^T\hat{W}_j-\varepsilon_{HJ_i}$$
$$-\tfrac{1}{2}\tilde{W}_i^T\frac{\partial\phi_i}{\partial\delta_i}\sum_{j\in N_i}e_{ij}B_jR_{jj}^{-1}B_j^T\frac{\partial\phi_j}{\partial\delta_j}^T\tilde{W}_j-\tfrac{1}{2}W_i^T\frac{\partial\phi_i}{\partial\delta_i}\sum_{j\in N_i}e_{ij}B_jR_{jj}^{-1}B_j^T\frac{\partial\phi_j}{\partial\delta_j}^T\hat{W}_j$$

$$-\tfrac{1}{2}\hat{W}_i^T\frac{\partial\phi_i}{\partial\delta_i}\sum_{j\in N_i}e_{ij}B_jR_{jj}^{-1}B_j^T\frac{\partial\phi_j}{\partial\delta_j}^TW_j \tag{54}$$

But

$$\hat{W}_i=-\tilde{W}_i+W_i,\quad\forall i. \tag{55}$$

After taking norms in (55) and letting $\|W_i\|<W_{i\max}$ one has

$$\left\|\hat{W}_i\right\|=\left\|-\tilde{W}_i+W_i\right\|\le\left\|\tilde{W}_i\right\|+\left\|W_i\right\|\le\left\|\tilde{W}_i\right\|+W_{i\max}$$

Now (54) with $\sup\left\|\varepsilon_{HJ_i}\right\|<\bar{\varepsilon}_i$ becomes

$$\left\|H_i(\delta_i,\hat{W}_i\,\hat{W}_{-i})\right\|\le\left\|\tilde{W}_i\right\|\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|A\|\delta_i\|+\tfrac{1}{4}(d_i+g_i)^2\left\|\tilde{W}_i\right\|^2\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|^2\|B_i\|^2\left\|R_{ii}^{-1}\right\|$$
$$+\tfrac{1}{2}(d_i+g_i)^2\|W_{i\max}\|^2\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|^2\|B_i\|^2\left\|R_{ii}^{-1}\right\|+\tfrac{1}{2}(d_i+g_i)^2\|W_{i\max}\|\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|^2\|B_i\|^2\left\|R_{ii}^{-1}\right\|\left(\left\|\tilde{W}_i\right\|+W_{i\max}\right)$$

$$+\tfrac{1}{4}\sum_{j\in N_i}(d_j+g_j)^2\left\|\tilde{W}_j\right\|^2\left\|\frac{\partial\phi_j}{\partial\delta_j}\right\|^2\left\|B_j\right\|^2\left\|R_{jj}^{-1}R_{ij}R_{jj}^{-1}\right\|+\tfrac{1}{2}\sum_{j\in N_i}(d_j+g_j)^2\left\|W_{j\max}\right\|^2\left\|\frac{\partial\phi_j}{\partial\delta_j}\right\|^2\left\|B_j\right\|^2\left\|R_{jj}^{-1}R_{ij}R_{jj}^{-1}\right\|$$

$$+\left(\left\|\tilde{W}_i\right\|+W_{i\max}\right)\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|\sum_{j\in N_i}e_{ij}\left\|B_j\right\|^2\left\|R_{jj}^{-1}\right\|\left\|\frac{\partial\varphi_j}{\partial\delta_j}\right\|\left(\left\|\tilde{W}_j\right\|+W_{j\max}\right)+\tfrac{1}{2}\left\|\tilde{W}_i\right\|\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|\sum_{j\in N_i}e_{ij}\left\|B_j\right\|^2\left\|R_{jj}^{-1}\right\|\left\|\frac{\partial\phi_j}{\partial\delta_j}\right\|\left\|\tilde{W}_j\right\|$$

$$+\tfrac{1}{2}\left\|W_{i\max}\right\|\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|\sum_{j\in N_i}e_{ij}\left\|B_j\right\|^2\left\|R_{jj}^{-1}\right\|\left\|\frac{\partial\phi_j}{\partial\delta_j}\right\|\left(\left\|\tilde{W}_j\right\|+W_{j\max}\right)$$

$$+\tfrac{1}{2}\left(\left\|\tilde{W}_i\right\|+W_{i\max}\right)\left\|\frac{\partial\phi_i}{\partial\delta_i}\right\|\sum_{j\in N_i}e_{ij}\left\|B_j\right\|^2\left\|R_{jj}^{-1}\right\|\left\|\frac{\partial\phi_j}{\partial\delta_j}\right\|W_{j\max}+\bar{\varepsilon}_2 \tag{56}$$

All the signals on the right hand side of (56) are UUB and convergence to the approximate coupled HJ solution is obtained for every agent.

b. According to Theorem 6, $\left\|\hat{W}_{i+N}-W_i\right\|, \forall i$ are UUB. Then it is obvious that $\hat{u}_{i+N}, \forall i$ give the approximate cooperative Nash equilibrium (Definition 2).

■

6. Simulation results

This section shows the effectiveness of the online approach described in Theorem 6 for two different cases.

Consider the three-node strongly connected digraph structure shown in Figure 1 with a leader node connected to node 3. The edge weights and the pinning gains are taken equal to 1 so that $d_1 = d_2 = 1, d_3 = 2$.

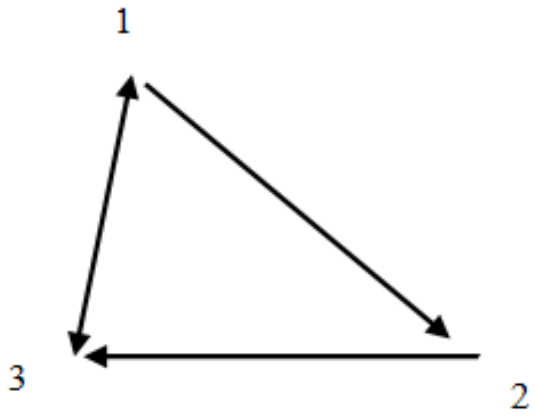

Fig. 1. Three agent communication graph showing the interactions.

Select the weight matrices in (9) as

$$Q_{11} = Q_{22} = Q_{33} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R_{11} = 4, R_{12} = 1, R_{13} = -1,$$
$$R_{31} = -4, R_{22} = 9, R_{23} = 1, R_{33} = 9, R_{32} = 1, R_{21} = 1$$

In the examples below, every node is a second-order system. Then, for every agent $\delta_i = \begin{bmatrix} \delta_{i1} & \delta_{i2} \end{bmatrix}^T$.

According to the graph structure, the information vector at each node is

$$z_1 = \begin{bmatrix} \delta_1^T & \delta_3^T \end{bmatrix}^T, z_2 = \begin{bmatrix} \delta_1^T & \delta_2^T \end{bmatrix}^T, z_3 = \begin{bmatrix} \delta_1^T & \delta_2^T & \delta_3^T \end{bmatrix}^T$$

Since the value is quadratic, the critic NNs basis sets were selected as the quadratic vector in the agent's components and its neighbors' components. Thus the NN activation functions are

$$\phi_1(\delta_1, 0, \delta_3) = \begin{bmatrix} \delta_{11}^2 & \delta_{11}\delta_{12} & \delta_{12}^2 & 0 & 0 & 0 & \delta_{31}^2 & \delta_{31}\delta_{32} & \delta_{32}^2 \end{bmatrix}^T$$
$$\phi_1(\delta_1, \delta_2, 0) = \begin{bmatrix} \delta_{11}^2 & \delta_{11}\delta_{12} & \delta_{12}^2 & \delta_{21}^2 & \delta_{21}\delta_{22} & \delta_{21}^2 & 0 & 0 & 0 \end{bmatrix}^T$$
$$\phi_3(\delta_1, \delta_2, \delta_3) = \begin{bmatrix} \delta_{11}^2 & \delta_{11}\delta_{12} & \delta_{12}^2 & \delta_{21}^2 & \delta_{21}\delta_{22} & \delta_{22}^2 & \delta_{31}^2 & \delta_{31}\delta_{32} & \delta_{32}^2 \end{bmatrix}^T$$

6.1 Position and velocity regulated to zero

For the graph structure shown, consider the node dynamics

$$\dot{x}_1 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} x_1 + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u_1, \dot{x}_2 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} x_2 + \begin{bmatrix} 2 \\ 3 \end{bmatrix} u_2, \dot{x}_3 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} x_3 + \begin{bmatrix} 2 \\ 2 \end{bmatrix} u_3$$

and the command generator $\dot{x}_0 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix} x_0$.

The graphical game is implemented as in Theorem 6. Persistence of excitation was ensured by adding a small exponentially decreasing probing noise to the control inputs. Figure 2 shows the convergence of the critic parameters for every agent. Figure 3 shows the evolution of the states for the duration of the experiment.

6.2 All the nodes synchronize to the curve behavior of the leader node

For the graph structure shown above consider the following node dynamics

$$\dot{x}_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_1 + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u_1, \dot{x}_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_2 + \begin{bmatrix} 2 \\ 3 \end{bmatrix} u_2, \dot{x}_3 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_3 + \begin{bmatrix} 2 \\ 2 \end{bmatrix} u_3$$

with target generator $\dot{x}_0 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x_0$.

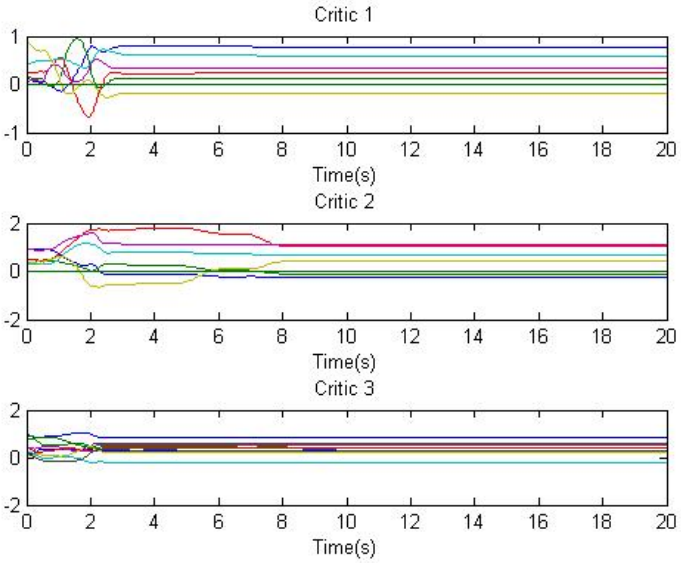

Fig. 2. Convergence of the critic parameters.

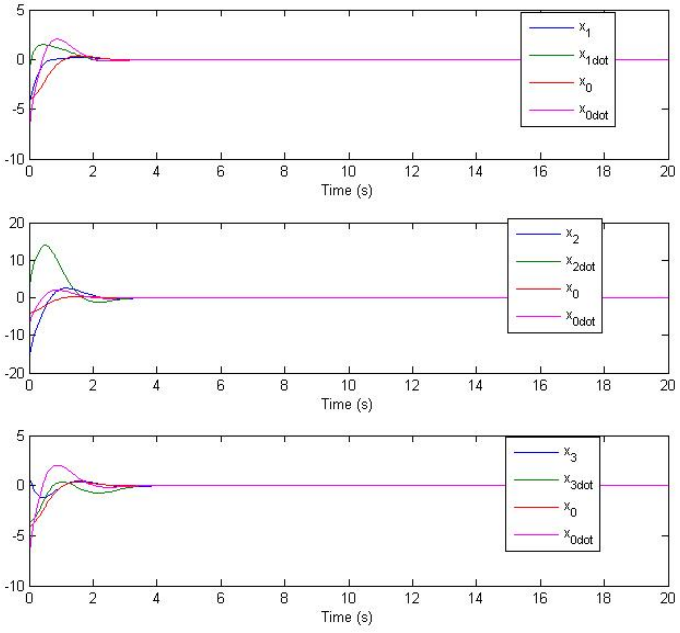

Fig. 3. Evolution of the system states and regulation.

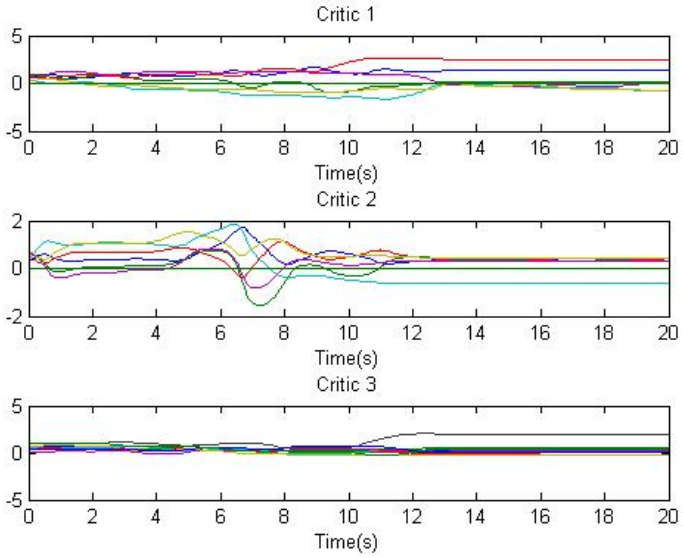

Fig. 4. Convergence of the critic parameters.

The command generator is marginally stable with poles at $s = \pm j$, so it generates a sinusoidal reference trajectory.

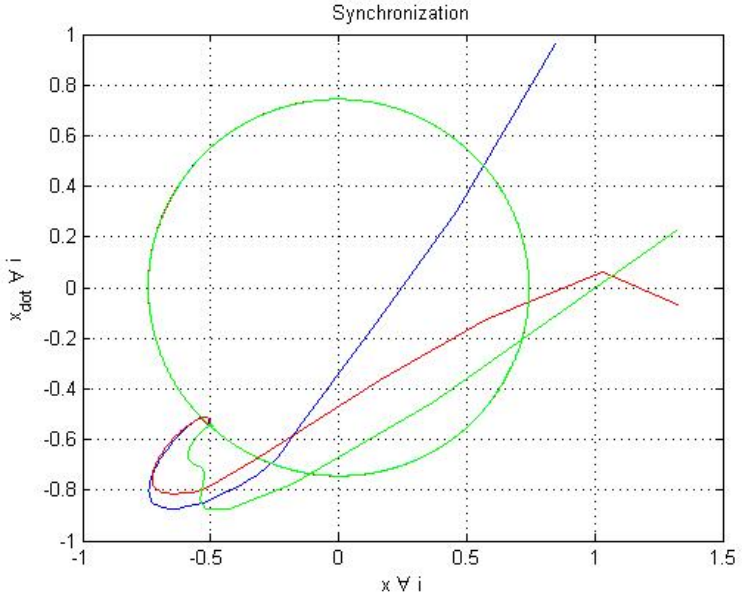

Fig. 5. Synchronization of all the agents to the leader node.

The graphical game is implemented as in Theorem 6. Persistence of excitation was ensured by adding a small exponential decreasing probing noise to the control inputs. Figure 4 shows the critic parameters converging for every agent. Figure 5 shows the synchronization of all the agents to the leader's behavior as given by the circular Lissajous plot.

7. Conclusion

This chapter brings together cooperative control, reinforcement learning, and game theory to solve multi-player differential games on communication graph topologies. It formulates graphical games for dynamic systems and provides policy iteration and online learning algorithms along with proof of convergence to the Nash equilibrium or best response. Simulation results show the effectiveness of the proposed algorithms.

8. References

Abou-Kandil H., Freiling G., Ionescu V., & Jank G., (2003). *Matrix Riccati Equations in Control and Systems Theory*, Birkhäuser.

Abu-Khalaf M., & Lewis F. L., (2005). Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach, *Automatica*, *41*(5), 779-791.

Başar T., & Olsder G. J.,(1999). *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA: SIAM.

Bertsekas D. P., & Tsitsiklis J. N. (1996). *Neuro-Dynamic Programming*, Athena Scientific, MA.

Brewer J.W., (1978). Kronecker products and matrix calculus in system theory, *IEEE Transactions Circuits and Systems*, *25*, *772-781*.

Busoniu L., Babuska R., & De Schutter B., (2008). A Comprehensive Survey of Multi-Agent Reinforcement Learning, *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, *38(2)*, *156–172*.

Dierks T. & Jagannathan S., (2010). Optimal Control of Affine Nonlinear Continuous-time Systems Using an Online Hamilton-Jacobi-Isaacs Formulation1, *Proc. IEEE Conf Decision and Control*, Atlanta, *3048-3053*.

Fax J. & Murray R., (2004). Information flow and cooperative control of vehicle formations, *IEEE Trans. Autom. Control*, *49(9)*, *1465–1476*.

Freiling G., Jank G., & Abou-Kandil H., (2002). On global existence of Solutions to Coupled Matrix Riccati equations in closed loop Nash Games, *IEEE Transactions on Automatic Control*, *41(2)*, *264- 269*.

Hong Y., Hu J., & Gao L., (2006). Tracking control for multi-agent consensus with an active leader and variable topology, *Automatica*, *42 (7)*, *1177–1182*.

Gajic Z., & Li T-Y., (1988). Simulation results for two new algorithms for solving coupled algebraic Riccati equations, *Third Int. Symp. On Differential Games*, Sophia, Antipolis, France.

Jadbabaie A., Lin J., & Morse A., (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules, *IEEE Trans. Autom. Control*, *48(6)*, *988–1001*.

Johnson M., Hiramatsu T., Fitz-Coy N., & Dixon W. E., (2010). Asymptotic Stackelberg Optimal Control Design for an Uncertain Euler Lagrange System, *IEEE Conference on Decision and Control*, *6686-6691*.

Kakade S., Kearns M., Langford J., & Ortiz L., (2003). Correlated equilibria in graphical games, *Proc. 4th ACM Conference on Electronic Commerce*, *42–47*.

Kearns M., Littman M., & Singh S., (2001) Graphical models for game theory, *Proc. 17th Annual Conference on Uncertainty in Artificial Intelligence*, *253–260*.

Khoo S., Xie L., & Man Z., (2009) Robust Finite-Time Consensus Tracking Algorithm for Multirobot Systems, *IEEE Transactions on Mechatronics*, *14*, *219-228*.

Leake R. J., Liu Ruey-Wen, (1967). Construction of Suboptimal Control Sequences, *J. SIAM Control*, *5 (1)*, *54-63*.

Lewis F. L., Syrmos V. L. (1995). *Optimal Control*, John Wiley.

Lewis F. (1992). *Applied Optimal Control and Estimation: Digital Design and Implementation*, New Jersey: Prentice-Hall.

Li X., Wang X., & Chen G., (2004). Pinning a complex dynamical network to its equilibrium, *IEEE Trans. Circuits Syst. I, Reg. Papers, 51(10)*, *2074–2087*.

Littman M.L., (2001). Value-function reinforcement learning in Markov games, *Journal of Cognitive Systems Research 1*.

Olfati-Saber R., Fax J., & Murray R., (2007). Consensus and cooperation in networked multi-agent systems, *Proc. IEEE*, vol. *95(1)*, *215–233*.

Olfati-Saber R., & Murray R.M., (2004). Consensus Problems in Networks of Agents with Switching Topology and Time-Delays, *IEEE Transaction of Automatic Control*, *49*, *1520-1533*.

Qu Z., (2009). *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*, New York: Springer-Verlag.

Ren W., Beard R., & Atkins E., (2005). A survey of consensus problems in multi-agent coordination, in *Proc. Amer. Control Conf.*, *1859–1864*.

Ren W. & Beard R., (2005). Consensus seeking in multiagent systems under dynamically changing interaction topologies, *IEEE Trans. Autom. Control*, *50(5)*, *655–661*.

Ren W. & Beard R.W., (2008) *Distributed Consensus in Multi-vehicle Cooperative Control*, Springer, Berlin.

Ren W., Moore K., & Chen Y., (2007). High-order and model reference consensus algorithms in cooperative control of multivehicle systems, *J. Dynam. Syst., Meas., Control*, *129(5)*, *678–688*.

R. Shinohara, "Coalition proof equilibria in a voluntary participation game," *International Journal of Game Theory*, vol. 39, no. 4, pp. 603-615, 2010.

Shoham Y., Leyton-Brown K., (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press.

Sutton R. S., Barto A. G. (1998) *Reinforcement Learning – An Introduction*, MIT Press, Cambridge, Massachusetts.

Tijs S (2003), *Introduction to Game Theory*, Hindustan Book Agency, India.

Tsitsiklis J., (1984). Problems in Decentralized Decision Making and Computation, Ph.D. dissertation, Dept. Elect. Eng. and Comput. Sci., MIT, Cambridge, MA.

Vamvoudakis Kyriakos G., & Lewis F. L., (2010). Online Actor-Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem, *Automatica*,46(5), 878-888.

Vamvoudakis K.G., & Lewis F. L., (2011). Multi-Player Non-Zero Sum Games: Online Adaptive Learning Solution of Coupled Hamilton-Jacobi Equations, to appear in *Automatica*.

Vrabie, D., Pastravanu, O., Lewis, F. L., & Abu-Khalaf, M. (2009). Adaptive Optimal Control for continuous-time linear systems based on policy iteration. Automatica, 45(2), 477-484.

Vrancx P., Verbeeck K., & Nowe A., (2008). Decentralized learning in markov games, *IEEE Transactions on Systems, Man and Cybernetics*, *38(4)*, *976-981.*

Wang X. & Chen G., (2002). Pinning control of scale-free dynamical networks, Physica A, *310(3-4)*, *521–531.*

Werbos P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavior Sciences*, Ph.D. Thesis.

Werbos P. J. (1992). Approximate dynamic programming for real-time control and neural modeling, *Handbook of Intelligent Control*, ed. D.A. White and D.A. Sofge, New York: Van Nostrand Reinhold.