An Imbalanced Mean-Field Game Theoretical Large-Scale Multiagent Optimization With Constraints

Shawon Dey and Hao Xu, *Member, IEEE*

***Abstract*—A novel optimization algorithm has been developed for distributed large-scale multiagent systems (LS-MASs), specifically focusing on achieving a terminal density constraint. While the recent advancement in mean field game (MFG) offers a feasible distributed solution to address the "Curse of Dimensionality" problem, it compromises the optimality of large-scale homogeneous agents and lacks the capability to achieve arbitrary fixed terminal probability density function (PDF) constraint, especially when deviating from the normal distribution. To tackle this issue, a novel approach called the imbalanced mean-field game (Imb-MFG) theory has been designed alongside an adaptive PDF decomposition method and distributed reinforcement learning (RL) that can effectively obtain optimal solutions in LS-MAS even with fixed terminal density constraints in a distributed manner. In particular, a method based on the induction theory has been developed for estimating the parameter of the final PDF constraint, enabling the decomposition of MFG-PDF into multiple imbalanced normal distributions. Subsequently, the Imb-MFG theory is developed by integrating the decomposed multigroup MFG agents with a K-means clustering algorithm with constraint. The developed Imb-MFG approach decomposes a single PDF into multiple imbalanced normal distributions, which are combined to achieve arbitrary terminal PDF constraints. To achieve the solution of the Imb-MFG theory, a multiactor–critic-mass (M-ACM) algorithm is developed. This algorithm is developed to concurrently learn the solution for coupled Fokker–Planck–Kolmogorov (FPK) and Hamilton–Jacobi–Bellman equations. The algorithm's convergence is ensured through the Lyapunov analysis. The effectiveness of this algorithm is validated through a simulation study.**

***Index Terms*—Large scale multi agent systems, mean field game, reinforcement learning.**

I. Introduction

IN RECENT times, a growing interest has emerged in the field of multiagent systems (MASs) [1]. The emerging attention is observed by both research communities and industrial sectors, particularly focusing on applications like traffic management [2], the use of autonomous unmanned-aerial-vehicle (UAV) [3], and so on. Besides that, rapid advances in control [4] and the game theory [5] have effectively facilitated the study of control strategies and decision making for MAS, grounded in solid mathematical principles [6]. However, while extending a conventional multiagent control system to a large-scale multiagent system (LS-MAS) through increasing the number of agents, two major challenges arise. First, to accomplish a shared objective, the data exchange among a massive number of agents in LS-MAS is required but challenging to sustain because of communication complexity. Second, the "Curse of Dimensionality" [7] poses a challenge due to the exponential growth in agent interactions when attempting to solve the PDE-based large-scale system optimal control. In order to overcome these two issues, prior research [7], [8] incorporated the mean field game (MFG) theory [9], [10] into their respective studies. In the MFG theory, a large number of neighboring agents has been considered as one united group represented as a probability density function (PDF). Recall to [10], by considering all the agents in LS-MAS are homogeneous and start from a known initial distribution, each agent can independently estimate all the other agents' behaviors (i.e., PDF) by solving a PDE known as the Fokker–Planck–Kolmogorov (FPK) equation. By employing this local PDF, each agent can efficiently access group information without additional communication and computational burdens. However, MFG restricts the system capabilities by assuming all the agents are homogeneous and maintain a unified PDF. For instance, while navigating LS-MAS in a complex environment with numerous unstructured obstacles, these mean-field agents may need to be divided into various groups with multiple probability distributions corresponding to each group. Through this division, the overall LS-MAS can successfully adapt to challenging environments with numerous unstructured obstacles. Moreover, the assumption of homogeneous dynamic in current MFG control [7] typically results in the LS-MAS PDF being Gaussian. This makes it challenging to enforce a non-Gaussian distribution, limiting the practical applicability of existing MFG methods. For example, considering LS-MAS pursuit and evasion games [11], a large number of pursuers might need to formulate a non-Gaussian PDF to maximize their probability of capturing massive evaders. Utilizing standard MFG that can only ensure LS-MAS single Gaussian PDF, is not a feasible approach for solving this type of problem.

To address these issues, an imbalanced mean-field game (Imb-MFG) theory framework is developed alongside an adaptive PDF decomposition method for LS-MAS of MFG

Received 26 April 2024; accepted 28 September 2024. Date of publication 5 November 2024; date of current version 18 December 2024. This work was supported in part by the National Science Foundation under Grant 2144646. This article was recommended by Associate Editor L. C. Rego. *(Corresponding author: Shawon Dey.)*

The authors are with the Department of Electrical and Biomedical Engineering, University of Nevada, Reno, NV 89557 USA (e-mail: sdey@unr.edu; haoxu@unr.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2024.3481351.

Digital Object Identifier 10.1109/TSMC.2024.3481351

interaction [8]. Then, the solution of the Imb-MFG has been learned by employing the reinforcement learning (RL) [12] technique. Specifically, a PDF decomposition approach with an induction-based method is designed to evaluate the parameters that effectively achieve imbalanced normal distributions by decomposing terminal density functions. This method helps agents to obtain multiple desired final normal distributions with imbalanced mean and variances for individual group LS-MAS. Then, a K-means algorithm with constraint [13] is incorporated to achieve the multigroup LS-MAS. Subsequently, the initial normal distributions of each group of MFG agents are generated and transferred to the respective agents. Then, the agents in each group ensure the convergence of the mixture PDF to the desired imbalanced distribution using the Imb-MFG theory for the LS-MAS optimal control problem. The Imb-MFG approach also considers the challenges associated with computational complexity and communication difficulties among a large number of agents. Next, obtaining the optimal solution for decomposed LS-MAS with Imb-MFG interaction involves solving two coupled PDEs: 1) the FPK equation and 2) the HJB equation. However, solving these coupled forward and backward PDEs is difficult to accomplish [8]. To address this problem, a novel multiactor–critic-mass (M-ACM) learning has been designed by incorporating RL [12] and adaptive dynamic programming (ADP) [14] techniques. In M-ACM, multiple neural networks (NNs) referred to as the *mass NNs* are responsible for learning the behaviors of massive populations in individual groups by FPK approximation. Additionally, multiple *critic NNs* are employed for cost function evaluation for agents in individual groups by learning the HJB equation solution. Finally, multiple *actor NNs* are employed to determine the optimal control strategy for agents. The key contributions are as follows.

1) A distributed optimal control with a fixed final density constraint for a large-scale system is formulated using a novel Imb-MFG theory.
2) A novel PDF parameter estimation based on the induction theory is developed to decompose the fixed final PDF.
3) An M-ACM learning algorithm has been designed to solve the Imb-MFG theory and achieve a distributed optimal solution for LS-MAS.
4) The Lyapunov stability analysis is provided to demonstrate the convergence of the NNs.

II. Problem Formulation

Consider an LS-MAS with the dynamic of each agent $\mathcal{A}$ is represented by

$$dx(t) = [f(x) + g(x)u]dt + \sigma d\omega \tag{1}$$

with the system state $x(t) \in \mathbb{R}^n$ and control input $u(t) \in \mathbb{R}^m$. Similar to [15], $f(0) = 0$ and $f(x) + g(x)u$ is assumed to be Lipschitz continuous. Also, $\sigma \in \mathbb{R}^{n\times n}$ denotes the Wiener process $\omega \in \mathbb{R}^n$ coefficient matrix. The final PDF $m_d(x;\theta)$ of the LS-MAS optimization is considered as the mixture of the N groups normal distributions [16] with distinct means and variances

$$m_d(x;\theta) = \sum_{j=1}^{N} w_j m_{d,j}\big(x;\theta_j\big) \quad j = 1,\ldots,N \tag{2}$$

with $m_{d,j}(x,\theta_j)$ denotes the jth group normal distribution with the parameter set $\theta_j = \{\mu_j, \Sigma_j\}$, and μ_j and Σ_j are the mean vector and the covariance matrix of the respective group. The parameter set involving weight is denoted as $\theta_{c,j} = \{w_j, \mu_j, \Sigma_j\}$. Next, the collection of the parameters in mixture-PDF is represented by the notation $\theta = \{\boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$. The cost function for individual agent $\mathcal{A}$ is defined as

$$J(x, m(x;\theta)) = \mathbb{E}\{\int_0^\infty [r(x,u) + \Phi(m(x;\theta))]dt\} \tag{3}$$

with the initial term derived as $r(x(t),u(t)) = \|x - \mathbb{E}\{m_{d,j}(x,\theta_j)\}\|_Q^2 + \|u\|_R^2$. Note that, $\mathbb{E}\{m_{d,j}(x,\theta_j)\}$ denotes the expected mean of the jth group desired PDF.

Theorem 1: For the error $e = x - \mathbb{E}\{m_{d,j}(x,\theta_j)\}$ correspond to an agent $\mathcal{A}$, there exist an error dynamic [7] defined as

$$de = \big[f_a(e) + g_a(e)u\big]dt + \sigma d\omega. \tag{4}$$

Proof: Provided in Appendix A. ■

Following this, the coupling function in (3) is designed to achieve the desired mixture-PDF as $\Phi(m(x;\theta)) = \|m_j(x;\theta) - m_{d,j}(x;\theta_j)\|_2^2$. This function measures the difference between the running PDF $m_j(x;\theta)$ and the expected final PDF $m_{d,j}(x;\theta_j)$ of each group j. Given the continuous dynamic in (1) represented by the stochastic differential equation and the cost function derived in (3), it is necessary to determine an admissible control to minimize the optimal cost function. Based on the Bellman's principle of optimality [17] and the optimal control [15] theory, the Hamiltonian is derived as

$$\begin{aligned} H\big[x, \partial_x J\big(x, m_j(x;\theta)\big)\big] &= \mathbb{E}\{r(x(t),u(t)) + \Phi\big(m_j(x;\theta)\big) \\ &+ \partial_x J^T\big(x, m_j(x;\theta)\big)[f(x) + g(x)u]\}. \end{aligned} \tag{5}$$

Subsequently, each agent's optimal control is evaluated as

$$\begin{aligned} 2Ru + g^T(x)\partial_x J\big(x, m_j(x;\theta)\big) &= 0 \\ u(x) = -\frac{1}{2}\mathbb{E}\{R^{-1}g^T(x)\partial_x J\big(x, m_j(x;\theta)\big)\}. \end{aligned} \tag{6}$$

Next, the associated HJB equation in (7), as shown at the bottom of the next page, is derived by inserting the optimal value function into the Hamiltonian equation. Also, the mass function (PDF) is achieved by solving the FPK as presented in (8), as shown at the bottom of the next page.

III. Imbalanced-MFG Theory and Multiactor–Critic-Mass Learning Algorithm

This section introduces a framework based on the Imb-MFG theory, aimed at achieving the final mixture-PDF through individual agents' action in LS-MAS. First, an adaptive PDF decomposition based on the induction theory is developed to break down the final desired PDF into a mixture of imbalanced normal distributions. Subsequently, the final desired PDF is attained by decomposing the LS-MAS into multiple groups using a K-means clustering algorithm with constraint, which ensures the appropriate combination of agents in individual groups in order to converge to imbalanced

normal distributions. Then, the distributed optimal control framework is designed based on the developed Imb-MFG theory. Particularly, the optimal policy for individual agents is derived by solving the coupled PDEs known as HJB and FPK equations. Inspired by emerging RL and ADP techniques [18], a novel M-ACM learning framework is developed to learn the Imb-MFG solution.

A. Adaptive PDF Decomposition Based on Induction Theory

An induction theory-based approach is designed to estimate the respective parameters i.e., weights, means, and variances of the final mixture-PDF. The ideal target PDF is reformulated as a mixture-PDF

$$m_d(x;\theta)=\sum_{j=1}^{N} w_j m_{d,j}\left(x;\mu_j,\Sigma_j\right) \tag{9}$$

with $w_j\in\mathbb{R}$, $\mu_j\in\mathbb{R}^n$, and $\Sigma_j\in\mathbb{R}^{n\times n}$ are the ideal weight, mean, and covariance parameters of individual group's normal distribution. Now, the (9) is reformulated as

$$m_d(x;\theta)=\boldsymbol{w}^T m_d(x{:}\boldsymbol{\mu},\boldsymbol{\Sigma}) \tag{10}$$

with $\boldsymbol{w}\in\mathbb{R}^N$ represents the ideal mixture-PDF weight. The estimated final mixture-PDF function is defined as

$$\hat{m}_d\left(x;\hat{\theta}\right)=\hat{\boldsymbol{w}}^T m_d\left(x;\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}}\right). \tag{11}$$

Now, the approximation error of the ideal mixture-PDF is

$$e_m=\boldsymbol{w}^T m_d(x;\boldsymbol{\mu},\boldsymbol{\Sigma})-\hat{\boldsymbol{w}}^T m_d\left(x;\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}}\right). \tag{12}$$

The mixture-PDF estimation error is written as $\tilde{m}_d(x;\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}})=m_d(x;\boldsymbol{\mu},\boldsymbol{\Sigma})-m_d(x;\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}})$. The (12) is as

$$\begin{aligned}e_m&=\boldsymbol{w}^T m_d(x;\boldsymbol{\mu},\boldsymbol{\Sigma})-\hat{\boldsymbol{w}}^T m_d\left(x;\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}}\right)\\&=\boldsymbol{w}^T m_d(x;\boldsymbol{\mu},\boldsymbol{\Sigma})-\boldsymbol{w}^T m_d\left(x;\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}}\right)+\boldsymbol{w}^T m_d\\&\quad\left(x;\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}}\right)-\hat{\boldsymbol{w}}^T m_d\left(x;\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}}\right)\\&=\boldsymbol{w}^T\tilde{m}_d\left(x;\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}}\right)+\tilde{\boldsymbol{w}}^T m_d\left(x;\hat{\boldsymbol{\mu}},\hat{\boldsymbol{\Sigma}}\right).\end{aligned} \tag{13}$$

Assumption 1: The mixture-PDF function follows the Lipschitz continuity assumption. It implies that there are Lipschitz constant, L_μ and L_Σ, such that the inequality $\|\tilde{m}_d(x;\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\Sigma}})\|\le L_\mu\|\tilde{\boldsymbol{\mu}}\|+L_\Sigma\|\tilde{\boldsymbol{\Sigma}}\|$ is satisfied.

Now, the residual error can be derived as $E_m=(1/2)e_m^T e_m$. If $\hat{\boldsymbol{w}}\to\boldsymbol{w}$, $\hat{\boldsymbol{\mu}}\to\boldsymbol{\mu}$, and $\hat{\boldsymbol{\Sigma}}\to\boldsymbol{\Sigma}$, then e_m approaches zero. An induction-based gradient descent approach is developed to update the estimated final mixture PDF parameters. Here, the iteration index in mixture PDF estimation is represented as l. In the remaining sections of this article, the bold notation will be excluded to simplify the presentation. However, it should be noted that the notation without bold font still refers to vectors and matrices. The update law is as follows:

$$\hat{w}^{[l+1]}=\hat{w}^{[l]}-\alpha_w\frac{\partial E_m}{\partial\hat{w}}=\hat{w}^{[l]}+\alpha_w m_d\left(x;\hat{\mu},\hat{\Sigma}\right)e_m^T \tag{14}$$

$$\hat{\mu}^{[l+1]}=\hat{\mu}^{[l]}-\alpha_\mu\frac{\partial E_m}{\partial\hat{\mu}}=\hat{\mu}^{[l]}+\alpha_\mu\hat{w}^{[l]}m_\mu\left(x;\hat{\mu},\hat{\Sigma}\right)e_m^T \tag{15}$$

$$\hat{\Sigma}^{[l+1]}=\hat{\Sigma}^{[l]}-\alpha_\Sigma\frac{\partial E_m}{\partial\hat{\Sigma}}=\hat{\Sigma}^{[l]}+\alpha_\Sigma\hat{w}^{[l]}m_\Sigma\left(x;\hat{\mu},\hat{\Sigma}\right)e_m^T \tag{16}$$

with the learning gain α_w, α_μ, and α_Σ. Now, the parameters approximation errors dynamics are

$$\tilde{w}^{[l+1]}=\tilde{w}^{[l]}-\alpha_w m_d\left(x;\hat{\mu},\hat{\Sigma}\right)e_m^T \tag{17}$$

$$\tilde{\mu}^{[l+1]}=\tilde{\mu}^{[l]}-\alpha_\mu\hat{w}^{[l]}m_\mu\left(x;\hat{\mu},\hat{\Sigma}\right)e_m^T \tag{18}$$

$$\tilde{\Sigma}^{[l+1]}=\tilde{\Sigma}^{[l]}-\alpha_\Sigma\hat{w}^{[l]}m_\Sigma\left(x;\hat{\mu},\hat{\Sigma}\right)e_m^T. \tag{19}$$

Theorem 2: The update law for the parameters of the mixture PDFs in gradient descent is defined in (14)-(16), with the positive constants representing the tuning gains. Based on the theory of the mathematical induction [19], for the base case scenario, the respective approximation errors of the parameters, denoted as $\tilde{w}^{[l]}$, $\tilde{\mu}^{[l]}$, and $\tilde{\Sigma}^{[l]}$ at iteration l, are UUB. Also, B_w, B_μ, and B_Σ represents the respective bounds of the errors. For the induction step at iteration $(l+1)$, if the UUB condition is satisfied for the base case at iteration l, then it must also apply for the subsequent case at iteration $(l+1)$.

Proof: Consider the following Lyapunov function candidate:

$$\begin{aligned}\Delta L_w&=\tilde{w}^{T^{[l+1]}}\tilde{w}^{[l+1]}-\tilde{w}^{T^{[l]}}\tilde{w}^{[l]}\\&=\left[\tilde{w}^{[l]}-\alpha_w m_d\left(x;\hat{\mu},\hat{\Sigma}\right)e_m^T\right]^T\left[\tilde{w}^{[l]}-\alpha_w m_d\left(x;\hat{\mu},\hat{\Sigma}\right)e_m^T\right]\\&\quad-\tilde{w}^{T^{[l]}}\tilde{w}^{[l]}.\end{aligned} \tag{20}$$

Next, inserting (13) and considering the Lipschitz function in Assumption 1, the (20) can be written as

$$\begin{aligned}\Delta L_w\le&-2\alpha_w\tilde{w}^{T^{[l]}}m_w\{w\left[L_\mu\|\tilde{\mu}^{[l-1]}\|+L_\Sigma\|\tilde{\Sigma}^{[l-1]}\|\right]\\&+\tilde{w}^{T^{[l]}}m_w\}+\alpha_w^2\|m_w\|^2\left\{w\left[L_\mu\|\tilde{\mu}^{[l-1]}\|+L_\Sigma\|\tilde{\Sigma}^{[l-1]}\|\right]\right.\\&\left.+\tilde{w}^{T^{[l]}}m_w\right\}^2\end{aligned} \tag{21}$$

with $m_w=m_d(x;\hat{\mu},\hat{\Sigma})$. Note that, in (17)–(19), each parameter approximation error depends on the other parameters. In the weight updates, the previous iteration approximation errors for the mean and covariance are used. For the mean update step, the current weight update and covariance of the previous iteration are considered. Finally, the covariance update uses the current weight and mean approximations, as these have already been updated within the current iteration. Next, the

$$\text{HJB: }\ \mathbb{E}\left\{\Phi\left(x,m_j(x;\theta)\right)\right\}=\mathbb{E}\left\{-\partial_t J\left(x,m_j(x;\theta)\right)-\frac{1}{2}\sigma^2\Delta J\left(x,m_j(x;\theta)\right)+H\left[x,\partial_x J\left(x,m_j(x;\theta)\right)\right]\right\} \tag{7}$$

$$\text{FPK: }\ \mathbb{E}\left\{\partial_t m_j(x;\theta)-\frac{1}{2}\sigma^2\Delta m_j(x;\theta)-div\left(m_j D_p H\left[x,\partial_x J(x,m_j(x;\theta))\right]\right)\right\}=0 \tag{8}$$

(21) can be rewritten in (22), derived at the bottom of the page, where

$$\Phi^{w}_{\text{con}}\left(\tilde{\mu}^{[l-1]}, \tilde{\Sigma}^{[l-1]}\right) = \left[L_{\mu}^2 + 4\alpha_w^2\|m_w\|^2\|w\|^2L_{\mu}^2\right] \|\tilde{\mu}^{[l-1]}\|^2 + \left[L_{\Sigma}^2 + 4\alpha_w^2\|m_w\|^2\|w\|^2L_{\Sigma}^2\right]\|\tilde{\Sigma}^{[l-1]}\|^2. \quad (24)$$

From (22), the first difference ΔL_w is less than zero if the following condition holds:

$$\|\tilde{w}^{[l]}\| > \sqrt{\frac{\Phi^{w}_{\text{con}}\left(\tilde{\mu}^{[l-1]}, \tilde{\Sigma}^{[l-1]}\right)}{\alpha_w\begin{bmatrix}2\|m_w\|^2 - 2\alpha_w \\ \|m_w\|^2\|w\|^2 - 2\alpha_w\|m_w\|^4\end{bmatrix}}} \equiv B_w. \quad (25)$$

Now, a Lyapunov function is considered as follows:

$$\Delta L_{\mu} = \tilde{\mu}^{T^{[l+1]}}\tilde{\mu}^{[l+1]} - \tilde{\mu}^{T^{[l]}}\tilde{\mu}^{[l]}. \quad (26)$$

Substituting (18) and (13), (23), shown at the bottom of the page, is achieved. with

$$\Phi^{\mu}_{\text{con}}\left(\tilde{w}^{[l]}, \Sigma^{[l-1]}\right) = \left[L_{\Sigma}^2\|w\|^2 + 4\alpha_{\mu}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2\|w\|^2L_{\Sigma}^2\right] \|\tilde{\Sigma}^{[l-1]}\|^2 + \left[m_w^2 + 2\alpha_{\mu}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2m_w^2\right]\|\tilde{w}^{[l]}\|^2. \quad (27)$$

From (23), the first difference of ΔL_{μ} is less than zero if the following condition holds:

$$\|\tilde{\mu}^{[l]}\| > \sqrt{\frac{\Phi^{\mu}_{\text{con}}\left(\tilde{w}^{[l]}, \Sigma^{[l-1]}\right)}{\alpha_{\mu}\begin{bmatrix}2\|\hat{w}^{[l]}\|L_{m_{\hat{\mu}}}wL_{\mu} - \alpha_{\mu}\|\hat{w}^{[l]}\|^2 \\ L_{m_{\hat{\mu}}}^2 - 4\alpha_{\mu}\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2\|w\|^2L_{\mu}^2\end{bmatrix}}} \equiv B_{\mu}. \quad (28)$$

Finally, a Lyapunov function is considered as follows:

$$\Delta L_{\Sigma} = \tilde{\Sigma}^{T^{[l+1]}}\tilde{\Sigma}^{[l+1]} - \tilde{\Sigma}^{T^{[l]}}\tilde{\Sigma}^{[l]}. \quad (29)$$

Substituting (19) and (13), the (29) is rewritten as

$$\begin{aligned}\Delta L_{\Sigma} &\le -2\alpha_{\Sigma}\tilde{\Sigma}^{[l]}\hat{w}^{[l]}L_{m_{\hat{\Sigma}}}\{w\left[L_{\mu}\|\tilde{\mu}^{[l]}\| + L_{\Sigma}\|\tilde{\Sigma}^{[l]}\|\right] \\ &+\tilde{w}^{T^{[l]}}m_w\} + \alpha_{\Sigma}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\Sigma}}}^2\{w\left[L_{\mu}\|\tilde{\mu}^{[l]}\| + L_{\Sigma}\|\tilde{\Sigma}^{[l]}\|\right] \\ &+\tilde{w}^{T^{[l]}}m_w\}^2 \\ &\le \alpha_{\Sigma}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\Sigma}}}^2\|\tilde{\Sigma}^{[l]}\|^2 + L_{\mu}^2\|w\|^2\|\tilde{\mu}^{[l]}\|^2 - 2\alpha_{\Sigma}\hat{w}^{T^{[l]}}w \\ &L_{m_{\hat{\Sigma}}}L_{\Sigma}\|\tilde{\Sigma}^{[l]}\|^2 + \alpha_{\Sigma}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\Sigma}}}^2\|\tilde{\Sigma}^{[l]}\|^2 + m_w^2\|\tilde{w}^{[l]}\|^2 + \\ &4\alpha_{\Sigma}^2\|\tilde{w}^{[l]}\|^2L_{m_{\hat{\Sigma}}}^2\|w\|^2L_{\mu}^2\|\tilde{\mu}^{[l]}\|^2 + 4\alpha_{\Sigma}^2\|\tilde{w}^{[l]}\|^2L_{m_{\hat{\Sigma}}}^2\|w\|^2 \\ &L_{\Sigma}^2\|\tilde{\Sigma}^{[l]}\|^2 + 2\alpha_{\Sigma}^2\|\tilde{w}^{[l]}\|^2L_{m_{\hat{\Sigma}}}^2m_w^2\|\tilde{w}^{[l]}\|^2 \\ &\le -[2\alpha_{\Sigma}\hat{w}^{T^{[l]}}wL_{m_{\hat{\Sigma}}}L_{\Sigma} - 2\alpha_{\Sigma}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\Sigma}}}^2 - 4\alpha_{\Sigma}^2\|\tilde{w}^{[l]}\|^2 \\ &L_{m_{\hat{\Sigma}}}^2\|w\|^2L_{\Sigma}^2]\|\tilde{\Sigma}^{[l]}\|^2 + \Phi^{\Sigma}_{\text{con}}\left(\tilde{w}^{[l]}, \tilde{\mu}^{[l]}\right) \end{aligned} \quad (30)$$

with

$$\Phi^{\Sigma}_{\text{con}}\left(\tilde{w}^{[l]}, \tilde{\mu}^{[l]}\right) = \left[L_{\mu}^2\|w\|^2 + 4\alpha_{\Sigma}^2\|\tilde{w}^{[l]}\|^2L_{m_{\hat{\Sigma}}}^2\|w\|^2L_{\mu}^2\right] \|\tilde{\mu}^{[l]}\|^2 + \left[m_w^2 + 2\alpha_{\Sigma}^2\|\tilde{w}^{[l]}\|^2L_{m_{\hat{\Sigma}}}^2m_w^2\right]\|\tilde{w}^{[l]}\|^2. \quad (31)$$

The first difference of ΔL_{Σ} is less than zero if the following condition holds:

$$\|\tilde{\Sigma}^{[l]}\| > \sqrt{\frac{\Phi^{\Sigma}_{\text{con}}\left(\tilde{w}^{[l]}, \mu^{[l]}\right)}{\alpha_{\Sigma}\begin{bmatrix}2\|\hat{w}^{[l]}\|wL_{m_{\hat{\Sigma}}}L_{\Sigma} - 2\alpha_{\Sigma}\|\hat{w}^{[l]}\|^2 \\ L_{m_{\hat{\Sigma}}}^2 - 4\alpha_{\Sigma}\|\tilde{w}^{[l]}\|^2L_{m_{\hat{\Sigma}}}^2\|w\|^2L_{\Sigma}^2\end{bmatrix}}} \equiv B_{\Sigma}. \quad (32)$$

Here, $L_{m_{\hat{\mu}}}$ and $L_{m_{\hat{\Sigma}}}$ represent the Lipschitz constants. Using an approach similar to the previous method, the following condition can be derived for the next iteration as follows:

$$\|\tilde{w}^{[l+1]}\| > \sqrt{\frac{\Phi^{w}_{\text{con}}\left(\tilde{\mu}^{[l]}, \tilde{\Sigma}^{[l]}\right)}{\alpha_w\begin{bmatrix}2\|m_w\|^2 - 2\alpha_w \\ \|m_w\|^2\|w\|^2 - 2\alpha_w\|m_w\|^4\end{bmatrix}}} \quad (33)$$

with $\Phi^{w}_{\text{con}}(\tilde{\mu}^{[l]}, \tilde{\Sigma}^{[l]}) < \Phi^{w}_{\text{con}}(\tilde{\mu}^{[l-1]}, \tilde{\Sigma}^{[l-1]})$, $\|\tilde{\mu}^{[l]}\| < \|\tilde{\mu}^{[l-1]}\|$, and $\|\tilde{\Sigma}^{[l]}\| < \|\tilde{\Sigma}^{[l-1]}\|$. Similarly

$$\|\tilde{\mu}^{[l+1]}\| > \sqrt{\frac{\Phi^{\mu}_{\text{con}}\left(\tilde{w}^{[l+1]}, \Sigma^{[l]}\right)}{\alpha_{\mu}\begin{bmatrix}2\|\hat{w}^{[l+1]}\|L_{m_{\hat{\mu}}}wL_{\mu} - \alpha_{\mu}\|\hat{w}^{[l+1]}\|^2 \\ L_{m_{\hat{\mu}}}^2 - 4\alpha_{\mu}\|\hat{w}^{[l+1]}\|^2L_{m_{\hat{\mu}}}^2\|w\|^2L_{\mu}^2\end{bmatrix}}} \quad (34)$$

$$\begin{aligned}\Delta L_w &\le -2\alpha_w\tilde{w}^{T^{[l]}}wm_wL_{\mu}\|\tilde{\mu}^{[l-1]}\| - 2\alpha_w\tilde{w}^{T^{[l]}}wm_wL_{\Sigma}\|\tilde{\Sigma}^{[l-1]}\| - 2\alpha_w\|m_w\|^2\|\tilde{w}^{[l]}\|^2 + 4\alpha_w^2\|m_w\|^2wL_{\mu}^2\|\tilde{\mu}^{[l-1]}\|^2 \\ &+ 4\alpha_w^2\|m_w\|^2\|w\|^2L_{\Sigma}^2\|\tilde{\Sigma}^{[l-1]}\|^2 + 2\alpha_w^2\|m_w\|^4\|\tilde{w}^{[l]}\|^2 \\ &\le \alpha_w^2\|m_w\|^2\|w\|^2\|\tilde{w}^{[l]}\|^2 + L_{\mu}^2\|\tilde{\mu}^{[l-1]}\|^2 + \alpha_w^2\|\tilde{w}^{[l]}\|^2\|m_w\|^2\|w\|^2 + L_{\Sigma}^2\|\tilde{\Sigma}^{[l-1]}\|^2 + 4\alpha_w^2\|m_w\|^2\|w\|^2L_{\mu}^2\|\tilde{\mu}^{[l-1]}\|^2 + 4\alpha_w^2 \\ &\|m_w\|^2\|w\|^2L_{\Sigma}^2\|\tilde{\Sigma}^{[l-1]}\|^2 + 2\alpha_w^2\|m_w\|^4\|\tilde{w}^{[l]}\|^2 - 2\alpha_w\|m_w\|^2\|\tilde{w}^{[l]}\|^2 \\ &\le -\left[2\alpha_w\|m_w\|^2 - 2\alpha_w^2\|m_w\|^2\|w\|^2 - 2\alpha_w^2\|m_w\|^4\right]\|\tilde{w}^{[l]}\|^2 + \Phi^{w}_{\text{con}}\left(\tilde{\mu}^{[l-1]}, \tilde{\Sigma}^{[l-1]}\right) \end{aligned} \quad (22)$$

$$\begin{aligned}\Delta L_{\mu} &\le \tilde{\mu}^{T^{[l]}}\tilde{\mu}^{[l]} - 2\alpha_{\mu}\tilde{\mu}^{T^{[l]}}\|\hat{w}^{[l]}\|L_{m_{\hat{\mu}}}\{w\left[L_{\mu}\|\tilde{\mu}^{[l]}\| + L_{\Sigma}\|\tilde{\Sigma}^{[l-1]}\|\right] + \tilde{w}^{T^{[l]}}m_w\} + \alpha_{\mu}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2\{w\left[L_{\mu}\|\tilde{\mu}^{[l]}\| + L_{\Sigma}\|\tilde{\Sigma}^{[l-1]}\|\right] \\ &+ \tilde{w}^{T^{[l]}}m_w\}^2 - \tilde{\mu}^{T^{[l]}}\tilde{\mu}^{[l]} \\ &\le -2\alpha_{\mu}\|\hat{w}^{[l]}\|L_{m_{\hat{\mu}}}wL_{\mu}\|\tilde{\mu}^{[l]}\|^2 + \alpha_{\mu}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2\|\tilde{\mu}^{[l]}\|^2 + L_{\Sigma}^2\|\tilde{\Sigma}^{[l-1]}\|^2\|w\|^2 + \alpha_{\mu}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2\|\tilde{\mu}^{[l]}\|^2 + m_w^2\|\tilde{w}^{[l]}\|^2 \\ &+ 4\alpha_{\mu}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2\|w\|^2L_{\mu}^2\|\tilde{\mu}^{[l]}\|^2 + 4\alpha_{\mu}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2\|w\|^2L_{\Sigma}^2\|\tilde{\Sigma}^{[l-1]}\|^2 + 2\alpha_{\mu}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2m_w^2\|\tilde{w}^{[l]}\|^2 \\ &\le -\left[2\alpha_{\mu}\|\hat{w}^{[l]}\|L_{m_{\hat{\mu}}}wL_{\mu} - \alpha_{\mu}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2 - 4\alpha_{\mu}^2\|\hat{w}^{[l]}\|^2L_{m_{\hat{\mu}}}^2\|w\|^2L_{\mu}^2\right]\|\tilde{\mu}^{[l]}\|^2 + \Phi^{\mu}_{\text{con}}\left(\tilde{w}^{[l]}, \Sigma^{[l-1]}\right) \end{aligned} \quad (23)$$

with $\Phi^{\mu}_{\text{con}}(\tilde{w}^{[l+1]}, \tilde{\Sigma}^{[l]}) < \Phi^{\mu}_{\text{con}}(\tilde{w}^{[l]}, \tilde{\Sigma}^{[l-1]})$, $\|\tilde{\Sigma}^{[l]}\| < \|\tilde{\Sigma}^{[l-1]}\|$, and $\|\tilde{w}^{[l+1]}\| < \|\tilde{w}^{[l]}\|$. And

$$\|\tilde{\Sigma}^{[l+1]}\| > \sqrt{\frac{\Phi^{\Sigma}_{\text{con}}\left(\tilde{w}^{[l+1]}, \mu^{[l+1]}\right)}{\alpha_{\Sigma}\begin{bmatrix} 2\|\hat{w}^{[l+1]}\| w L_{m_{\hat{\Sigma}}} L_{\Sigma} - 2\alpha_{\Sigma}\|\hat{w}^{[l+1]}\|^2 \\ L^2_{m_{\hat{\Sigma}}} - 4\alpha_{\Sigma}\|\tilde{w}^{[l+1]}\|^2 L^2_{m_{\hat{\Sigma}}}\|w\|^2 L^2_{\Sigma} \end{bmatrix}}} \tag{35}$$

where $\Phi^{\Sigma}_{\text{con}}(\tilde{w}^{[l+1]}, \mu^{[l+1]}) < \Phi^{\Sigma}_{\text{con}}(\tilde{w}^{[l]}, \mu^{[l]})$, $\|\tilde{\mu}^{[l+1]}\| < \|\tilde{\mu}^{[l]}\|$, and $\|\tilde{\Sigma}^{[l+1]}\| < \|\tilde{\Sigma}^{[l]}\|$. The estimated weight of the mixture-PDF is represented as $\hat{w} = \{\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_N\}$. Next, a K-means clustering algorithm with [13] is incorporated to decompose the LS-MAS with mean-field interaction into N groups. In this context, the number of clusters is set to $K = N$, with $j = 1, 2, \ldots, K$ denoting the cluster index. The minimum agents number in the jth cluster is evaluated as $p_j = (\hat{w}_j/[\sum_{j=1}^{K} \hat{w}_j])M$, for $\sum_{j=1}^{K} p_j \leq M$ and M is the total agents number. Next, starting with the cluster centers $C_{1,t}, C_{2,t}, \ldots, C_{K,t}$ at iteration t, the cluster assignment and update step are outlined as follows.

1) For the agent i with state x_i, assign x_i to cluster j by minimizing the given function, ensuring that the cluster center $C_{j,t}$ is closest to the position x_i. In this step, the cluster center for group j is fixed, and the selection variable $q_{i,j}$ is the solution to the following cost function:

$$\begin{aligned} \min_{q} \quad & \sum_{i=1}^{M}\sum_{j=1}^{K} q_{i,j}\left(\frac{1}{2}\|x_i - C_{j,t}\|_2^2\right) \\ \text{s.t.} \quad & \sum_{i=1}^{M} q_{i,j} \geq p_j, \sum_{j=1}^{K} q_{i,j} = 1, q_{i,j} \geq 0. \end{aligned} \tag{36}$$

Be aware that the first constraint of the cost function guarantees that the number of agents chosen for a specific cluster j always meets the minimum threshold requirement of agents allocated to that cluster. The second constraint ensures that each agent is assigned exclusively to a single cluster.

2) Update the cluster center $C_{j,t+1}$ at iteration $t+1$

$$C_{j,t+1} = \begin{cases} \frac{\sum_{i=1}^{M} q^t_{i,j} x_i}{\sum_{i=1}^{M} q^t_{i,j}} & \text{if } \sum_{i=1}^{M} q^t_{i,j} > 0 \\ C_{j,t} & \text{otherwise.} \end{cases} \tag{37}$$

The second step update the cluster center at each iteration. Then, if the center of cluster j at iteration t stays unchanged in the subsequent iteration $t+1$, meaning $C_{j,t+1} = C_{j,t}$, the algorithm is terminated, otherwise, the iteration t is incremented by 1 and the procedure returns to step 1. This algorithm is executed for all clusters j. Following the decomposition of LS-MAS into N groups, each agent acquires an optimal control strategy based on the M-ACM algorithm to collectively achieve the desired final mixture-PDF.

Remark 1: Solving the combined HJB-FPK equations in the Imb-MFG poses a significant challenge [7]. To address this challenge, an M-ACM-based NN learning is designed to achieve the HJB-FPK equations' solution.

■

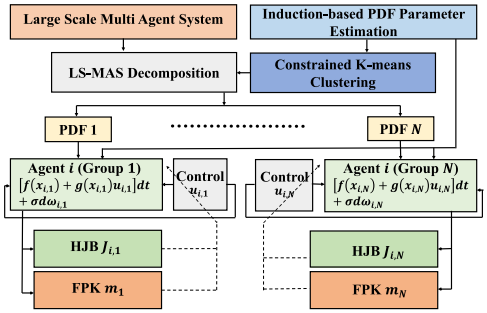


Fig. 1. M-ACM learning structure for Imb-MFG optimal control.

B. Multiactor–Critic-Mass (M-ACM) Algorithm

The M-ACM method is designed in this part. The goal of each agent from decomposed groups is to find the optimal control strategies to achieve the final desired PDF function collectively. To achieve the final objective, each agent from a particular group operates three NN models, see Fig. 1. Here, the *critic NN* is employed to estimate the optimal value function, the *actor NN* is utilized to evaluate the optimal control input strategy, and the *mass NN* is designed to estimate the PDF function from each group. Now, the ideal cost function, the PDF function, and the control input are defined as $J(x, m_j) = \mathbb{E}\{W_J^T \phi_J(x, m_j) + \varepsilon_{\text{HJB}}\}$, $m_j(x,t) = \mathbb{E}\{W_{m_j}^T \phi_{m_j}(x, J, t) + \varepsilon_{\text{FPK}}$, and $u(x, m_j) = \mathbb{E}\{W_u^T \phi_u(x, m_j) + \varepsilon_{\text{u}}\}$, respectively, where W_J, W_{m_j}, and W_u represents the weights of the critic-mass-actor NNs for the jth group agent. Moreover, the activation functions of the respective NNs are given as ϕ_J, ϕ_{m_j}, and ϕ_u. The critic, mass, and actor NNs reconstruction error are given as ε_{HJB}, ε_{FPK}, and ε_u. Then, the approximated cost, PDF functions, and the optimal control are defined as $\hat{J}(x, \hat{m}_i) = \mathbb{E}\{\hat{W}_J^T \hat{\phi}_J(x, \hat{m}_j)\}$, $\hat{m}_j(x,t) = \mathbb{E}\{\hat{W}_{m_j}^T \hat{\phi}_{m_j}(x, \hat{J}, t)\}$, and $\hat{u}(x, \hat{m}_j) = \mathbb{E}\{\hat{W}_u^T \hat{\phi}_u(x, \hat{m}_j)\}$, respectively. By inserting these estimated functions into the HJB, FPK, and optimal control input (7), (8), and (6), the residuals errors are generated. These errors are then utilized to tune the respective NNs

$$\mathbb{E}\{e_{\text{HJB}}\} = \mathbb{E}\left\{\Phi(x, \hat{m}_j) + \hat{W}_J^T\left[\partial_t \hat{\phi}_J + 0.5\sigma^2 \Delta\hat{\phi}_J - \hat{H}_W\right]\right\} \tag{38}$$

$$\mathbb{E}\{e_{\text{FPK}}\} = \mathbb{E}\left\{\hat{W}_{m_j}^T \partial_t \hat{\phi}_{m_j} - 0.5\sigma^2 \Delta\hat{\phi}_{m_j} - div\left(\hat{\phi}_{m_j}\right) D_p \hat{H}\right\} \tag{39}$$

$$\mathbb{E}\{e_{\text{u}}\} = \mathbb{E}\left\{\hat{W}_u^T \hat{\phi}_u + 1/2R^{-1} g^T(x) D\hat{J}(x, \hat{m}_j)\right\} \tag{40}$$

with $\hat{H} = W_J^T \hat{H}_W$ and $\hat{H} = H[x, \partial_x \hat{\phi}_J]$. Let

$$\begin{aligned} \Psi_J &= \partial_t \hat{\phi}_J + 0.5\sigma^2 \Delta\hat{\phi}_J - \hat{H}_W \\ \Psi_{m_j} &= \partial_t \hat{\phi}_{m_j} - 0.5\sigma^2 \Delta\hat{\phi}_{m_j} - div\left(\hat{\phi}_{m_j} D_p \hat{H}\right) \\ \Phi(x, \tilde{m}_j) &= \Phi(x, \hat{m}_j) - \Phi(x, m_j). \end{aligned} \tag{41}$$

Then, the residual errors (38) and (39) are as follows:

$$\mathbb{E}\{e_{\text{HJB}}\} = \mathbb{E}\left\{\Phi(x, m_j) + \Phi(x, \tilde{m}_j) + \hat{W}_J^T \Psi_J\right\} \tag{42}$$

$$\mathbb{E}\{e_{\text{FPK}}\} = \mathbb{E}\left\{\hat{W}_{m_j}^T \Psi_{m_j}\left(x, \hat{J}, t\right)\right\}. \tag{43}$$

Next, the effect of reconstruction errors is considered by inserting the ideal functions into (7) and (8)

$$\mathbb{E}\{\Phi(x, m_j) + W_J^T \Psi_J(x, m_j) + \varepsilon_{\text{HJB}}\} = 0 \tag{44}$$

$$\mathbb{E}\{W_{m_j}^T \Psi_{m_j}(x, J, t) + \varepsilon_{\text{FPK}}\} = 0 \tag{45}$$

where ε_{HJB} and ε_{FPK} are the reconstruction errors. Again, substituting (44) and (45) into (42) and (43), we have

$$\mathbb{E}\{e_{\text{HJB}}\} = \mathbb{E}\Big\{\Phi\big(x, \tilde{m}_j\big) - \tilde{W}_J^T \hat{\Psi}_J - W_J^T \tilde{\Psi}_J - \varepsilon_{\text{HJB}}\Big\} \tag{46}$$

$$\mathbb{E}\{e_{\text{FPK}}\} = \mathbb{E}\Big\{-\tilde{W}_{m_j}^T \hat{\Psi}_{m_j} - W_{m_j}^T \tilde{\Psi}_{m_j} - \varepsilon_{\text{FPK}}\Big\} \tag{47}$$

$$\mathbb{E}\{e_{\text{u}}\} = \mathbb{E}\Big\{-\tilde{W}_u^T \hat{\phi}_u\big(x, \hat{m}_j\big) - W_u^T \tilde{\phi}_u\big(x, \tilde{m}_j\big) - \frac{1}{2} R^{-1} g^T(x) \partial_x \tilde{J}\big(x, \tilde{m}_j\big) - \varepsilon_{\text{u}}\Big\}. \tag{48}$$

The update rules for the critic, mass, and actor in the context of gradient descent algorithm for any agent $\mathcal{A}$ in the group j can be derived with the learning rates α_J, α_{m_j}, and α_u

$$\mathbb{E}\{\dot{\hat{W}}_J\} = \mathbb{E}\left\{-\alpha_J \frac{\Psi_J\big(x, \hat{m}_j\big) e_{\text{HJB}}^T}{1 + \|\Psi_J\big(x, \hat{m}_j\big)\|^2}\right\} \tag{49}$$

$$\mathbb{E}\{\dot{\hat{W}}_{m_j}\} = \mathbb{E}\left\{-\alpha_{m_j} \frac{\Psi_{m_j}\Big(x, \hat{J}, t\Big) e_{\text{FPK}}^T}{1 + \|\Psi_{m_j}\Big(x, \hat{J}, t\Big)\|^2}\right\} \tag{50}$$

$$\mathbb{E}\{\dot{\hat{W}}_u\} = \mathbb{E}\left\{-\alpha_u \frac{\phi_u\big(x, \hat{m}_j\big) e_{\text{u}}^T}{1 + \|\phi_u\big(x, \hat{m}_j\big)\|^2}\right\}. \tag{51}$$

Theorem 3: The critic NN weight $\mathbb{E}\{\hat{W}_J\}$ is updated by the tuning rule provided in (49), assuming $\Psi_J(x, \hat{m}_j)$ is persistently exciting (PE) [20], [21]. The approximated critic NN's weight error $\mathbb{E}\{\tilde{W}_J\}$ and the cost estimation error $\mathbb{E}\{\tilde{J}\}$ are UUB. Additionally, if the reconstruction error [8] is ignored, implying the error is zero, then $\mathbb{E}\{\tilde{W}_J\}$ and $\mathbb{E}\{\tilde{J}\}$ are asymptotically stable. The bound of $\mathbb{E}\{\tilde{J}\}$ is derived as $\mathbb{E}\{\|\tilde{J}(t)\|\} = \mathbb{E}\{\|\tilde{W}_J^T \hat{\phi}_J + W_J^T \tilde{\phi}_J + \varepsilon_{\text{HJB}}\|\} \leq b_{W_J} \mathbb{E}\{\|\hat{\phi}_J\|\} + l_{\phi_J} \mathbb{E}\{\|W_J\|\} b_{m_j} + \mathbb{E}\{\|\varepsilon_{\text{HJB}}\|\} \equiv b_J$, where l_{ϕ_J} represents the Lipschitz constant of the critic NN activation functions' ϕ_J.

Proof: Provided in Appendix B. ∎

Theorem 4: The mass NN weight $\mathbb{E}\{\hat{W}_{m_j}\}$ is updated by the tuning rule (50), assuming $\Psi_{m_j}(x, \hat{J}, t)$ is PE [20], [21]. The approximation error of the mass NN weight $\mathbb{E}\{\tilde{W}_{m_j}\}$ and the PDF approximation error $\mathbb{E}\{\tilde{m}_j\}$ are UUB. Moreover, if the reconstruction error [22] is ignored, then $\mathbb{E}\{\tilde{W}_{m_j}\}$ and $\mathbb{E}\{\tilde{m}_j\}$ are asymptotically stable. The mass approximation error bound $\tilde{m}_j$ is calculated as $\mathbb{E}\{\|\tilde{m}_j\|\} = \mathbb{E}\{\|\tilde{W}_{m_j}^T \hat{\phi}_{m_j} + \varepsilon_{\text{FPK}}\|\} \leq b_{W_{m_j}} \mathbb{E}\{\|\hat{\phi}_{m_j}\|\} + \mathbb{E}\{\|\varepsilon_{\text{FPK}}\|\} \equiv b_{m_j}$.

Proof: Provided in Appendix C. ∎

Theorem 5: The actor NN weight $\mathbb{E}\{\hat{W}_u\}$ is updated by the tuning rule provided in (51), assuming $\phi_u(x, \hat{m}_j)$ is PE [20], [21]. Now, the actor NN weight error $\mathbb{E}\{\tilde{W}_u\}$ and the approximation errors $\mathbb{E}\{\tilde{u}\}$ of the control input are UUB. Moreover, if the reconstruction error [22] is ignored, implying the error is zero, then $\mathbb{E}\{\tilde{W}_u\}$ and $\mathbb{E}\{\tilde{u}\}$ are asymptotically stable. Finally, the bound for $\tilde{u}$ is calculated as $\|\mathbb{E}\{\tilde{u}\}\| = \mathbb{E}\{\|\tilde{W}_u^T(t) \hat{\phi}_u + W_u^T \tilde{\phi}_u + \varepsilon_{\text{u}}\|\} \leq b_{W_u}(t) \mathbb{E}\{\|\hat{\phi}_u\|\} + l_{\phi_u} \mathbb{E}\{\|W_u\|\} b_{m_j}(t) + \mathbb{E}\{\|\varepsilon_u\|\} \equiv b_u$.

Proof: Provided in Appendix D. ∎

Algorithm 1 Imb-MFG Theory: Induction-Based PDF Decomposition and M-ACM Learning

1: Initialize the state of M agents in LS-MAS.
2: Set iteration $l = 0$ and initialize the final PDF constraint parameters $\hat{w}^{[l]}$, $\hat{\mu}^{[l]}$ and $\hat{\Sigma}^{[l]}$.
3: Initialize the PDF function approximation error $e_m \leftarrow \infty$.
4: Initialize the threshold δ_{e_m} of the approximation error.
5: Set the tuning gain α_w, α_μ and α_Σ.
6: **while** $e_m \geq \delta_{e_m}$ **do**
7: Update $\hat{w}$ using (14) and employ $\tilde{\mu}$ and $\tilde{\Sigma}$ from iteration $l-1$.
8: Update $\hat{\mu}$ using (15) and employ $\tilde{w}$ from iteration l and $\tilde{\Sigma}$ from iteration $l-1$.
9: Update $\hat{\Sigma}$ using (16) and employ $\tilde{w}$ and $\tilde{\mu}$ from current iteration l.
10: Update error e_m using (13).
11: Update iteration $l \leftarrow l+1$
12: **end while**
13: Define the number of clusters $K = N$.
14: Define the minimum number of agents in any cluster j using $p_j = (\frac{\hat{w}_j}{\sum_{j=1}^K \hat{w}_j})M$, with $\sum_{j=1}^K p_j \leq M$.
15: Initialize the iteration t.
16: Initialize the cluster center $C_{j,t}$ for any cluster j at iteration t.
17: Solve (36), to assign any agent i to the nearest cluster j.
18: Update cluster center C_j for next iteration $t+1$ using (37).
19: Repeat step 17 and 18 until convergence i.e., $C_{j,t+1} = C_{j,t}$.
20: Initialize M-ACM NN weights $\hat{W}_J$, $\hat{W}_{m_j}$ and $\hat{W}_u$ randomly.
21: Initialize NN errors- e_{HJB}, e_{FPK} and $e_{\text{u}} \leftarrow \infty$
22: Initialize thresholds δ_{HJB}, δ_{FPK} and δ_{u}
23: **while** TRUE **do**
24: **while** $e_{\text{HJB}} \geq \delta_{\text{HJB}}, e_{\text{FPK}} \geq \delta_{\text{FPK}}, e_{\text{u}} \geq \delta_{\text{u}}$ **do**
25: Update the weights using (49),(50) and (51).
26: Update the residual errors using (46),(47) and (40).
27: **end while**
28: $\hat{u}(x) \leftarrow \hat{W}_u^T \hat{\phi}_u(x, \hat{m}_j)$
29: Implement the control $\hat{u}$.
30: Observe the new state x.
31: **end while**

Lemma 1: For the optimal control policy u in (4)

$$\mathbb{E}\left\{e^T\left[f_a(e(t)) + g_a(e(t))u(t) + \frac{\sigma d\omega}{dt}\right]\right\} \leq -\gamma \mathbb{E}\{\|e\|^2\}. \tag{52}$$

Theorem 6: The NNs' weights are updated by the tuning rule in (49)–(51), assuming the learning rates α_J, α_{m_j}, and α_u are positive. Then, $\mathbb{E}\{\tilde{W}_J\}$, $\mathbb{E}\{\tilde{W}_u\}$, $\mathbb{E}\{\tilde{W}_{m_j}\}$, and $\mathbb{E}\{e\}$ are UUB. Also, if the reconstruction error [22] is ignored, implying the error is zero, then $\mathbb{E}\{\tilde{W}_J\}$, $\mathbb{E}\{\tilde{W}_{m_j}\}$, $\mathbb{E}\{\tilde{W}_u\}$, and $\mathbb{E}\{e\}$ are asymptotically stable.

Proof: Provided in Appendix E. The proposed method is implemented using Algorithm 1 in the simulation section. ∎

IV. Simulation Results

The efficiency of the developed algorithm is showcased using the large-scale UAV (LS-UAV). A total number of 1200 UAVs were initially used. The objective for individual agents within LS-UAVs is to collaboratively achieve a final PDF distribution as a desired formation. In practical situations, e.g., large-scale pursuit-evasion [23], etc., successfully achieving an arbitrary distribution assists the agents in collectively forming diverse shapes. Now, a normal distribution $\mathcal{N}(\mu = [4.5 \quad 7]$ and $\Sigma = \begin{bmatrix} 1 & 0.9 \\ 0.9 & 3 \end{bmatrix})$ is used to generate the initial states of agents. Also, the functions for SDE in (1) is defined as $f(x) =$

TABLE I
PARAMETERS AND THEIR VALUES

Parameters	Value
Coefficient, Q	I_2
Coefficient, R	1
Wiener process coefficient, σ	$0.05 \times I_2$
Tuning gain, α_w, α_μ, α_Σ	1×10^{-3}, 1.7×10^{-4}, 1×10^{-4}
Error threshold, δ_{e_m}	1×10^{-5}
Critic learning rate, α_J	2×10^{-5}
Actor learning rate, α_u	2×10^{-4}
Mass learning rate, α_{m_j}	2×10^{-3}
HJB error threshold, δ_{HJB}	1×10^{-5}
FPK error threshold, δ_{FPK}	1×10^{-3}
Actor error threshold, δ_u	1×10^{-2}

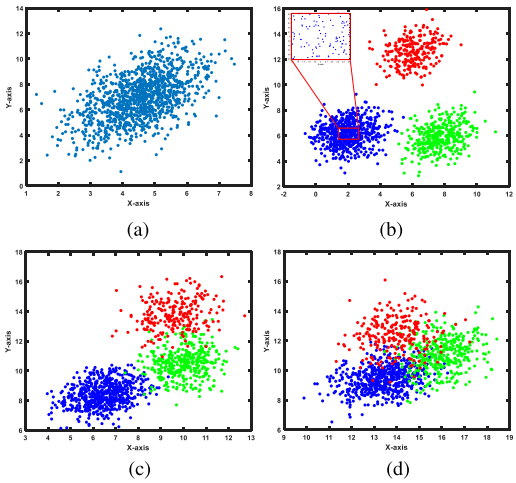


Fig. 2. (a) Initial locations at time $t = 0$ s. (b) Position of decomposed UAVs at time $t = 20$ s. (C) Positions at $t = 45$ s. (d) Final distributions of UAVs at time $t = 60$ s.

$\begin{bmatrix} -x_1 + (1/2)x_2^2 \\ -0.40x_2^2 \end{bmatrix}$ and $g(x) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, where $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ being the state of arbitrary agent $\mathcal{A}$.

However, each agent is unaware of the desired distribution before initiating the mission. This assumption aligns well with real-world scenarios, particularly in battlefield situations where the agents might lack prior knowledge of their surroundings. Here, the final PDF distribution is represented as a Gaussian mixture defined in (2) as $m_d(x;\theta) = \sum_{j=1}^{N} w_j m_{d,j}(x;\theta_j)$. Here, N is the number of Gaussian components and $m_{d,j}(x;\theta_j) = (1/[(2\pi)^{(n/2)}|\Sigma_j|^{(1/2)}]\exp(-(1/2)(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j))$. Now, an iterative induction-based parameters estimation framework has been used. The selected parameters for iterative induction-based estimation and the M-ACM algorithm are given in Table I. The estimated weights of the mixture are obtained as $\hat{w}_1 = 0.495$, $\hat{w}_2 = 0.3025$, and $\hat{w}_3 = 0.2025$, with the cluster number $N = 3$. The estimated mean and covariance are as $\hat{\mu}_1 = \begin{bmatrix} 13.4256 & 9.5846 \end{bmatrix}^T$, $\hat{\Sigma}_1 = \begin{bmatrix} 1.1956 & 0.334 \\ 0.334 & 0.7661 \end{bmatrix}$, $\hat{\mu}_2 = \begin{bmatrix} 15.9451 & 10.8812 \end{bmatrix}^T$, $\hat{\Sigma}_2 = \begin{bmatrix} 0.7845 & 0.1826 \\ 0.1826 & 1.3426 \end{bmatrix}$, $\hat{\mu}_3 = \begin{bmatrix} 14.0557 & 12.4397 \end{bmatrix}^T$, and $\hat{\Sigma}_3 = \begin{bmatrix} 0.9882 & 0.2105 \\ 0.2105 & 1.2259 \end{bmatrix}$.

The large number of agents deployed in the LS-UAVs are noncooperative and their interactions are captured through MFG. However, attaining the intended PDF through mean-field agents can be challenging, primarily because it necessitates intricate nonlinear interactions among agents,

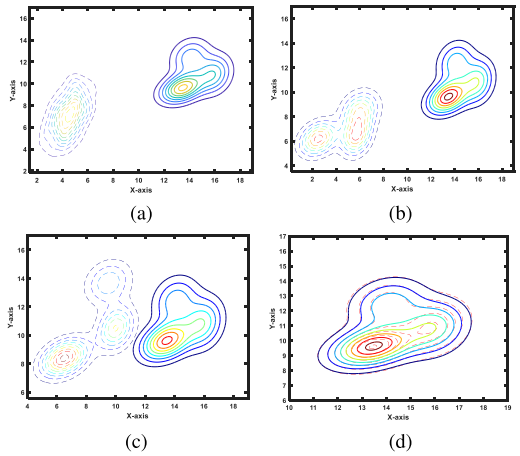


Fig. 3. PDF function contour plot. The real-time PDF and desired final PDF are shown in dashed and solid lines. (a) $t = 0$ s. (b) $t = 30$ s. (c) $t = 45$ s. (d) $t = 60$ s.

especially when dealing with large-scale systems with high dimensions. Nevertheless, breaking down agents into several groups, which in turn leads to the decomposition of a single mean field PDF into multiple PDFs, enables them to reach the desired final PDF more effectively. Next, we set the number of clusters for the constrained K-means algorithm, denoted as K, to be equal to N, which is 3 in this case. Here, the estimated minimum number of agents in each clusters are 596, 363, and 241, respectively. Next, the agents are assigned to clusters using (36) and (37). Also, each agent solves a pair of coupled HJB and FPK equations attained from the Imb-MFG. To address these imbalanced PDEs and attain optimal control, the proposed M-ACM algorithm is utilized. Fig. 2 illustrates how the positions of UAVs evolve, and these changes are visually represented through dots on a plot at specific time intervals. The initial and the decomposed states of UAVs are illustrated in Fig. 2(a) and (b), respectively. The different colors depict different groups of UAVs. Also, a small window is plotted to show the position of group 1 UAVs in detail. Fig. 2(c) shows the position of the UAVs at time $t = 45$ s. As time passes, each UAV adjusts its position with the goal of collectively reaching the desired distribution. At the end of the simulation (t = 60 s), each group of UAVs successfully reaches a position meeting the ε-Nash equilibrium, thereby achieving the desired final PDF. The contour plot of the PDF function is demonstrated in Fig. 3. Fig. 3(a) shows the initial PDF of LS-UAVs in dashed lines and also the final desired PDF with solid lines. Fig. 3(d) shows that the desired PDF is achieved by LS-UAVs. The final PDF contour plot is shown with dashed red lines to differentiate it from the desired PDF.

Next, Fig. 4 demonstrates the PDF distribution of agents. In Fig. 4(a) and (e), the initial PDF is presented in a two-dimensional (2-D) and three-dimensional (3-D) views, respectively. After estimating the parameters of the final mixture-PDF, we applied a constrained K-means clustering algorithm to partition the UAVs into multiple groups. The respective decomposed PDFs for these groups are demonstrated in Fig. 4(b) and (f) for 2-D and 3-D views. Then, Fig. 4(c) and (g) shows the PDF of all UAVs at time $t = 45$ s. Then, the final mixture PDF is illustrated in Fig. 4(d) and (h). Next, Fig. 5 demonstrates the final PDF percentage estimation

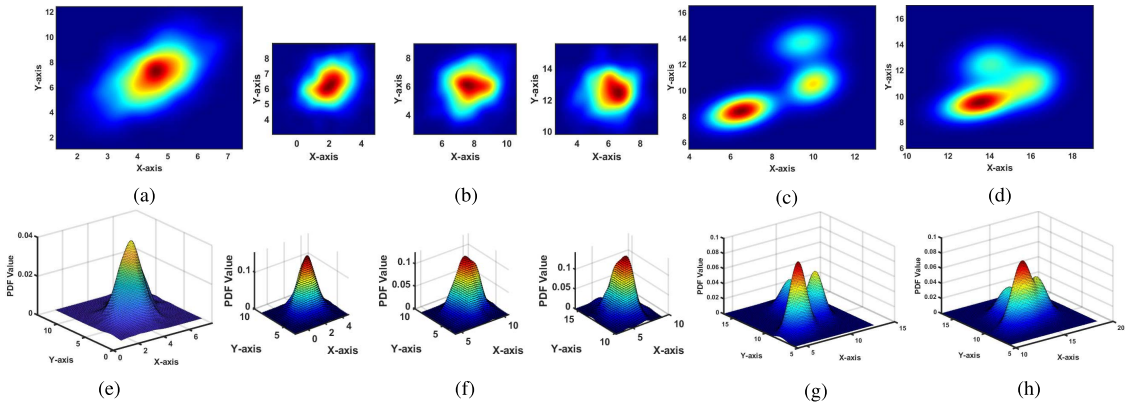


Fig. 4. PDF of the LS-MAS in 2-D and 3-D view. (a) and (e) Initial normal distributed UAVs PDF at $t = 0$ s (2-D and 3-D view). (b) and (f) Decomposed PDF of three groups of UAVs. (c) and (g) UAVs are moving to achieve the final mixture PDF. This figure shows the PDF at time $t = 45$ s. (d) and (h) Final mixture-PDF at time $t = 60$ s.

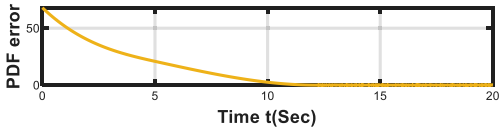


Fig. 5. Estimation error of final PDF.

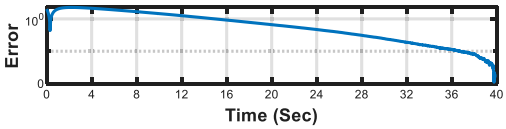


Fig. 8. FPK equation error of an UAV.

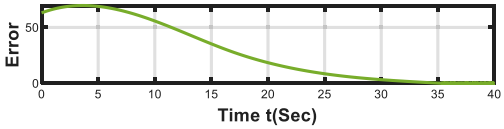


Fig. 6. Percentage error in PDF transition from initial distribution to achieving the final mixture distribution.

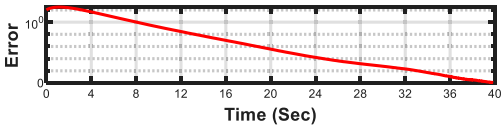


Fig. 7. HJB equation error of an UAV in the LS-UAV system.

error using, $\text{Error}_1 = ([m_d(x;\theta) - \hat{m}_d(x;\theta)]/m_d(x;\theta)) \times 100$. The PDF is estimated using the proposed iterative induction-based parameter estimation method. This figure clearly shows that the PDF function approximation error converges to zero with time. A total of 350 iterations were executed to reduce the error in approximating the PDF to a level below the threshold δ_{e_m}. Then, the percentage error of achieving that desired PDF by LS-UAVs is implemented in Fig. 6. The error is calculated as $\text{Error}_2 = [(m_d(x;\theta) - \hat{m}(x;\theta))/(m_d(x;\theta))] \times 100$. As the error converges to zero over time, the figure effectively illustrates that the LS-UAVs attain the desired PDF after a certain period. Then, the critic and mass NN performance is demonstrated by choosing a single UAV to illustrate the errors in the HJB and FPK equations. Fig. 7 displays the logarithmic error associated with the HJB equation. This figure illustrates that the HJB error gradually converges to zero over a specific period, implying the optimality of the UAVs concerning the cost function. Similarly, in Fig. 8, we observe the logarithmic error convergence associated with the FPK equation.

V. Conclusion

This article introduced a novel LS-MAS distributed optimization algorithm with a fixed final PDF constraint. This algorithm overcomes the MFG theory limitations, including challenges in maintaining optimality and difficulties in attaining an arbitrary final PDF constraint. The proposed algorithm integrates a novel Imb-MFG theory by decomposing the MFG-PDF using the induction theory and further incorporates a distributed RL algorithm to achieve the optimal solution. Specifically, an induction-based approach for PDF parameter estimation is employed, and a constrained K-means clustering method is used to break down the LS-MAS into different groups to attain the desired final PDF constraint. Furthermore, an RL-based M-ACM learning is designed to achieve the optimal solution of the Imb-MFG. This learning structure is employed to solve HJB-FPK equations, with the critic NN evaluating the optimal cost function, the actor NN calculating the optimal control, and the mass NN assessing the PDF function. To show the efficacy of the developed algorithm, numerical simulations are conducted, accompanied by the Lyapunov stability analysis, highlighting its efficiency and applicability in real-world scenarios.

Appendix A
Proof of Theorem 1

The error term of an agent from group j is defined as $e = x - \mathbb{E}\{m_{d,j}(x,\theta_j)\}$. Here, $\mathbb{E}\{m_{d,j}(x,\theta_j)\}$ represents the mean of the desired PDF. Now, the tracking error dynamic, which is the first derivative of the tracking error [7] is obtained as

$$de(t) = dx(t) - d\mathbb{E}\{m_{d,j}(x,\theta_j)\}. \tag{53}$$

Substituting (1), the tracking error dynamic is obtained as

$$de(t) = [f(x(t)) + g(x(t))u(t)]dt + \sigma d\omega. \tag{54}$$

Note that, the derivative value $d\mathbb{E}\{m_{d,j}(x,\theta_j)\}$ becomes zero since $\mathbb{E}\{m_{d,j}(x,\theta_j)\}$ is a stationary point. Now

$$\begin{aligned} de(t) &= [f(e(t)+\mathbb{E})\{m_{d,j}(x,\theta_j)\}+g(e(t)+\mathbb{E})\{m_{d,j}(x,\theta_j)\} \\ &\quad u(t)]dt+\sigma d\omega \\ &= \big[f_a(e(t))+g_a(e(t))u(t)\big]dt+\sigma d\omega \end{aligned} \tag{55}$$

where $x(t)=e(t)+\mathbb{E}\{m_{d,j}(x,\theta_j)$, $f_a(e)=f(e+\mathbb{E}\{m_{d,j}(x,\theta_j)\})$, and $g_a(e)=g(e+\mathbb{E}\{m_{d,j}(x,\theta_j)\})$.

Appendix B
Proof of Theorem 3

Consider the Lyapunov function candidate as

$$L_J(t)=\frac{1}{2}\mathrm{tr}\{\mathbb{E}\{\tilde{W}_J^T\tilde{W}_J\}\}. \tag{56}$$

The critic NN weight estimation error is obtained from (49)

$$\mathbb{E}\{\dot{\tilde{W}}_{V,i}\}=\mathbb{E}\{-\dot{\hat{W}}_J\}=\mathbb{E}\left\{\alpha_J\frac{\Psi_J(x,\hat{m}_j)e_{\mathrm{HJB}}^T}{1+\|\Psi_J(x,\hat{m}_j)\|^2}\right\}. \tag{57}$$

Based on the Lyapunov stability analysis [24], taking the derivative of (56) with respect to time and substituting (57)

$$\dot{L}_J(t)=\alpha_J\mathrm{tr}\left(\mathbb{E}\left\{\tilde{W}_J^T\frac{\Psi_J(x,\hat{m}_j)e_{\mathrm{HJB}}^T}{1+\|\Psi_J(x,\hat{m}_j)\|^2}\right\}\right). \tag{58}$$

Let $\hat{\Psi}_J=\Psi_J(x,\hat{m}_j)$ and $\tilde{\Psi}_J=\Psi_J(x,\tilde{m}_j)$. Now, substituting (46) into (58), and using the triangular inequality

$$\begin{aligned} \dot{L}_J(t) &\le -\frac{1}{4}\alpha_J\mathbb{E}\left\{\frac{\|\hat{\Psi}_J\|^2\|\tilde{W}_J\|^2}{1+\|\hat{\Psi}_J\|^2}\right\}-\alpha_J\mathbb{E}\left\{\frac{\|\frac{\tilde{W}_J^T\hat{\Psi}_J}{2}-\tilde{\Phi}\|^2}{1+\|\hat{\Psi}_J\|^2}\right\} \\ &-\alpha_j\mathbb{E}\left\{\frac{\|\frac{\tilde{W}_J^T\hat{\Psi}_J}{2}+W_J^T\tilde{\Psi}_J\|^2}{1+\|\hat{\Psi}_J\|^2}\right\}-\alpha_J\mathbb{E}\left\{\frac{\|\frac{\tilde{W}_J^T\hat{\Psi}_J}{2}+\varepsilon_{\mathrm{HJB}}\|^2}{1+\|\hat{\Psi}_J\|^2}\right\} \\ &+\alpha_J\left[\mathbb{E}\left\{\frac{\|\tilde{\Phi}\|^2}{1+\|\hat{\Psi}_J\|^2}\right\}+\mathbb{E}\left\{\frac{\|W_J^T\tilde{\Psi}_J\|^2}{1+\|\hat{\Psi}_J\|^2}\right\}+\mathbb{E}\left\{\frac{\|\varepsilon_{\mathrm{HJB}}\|^2}{1+\|\hat{\Psi}_J\|^2}\right\}\right]. \end{aligned} \tag{59}$$

Equation (59) with dropped negative terms

$$\dot{L}_J(t)\le -\frac{1}{4}\alpha_J\mathbb{E}\left\{\frac{\|\hat{\Psi}_J\|^2\|\tilde{W}_J\|^2}{1+\|\hat{\Psi}_J\|^2}\right\}+B_{W_J}(t) \tag{60}$$

$$\begin{aligned} B_{W_J}(t) &= \frac{\alpha_J}{1+\mathbb{E}\{\|\hat{\Psi}_J\|^2\}}\Big[[l_\Phi+l_{\Psi_J}\mathbb{E}\{\|W_J\|^2\}]\mathbb{E}\{\|\tilde{m}_j\|^2\} \\ &\quad +\|\varepsilon_{\mathrm{HJB}}\|^2\Big] \end{aligned} \tag{61}$$

where l_Φ and l_{Ψ_J} are the respective functions Lipschitz constants and $\tilde{m}_j$ is the PDF approximation error bound. The approximation error of the critic NN weight will be UUB and the bound is

$$\mathbb{E}\{\|\tilde{W}_J\|\}\le 2\mathbb{E}\left\{\sqrt{\frac{\left(1+\|\hat{\Psi}_J\|^2\right)}{\alpha_J\|\hat{\Psi}_J\|^2}B_{W_J}}\right\}\equiv b_{W_J}. \tag{62}$$

Appendix C
Proof of Theorem 4

Consider the Lyapunov function candidate as

$$L_{m_j}(t)=\frac{1}{2}\mathrm{tr}\{\mathbb{E}\left\{\tilde{W}_{m_j}^T\tilde{W}_{m_j}\right\}\}. \tag{63}$$

Using Lyapunov stability analysis [24], taking the derivative of (63) and using the weight estimation error

$$\dot{L}_{m_j}(t)=\alpha_{m_j}\mathrm{tr}\left(\mathbb{E}\left\{\tilde{W}_{m_j}^T\frac{\Psi_{m_j}(x,\hat{J},t)e_{\mathrm{FPK}}^T}{1+\|\Psi_{m_j}(x,\hat{J},t)\|^2}\right\}\right). \tag{64}$$

Substituting (47) into (64) and using the triangular inequality

$$\dot{L}_{m_j}(t)\le -\frac{1}{2}\alpha_{m_j}\mathbb{E}\left\{\frac{\|\hat{\Psi}_{m_j}\|^2\|\tilde{W}_{m_j}\|^2}{1+\|\hat{\Psi}_{m_j}\|^2}\right\}+B_{W_{m_j}} \tag{65}$$

$$\begin{aligned} B_{W_{m_j}} &= \alpha_{m_j}/1+\mathbb{E}\{\|\hat{\Psi}_{m_j}\|^2\}[\left[l_{\Psi_{m_j}}\mathbb{E}\{\|W_{m_j}\|^2\}\right]\mathbb{E}\{\|\tilde{J}\|^2\} \\ &\quad +\mathbb{E}\{\|\varepsilon_{\mathrm{FPK}}\|^2\} \end{aligned} \tag{66}$$

with $l_{\Psi_{m_j}}$ being the Lipschitz constant. The approximation error of the mass NN weight is UUB

$$\mathbb{E}\{\|\tilde{W}_{m_j}\|\}\le\sqrt{2}\mathbb{E}\left\{\sqrt{\frac{\left(1+\|\hat{\Psi}_{m_j}\|^2\right)}{\alpha_{m_j}\|\hat{\Psi}_{m_j}\|^2}B_{W_{m_j}}}\right\}\equiv b_{W_{m_j}}. \tag{67}$$

Appendix D
Proof of Theorem 5

Consider the Lyapunov function candidate as

$$L_u(t)=\frac{1}{2}\mathrm{tr}\{\mathbb{E}\{\tilde{W}_u^T\tilde{W}_u\}\}. \tag{68}$$

Based on the Lyapunov stability analysis [24], taking the derivative of (68) and using the weight estimation error

$$\dot{L}_u(t)=\alpha_u\mathrm{tr}\left(\mathbb{E}\left\{\tilde{W}_u^T\frac{\phi_u(x,\hat{m}_j)e_{\mathrm{u}}^T}{1+\|\phi_u(x,\hat{m}_j)\|^2}\right\}\right). \tag{69}$$

Substituting (40) into (69) and using the triangular inequality

$$\dot{L}_u(t)\le -\frac{1}{4}\alpha_u\mathbb{E}\left\{\frac{\|\hat{\phi}_u\|^2\|\tilde{W}_u\|^2}{1+\|\hat{\phi}_u\|^2}\right\}+B_{W_u} \tag{70}$$

$$B_{W_u}(t)=\mathbb{E}\left[\frac{\alpha_u}{1+\|\hat{\phi}_u\|^2}\left\{\|R^{-1}g^T\|^2\|\tilde{J}\|^2+\|\hat{\phi}_u\|^2\right\}\right]. \tag{71}$$

where $\tilde{J}$ is the critic approximation error bound. The approximation error of the actor NN weight is UUB given as

$$\mathbb{E}\{\|\tilde{W}_u\|\}\le 2\mathbb{E}\left\{\sqrt{\frac{\left(1+\|\hat{\phi}_u\|^2\right)}{\alpha_u\|\hat{\phi}_u\|^2}B_{W_u}}\right\}\equiv b_{W_u}. \tag{72}$$

Appendix E
Proof of Theorem 6

Consider the Lyapunov function as

$$\begin{aligned} L_{\mathrm{sys}}(t) &= \frac{\beta_1}{2}\mathrm{tr}\big(\mathbb{E}\{e^T(t)e(t)\}\big)+\frac{\beta_2}{2}\mathrm{tr}\big(\mathbb{E}\big\{\tilde{W}_J^T(t)\tilde{W}_J(t)\big\}\big) \\ &\quad +\frac{\beta_3}{2}\mathrm{tr}\Big(\mathbb{E}\{\Big\{\tilde{W}_{m_j}^T(t)\tilde{W}_{m_j}(t)\Big\}\Big)+\frac{\beta_4}{2}\mathrm{tr}\big(\mathbb{E}\{\{\tilde{W}_u^T(t)\tilde{W}_u(t)\}\big). \end{aligned} \tag{73}$$

Taking the derivative with respect to time and substituting Lemma 1 and Theorems 1–3 given in (60), (65), and (70) and using the corresponding bound

$$\begin{aligned} \dot{L}_{\mathrm{sys}}(t) &\le -\frac{\gamma\beta_1}{2}\mathbb{E}\{\|e\|^2\}+\frac{2\beta_1 g_l^2}{\gamma}\mathbb{E}\|\tilde{u}\|^2-\frac{\beta_2\alpha_J}{4} \\ &\quad \mathbb{E}\left\{\frac{\|\hat{\Psi}_J\|^2\|\tilde{W}_J\|^2}{1+\|\hat{\Psi}_J\|^2}\right\}-\frac{\beta_3\alpha_{m_j}}{2}\mathbb{E}\left\{\frac{\|\hat{\Psi}_{m_j}\|^2\|\tilde{W}_{m_j}\|^2}{1+\|\hat{\Psi}_{m_j}\|^2}\right\}-\frac{\beta_4\alpha_u}{4} \end{aligned}$$

$$\mathbb{E}\left\{\frac{\|\hat{\phi}_u\|^2\|\tilde{W}_u\|^2}{1+\|\hat{\phi}_u\|^2}\right\}+\beta_2\varepsilon_{\text{NHJB}}+\beta_3\varepsilon_{\text{NFPK}}+\beta_4\varepsilon_{\text{Nu}}+3b_4\mathbb{E}$$
$$\left\{\|\tilde{W}_J\|^2\|\hat{\Psi}_J\|^2\right\}+6b_4l_{\Psi_J}^2\mathbb{E}\{\|W_J\|^2\}\mathbb{E}\left\{\|\tilde{W}_{m_j}\|^2\|\hat{\Psi}_{m_j}\|^2\right\}$$
$$+2b_2\mathbb{E}\left\{\|\tilde{W}_{m_j}\|^2\|\hat{\Psi}_{m_j}\|^2\right\}+6b_4l_{\Psi_J}^2\mathbb{E}\{\|W_J\|^2\}\mathbb{E}$$
$$\{\|\varepsilon_{\text{FPK}}\|^2\}+2b_2\mathbb{E}\{\|\varepsilon_{\text{FPK}}\|^2\}+3b_4\mathbb{E}\{\|\varepsilon_{\text{HJB}}\|^2\}$$
$$\leq-\frac{\gamma\beta_1}{2}\mathbb{E}\{\|e\|^2\}-\kappa_J\mathbb{E}\{\|\tilde{W}_J\|^2\}-\kappa_u\mathbb{E}\{\|\tilde{W}_u\|^2\}$$
$$-\kappa_{m_j}\mathbb{E}\{\|\tilde{W}_{m_j}\|^2\}+\varepsilon_{\text{CS}} \tag{74}$$

$$\text{with, }\ \kappa_J=-\frac{\beta_2\alpha_J}{4}\mathbb{E}\left\{\frac{\|\hat{\Psi}_J\|^2\|\tilde{W}_J\|^2}{1+\|\hat{\Psi}_J\|^2}\}-3b_4\mathbb{E}\{\|\hat{\Psi}_J\|^2\right\}$$

$$\kappa_u=\frac{\beta_4\alpha_u}{4}\mathbb{E}\left\{\frac{\|\hat{\phi}_u\|^2\|\tilde{W}_u\|^2}{1+\|\hat{\phi}_u\|^2}\right\}-\frac{6\beta_1g_l^2}{\gamma}\mathbb{E}\{\|\hat{\phi}_u\|^2\}$$

$$\kappa_{m_j}=\frac{\beta_3\alpha_{m_j}}{2}\mathbb{E}\left\{\frac{\|\hat{\Psi}_{m_j}\|^2\|\tilde{W}_{m_j}\|^2}{1+\|\hat{\Psi}_{m_j}\|^2}\right\}-\frac{6\beta_1g_l^2}{\gamma}l_{\phi_u}^2$$
$$\mathbb{E}\left\{\|W_u\|^2\|\hat{\Psi}_{m_j}\|^2\right\}-6b_4l_{\Psi_J}^2\mathbb{E}\{\|W_J\|^2\|\hat{\Psi}_{m_j}\|^2\}$$
$$-2b_2\mathbb{E}\{\|\hat{\Psi}_{m_j}\|^2\}$$

$$\varepsilon_{\text{CS}}=\frac{6\beta_1g_l^2}{\gamma}l_{\phi_u}^2\mathbb{E}\{\|W_u\|^2\}\mathbb{E}\left\{\|\varepsilon_{\text{FPK}}\|^2\right\}+\frac{6\beta_1g_l^2}{\gamma}\mathbb{E}\{\|\varepsilon_{\text{u}}\|^2\}$$
$$+\beta_2\varepsilon_{\text{NHJB}}+\beta_3\varepsilon_{\text{NFPK}}+\beta_4\varepsilon_{\text{Nu}}+6b_4l_{\Psi_J}^2\mathbb{E}\{\|W_J\|^2\}$$
$$\mathbb{E}\left\{\|\varepsilon_{\text{FPK}}\|^2\right\}+2b_2\mathbb{E}\left\{\|\varepsilon_{\text{FPK}}\|^2\right\}+3b_4\mathbb{E}\left\{\|\varepsilon_{\text{HJB}}\|^2\right\} \tag{75}$$

where $l_{\Psi_{m_j}}$, l_{Ψ_J}, and l_{ϕ_u} are the Lipschitz constants. The derivative of the Lyapunov function $\dot{L}_{\text{sys}}(t)$ is less than zero outside a compact set, which is obtained from (74)

$$\mathbb{E}\{\|e\|\}>\sqrt{\frac{2}{\gamma\beta_1}\varepsilon_{\text{CS}}}\ \text{ or }\ \mathbb{E}\{\|\tilde{W}_J\|\}>\sqrt{\frac{1}{\kappa_J}\varepsilon_{\text{CS}}}$$
$$\mathbb{E}\{\|\tilde{W}_u\|\}>\sqrt{\frac{1}{\kappa_u}\varepsilon_{\text{CS}}}\ \text{ or }\ \mathbb{E}\{\|\tilde{W}_{m_j}\|\}>\sqrt{\frac{1}{\kappa_{m_j}}\varepsilon_{\text{CS}}}.$$

REFERENCES

[1] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.

[2] M. Balmer, K. Nagel, and B. Raney, "Large-scale multi-agent simulations for transportation applications," *J. Intell. Transp. Syst.*, vol. 8, no. 4, pp. 205–221, 2004.

[3] W. Wang, L. Wang, J. Wu, X. Tao, and H. Wu, "Oracle-guided deep reinforcement learning for large-scale multi-UAVs flocking and navigation," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10280–10292, Oct. 2022.

[4] F.-L. Lian, J. Moyne, and D. Tilbury, "Network design consideration for distributed control systems," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 2, pp. 297–307, Mar. 2002.

[5] S. Parsons and M. Wooldridge, "Game theory and decision theory in multi-agent systems," *Auton. Agents Multi-Agent Syst.*, vol. 5, pp. 243–254, Sep. 2002.

[6] E. Semsar-Kazerooni and K. Khorasani, "Multi-agent team cooperation: A game theory approach," *Automatica*, vol. 45, no. 10, pp. 2205–2213, 2009.

[7] Z. Zhou and H. Xu, "Large-scale multiagent system tracking control using mean field games," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5602–5610, Oct. 2022.

[8] S. Dey and H. Xu, "Hierarchical game theoretical distributed adaptive control for large scale multi-group multi-agent system," *IET Control Theory Appl.*, vol. 17, no. 17, pp. 2332-2352, 2023.

[9] J.-M. Lasry and P.-L. Lions, "Mean field games," *Jpn. J. Math.*, vol. 2, no. 1, pp. 229–260, 2007.

[10] P. E. Caines, "Mean field games," in *Encyclopedia of Systems and Control*. New York, NY, USA: Springer, 2021, pp. 1197–1202.

[11] Z. Zhou and H. Xu, "Decentralized optimal large scale multi-player pursuit-evasion strategies: A mean field game approach with reinforcement learning," *Neurocomputing*, vol. 484, pp. 46–58, May 2022.

[12] M. A. Wiering and M. Van Otterlo, "Reinforcement learning," *Adapt., Learn., Optim.*, vol. 12, no. 3, p. 729, 2012.

[13] P. S. Bradley, K. P. Bennett, and A. Demiriz, "Constrained k-means clustering," *Microsoft Res., Redmond*, vol. 20, pp. 1–9, May 2000.

[14] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.)*, vol. 32, no. 2, pp. 140–153, May 2002.

[15] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[16] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*. Hoboken, NJ, USA: Wiley, 2011.

[17] J. Seiffertt, S. Sanyal, and D. C. Wunsch, "Hamilton–Jacobi–Bellman equations and approximate dynamic programming on time scales," *IEEE Trans. Syst., Man, Cybern., Part B (Cybern.)*, vol. 38, no. 4, pp. 918–923, Aug. 2008.

[18] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[19] A. Baker, "Mathematical induction and explanation," *Analysis*, vol. 70, no. 4, pp. 681–689, 2010.

[20] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. North Chelmsford, MA, USA: Courier Corp., 2012.

[21] A. Sahoo, H. Xu, and S. Jagannathan, "Approximate optimal control of affine nonlinear continuous-time systems using event-sampled neurodynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 639–652, Mar. 2017.

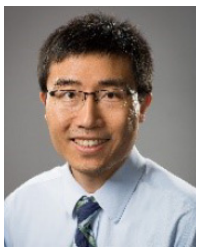
[22] H. Xu and S. Jagannathan, "Stochastic optimal controller design for uncertain nonlinear networked control system via neuro dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 471–484, Mar. 2013.

[23] Z. Zhou and H. Xu, "Mean field game and decentralized intelligent adaptive pursuit evasion strategy for massive multi-agent system under uncertain environment," in *Proc. Amer. Control Conf. (ACC)*, 2020, pp. 5382–5387.

[24] H. K. Khalil, "Lyapunov stability," *Control Syst., Robot. Autom.*, vol. 12, p. 115, Oct. 2009.

Shawon Dey received the M.Sc. degree in electrical engineering from South Dakota Mines, Rapid City, SD, USA, in 2020. He is currently pursuing the Ph.D. degree with the Electrical and Biomedical Engineering Department, University of Nevada, Reno, NV, USA.

His research interests include adaptive control, artificial intelligence, and game theory.



Hao Xu (Member, IEEE) received the master's degree in electrical engineering from Southeast University, Nanjing, China, in 2009, and the Ph.D. degree in electrical engineering from the Missouri University of Science and Technology, Rolla, MO, USA, in 2012.

He is currently an Associate Professor with the Department of Electrical and Biomedical Engineering, University of Nevada, Reno, NV, USA. His current research interests include game theory, large-scale multiagent systems, optimization, trusted AI, and autonomous unmanned aircraft systems.