Distributed learning algorithm for non-linear differential graphical games

Farzaneh Tatari¹, Mohammad-Bagher Naghibi-Sistani¹ and Kyriakos G. Vamvoudakis²

Abstract
This paper introduces differential graphical games for continuous-time non-linear systems and proposes an online adaptive learning framework. The error dynamics and the user-defined performance indices of each agent depend only on local information and the proposed cooperative learning algorithm learns the solution to the cooperative coupled Hamilton–Jacobi equations. In the proposed algorithm, each one of the agents uses an actor/critic neural network (NN) structure with appropriate tuning laws in order to guarantee closed-loop stability and convergence of the policies to the Nash equilibrium. Finally, a simulation example verifies the effectiveness of the proposed approach.

Keywords

Introduction
Multi-agent systems (MASs) provide a framework for solving problems that cannot be solved by a single agent. Such applications include formation of mobile robots (Deoort et al., 2008), synchronization of robot manipulators (Cui and Yan, 2012; Li et al., 2013), unmanned air vehicle (UAV) formation flying (Lin, 2014), attitude synchronization in spacecraft (Abdessameud and Tayebi, 2009; Ren, 2007). Consensus, or leader synchronization, are two of the main problems in MASs where one desires to design a simple control law for every agent in order to provide a global behaviour. Cooperative control can be regarded as a regulation problem or as a tracking problem. In regulation, also known as leaderless consensus, the designed distributed controllers drive all agents to an unsupervised common value (Rong et al., 2012; Xie and Chen, 2013; Zhang et al., 2012b). This common or consensus value is a function of the initial states of the agents in the communication network (Ren et al., 2007a).

In optimization-based cooperative control, agents generally minimize their own cost functions to find optimal policies in order to guarantee optimization and synchronization. Semsar-Kazerooni and Khorasani (2009) have proposed an output consensus framework for linear MAS using a cooperative game theory approach. Wang and Xin (2012) employed inverse optimal control strategy to present a distributed optimal control approach for cooperative tracking with guaranteed closed-loop stability. An inverse optimal control approach has also been used in Mao et al. (2011) in order to construct a decentralized optimal control strategy to reach consensus in a leaderless structure, which has the advantage to handle high relative degree systems. Shi et al. (2011) proposed an optimal consensus algorithm within the global solution set of a sum of non-linear objective functions. All the proposed previous methods are applied to linear MAS and require offline calculations to find the optimal solution. Therefore, there has been no research regarding the synchronization of MASs with non-linear dynamics while also guaranteeing optimality.

Game theory (Isaacs, 1965; Tijs, 2003) lets us solve optimization problems for systems with more than one decision.

¹Department of Electrical Engineering, Ferdowsi University of Mashhad, P.O. Box: 9177948974, Mashhad, Islamic Republic of Iran. Corresponding author: Mohammad-Bagher Naghibi-Sistani, Department of Electrical Engineering, Ferdowsi University of Mashhad, P.O. Box: 9177948974, Mashhad, Islamic Republic of Iran. Email: mb-naghibi@um.ac.ir

²Center for Control, Dynamical-systems and Computation (CCDC), University of California, Santa Barbara, CA 93106-9560, USA.
maker or controller. This is achieved by modelling the strategic behaviour of the agents where the outcome of each player depends on the actions of herself and the rest of the players (Esparza et al., 2013; Starr and Ho, 1969; Tolwinski et al., 1986). Non-linear differential games require the solution of coupled Hamilton–Jacobi (HJ) equations, which are difficult or impossible to solve, and may not have global analytical solutions, even in simple cases. Generally, the employed methods for approximating the coupled HJ equations are offline and require complete knowledge of the system dynamics. Online stabilizing controllers can be designed by adaptive control methods (Ioannou and Fidan, 2006; Sastry and Bodson, 1989; Slotine and Li, 1991), but without guaranteeing any level of optimality. In our work, we shall use ideas from reinforcement learning (RL) to provide solution to these difficult optimization problems. RL is concerned with modifying an agent’s actions based on the responses received from the environment (Lewis and Vrabie, 2009; Powell, 2007; Sutton and Barto, 1998; Tan et al. 2014). In other words, RL is a means of learning optimal behaviours by observing the response from the environment to non-optimal control policies. In game theory, RL can be considered a bounded rational interpretation of how equilibrium may arise.

In order to solve approximately the non-linear partial differential equation (PDEs) arising from the coupled HJ equations, neural network (NN) approximators, (Abu-Khalaf and Lewis, 2005), are used in the context of approximate/adaptive dynamic programming (ADP) (Murray et al. 2002; Vrabie et al., 2008; Werbos, 1992). A way to solve ADP is to use actor–critic structures (Barto et al., 1983). The actor NN approximates the control policy while the critic NN assesses the value of that action. Based on this value assessment, the algorithm finds another improved control policy. Moreover, actor–critic structures have been employed to solve the multi-agent non-zero-sum (NZS) differential games online. In Vamvoudakis and Lewis (2011), an online adaptive control algorithm based on policy iteration (PI) and actor–critic structures are used to solve the continuous-time (CT) multi-player NZS games. Zhang et al. (2012a) proposed an ADP network in the form of a single critic network instead of an actor–critic dual network to obtain the Nash equilibrium of NZS differential games with known dynamics. In Vrabie and Lewis (2010), an ADP algorithm finds the Nash equilibrium for a two-player linear NZS differential game. The work of Vamvoudakis and Lewis (2012) introduced an online algorithm that uses integral reinforcement learning (IRL) and actor–critic structures to learn the CT Nash equilibrium solutions for non-linear systems with partial knowledge of the system dynamics. Kamalapurkar et al. (2013) developed an ADP-based approach to solve the cooperative regulation problem for a non-linear control-affine network topology where an identifier is used in conjunction with the controller to find approximate optimal policies online without the knowledge of drift dynamics. In order to avoid centralized information that used in the aforementioned papers, the work of Vamvoudakis et al., 2012 introduced the concept of differential graphical games for linear systems with known dynamics and solved the coupled HJ equations by employing an online adaptive learning algorithm while also guaranteeing closed-loop stability and convergence to the Nash equilibrium.

Contributions

In our work, in order to use only local information, while we guarantee global synchronization to the leader behaviour, we shall use the theory of graphical games where the performance indices to be optimized are totally distributed. To our knowledge, there has not been any work on multi-agent non-linear differential graphical games and their online optimal solution. Therefore, the main contribution of this paper is the introduction of a multi-agent non-linear differential graphical game that solves the global synchronization problem by using only local control inputs for each agent. Herein, the agent dynamics are heterogeneous and a general value function is considered for each agent. Each player maintains a critic and an actor NN to learn respectively the optimal value and optimal control policy. We derive gradient descent base tuning laws in order to tune the critic and actor NNs simultaneously online. Finally, all the closed-loop signals are bounded by using Lyapunov stability theory and the policies of all the agents form a Nash equilibrium.

Notation. Throughout the paper, \( \mathbb{R} \) denotes the real numbers, \( \mathbb{R}^n \) denotes the real n vectors and \( \mathbb{R}^{m \times n} \) denotes the real \( m \times n \) matrices. For a vector \( x \), \( \|x\| \) indicates the Euclidean norm of \( x \). For a matrix \( M \), \( \|M\| \) indicates the induced 2-norm of \( M \) and \( \sigma_{\text{min}}(M) \) denotes the minimum eigenvalue of \( M \). The topology of information exchange between N agents is described by a graph \( G = (V, \Sigma) \), where \( V = \{1, 2, ..., N\} \) is the set of vertices representing \( N \) agents and \( \Sigma \subseteq V \times V \) is the set of edges of the graph. \( (i,j) \in \Sigma \), means there is an edge from node \( i \) to node \( j \).

Background on graphs. We assume that the communication graph is simple, e.g. no repeated edges and no self-loops. The topology of a graph is often represented by an adjacency matrix \( E = [e_{ij}] \in \mathbb{R}^{N \times N} \) with \( e_{ij} = 1 \) if \( (j, i) \in \Sigma \) and \( e_{ij} = 0 \) otherwise. Note that when \( (i,j) \notin \Sigma \), \( e_{ij} = 0 \). The set of neighbours of a node \( i \) is \( N_i = \{j : (j, i) \in \Sigma \} \), i.e. the set of nodes with arcs incoming to \( i \). If the node \( j \) is a neighbour of node \( i \), then the node \( i \) can get information from node \( j \), though not necessarily vice versa for directed graphs. The diagonal in-degree matrix is defined as \( D = \text{diag}(d_i) \in \mathbb{R}^{N \times N} \) with \( d_i = \sum_{j \in \Sigma} e_{ij} \), the weighted in-degree of node \( i \), i.e. \( i \)th row sum of \( E \). A graph is connected if there is a path between every pair of vertices. The leader is represented by vertex 0 and information is sent from the leader to the agents for which the leader is in their neighbourhood (Figure 1).

Structure. The paper is organized as follows. The next section introduces multi-player differential graphical games for non-linear systems. Then the online learning algorithm and online solution for the multi-agent non-linear differential graphical games are given. Simulation results are discussed and finally, we provide conclusions and talk about future work.
Problem formulation

Consider the dynamics of each agent in a directed and strongly connected communication graph (e.g. Figure 1) as

$$\dot{x}_i(t) = f_i(x_i(t)) + g_i(x_i(t)) u_i(t), \quad i = 1, ..., N, \quad t \geq 0 \quad (1)$$

where $x_i(t) \in \mathbb{R}^n$ is the measurable state vector and $u_i(t) \in \mathbb{R}^m$ are the control inputs or players as we shall see later. Note also that we assume that the closed-loop system $f_i(x_i) + g_i(x_i) u_i, \quad i = 1, ..., N$ is locally Lipschitz.

The leader agent who generates the target state $x_0$ has the following uncontrolled dynamics:

$$\dot{x}_0(t) = f_0(x_0(t)) \quad (2)$$

Our next step shall be to define the local tracking error for each agent as

$$\delta_i = \sum_{j \in N_i} e_{ij}(x_i - x_j) + e_{i0}(x_i - x_0), \quad i = 1, ..., N \quad (3)$$

where $e_{i0} \geq 0$ is the pinning gain that is non-zero for a limited group of agents (i.e. $e_{i0} = 1$ for at least one that node that communicates directly with the leader agent and $e_{i0} = 0$ otherwise).

The time derivative of (3) is given by

$$\dot{\delta}_i = \sum_{j \in N_i} e_{ij}(f_i(x_i) - f_j(x_j)) + e_{i0}(f_i(x_i) - f_0(x_0)) + (d_i + e_{i0}) g_i(x_i) u_i - \sum_{j \in N_i} e_{ij} g_j(x_j) u_j, \quad i = 1, ..., N \quad (4)$$

Now, using (1) and (2), the dynamics (4) can be re-written as

$$\dot{\delta}_i = \sum_{j \in N_i} e_{ij}(\dot{x}_i - \dot{x}_j) + e_{i0}(\dot{x}_i - \dot{x}_0), \quad i = 1, ..., N \quad (5)$$

We shall now define the distributed cost functional that every agent would like to optimize as

$$J_i(\delta_i(0), u_i, u_{N_i}) = \frac{1}{2} \int_0^\infty \left( Q_i(\delta_i) + u_i^T R_i u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j \right) dt, \forall \delta_i(0), u_i, u_{N_i}, \quad i = 1, ..., N \quad (6)$$

where $u_{N_i} = \{u_j | j \in N_i\}$, and the weighting matrices $Q_i(\delta_i) > 0, R_i > 0$ are symmetric and constant.

We are thus interested in finding Nash equilibrium policies $u^*_i$ for the $N$ player game in the sense that $J_i^* = J_i(\delta_i(0), u^*_i, u_{N_i})$.$ J_i(\delta_i(0), u_i, u_{N_i}) \leq J_i(\delta_i(0), u^*_i, u_{N_i}), \forall u_i, \quad i = 1, ..., N. \quad (7)$

The corresponding value function for every agent $i$ is given by

$$V_i(\delta_i(t)) = \frac{1}{2} \int_0^\infty \left( Q_i(\delta_i) + u_i^T R_i u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j \right) dt, \forall \delta_i, \quad i = 1, ..., N \quad (8)$$

Hence, the ultimate goal of the multi-agent graphical game is to find the following tuple of optimal values (i.e. solve the $N$ coupled optimization problems):

$$V_i^*(\delta_i(t)) = \min_{u_i} \int_0^\infty \left( Q_i(\delta_i) + u_i^T R_i u_i + \sum_{j \in N_i} u_j^T R_{ij} u_j \right) dt, \forall \delta_i, \quad i = 1, ..., N \quad (9)$$

given each agent’s local tracking error dynamics (5).

Online learning algorithm for non-linear differential graphical games

The differential equivalents to (7) are given by the following non-linear Lyapunov equations:

$$\nabla V_i^T \left[ \sum_{j \in N_i} e_{ij}(f_i(x_i) - f_j(x_j)) + e_{i0}(f_i(x_i) - f_0(x_0)) + (d_i + e_{i0}) g_i(x_i) u_i - \sum_{j \in N_i} e_{ij} g_j(x_j) u_j \right] + \frac{1}{2} Q_i(\delta_i) + \frac{1}{2} u_i^T R_i u_i + \frac{1}{2} \sum_{j \in N_i} u_j^T R_{ij} u_j = 0, \quad V_i(0) = 0, \quad i = 1, ..., N \quad (10)$$

where $\nabla V_i = \partial V_i / \partial \delta_i \in \mathbb{R}^n$ is the partial derivative of the value function $V_i(\delta_i)$ with respect to $\delta_i$.

The value functions (7) have the following associated Hamiltonians:
Given admissible feedback control policies then the non-linear Lyapunov equations (9) have locally smooth solutions to (9) and (12). See Vamvoudakis et al. (2012) for a similar framework of solutions.

The following theorem adopted from Vamvoudakis et al. (2012) provides an existence theorem for the Nash equilibrium.

**Theorem 1.** Let Assumptions 1 and 2 hold. Let \( V_i > 0 \), \( i = 1, \ldots, N \) be the smooth solutions to the coupled HJ equations (12) and the optimal control policies be given by (11) in terms of these solutions. Then the closed-loop systems (5), (11) have asymptotically stable equilibrium points and the policies (11) form a Nash equilibrium.

**Proof.** See Vamvoudakis et al. (2012).

### Value function approximation by critic neural networks

Based on the Weierstrass higher-order approximation theorem (Abu-Khalaf and Lewis, 2005; Adams and Fournier, 2003; Finlayson, 1990; Hornik et al., 1990), there exist a complete independent basis set \( \sigma_i(\delta_i) : \Omega \rightarrow \mathbb{R}^{K_i}, i = 1, \ldots, N \) such that \( \sigma_i(0) = 0, \nabla \sigma_i(0) = 0 \) and constant NN weights \( W_i \in \mathbb{R}^{K_i}, i = 1, \ldots, N \) such that the solutions \( V_i \) to (9) and \( \nabla V_i \) are uniformly approximated on a compact set \( \Omega \) as follows:

\[
V_i = W_i^T \sigma_i(\delta_i) + \omega_i(\delta_i), i = 1, \ldots, N
\]

\[
\nabla V_i = \nabla \sigma_i^T W_i + \nabla \omega_i, i = 1, \ldots, N
\]

where \( \sigma_i(\delta_i) \) are the NN activation function basis set vectors, \( K_i, i = 1, \ldots, N \) is the number of neurons in the hidden layer of every agent, and \( \omega_i(\delta_i) \) are the NNs approximation errors.

As the number of hidden-layer neurons of every agent \( K_i \rightarrow \infty \), the approximation errors \( \omega_i \rightarrow 0 \) and \( \frac{\|\omega_i\|}{\|\sigma_i^T W_i\|} \rightarrow 0 \) uniformly (Abu-Khalaf and Lewis, 2005; Finlayson, 1990). Note that, based on Assumption 1, we have \( \|\omega_i\| \leq b_{\omega_i}, \|\nabla \omega_i\| \leq b_{\nabla \omega_i}, \|\sigma_i\| \leq b_{\sigma_i}, \) and \( \|\nabla \sigma_i\| \leq b_{\nabla \sigma_i}, \forall i \).

By using the critic NN (13)–(14), and fixed policies \( u_i \) and \( u_{Ni} \), the Hamiltonians (10) can be approximated as follows:

\[
H_i(\delta_i, W_i, u_i, u_{Ni}) = W_i^T \nabla \sigma_i
\]

\[
\left[ \sum_{j \in \mathcal{N}} e_j(f_j(x_i) - f_j(x_i)) + e_0(f_j(x_i) - f_0(x_0)) + (d_j + e_0)g_j(x_i)u_i - \sum_{j \in \mathcal{N}} e_j g_j(x_i)u_i \right] + \frac{1}{2} Q(\delta_i) + \frac{1}{2} \sigma_i^T R_i \sigma_i + \frac{1}{2} \sum_{j \in \mathcal{N}} u_j^T R_j u_j = e_{Bi}
\]

where the residual errors are given as

\[
e_{Bi} = - (\nabla \omega_i)^T \left[ \sum_{j \in \mathcal{N}} e_j(f_j(x_i) - f_j(x_i)) + e_0(f_j(x_i) - f_0(x_0)) + (d_j + e_0)g_j(x_i)u_i - \sum_{j \in \mathcal{N}} e_j g_j(x_i)u_i \right].
\]
Note that according to Assumption 1, these residual errors are bounded on the compact set $\Omega$, i.e. $\sup_{i \in \Omega} \|e_i\| \leq \tilde{e}_i$, $i = 1, \ldots, N$.

The following assumption for the drift and control dynamics is needed for every agent.

**Assumption 3.** For a given compact set $\Omega \subset \mathbb{R}^d$ and $i = 1, \ldots, N$:

(a) $f(x_i) \leq b_f |x_i|$

(b) $g(x_i)$ is bounded by constant: $\|g(x_i)\| \leq b_g$.

(c) The critic NNs weights are bounded by known constants $\|W_i\| < W_{\text{max}}$. 

The ideal weights of the critic NNs, i.e. $W_i$, $i = 1, \ldots, N$, which provide the solution to (15) are unknown and must be approximated in real time. Therefore, the output of the critic NNs $\hat{V}_i$ and the approximate Bellman equations (18) can be written as

$$
\dot{V}_i = \hat{W}_i^T \sigma(\delta_i)
$$

where $\hat{W}_i \in \mathbb{R}^K$ is the current estimated value of the ideal weight $W_i \in \mathbb{R}^K$ for every agent.

It is desired to select $\dot{\hat{W}}_i$ to minimize the following squared residual error:

$$
E_i = \frac{1}{2} e_i^T e_i
$$

Hence, we shall select the tuning law for the critic weights as the following normalized gradient descent algorithm:

$$
\dot{\hat{W}}_i = -\alpha_i \frac{\partial E_i}{\partial \hat{W}_i} = -\alpha_i \frac{B_i}{(1 + B_i^T B_i)} e_i - \alpha_i \frac{B_i}{m_i} e_i
$$

where $B_i = \nabla \sigma(\sum_{j \in N_i} e_0(f_j(x_j) - f_j(x_i)) + e_0(f_i(x_i) - f_0(x_0)) + (d_i + e_0)g(x_i)u_i - \sum_{j \in N_i} e_0g_j(x_j)u_j)$, $m_i = 1 + B_i^T B_i$, $B_i = \frac{\partial \hat{V}_i}{\partial \hat{W}_i}$, $\alpha_i > 0$ is the learning rate and $(1 + B_i^T B_i)^2$ is used for normalization.

The following definition, is needed before we proceed to Lemma 1.

**Definition 1** (Persistence of Excitation (PE)). The bounded vector signal $B_i(t)$, $i = 1, \ldots, N$ is PE (Zhang et al., 2012a) over the interval $[t, t + T_i]$ if there exists $T_i > 0$, $\gamma_i > 0$ and $\gamma_i + N_i > 0$ such that for all $\epsilon$:

$$
\gamma_i I \leq \int_t^{t + T_i} B_i(\tau) B_i^T(\tau)d\tau \leq \gamma_i + N_i I, \quad i = 1, \ldots, N
$$

**Lemma 1.** Let $(u_i, u_N)$, $\forall i$ be a given admissible feedback policy set, let (20) be the tuning of the critic NNs and assume that $\tilde{B}_i$ is PE. Then the critic parameter errors converge exponentially to the residual set $\|W_i\| \leq \eta_i e^{-\eta_i \tau} + \frac{\eta_i}{m_i \eta_i} \tilde{e}_i$, $i = 1, \ldots, N$ for some $\eta_i, \tilde{e}_i > 0$.

**Proof.** From the coupled HJ equations we have

$$
-W_i^T \nabla \sigma(\sum_{j \in N_i} e_0(f_j(x_j) - f_j(x_i)) + e_0(f_i(x_i) - f_0(x_0)) + (d_i + e_0)g(x_i)u_i - \sum_{j \in N_i} e_0g_j(x_j)u_j)
$$

$$
+ e_i = \frac{1}{2}Q(\delta_i) + \frac{1}{2} u_i^T R_0 u_i + \frac{1}{2} \sum_{j \in N_i} u_j^T R_0 u_j.
$$

Substituting (21) into (18) and doing some simple algebra, we can write

$$
e_i = -\hat{W}_i^T B_i + e_i.
$$

Assuming that (23) is a linear time-varying system with an input given by $e_i$, $i = 1, \ldots, N$, then the closed-form solution $\hat{W}_i$ is given as

$$
\hat{W}_i(t) = \phi_i(t, t_0) \hat{W}_i(0) + \alpha_i \int_{t_0}^{t} \phi_i(\tau, t_0) \tilde{B}_i m_i e_i d\tau
$$

where the state transition matrix can be found from

$$
\frac{\partial \phi_i(t, t_0)}{\partial t} = -\alpha_i B_i B_i^T \phi_i(t, t_0), \quad \phi_i(t_0, t_0) = I, \quad i = 1, \ldots, N
$$

The state transition matrix $\phi_i$, $i = 1, \ldots, N$ is exponentially stable provided that $B_i$ is PE (Ioannou and Sun, 1996). As $B_i$ is PE and the fact that $\|B_i\| \leq 1$ and $\sup_{x \in \Omega} \|e_i\| \leq \tilde{e}_i$, $i = 1, \ldots, N$, we finally obtain

$$
\|W_i\| \leq \eta_i e^{-\eta_i \tau} + \frac{\eta_i}{m_i \eta_i} \tilde{e}_i, \quad i = 1, \ldots, N
$$

for some $\eta_i, \tilde{e}_i > 0$.

This completes the proof.

**Control policy approximation by actor neural networks**

We shall now define the optimal control policy by using the value function approximation (13)–(14), as follows:
\[ u_i = -(d_i + e_{0})R^{-1}_{ii}g_i^T(x_i)(\nabla \sigma_i^T W_i + \nabla a_i), \quad i = 1, ..., N \]  
(27)

However, as (27) is unknown, we can approximate the control policy with an actor NN as

\[ u_i = W^T_{ui} \sigma_u(\delta_i) + a_u(\delta_i), \quad i = 1, ..., N \]  
(28)

and

\[ \dot{\delta}_i = W^T_{ui} \sigma_u(\delta_i), \quad i = 1, ..., N \]  
(29)

where \( \dot{W}_{ui} \in \mathbb{R}^{L \times m} \) is the current estimated value of the ideal NN weight \( W_{ui} \in \mathbb{R}^{L \times m} \), \( \sigma_u(\delta_i) \) are the activation functions and \( L \) is the number of neurons in the hidden layer.

Define the critic and the actor NNs estimation errors respectively as

\[ \dot{W}_i = W_i - \dot{W}_i \]  
(30)

and

\[ \dot{W}_{ui} = W_{ui} - \dot{W}_{ui} \]  
(31)

In order to tune the actor NNs, we shall define the following error signal \( e_u \in \mathbb{R}^n \):

\[ e_u = \dot{W}^T_{ui} \sigma_u(\delta_i) + (d_i + e_{0})R^{-1}_{ii}g_i^T(x_i)\nabla \sigma_i^T \dot{W}_i \]  
(32)

Now, the objective function to be minimized is given as

\[ E_{eu} = \frac{1}{2} e_u^T e_u \]  
(33)

Hence, the weight update law for the actor NN can be found by using a normalized gradient descent algorithm, as follows

\[ \dot{W}_{ui} = -\alpha_u \frac{\sigma_u(\delta_i)^T \sigma_u(\delta_i) \dot{W}_{ui} + (d_i + e_{0})R^{-1}_{ii}g_i^T(x_i)\nabla \sigma_i^T \dot{W}_i}{m_u} \]  
(34)

where \( m_u = 1 + \sigma_u(\delta_i)^T \sigma_u(\delta_i) \), \( \dot{\sigma}_u = \frac{\sigma_u(\delta_i)}{1 + \sigma_u(\delta_i)^T \sigma_u(\delta_i)} \)

Theorem 2. Consider the dynamical system (5) and the multiplayer graphical game formulation. Let the critic NN of each agent be given by (17) and the corresponding control input be given by (29). Consider that the tuning for agent i critic NN is given by (20) and the corresponding actor NN is tuned by (34). Let Assumptions 1–3 hold. Then the closed-loop system states \( \delta_i(t) \), the critic NN errors \( \dot{W}_i \), the actor NN errors \( \dot{W}_{ui} \) are UUB (Uniformly Ultimately Bounded), for a sufficiently large number of NN neurons.

Proof. See Appendix.

Simulation results

Consider a five-node MAS, as shown in Figure 1 with dynamics given by \( \dot{x}_i = f(x_i) + g(x_i)u_i, \quad x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}, \) \( i = 1, 2, ..., 5, \) where

\[ f(x_i) = \begin{pmatrix} x_{i2} \\ -x_{i1} + \epsilon(1-x_{i1}^2)x_{i2} \end{pmatrix}, \quad g_1(x_i) = \begin{bmatrix} 0 \\ -0.8x_{i1}x_{i2} \end{bmatrix}, \]

\[ g_2(x_2) = \begin{bmatrix} 0 \\ x_{21}x_{22} \end{bmatrix}, \quad g_3(x_3) = \begin{bmatrix} 0 \\ 0.5x_{31}x_{32} \end{bmatrix}, \]

\[ g_4(x_4) = \begin{bmatrix} 0 \\ -0.2x_{41}x_{42} \end{bmatrix}, \quad g_5(x_5) = \begin{bmatrix} 0 \\ 1.4x_{51}x_{52} \end{bmatrix}, \]

with the pinning gains and the edge weights chosen to be \( \epsilon = 0.5 \) and the leader node dynamics is \( f(x_0) = -x_{01} + \epsilon(1-x_{01}^2)x_{02} \).

For \( i, j = 1, 2, ..., 5, \) we pick \( Q(\delta_i) = \delta_i^T Q \delta_i = \delta_i^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \)

and the tuning gains are picked as \( \alpha_u = 1 \) and \( \alpha_i = 5, \) \( i = 1, ..., 5. \)

The available information vector for each agent, \( \delta_i = \begin{bmatrix} \delta_{i1} \\ \delta_{i2} \end{bmatrix}, i = 1, ..., 5, \) is restricted by the graph topology.

The critic NN activation functions are chosen as follows

\[ \sigma_1 = [\delta_{11}^2, \delta_{11}1_{12}, \delta_{11}1_{12}, \delta_{11}2_{12}, \delta_{11}2_{12}, \delta_{11}1_{12}, \delta_{111}1_{12}, \delta_{111}2_{12}, \delta_{112}1_{12}, \delta_{112}2_{12}] \]

\[ \sigma_2 = [\delta_{21}^2, \delta_{211}1_{22}, \delta_{211}1_{22}, \delta_{212}1_{22}, \delta_{212}1_{22}, \delta_{211}2_{22}, \delta_{212}2_{22}, \delta_{213}1_{22}, \delta_{213}1_{22}, \delta_{214}1_{22}, \delta_{214}2_{22}] \]

\[ \sigma_3 = [\delta_{31}^2, \delta_{311}1_{32}, \delta_{311}1_{32}, \delta_{312}1_{32}, \delta_{312}1_{32}, \delta_{313}1_{32}, \delta_{313}1_{32}, \delta_{314}1_{32}, \delta_{314}2_{32}, \delta_{315}1_{32}, \delta_{315}1_{32}, \delta_{316}1_{32}, \delta_{316}2_{32}] \]

\[ \sigma_4 = [\delta_{41}^2, \delta_{411}1_{42}, \delta_{411}1_{42}, \delta_{412}1_{42}, \delta_{412}1_{42}, \delta_{413}1_{42}, \delta_{413}1_{42}, \delta_{414}1_{42}, \delta_{414}2_{42}, \delta_{415}1_{42}, \delta_{415}1_{42}, \delta_{416}1_{42}, \delta_{416}2_{42}, \delta_{417}1_{42}, \delta_{417}1_{42}, \delta_{418}1_{42}, \delta_{418}2_{42}, \delta_{419}1_{42}, \delta_{419}1_{42}, \delta_{4110}1_{42}, \delta_{4110}2_{42}] \]

\[ \sigma_5 = [\delta_{51}^2, \delta_{511}1_{52}, \delta_{511}1_{52}, \delta_{512}1_{52}, \delta_{512}1_{52}, \delta_{513}1_{52}, \delta_{513}1_{52}, \delta_{514}1_{52}, \delta_{514}2_{52}, \delta_{515}1_{52}, \delta_{515}1_{52}, \delta_{516}1_{52}, \delta_{516}2_{52}, \delta_{517}1_{52}, \delta_{517}1_{52}, \delta_{518}1_{52}, \delta_{518}2_{52}, \delta_{519}1_{52}, \delta_{519}1_{52}, \delta_{5110}1_{52}, \delta_{5110}2_{52}] \]

and the actor NN activation functions are \( \dot{\sigma}_u = \nabla \sigma_u, \quad i = 1, ..., N. \)

It should be noted that a small exponential decreasing probing noise is added to the control inputs to ensure excitation. Figures 2 and 3 show the convergence of all the agents’ critic weights after using the proposed learning algorithm.

Stability and convergence analysis

The critic and the actor NNs’ tuning laws are designed to ensure global synchronization, closed-loop system stability and convergence of the policies to a Nash equilibrium.
Figures 4 and 5 show the evolution of the local tracking errors, which approximately converge to zero.

**Conclusion and future work**

This paper integrates the abilities of optimal adaptive control, differential games and non-linear MASs to introduce a formulation of leader–follower non-linear differential graphical games. An online distributed optimal adaptive control algorithm based on RL techniques is presented to solve the continuous-time multi-agent non-linear differential graphical games. Each agent uses a critic and an actor NN to learn online respectively the optimal value and optimal control policy. The closed-loop signals are proved to be bounded according to Lyapunov stability and the policies form a Nash equilibrium.

For future work, we intend to extend the approach of this paper to distributed control of non-linear differential graphical games under unknown dynamics.

**Acknowledgement**

The authors would like to thank Prof. Frank Lewis for his helpful suggestions and guidance.

**Conflict of interest**

The authors declare that there is no conflict of interest.
**Funding**

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

**References**


Vrabie D and Lewis FL (2010) Integral reinforcement learning for online computation of feedback Nash strategies of nonzero-sum differential games. In: 49th IEEE Conference on Decision and Control, Atlanta, GA.


**Appendix**

**Proof of Theorem 1**

Consider the following Lyapunov function

\[
L(t) = \sum_{i=1}^{N} \left\{ V_i(t) + \gamma_i \left( \frac{1}{2} \bar{W}^T_i \bar{\sigma}_i + \bar{\omega}_i \right) + \lambda_i \left( \frac{1}{2} \bar{W}^T_i \bar{\sigma}_i + \bar{\omega}_i \right) \right\}
\]  

(A.1)

where \( V_i(t) = W^T_i \sigma_i + \omega_i, i = 1, \ldots, N \) are the approximate solutions to (12).

The time derivative of the Lyapunov function (A.1) is given by

\[
\dot{L}(t) = \sum_{i=1}^{N} \left\{ \dot{V}_i(t) - \gamma_i \left( \frac{1}{2} \bar{W}^T_i \bar{\sigma}_i + \bar{\omega}_i \right) - \lambda_i \left( \frac{1}{2} \bar{W}^T_i \bar{\sigma}_i + \bar{\omega}_i \right) \right\}
\]

(A.2)

Using (7), the first term of (A.2) is written as

\[
\sum_{i=1}^{N} \dot{V}_i(t) = \sum_{i=1}^{N} \left( -\frac{1}{2} \dot{g}^T_i \dot{\sigma}_i - \frac{1}{2} \dot{\omega}_i - \frac{1}{2} \dot{\omega}_i \right)
\]

(A.3)

We shall note that after, using (29)–(31), \( \dot{V}_i \) can be written as

\[
\sum_{i=1}^{N} \dot{V}_i(t) = \sum_{i=1}^{N} \left( -\frac{1}{2} \dot{g}^T_i \dot{\sigma}_i - \frac{1}{2} \dot{\omega}_i - \frac{1}{2} \dot{\omega}_i \right)
\]

(A.4)

where

\[
\dot{\sigma}_i = -\alpha_i B_i B_i^T \bar{W}_i + \alpha_i \frac{B_i}{m_i} \bar{e}_i
\]

(A.5)

In order to simplify the second term of (A.2), we know that

\[
\dot{\omega}_i = -\dot{\omega}_i - \alpha_i \bar{B}_i \bar{B}_i^T \bar{W}_i + \alpha_i \frac{\bar{B}_i}{m_i} \bar{e}_i
\]

(A.6)

Before we proceed to the third term of (A.2), we shall use (27) and (28) to write

\[
\sum_{i=1}^{N} \left\{ -\gamma_i \bar{W}_i^T \bar{B}_i \bar{B}_i^T \bar{W}_i + \gamma_i \bar{W}_i^T \frac{\bar{B}_i}{m_i} \bar{e}_i \right\}
\]
\[ W_i^T \sigma_u(\delta_i) + \omega_u(\delta_i) + (d_i + e_0)R_i^{-1}g_i(x_i)(\nabla \sigma_i^TW_i + \nabla \omega_i) = 0 \]  
\[ (A.7) \]

Thus, we can write the actor error dynamics using (34) and (A.7) as

\[ \dot{W}_i = -\dot{W}_i = \alpha_u \frac{\partial \sigma_u(\delta_i)}{m_{\sigma_u}} \]
\[ [-\sigma_u(\delta_i)^TW_i + (d_i + e_0)R_i^{-1}g_i(x_i) \frac{\partial \sigma_i^T}{\partial \delta_i} \dot{W}_i + \omega_u, w_i] \]
\[ = \alpha_u [-\sigma_u \alpha_i^T \dot{W}_i - (d_i + e_0) \sigma_u R_i^{-1}g_i(x_i) \frac{\partial \sigma_i^T}{\partial \delta_i} \dot{W}_i + \alpha_u \omega_u, w_i] \]  
\[ (A.8) \]

where \( \omega_u, w_i = -\omega_u(\delta_i) - (d_i + e_0)R_i^{-1}g_i(x_i) \nabla \omega_i. \)

The summation of the third term of (A.2)

\[ \sum_{i=1}^N \dot{L}_u = -\sum_{i=1}^N \lambda_i \langle \dot{W}_i^T, \alpha_i^T \dot{W}_i \rangle \]  
\[ (A.9) \]

Now for the total Lyapunov function, we shall use (A.4), (A.6) and (A.9) to write (A.2) as

\[ \dot{L}(t) = \sum_{i=1}^N \{ -\dot{Z}_i^TM_i \dot{Z}_i + D_i \dot{Z}_i \} \]  
\[ (A.11) \]

where \( M_i = \begin{bmatrix} m_{i1} & m_{i2} & m_{i3} & m_{i4} \\ m_{i1} & m_{i2} & m_{i3} & m_{i4} \\ m_{i1} & m_{i2} & m_{i3} & m_{i4} \\ m_{i1} & m_{i2} & m_{i3} & m_{i4} \end{bmatrix}, D_i = [d_1, d_2, d_3, d_4], \)

\( m_{i1} = \frac{1}{2} q_i, m_{i2} = + \gamma_B \dot{B}_i^T, m_{i3} = m_{i4} = m_{i1}^T = 0, \)
\( m_{i4} = m_{i2}^T = 0, m_{i3} = m_{i2}^T = 0, m_{i23} = \frac{\lambda_i}{\alpha_u(X_i)}, \)
\( (d_i + e_0) \frac{\partial \sigma_i}{\partial \delta_i} g_i(X_i) R_i^{-T} = m_{i22}m_{i33} = \frac{1}{2} R_i + \frac{\lambda_i}{\alpha_u(X_i)}, m_{i44} = \frac{1}{2} \sum_{j \in N_i} R_j, \)
\( d_1 = 0, d_2 = \gamma_i \frac{\dot{g}_i}{\alpha_u}, d_3 = R_i \dot{W}_i^T \sigma_i + \frac{\lambda_i}{\alpha_u(X_i)}, d_4 = \sum_{j \in N_i} R_j \dot{W}_i^T \alpha_{u_j}, D_i \leq D_{i_{\text{max}}} \)

Let the parameters be chosen such that \( M_i > 0 \) with

\[ M_i = \begin{bmatrix} M_{i1} & 0 & 0 & 0 \\ 0 & M_{i2} & 0 & 0 \\ 0 & 0 & M_{i3} & 0 \\ 0 & 0 & 0 & M_{i4} \end{bmatrix}, \]
\[ (A.12) \]
\[ M_{i1} = m_{i1}, \]
\[ M_{i2} = m_{i22}, M_{i3} = m_{i33}, M_{i4} = m_{i44} \]

where the following properties must hold:

(a) \( M_{i1} = \frac{1}{2} q_i > 0. \)
(b) \( M_{i2} > 0, \) which requires that \( m_{i33} > 0 \) and \( m_{i22} \) which hold after proper selection of \( \gamma_i \) and \( \lambda_i. \)
(c) \( M_{i3} = \frac{1}{2} \sum_{j \in N_i} R_j \) > 0.

Finally, (A.11) becomes

\[ L < \sum_{i=1}^N \{ -\| \dot{Z}_i \| \sigma_{\text{min}}(M_i) + D_{i_{\text{max}}} \| \dot{Z}_i \| \} \]  
\[ (A.13) \]

and it is negative as long as

\[ \| \dot{Z}_i \| > \frac{D_{i_{\text{max}}} \sigma_{\text{min}}(M_i)}{\| \dot{Z}_i \|} \]
\[ \text{max} = \text{max} \]
\[ (A.14) \]

It is shown that if (A.14) exceeds a certain bound, then \( \dot{L} \) is negative. Therefore, according to the standard Lyapunov extension theorem the analysis above demonstrates that the states and the weights are UUB (Khalil, 1996). Condition (A.14) holds if the norm of any component of \( \dot{Z}_i \) exceeds the bound, i.e. specifically \( \dot{\delta}_i > B_{i2}, \dot{W}_i > B_{i2}, \alpha_i^T \dot{W}_i > B_{i2}, \alpha_i^T \dot{W}_i > B_{i2} \). Note that according to Assumption 1, the actor NN activation functions \( \sigma_u \) and \( \sigma_i \) are also bounded.