*Journal of AI and Data Mining*
*Vol 3, No 1, 2015, 101-111.*
doi:10.5829/idosi.JAIDM.2015.03.01.11

# Optimal adaptive leader-follower consensus of linear multi-agent systems: Known and unknown dynamics

F. Tatari and M. B. Naghibi-Sistani*

*Electrical Engineering Department, Ferdowsi university of Mashhad, Azadi square, Mashhad, Iran.*

Received 5 May 2014; Accepted 13 May 2015
*Corresponding author: mb-naghibi@um.ac.ir (M. B. Naghibi).*

**Abstract**
In this paper, the optimal adaptive leader-follower consensus of linear continuous time multi-agent systems is considered. The error dynamics of each player depends on its neighbors' information. Detailed analysis of online optimal leader-follower consensus under known and unknown dynamics is presented. The introduced reinforcement learning-based algorithms learn online the approximate solution to algebraic Riccati equations. An optimal adaptive control technique is employed to iteratively solve the algebraic Riccati equation based on the online measured error state and input information for each agent without requiring the priori knowledge of the system matrices. The decoupling of the multi-agent system global error dynamics facilitates the employment of policy iteration and optimal adaptive control techniques to solve the leader-follower consensus problem under known and unknown dynamics. Simulation results verify the effectiveness of the proposed methods.

**Keywords:** *Graph Theory, Leader-follower Consensus, Multi-agent Systems, Policy Iterations.*

## 1. Introduction

In recent decades multi-agent systems (MASs) are applied as new methods for solving problems which cannot be solved by a single agent. MASs contain agents forming a network which exchange information through the network to satisfy a predefined objective. Information exchanging among agents can be divided to centralized and distributed approaches. Centralized approaches are mainly concentrated and discussed where all agents have to continuously communicate with a central agent. This kind of communication results in a heavy traffic, information loss and delay. Also, the central agent must be equipped with huge computational capabilities to receive all the agents' information and provide them with a command in response. Recently these challenges deviates the stream of studies toward distributed techniques where agents only need to communicate with their local neighbors.

A main problem in cooperative control of MASs is Consensus or synchronization. In consensus problems, it is desired to design simple control law for each agent, using local information, such that the system can achieve prescribed collective behaviors. In the field of control, consensus of MAS is categorized to cooperative regulation and cooperative tracking. In cooperative regulator problems, known as leaderless consensus, distributed controllers are designed for each agent, such that all agents are eventually driven to an unprescribed common value [1]. This value may be a constant, or may be time varying, but is generally a function of the initial states of the agents in the communication network [2]. Alternatively in a cooperative tracking problem, which is considered in this paper, there exists a leader agent. The leader agent acts as a command generator, which generates the desired reference trajectory. The leader ignores information from the follower agents and all other agents are required to follow the leader agent [3,4]. This problem is known as the leader-follower consensus [5], model reference consensus [6], or pinning control [7].

In MASs, the network structure and agents communications can be shown by graph theory tools.

Multi player linear differential games rely on solving the coupled algebraic Riccati equations (AREs). The solution of each player coupled equations requires knowledge of the player's neighbors strategies. Since AREs are nonlinear, it is difficult to solve them directly. To solve ARE, the following approaches have been proposed and extended: backwards integration of the Differential Riccati Equation, or Chandrasekhar equations [8]; eigenvector-based algorithms [9,10] and the numerically advantageous Schur-vector-based modification [11]; matrix-sign-based algorithms [12-14]; Newton's method [15-18]. These methods are mostly offline procedures and are proven to converge to the desired solution of the ARE. They either operate on the Hamiltonian matrix associated with the ARE (eigenvector and matrix-sign-based algorithms) or require solving Lyapunov equations (Newton's method). In all methods, the system dynamics must be known and a preceding identification procedure is always necessary.

Adaptive control [19,20] allows the design of online stabilizing controllers for uncertain dynamic systems. A conventional way to design an adaptive optimal control law is to identify the system parameters first and then solve the related algebraic Riccati equation. However, such adaptive systems are known to respond slowly to parameter variations from the plant. Optimal adaptive controllers can be obtained by designing adaptive controllers with the ability of learning online the solutions to optimal control problems.

Reinforcement learning (RL) is a sub-area of machine learning involved with how to methodically modify the actions of an agent (player) based on observed responses from its environment [21]. RL is a class of methods, which provides online solution for optimal control problems by means of a reinforcement scalar signal measured from the environment, which indicates the level of control performance. This is because a number of RL algorithms [22-24] do not require knowledge or identification/learning of the system dynamics, and RL is strongly connected with direct and indirect optimal adaptive control methods.

In this paper, the optimal adaptive control means the algorithms based on RL that provide online synthesis of optimal control policies. Also, the scalar value associated with the online adaptive controller acts as a reinforcement signal to optimally modify the adaptive controller in an online fashion.

RL algorithms can be employed to solve optimal control problems, by means of function approximation structures that can learn the solution of ARE. Since function approximation structures are used to implement these online iterative learning algorithms, the employed methods can also be addressed as approximate dynamic programming (ADP) [24].

Policy Iteration (PI), a computational RL technique [25], provides an effective means of online learning solutions to AREs. PI contains a class of algorithms with two steps, policy evaluation and policy improvement. In control theory, PI algorithm amounts to learning the solution to a nonlinear Lyapunov equation, and then updating the policy through minimizing a Hamiltonian function. Using PI technique, a nonlinear ARE is solved successively by breaking it into a sequence of linear equations that are easier to handle. However, PI has primarily been developed for discrete-time systems [24,25], recent research findings present Policy Iteration techniques for continuous-time systems [26].

ADP and RL methods have been used to solve multi player games for finite-state systems [27,28]. In [29-32], RL methods have been employed to learn online in real-time the solutions of optimal control problems for dynamic systems and differential games.

The leader-follower consensus has been an active area of research. Jadbabaie et al. considered a leader-follower consensus problem and proved that if all the agents were jointly connected with their leader, their states would converge to that of the leader over the course of time [33]. To solve the leader-follower problem, Hong et al. proposed a distributed control law using local information [34] and Cheng et al. provided a rigorous proof for the consensus using an extension of LaSalle's invariance principle [35]. Cooperative leader follower attitude control of multiple rigid bodies was considered in [36]. Leader-follower formation control of nonholonomic mobile robots was studied in [37]. Peng et al. studied the leader-follower consensus for an MAS with a varying-velocity leader and time-varying delays [38]. The consensus problem in networks of dynamic agents with switching topology and time-delays was proposed in [39].

In the progress of the research on leader-follower consensus of MASs, the mentioned methods were mostly offline and non-optimal and required the complete knowledge of the system dynamics.

The optimal adaptive control contains the algorithms that provide online synthesis of optimal control policies [40]. For a single system, [26] introduced an online iterative PI method which does not require the knowledge of internal

system dynamics but does require the knowledge of input dynamics to solve the linear quadratic regulator (LQR) problem. Vrabie et al. showed that after each time the control policy is updated, and the information of state and input must be recollected for the next iteration [26]. Jiang et al. introduced a computational adaptive optimal control method for the LQR problem, which does not require either the internal or the input dynamics [41]. For MASs, [42] introduced an online synchronous PI for optimal leader-follower consensus of linear MASs with the known dynamics. Based on the previous studies, the online optimal leader-follower consensus of MASs under the unknown linear dynamics has remained an open problem.

This paper presents an online optimal adaptive algorithm for continuous time leader-follower consensus of MASs under known and unknown dynamics. The main contribution of the paper is the introduction of a direct optimal adaptive algorithm (data-based approach) which converges to optimal control solution without using an explicit, a priori obtained, model of the matrices (drift and input matrices) of the linear system. We implement the decoupling of multi-agent global error dynamics which facilitates the employment of policy iteration and optimal adaptive control techniques to solve the leader-follower consensus problem under known and unknown dynamics. The introduced method employs PI technique to iteratively solve the ARE of each agent using the online information of error state and input without requiring a primary knowledge of system matrices. For each agent, all iterations are implemented using repeatedly the same error state and input information on some fixed time intervals. In this paper, the employed online optimal adaptive computational tool is motivated with [41], where the method is generalized for leader-follower consensus in MASs.

The paper is organized as follows. Section 2 contains the results from Graph theory, also the problem formulation, node error dynamics and leader-follower error dynamics decoupling are clarified in this section. Section 3 introduces Policy iteration algorithm for leader-follower consensus under known dynamics. Optimal adaptive control design for leader-follower consensus under unknown dynamics is presented in section 4. Simulation results are discussed in Section 5. Finally the conclusions are drawn in section 6.

## 2. Problem formulation and preliminaries

### 2.1. Graphs

Graph theory is a useful mathematical tool in multi-agent systems research where information exchange between agents and the leader is shown through a graph. The topology of a communication network can be expressed by either a directed or undirected graph, according to whether the information flow is unidirectional or bidirectional. The topology of information exchange between N agents is described by a graph $Gr = (V, E)$, where $V = \{1,2,...,N\}$ is the set of vertices representing N agents and $E \subset V \times V$ is the set of edges of the graph. $(i, j) \in E$ means there is an edge from node $i$ to node $j$. We assume the graph is simple, e.g., no repeated edges and no self-loops. The topology of a graph is often represented by an adjacency matrix $A_G = [a_{ij}] \in R^{N \times N}$ with $a_{ij} = 1$ if $(j,i) \in E$ and $a_{ij} = 0$ otherwise. Note $(i,i) \notin E, \forall i$, $a_{ii} = 0$. The set of neighbors of a node $i$ is $N_i = \{j : (j,i) \in E\}$, i.e. the set of nodes with arcs incoming to $i$. If node $j$ is a neighbor of node $i$, the node $i$ can get information from node $j$ not necessarily vice versa for directed graphs. In undirected graphs, neighbor is a mutual relation. Define the in-degree matrix as a diagonal matrix $D = diag(d_i) \in R^{N \times N}$ with $d_i = \sum_{j \in N_i} a_{ij}$ the weighted in-degree of node $i$ (i.e. i$^{th}$ row sum of $A_G$). Define the graph Laplacian matrix as $L = D - A_G$, which has all row sums equal to zero. Apparently in bidirectional (undirected) graphs, $L$ is a symmetric matrix. A path is a sequence of connected edges in a graph. A graph is connected if there is a path between every pair of vertices.

The leader is represented by vertex 0. Information is exchanged between the leader and the agents which are in the neighbors of the leader (See Figure 1.).

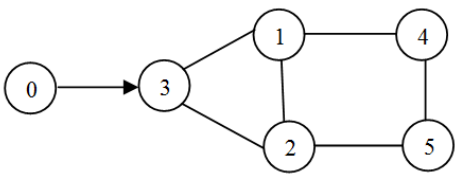


**Figure 1. Communication graph.**

**2.2. Synchronization and node error dynamics**

In cooperative tracking control of networked linear systems, we wish to achieve synchronization in the multi-agent system simultaneously optimizing some performance specifications on the agents. Consider an MAS consisting of $N$ agents and a leader, which are in communication through an undirected graph. The dynamics of each agent is

$$\dot{x}_i = Ax_i + B_i u_i \quad (1)$$

where $x_i \in R^n$ is the measurable state of agent $i$, and $u_i \in R^m$ is the input of player $i$. In this section, we assume that $A$ and $B_i$ are accurately known. The matrix $B_i$ is full column rank. The leader labeled, as $i = 0$ has linear dynamics as

$$\dot{x}_0 = Ax_0 \quad (2)$$

where $x_0 \in R^n$ is the measurable state of the leader. Obviously, the leader's dynamics is independent of others. We take the same internal dynamic matrix ($A$) for all the agents and the leader to be identical because this case has practical background such as group of birds, school of fishes etc. The following assumption is used throughout the paper.

**Assumption 1.** The pair $(A, B_i), i = 1,2,..., N$ is stabilizable.

The dynamics of each agent (node) can describe the motion of a robot, unmanned autonomous vehicle, or missile that satisfies a performance objective.

**Definition 1.** The leader-follower consensus of system (1)-(2) is said to be achieved if, for each agent $i \in \{1,2,..., N\}$, there is a local state feedback $u_i$ of $\{x_j : j \in N_i\}$ such that the closed-loop system satisfies $\lim_{t\to\infty} \|x_i(t) - x_0(t)\| = 0,\ i = 1,..., N$ for any initial condition $x_i(0),\ i = 0,1,..., N$.

The design objective is to employ the following distributed control law for agent $i, i = 1,..., N$

$$u_i = K_i\Big(\sum_{j\in N_i} (x_j - x_i) + g_i\,(x_0 - x_i)\Big) \quad (3)$$

where $K_i \in R^{m\times n},\ i = 1,2,..., N$ is a feedback matrix to be designed and $g_i$ is defined to be 1 when the leader is a neighbor of the agent $i$, and 0 otherwise. Since the proposed feedback controller $u_i$, depends on both the states of its neighbors and the leader agent states, $u_i$ is a distributed controller. In order to analyze the leader-follower consensus problem, we denote the error state between the agent $i$ and the leader as $\varepsilon_i = x_i - x_0$. The dynamics of $\varepsilon_i$, $i = 1,..., N$ is

$$\dot{\varepsilon}_i = A\varepsilon_i + B_i K_i \sum_{j\in N_i} (\varepsilon_j - \varepsilon_i) - B_i g_i \varepsilon_i . \quad (4)$$

Considering $\varepsilon = (\varepsilon_1^T, \varepsilon_2^T,..., \varepsilon_N^T)^T, G = diag(g_1,..., g_N)$ and by using the Lapalcian $L$ of Graph $Gr$, we have

$$\dot{\varepsilon} = [I_N \otimes A - diag(B_i K_i)(H \otimes I_n)]\varepsilon \quad (5)$$

where $H = L + G$ and $\otimes$ is the Kronecker product. $diag(B_i K_i)$ is an $N \times N$ block diagonal matrix. The matrix $H$ corresponding to Graph topology has the following properties, which are proved in [43]:

1. The matrix $H$ has nonnegative eigenvalues.
2. The matrix $H$ is positive definite if and only if the graph $Gr$ is connected.

**Assumption 2.** The graph $Gr$ is connected.

The design objective for each agent $i$ is to find the feedback matrix $K_i$ which minimizes the following performance index for linear system (4),

$$J_i = \int_0^\infty (\varepsilon_i^T Q \varepsilon_i + u_i^T R u_i) dt, \quad i = 1, 2,..., \mathrm{N} \quad (6)$$

where $Q \in R^{n\times n}, R \in R^{m\times m}, Q = Q^T > 0$, $R = R^T > 0$, with $(A, Q^{1/2})$ observable.

Before we proceed to the design of online controllers, we need to decouple the global error dynamics (5), as discussed in the following.

**2.3. Decoupling of Leader-follower error dynamic**

Since $H$ is symmetric, there exists an orthogonal matrix $T \in R^{N\times N}$ such that $THT^T = \Lambda = diag\{\lambda_1, \lambda_2,..., \lambda_N\}$ where $\{\lambda_1, \lambda_2,..., \lambda_N\}$ are the eigenvalues of matrix $H$. Based on Assumption 2, $Gr$ is connected therefore $H$ is a positive definite matrix and $\lambda_i > 0, i = 1,2,..., N$. Now let $\delta = (T \otimes I_n)\varepsilon$ then (5) becomes

$$\dot{\delta} = (I_N \otimes A)\delta - diag(B_i K_i)(\Lambda \otimes I_n)\delta \quad (7)$$

Since the obtained global error dynamics (7) is block diagonal, it can be easily decoupled for each agent $i$, where for each agent we have

$$\dot{\delta}_i = (A - \lambda_i B_i K_i)\delta_i\ ,\ i = 1, 2,..., \mathrm{N} \quad (8)$$

$$J_i = \int_0^\infty (\delta_i^T Q \delta_i + u_i^T R u_i) dt \ , \quad i = 1, 2, ..., \mathrm{N} \tag{9}$$

In order to find the optimal $K_i$ which guarantees the leader-follower consensus for every agent $i$, we can minimize (9) with respect to (8), which is easier in comparison with minimizing (6) with respect to (4).

Based on linear optimal control theory, minimizing (9) with respect to (8) to find the feedback matrix $K_i$ can be done by solving the following algebraic Riccati equation for each agent:

$$A^T P_i + P_i A - P_i (\lambda_i B_i) R^{-1} (\lambda_i B_i)^T P_i + Q = 0 \tag{10}$$

Based on the mentioned assumptions, (10) has a unique symmetrical positive definite solution $P_i^*$. Therefore, the optimal feedback gain matrix can be determined by $K_i^* = R^{-1} \lambda_i B_i^T P_i^*$, due to the dependence of $K_i$ to $\lambda_i$, each feedback gain depends on the graph topology. Since ARE is nonlinear in $P_i$, it is usually difficult to directly solve $P_i^*$ from (10), especially for large size matrices. Furthermore, solving (10) and obtaining $K_i^*$ requires the knowledge of $A$ and $B_i$ matrices.

## 3. Policy iteration algorithm for leader-follower consensus of continuous time linear systems under known dynamic

One of the efficient algorithms to numerically approximate the solution of ARE is the Kleinman algorithm [17]. Here we employ the Kleinman algorithm to numerically solve the corresponding ARE for each agent. The Kleinman method performs as a PI algorithm as discussed in the following.

**Algorithm 1. (Policy iteration Kleinman Algorithm)**

*Step 0:* Let $K_i^0 \in R^{m \times n}$ be any initial stabilizing feedback gain.

*Step 1:* Let $P_i^k$ be the symmetric positive definite solution of Lyapunov equation (11) for the agent $i$, $i = 1, 2, ..., \mathrm{N}$

$$(A - \lambda_i B_i K_i^k)^T P_i^k + P_i^k (A - \lambda_i B_i K_i^k) + Q + K_i^{kT} R K_i^k = 0 \tag{11}$$

*Step 2:* $K_i^{k+1}$ with $k = 1, 2, ...$ is defined recursively by

$$K_i^{k+1} = R^{-1} \lambda_i B_i^T P_i^k \tag{12}$$

*Step 3:* $k = k + 1$ and go to step 1.

On convergence. End.

$A - \lambda_i B_i K_i^k$ is Hurwitz and by iteratively solving the Lyapunov equation (11) which is linear in $P_i^k$ and updating $K_i^{k+1}$ by (12) the solution to the nonlinear equation (10) is approximated as $P_i^* \le P_i^{k+1} \le P_i^k$ and $\lim_{k \to \infty} K_i^k = K_i^*$.

**Theorem 1.** Consider the MAS (1)-(2). Suppose Assumptions 1 and 2 are satisfied. Let $P_i > 0$ and $K_i$ be the final solutions of the Kleinman's algorithm for agent $i$, $i = 1, 2, ..., N$. Then under control law (3) all the agents follow the leader from any initial conditions.

**Proof:** Consider the Lyapunov function candidate $V_i = \delta_i^T P_i \delta_i$. The time derivative of this Lyapunov candidate along the trajectory of system (8) is

$$\begin{aligned} \dot{V}_i &= \delta_i^T [(A^T - \lambda_i K_i^T B_i^T) P_i] \delta_i + \\ &\delta_i^T [P_i (A - \lambda_i B_i K_i)] \delta_i = \\ &\delta_i^T [A^T P_i + P_i A - 2 \lambda_i^2 P_i B_i R^{-1} B_i^T P_i] \delta_i \\ &= -\delta_i^T [Q + \lambda_i^2 P_i B_i R^{-1} B_i^T P_i] \delta_i \le 0 \end{aligned} \tag{13}$$

Thus for any $\delta_i \neq 0$, $\dot{V}_i < 0$, $i = 1, 2, ..., \mathrm{N}$. Therefore, system (8) is globally asymptotically stable which implies that all the agents follow the leader.

## 4. Optimal adaptive control for leader-follower consensus under unknown dynamics

To solve (11) without the knowledge of $A$, we have [40]

$$\int_t^{t+\Delta t} (\delta_i^T Q \delta_i + u_i^{kT} R u_i^k) d\tau = \delta_i^T(t) P_i^k \delta_i(t) - \delta_i^T(t + \Delta t) P_i^k \delta_i(t + \Delta t), \tag{14}$$

By online measurement of both $\delta_i$ and $u_i^k$, $P_i^k$ is uniquely determined under some persistence excitation (PE) condition though matrix $B_i$ is still needed to calculate $K_i^{k+1}$ in (12).

To freely solve (11) and (12) without the knowledge of $A$ and $B_i$, here the result of [41] is generalized for MAS leader-follower consensus. An online learning algorithm for the leader-follower consensus problem is developed but does not rely on either $A$ or $B_i$.

For each agent $i$, we assume a stabilizing $K_i^0$ is known. Then we seek to find symmetric positive definite matrix $P_i^k$ and feedback gain matrix $K_i^{k+1} \in R^{m\times n}$ without requiring $A$ and $B_i$ matrices to be known.

System (8) is rewritten as

$$\dot{\delta}_i = A_k\delta_i + \lambda_i B_i(K_i^k\delta_i + u_i) \tag{15}$$

where $A_k = A - \lambda_i B_i K_i^k$. Then using (14), along the solutions of (15), by (11) and (12) we have

$$\begin{aligned}&\delta_i^T(t+\Delta t)P_i^k\delta_i(t+\Delta t) - \delta_i^T(t)P_i^k\delta_i(t) = \\ &-\int_t^{t+\Delta t}\delta_i^T Q_i^k\delta_i\, d\tau + \\ &2\int_t^{t+\Delta t}(u_i + K_i^k\delta_i)^T RK_i^{k+1}\delta_i\, d\tau\end{aligned} \tag{16}$$

where $Q_i^k = Q + K_i^{kT}RK_i^k$. Note that in (16), the term $\delta_i^T(\mathrm{A}_k{}^T P_i^k + P_i^k \mathrm{A}_k)\delta_i$ depending on unknown matrices $A$ and $B_i$ is replaced by $-\delta_i^T Q_i^k\delta_i$, which can be obtained by measuring $\delta_i$ online. Also, the term $\lambda_i B_i{}^T P_i^k$ containing $B_i$ is replaced by $RK_i^{k+1}$, in which $K_i^{k+1}$ is treated as another unknown matrix to be solved together with $P_i^k$ [41].

Therefore, (16) plays an important role in separating the system dynamics from the iterative process. As a result, the requirement of the system matrices in (11) and (12) can be replaced by the $\delta_i$ and input information $u_i$ measured online. In other words, the information regarding the system dynamics ($A$ and $B_i$ matrices) is embedded in the error states and input which are measured online.

We employ $u_i = -K_i^0\delta_i + e$, with $e$ the exploration noise (for satisfying PE condition), as the input signal for learning in (15), without affecting the convergence of the learning process. Given a stabilizing $K_i^k$, a pair of matrices ($P_i^k$, $K_i^{k+1}$), with $P_i^k = P_i^{kT} > 0$, satisfying (11) and (12) can be uniquely determined without knowing $A$ or $B_i$, under certain condition (Equation (27)).

We employ $\hat{P}_i \in R^{\frac{n\times(n+1)}{2}}$ and $\bar{\delta}_i \in R^{\frac{n\times(n+1)}{2}}$ instead of $P_i \in R^{n\times n}$ and $\delta_i \in R^n$ respectively where

$$\begin{aligned}\hat{P}_i = [&p_{11}, 2p_{12}, ..., 2p_{1n}, \\ &p_{22}, 2p_{23}, ..., 2p_{n-1,n}, p_{nn}]^T_{\frac{n\times(n+1)}{2}}\end{aligned} \tag{17}$$

$$\begin{aligned}\bar{\delta}_i = [&\delta_1^2, \delta_1\delta_2, ..., \delta_1\delta_n, \\ &\delta_2^2, \delta_2\delta_3, ..., \delta_{n-1}\delta_n, \delta_n^2]^T_{\frac{n\times(n+1)}{2}}\end{aligned} \tag{18}$$

Furthermore, by using Kronecker product representation we have:

$$\delta_i^T Q_i^k\delta_i = (\delta_i^T \otimes \delta_i^T)vec(Q_i^k) \tag{19}$$

$$\begin{aligned}&(u_i + K_i^k\delta_i)^T RK_i^{k+1}\delta_i = \\ &u_i^T RK_i^{k+1}\delta_i + \delta_i^T K_i^{kT}RK_i^{k+1}\delta_i = \\ &[(\delta_i^T \otimes \delta_i^T)(I_n \otimes K_i^{kT}R) + \\ &(\delta_i^T \otimes u_i^T)(I_n \otimes R)]vec(K_i^{k+1})\end{aligned} \tag{20}$$

Also, for positive integer $l$, we define matrices $\Delta_{\delta_i\delta_i} \in R^{l\times\frac{n\times(n+1)}{2}}$, $I_{\delta_i\delta_i} \in R^{l\times n^2}$, $I_{\delta_i u_i} \in R^{l\times mn}$ such that

$$\begin{aligned}\Delta_{\delta_i\delta_i} = [&\bar{\delta}_i(t_1) - \bar{\delta}_i(t_0), \bar{\delta}_i(t_2) - \bar{\delta}_i(t_1), \\ &..., \bar{\delta}_i(t_l) - \bar{\delta}_i(t_{l-1})]^T_{l\times\frac{n\times(n+1)}{2}},\end{aligned} \tag{21}$$

$$\begin{aligned}I_{\delta_i\delta_i} = [&\int_{t_0}^{t_1}\delta_i\otimes\delta_i\, d\tau, \int_{t_1}^{t_2}\delta_i\otimes\delta_i\, d\tau, \\ &..., \int_{t_{l-1}}^{t_l}\delta_i\otimes\delta_i\, d\tau]^T_{l\times n^2},\end{aligned} \tag{22}$$

$$\begin{aligned}I_{\delta_i u_i} = [&\int_{t_0}^{t_1}\delta_i\otimes u_i\, d\tau, \int_{t_1}^{t_2}\delta_i\otimes u_i\, d\tau, \\ &..., \int_{t_{l-1}}^{t_l}\delta_i\otimes u_i\, d\tau]^T_{l\times(mn)}\end{aligned} \tag{23}$$

where, $0 \le t_0 < t_1 < ... < t_l$.

Inspired by [41], (16) implies the following matrix form of linear equations for any given stabilizing gain matrix $K_i^k$

$$\Theta_i^k\begin{bmatrix}\hat{P}_i^k \\ vec(K_i^{k+1})\end{bmatrix} = \Xi_i^k \tag{24}$$

where, $\Theta_i^k \in R^{l\times[\frac{n(n+1)}{2}+mn]}$ and $\Xi_i^k \in R^l$ are defined as:

$$\begin{aligned}&\Theta_i^k = [\Delta_{\delta_i\delta_i}, \\ &-2I_{\delta_i\delta_i}(I_n \otimes K_i^{kT}R) - 2I_{\delta_i u_i}(I_n \otimes R)], \\ &\Xi_i^k = -I_{\delta_i\delta_i}\, vec(Q_i^k)\end{aligned} \tag{25}$$

Notice that if $\Theta_i^k$ has full column rank, (24) can be directly solved as follows:

$$\begin{bmatrix} \hat{P}_k \\ vec(K_{k+1}) \end{bmatrix} = (\Theta_i^{k^T}\Theta_i^k)^{-1}\Theta_i^{k^T}\Xi_i^k \qquad (26)$$

The steps of the proposed optimal adaptive control algorithm for practical online implementation are presented as follows:

**Algorithm 2 (Optimal adaptive learning algorithm):**

*Step 1:* For the agent $i$ employ $u_i = -K_i^0\delta_i + e$ as the input on the time interval $[t_0, t_l]$, where $K_i^0$ is stabilizing and $e$ is the exploration noise (to satisfy PE condition). Compute $\Delta_{\delta_i\delta_i}$, $I_{\delta_i\delta_i}$ and $I_{\delta_i u_i}$ until the rank condition in (27) below is satisfied.

Let $k = 0$.

*Step 2:* Solve $\hat{P}_i^k$ and $K_i^{k+1}$ from (26).

*Step 3:* Let $k+1 \to k$, and repeat Step 2 until $\left\|P_i^k - P_i^{k-1}\right\| \le \beta$ for $k \ge 1$, where the constant $\beta > 0$ is a predefined small threshold.

*Step 4:* Use $u_i = -K_i^k\delta_i = -R^{-1}B_i^{\,T}P_i^{k*}\delta_i$ as the approximated optimal control policy for each agent $i$.

It must be noted that in the cases where the solution of (24) does not exist due to the numerical error in $I_{\delta_i\delta_i}$ and $I_{\delta_i u_i}$ computations, the solution of (26) can be obtained by employing the least square solution of (24).

**Lemma 1.** As proved in [41], the convergence is guaranteed, if $\Theta_i^k$, $i = 1,2,...,N$ has full column rank for all $k$, $k = 0,1,2,...$; therefore, there exists an integer $l_0 > 0$, such that, for all $l > l_0$,

$$rank([I_{\delta_i\delta_i}, I_{\delta_i u_i}]) = \frac{n(n+1)}{2} + mn \qquad (27)$$

**Theorem 2.** Using an initial stabilizing control policy $K_i^0$ with exploration noise, once the online information of $\Delta_{\delta_i\delta_i}$, $I_{\delta_i\delta_i}$ and $I_{\delta_i u_i}$ matrices (satisfying the rank condition (27)) is computed, the iterative process of Algorithm 2 results in a sequence of $\{P_i^k\}_{k=0}^{\infty}$ and $\{K_i^k\}_{k=1}^{\infty}$ which respectively converges to the optimal values $P_i^*$ and $K_i^*$.

**Proof:** See [41] for the similar proof.

Several types of exploration noise, such as random noise [44,45], exponentially decreasing probing noise [32] and sum of sinusoids noise [41] are added to the input in reinforcement learning problems. The input signal should be persistently exciting; therefore, the generated signals from the system, which contains the information of the unknown system dynamics, are rich enough to lead us to the exact solution. Here is a sum of sinusoids noise applied in the simulations to satisfy PE condition.

**Remark 1.** In comparison with the previous research on MASs leader-follower consensus, which is mostly offline and requires the complete knowledge of the system dynamics, this paper has presented an online optimal adaptive controller for the leader-follower consensus, which does not require the knowledge of drift and input matrices of the linear agents.

**Remark 2.** The main advantage of the proposed method is that the introduced optimal adaptive learning method is an online model-free ADP algorithm.

Moreover, this technique iteratively solves the algebraic Riccati equation using the online information of state and input, without requiring the priori knowledge of the system matrices and all iterations can be conducted by using repeatedly the same state and input information ($I_{\delta_i\delta_i}$, $I_{\delta_i u_i}$, $\Delta_{\delta_i\delta_i}$) on some fixed time intervals. However, the main burden in implementing the introduced optimal adaptive method (Algorithm 2) is the computation of $I_{\delta_i\delta_i} \in R^{l\times n^2}$ and $I_{\delta_i u_i} \in R^{l\times mn}$ matrices. The two matrices can be implemented using $n^2 + mn$ integrators in the learning system to collect information of the error state and the input.

## 5. Simulation results

In this section, we give an example to illustrate the validity of the proposed methods. Consider the graph structure shown in figure 1, similar to [42] focusing on the dynamic of each agent, which is as follows

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

$$\dot{x}_3 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix}x_3 + \begin{bmatrix} 2 \\ 2 \end{bmatrix}u_3,\ \dot{x}_4 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix}x_4 + \begin{bmatrix} 1 \\ 1 \end{bmatrix}u_4,$$

$$\dot{x}_5 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix}x_5 + \begin{bmatrix} 3 \\ 2 \end{bmatrix}u_5$$

with target generator (leader) $\dot{x}_0 = \begin{bmatrix} -2 & 1 \\ -4 & -1 \end{bmatrix}x_0$.

The Laplacian $L$ and matrix $G$ are as follows:

$$L=\begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ -1 & -1 & 2 & 0 & 0 \\ -1 & 0 & 0 & 2 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix},\quad G=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The cost function of parameters for each agent, namely the Q and R matrices, is chosen to be identity matrices of appropriate dimensions. Since agents dynamics are already stable, the initial stabilizing feedback gains are considered as $K_i^0=[0 \quad 0], i=1,2,...,5$.

First we assume that $A$ and $B_i$ matrices are precisely known and we employ the Kleinman policy iteration (Algorithm 1) to reach leader-follower consensus. Figure 2 shows the convergence of $\delta_1,\delta_2,\delta_3,\delta_4,\delta_5$ components trajectories to zero by time in 6 iterations, which confirm the synchronization of all agents to the leader.

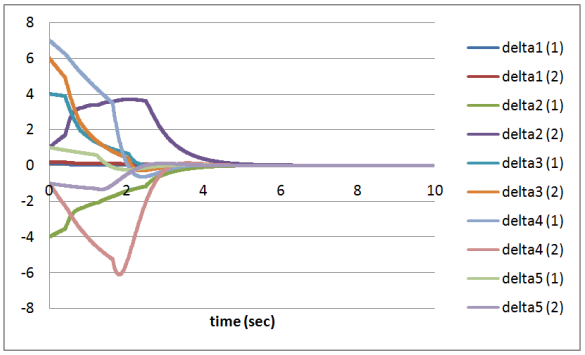


**Figure 2. Agents $\delta_i, i=1,...,5$ trajectories under known dynamics.**

The error difference between the parameters of the solution $P_i^k, i=1,2,3,4,5$ obtained iteratively and the optimal solution $P_i^*$, obtained by directly solving the ARE, is in the range of $10^{-4}$.

Now we assume that $A$ and $B_i$ matrices are unknown and we employ the optimal adaptive learning method (Algorithm 2).

It must be mentioned that the precise knowledge of $A$ and $B_i$ is not used in the design of optimal adaptive controllers. The initial values for the state variables of each agent are randomly selected near the origin. From $t=0\ s$ to $t=2\ s$ the following exploration noise is added to the agents' inputs to meet the PE requirement, where $w_i, i=1,2,...,100$ is randomly selected from $[-500,500]$.

$$e=0.01\sum_{i=1}^{100}\sin(w_i t) \tag{28}$$

$\delta_i$ and $u_i$ information of each agent is collected over each interval of 0.1 s. The policy iteration started at $t=2\ s$, and convergence is attained after 10 iterations, when the stopping criteria $\left\|P_i^k-P_i^{k-1}\right\|\le 0.001$ are satisfied for each $i=1,2,3,4,5$. Figures 3 and 4 illustrate the convergence of $P_i^k$ to $P_i^*$ and $K_i^k$ to $K_i^*$ for $i=1,2,3,4,5$ respectively during 10 iterations.

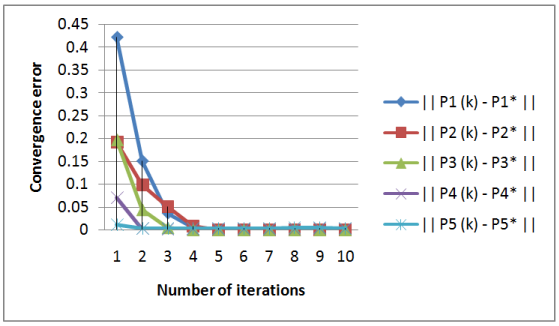


**Figure 3. Convergence of $P_i^k$ to $P_i^*$ during learning iterations.**

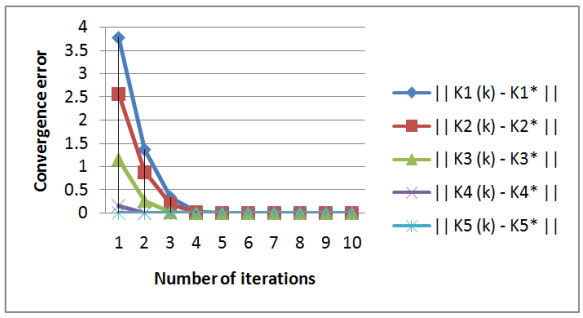


**Figure 4. Convergence of $K_i^k$ to $K_i^*$ during learning iterations.**

The controller $u_i=-K_i^*\delta_i=-R^{-1}B_i^T P_i^*\delta_i$ is used as the actual control input for each agent $i,\ i=1,2,...,5$ starting from $t=2\ s$ to the end of the simulation. The convergence of $\delta_1,\delta_2,\delta_3,\delta_4,\delta_5$ components to zero is depicted in figure 5 where the synchronization of all agents to the leader is guaranteed.

As mentioned in table 1, the Kleinman PI method after 6 iterations results in leader-follower consensus in 6 seconds under known dynamics. The introduced optimal adaptive PI learns the optimal policy and guarantees the leader-follower consensus in 12 seconds after 10 iterations under unknown dynamics. Clearly, the introduced optimal adaptive method for unknown dynamics

requires more time and iterations in comparison with the method for known dynamics to converge to the optimal control policies.

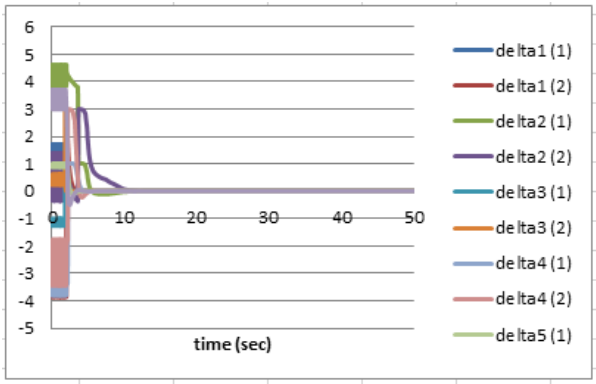


**Figure 5. Agents $\delta_i, i = 1,...,5$ trajectories under unknown dynamics.**

**Table 1. Online PI methods comparison under known and unknown dynamics.**

| Online method | $\delta_i$ Convergence time to zero | $A$ and $B_i$ matrices | Number of iterations |
|---|---|---|---|
| **Kleinman PI** | 6 seconds | Known | 6 |
| **Optimal Adaptive PI** | 12 seconds | Unknown | 10 |

As illustrated in the simulation results by employing PI technique and optimal adaptive learning algorithm, all agents synchronize to the leader.

## 6. Conclusions

In this paper, the online optimal leader-follower consensus problem for linear continuous time systems under known and unknown dynamics is considered. The multi-agent global error dynamic is decoupled to simplify the employment of policy iteration and optimal adaptive control techniques for leader-follower consensus under known and unknown dynamics respectively. The online optimal adaptive control solves the algebraic Riccati equation iteratively using system error state and input information collected online for each agent, without knowing the system matrices. Graph theory is employed to show the network topology of the multi-agent system, where the connectivity of the network graph is assumed as a key condition to ensure leader-follower consensus. Simulation results indicate the capabilities of the introduced algorithms.

## References

[1] Zhang, H., Lewis, F. & Qu, Z. (2012). Lyapunov, Adaptive, and Optimal Design Techniques for Cooperative Systems on Directed Communication Graphs. IEEE transactions on industrial electronics, vol. 59, pp. 3026 – 3041.

[2] Ren, W., Beard, R. & Atkins, E. (2007). Information consensus in multi vehicle cooperative control. IEEE Control Syst. Mag, vol. 27. No. 2, pp. 71–82.

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

[4] Li, X., Wang, X. & Chen, G. (2004). Pinning a complex dynamical network to its equilibrium. IEEE Transactions on Circuits and Systems. I. Regular Papers, vol. 51, no. 10, pp. 2074–2087.

[5] Zhang, H. & Lewis, F. (2012). Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics. Automatica, vol. 48, pp. 1432–1439.

[6] Ren, W., Moore, K. & Chen, Y. (2007). High-order and model reference consensus algorithms in cooperative control of multi vehicle systems. Journal of Dynamic Systems, Measurement, and Control, vol. 129, no. 5, pp. 678–688.

[7] Wang, X. and Chen, G. (2002). Pinning control of scale-free dynamical net-works. PhysicaA, vol. 310(3-4), pp. 521–531.

[8] Kailath, T. (1973). Some new algorithms for recursive estimation in constant linear systems. IEEE Transactions on Information Theory, vol. 19, no. 6, pp. 750-760.

[9] MacFarlane, A. G. J. (1963). An eigenvector solution of the optimal linear regulator problem. Journal of Electronics and Control, vol. 14, pp. 643-654.

[10] Potter, J. E. (1966). Matrix quadratic solutions. SIAM Journal on Applied Mathematics, vol. 14, pp. 496-501.

[11] Laub, A. J. (1979). A Schur method for solving algebraic Riccati equations. IEEE Transactions on Automatic Control, vol. 24, no. 6, pp. 913-921.

[12] Balzer, L. A. (1980). Accelerated convergence of the matrix sign function method of solving Lyapunov, Riccati and other equations. International Journal of Control, vol. 32, no. 6, pp. 1076-1078.

[13] Byers, R. (1987). Solving the algebraic Riccati equation with the matrix sign. Linear Algebra and its Applications, vol. 85, pp. 267-279.

[14] Hasan, M. A., Yang, J. S. & Hasan, A. (1999). A method for solving the algebraic Riccati and Lyapunov equations using higher order matrix sign function algorithms. In: Proc. Of ACC, pp. 2345-2349.

[15] Banks, H. T. & Ito, K. (1991). A numerical algorithm for optimal feedback gains in high dimensional linear quadratic regulator problems. SIAM

Journal on Control and Optimization, vol. 29, no. 3, pp. 499-515.

[16] Guo, C. H. & Lancaster, P. (1998). Analysis and modification of Newton's method for algebraic Riccati equations. Mathematics of Computation, vol. 67, no. 223, pp. 1089-1105.

[17] Kleinman, D. (1968). On an iterative technique for Riccati equation computations. IEEE Transactions on Automatic Control, vol. 13, no. 1, pp. 114–115.

[18] Moris, K. & Navasca, C. (2006). Iterative solution of algebraic Riccati equations for damped systems. In: Proc. of CDC, pp. 2436-2440.

[19] Sastry, S. & Bodson, M. (1989). Adaptive Control: Stability, Convergence, and Robustness. Englewood Cliffs, NJ: Prentice-Hall.

[20] Slotine, J. E. & Li W. (1991). Applied Nonlinear Control. Englewood Cliffs, NJ: Prentice-Hall.

[21] Sutton, R. S. & Barto, A. G. (1998). Reinforcement learning-an introduction. Cambridge, Massachusetts: MIT Press.

[22] Watkins C . J. C. H. (1989). Learning from delayed rewards, PhD Thesis, University of Cambridge, England.

[23] Werbos P. (1989), Neural networks for control and system identification, Proc. Of CDC'89, pp. 260–265.

[24] Werbos, P. J. (1992). Approximate dynamic programming for real-time control and neural modeling. In [24] D. A. White, and D. A. Sofge (Eds.), Hand book of intelligent control. New York: Van Nostrand Reinhold.

[25] Bertsekas, D. P. & Tsitsiklis, J. N. (1996). Neuro-dynamic programming. MA: Athena Scientific.

[26] Vrabie, D., Pastravanu, O., Lewis, F. L. & Abu-Khalaf, M. (2009). Adaptive optimal control for continuous-time linear systems based on policy iteration. Automatica, vol. 45, no. 2, pp. 477–484.

[27] Busoniu, L., Babuska, R. & DeSchutter, B. (2008). 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.

[28] Littman, M. L. (2001). Value-function reinforcement learning in Markov games. Journal of Cognitive Systems Research, vol. 2, pp. 55-66.

[29] Dierks, T. & Jagannathan, S. (2010). Optimal control of affine nonlinear continuous- time systems using an online Hamilton–Jacobi–Isaacs formulation. In Proc. IEEE conf. decision and control, pp. 3048–3053.

[30] Johnson, M., Hiramatsu, T., Fitz-Coy, N. & Dixon, W. E. (2010). Asymptotic stackelberg optimal control design for an uncertain Euler lagrange system. In IEEE conference on decision and control, pp. 6686–6691.

[31] Vamvoudakis, K. G. & Lewis, F. L. (2010). On line actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. Automatica, vol. 46, no. 5, pp. 878–888.

[32] Vamvoudakis, K. G. & Lewis, F. L. (2011). Multi-player non-zero sum games:online adaptive learning solution of coupled Hamilton–Jacobi equations. Automatica, vol. 47, no. 8, pp.1556–1569.

[33] Jadbabaie, A., Lin, J. & Morse, A. S. (2007). Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on Automatic Control, vol. 48, no. 6, pp. 943-948.

[34] Hong, Y., Gao, L., Cheng, D. & Hu, J. (2007). Lyapunov-based approach to multi agent systems with switching jointly connected interconnection. IEEE Transactions on Automatic Control, vol. 52, no. 5, pp. 943-948.

[35] Cheng, D., Wang, J. & Hu, X., (2008). An extension of Lasall's invariance principle and its application to multi-agent consensus. IEEE Transactions on Automatic Control, vol. 53, no. 7, pp. 1765-1770.

[36] Dimarogonasa, D. V., Tsiotras, P. & Kyriakopoulos, K. J. (2009). Leader-follower cooperative attitude control of multiple rigid bodies. Systems and Control Letters, vol. 58, no. 6, pp. 429-435.

[37] Consolini, L., Morbidi, F., Prattichizzo, D. & Tosques, M. (2008). Leader-follower formation control of nonholonomic mobile robots with input constraints. Automatica, vol. 44, no. 5, pp. 1343-1349.

[38] Peng, K. & Yanga, Y. (2009). Leader-follower consensus problem with a varying-velocity leader and time-varying delays. PhysicaA, vol. 388, no. 2-3, pp., 193-208.

[39] Olfati-Saber, R. & Murray, R. M. (2003). Consensus protocols for networks of dynamic agents. in: Proceedings of 2003 American Control Conference.

[40] Vrabie, D.(2009). Online adaptive optimal control for continuse time systems, Ph. D. thesis, The University of Texas at Arlington.

[41] Jiang, Y. & Jiang, Z. P. (2012). Computational adaptive optimal control for continuous time linear systems with completely unknown dynamics. Automatica, vol. 48, pp. 2699-2704.

[42] Vamvoudakis, K. G., Lewis, F. L. & Hudas, G. R. (2012). Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality. Automatica, vol. 48, pp. 1598–1611.

[43] Ni, W. & Cheng, D. (2010). Leader-follower consensus of multi-agent systems under fixed and switching topologies. Systems and Control Letters, vol. pp. 59, 209-217.

[44] Al-Tamimi, A., Lewis, F. L. & Abu-Khalaf, M. (2007). Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. Automatica, vol. 43, no. 3, pp. 473–481.

[45] Xu, H., Jagannathan, S. & Lewis, F. L. (2012). Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses. Automatica, vol. 48, no. 6, pp. 1017–1030.

نشریه هوش مصنوعی و داده کاوی

دانشگاه شاهرود

# اجماع پیرو-رهبر بهینه تطبیقی سیستم‌های چندعاملی خطی با دینامیک معین و نامعین

**فرزانه تاتاری و محمدباقر نقیبی سیستان***

**گروه مهندسی برق، دانشکده مهندسی، دانشگاه فردوسی مشهد، مشهد، ایران.**

ارسال ۲۰۱۴/۰۵/۰۵؛ پذیرش۲۰۱۵/۰۵/۱۳

**چکیده:**

در این مقاله اجماع پیرو-رهبر بهینه تطبیقی سیستم‌های چندعاملی زمان پیوسته خطی مورد بررسی قرار گرفته است. دینامیک خطای هر عامل به اطلاعات همسایه‌های آن بستگی دارد. تحلیل دقیق اجماع پیرو-رهبر بهینه برخط تحت دینامیک‌های معین و نامعین در این مقاله ارایه شده است. الگوریتم‌های ارایه شده، به یادگیری حل تقریبی معادلات جبری ریکاتی بر مبنای یادگیری تقویتی، می‌پردازند. تکنیک کنترل بهینه تطبیقی برای حل تکرار شونده معادله جبری ریکاتی بر اساس اطلاعات حالت خطای اندازه‌گیری شده و اطلاعات ورودی برخط، طراحی شده است که نیاز به دانش اولیه از ماتریس‌های دینامیک عامل‌ها ندارد. جداسازی دینامیک خطای همه جایی سیستم چندعاملی، باعث سهولت در به کارگیری الگوریتم تکرار سیاست و تکنیک‌های کنترل بهینه تطبیقی برای حل مسئله اجماع پیرو-رهبر تحت دینامیک معین و نامعین می‌گردد.

**کلمات کلیدی:** تئوری بازی، اجماع پیرو-رهبر، سیستم‌های چندعاملی، تکرار سیاست.