Distributed Games for Multi-Agent Systems: Games on Communication Graphs

K. G. Vamvoudakis[1], **D. G. Mikulski[2,3]**, G. R. Hudas[3], F. L. Lewis[1], E. Y. Gu[2]

27th Army Science Conference
Orlando, FL

Thursday, December 2, 2010

1 University of Texas-Arlington, Automation & Robotics Research Institute

2 U.S. Army RDECOM-TARDEC

3 Oakland University, School of Engineering

<u>Supported by</u>: ARO grant W91NF-05-1-0314
and the U.S. Army National Automotive Center

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **02 DEC 2010**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Distributed Games for Multi-Agent Systems: Games on Communication Graphs**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **K. G. Vamvoudakis; D. G. Mikulski; G. R. Hudas, F. L. Lewis1; E. Y. Gu**	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Texas-Arlington, Automation & Robotics Research Institute; U.S. Army RDECOM-TARDEC; Oakland University, School of Engineering**	8. PERFORMING ORGANIZATION REPORT NUMBER **21391RC**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **US Army RDECOM-TARDEC 6501 E 11 Mile Rd Warren, MI 48397-5000, USA**	10. SPONSOR/MONITOR'S ACRONYM(S) **TACOM/TARDEC** 11. SPONSOR/MONITOR'S REPORT NUMBER(S) **21391RC**

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at 27th Army Science conference (ASC), 29 November 2 December 2010 Orlando, Florida, USA, The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**31**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Motivation

- U.S. Army currently wages asymmetric battles against insurgencies
- Enemy is hard to detect
  - Knowledge of local terrain
  - Ability to mix in with the civilian population
- Enemy quickly adapts to Army tactics and strategies

Motivation (cont)

- The needs of Soldiers change in response to new insurgent strategies
- Real-time adaptive team responses to insurgent threats are key to mitigate the risk in uncertain and dynamic battle spaces

Research Objective

- Goal: Develop ways for teams to learn optimal game strategies, even under changing mission requirements and team objectives

- Problem: Centralized formulation of multi-agent games is complex and needs global data. **Can we decentralize the dynamics in multi-agent games and still achieve optimal performance?**

Outline

- Background Information
  - Game Theory for Multi-Agent Systems (MAS)
  - Graph Theory for Communication Graphs
  - Synchronization Control Design Problem
- Cooperative Optimal Control
  - Local Performance Functions for Team Behaviors
  - Distributed Hamilton-Jacobi (HJ) Equation
- Multi-Agent Game Distributed Solution
  - Reinforcement Learning Solution
  - Online Solution using Neural Networks
  - Simulation Results

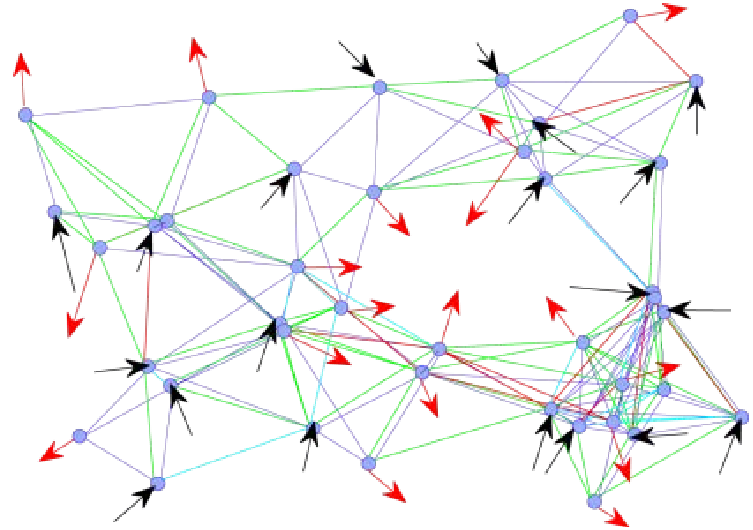
Background Information

Game Theory for MAS

- MAS comprised of autonomous agents that cooperate to meet a system-level objective
- Game Theory used to model the strategic behavior of MAS
  - Outcomes depend not only an agent's own actions, but also the actions of every other agent
  - Each agent chooses a strategy that independently optimizes his own performance objectives without the knowledge of other agent strategies
- Team decisions normally solved offline
  - Coupled Riccati equations for linear systems
  - Coupled Hamilton-Jacobi equations non-linear systems

Graphs for Communications

- Consider a graph Gr=(V,E) with:
  - Nonempty set of N agents $V=\{v_1,\ldots,v_N\}$
  - Set of edges $E\subseteq VxV$
  - Connectivity matrix $E=[e_{ij}]$
  - Set of neighbors N_i
  - In degree matrix is denoted as $D=[d_i]=[\sum_{j\in N_i} e_{ij}]$



- Define the graph Laplacian: $L=D-E$

- If the graph is strongly connected: no permutation matrix such that: $L=U\begin{bmatrix} * & 0 \\ * & * \end{bmatrix}U^T$

Synchronization Problem

- Consider N agents on Gr with dynamics
  $\dot{x}_i = Ax_i + B_i u_i, x_i(t) \in \mathbb{R}^n, u_i(t) \in \mathbb{R}^{m_i}, A(t) \in \mathbb{R}^{nxn}, B(t) \in \mathbb{R}^{m_i xn}$

- Target node is $x_0(t) \in \mathbb{R}^n$, which satisfies the dynamics: $\dot{x}_0 = Ax_0$

- <u>Synchronization Problem</u>: design local control protocols for all agents in Gr to synch to target node. $x_i(t) \to x_0(t), \forall i$

Synchronization Problem (cont)

- Cooperative team objectives can be described in terms of the *local neighborhood tracking error (LNTE)*

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

- Dynamics of the LNTE

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

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

Cooperative Optimal Control

- Multi-Agent Games on Graphs

Local Cost Function for Teams

- Goal: To achieve synchronization while optimizing some performance measures on the agents

- Local Cost Function

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

$$Q_{ii} \geq 0,\ R_{ii} > 0,\ R_{ij} \geq 0$$

Local Value and Hamiltonian

- Let us interpret the control input as policies/strategies

- Local Value Function

$$V_i(\delta_i(t),\delta_{-i}(t)) = \int_t^{\infty} (\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii}u_i + \sum_{j\in N_i} u_j^T R_{ij}u_j)\, dt$$

- Local Hamiltonian Function

$$H_i(\delta_i, u_i, u_{-i}) \equiv \frac{\partial V_i}{\partial \delta_i}^T \left(A\delta_i + (d_i + g_i)B_i u_i - \sum_{j\in N_i} e_{ij} B_j u_j \right)$$

$$+\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii}u_i + \sum_{j\in N_i} u_j^T R_{ij}u_j = 0$$

Local Nash Equilibrium

- The control objective of agent *i* is to find the optimal strategy and smallest value:

$$V_i^*(\delta_i(t), \delta_{-i}(t)) = \min_{u_i} \int_t^{\infty} (\delta_i^T Q_{ii} \delta_i + u_i^T R_{ii} u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j)\, dt$$

- Nash equilibrium solution for a finite *N*–agent distributed game is an *N*–tuple of strategies where:

$$J_i^* \triangleq J_i(\mu_i^*, \mu_{-i}^*) \le J_i(\mu_i, \mu_{-i}^*), i \in N$$

Distributed HJ Equation

- Using the stationarity condition $\partial H_i / \partial u_i = 0$ to find the optimal control:

$$u_i = -\tfrac{1}{2}(d_i + g_i) R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} \equiv -h_i\left(\frac{\partial V_i}{\partial \delta_i}\right)$$

- Substitute into Hamiltonian to get distributed Hamilton–Jacobi (HJ) equation

$$\frac{\partial V_i}{\partial \delta_i}^T \left(A\delta_i - \tfrac{1}{2}(d_i + g_i)^2 B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i} + \tfrac{1}{2} \sum_{j \in N_i} e_{ij}(d_j + g_j) B_j R_{jj}^{-1} B_j^T \frac{\partial V_j}{\partial \delta_j} \right)$$
$$+ \delta_i^T Q_{ii} \delta_i + \tfrac{1}{4}(d_i + g_i)^2 \frac{\partial V_i}{\partial \delta_i}^T B_i R_{ii}^{-1} B_i^T \frac{\partial V_i}{\partial \delta_i}$$
$$+ \tfrac{1}{4} \sum_{j \in N_i} (d_j + g_j)^2 \frac{\partial V_j}{\partial \delta_j}^T B_j R_{jj}^{-1} R_{ij} R_{jj}^{-1} B_j^T \frac{\partial V_j}{\partial \delta_j} = 0,\ i \in N$$

Distributed HJ Equation (cont)

- There is one coupled HJ equation corresponding to each agent.
- Therefore, a solution to this multi-agent game problem requires a solution to N coupled partial differential equations.
- **Next, we show how to solve this online in a distributed way**
  - Each agent requires only information from neighbors
  - Use techniques from reinforcement learning

Distributed Solution of the Multi-Agent Game

- Using Reinforcement Learning

Reinforcement Learning (RL)

- RL is concerned with how to methodically modify the actions of an agent based on observed responses from its environment.
- In game theory, RL is considered a bounded rational interpretation of how equilibrium may arise.
- One technique that has been developed from RL research in controls is *Policy Iteration* (PI)

Policy Iteration (PI)

- A class of two-step iteration algorithms: *policy evaluation* and *policy improvement*
  - Evaluation: Apply a control. Evaluate the benefit of that control.
  - Improvement: Improve the control policy.

- In control theory, PI algorithms amount to:
  - Learning the solution to a non-linear Lyapunov equation
  - Updating the policy by minimizing a Hamiltonian function

Offline PI Algorithm

- To solve the multi-agent game in a distributed way, the value functions must be parameterized.

- However, in our case, it is not clear what parametric form the value should take in the Hamiltonian.

- The value function needs to be in terms of local variables in order to use a local solution procedure

Offline PI Algorithm (cont)

- Step 0: Start with stabilizing initial policies

$$u^0{}_1(x),\ldots,u^0{}_N(x)$$

- Step 1: Given the N–tuple of policies, solve for the costs $V^k{}_1, V^k{}_2 \ldots, V^k{}_N$

$$0=\delta_i^T Q_{ii}\delta_i + u_i^T R_{ii}u_i + \sum_{j\in N_i} u_j^T R_{ij}u_j + \left(\frac{\partial V_i^k}{\partial \delta_i}\right)^T \left(A\delta_i + (d_i+g_i)B_i u_i - \sum_{j\in N_i} e_{ij}B_j u_j \right)$$

$$V^k{}_i(0)=0 \qquad i\in N$$

Offline PI Algorithm (cont)

- Step 2: Update the N−tuple control policies by trying to minimize the Hamiltonian:

$$u_i^{k+1}(x) = -\tfrac{1}{2}(d_i + g_i)R_{ii}^{-1}B_i^T \frac{\partial V_i^k}{\partial \delta_i} \qquad i \in N$$

- Step 3: Increment k and repeat to Step 1 until convergence

Online Solution using Neural Nets

- Online solution uses an Actor–Critic method
  - Actor: selects the policy of the agent
  - Critic: criticizes the policy of the actor

- The output of the Critic drives the learning for both the Actor and Critic

- In this solution, Actors and Critics are neural networks (NNs)
  - Approximate value functions and their gradients
  - Use proper approximator structures

Value Function Approximator (VFA)

- Assumption: For each admissible policy, the non-linear Lyapunov equations have smooth solutions

$$V_i(\bar{\delta}_i) \geq 0, \quad \bar{\delta}_i = \begin{bmatrix} \delta_i & \delta_{-i} \end{bmatrix}$$

- Critic NN

$$\hat{V}_i(\bar{\delta}_i) = \hat{W}_i^T \phi_i(\bar{\delta}_i)$$

- Actor NN

$$\hat{u}_{i+N} = -\frac{1}{2}(d_i + g_i) R_{ii}^{-1} B_i^T \nabla \phi_i^T \hat{W}_{i+N}$$

Online Cooperative Games

- Update Critic: learn the value

$$\dot{\hat{W}}_i = -a_i \frac{\sigma_{i+N}}{(\sigma_{i+N}^T \sigma_{i+N} + 1)^2} [\sigma_{i+N}{}^{\mathrm{T}} \hat{W}_i + \delta_i^T Q_{ii} \delta_i + \tfrac{1}{4} \hat{W}_{i+N}{}^T \bar{D}_i \hat{W}_{i+N}$$

$$+ \tfrac{1}{4} \sum_{j \in N_i} (d_j + g_j)^2 \hat{W}_{j+N}{}^T \nabla \varphi_j B_j R_{jj}^{-T} R_{ij} R_{jj}^{-1} B_j{}^T \nabla \varphi_j{}^T \hat{W}_{j+N}]$$

- Update Actor: learn the control policy

$$\dot{\hat{W}}_{i+N} = -\alpha_{i+N} \{ (F_{i+1} \hat{W}_{i+N} - F_i \bar{\sigma}_{i+N}^T \hat{W}_i) - \frac{1}{4} \bar{D}_i \hat{W}_{i+N} \frac{\bar{\sigma}_{i+N}}{m_{si}}^T \hat{W}_i$$

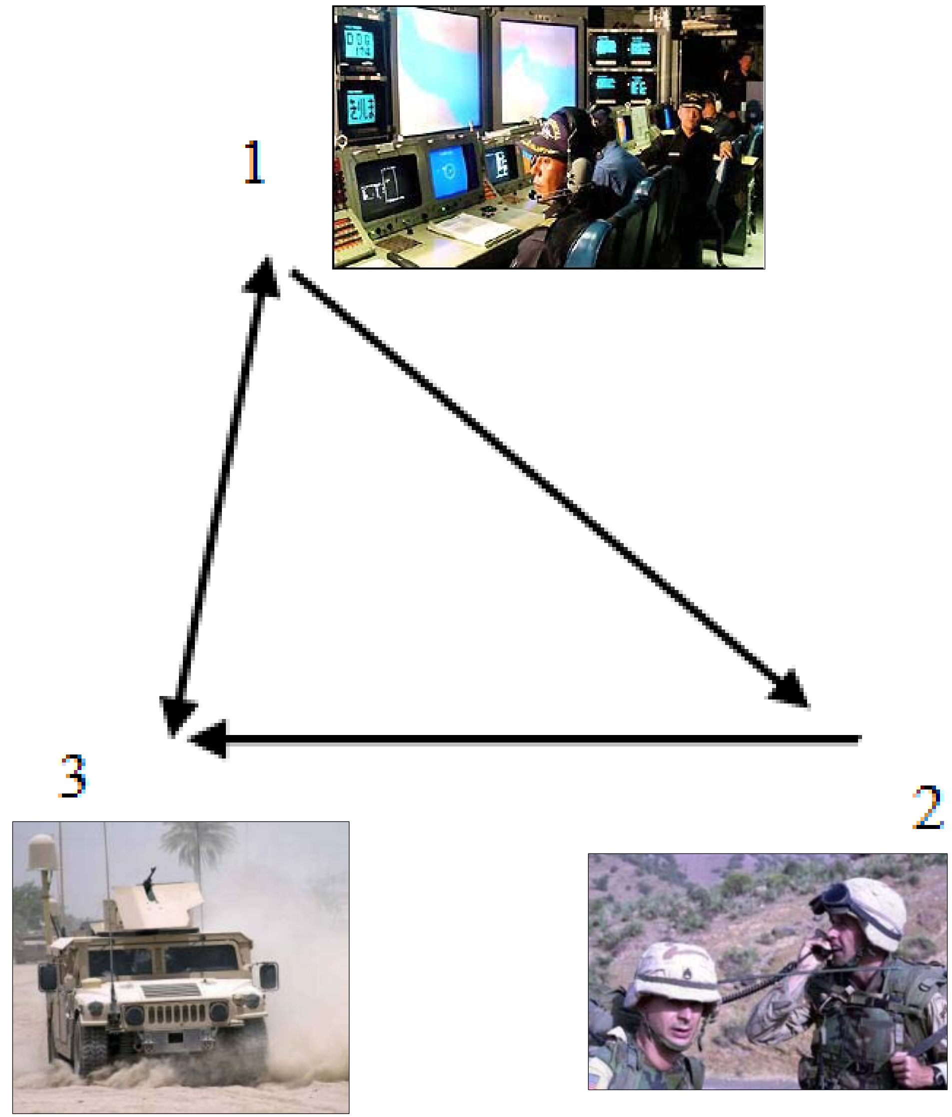
$$- \frac{1}{4} \hat{W}_{i+N}{}^T \sum_{\substack{j \in N_i \\ j \neq i}} (d_j + g_j)^2 \hat{W}_j \frac{\bar{\sigma}_{i+N}}{m_{si+N}}^T \nabla \varphi_j B_j R_{jj}^{-T} R_{ij} R_{jj}^{-1} B_j{}^T \nabla \varphi_j{}^T$$

Some Remarks for Online Solution

- We have provided the base for tuning the actor/critic network of N agents at the same time, meaning that teams can learn online in real time.
- Persistence of excitation is need for the proper identification of the value functions by the **Critic NN**
- Nonstandard tuning algorithms are required to guarantee stability for the **Actor NN**
- NN usage suggest starting with random, non-zero control weights

Simulation

- Node 2 can receive orders from Node 1
- Node 2 does not have a transmitter strong enough to acknowledge the order directly.
- Thus Node 2 must use a router (Node 3), which under a security protocol, cannot acknowledge Node 2 directly.

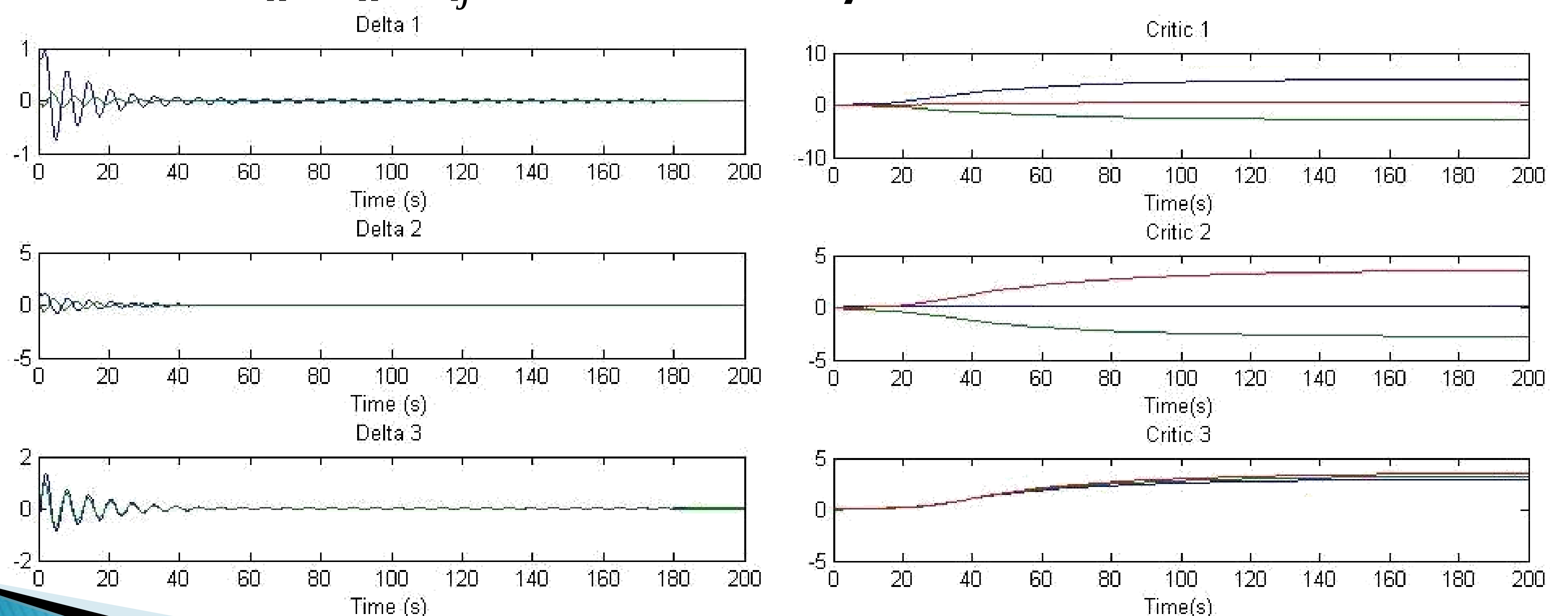


Simulation Results

- Node Dynamics

$$\dot{x}_1 = \begin{bmatrix} -1 & -2 \\ 1 & -4 \end{bmatrix} x_1 + \begin{bmatrix} 2 \\ -1 \end{bmatrix} u_1 \quad \dot{x}_2 = \begin{bmatrix} -1 & -2 \\ 1 & -4 \end{bmatrix} x_2 + \begin{bmatrix} 1 \\ -3 \end{bmatrix} u_2 \quad \dot{x}_3 = \begin{bmatrix} -1 & -2 \\ 1 & -4 \end{bmatrix} x_3 + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u_3$$

- Select Q_{ii}, R_{ii}, R_{ij} as identity matrices. Results:



Summary

- Posed the Synchronization Control Problem
- Derived the distributed Hamilton–Jacobi equation in terms of local value functions
- Proposed distributed solutions to the Multi-Agent Game
  - Offline Policy Iteration Algorithm
  - Online Solution using Actor/Critic NNs

Future Work

- Develop more simulations using more agents in time-varying graphs

- Extend the results of this research to graphs with a spanning tree (i.e. not necessarily strongly connected)

- Incorporate concepts of trust into cooperative multi-agent systems

Questions?

Kyriakos G. Vamvoudakis
kyriakos@arri.uta.edu

Dariusz G. Mikulski
dgmikuls@oakland.edu, dariusz.mikulski@us.army.mil

Dr. Greg R. Hudas
greg.hudas@us.army.mil

Dr. Frank L. Lewis
lewis@uta.edu

Dr. Edward Y. Gu
guy@oakland.edu