2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Policy Iteration Algorithm for Distributed Networks and Graphical Games

Kyriakos G. Vamvoudakis, *Student* Member, *IEEE*, F. L. Lewis, *Fellow*, *IEEE*

Abstract—**This paper brings together cooperative control, reinforcement learning, and game theory to present a multi-agent distributed formulation for graphical games. The notion of *graphical games* is developed for dynamical systems, where the dynamics and performance indices for each node depend only on local neighbor information. We propose a cooperative policy iteration algorithm for graphical games. This algorithm converges to the best response when the neighbors of each agent do not update their policies and to the Nash equilibrium when all agents update their policies simultaneously. It is also shown that the convergence of this algorithm is based on the speed of convergence of the neighbors of every player in the graph, graph topology, and user defined matrices in the performance index. This framework will be used to develop methods for online adaptive learning solutions of graphical games in real time.**

Index Terms—**cooperative Hamilton-Jacobi equations, Policy Iteration, Nash-equilibrium, best response, graphical games.**

I. 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 [7], [11], [22], [23], [24], [25], [26] [31]. See [21], [24] 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 [9], [19], [27], [36]. 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 [30] 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 ([2], [8], [10]). 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 [29]. 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) [3], 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 [3], [37], [37]. 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 [34].

RL methods have been used to solve multiplayer games for finite-state systems in [5], [20]. RL methods have been applied to learn online in real-time the solutions for optimal control problems for dynamic systems and differential games in [6], [12], [32], [33].

This paper brings together cooperative control, reinforcement learning, and game theory to solve multi-player differential games on communication graph topologies. There are three main contributions in this paper. 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. 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

This work was supported by the National Science Foundation ECS-0801330, the Army Research Office W91NF-05-1-0314 and the Air Force Office of Scientific Research FA9550-09-1-0278.

K. G. Vamvoudakis and F. L. Lewis are with the Automation and Robotics Research Institute, University of Texas at Arlington, 7300 Jack Newell Blvd. S., Fort Worth, TX 76118 USA (phone/fax: +817-272-5938; e-mail: {kyriakos, lewis@arri.uta.edu}).

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 paper 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. 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.

II. SYNCHRONIZATION AND NODE ERROR DYNAMICS

A. 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 $D=[d_i]$ 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 paper.

B. Synchronization and Node Error Dynamics

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

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

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* $\delta_i\in\mathbb{R}^n$ [15]) 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 [19]. 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* [18] 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

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

where $\delta=\begin{bmatrix}\delta_1^T & \delta_2^T & \cdots & \delta_N^T\end{bmatrix}^T\in\mathbb{R}^{nN}$ 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 $\otimes$. $G\in R^{N\times N}$ 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 [22]) is

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

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 [15]. The next result therefore follows from (4) and the Cauchy Schwartz inequality and the properties of the Kronecker product [4].

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

$$\dot{\delta}_i = \sum_{j\in N_i} e_{ij}(\dot{x}_i - \dot{x}_j) \quad + g_i(\dot{x}_i - \dot{x}_0)$$

$$\dot{\delta}_i = \sum_{j\in N_i} e_{ij}(Ax_i + B_i u_i - (Ax_j + B_j u_j))$$

$$+ g_i(Ax_i + B_i u_i - Ax_0)$$

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

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.

III. 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 [2].

A. Cooperative Performance Index

Define the local performance indices

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

$$\equiv \tfrac{1}{2}\int_0^\infty L_i(\delta_i(t), u_i(t), u_{-i}(t))\, dt \qquad (9)$$

where $u_{-i}(t)$ is the vector of the control inputs $\{u_j : j \in N_i\}$ of the neighbors of node *i*, and u_{-i} denotes $\{u_{-i}(t) : 0 \le t\}$. All weighting matrices are constant and symmetric with $Q_{ii} > 0, R_{ii} > 0, R_{ij} \ge 0$. 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

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

where p_i is the co-state variable.

Necessary conditions [17] for a minimum of (9) are (1) and

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

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

B. Graphical Games and Nash Equilibrium

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

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

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

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

$$H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}, u_i, u_{-i}) \equiv \frac{\partial V_i}{\partial \delta_i}^T \left(A\delta_i + (d_i + g_i)B_i u_i - \sum_{j\in N_i} e_{ij} B_j u_j \right)$$
$$+\tfrac{1}{2}\delta_i^T Q_{ii}\delta_i + \tfrac{1}{2}u_i^T R_{ii} u_i + \tfrac{1}{2}\sum_{j\in N_i} u_j^T R_{ij} u_j = 0 \qquad (14)$$

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 1 that (14) is a Lyapunov equation. According to (13) and (10) one equates $p_i = \partial V_i / \partial \delta_i$.

The control objective of agent *i* is to determine

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

which corresponds to Nash equilibrium.

Definition 2. [2] **(**Global Nash equilibrium) An *N-tuple* of policies $\left\{u_1^*, u_2^*, ..., u_N^*\right\}$ is said to constitute a global Nash equilibrium solution for an *N* player game if for all $i \in N$

$$J_i^* \triangleq J_i(u_1^*, u_2^*, ..., u_i^*, ..., u_N^*) \le J_i(u_1^*, u_2^*, ..., u_i, ..., u_N^*) \; (16)$$

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

The distributed multiplayer game with local dynamics (8) and local performance indices (9) should be contrasted with standard multiplayer games [1], [2] which have centralized dynamics

$$\dot{z} = Az + \sum_{i=1}^N B_i u_i \qquad (17)$$

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 dynamics and performance index only depends on its own state, its control, and the controls of its immediate neighbors.

We want 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 [13], [14], [28]. 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.

Our intention in this paper is to provide algorithms 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 on 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.

The following notions are needed in the study of differential graphical games. Define $u_{-i}=\{u_j : j\in N_i\}$ as the set of policies of the neighbors of node *i*.

Definition 3. [28] Agent *i*'s *best response* to fixed policies u_{-i} of his neighbors is the policy u_i^* such that

$$J_i\,(u_i^*,u_{-i})\le J_i\,(u_i,u_{-i}) \qquad (18)$$

for all policies u_i of agent *i*.

For centralized multi-agent games, where the dynamics is given by (17) and the performance index of each agent depends on the actions of all other agents, an alternative definition of Nash equilibrium is that each agent is in best response to all other agents. However, in Definition 3 each node *i* is only in best response to all his neighbors.

C. Stability and Solution of Graphical Games

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

Assumptions 1.

a. The graph is strongly connected and at least one pinning gain g_i is nonzero. Then $(L+G)$ is nonsingular.

The game is well-formed in the sense that:

b. $B_j\ne 0 \rightleftharpoons e_{ij}\in E$.

c. $R_{ij}\ne 0 \rightleftharpoons e_{ij}\in E$.

Employing the stationarity condition (12) [17] one obtains the control policies

$$u_i=u_i(V_i)\equiv-(d_i+g_i)R_{ii}^{-1}B_i^T\frac{\partial V_i}{\partial\delta_i}\equiv -h_i(p_i) \quad (19)$$

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

$$\frac{\partial V_i}{\partial\delta_i}^T A_i^c+\tfrac{1}{2}\delta_i^TQ_{ii}\delta_i+\tfrac{1}{2}(d_i+g_i)^2\frac{\partial V_i}{\partial\delta_i}^T B_iR_{ii}^{-1}B_i^T\frac{\partial V_i}{\partial\delta_i}$$
$$+\tfrac{1}{2}\sum_{j\in N_i}(d_j+g_j)^2\frac{\partial V_j}{\partial\delta_j}^T B_jR_{jj}^{-1}R_{ij}R_{jj}^{-1}B_j^T\frac{\partial V_j}{\partial\delta_j}=0,\ i\in N \qquad (20)$$

where the closed-loop matrix is

$$A_i^c=A\delta_i-(d_i+g_i)^2B_iR_{ii}^{-1}B_i^T\frac{\partial V_i}{\partial\delta_i}$$
$$+\sum_{j\in N_i}e_{ij}(d_j+g_j)B_jR_{jj}^{-1}B_j^T\frac{\partial V_j}{\partial\delta_j},\ i\in N \qquad (21)$$

For a given V_i, define $u_i^*=u_i(V_i)$ as (19) given in terms of V_i. Then HJ equations (20) can be written as

$$H_i(\delta_i,\frac{\partial V_i}{\partial\delta_i},u_i^*,u_{-i}^*)=0 \qquad (22)$$

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 section 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.

For the global state δ given in (4) we can write the dynamics as

$$\dot\delta=(I_N\otimes A)\delta+diag(B_i)(L+G)\otimes I_nu \qquad (23)$$

where u is the control given by

$$u=-diag(R_{ii}^{-1}B_i^T)\big((D+G)\otimes I_np\big) \qquad (24)$$

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

$$-\dot p=\frac{\partial H}{\partial\delta}\equiv(I_N\otimes A)^Tp+diag(Q_{ii})\delta \qquad (25)$$

This is a set of coupled dynamic equations reminiscent of standard multi-player games [2] or single agent optimal control [17]. Therefore the solution can be written without any loss of generality as

$$p=\overline{P}\delta \qquad (26)$$

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

Lemma 2. HJ equations (20) are equivalent to the coupled Riccati equations

$$\delta^T\overline{P}^T\overline{A}_i\delta-\delta^T\overline{P}^T\overline{B}_i\overline{P}\delta+\tfrac{1}{2}\delta^T\overline{Q}_i\delta+\tfrac{1}{2}\delta^T\overline{P}^T\overline{R}_i\overline{P}\delta=0 \quad (27)$$

or equivalently

$$(\overline{P}^T\overline{A}_{ic}+\overline{A}_{ic}{}^T\overline{P}+\overline{Q}_i+\overline{P}^T\overline{R}_i\overline{P})=0 \qquad (28)$$

where $\overline{P}$ is defined by (26), and

$$\overline{A}_i=\begin{bmatrix}0&&&\\&0&&\\&&[A]^{ii}&\\&&&0\end{bmatrix}$$

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

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

$$\bar{Q}_i = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & [Q_{ii}]^{ii} & \\ & & & 0 \end{bmatrix},$$

$$\bar{R}_i = diag((d_i+g_i)B_i R_{ii}^{-1}) \begin{bmatrix} R_{i1} & & & & & \\ & \ddots & & & & \\ & & R_{ij} & & & \\ & & & \ddots & & \\ & & & & R_{ii} & \\ & & & & & R_{iN} \end{bmatrix}$$

$$diag((d_i+g_i)R_{ii}^{-1}B_i^T)$$

where $[\ ]^{ij}$ denotes the position of the element in the block matrix.

Proof:

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

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

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

$$+\tfrac{1}{2}\delta^T \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & [Q_{ii}]^{ii} & \\ & & & 0 \end{bmatrix} \delta + \tfrac{1}{2} \begin{bmatrix} u_1 \\ \vdots \\ u_i \\ u_N \end{bmatrix}^T \begin{bmatrix} R_{i1} & & & \\ & R_{ij} & & \\ & & R_{ii} & \\ & & & R_{iN} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_i \\ u_N \end{bmatrix} = 0 \tag{29}$$

By definition of the co-state one has

$$p \equiv \begin{bmatrix} \dfrac{\partial V_1}{\partial \delta_1} \\ \vdots \\ \vdots \\ \dfrac{\partial V_N}{\partial \delta_N} \end{bmatrix} = \bar{P}\delta \tag{30}$$

From the control policies (19), (29) becomes (27), which can be written in closed-loop form as (28).

■

Theorem 1. Stability and Solution for Nash Equilibrium.

Let $V_i > 0 \in C^1$, $i \in N$ be smooth solutions to HJ equations (20) and control policies u_i^*, $i \in N$ be given by (19) in terms of these solutions V_i. Then

a. Systems (8) are asymptotically stable.

b. u_i^*, u_{-i}^* are in Nash equilibrium and the corresponding game values are

$$J_i^*(\delta_i(0)) = V_i \ , i \in N \tag{31}$$

Proof:

a. If $V_i > 0$ satisfies (20) then it also satisfies (14). Take the time derivative to obtain

$$\dot{V}_i = \frac{\partial V_i}{\partial \delta_i}^T \dot{\delta}_i = \frac{\partial V_i}{\partial \delta_i}^T \left(A\delta_i + (d_i+g_i)B_i u_i - \sum_{j\in N_i} e_{ij}B_j u_j \right)$$

$$= -\tfrac{1}{2}\left(\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii} u_i + \sum_{j\in N_i} u_j^T R_{ij} u_j \right) \tag{32}$$

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

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

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

$$+V_i(\delta_i(0)) + \int_0^\infty \dot{V}_i dt$$

or

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

$$+V_i(\delta_i(0)) + \int_0^\infty \frac{\partial V_i}{\partial \delta_i}^T (A\delta_i + (d_i+g_i)B_i u_i - \sum_{j\in N_i} e_{ij}B_j u_j)dt$$

Now let V_i satisfy (20) and u_i^*, u_{-i}^* be the optimal controls given by (19). 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 (16).

■

IV. POLICY ITERATION SOLUTION 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 [37], [37]. RL techniques have also been applied for multiplayer games with centralized dynamics (17). See for example [5], [34]. 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.

A. Best Response

Theorem 1 reveals that the systems are in 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 (19) and arbitrary policies $u_{-i} = \{u_j : j \in N_i\}$

$$\begin{aligned} 0 = H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}, u_i^*, u_{-i}) \equiv \frac{\partial V_i}{\partial \delta_i}^T A_i^c + \tfrac{1}{2}\delta_i^T Q_{ii}\delta_i \\ + \tfrac{1}{2}(d_i + g_i)^2 \frac{\partial V_i}{\partial \delta_i}^T B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} + \tfrac{1}{2}\sum_{j\in N_i} u_j^T R_{ij} u_j \end{aligned} \qquad (33)$$

where the closed-loop matrix is

$$A_i^c = A\delta_i - (d_i + g_i)^2 B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} - \sum_{j\in N_i} e_{ij} B_j u_j \qquad (34)$$

Theorem 2. 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 (33) and let control policy u_i^* be given by (19) in terms of this solution V_i. Then

a. System (8) is asymptotically stable.

b. u_i^* is the best response to the fixed policies u_{-i} of its neighbors.

Proof:

a. $V_i > 0$ satisfies (33). Proof follows Theorem 1, 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

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

Now let V_i satisfy (33), u_i^* be the optimal controls given by (19) 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 (18).

■

B. Policy Iteration for Solution of Graphical games

The following algorithm for the *N*-player distributed games is motivated by the structure of policy iteration algorithms in reinforcement learning [3], [29], which rely on repeated policy evaluation (e.g. solution of (14)) and policy improvement (solution of (19)). 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 (20), 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)

$$H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_i^k, u_{-i}^k) = 0\, , \forall i = 1, \ldots, N \qquad (35)$$

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

$$u_i^{k+1} = -(d_i + g_i)R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i}^k, \forall i = 1, \ldots, N\, . \qquad (36)$$

Go to step 1.

On convergence End

■

The following two theorems prove convergence of the policy iteration algorithm for graphical 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 3. Convergence of Policy Iteration algorithm when only i^{th} agent updates its policy and all players u_{-i} in the 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 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 (33).

Proof:

It is clear that

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

$$\equiv \min_{u_i} H_i(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_i^k, u_{-i}^k) = H_i(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_i^{k+1}, u_{-i}^k) \tag{37}$$

Let $H_i(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_i^k, u_{-i}^k) = 0$ from (35) then according to (37) it is clear that

$$H_i^o(\delta_i, \frac{\partial V_i^k}{\partial \delta_i}, u_{-i}^k) \le 0 \tag{38}$$

Using the next control policy u_i^{k+1} and the current policies u_{-i}^k one has the orbital derivative [16]

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

From (37) and (38) one has

$$\begin{aligned} \dot{V}_i^k &= H_i^0(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_{-i}^k) - L_i(\delta_i, u_i^{k+1}, u_{-i}^k) \\ &\le -L_i(\delta_i, u_i^{k+1}, u_{-i}^k) \end{aligned} \tag{39}$$

Because only agent *i* update its control it is true that $u_{-i}^{k+1} = u_{-i}^k$ and $H_i(\delta_i, \frac{\partial V_i^{k+1}}{\partial \delta_i}, u_i^{k+1}, u_{-i}^k) = 0$.

But since $\dot{V}_i^{k+1} = -L_i(\delta_i, u_i^{k+1}, u_{-i}^{k+1})$, from (39) one has

$$\begin{aligned} \dot{V}_i^k &= H_i^0(\delta_i, \frac{\partial V_i}{\partial \delta_i}^k, u_{-i}^k) - L_i(\delta_i, u_i^{k+1}, u_{-i}^k) \\ &\le -L_i(\delta_i, u_i^{k+1}, u_{-i}^k) = \dot{V}_i^{k+1} \end{aligned} \tag{40}$$

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

$$V_i^{k+1} \le V_i^k \tag{41}$$

Since $V_i^* \le V_i^k$, the algorithm converges, to V_i^*, to the best response HJ equation (33).

■

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 4. 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}, μ_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

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

$$+\tfrac{1}{2} \sum_{j \in N_i} (u_j^{k+1} - \mu_j^k)^T R_{ij} (u_j^{k+1} - \mu_j^k)$$

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

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,

$$\dot{V}_i^k \le \dot{V}_i^{k+1} - \tfrac{1}{2} \sum_{j \in N_i} (u_j^{k+1} - u_j^k)^T R_{ij} (u_j^{k+1} - u_j^k)$$

$$+ \frac{\partial V_i}{\partial \delta_i}^{k+1T} \sum_{j \in N_i} e_{ij} B_j (u_j^{k+1} - u_j^k) - \sum_{j \in N_i} \mu_j^{kT} R_{ij} (u_j^{k+1} - u_j^k)$$

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

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

or

$$\tfrac{1}{2} \Delta u_j^T R_{ij} \Delta u_j - e_{ij} (p_i^{k+1})^T B_j \Delta u_j$$

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

and after taking norms one has

$$\tfrac{1}{2} \underline{\sigma}(R_{ij}) \left\| \Delta u_j \right\| > e_{ij} \left\| p_i^{k+1} \right\| \cdot \left\| B_j \right\| + (d_j + g_j) \rho_{ij} \left\| p_j^{k-1} \right\| \cdot \left\| B_j \right\|$$

where $\Delta u_j = (u_j^{k+1} - u_j^k)$, p_i the co-state 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. An alternative condition for convergence in Theorem 4 is that the norm $\|B_j\|$ should be small. This is similar to the case of weakly coupled dynamics in multi-player games in [2].

V. CONCLUSION

In this paper we have developed a multi-agent distributed formulation for graphical games, where 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. A policy iteration algorithm is proposed. This algorithm converges to the best response given fixed policies for the neighbors and to the Nash equilibrium given that all the agents update their policies simultaneously. Convergence depends on graph topology and user defined matrices.

REFERENCES

[1] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank, *Matrix Riccati Equations in Control and Systems Theory*, Birkhäuser, 2003.

[2] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA: SIAM, 1999.

[3] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, MA, 1996.

[4] J.W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Transactions Circuits and Systems*, vol. 25, 1978, pp. 772-781.

[5] L. Busoniu, R. Babuska, B. De Schutter, "A Comprehensive Survey of Multi-Agent Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics — Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156–172, 2008.

[6] T. Dierks and S. Jagannathan, Optimal Control of Affine Nonlinear Continuous-time Systems Using an Online Hamilton-Jacobi-Isaacs Formulation1, Proc. IEEE Conf Decision and Control, Atlanta, pp. 3048-3053, 2010.

[7] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.

[8] G. Freiling, G. Jank, H. Abou-Kandil, "On global existence of Solutions to Coupled Matrix Riccati equations in closed loop Nash Games," *IEEE Transactions on Automatic Control*, vol. 41, no. 2, pp. 264- 269, 2002.

[9] Y. Hong, J. Hu, and L. Gao, "Tracking control for multi-agent consensus with an active leader and variable topology," Automatica, vol. 42, no. 7,pp. 1177–1182, 2006.

[10] Z. Gajic and T-Y. Li, "Simulation results for two new algorithms for solving coupled algebraic Riccati equations," *Third Int. Symp. On Differential Games*, Sophia, Antipolis, France, June 1988.

[11] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[12] M. Johnson, T. Hiramatsu, N. Fitz-Coy, and W. E. Dixon, "Asymptotic Stackelberg Optimal Control Design for an Uncertain Euler Lagrange System," IEEE Conference on Decision and Control, pp. 6686-6691, 2010

[13] S. Kakade, M. Kearns, J. Langford, and L. Ortiz, "Correlated equilibria in graphical games," *Proc. 4th ACM Conference on Electronic Commerce*, pp. 42–47, 2003.

[14] M. Kearns, M. Littman, and S. Singh. "Graphical models for game theory," *Proc. 17th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 253–260, 2001.

[15] S. Khoo, L. Xie, and Z. Man, "Robust Finite-Time Consensus Tracking Algorithm for Multirobot Systems," *IEEE Transactions on Mechatronics*, vol. 14, pp. 219-228.

[16] R. J. Leake, Ruey-Wen Liu, "Construction of Suboptimal Control Sequences," *J. SIAM Control*, vol. 5, no, 1, pp. 54-63, 1967.

[17] F. L. Lewis, V. L. Syrmos, *Optimal Control*, John Wiley, 1995.

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

[19] X. Li, X. Wang, and G. Chen, "Pinning a complex dynamical network to its equilibrium," *IEEE Trans. Circuits Syst. I, Reg. Papers,* vol. 51, no. 10, pp. 2074–2087, Oct. 2004.

[20] M.L. Littman, "Value-function reinforcement learning in Markov games," Journal of Cognitive Systems Research 1, 2001.

[21] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[22] R. Olfati-Saber and R.M. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Transaction of Automatic Control*, vol. 49, pp. 1520-1533, 2004.

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

[24] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proc. Amer. Control Conf.*, Portland, OR, pp. 1859–1864, 2005.

[25] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[26] W. Ren and R.W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*, Springer, Berlin, 2008.

[27] W. Ren, K. Moore, and Y. Chen, "High-order and model reference consensus algorithms in cooperative control of multivehicle systems," J. Dynam. Syst., Meas., Control, vol. 129, no. 5, pp. 678–688, 2007.

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

[29] R. S. Sutton, A. G. Barto, *Reinforcement Learning – An Introduction*, MIT Press, Cambridge, Massachusetts, 1998.

[30] S. Tijs, *Introduction to Game Theory*, Hindustan Book Agency, India, 2003.

[31] J. Tsitsiklis, "Problems in Decentralized Decision Making and Computation," Ph.D. dissertation, Dept. Elect. Eng. and Comput. Sci., MIT, Cambridge, MA, 1984.

[32] K.G. Vamvoudakis, and F. L. Lewis, "Online Actor-Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem," *Automatica*, vol. 46, no. 5, pp. 878-888, 2010.

[33] K.G. Vamvoudakis, and F. L. Lewis, "Multi-Player Non-Zero Sum Games: Online Adaptive Learning Solution of Coupled Hamilton-Jacobi Equations," *Automatica*, vol. 47, no. 8, pp. 1556-1569, 2011.

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

[35] P. Vrancx, K. Verbeeck, and A. Nowe, "Decentralized learning in markov games," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 38, no. 4, pp. 976-981, August 2008.

[36] X. Wang and G. Chen, "Pinning control of scale-free dynamical networks," Physica A, vol. 310, no. 3-4, pp. 521–531, 2002.

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

[38] P. J. Werbos, "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, 1992.