Optimal consensus control for double-integrator multi-agent systems with unknown dynamics using adaptive dynamic programming

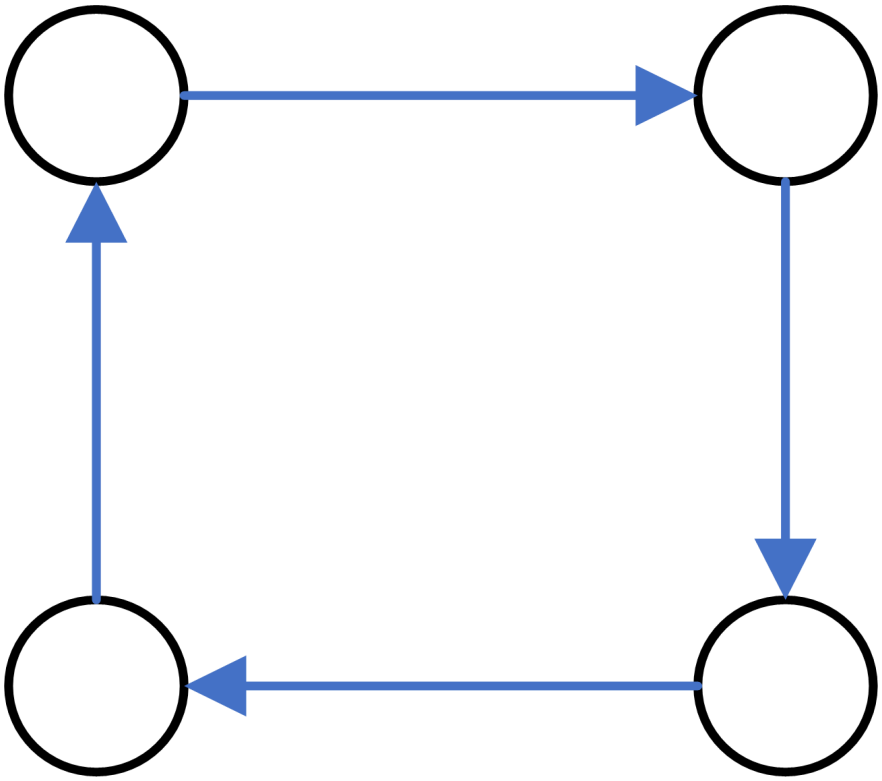
Yang Yang[1], Qi Zhang[1], Xue Song[1], Xiaoran Xie[1], Naibo Zhu[1], and Zhi Liu[1]

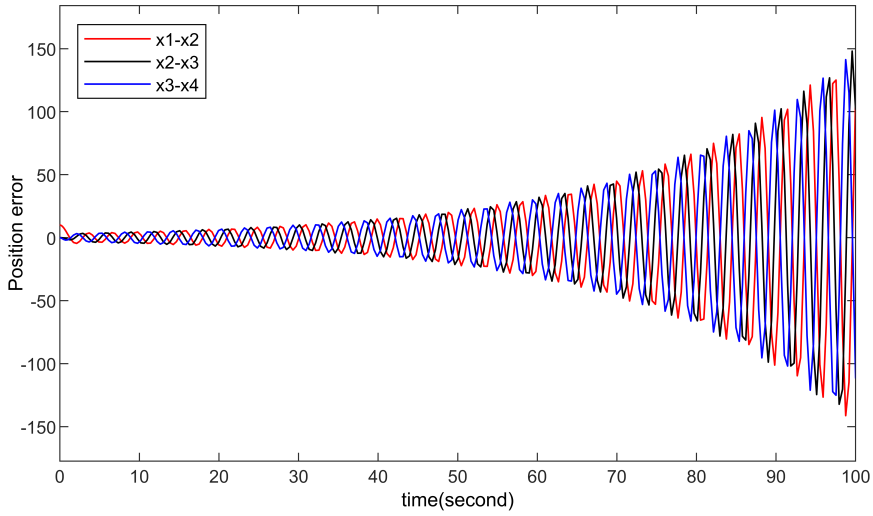
[1]Changchun University of Science and Technology

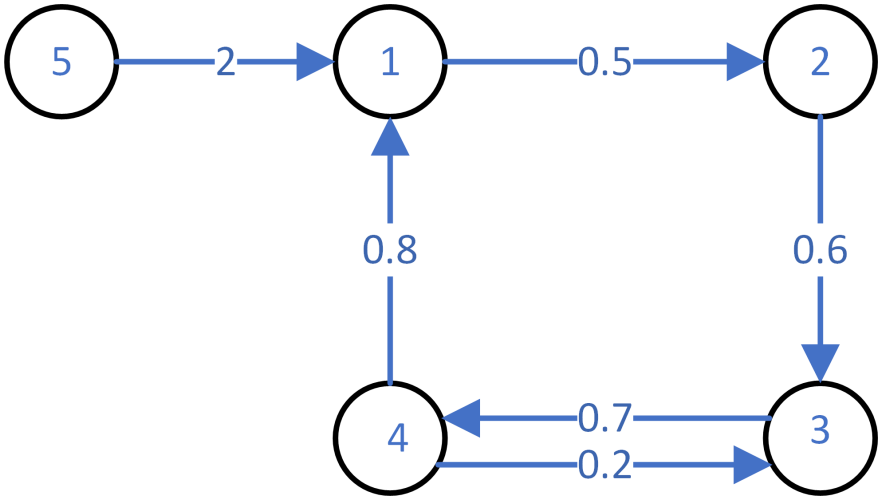
January 23, 2023

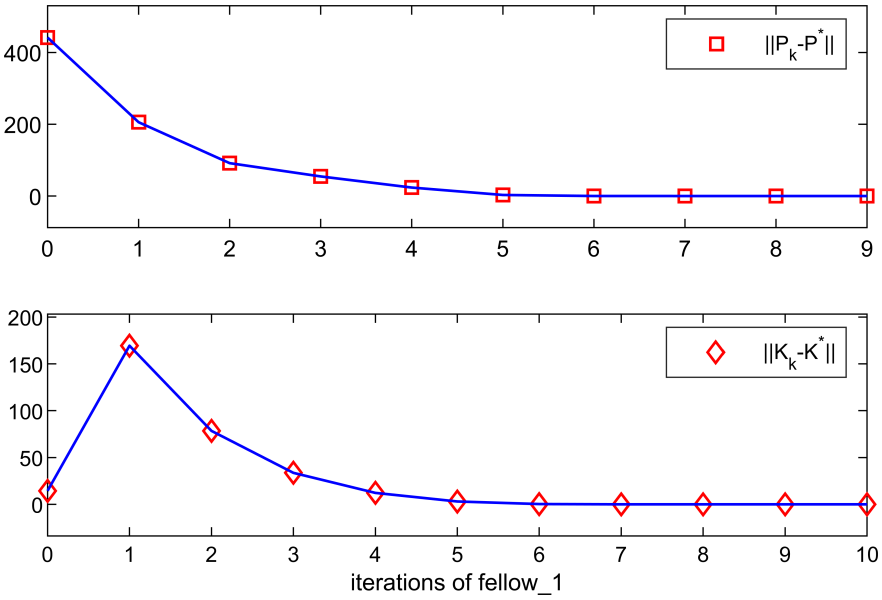
Abstract

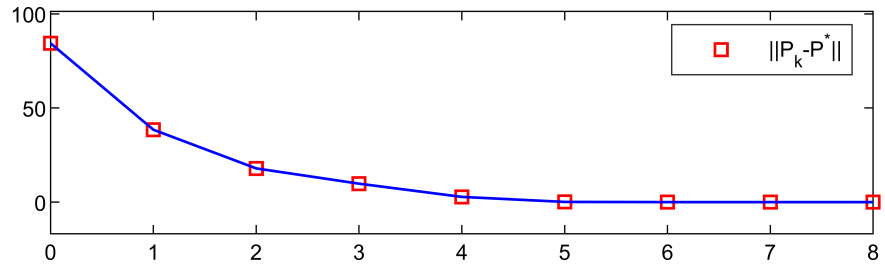
The purpose of this paper is to utilize adaptive dynamic programming to solve an optimal consensus problem for double-integrator multi-agent systems with completely unknown dynamics. In double-integrator multi-agent systems, flocking algorithms that neglect agents' inertial effect can cause unstable group behavior. Despite the fact that an inertias-independent protocol exists, the design of its control law is decided by dynamics and inertia. However, inertia in reality is difficult to measure accurately, therefore, the control gain in the consensus protocol was solved by developing adaptive dynamic programming to enable the double-integrator systems to ensure the consensus of the agents in the presence of entirely unknown dynamics. Firstly, we demonstrate in a typical example how flocking algorithms that ignore the inertial effect of agents can lead to unstable group behavior. And even though the protocol is independent of inertia, the control gain depends quite strongly on the inertia and dynamic of the agent. Then, to address these shortcomings, an online policy iteration-based adaptive dynamic programming is designed to tackle the challenge of double-integrator multi-agent systems without dynamics. Finally, simulation results are shown to prove how effective the proposed approach is.

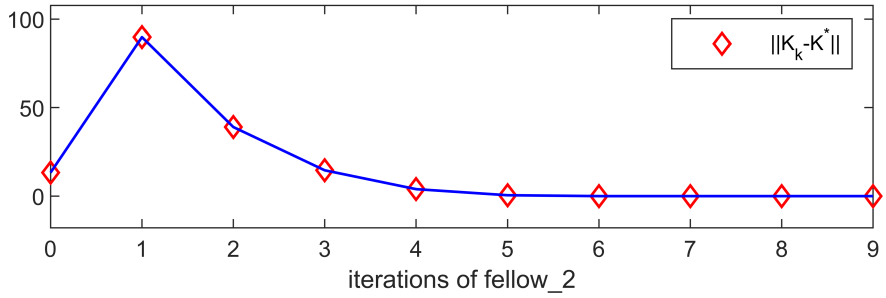


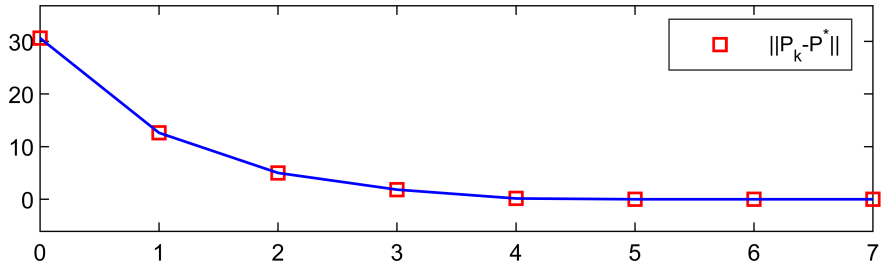


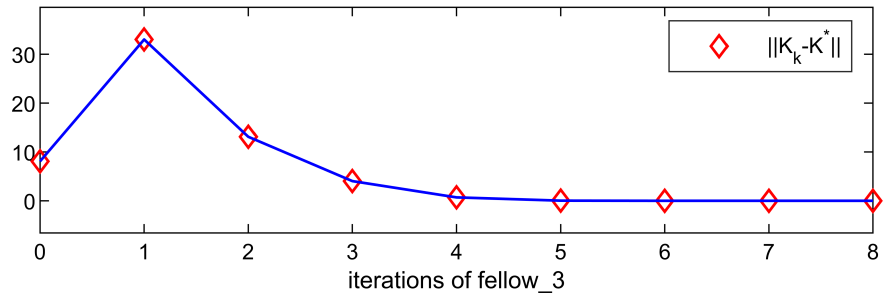


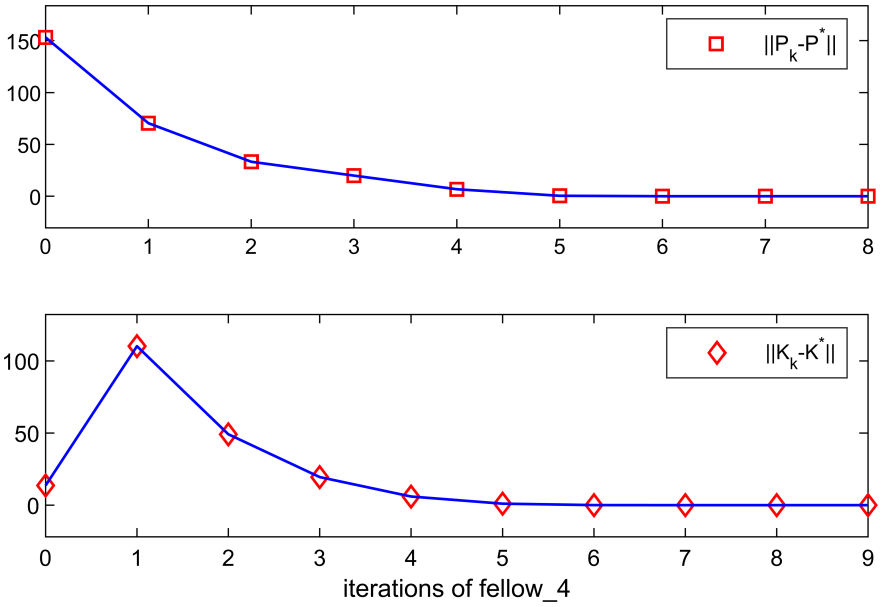


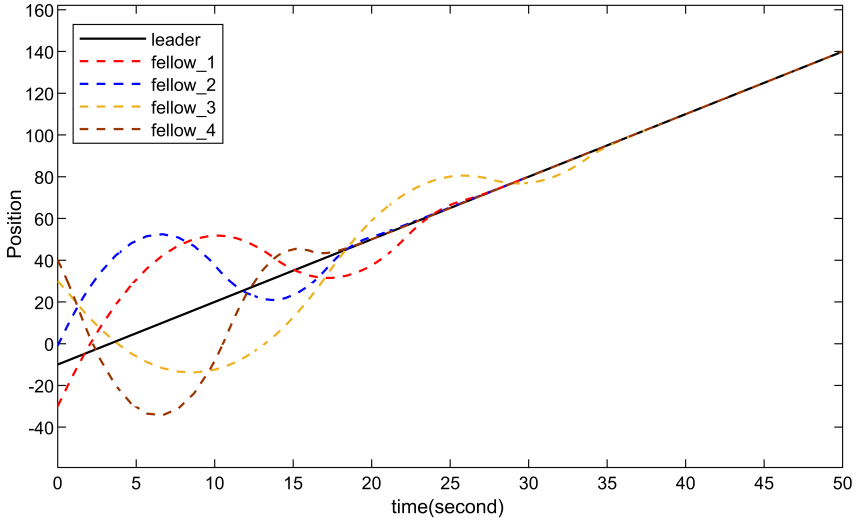


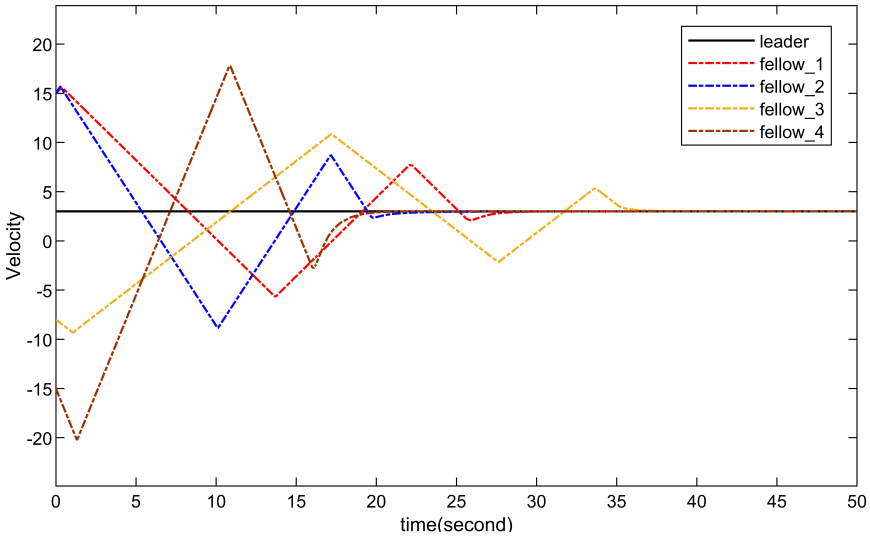












Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Optimal consensus control for double-integrator multi-agent systems with unknown dynamics using adaptive dynamic programming

Qi Zhang | Yang Yang* | Xue Song | Xiaoran Xie | Naibo Zhu | Zhi Liu

The College of Electronic and Information Engineering,Changchun University of Science and Technology, Changchun, Jilin, China

Correspondence
*Yang Yang, The College of Electronic and Information Engineering, Changchun University of Science and Technology, Changchun130022, Jilin, China. Email: yangyang@cust.edu.cn

Present Address
National Natural Science Foundation of China under Grant U2141231.

Abstract

The purpose of this paper is to utilize adaptive dynamic programming to solve an optimal consensus problem for double-integrator multi-agent systems with completely unknown dynamics. In double-integrator multi-agent systems, flocking algorithms that neglect agents' inertial effect can cause unstable group behavior. Despite the fact that an inertias-independent protocol exists, the design of its control law is decided by dynamics and inertia. However, inertia in reality is difficult to measure accurately, therefore, the control gain in the consensus protocol was solved by developing adaptive dynamic programming to enable the double-integrator systems to ensure the consensus of the agents in the presence of entirely unknown dynamics. Firstly, we demonstrate in a typical example how flocking algorithms that ignore the inertial effect of agents can lead to unstable group behavior. And even though the protocol is independent of inertia, the control gain depends quite strongly on the inertia and dynamic of the agent. Then, to address these shortcomings, an online policy iteration-based adaptive dynamic programming is designed to tackle the challenge of double-integrator multi-agent systems without dynamics. Finally, simulation results are shown to prove how effective the proposed approach is.

KEYWORDS:
multi-agent systems, adaptive dynamic programming, optimal consensus control, data-driven, reinforcement learning

1 | INTRODUCTION

In recent years, multi-agent systems have seen increased usage in physics, social sciences, biology, and engineering [1,2,3]. It can be utilized to address issues that are difficult for a single individual to solve. Therefore, distributed control of multi-agent systems has gained much interest. In the multi-agent distributed control problem, the consensus problem is of great practical significance and theoretical value as the basis of cooperative control among agents. These publications are regarded as seminal works on consensus problem in control theory.[4,5] Since the communication graph affects information transfer globally, it affects flocking behavior in multi-agent systems. Solving the consensus problem is a challenge due to the complex communication graph between the agents. To achieve consensus convergence, several assumptions are placed on the communication graph using graph theory, and some protocols are suggested in some studies.[6,7,8,9] However, the protocols discussed above are not distributed, and the threshold value set by the Laplacian matrix must be exceeded by these designed protocols. In some studies, [10,11] there have been

designs for completely distributed adaptive protocols to address this limitation. These protocols are constructed using high-gain adaptive feedback, which involves only information about itself and the neighboring regions and tends to rise monotonically over time to a limited value.

Most multi-agent consensus studies presume that each agent's dynamics can be captured by a single integrator.[12,13,14] However, in reality, the double-integrator frequently simulates more critical systems. For example, single-axis spacecraft rotation,[15] rotary crane motion,[16] and the dynamics of a spacecraft.[17] Therefore, research interest has been aroused by the consensus problem of double integrator multi-agent systems. However, since most actuators can only change acceleration through agents' inertias, it is not possible to regulate the velocity directly in many significant applications.[18,19] But, inertial action may generate unstable group behavior in a certainly directed information topology. Think of a disturbed protocol whose gain is determined by the Laplacian matrix and the agent inertias, in this case, double integrator multi-agent systems can achieve consensus convergence over strongly connected balanced graph.[20] Unfortunately, the protocol has the disadvantage of relying on global rather than local information, and thus it is not fully distributed.

However, current research requires a thorough understanding of the dynamics of the agents, which is difficult in many practical situations. Previous research has gone into great detail about the design of adaptive controllers for uncertain linear systems. The traditional approach to designing adaptive optimum control laws is to first calculate the algebraic Riccati equation based on the system parameters.[21,22] This issue requires precise dynamics, which is difficult since most systems in practice are too complicated, and the resulting dynamics may be inaccurate. So it is necessary to find a proven method to solve this problem.

Reinforcement learning (RL) has been frequently employed in recent years to address optimal solution problems.[23,24,25,26] It was first observed in the learned behavior of humans and other mammals, RL adopts a learning-by-acquiring approach, updating its model after acquiring a sample, using the current model to guide the next action, and updating the model after the next action is rewarded, iterating and repeating until the model converges. A very important point in this process is "If the current model is available, what is the best way to choose the next step to improve the current model". This brings us to two very important concepts in RL: exploration, which is the selection of previously unexecuted actions to explore more possibilities, and exploitation, which is the selection of executed actions to refine the model of a known action.

With the development of machine learning, information science, and data science, some academics are beginning to undertake study using data-driven concepts.[27,28] Data collected from the system, whether online or offline, can be utilized immediately to perform status analysis and optimization, system modeling, and controller design. Quickly emerging are several data-driven model-free control techniques. Without depending on standard mathematical models, these methods may efficiently deal with unmodeled system dynamics and disturbances. The method known as adaptive dynamic programming (ADP) has been created recently and is a desirable data-driven method. ADP is an example of a reinforcement learning technique that combines the benefits of adaptive and optimal control.[29,30] Among several RL approaches, ADP is regarded as one of the fundamental ways for achieving optimum control laws for a variety of optimal control issues since it has strong self-learning and self-adaptive capabilities and has evolved into an essential optimal control method that is similar to the brain. ADP was known by numerous names, including "adaptive critic designs",[31] "approximate dynamic programming",[32,33] and "neural dynamic programming".[33] It contains both value iteration and policy iteration.[34] It was verified that the value iterative ADP method is convergence.[34] It is impossible, however, to guarantee the system's stability under the value iterative control law.[35] A policy iteration is provided for optimal control of the continuous-time system.[36] Using policy iteration, continuous-time complex-valued systems have been successfully solved.[37] And discrete-time policy iteration with convergence and stability proof was developed.[38] The optimal distributed control problems on multi-agent systems have recently been addressed using ADP techniques.[39,40,41,42] To address directed graph multi-agent systems' optimal control problems, an unique ADP approach was created.[39] By utilizing policy iteration algorithms, adaptive learning solutions for multi-agent differential graphical games were obtained.[40] The optimal consensus problem for continuous-time nonlinear multi-agent systems was solved using fuzzy adaptive dynamic programming with a policy iteration online framework.[41] A unique method for continuous-time heterogeneous multi-agent differential graphical games was proposed.[42] Through the use of value iteration techniques, some papers studied multi-agent discrete-time graphical games.

In this paper, we propose an online ADP technique for solving the optimal consensus issue for a type of continuous-time double integrator multi-agent systems with uncertain system dynamics, using system data rather than exact dynamics and inertia. Because the double-integrator multi-agent systems is unstable under certain topologies when inertia is not considered, we propose an inertia-independent protocol. But the design of their control laws is reliant on the dynamics of agents, including the precise inertia.[43] In practice, however, accurate inertia is frequently unavailable, resulting in dynamic uncertainty.[44] It is difficult to develop distributed controllers for double-integrator multi-agent systems. Finally, the control gain of the consensus protocol can be solved by the proposed adaptive dynamic programming based on online policy iterations.

As a result of the above observations, the contributions of this study, as compared to earlier studies, can be summed up in two aspects: 1) As compared to adaptive nonlinear gain, the offered consensus protocols are developed with linear constant gain, making them easier to implement. This protocol is applicable to multi-agent systems with any number of agents and any communication topology since they only need local information from their own and neighbors. 2) An online policy iteration-based adaptive dynamic programming is designed to tackle the challenge of double-integrator multi-agent systems without dynamics.

Organize the remainder of the paper as follows: In Section 2, there is some algebraic graph theory knowledge presented and problem formulation is derived. Section 3 shows the importance of the agents' inertial effect and an inertia-independent protocol. Section 4 presents an online ADP algorithm to find the best solution to the optimal consensus problem of the double-integrator multi-agent systems without requiring prior knowledge of the system dynamics. Section 5 provides an example to show the effectiveness of our proposed approach. Finally, in Section 6, we provide a brief conclusion.

Notation. This paper refers to the set of real numbers as $\mathbf{R}$. $\otimes$ denotes the Kronecker product. Let $\min_{i=1,2,\ldots n}\left\{\mathrm{Re}\left(\lambda_i(A)\right)\right\}$ be denoted by $\alpha(A)$ and $\max_{i=1,2,\ldots n}\left\{\mathrm{Re}\left(\lambda_i(A)\right)\right\}$ be denoted by $\beta(A)$. $\sigma_\tau(u)$ is denoted by $\sigma_\tau(u)=\left[\mathrm{sign}\left(u_1\right)\min\left\{\tau,\left|u_1\right|\right\},\mathrm{sign}(u)\min\left\{\tau,\left|u_2\right|\right\},\ldots,\mathrm{sign}\left(u_m\right)\min\left\{\tau,\left|u_m\right|\right\}\right]^{\mathrm{T}}$, brevity, we use $\sigma(u)$ to denote $\sigma_1(u)$.

2 | PRELIMINARY KNOWLEDGE

This section will introduce the necessary algebraic graph theory and problem formulation.

2.1 | Graph theory

In this study, graph theory is employed as a highly useful mathematical tool to examine multi-agent systems. A weighted graph may explain the architecture of a communication network, regardless of whether the information flow is unidirectional or bidirectional.

Let $G(\mathcal{N},\epsilon,\mathcal{R})$ be a weighted graph, with $\mathcal{N}=\{1,2,\ldots,N+1\}$ representing the set of nodes, the edge set is denoted by $\epsilon\subseteq\mathcal{N}\times\mathcal{N}$, and $\mathcal{R}=\left[r_{ij}\right]\in R^{(N+1)\times(N+1)}$ is the matrix of weighted adjacency. Node j can obtain information from node i, as noted by $(i,j)\in\epsilon$. If $(j,i)\in\epsilon$, $r_{ij}>0$ otherwise, $r_{ij}=0$, then $r_{ii}=0$. It is denoted as $\mathcal{D}=\mathrm{diag}\left(\mathcal{D}_1,\ldots,\mathcal{D}_{N+1}\right)\in R^{(N+1)\times(N+1)}$, $\mathcal{D}_i=\sum_{j\in\mathcal{F}_i}r_{ij}$, where F_i represents the neighborhood set of node i. $L=\left[l_{ij}\right]=\mathcal{D}-\mathcal{R}$ corresponds to the Laplacian matrix. Our study examines the consensus problem when a leader with no in-neighbors. As a result, It is also possible to write L as: $L=\begin{bmatrix}L_1 & L_2\\ 0_{1\times N} & 0\end{bmatrix}$, where $L_1\in R^{N\times N}$, $L_2\in R^{N\times N}$

Assumption 1. A directed spanning tree is rooted at the leader of the graph $G(\mathcal{N},\epsilon,\mathcal{R})$.

Assumption 2. There is no known value for the inertias $m_i, i=1,2,\ldots,N+1$, but they are positive constants.

Lemma 1. [45] Assume that Assumption 1 holds. Then L is a nonsingular matrix with positive real parts in its eigenvalues.

2.2 | Problem formulation

Multi-agent systems with double-integrators are discussed. The dynamics of the $i-th$ followers:

$$\dot{x}_i(t)=Ax_i(t)+\frac{1}{m_i}B\sigma\left(u_i(t)\right),i=1,2,\ldots,N \tag{1}$$

where $x_i(t)$ is the system state vector, $u_i(t)$ is the system state vector, and m_i is the inertia of the $i-th$ follower, system dynamics matrix A and input matrix B are constant matrices.

A reference system, also called leader, is defined as:

$$\dot{x}_{N+1}(t)=Ax_{N+1}(t) \tag{2}$$

Assume agent i exclusively gathers local information about itself and its neighbors, specifically, the signal:

$$z_i=\sum_{j=1}^{N+1}r_{ij}\left(x_i-x_j\right) \tag{3}$$

The consensus problem addressed in this study relates to the fact that the followers' states converge toward the leader's state:

$$\lim_{t\to\infty} \left\| x_i(t) - x_{N+1}(t) \right\|_2 = 0, i = 1, 2, \ldots, N \tag{4}$$

3 | LINEAR PROTOCOL OF DIRECTED GRAPH

In this section, it is shown the significance of the interplay between agent inertia and information topology. In undirected communication graphs, heterogeneous inertia does not affect double-integrator multi-agent systems, but given a certain communication digraph, there exist unstable group behaviors. Following that, a distributed protocol for agents with inertial roles and development on a balanced information graph is shown.

3.1 | The effect of inertia

The agent-i's closed-loop dynamics with its inertia, $m_i > 0$, are shown below:

$$m_i \ddot{x}_i = \sum_{j\in F_i} -br_{ij}\left(\dot{x}_i - \dot{x}_j\right) - kr_{ij}\left(x_i - x_j\right) \tag{5}$$

in this equation, k and b are stiffness gains and the damping, respectively. As follows is the closed-loop group kinematics by stacking up the individual dynamics (5):

$$M\ddot{x} + bL\dot{x} + kLx = 0 \tag{6}$$

On the following cyclic graph, Figure 1, the dynamics-based flocking model (6) is applied to the four agents, where $\left(m_i, w_{ij}, b, k\right) = (1, 1, 1, 2.2)$, $M = \text{diag}\left[m_1, m_2, \ldots, m_n\right]$. As seen in Figure 2, the group behavior is unstable.

Remark 1. The dynamics (6) resembles a standard mass-spring-damper system if G is undirected. The system is stable because it has asymmetric and positive-semidefinite L.[46,47]

A clear illustration of the significance of the interaction between the inertias of the agents and the information structure can be seen through this example, and a framework that caters to agents with the inertia that is not negligible and evolves with broad information digraphs is needed.

3.2 | The inertias-independent protocol

In this section, a protocol are developed for double-integrator multi-agent systems. Generally, control gains are larger as the inertia is larger, as indicated in earlier articles, which should be avoided in control systems with high frequency noise. More research is required for inertia-independent distributed observer methods. This linear protocol for solving the consensus problem of multi-agent systems specified by (1) and (2) with any beginning state by using the signal (3).

Based partly on Zhang et al.,[43] we propose the following linear protocol, which relies on a distributed observer:

$$\dot{\varepsilon}_i(t) = \frac{1}{\sum_{j=1}^{N+1} r_{ij}} \left(\sum_{j=1}^{N+1} r_{ij}\dot{\varepsilon}_j(t) - \eta \sum_{j=1}^{N+1} r_{ij}\left(\varepsilon_i(t) - \varepsilon_j(t)\right) \right), i = 1, 2, \ldots, N \tag{7}$$

where $\varepsilon_{N+1}(t) = x_{N+1}(t)$ and $\eta > 0$ is a design constant. We explore the following linear procedure utilizing the estimated values:

$$u_i(t) = K\left(x_i(t) - \varepsilon_i(t)\right), i = 1, 2, \ldots, N \tag{8}$$

where $K = \left[-k_1, -k_2\right], k_1 > 0, k_2 > 0$.

Theorem 1. Imagine that there are n followers with dynamics (1), and m leaders with dynamics (2) in a multi-agent system. All followers will still converge on the leader under protocol (8).

Proof. (7) can be rewritten using a simple change:

$$\sum_{j=1}^{N+1} r_{ij}\dot{\varepsilon}_i(t) - \sum_{j=1}^{N+1} r_{ij}\dot{\varepsilon}_j(t) = -\eta \sum_{j=1}^{N+1} r_{ij}\left(\varepsilon_i(t) - \varepsilon_j(t)\right) \tag{9}$$

whose compact form is:

$$\begin{bmatrix} \sum_{j=1}^{N+1} r_{1j} & -r_{12} & \cdots & -r_{1N} \\ -r_{21} & \sum_{j=1}^{N+1} r_{2j} & \cdots & -r_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ -r_{N1} & \cdots & -r_{N(N-1)} & \sum_{j=1}^{N+1} r_{Nj} \end{bmatrix} \times \begin{bmatrix} \dot{\varepsilon}_1(t) \\ \dot{\varepsilon}_2(t) \\ \vdots \\ \dot{\varepsilon}_N(t) \end{bmatrix} = -\eta \begin{bmatrix} \sum_{j=1}^{N+1} r_{1j}\left(\varepsilon_1(t) - \varepsilon_j(t)\right) \\ \sum_{j=1}^{N+1} r_{2j}\left(\varepsilon_2(t) - \varepsilon_j(t)\right) \\ \vdots \\ \sum_{j=1}^{N+1} r_{Nj}\left(\varepsilon_N(t) - \varepsilon_j(t)\right) \end{bmatrix} \tag{10}$$

where

$$\begin{bmatrix} \sum_{j=1}^{N+1} r_{1j} & -r_{12} & \cdots & -r_{1N} \\ -r_{21} & \sum_{j=1}^{N+1} r_{2j} & \cdots & -r_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ -r_{N1} & \cdots & -r_{N(N-1)} & \sum_{j=1}^{N+1} r_{Nj} \end{bmatrix} = L_1$$

denote $v_i(t) = \sum_{j=1}^{N+1} r_{ij}\left(\varepsilon_i(t) - \varepsilon_j(t)\right)$, then (10) can be modified as follows:

$$\dot{v}_i(t) = -\eta v_i(t) \tag{11}$$

denote:

$$v = \left[v_1^{\mathrm{T}}, v_2^{\mathrm{T}}, \ldots, v_N^{\mathrm{T}}\right]^{\mathrm{T}} \tag{12}$$

as a result of (11):

$$\dot{v}(t) = -\eta v(t) \tag{13}$$

$$v(t) = \left(L_1 \otimes I_n\right)\phi(t) \tag{14}$$

where:

$$\phi(t) = \begin{bmatrix} \varepsilon_1(t) - \varepsilon_{N+1}(t) \\ \varepsilon_2(t) - \varepsilon_{N+1}(t) \\ \cdots \\ \varepsilon_n(t) - \varepsilon_{N+1}(t) \end{bmatrix} = \begin{bmatrix} \varepsilon_1(t) - x_{N+1}(t) \\ \varepsilon_2(t) - x_{N+1}(t) \\ \cdots \\ \varepsilon_n(t) - x_{N+1}(t) \end{bmatrix} = \varepsilon(t) - I_N \otimes x_{N+1}(t) \tag{15}$$

with $\varepsilon = \left[\varepsilon_1^T, \varepsilon_2^T, \ldots, \varepsilon_N^T\right]^{\mathrm{T}}$. Based on Lemma 1, we have from (14) that:

$$\phi(t) = \left(L_1^{-1} \otimes I_n\right)v(t) \tag{16}$$

Its time derivative is $\dot{\phi}(t) = -\eta\phi(t)$ as defined by $e(t), u(t), \varepsilon(t), \phi(t)$ and $v(t)$. The closed-loop system composed of (1),(2),(7), and (8) may be expressed as:

$$\begin{cases} \dot{\phi}(t) = -\eta\phi(t) \\ \dot{e}(t) = \left(I_N \otimes A\right)e(t) + \left(M^{-1} \otimes B\right)\sigma(u) \\ u(t) = \left(I_N \otimes K\right)\left(e(t) - \phi(t)\right) \end{cases} \tag{17}$$

we have that $\eta > 0$, which implies that $\gamma = \eta * I_N$ is a Hurwitz matrix. Denote $\delta > 0$. In all cases, the matrix P_0 is positive definite in which:

$$T_1 \triangleq \gamma^{\mathrm{T}} P_0 + P_0 \gamma \leq -\delta I_{2N} \tag{18}$$

if $\lim_{x\to\infty} \|e(t)\|_2 = 0$ and $\lim_{x\to\infty} \|\phi(t)\|_2 = 0$, we have solved the consensus problem.

Take into account the following Lyapunov-like function:

$$V(e,\phi) = e^{\mathrm{T}}(M \otimes P)e + 2\sum_{i=1}^{N} \int_0^{u_i} \sigma(s)\mathrm{d}s + \phi^{\mathrm{T}} P_0 \phi \tag{19}$$

The Lyapunov-like (19) is easily proved to be positive definite. Then:

$$\begin{aligned}
\dot{V}(e,\phi) &= \dot{e}^{\mathrm{T}}(M \otimes P)e + e^{\mathrm{T}}(M \otimes P)\dot{e} + 2\sum_{i=1}^{N} \sigma\left(u_i\right)\dot{u}_i + \dot{\phi}^{T} P_0 \phi + \phi^{\mathrm{T}} P_0 \dot{\phi} \\
&= e^{\mathrm{T}}\left(M \otimes \left(A^{\mathrm{T}} P + PA\right)\right) e + 2\dot{u}^{\mathrm{T}}\sigma(u) + 2\sigma^{\mathrm{T}}(u)\left(I_N \otimes B^{\mathrm{T}} P\right) e + \phi^{\mathrm{T}} T_1 \phi \\
&= 2\sigma^{\mathrm{T}}(u)\left(I_N \otimes B^{\mathrm{T}} P\right) e + 2\sigma^{\mathrm{T}}(u)\left(I_N \otimes KA\right) e + 2\sigma^{\mathrm{T}}(u)\left(M^{-1} \otimes KB\right)\sigma(u) \\
&+ 2\sigma^{\mathrm{T}}(u)\left(I_N \otimes K\right)\gamma\phi + \phi^{\mathrm{T}} T_1 \phi \\
&= 2\sigma^{\mathrm{T}}(u)\left(I_N \otimes \left(B^{\mathrm{T}} P + KA\right)\right) e + \sigma^{\mathrm{T}}(u)\left(M^{-1} \otimes \left(B^{\mathrm{T}} K^{\mathrm{T}} + KB\right)\right)\sigma(u) \\
&+ 2\sigma^{\mathrm{T}}(u)\left(I_N \otimes K\right)\gamma\phi + \phi^{\mathrm{T}} T_1 \phi
\end{aligned} \tag{20}$$

Because of $A^{\mathrm{T}} P + PA = 0$, the third equation holds. There is a constant $\pi > 0$ to be designed. Using Young's inequality, we have calculated:

$$2\sigma^{\mathrm{T}}(u)\left(I_N \otimes K\right)\gamma\phi \leq \pi\sigma^{\mathrm{T}}(u)\left(M^{-1} \otimes I_2\right)\sigma(u) + \frac{1}{\pi}\phi^{\mathrm{T}}\gamma^{\mathrm{T}}\left(M \otimes K^{\mathrm{T}}K\right)\gamma\phi$$

If (20) is substituted for the inequality above, it gives:

$$\begin{aligned}
\dot{V}(e,\theta) &\leq 2\sigma^{\mathrm{T}}(u)\left(I_N \otimes \left(B^{\mathrm{T}} P + KA\right)\right) e + \sigma^{\mathrm{T}}(u)\left(M^{-1} \otimes \left(B^{\mathrm{T}} K^{\mathrm{T}} + KB\right)\right)\sigma(u) \\
&+ \pi\sigma^{\mathrm{T}}(u)\left(M^{-1} \otimes I_2\right)\sigma(u) + \frac{1}{\pi}\phi^{\mathrm{T}}\gamma^{\mathrm{T}}\left(M \otimes K^{\mathrm{T}}K\right)\gamma\phi + \phi^{\mathrm{T}} T_1 \phi \\
&\leq 2\sigma^{\mathrm{T}}(u)\left(I_N \otimes \left(B^{\mathrm{T}} P + KA\right)\right) e + \sigma^{\mathrm{T}}(u)\left(M^{-1} \otimes \left(B^{\mathrm{T}} K^{\mathrm{T}} + KB\right)\right)\sigma(u) \\
&+ \pi\sigma^{\mathrm{T}}(u)\left(M^{-1} \otimes I_2\right)\sigma(u) + \frac{1}{\pi}\phi^{\mathrm{T}}\gamma^{\mathrm{T}}\left(M \otimes K^{\mathrm{T}}K\right)\gamma\phi - \delta\phi^{\mathrm{T}}\phi \\
&= \sigma^{\mathrm{T}}(u)\left(M^{-1} \otimes \left(B^{\mathrm{T}} K^{\mathrm{T}} + KB + \pi\right)\right)\sigma(u) - \phi^{\mathrm{T}}\left(\delta I_{2N} - \frac{1}{\pi}\gamma^{\mathrm{T}}\left(M \otimes K^{\mathrm{T}}K\right)\gamma\right)\phi
\end{aligned} \tag{21}$$

due to $B^{\mathrm{T}} P + FA = 0$, the third equation holds, we can choose $\pi < 2k_2$ such that:

$$-T_2 \triangleq B^{\mathrm{T}} K^{\mathrm{T}} + KB + \pi < 0 \tag{22}$$

choosing δ such that $\delta > \frac{1}{\pi}\beta\left(\gamma^{\mathrm{T}}\left(M \otimes K^{\mathrm{T}}K\right)\gamma\right)$ is sufficiently large, we have that:

$$T_3 \triangleq \delta I_{2N} - \frac{1}{\pi}\gamma^{\mathrm{T}}\left(M \otimes K^{\mathrm{T}}K\right)\gamma > 0 \tag{23}$$

the inequality in (21) may be extended using (22) and (23) as:

$$\begin{aligned}
\dot{V}(e,\phi) &\leq -\sigma^{\mathrm{T}}(u)\left(M^{-1} \otimes T_2\right)\sigma(u) - \phi^{\mathrm{T}} T_3 \phi \\
&\leq -\alpha\left(M^{-1}\right)\alpha\left(T_2\right)\sigma^{\mathrm{T}}(u)\sigma(u) - \alpha\left(T_3\right)\phi^{\mathrm{T}}\phi
\end{aligned} \tag{24}$$

The states of the closed-loop system (17) converge to the set $\mathcal{E}_1 = \left\{[e,\phi] : \left(I_N \otimes K\right)(e+\phi) = 0, \phi = 0\right\}$ according to LaSalle's invariant principle. in such a case, the closed-loop system (17) in the set $\mathcal{E}_1$ becomes:

$$\begin{cases} \phi(t) = 0 \\ \dot{e}(t) = \left(I_N \otimes A\right) e(t) \end{cases} \tag{25}$$

the matrix pair (A, K) is observable, and so is the matrix pair $\left(I_N \otimes A, I_N \otimes K\right)$. As a result, the elements in the set $\mathcal{E}_1$ are $\phi(t) = 0, e(t) = 0$. Proof has been completed. □

Remark 2. The advantage of this protocol over others is that the observer (7) is fully independent of inertia and hence free to design. However, for the choice of K in protocol (8), the traditional optimal control approach requires the complete dynamics model, including precise inertia. In reality, however, obtaining the correct inertia may be difficult. There is uncertainty in unidentified inertia.

$$\dot{x}_i(t) = Ax_i(t) + \frac{1}{\left(m_i + \Delta m_i\right)} B\sigma\left(u_i(t)\right), i = 1, 2, \ldots, N \tag{26}$$

Due to the inertia not being accurately calculated, Δm_i presents the uncertainties of its inertia. Because of the presence of inertia uncertainty, designing the protocol's control law is problematic. Despite the fact that the protocol is inertia-independent, inertia must be measured while developing the control law. This demonstrates the significance of a data-driven approach. The online ADP for solving the control law is presented in the next section.

4 | ADAPTIVE DYNAMIC PROGRAMMING

In this section, when the system dynamics are unknown, a policy iterative approach is provided to approximate the algebraic Riccati problem's solution. The policy iterative approach is paired with online ADP algorithm. The developed method can approximate the control gain K^* for each follower without relying on system matrix knowledge or the precise inertia, by utilizing all the finite data available, imposing an initial control policy on the agent at a limited time interval, collecting online measurements, and iterating by reusing the same online data.

4.1 | Online off-policy algorithm

The continuous-time linear system (1), in which the system dynamics matrix A and the input matrix B are unknown constant matrices of acceptable dimensions, and m_i are unknown constants.

denote:

$$\frac{1}{m_i} B = B_{m_i} \tag{27}$$

(1) can be simplified as follow:

$$\dot{x}_i = Ax + B_{m_i} u(t) \tag{28}$$

furthermore, (28) is regarded as stable in that there exists a constant matrix K with adequate dimensions such that $A - B_{m_i} K$ is Hurwitz

We are looking for a linear quadratic regulator (LQR):

$$u = -Kx \tag{29}$$

which reduces the performance index shown below to the minimum:

$$J\left(x_0; u\right) = \int_0^\infty \left(x^T Q x + u^T R u\right) dt \tag{30}$$

where $Q = Q^T \geq 0, R = R^T > 0$ with $\left(A, Q^{1/2}\right)$ observable, taking (29) and applying it to (28), we can easily write (30) as:

$$J\left(x_0; u\right) = x_0^T P x_0 \tag{31}$$

where:

$$P = \int_0^\infty e^{\left(A - B_{m_i} K\right)^T t} \left(x^T Q x + K^T R K\right) e^{\left(A - B_{m_i} K\right) t} dt \tag{32}$$

The Lyapunov equation has just one positive definite solution P , when the derivative of $X^T P X$ is taken along the solution of (28):

$$\left(A - B_{m_i} K\right)^T P + P\left(A - B_{m_i} K\right) + Q + K^T R K = 0 \tag{33}$$

In classical optimal control theory, the solution to an optimal control issue can be found by solving the following Riccati equation when A and B are known exactly:

$$A^T P + PA + Q - PB_{m_i} R^{-1} B_{m_i}^T P = 0 \tag{34}$$

as a consequence, it is possible to compute the ideal state feedback gain matrix K^* in (29) by using

$$K^* = R^{-1} B_{m_i}^T P^* \tag{35}$$

As a result, solving (34) is often challenging, especially for high-dimensional matrices. The answer to (34) has nevertheless been numerically approximated by numerous efficient approaches. Kleinman's algorithm is one such approach,[48] and it is discussed further below.

Theorem 2. $K_0 \in \mathbb{R}^{m\times n}$ is selected in such a way that the matrix $A - B_{m_i} K_0$ is Hurwitz. and continue by repeating the following steps for $k = 0, 1, ...$

(1) For a real symmetric positive definite solution P_k , solve the Lyapunov equation:

$$A_k^T P_k + P_k A_k + Q + K_k^T R K_k = 0 \tag{36}$$

where:$A_k = A - B_{m_i} K_k$

(2) The feedback gain matrix should be updated by :

$$K_{k+1} = R^{-1} B_{m_i}^T P_k \tag{37}$$

Then, the following properties hold:

(1) $A - B_{m_i} K_k$ is Hurwitz;

(2) $P^* \le P_{k+1} \le P_k$;

(3) $\lim_{k\to\infty} K_k = K^*$, $\lim_{k\to\infty} P_k = P^*$

Proof. : Consider the Lyapunov equation (36) with $k = 0$. Since $A - BK_0$ is Hurwitz, by (32) we know P_0 is finite and positive definite. In addition, by (32) and (36) we have

$$P_0 - P_1 = \int_0^\infty e^{A_1^T \tau} \left(K_0 - K_1\right)^T R \left(K_0 - K_1\right) e^{A_1 \tau} d\tau \ge 0$$

Similarly, by (32) and (34) we obtain

$$P_1 - P^* = \int_0^\infty e^{A_1^T \tau} \left(K_1 - K^*\right)^T R \left(K_1 - K^*\right) e^{A_1 \tau} d\tau \ge 0$$

Therefore, we have $P^* \le P_1 \le P_0$. Since P^* is positive definite and P_0 is finite, P_1 must be finite and positive definite. This implies that $A - BK_1$ is Hurwitz. Repeating the above analysis for $k = 1, 2, \ldots$ proves Properties (1) and (2) in Theorem 2. Finally, since $\{P_k\}$ is a monotonically decreasing sequence and lower bounded by P^*, $\lim_{k\to\infty} P_k = P_\infty$ exists. By (36) and (37), $P = P_\infty$ satisfies (34), which has a unique solution. Therefore, $P_\infty = P^*$. The proof is thus complete. □

Remark 3. The Lyapunov equation (36) is linear in P_k , when A and B are known, it is possible to solve P_k by (36) and update K_k by (37) iteratively. So, it is numerically approximated to find a solution for equation (34).

Then, based on Theorem 2, we will provide an offline policy iteration algorithm to solve the optimal control issue.

Algorithm 1 The offline policy iteration algorithm

Step 1: Considering a stabilizing feedback gain matrix K_0;

Step 2: Solve P_K from Equation (36);

Step 3: Update the feedback gain K_{K+1} matrix by (37);

Step 4: Let $K = K + 1$, if $\left|P_k - P_{k-1}\right| \le \epsilon$ for $k \ge 0$, go to Step 5; else go to Step 2. Where ϵ is a small positive number;

Step 5: use $u = -K_k x$ as the approximate optimal control policy.

Based on (37) and (36), it is evident that the algorithm is offline and needs exact knowledge of the dynamic. But, the reality is that it is often hard to build a model of system dynamics or to obtain precise information regarding the dynamics of systems. To address this issue without the need for prior knowledge of system dynamics, an online ADP method is developed in the spirit of Jiang et al.[49]

4.2 | Adaptive dynamic programming based on policy iteration

In this section, we will describe an online ADP algorithm without the need for A and B based on the policy iteration algorithm shown above.

Consider K_0 to be a known stabilizing feedback gain matrix. System (28) is rewritten in the following form:

$$\dot{x}_i = Ax + B_{m_i} u_0 \tag{38}$$

then, along with the solutions of (38) by (36) and (37), one can obtain:

$$\begin{aligned} &x^T(t+\Delta t)P_k x(t+\Delta t) - x^T(t)P_k x(t) \\ &= -\int_t^{t+\Delta t} x^T Q_k x d\tau + 2\int_t^{t+\Delta t} \left(u_0 + K_k x\right)^T R K_{k+1} x d\tau \end{aligned} \tag{39}$$

where $Q_k = Q + K_k^T R K_k$

Remark 4. In Equation (39), it is worth noting that the system matrices can be replaced with the states and inputs measured online. As a result, Equation (39) can be used to obtain P_k and K_{k+1} without knowing A and B exactly.

Thus, we introduce the Kronecker product in order to find $\left(P_k, K_{k+1}\right)$ with the unknown system matrices under a given stabilizing feedback gain matrix K_k :

$$x^T Q_k x = \left(x^T \otimes x^T\right) \operatorname{vec}\left(Q_k\right) \tag{40}$$

$$\left(u + K_k x\right)^T R K_{k+1} x = \left[\left(x^T \otimes x^T\right)\left(I_n \otimes K_k^T R\right) + \left(x^T \otimes u_0^T\right)\left(I_n \otimes R\right)\right] K_{k+1} \tag{41}$$

denote $\zeta_{xx} \in \mathbb{R}^{1\times n^2}$, $\varphi_{xx} \in \mathbb{R}^{1\times n^2}$ and $\varphi_{xu} \in \mathbb{R}^{1\times mn}$

$$\zeta_{xx} = \left[x \otimes x|_{t_1}^{t_1+\delta t}, \quad x \otimes x|_{t_2}^{t_2+\delta t}, \quad \ldots, \quad x \otimes x|_{t_l}^{t_l+\delta t}\right]^T \tag{42}$$

$$\varphi_{xx} = \left[\int_{t_1}^{t_1+\delta t} x \otimes x d\tau, \quad \int_{t_2}^{t_2+\delta t} x \otimes x d\tau, \quad \ldots, \quad \int_{t_l}^{t_l+\delta t} x \otimes x d\tau\right]^T \tag{43}$$

$$\varphi_{xu} = \left[\int_{t_1}^{t_1+\delta t} x \otimes u_0 d\tau, \quad \int_{t_2}^{t_2+\delta t} x \otimes u_0 d\tau, \quad \ldots, \quad \int_{t_l}^{t_l+\delta t} x \otimes u_0 d\tau\right]^T \tag{44}$$

where $0 \le t_1 \le t_2 \le \cdots \le t_l$ then $\Phi_k \in \mathbb{R}^{l\times\left(n^2+mn\right)}$ and $\Psi_k \in \mathbb{R}^l$

$$\Phi_k = \left[\zeta_{xx}, -2\varphi_{xx}\left(I_n \otimes K_k^T R\right) - 2\varphi_{xu}\left(I_n \otimes R\right)\right] \tag{45}$$

$$\Psi_k = -\varphi_{xx} \operatorname{vec}\left(Q_k\right) \tag{46}$$

this is the form of Equation (39) when written as a linear equation:

$$\Phi_k \begin{bmatrix} \operatorname{vec}\left(P_k\right) \\ \operatorname{vec}\left(K_{k+1}\right) \end{bmatrix} = \Psi_k \tag{47}$$

Lemma 2. [49]It is possible to have a sufficiently large integer $l > 0$ such that:

$$\operatorname{rank}\left(\left[\varphi_{xx}, \varphi_{xu}\right]\right) = \frac{n(n+1)}{2} + mn$$

Lemma 3. [17]Whenever k_0 is an initial stabilizing feedback control gain, and Lemma 2 holds, the sequences and obtained by solving (49) will respectively converge to optimal $\left\{P_k\right\}_{k=0}^{\infty}$ and $\left\{k_k\right\}_{k=0}^{\infty}$ .obtained by solving (49) will respectively converge to optimal P^* and K^* .

As a result, we are able to implement an online ADP method for solving the optimal control issue under uncertain system dynamics.

Algorithm 2 Off-policy ADP algorithm

Step 1: Find K_0 such that $A - BK_0$ is Hurwitz. Let$k = 0$;
Step 2: Utilize $u_0 = -K_0 + e$ as the control input, e is the exploration noise. Compute , ζ_{xx}, φ_{xx} and φ_{xu} to satisfy the rank condition in Lemma 2;
Step 3: Solve for P_k and K_{k+1} from (49);
Step 4: Let $K = K + 1$and repeat Step (3), until $\left|P_k - P_{k-1}\right| \leq \epsilon$, where the constant $\varepsilon > 0$ is a predefined small threshold;
Step 5: use $u = -K_k x$ as the approximate optimal control policy.

5 | SIMULATIONS

In this section, we use a numerical example to demonstrate the efficiency of our method and develop a data-driven distributed controller of followers using ADP.

Consider the following four-node digraph structure with node 1 connected to the leader node, such as Figure 3. The dynamic of the $i-th$ follower is described by (1) with $m_1 = 2.5, m_2 = 0.8, m_3 = 0.8, m_4 = 1.25, A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, and the dynamic of the leader is described by (2). The system matrix and inertia are for the analysis of the results. Agents' initial conditions are selected at random as $x_1(0) = [-30, 15]^{\mathrm{T}}$, $x_2(0) = [-1, 15]^{\mathrm{T}}$, $x_3(0) = [30, -8]^{\mathrm{T}}$, $x_4(0) = [40, -15]^{\mathrm{T}}$, $x_5(0) = [-10, 3]^{\mathrm{T}}$.

Each follower's input limit is randomly chosen as $\left|u_1(t)\right| \leq 4$, $\left|u_2(t)\right| \leq 2$,$\left|u_3(t)\right| \leq 1$, $\left|u_4(t)\right| \leq 5$. The protocol is designed by (8) and the observer gain is determined at random as $\eta = 0.2$. The selection of Q and R in the performance index function impacts the speed of state convergence and the size of the input. To simplify the simulation, we design Q and R for each follower as: $\mathrm{Q}_1 = \mathrm{Q}_2 = \mathrm{Q}_4 = \mathrm{diag}\left(\begin{bmatrix} 100 & 100 \end{bmatrix}\right)$, $\mathrm{Q}_3 = \mathrm{diag}\left(\begin{bmatrix} 40 & 40 \end{bmatrix}\right)$, $\mathrm{Q}_1 = \mathrm{Q}_2 = \mathrm{Q}_3 = \mathrm{Q}_4 = \mathrm{diag}([1])$. It is chosen to be $e = 100 \sum_{i=1}^{100} \sin\left(\omega_i t\right), i = 1, 2, \ldots, 100$ as the exploration noise, with ω_i, with $k = 1, \ldots, 100$, random numbers uniformly distributed on $[-50, 50]$. During each interval of 0.1s, input and state information is collected. At t=1 s , the policy iteration started, and convergence was achieved when the stopping criterion $\left|P_k - P_{k-1}\right| \leq 0.0005$ is satisfied. From t=1 s up until the end of the simulation, learned controllers are actually used as control inputs in the system. In Figure 4 - 7, we see that $\left\{P_k\right\}$ and $\left\{K_k\right\}$ have reached their optimal values.

As follows are the optimal values for P^* and K^* :

$$P_1^* = P_1 = \begin{bmatrix} 122.4745 & 25 \\ 25 & 30.6186 \end{bmatrix}, \quad K_1^* = K_1 = \begin{bmatrix} 10 & 12.2474 \end{bmatrix}$$

$$P_2^* = P_2 = \begin{bmatrix} 107.7033 & 8 \\ 8 & 8.6163 \end{bmatrix}, \quad K_2^* = K_2 = \begin{bmatrix} 10 & 10.7703 \end{bmatrix}$$

$$P_3^* = P_3 = \begin{bmatrix} 44.7747 & 5.0596 \\ 5.0596 & 5.6636 \end{bmatrix}, \quad K_3^* = K_3 = \begin{bmatrix} 6.3246 & 7.0795 \end{bmatrix}$$

$$P_4^* = P_4 = \begin{bmatrix} 111.8034 & 12.5 \\ 12.5 & 13.9754 \end{bmatrix}, \quad K_4^* = K_4 = \begin{bmatrix} 10 & 11.1803 \end{bmatrix}$$

As can be seen in Figure 8 and Figure 9, the state trajectories of the system are shown.

Simulation experiments show that our proposed algorithm can converge to the optimal control law without agents' dynamics. In summary, the efficiency of our proposed online model-free ADP method is confirmed by simulation results.

6 | CONCLUSIONS

In this paper, the data-driven ADP approach is applied to solve the leader-following consensus problem of double-integrator multi-agent systems with completely unknown dynamics. We solved the problem that inertia is difficult to measure, making it hard to design controller gain. It is noteworthy that, in the achievement of getting consensus in double-integrator multi-agent systems, the developed method only needs current and past data instead of accurate system models. Further,we design a new controller for each follower that can achieve consensus. As a result of ADP's ability to scale, our findings are potentially applicable to other higher dimensional systems.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China under Grant U2141231.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

1. Chu T, Wang J, Codeca L, Li Z. Multiagent deep reinforcement learning for largescale traffic signal control. *IEEE trans Intell Transp Syst* 2020; 21(3): 1086-1095. doi: 10.1109/tits.2019.2901791

2. Liu H, Meng Q, Peng F, Lewis FL. Heterogeneous formation control of multiple UAVs with limited-input leader via reinforcement learning. *Neurocomputing* 2020; 412: 63-71. doi: 10.1016/j.neucom.2020.06.040

3. Zhang X, Cheng Z, Ma J, Huang S, Lewis FL, Lee TH. Semi-definite relaxation-based ADMM for cooperative planning and control of connected autonomous vehicles. *IEEE trans Intell Transp Syst* 2021. doi: 10.1109/tits.2021.3094215

4. Jin X, Wang S, Qin J, Zheng WX, Kang Y. Adaptive fault-tolerant consensus for a class of uncertain nonlinear second-order multi-agent systems with circuit implementation. *IEEE Trans Circuits Syst I Regul Pap* 2017; 65(7): 2243–2255.

5. Jin X, Lü S, Yu J. Adaptive NN-based consensus for a class of nonlinear multiagent systems with actuator faults and faulty networks. *IEEE Trans Neural Netw Learn Syst* 2021.

6. Dong X, Zhou Y, Ren Z, Zhong Y. Time-varying formation tracking for second-order multi-agent systems subjected to switching topologies with application to quadrotor formation flying. *IEEE Trans Ind Electron* 2017; 64(6): 5014-5024. doi: 10.1109/tie.2016.2593656

7. Feng Y, Duan Z, Lv Y, Ren W. Some necessary and sufficient conditions for synchronization of second-order interconnected networks. *IEEE Trans Cybern* 2019; 49(12): 4379-4387. doi: 10.1109/tcyb.2018.2864625

8. Zhang W, Zuo Z, Wang Y, Zhang Z. Double-integrator dynamics for multiagent systems with antagonistic reciprocity. *IEEE Trans Cybern* 2020; 50(9): 4110-4120. doi: 10.1109/tcyb.2019.2939487

9. Zhou B. Consensus of delayed multi-agent systems by reduced-order observer-based truncated predictor feedback protocols. *IET Control Theory Appl* 2014; 8(16): 1741-1751. doi: 10.1049/iet-cta.2014.0038

10. Li Z, Ren W, Liu X, Fu M. Consensus of multi-agent systems with general linear and lipschitz nonlinear dynamics using distributed adaptive protocols. *IEEE Trans Automat Contr* 2013; 58(7): 1786-1791. doi: 10.1109/tac.2012.2235715

11. Li Z, Wen G, Duan Z, Ren W. Designing fully distributed consensus protocols for linear multi-agent systems with directed graphs. *IEEE Trans Automat Contr* 2015; 60(4): 1152-1157. doi: 10.1109/tac.2014.2350391

12. Olfati-Saber R, Murray RM. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans Automat Contr* 2004; 49(9): 1520-1533.

13. Jadbabaie A, Jie L, Morse AS. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans Automat Contr* 2003; 48(6): 988-1001.

14. Lin Z, Broucke M, Francis B. Local control strategies for groups of mobile autonomous agents. *IEEE Trans Automat Contr* 2004; 49(4): 622-629.

15. Hughes PC. spacecraft attitude dynamics. *J. Wiley* 1986.

16. Gustafsson , Thomas . On the design and implementation of a rotary crane controller. *Eur J Control* 2016; 2(3): 166-175.

17. Jing WX, Mcinnes CR. Memorised quasi-time-fuel-optimal feedback control of perturbed double integrator. *Automatica* 2002; 38(8): 1389-1396.

18. Zhao L, Jia Y. Decentralized adaptive attitude synchronization control for spacecraft formation using nonsingular fast terminal sliding mode. *Nonlinear Dyn* 2014; 78(4): 2779-2794. doi: 10.1007/s11071-014-1625-5

19. Feng X, Butler-Purry KL, Zourntos T. A multi-agent system framework for real-time electric Load management in MVAC all-electric ship power systems. *IEEE Trans Power Syst* 2015; 30(3): 1327-1336. doi: 10.1109/tpwrs.2014.2340393

20. Dongjun , Lee , Spong , Mark , W . Stable flocking of multiple inertial agents on balanced graphs. *IEEE Trans Automat Contr* 2007; 52(8): 1469-1475.

21. Huang YJ, Kuo TC, Chang SH. Adaptive sliding-mode control for nonlinears ystems with uncertain parameters. *IEEE Trans Syst Man Cybern* 2008; 38(2): 534-9.

22. Li XJ, Yang GH. Robust adaptive fault-tolerant control for uncertain linear systems with actuator failures. *IET Control Theory Appl* 2012; 6(10): 1544-1551.

23. Si J, Wang YT. Online learning control by association and reinforcement. *IEEE Trans Neural Netw* 2001; 12(2): 264-76. doi: 10.1109/72.914523

24. Liu F, Sun J, Si J, Guo W, Mei S. A boundedness result for the direct heuristic dynamic programming. *Neural Networks* 2012; 32: 229-235. doi: 10.1016/j.neunet.2012.02.005

25. Sokolov Y, Kozma R, Werbos LD, Werbos PJ. Complete stability analysis of a heuristic approximate dynamic programming control design. *Automatica* 2015; 59: 9-18. doi: 10.1016/j.automatica.2015.06.001

26. Modares H, Lewis FL, Naghibi-Sistani MB. Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems. *Automatica* 2014; 50(1): 193-202. doi: 10.1016/j.automatica.2013.09.043

27. Peng Z, Zhao Y, Hu J, Luo R, Ghosh BK, Nguang SK. Input–Output Data-Based Output Antisynchronization Control of Multiagent Systems Using Reinforcement Learning Approach. *IEEE Trans Industr Inform* 2021; 17(11): 7359–7367.

28. Peng Z, Luo R, Hu J, Shi K, Nguang SK, Ghosh BK. Optimal tracking control of nonlinear multiagent systems using internal reinforce Q-learning. *IEEE Trans Neural Netw Learn Syst* 2021.

29. Werbos PJ, Miller W, Sutton R. *A menu of designs for reinforcement learning over time*. 3. : 67-95; MIT press Cambridge, MA . 1990.

30. Werbos P. Advanced forecasting methods for global crisis warning and models of intelligence. *General System Yearbook* 1977: 25-38.

31. Prokhorov DV, Wunsch D. Adaptive critic designs. *IEEE Trans Neural Netw* 1997; 8(5): 997-1007.

32. Molina D, Venayagamoorthy GK, Liang J, Harley RG. Intelligent local area signals based damping of power system oscillations using virtual generators and approximate dynamic programming. *IEEE Trans Smart Grid* 2013; 4(1): 498-508.

33. Bertsekas DP, Tsitsiklis JN. Neuro-dynamic programming: an overview. In: . 1. IEEE. ; 1995: 560–564.

34. Lewis FL, Vrabie D, Vamvoudakis KG. Reinforcement Learning and Feedback Control using natural decision methods to design optimal adaptive controllers. *IEEE Control Syst Mag* 2012; 32(6): 76-105. doi: 10.1109/mcs.2012.2214134

35. Al-Tamimi A, Abu-Khalaf M, Lewis FL. Adaptive critic designs for discrete-time zero-sum games with application to H_{infty} control. *IEEE Trans Syst Man Cybern.* 2007; 37(1): 240-247. doi: 10.1109/tsmcb.2006.880135

36. Murray JJ, Cox CJ, Lendaris GG, Saeks R. Adaptive dynamic programming. *IEEE Trans Syst Man Cybern* 2002; 32(2): 140-153.

37. Song R, Xiao W, Zhang H, Sun C. Adaptive dynamic programming for a class of complex-valued nonlinear systems. *IEEE Trans Neural Netw Learn Syst* 2014; 25(9): 1733-1739. doi: 10.1109/tnnls.2014.2306201

38. Liu D, Wei Q. Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. *IEEE Trans Neural Netw Learn Syst* 2014; 25(3): 621-634. doi: 10.1109/tnnls.2013.2281663

39. Zhang H, Feng T, Yang GH, Liang H. Distributed cooperative optimal control for multiagent systems on directed graphs: An inverse optimal approach. *IEEE Trans Cybern* 2014; 45(7): 1315–1326.

40. Vamvoudakis KG, Lewis FL, Hudas GR. Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality. *Automatica* 2012; 48(8): 1598–1611.

41. Zhang H, Zhang J, Yang GH, Luo Y. Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming. *IEEE Trans Fuzzy Syst* 2014; 23(1): 152–163.

42. Abouheaf MI, Lewis FL, Vamvoudakis KG, Haesaert S, Babuska R. Multi-agent discrete-time graphical games and reinforcement learning solutions. *Automatica* 2014; 50(12): 3038–3053.

43. Zhang K, Zhou B, Wen G. Global Leader-following consensus of double-integrator multi-agent systems by fully distributed bounded linear protocols. *IEEE Trans Automat Contr* 2022.

44. Subramanian RG, Elumalai VK. Robust MRAC augmented baseline LQR for tracking control of 2 DoF helicopter. *Rob Auton Syst* 2016; 86: 70-77.

45. Grip HF, Yang T, Saberi A, Stoorvogel AA. Output synchronization for heterogeneous networks of non-introspective agents. *Automatica* 2012; 48(10): 2444-2453.

46. Tanner HG, Jadbabaie A, Pappas GJ. Stable flocking of mobile agents, Part I: Fixed topology. In: . 2. IEEE. ; 2003: 2010–2015.

47. Tanner HG, Jadbabaie A, Pappas GJ. Stable flocking of mobile agents part I: dynamic topology. In: . 2. IEEE. ; 2003: 2016–2021.

48. Kleinman D. On an iterative technique for Riccati equation computations. *IEEE Trans Automat Contr* 1968; 13(1): 114-115.

49. Jiang Y, Jiang ZP. Global adaptive dynamic programming for continuous-time nonlinear systems. *IEEE Trans Automat Contr* 2015; 60(11): 2917-2929.

How to cite this article: Williams K., B. Hoskins, R. Lee, G. Masato, and T. Woollings (2016), A regime analysis of Atlantic winter jet variability applied to evaluate HadGEM3-GC2, *Q.J.R. Meteorol. Soc.*, *2017;00:1–6*.