Abstract—This paper investigates the control of a massive population of UAVs such as drones. The straightforward method of control of UAVs by considering the interactions among them to make a flock requires a huge inter-UAV communication which is impossible to implement in real-time applications. One method of control is to apply the mean field game (MFG) framework which substantially reduces communications among the UAVs. However, to realize this framework, powerful processors are required to obtain the control laws at different UAVs. This requirement limits the usage of the MFG framework for real-time applications such as massive UAV control. Thus, a function approximator based on neural networks (NN) is utilized to approximate the solutions of Hamilton-Jacobi-Bellman (HJB) and Fokker-Planck-Kolmogorov (FPK) equations. Nevertheless, using an approximate solution can violate the conditions for convergence of the MFG framework. Therefore, the federated learning (FL) approach which can share the model parameters of NNs at drones, is proposed with NN based MFG to satisfy the required conditions. The stability analysis of the NN based MFG approach is presented and the performance of the proposed FL-MFG is elaborated by the simulations.

Index Terms—Autonomous UAV, communication-efficient online path control, mean-field game, federated learning.

I. INTRODUCTION

Real-time control of a large number of unmanned aerial vehicles (UAVs) is instrumental in enabling mission-critical applications, such as covering wide disaster sites in emergency cell networks [1], search-and-rescue missions to deliver first-aid packets, and firefighting scenarios [2], [3]. One key challenge is inter-UAV collision, notably under random wind dynamics [1], [4]. A straight forward solution is to exchange instantaneous UAV locations, incurring huge communication overhead, which is thus unfit for real-time operations. Alternatively, in this article we propose a novel real-time massive UAV control framework leveraging mean-field game (MFG) theory [5–7] and federated learning (FL) [8], [9].

In our proposed FL-MFG control method, each UAV determines its optimal control decision (e.g., acceleration) not by exchanging UAV states (e.g., position and velocity), but by locally estimating the entire UAV population’s state distribution, hereafter referred to as MF distribution. According to MFG [6], such a distributed control decision asymptotically achieves the epsilon-Nash equilibrium as the number of UAVs goes to infinity. To implement this, the UAV needs to solve a pair of coupled stochastic differential equations (SDEs), namely, the Fokker-Planck-Kolmogorov (FPK) and Hamilton-Jacobi-Bellman (HJB) equations, for the population distribution estimation and optimal control decision, respectively. The complexity of solving FPK and HJB increases with the state dimension, creating another bottleneck in real-time applications.

To resolve this complexity issue, FL-MFG control utilizes neural-network (NN) based approximations [4], [10], [11] and FL [8]. Specifically, instead of solving HJB and FPK equations, every UAV runs a pair of two NNs, HJB NN and FPK NN whose outputs approximate the solutions of HJB and FPK equations, respectively. The approximation accuracy increases with the number of UAV state observations, i.e., NN training samples. To accelerate the NN training speed, by leveraging FL, each UAV periodically exchanges the HJB NN and FPK NN model parameters with other UAVs, thereby reflecting the locally non-observable training samples. In a source-destination UAV dispatching scenario shown in Fig. 1, simulation results corroborate that FL-MFG control achieves up to 50% shorter travel time, 25% less motion energy, 75% less total transmitted bits, and better collision avoidance measured by 50% lower collision probability number, compared to baseline schemes: FL-MFG exchanging only either HJB NN or FPK NN model parameters, and a control scheme running only HJB NN while exchanging state observations.

A. Background and Related Works

There are many ongoing works on the control and applications of the UAVs. For many missions and tasks such as agriculture, and search and rescue, the working hours, labor requirements, and cost can be reduced significantly, and the efficiency of the work can also be improved by utilizing the UAVs [12], [13]. However, many current research works of UAV such as [14–21] are on single UAV with autonomous or manual control, and multiple UAV system
or mass of UAVs are at the early stages of research [22–24]. In [22], an autonomous system is developed to perform inspections for precision agriculture based on the use of single and multiple UAVs. Furthermore, [23] the collective motion of large flocks of autonomous UAVs in order to navigate in confined spaces is investigated, and it is shown that the swarm behavior remained stable under realistic conditions for large flocks and high velocities. In [24] the performance of a multiple UAV system by using the distributed swarm control algorithm is evaluated and analyzed through four experimental cases: single UAV with autonomous control, multiple UAVs with autonomous control, single UAV with remote control, and multiple UAVs with remote control and it is shown that performance of the multiple UAV system is better than the single UAV system for agricultural application.

Moreover, there are several works considering the flocking behavior of the agents. In [1] an instantaneous movement control for massive UAVs to provide a cellular connection in a disaster region is investigated. [25] discusses a linear analysis in the synthesis of Cucker-Smale (C-S) type flocking via MF stochastic control theory. It is shown that the C-S flocking behavior may be obtained as a Nash dynamic competitive game equilibrium. In [23] a flocking model for real drones with an evolutionary optimization framework is proposed, and shown that swarm behavior remained stable under realistic conditions for a large population at even high velocities. These works have the common challenge of controlling multiple agents affected randomly by the environment.

MFG framework has been developed mainly in [5–7] to study the behavior of large population systems that play non-cooperative games under exchangeability assumptions. However, the main challenge in this framework is to solve the partial differential equations (PDEs) with acceptable accuracy and speed for a specific application. There are many numerical methods to solve PDEs as in [26–29]. However, new methods are arising to obtain the solution for PDE with more accuracy or speed. The (deep) reinforcement learning methods are developed to learn the solution of the HJB equation and control rule in [30–32]. In [33] a numerical approximation method is proposed to approximate the Kolmogorov PDE without suffering from the curse of conditionality. In [10] a neural-network-based online solution of the HJB equation is used to explore the infinite horizon optimal robust guaranteed cost control of uncertain nonlinear systems. However, most methods to solve the HJB and FPK are computationally expensive, and they are not proper for real-time applications.

Most works in UAV controlling rely on the knowledge of the measurements and the environment [34–36]. However, obtaining the exact measurements about the environment in realistic real-time applications is not possible. However, for uncertain or unknown environments the reinforcement learning models [14], [37], Bayesian method [38] or models based on MFG [1], [25] can be promising alternatives since they can adapt themselves with the dynamic environment.

To the best of our knowledge, yet there are not many works considering the massive number of UAVs scenario in a windy environment in an online manner all together for real-time applications. We think that solving this problem can pave the way for new applications of UAVs.

Following [25], in general, two models are used to handle the massive UAV flocking problem. First, the direct control model where a stochastic differential equation (SDE) at each UAV should be solved to obtain the control inputs. Second, the MFG theoretic method, where a pair of the partial differential equation is required to be solved at the UAVs. However, these two models have both advantages and drawbacks for different applications.

- In the direct control model, the agents are coupled by a non-linear term in their cost functions. This model requires that all the agents have exact information about the states of all other agents in order to solve their corresponding SDEs. Since this method is base on the exact observations of the other agents, it can result in optimal solutions. Nevertheless, the application of this method is limited to the cases with a small number of agents because the number of communications increases (exponentially) with the number of agents. Therefore, the direct SDE method cannot be utilized when the number of agents is large.

- The mean-field (MF) stochastic control theory can be utilized to approximate the large population. In this model, all the agents have similar dynamics and are coupled by a non-linear flocking term in their cost functions. However, this control method requires higher processing at the agents than the direct control method and it should satisfy the convergence conditions. Following [39], MFG theory, as a branch of game theory, is a suitable mathematical tool to help agents of a large population to take proper decisions in the context of strategic interactions. When the number of agents  in an multi-agent system increases, the number of interactions also increases exponentially and this makes every exact model for the agents impossible to be implemented. This problem can be simplified when the agents of the system are interchangeable in the sense that nothing is dependent on an individual agent. However, utilizing the MFG framework is quite challenging.

- One main challenge for the MFG framework is that unless for special cases obtaining an analytic solution for HJB and FPK differential equations is impossible. Therefore, there is a need for approximate numerical solutions that require high computations power. There are many numerical methods such as [27], [28] which can be used to solve HJB and FPK equations. In addition, deep neural network (DNN) methods are also proposed recently to solve HJB equations. However, these methods usually require high processing and might not be suitable
for real-time applications with low processing capability agents with limited energy.

- Another important challenge in the MFG framework is the interchangeability condition for the agents. This means that all the agents play with similar rules. However, if the algorithm used to solve the HJB and FPK equations cannot obtain the optimal solution the agents might have different control rules and this will violate the interchangeability condition.

The proposed method in this paper is to address these challenges.

B. Contributions and Organization

In this paper, we study the real-time control of a large population of UAVs in a windy environment. We start this by explaining the scenario of multiple UAVs to be moved from a starting region to a destination region as shown in Fig.1. Next, two control methods are described in detail, i.e., the HJB control method and the MFG control method to dispatch multiple UAVs quickly, safely, and with low energy consumption. Controlling the UAVs by the HJB control method requires that the UAVs know the instantaneous location of all the UAVs to achieve the mentioned objective. To reduce the communications cost of HJB, the MFG is a useful alternative. However, obtaining the optimal solution for MFG entails solving a pair of HJB and FPK equation which is computationally costly, which limits MFG’s usage for real-time UAV control. The main contribution of this paper are summarized as follows:

- In order to reduce the computational cost of HJB control and MFG control, an NN-based function approximator is utilized to approximate the solution of HJB and FPK equations in an adaptive way. The method used here is a variant of our previous work in [4], where one single-layer network with two outputs is used to estimate the solution of each HJB and FPK equation. This method gives an approximate solution to the HJB and MFG framework.

- In order to validate the feasibility of the proposed method, the Lyapunov stability analysis is used for HJB and FPK approximation error. These analyses show that the error of approximate solutions for HJB and FPK is bounded, which means that the obtained approximate control actions from NNs are an approximation of the optimal control actions. However, one main requirement for the stability of the approximate solution of MFG is that enough data samples should be provided to update NN’s weights, which is challenging in real-time applications.

- To make sure the UAVs do not lack data samples and to mitigate the communication costs, and stability concerns of MFG, an FL-based MFG strategy is proposed, which will be named as MfgFL-HF control in this paper. The performance and stability of MfgFL-HF are verified by the simulations. It will be shown that adopting FL can yield faster, safer and energy-efficient control over the baseline methods.

The remainder of this paper is organized as follows. Section II describes the system model for controlling the population of UAVs. Section III explains the HJB and MFG control methods and the necessity to propose an alternative method. Section IV proposes the online NN-base method to obtain an approximate solution for HJB and FPK equations, with their stability analysis brought in the appendices. Section V proposes FL-based MFG methods in detail. Section VI validates the performance of the proposed method by simulations, followed by our conclusions in Section VII.

II. System Model

Consider the scenario of Fig.1 where a set $\mathcal{N}$ of $N$ UAVs are set to go from a starting position to a specified destination in a windy environment. There are three major issues in this problem as: A) Dynamics of the control system, which reflects the relationship between the parameters of the system, and also effect of the environment in the system. The more information we have about the environment, the better model we can utilize for control. Here we will assume the wind perturbations as the main source of randomness in the system. B) Control problem, which will consider the costs and interests to formulate a problem where its solution can control the UAVs to the destinations. Here, one major assumption for control is that the number of UAVs, i.e., $N$, is large. When the number of UAVs is getting larger, the complexity and the risk of the problem increases consequently, especially in the real-time application with expensive UAVs such as UAVs. C) Channel, in a multi-UAV control, the communication among the UAVs is of critical importance to achieve the control objectives. Here, following the explanations in Introduction, we only will consider inter-UAV channels, which is modeled as Rician in [40]. In the following we will consider these challenges to address the objective of the paper. However, the more focus will be on control with more details on the next sections.

A. Dynamics of Control System

In order to solve the UAV control problem we should obtain the relationships among its location, speed, acceleration, and effect of wind on them on the coordinate system. Then Let us use a Cartesian coordinate system with the origin at the target position as the global reference coordinate. We define $\mathbf{r}_i(t) \in \mathbb{R}^2$ as the vector from the target destination to the current position of $i$-th UAV $u_i$ at time $t \geq 0$. Therefore, the objective of each $u_i$, $1 \leq i \leq N$, is to gradually reduce the distance between destination point and the $u_i$’s current position, by tuning its speed $\mathbf{v}_i(t) \in \mathbb{R}^2$ by controlling the acceleration $a_i(t) \in \mathbb{R}^2$ under random wind dynamics. Following [41], the wind dynamics are assumed to follow an Ornstein-Uhlenbeck process with an average wind velocity $\mathbf{v}_o$. The temporal state dynamics are thereby given as:

$$\frac{d\mathbf{v}_i(t)}{dt} = a_i(t)dt - c_0(\mathbf{v}_i(t) - \mathbf{v}_o)dt + \mathbf{v}_o dW_i(t) \quad (1a)$$

$$dr_i(t) = v_i(t)dt, \quad (1b)$$
TABLE I
LIST OF SIMPLIFIED NOTATIONS.

| Notation | Simplified Definition | Meaning |
|----------|----------------------|---------|
| \(\psi(s_i, \tau; m(s_i, \tau))\) | \(\psi\) | Value function |
| \(\phi_L(s_i)\) | | Local term |
| \(\phi_G(s_i; m(s_i, \tau))\) | \(\phi_G\) | Global interaction term |
| \(f(s_i)\) | \(f\) | Drift function |
| \(\nabla s_i\) | \(\nabla\) | Divergence with respect to |
| \(\Delta_s\) | \(\Delta\) | Laplacian with respect to |
| \(\nabla_s\) | \(\nabla\) | Gradient with respect to |
| \(H(s_i, \tau; m(s_i, \tau))\) | \(H\) | HJB equation |
| \(F(s, \tau; a(t))\) | \(F\) | FPK equation |
| \(m(s, \tau)\) | \(m\) | Distribution of states at time \(t\) |
| \(u_{H_0}(t)\) | \(u_{H_0}\) | HJB model weights, \(d \in \{0, 1\}\) |
| \(\sigma_H(s_i; m(s_i, \tau))\) | \(\sigma_H\) | HJB activation vector function |
| \(\sigma_G(s_i; s_{-i}(\tau))\) | \(\sigma_G\) | FPK activation vector function |
| \(e_H(s_i)\) | \(e_H\) | Error of approximating HJB |
| \(e_F(s_i)\) | \(e_F\) | Error of approximating FPK |
| \(J_{H_0}(u_{H_0}, u_{H_1})\) | \(J_H\) | Loss function of HJB approximation |
| \(J_F(u_{F_0}, u_{F_1})\) | \(J_F\) | Loss function of FPK approximation |
| \(a_i(t)\) | \(a_i\) | Acceleration or control command |
| \(e_{H}(s_i, \tau)\) | \(e_{H}\) | Error of HJB with optimal NN |
| \(\varepsilon_F(s_i)\) | \(\varepsilon_F\) | Error of FPK with optimal NN |
| \(L_1(s_i(\tau))\) | \(L_1\) | A Lyapunov candidate function |
| \(R_1(t)\) | \(R_1\) | Regularizer term |

where \(c_0\) is a positive constant, \(V_0 \in \mathbb{R}^{2 \times 2}\) is the covariance matrix of the wind velocity, and \(W_i(t) \in \mathbb{R}^2\) is the standard Wiener process independently and identically distributed (i.i.d.) across UAVs.

Now in order to write the dynamics of the controlled system \(\{1a\} \sim \{1b\}\) in a compact form, let us define the state of each \(u_i\) as \(s_i(t) = [r_i(t)^T, v_i(t)^T]^T \in \mathbb{R}^4\); so the SDEs \(\{1a\} \sim \{1b\}\) can be rewritten as

\[
ds_i(t) = (A s_i(t) + B (a_i(t) + c_0 v_0)) dt + G dW_i(t),
\]

(2)

where \(A = \begin{pmatrix} 0 & I \\ 0 & 0 \end{pmatrix}\), \(B = \begin{pmatrix} 0 \\ I \end{pmatrix}\), \(G = \begin{pmatrix} 0 \\ V_0 \end{pmatrix}\), and \(I\) denotes the two-dimensional identity matrix. Furthermore, by defining \(f(s_i(t)) = A s_i(t) + c_0 B v_0\) we rewrite the equation (2), in a compact form as:

\[
ds_i(t) = (f(s_i(t)) + B a_i(t)) dt + G dW_i(t),
\]

(3)

B. Control Problem

In general, a model to solve the mentioned control problem should consider three high-level interests for the UAVs. **First, travel time minimization:** each UAV \(i\) should increase velocity in the direction to the destination point, to reduce the remaining distance to the destination point while considering to limit the total velocity of the UAV. **Second, motion energy:** each UAV \(i\) should reduce the (motion) energy consumption since the UAVs flight time depends on its battery capacity. **Lastly, collision avoidance:** the collective interest of the whole population is to make a flock of the UAVs traveling together to avoid UAVs colliding each other and also to complete the mission quickly. Nevertheless, there is a trade-off among the interests which should be considered in the control problem.

To achieve the aforementioned interests, UAV \(u_i\) at time \(t < T\) aims to minimize its average cost \(\psi^{u_i}_s(s_i(t); s_{-i}(t))\), where \(s_i(t) = s_i, s_{-i}(t) = s_{-i}\) are the state of UAV \(u_i\) and the set of states of all UAVs excluding UAV \(u_i\) at time \(t\), respectively, and the average is taken with respect to the measure induced by control law \(a_i\) for \(\tau \in [t, T]\). The cost \(\psi^{u_i}_s(s_i(t); s_{-i}(t))\) consists of the term \(g(a_i(\tau), s_i(\tau); s_{-i}(\tau))\) depending only on the local state \(s_i(\tau)\) and the control action \(a_i(\tau)\) with given states of other UAV’s as \(s_{-i}(\tau)\).

\[
\psi^{a_s}(s_i, t; s_{-i}) = \mathbb{E} \left[ \int_t^T g(a_i(\tau), s_i(\tau); s_{-i}(\tau)) d\tau \right]
\]

(4)

where \(\mathbb{E}\) is the expectation operator, and \(g(a_i(\tau), s_i(\tau); s_{-i}(\tau))\) is

\[
g(a_i(\tau), s_i(\tau); s_{-i}(\tau)) = \phi_L(s_i(\tau)) + c_0 ||a_i(\tau)||^2 + c_2 \phi_G(s_i(\tau); s_{-i}(\tau))
\]

(5)

in which, the term \(\phi_L(s_i(\tau))\) depending only on the local state \(s_i(\tau)\) and the term \(\phi_G(s_i(\tau); s_{-i}(\tau))\) relying on the global state \(\{s_i(t), s_{-i}(t)\}\) are given as:

\[
\phi_L(s_i(\tau)) = \frac{v_i(\tau) \cdot r_i(\tau)}{||r_i(\tau)||} + c_1 ||v_i(\tau)||^2,
\]

(6)

\[
\phi_G(s_i(\tau); s_{-i}(\tau)) = \frac{1}{N} \sum_{u_j \in \mathcal{N}} \frac{||v_j(\tau) - v_i(\tau)||^2}{(e + ||r_j(\tau) - r_i(\tau)||)^2},
\]

(7)

and the terms \(c_1, c_2, c_3, \beta, \varepsilon\) are positive constants.

The local term \(\phi_L(s_i(\tau))\) and the second term in (5) focus on the the two objectives, i.e. travel time and motion energy minimization. It is intended to minimize the remaining travel distance \(||r_i(\tau)||\) by maximizing the velocity towards the destination, i.e., minimizing the projected velocity \(v_i(\tau) \cdot r_i(\tau)/||r_i(\tau)||\) towards the opposite direction to the destination. Also, it is planned to minimize the kinetic energy and the acceleration control energy that are proportional to \(||v_i(\tau)||^2\) and \(||a_i(\tau)||^2\), respectively [21], [25]. Then, the motion energy \(E(t)\) for each UAV \(i\) at time \(t\), is defined as

\[
E(t) = \int_{\tau=0}^t (c_2 ||v_i(\tau)|| + c_3 ||a_i(\tau)||) d\tau,
\]

(8)

which is used as a metric to compare different algorithms in this paper.

The global term \(\phi_G(s_i(\tau); s_{-i}(\tau))\) in (5) refers to collision avoidance, and is intended to form a flock of UAvs moving together [42]. The flocking leads to small relative inter-UAV velocities for avoiding collision even when their controlled velocities are slightly perturbed by wind dynamics. A collision happens when the inter-UAV distance is less than a defined distance \(r_{coll}\). Furthermore, the flocking yields closer inter-UAV distances without collision. This is beneficial for allowing more UAvs to exchange their states through better channel quality, thereby contributing also to collision avoidance. The formation of a flock as mentioned in [43] is a result of three components: a) separation, i.e. steer to avoid crowding; b) alignment, i.e. steer toward the average heading of neighbors; c) cohesion, i.e. steer toward the average position of neighbors. In view of this, we adopt the Cucker-Smale...
flocking [1, 22] that reduces the relative velocities for the
UAVs. The relative velocity $\|v_j(t) - v_i(t)\|$ and the inter-UAV
distance $\|r_j(t) - r_i(t)\|$ are thus incorporated in the numerator
and denominator of $\phi_C(s_i(t); s_{-i}(t))$, respectively. In
addition, inspired by [23], the velocity alignment $\phi(t)$
and number of collision risks $\phi_C(t)$ as metrics to compare
different algorithms, are defined as
\begin{equation}
\phi(t) = \int_0^t \sum_{u_i \in N} \sum_{u_j \in N} \|v_j(t) - v_i(t)\| \, dt,
\end{equation}
\begin{equation}
\phi_C(t) = \int_0^t \sum_{u_i \in N} \sum_{u_j \in N} \left[ \frac{1}{r_j(t) - r_i(t)} \right] \leq r_c \, dt,
\end{equation}
where the hazard radius $r_c$ defines a dangerous potential
collision zone around the UAV. Lower values of $\phi(t)$ mean
that the amplitude of velocity difference between UAVs is
small and hence they have made a better flock to travel
together. Lower values of $\phi_C(t)$ mean that the UAVs do not
tend to be too close to each other and the risk of them colliding
each other is smaller.

Incorporating the cost (4) under the temporal dynamics (3),
the control problem of UAV $u_i$ at time $t$ is formulated as:
\begin{equation}
\psi(s_i, t; s_{-i}) = \min_{a_i} \psi_{a_i}(s_i, t; s_{-i}) \tag{11}
\end{equation}
s.t. $d_s(t) = (f(s(t)) + B a_i(t)) dt + G dW_i(t), \tag{12}$
The minimum cost $\psi(s_i, t; s_{-i})$ is referred to as the value function
of the optimal control, and should be derived to obtain
the optimal action $a_i(t)$ for UAV $u_i$. The methods to encounter
this problem will be introduced in the following sections.

C. UAV to UAV Wireless Channel Model

In many multi-UAV control problems, communication
among the UAVs is a critical condition. However, in the
problem of this paper the UAVs will be required to communicate
their data with each other while they are moving at a height $h$.
Following [24] for the UAV to UAV communication channels,
the Rice model can be used to model both dominant LOS and
NLOS paths. The Rice distribution is given by:
\begin{equation}
p_r(\zeta) = \frac{\zeta}{\chi^2} \exp \left( -\frac{\zeta^2 - \zeta^2}{2\chi^2} \right), \tag{13}
\end{equation}
where $\zeta \geq 0$, and $\chi$ and $\zeta$ are the strength of LOS and NLOS
paths respectively. Assuming the frequency division multiple
access (FDMA) is used for each UAV to UAV communication,
with the transmission power $P_o$, and the distance $r_d$ from a
UAV to another UAV, the received signal-to-noise (SNR)
at each time is:
\begin{equation}
\text{SNR} = \frac{P_o z r_d^{-\alpha}}{W_o \sigma_n}, \tag{14}
\end{equation}
where, $\sigma_n$ is the noise power, $W_o$ is the bandwidth, $\alpha \geq 2$ is
the path loss exponent, and $z$ is the Rice random variable
defined in (13) and is assumed to be independent and identically
distributed (i.i.d) across the different UAVs and times.

Another parameter which we will use in this paper is the
communication latency. A signal is successfully decoded if the
SNR at time $t$ is greater than a target SNR $\eta$, i.e., $\text{SNR}(t) \geq \eta$.
The number of bits $b_i(D_0)$, transmitted during $D_0$ time slots,
is given as:
\begin{equation}
b_i(D_0) = \theta \sum_{t=1}^{D_0} \mathbb{1}_{\text{SNR}(t) \geq \eta} W_o \log_2 (1 + \eta), \tag{15}
\end{equation}
where $\theta$ is the channel coherence time. The latency of trans-
mittting $b$ bits is the minimum $D_0$, i.e. $D_m$ that satisfies
$b_i(D_0) \geq b$. A latency outage occurs when $D_m$ is greater than
a predefined threshold $D_M$, i.e., $D_m > D_M$ in an algorithm.

III. HJB AND MFG CONTROL

At each time instant, each UAV seeks a solution for the
defined objective function. The first intuitive method is to
analyze the problem directly as solving an HJB equation and
then see if there is a need for other alternatives and also have
the basis for the proposed methods. However, in the following,
it will be clarified that when the number of UAVs is high,
the communications and processing complexity will increase
to the extent that the real-time implementation will not be
possible. Therefore, an alternative MFG method which can
reduce the number of communications significantly will be
explained.

A. HJB Control

In this method, we assume that all UAVs perform their
optimal action based on the last observed states of all other
UAVs and not their instantaneous action and states. Therefore,
the HJB equation with this method is obtained as
\begin{align*}
\hat{\psi}(s_i, t; s_{-i}) &= \min_{a_i} \{ g(a_i(t), s_i; s_{-i}) \\
&+ [\nabla s_i \hat{\psi}(s_i, t; s_{-i})]^\top (f(s_i) + B a_i(t)) \\
&+ \frac{1}{2} \text{tr}(GG^\top [\Delta s_i \hat{\psi}(s_i, t; s_{-i})]) \} = 0, \tag{16}
\end{align*}
where $\nabla$ and $\Delta$ denote the gradient and Laplacian operators,
respectively. The optimal action at UAV $i$ is obtained as
\begin{equation}
a_i(t) = -\frac{1}{2c_3} B^\top \nabla_{s_i} \hat{\psi}(s_i, t; s_{-i}). \tag{17}
\end{equation}

Therefore, substituting (17) in (16) yields in HJB equation
\begin{align*}
\hat{\psi}(s_i, t; s_{-i}) &+ \phi_L(s_i) + c_2 \phi_C(s_i; s_{-i}) \\
&+ \left( f(s_i) - \frac{1}{4c_3} BB^\top \nabla_{s_i} \psi(s_i, t; s_{-i}) \right)^\top \nabla_{s_i} \psi(s_i, t; s_{-i}) \\
&+ \frac{1}{2} \text{tr}(GG^\top [\Delta s_i \psi(s_i, t; s_{-i})]) = 0. \tag{18}
\end{align*}
Solving this HJB equation requires an enormous exchange of
states among the UAVs, which becomes impossible when the
number of UAVs, i.e. $N$, is high. In order to address this
challenge, we leverage the capabilities of the MFG framework
explained in the next subsection. Therefore, the UAVs only
will need to exchange the states only at the beginning of the
mission, and after that, they will calculate the optimal actions
based on their own state.
B. MFG Control

Another method to encounter this problem is to use MFG framework when the number of UAVs is very high. Let \(m_N(s, t)\) be the empirical state distribution function of the UAVs at time instant \(t\) defined as
\[
m_N(s, t) \triangleq \frac{1}{N} \sum_{j=1}^{N} \delta(s - s_j(t)),
\]
where \(\delta(\cdot)\) is the Dirac delta function. Then, the interaction term \((17)\) can be rewritten as
\[
\phi_G(s_i(t); s_{-i}(t)) = \int m_N(s, t) \frac{||v_i(t) - v_j(t)||^2}{(\varepsilon + ||r_i(t) - r_j(t)||)^2} ds.
\]

Since the states \(s_i(t)\) for \(i = 1, \ldots, N\) are independent and identically distributed which evolve according to SDE \((12)\), utilizing the ergodic theory gives
\[
\lim_{N \to \infty} m_N(s, t) = m(s, t),
\]
where \(m(s, t)\) is the distribution of generic UAV’s state, i.e., s, according to the SDE \((12)\) and the optimal policy \(a(t)\) obtained by \((17)\). The distribution \(m(s, t)\), which is called mean field (MF), is a solution of the Fokker-Plank-Kolmogorov (FPK) equation which is derived from the SDE \((12)\) as
\[
m(s, t) + \nabla_s \cdot [(f(s) + Ba(t))m(s, t)] + \frac{1}{2} tr(GG^T \Delta_m m(s, t)) = 0,
\]
where \(\nabla_s \cdot\) denotes the divergence operator, and the initial distribution of the UAVs is given as \(m(s, 0) = \frac{1}{N} \sum_{j=1}^{N} \delta(s - s_j(0))\). Hence, the solution to equation \((22)\), i.e., \(m(s, t)\), can be used to obtain the interaction term as
\[
\phi_G(s_i(t); m(s, t)) \triangleq \int m(s, t) \frac{||v_i(t) - v_j(t)||^2}{(\varepsilon + ||r_i(t) - r_j(t)||)^2} ds
\]

By this definition, the HJB equation \((13)\) can be rewritten as
\[
\dot{\psi}(s_i; t; m(s, t)) + \phi_L(s_i) + c_2 \phi_G(s_i; m(s, t)) + \left(f(s_i) - \frac{1}{4c_3} BB^T \nabla_s \psi(s_i; t; m(s, t))\right)^T \nabla_s \psi(s_i; t; m(s, t)) + \frac{1}{2} tr(GG^T[\nabla_s \psi(s_i; t; m(s, t))] = 0
\]
and the corresponding action is
\[
a_i(t) = -\frac{1}{2c_3} BB^T \nabla_s \psi(s_i; t; m(s, t)),
\]
Therefore, by substituting \((25)\) in \((22)\) the FPK equation can also be rewritten in the following form:
\[
m(s, t) + \nabla_s \cdot [(f(s) - \frac{1}{2c_3} BB^T \nabla_s \psi(s; t; m(s, t)))m(s, t)] + \frac{1}{2} tr(GG^T \Delta_m m(s, t)) = 0
\]
By solving HJB and FPK equation pairs, i.e., \((24)\) and \((26)\), the optimal action for each UAV can be calculated.

IV. NN-BASED HJB AND MFG LEARNING CONTROL - STATE SHARING METHODS

In this section, inspired by \([45]\), we discuss NN-based methods to obtain approximate solutions for HJB and FPK equations for the multiple UAV control application. However, to have better readability and also to save space, we use the simplifications of Table I, and whenever required we will use the complete forms to avoid confusion.

A. HJB Learning Control

Here, following our previous work \([REF]\), we find an approximate solution to the HJB equation to obtain the corresponding action. Any approximate solution will result in some error, and the approximated HJB equation may not be exactly equal to zero. However, first, based on the defined simplifications above, we represent the HJB equation \((18)\) and \((24)\) by \(H\) as
\[
H \triangleq \dot{\psi} + \left(f - \frac{1}{4c_3} BB^T \nabla \psi\right)^T \nabla \psi + \phi_L + c_2 \phi_G + \frac{1}{2} tr(GG^T \Delta \psi) = 0,
\]
where we obtain the \(\phi_G\) empirically by \((17)\) in this subsection.

Similar to \([45]\), given the state distribution of UAVs at each time \(t\), let the function \(\psi(s_i; t; s_{-i})\) and its derivative correspondingly be approximated by functions as
\[
\hat{\psi}(s_i; t; s_{-i}) \triangleq \hat{w}_{h_k}(t)^T \sigma_i(s_i; s_{-i}),
\]
\[
\dot{\psi}(s_i; t; s_{-i}) \triangleq \hat{w}_{h_k}(t)^T \sigma_i(s_i; s_{-i}),
\]
where vector functions \(\hat{w}_{h_k}(t)\) and \(\hat{w}_{h_k}(t)\) are approximations to the optimal vector weight functions \(w_{h_k}(t)\) and \(w_{h_k}(t)\), respectively, and the value error of these approximations are
\[
\varepsilon_{h_k}(s_i, t) \triangleq \psi(s_i; t; s_{-i}) - \hat{w}_{h_k}(t)^T \sigma_i(s_i; s_{-i}),
\]
\[
\varepsilon_{h_k}(s_i, t) \triangleq \dot{\psi}(s_i; t; s_{-i}) - \hat{w}_{h_k}(t)^T \sigma_i(s_i; s_{-i}).
\]
Then, using these definitions, and notation simplifications as in Table I, the HJB equation \((27)\) and its approximation are written as
\[
H = w_{h_1}^T \sigma_h + \left(f - \frac{1}{4c_3} BB^T ([\nabla \sigma_h]^T w_{h_0})\right)^T [\nabla \sigma_h]^T w_{h_0}
\]
\[
+ \frac{1}{2} \sum_{k=1}^{N} tr(GG^T[\Delta_{\sigma_h}^{[k]}] e_k)^T w_{h_k} + \phi_L + c_2 \hat{\phi}_G + \varepsilon_H = 0.
\]

\[
\hat{H} = \hat{w}_{h_1}^T \sigma_h + \left(f - \frac{1}{4c_3} BB^T ([\nabla \sigma_h]^T \hat{w}_{h_0})\right)^T [\nabla \sigma_h]^T \hat{w}_{h_0}
\]
\[
+ \frac{1}{2} \sum_{k=1}^{N} tr(GG^T[\Delta_{\sigma_h}^{[k]}] e_k)^T \hat{w}_{h_k} + \phi_L + c_2 \hat{\phi}_G.
\]

where the superscript \([k]\) shows the \(k\)’s element of the corresponding vector, \(e_k\) is a vector with \(k\)’s element equal to 1 and other elements equal to zero, and \(\varepsilon_H\) is the error of HJB equation with the function approximator defined as
\[
\varepsilon_H = c_2 \varepsilon_{\phi_G} + \varepsilon_{h_k} - \frac{1}{4c_3} [\nabla \varepsilon_{h_k}]^T BB^T [\nabla \sigma_h]^T w_{h_0} - \frac{1}{4c_3} [\nabla \varepsilon_{h_k}]^T BB^T [\nabla \sigma_h]^T \varepsilon_{h_k} - \frac{1}{2} tr(GG^T \Delta_{\sigma_h})
\]
\[
+ \left(f - \frac{1}{4c_3} BB^T [\nabla \sigma_h]^T \varepsilon_{h_k}\right)^T [\nabla \sigma_h]^T \varepsilon_{h_k} + \frac{1}{2} tr(GG^T \Delta_{\sigma_h})
\]
where \(\varepsilon_{\phi_G}\) is the uncertainty of the interaction term. Then, the corresponding approximate action can be obtained by
\[
a = -\frac{1}{2c_3} BB^T [\nabla \sigma_h]^T \varepsilon_{h_k} - \frac{1}{2c_3} BB^T [\nabla \varepsilon_{h_k}],
\]
\[
\hat{a} = -\frac{1}{2c_3} BB^T [\nabla \sigma_h]^T \hat{w}_{h_0}.
\]
Therefore, the error of approximating HJB equation by NNs is
Algorithm 1 Hjb control

1: **Initialization**: \( \hat{w}_{h_0}(n) = 0 \) and \( \hat{w}_{h_1}(n) = 0 \).
2: for Each UAV \( i = 1, \ldots, N \), in parallel, do
3: \( \text{Collect} \) the states \( s_{-i}(t) \) from neighboring UAVs.
4: \( \text{Update} \) the weights \( \hat{w}_{h_0}(n) \) and \( \hat{w}_{h_1}(n) \) by (39) and (40).
5: Calculate the value \( \hat{\psi} = \hat{w}_{h_0}^T \sigma H_0 \).
6: Take the optimal action \( \hat{a} = -\alpha \frac{1}{2c_3} B^T [\nabla \sigma^T] \hat{w}_{h_0} \).
7: **end for**

\[
e_H = H - H = -\hat{w}_{h_0} \sigma H - \hat{w}_{h_0}^T [\nabla \sigma^T] f - \frac{1}{4c_3} \hat{w}_{h_0}^T [\nabla \sigma^T] B B^T [\nabla \sigma^T] \hat{w}_{h_0}
+ \frac{1}{2c_3} \hat{w}_{h_0}^T [\nabla \sigma^T] B B^T [\nabla \sigma^T] \hat{w}_{h_0}
- \frac{1}{2} \sum_{k=1}^N \text{tr}(GG^T [\Delta \sigma_{0}^{[k]}]) \hat{a}_k \right)^T \hat{w}_{h_0} - e_H,
\]

where \( \hat{w}_{h_0} = w_{h_0} - \hat{w}_{h_0} \), and \( \hat{w}_{h_1} = w_{h_1} - \hat{w}_{h_1} \). Then, the optimal weights \( \hat{w}_{h_0} \) and \( \hat{w}_{h_1} \) should minimize the loss function defined as

\[
J_{h}(\hat{w}_{h_0}, \hat{w}_{h_1}) = \frac{1}{2} e_H^T e_H + \epsilon_H \max \left\{ 0, \hat{L}_s \right\} \| s_{i}(t) \| \geq s_{\text{dest}}.
\]

where \( \epsilon_H \) is a positive constant, \( L_s \) as the simplified notation of \( L_s(s(t)) \) is a Lyapunov candidate function, and \( \hat{L}_s \) is its derivative with respect to time. The regularizer term as shown as \( R_{i} \) or \( R_{t} \)(t) means to stop the movement when reaching the destination, i.e., \( s_i(T) = [r_i(T)^t, v_i(T)^t]^t \leq s_{\text{dest}} \). Then, by discretizing the time with \( d_t \) steps, the gradient descent updates are written as

\[
\hat{w}_{h_0}(n+1) = \hat{w}_{h_0}(n) - \mu_H [\nabla \phi_{h_0} \hat{e}_H] \hat{e}_H - \mu_H c_H \nabla \phi_{h_0} R_i,
\]

\[
\hat{w}_{h_1}(n+1) = \hat{w}_{h_1}(n) - \mu_H [\nabla \phi_{h_1} \hat{e}_H] \hat{e}_H,
\]

where the gradients \( \nabla \phi_{h_0} \hat{e}_H \) and \( \nabla \phi_{h_1} \hat{e}_H \) are obtained as

\[
\nabla \phi_{h_0} \hat{e}_H = [\nabla \phi_{h_0} f + \frac{1}{2} \sum_{k=1}^N \text{tr}(GG^T [\Delta \phi_{h_0}^{[k]}]) \hat{a}_k \rceil^T \hat{w}_{h_0} - \frac{1}{2c_3} [\nabla \sigma^T] B B^T [\nabla \sigma^T] \hat{w}_{h_0}.
\]

\[
\nabla \phi_{h_1} \hat{e}_H = \sigma H.
\]

The corresponding Hjb learning control based on these update equations is described in Algorithm 1. After initialization of the weights, each UAV has to collect instantaneous states of other UAVs and use it to update the equations (39) and (40). Then, it uses the updated model to take the proper action. However, the stability of this algorithm is explored by the following Proposition 1.

**Proposition 1 (Hjb Lyapunov stability)**: For small uncertainty of interaction term, i.e., \( \| c_{G_{ij}} \| \ll 1 \), and a bounded interaction term, i.e., \( \| c_{G_{ij}} \| \leq M_1 \), the system state and the model weights of constructed adaptive HJB neural network obtained by Algorithm 1 are uniformly ultimately bounded (UUB).

The UUB means that there exist \( s_{\text{dest}}, w_0, \) and \( w_1 \) at time \( T \) such that \( \| s(t) \| \leq s_{\text{dest}}, \| w_{h_0}(t) - \hat{w}_{h_0}(t) \| \leq w_0, \) and \( \| w_{h_1}(t) - \hat{w}_{h_1}(t) \| \leq w_1 \) for all \( t \geq T + T' \).

**Proof.** See Appendix A

B. MFG Learning Control

Here, we find an approximate solution to the pair of HJB-FPK equations in MFG framework. Regarding the HJB equation, we follow the method explained in previous subsection and by considering that the interaction term is obtained using (23). Then, we follow the similar approximation procedure to approximate the solution for FPK equation. Let us first rewrite the FPK equation (26) by using the simplified notations in Table I, and define \( F \) as

\[
F = \dot{m} + \nabla \cdot [(f - \frac{1}{2c_3} BB^T \nabla \psi)] - \frac{1}{2} \text{tr}(G G^T \Delta m) = 0
\]

Using the equality \( \nabla \cdot [a \hat{b}] = a \nabla \cdot \hat{b} + \hat{b} \nabla a \), where \( \hat{b} \) is a vector and \( a \) is a scalar, we rewrite this FPK equation as

\[
F = \dot{m} + m \nabla \cdot [(f - \frac{1}{2c_3} BB^T \nabla \psi)]
+ [(f - \frac{1}{2c_3} BB^T \nabla \psi)]^T \nabla m - \frac{1}{2} \text{tr}(G G^T \Delta m) = 0
\]

Now, we seek to find an approximate solution to the equation (44). Let us define the linear function approximator \( \hat{m}(s, t) \), which approximates the density function \( m(s, t) \), as

\[
\hat{m}(s, t) = \hat{w}_{F_0}(t) \sigma F(s),
\]

\[
\hat{m}(s, t) = \hat{w}_{F_1}(t) \sigma F(s).
\]

where \( \sigma F(s) \) is a vector of linear or nonlinear functions, and \( \hat{w}_{F_0}(t) \) and \( \hat{w}_{F_1}(t) \) are the approximations of \( w_{F_0}(t) \) and \( w_{F_1}(t) \) respectively. Then, the errors of approximating the distribution function \( m(s, t) \) and its derivative \( \dot{m}(s, t) \) are

\[
\varepsilon F_0(s, t) = m(s, t) - \hat{m}(s, t),
\]

\[
\varepsilon F_1(s, t) = \dot{m}(s, t) - \hat{m}(s, t).
\]

Considering this definition, and notation simplifications of Table I, the FPK equation (43) and its corresponding approximation are written as

\[
F = \dot{w}_{F_0}^T \sigma F + w_{F_0}^T \sigma F \nabla \cdot [(f - \frac{1}{2c_3} BB^T \nabla \psi)]
+ [(f - \frac{1}{2c_3} BB^T \nabla \psi)]^T \nabla m_{F_0} - \frac{1}{2} \text{tr}(G G^T \Delta m_{F_0}) = 0,
\]

\[
F = \dot{w}_{F_1}^T \sigma F + w_{F_1}^T \sigma F \nabla \cdot [(f - \frac{1}{2c_3} BB^T \nabla \psi)]
+ [(f - \frac{1}{2c_3} BB^T \nabla \psi)]^T \nabla m_{F_1} - \frac{1}{2} \text{tr}(G G^T \Delta m_{F_1}) = 0,
\]

where \( \varepsilon F \) denotes the error of FPK equation caused by the NN, and it is defined as

\[
\varepsilon F \triangleq \nabla \cdot [(f - \frac{1}{2c_3} BB^T \nabla \psi)] (\hat{w}_{F_0}^T \sigma F + \varepsilon F_0)
+ \varepsilon F_1 + \varepsilon F \nabla \cdot [(f - \frac{1}{2c_3} BB^T \nabla \psi)]
+ [(f - \frac{1}{2c_3} BB^T \nabla \psi)]^T \hat{w}_{F_0}
- \frac{1}{2} \text{tr}(G G^T \Delta m_{F_0}),
\]

where \( \varepsilon \psi \) is the uncertainty in finding \( \psi \). Therefore, the error of approximating FPK equation by neural networks is
Algorithm 2: Mfg control

1: **Initialization:** \( \hat{m}(s, 0) = \frac{1}{N} \sum_{j=1}^{N} \delta(s - s_j(0)), \hat{w}_{\mu_0}(n) = 0, \hat{w}_{\mu_1}(n) = 0, \hat{w}_{\mu_2}(n) = 0, \hat{w}_{\mu_3}(n) = 0 \).

2: **for** Each UAV \( i = 1, \ldots, N \), **in parallel,** **do**

3: **for** \( n = 1, \ldots, T_0 \) **do**

4: Update weights \( \hat{w}_{\mu_0}(n) \) and \( \hat{w}_{\mu_1}(n) \) by \((39)\) and \((40)\).

5: Calculate value \( \hat{v} = \hat{w}_{\mu_0}^T \sigma_{\mu_0} \).

6: Update weight \( \hat{w}_{\mu_2}(n) \) and \( \hat{w}_{\mu_3}(n) \) by \((54)\) and \((55)\).

7: Obtain MF distribution \( \hat{m} = \hat{w}_{\mu_0}^T \sigma_{\mu} \).

8: **end for**

9: Take the optimal action \( \hat{a} = -\frac{1}{2c_3}B^T[\nabla \sigma_{\mu}]^T \hat{w}_{\mu_0} \).

10: **end for**

\[
e_F = \hat{F} - F = -\hat{w}_{\mu_0}^T \sigma_{\mu} \nabla \cdot [(f - \frac{1}{2c_3}BB^T \nabla \hat{v})] - \hat{w}_{\mu_0}^T \nabla \sigma_{\mu} [(f - \frac{1}{2c_3}BB^T \nabla \hat{v})] + \frac{1}{2} \hat{w}_{\mu_0}^T \sum_{k=1}^{N} \lambda \{(G^T \Delta \sigma_{\mu}^k) e_k\} - \varepsilon F, \quad (52)
\]

where \( \hat{w}_{\mu_0} = w_{\mu_0} - \hat{w}_{\mu_0} \) and \( \hat{w}_{\mu_1} = w_{\mu_1} - \hat{w}_{\mu_1} \). Based on these definitions, the optimal weights \( \hat{w}_{\mu_0} \) and \( \hat{w}_{\mu_1} \) should minimize the loss function defined as

\[
J_F(\hat{w}_{\mu_0}, \hat{w}_{\mu_1}) = \frac{1}{2} e_F^T e_F \quad (53)
\]

Therefore, by discretizing the time with \( dt \) time steps, and the gradient descent updates for FPK weights are obtained as

\[
\hat{w}_{\mu_0}(n+1) = \hat{w}_{\mu_0}(n) - \mu_F(\nabla \hat{w}_{\mu_0} e_F) e_F \quad (54)
\]

\[
\hat{w}_{\mu_1}(n+1) = \hat{w}_{\mu_1}(n) - \mu_F(\nabla \hat{w}_{\mu_1} e_F) e_F \quad (55)
\]

where the gradients \( \nabla \hat{w}_{\mu_0} e_F \) and \( \nabla \hat{w}_{\mu_1} e_F \) are

\[
\nabla \hat{w}_{\mu_0} e_F = \sigma_{\mu} \nabla \cdot [(f - \frac{1}{2c_3}BB^T \nabla \hat{v}) + \nabla \sigma_{\mu} [(f - \frac{1}{2c_3}BB^T \nabla \hat{v})]]
\]

\[
- \frac{1}{2} \sum_{k=1}^{N} \lambda \{(G^T \Delta \sigma_{\mu}^k) e_k\} \quad (56)
\]

\[
\nabla \hat{w}_{\mu_1} e_F = \sigma_{\mu} \quad (57)
\]

However, based on these update pairs and update pairs for HJB equation, the Mfg learning algorithm is described as in Algorithm 2. In this algorithm, first, the UAVs share their states \( s_j(0) \) at time \( t = 0 \) to obtain the distribution of the population and initial samples. Then, the UAVs start collecting samples and updating their weights until they reach the destination at time \( T = T_0 dt \).

We have shown in our previous works \([4]\) that this method provides better results in terms of energy consumption, communications cost, and flocking of UAVs when enough samples are used to train the models. However, there is stability concerns about this algorithm which is guaranteed in the followings.

**Proposition 2 (FPK Lyapunov stability):** For almost certain \( \psi \), i.e., \( \| \psi \| \ll 1 \), and differentiable and bounded value function \( \psi \), i.e., \( \| \psi \| \leq M_2 \), the weights of constructed adaptive FPK neural network obtained in Algorithm 2, which is controlled by its corresponding HJB equation, are UUB.

The UUB means that there exist \( w_2 \) and \( w_3 \) at time \( T \) such that \( \| w_{\mu_0}(t) - \hat{w}_{\mu_0}(t) \| \leq w_2 \) and \( \| w_{\mu_1}(t) - \hat{w}_{\mu_1}(t) \| \leq w_3 \) for all \( t \geq T + t' \).

**Proof.** See Appendix \([B]\). □

**Proposition 3 (FPK Convergence):** Under the assumptions of Proposition 2 and with small step-sizes \( \mu_F \), the weights of FPK neural network function approximator converges to its optimal weights in mean with no bias and it is stable in mean square deviation sense.

**Proof.** See Appendix \([C]\). □

**Corollary 1:** Considering Propositions 1, 2, and 3, we can conclude that the system state and weights of constructed HJB-FPK neural networks obtained by Algorithm 2 are UUB.

**Proof.** The Algorithm 2 has two parts as HJB part and FPK parts. It is initialized by the states of the UAVs at the source region. At the initial iterations the states are used directly in the algorithm to update the weights of HJB and FPK neural networks, and hence the uncertainty of interaction term is small, i.e., \( \| \phi_{0c} \| \ll 1 \). Also, by starting with well trained or zero initialized neural network weights, both the interaction term and value functions are upper-bounded, i.e., \( \| \phi_{0c} \| \leq M_1 \) and \( \| \psi \| \leq M_2 \). Hence, there is a design, i.e., a choice of parameters in proofs in Appendices \([A]\), \([B]\), and \([C]\) such that all assumptions necessary for Propositions 1, 2, and 3 hold together and completely □

V. FEDERATED MFG LEARNING - MODEL SHARING METHODS

In this section, we propose federated mean field game learning strategy (MfgFL) and its different implementations, i.e., MfgFL-H, MfgFL-F, and MfgFL-HF, to make the UAVs’ control models close to each other and to use sample diversity among the UAVs efficiently. In MfgFL-H the model parameters of HJB neural network are shared with central unit to obtain the global HJB NN model, so the action rules of UAVs are close to each other. In MfgFL-F the FPK NN models are transmitted to the central unit to obtain the global FPK NN model, so the estimation of the population density function at UAVs be more accurate. In MfgFL-HF, both HJB and FPK neural network models are averaged to obtain better global online MFG learning model. In the following, we explain general form of MfgFL strategy, which covers three different implementations.

Although Algorithm 2 can reduce the communications cost of the control algorithm by leveraging the Mfg framework, it still requires big sample sets to train and provide conditions of stability. In other words, there is still a need to share a subset of samples among the UAVs or with a central unit, which requires extra communication costs in addition to privacy concerns. Therefore, instead of state sharing, we adopt the federated learning method to address these issues.

In the MfgFL algorithm, one UAV out of all is set to act as a control center, which we call as leader (or header)
Algorithm 3 MfgFL control

1: Initialization: \( \hat{m}(s, 0) = \frac{1}{N} \sum_{i=1}^{N} \delta(s - s_i(0)), \hat{w}_{h_0}(n) = 0, \hat{w}_{i_0}(n) = 0, \hat{w}_{0}(n) = 0. \)
2: for \( n = 0, 1, 2, \ldots, T_0 \) do
3: if \( n = k\eta_0 \) then
4: \( N_h \) UAVs, in parallel, send their model \( \hat{w}_{i,a}(kn_0) \) to the leader.
5: leader updates the model parameters \( \hat{w}_{h,a}(k) \), via
\[
\hat{w}_{h,a}(k) \leftarrow \frac{1}{N_h} \sum_{i \in I_h} \hat{w}_{i,a}(kn_0).
\] (58)
6: leader broadcasts the model \( \hat{w}_{h,a}(k) \).
7: end if
8: for each UAV \( i = 1, \ldots, N_h \), in parallel, do
9: if UAV \( i \) receives \( \hat{w}_{h,a}(k) \) then
10: Update \( \hat{w}_{i,a}(n) \) as
\[
\hat{w}_{i}(n) \leftarrow \hat{w}_{h,a}(k)
\] (59)
11: end if
12: Update \( \hat{w}_{F_0}(n), \hat{w}_{F_e}(n), \) and \( \hat{w}_{F_e}(n) \) by (39) and (40), (54), and (55).
Take the optimal action \( \hat{a} = -\frac{1}{\sqrt{2}} B^\top [\nabla \sigma_h]^{\top} \hat{w}_{h_0}. \)
13: end for
14: end for

In other words, the complete model of MFG is shared by the leader.

The leader obtains the average model \( \hat{w}_{h,a} \) by (58) after collecting models from other UAVs. However, after the average model is calculated at the leader and broadcasted to the UAVs, the updates are done locally by local samples at each UAV. Then, this procedure is repeated until all the UAVs reach the destination at repetition \( T_0 \).

In addition to reducing communication costs and increasing the privacy of the UAVs, the MfgFL method can provide other benefits as well such ensuring stability conditions for MFG framework and increasing training speed. One major condition for the MFG based approach is that the UAVs are indistinguishable. It means that the UAVs should have the same action rule, and hence, it is reasonable that they are trained by big enough samples. Nonetheless, due to energy/bandwidth limitations, it is not possible to provide this huge samples for model training. From this viewpoint, FL-based approaches can increase the model similarity among the UAVs and make them indistinguishable by efficiently using their samples for training.

Another benefit of using the MfgFL approach is the increased training speed of the models at UAVs. This is closely related to the communication cost of the algorithm, since utilizing model averaging means that the algorithm benefit from the various sample of UAVs in a shorter time span. Therefore, it is safe to say that it can provide higher model training speed. However, the performance of MfgFL is explored in next section.

VI. NUMERICAL RESULTS

In this section, we numerically validate the effectiveness of the proposed algorithm MfgFL-HF compared to the baseline methods Hjb control, MfgFL-F, and MfgFL-H, in terms of travel time, motion energy, collision avoidance, and communications cost. Throughout the simulations, we consider \( N \) UAVs controlled in a two-dimensional plane at the fixed altitude of \( h = 40m. \) Initially, the UAVs are equally separated with the distance \( \sqrt{2}m \) each other, and located at a source, which is a square region centered at \((150, 100)m\) in a 2-dimensional plane (see Fig. 2(a)). Each UAV aims to reach the destination at the origin, under the wind dynamics described by \( V_0 = 0.1 \) and \( V_0 = (1, -1)m/s \) (see Sec. II).

Following \( \{3, 10\} \) single hidden layer models (28) and (29) are considered for HJB model, where each hidden node’s activation function, i.e., \( \sigma_{H,J}(s_i(t)) \) for \( j = 1, \ldots, M_h \), corresponds to each scalar term in a polynomial expansion. The polynomial for \( \sigma_{H,J}(s_i(t)) \) is heuristically chosen as: \((1 + x_i(t) + v_{x_i}(t))^6 + (1 + y_i(t) + v_{y_i}(t))^6 \), where \( r_i(t) = [x_i(t), y_i(t)] \) and \( v_i(t) = [v_{x_i}(t), v_{y_i}(t)] \), thus the model size for HJB model is \( M_H = 54. \)

For the MFG based methods, the same neural network structure described above is considered to approximate HJB model. In a similar way, single hidden layer models (47) and (48) are considered for FPK model, where each hidden node’s activation function, i.e., \( \sigma_{F,J}(s_i(t)) \) for \( j = 1, \ldots, M_f \), corresponds to each scalar term in a polynomial expansion. The polynomial for \( \sigma_{F,J}(s_i(t)) \) is heuristically chosen as: \((1 + x_i(t) + v_{x_i}(t))^6 + (1 + y_i(t) + v_{y_i}(t))^6 \). Thus the model size for FPK model is \( M_F = 69. \)

Unless otherwise stated, the default simulation parameters are: \( P_s = 20dBm, W_s = 2MHz, \sigma_n = -110 dBm/Hz, \alpha = 0, \chi = 1.347, \xi = 6.649; n_0 = 100, N_h = 0.8N; r_{can} = 0.1m, rc = \sqrt{2}/2m; c_0 = 0.1, c_1 = c_2 = 0.015, c_3 = 0.005, m_H = 0.01, c_4 = 0.5 and d = 0.1s \) for the purpose of discretizing time in simulations.

Fig. 2 shows the trajectories of \( N = 25 \) UAVs under Hjb, MfgFL-H, MfgFL-F, and MfgFL-HF control methods. With
Fig. 2. Trajectory snapshots (left, 4 subplots for each control method) of 25 UAVs under (a) Hjb: HJB learning control with the communication range $d = 100m$, (b) MfgFL-H: MFG learning control with HJB model averaging, (c) MfgFL-F: MFG learning control with FPK model averaging, and (d) MfgFL-HF: MFG learning control with both HJB and FPK model averaging. During the travel time $t = 0 \sim 125s$, MfgFL-HF shows the best flocking behavior and the most stable HJB model parameters $u_{1,1}$ (rightmost subplot for each control method) of a randomly selected reference UAV $u_1$. Consequently, MfgFL-HF yields no collision during its entire travel, in sharp contrast to the others.

Hjb control, all the UAVs should communicate instantaneous states with each other, and use the received states to update their local HJB model. Therefore, Hjb control is extremely costly to be implement in real-time. However, for comparison purposes, it is assumed that the UAVs communicate their states at each time step to calculate the instantaneous interaction term, but the processing at each UAV is limited to one update of (39) and (40) per time step. This results in a fair comparison with FL-based methods, as they are also limited to one update of (39) and (40) per time step.

In all the methods, at first, the untrained UAVs follow the average wind direction while they train the models until the models are trained to the extend that their output commands turn the UAVs towards the destination. Then, the differences among algorithms in terms of collision, model weights, and interaction terms become observable from the trajectory and model weight plots, as explained in the following.

Collision occurrences is shown by star marks in the trajectories. It can be seen that in the proposed MfgFL-HF method, no collision has happened thanks to more sample utilization for HJB and FPK model training by adopting FL averaging for both models. Unlike MfgFL-HF, only one of the HJB or FPK models in MfgFL-H and MfgFL-F methods is trained with enough samples by utilizing FL method. The less-trained model results in more collisions of MfgFL-H and MfgFL-F methods as seen in the trajectory plots Fig. 2-b and Fig. 2-c. In Hjb method, although enough samples are provided, the UAVs can not use them to train the model in real-time due to limited processing power of the UAVs. Therefore, the models are not trained with enough samples, and a few collisions occur on the path to the destination as seen in the trajectory plots Fig. 2-a. These training behaviors can also be seen in HJB model parameters on the most right side of the Fig. 2, where in comparison to the other control methods, the model parameters in MfgFL-HF are less divergent after a period of time.

Fig. 2 shows the interaction term $\phi_G$ for each UAV using the color map on the trajectories. The bluer trajectories of MfgFL-HF method compared to other methods indicates lower interaction term values and better alignment of UAVs on the path to the destination. The reason is better training of the models in the proposed method as explained above. One main benefit of flocking of the UAVs instead of traveling individually or in different clusters is that it results in better communication channels among the UAV due to shorter distances, which can help in better model training and control. Further features of the proposed MfgFL-HF method corresponding to Fig. 2 is explained below using Fig. 3 and Fig. 4.

Fig. 3 represents the motion energy, communications payload, velocity alignment and number of collision risks of the UAVs corresponding to the scenario and methods in Fig. 2. Fig. 3-a represents the average motion energy and its variance among the UAVs. The proposed method MfgFL-HF method consumes at least 16% less energy than the other methods, and requires at least 4 times less communication costs than Hjb method (see
Fig. 3(b) at the cost of 10% and 6% more travel time compared to MfgFL-F and MfgFL-FH, respectively. The reason for less energy consumption of MfgFL-HF is that the UAVs can travel in a flock with a smaller interaction term on the trajectory as we observed in Fig. 2. This is due to better model training of MfgFL-HF by utilizing FL method. Furthermore, the reason for less communication costs for MfgFL-HF, MfgFL-F, and MfgFL-H methods is due to adopting the FL method. In FL-based methods, at every $n_0 = 100$ time steps, 80% of the UAVs transmit their models to the leader and the leader broadcasts it to all the UAVs. However, in Hjb method, all 25 UAVs broadcasts their states to all the neighbor UAVs at each time step.

Despite the disadvantage of more travel time for the MfgFL-HF method in this scenario, it demonstrates better velocity alignment and collision avoidance properties than the other defined FL-based methods as shown in Fig. 3c and Fig. 3d. As explained in definition of metrics $\phi_A(t)$ in (9) and $\phi_C(t)$ in (10), their smaller values correspond to better flocking behavior and lower probability of collision occurrence, respectively. Clearly, the cumulative value of $\phi_A(t)$ at time $T = 175s$ for MfgFL-HF method is at least 7% less than the other methods, which means better velocity alignment of MfgFL-HF. Furthermore, the cumulative value of $\phi_C(t)$ at time $T = 175s$ for MfgFL-HF is at least 8% less than the other methods, which means lower risk of collision occurrence of the proposed method. This complies with the training discussion above for Fig. 2.

Fig. 4 represents the absolute values of approximation errors of HJB in (37) and FPK in (52), equations corresponding to the scenario and methods in Fig. 2. It is noticeable that the values of approximation errors of HJB and FPK, despite not being too small, are acceptable small. This is in compliance with the analysis that the model weights are UUB. The approximation errors of HJB, i.e., $e_H$ in (37), depends on the error value of HJB model weights which is proved to be UUB in Proposition 1, and The approximation errors of FPK, i.e., $e_F$ in (52), depends on the error value of FPK model weights which is proved to be UUB in Proposition 2. Then, when the error value of model weights is below a threshold, the corresponding absolute values of approximate error of HJB in (37) and/or FPK in (52) will be bounded. This can be seen in Fig. 4a and Fig. 4b that the corresponding absolute error values are below 1.5 and 0.02, respectively.

Fig. 5 illustrates the performance of different methods versus $N$ number of UAVs. Clearly, MfgFL-HF requires less motion energy for $N = 16, \ldots, 64$, and its performance in terms of travel time $T$, velocity alignment $\phi_A(T)$, and number of collision risks $\phi_C(T)$, gets better as the number of UAVs increases. This is because, for higher $N$, more samples can be provided for both HJB and FPK models in MfgFL-HF which results in better training of both models. However, for the other two FL base methods, i.e., MfgFL-H and MfgFL-F, provided samples due to averaging improves only one of the HJB or FPK models, and the other corresponding model still remains less trained. Therefore, the coupled HJB-FPK equation in these two methods still is not well trained and non of MfgFL-H and MfgFL-F can benefit much when number of UAVs increases. Regarding Hjb, increasing the number of UAVs does not improve the performance much since it does not utilize the more provided samples for training the model due to the processing power limitations of the UAVs.

Fig. 6 shows the impact of model update period $n_0$ on
This is due to the fact that, for small values of \( n \), e.g., because of communication costs such as limited transmission risks than other FL-based methods (see Fig. 6-c and Fig. 6-d). An FL-based MFG learning method named MfgFL-HF risks than the other FL-based methods to complete the travel (see Fig. 6-a), while its travel time is only more than \( MfgFL-HF \) method consumes less energy on the right hand side of (A.3) as

\[
\begin{align*}
\dot{L}(t) &= \frac{1}{\mu_H} \hat{w}_{h_0}^T \hat{w}_{h_0} + \frac{1}{\mu_H} \hat{w}_{H_1}^T \hat{w}_{H_1} + c_H [\nabla L^T_\sigma x_\sigma] H \| x_\sigma \| \geq \gamma_{\text{dest}}. \\
\end{align*}
\]

This proof is based on methodology of [45], but with a few differences in system model and the update algorithms. The candidate Lyapunov function is chosen as

\[
\dot{L}(t) = \frac{1}{\mu_H} \hat{w}_{h_0}^T \hat{w}_{h_0} + \frac{1}{\mu_H} \hat{w}_{H_1}^T \hat{w}_{H_1} + c_H [\nabla L_\sigma] H \| x_\sigma \| \geq \gamma_{\text{dest}}.
\]

Then, the corresponding derivative function is

\[
\dot{L}(t) = \frac{1}{\mu_H} \hat{w}_{h_0}^T \hat{w}_{h_0} + \frac{1}{\mu_H} \hat{w}_{H_1}^T \hat{w}_{H_1} + c_H [\nabla L_\sigma] H \| x_\sigma \| \geq \gamma_{\text{dest}}.
\]

which can be rewritten in the following form

\[
\dot{L}(t) = \hat{w}_{h_0}^T [([\nabla \sigma_h] c_H) + c_H \nabla \omega_{h_0} R_h] \\
+ \hat{w}_{h_0}^T [([\nabla \omega_{h_0}] c_H) + c_H [\nabla L_\sigma] H \| x_\sigma \| \geq \gamma_{\text{dest}}]
\]

where \( b_H \) and matrix \( R_H \) are defined as

\[
b_H = [\nabla \sigma_h] (f - \frac{1}{2c_3} B B^T [\nabla \sigma_h] \sigma_h - \frac{1}{2c_3} B B^T [\nabla \omega_{h_0}]) \\
+ \frac{1}{2c_3} [\nabla \sigma_h] B B^T [\nabla \omega_{h_0}] + \frac{1}{2} \sum_{k=1}^N \mu (GG^T [\Delta \sigma_{h_1}]) \epsilon_k
\]

\[
\begin{align*}
\dot{L}(t) &= \hat{w}_{h_0}^T [([\nabla \sigma_h] c_H) + c_H \nabla \omega_{h_0} R_h] \\
+ \hat{w}_{h_0}^T [([\nabla \omega_{h_0}] c_H) + c_H [\nabla L_\sigma] H \| x_\sigma \| \geq \gamma_{\text{dest}}]
\end{align*}
\]

\[
R_H = \left[ \frac{1}{2c_3} [\nabla \sigma_h] B B^T [\nabla \sigma_h] \right]^T,
\]

and \( \tilde{s} \) is the dynamics of nominal system defined as

\[
\tilde{s} = f - \frac{1}{2c_3} B B^T [\nabla \sigma_h] \sigma_h - \frac{1}{2c_3} B B^T [\nabla \omega_{h_0}].
\]

Using the following relation as

\[
\begin{align*}
ab &= \frac{1}{2} \left( -\langle ha - \frac{b}{h} \rangle^2 + h^2 a^2 + \frac{b^2}{h^2} \right),
\end{align*}
\]

for scalars \( a \) and \( b \), we obtain the upper-bounds for the terms on the right hand side of (A.3) as

VII. CONCLUSION

In this paper, a novel path planning approach is proposed for a population of UAVs being effected by random wind perturbations in the environment. To this end, the objective is to minimize the transmission time, motion energy, and the interactions among the UAVs. First, the MFG framework is applied in order to reduce the high amount of communications required to control a massive number of UAVs. Next, a function approximator based on neural networks is proposed to approximate the solution of the HJB and FPK equations. The Lyapunov stability analysis for MFG learning is provided to show that the approximate solution for HJB and FPK equations are bounded. Then, on the bases of these assumptions and analyses, an FL-based MFG learning method named MfgFL-HF is proposed to use the samples of UAVs more efficiently for the purpose of training the model weights of neural networks at UAVs. The numerical results confirm the stability of the proposed method and show that it can be used to control a massive UAV population in a windy environment efficiently.

Appendix A

Proof of Proposition 1.
Depending on the state of the UAV, three cases can occur in (A.21) as

Case 1: \( \|s_i\| \geq \delta_{\text{min}} = 0 \). With this condition, we can conclude that the UAVs are in destination, and we focus only on the weights of the models

\[
\dot{L}(t) \leq -\lambda_0 \|\hat{w}_{\text{vd}}\|^4 + \lambda_1 \|\hat{w}_{\text{vd}}\|^2 + \lambda_2 - \lambda_3 \|\hat{w}_{\text{vd}}\|^2
\]

Then, when the following conditions hold, i.e.,

\[
\|\hat{w}_{\text{vd}}\| \geq \sqrt{\frac{\lambda_1 + \sqrt{\lambda_1^2 + 4\lambda_0\lambda_2}}{2\lambda_0}} \triangleq \omega_{0,1}, \quad (A.24)
\]
\[
\|\hat{w}_{\text{vd}}\| \geq \sqrt{\frac{4\lambda_0^2\lambda_2 + \lambda_1^2}{4\lambda_3\lambda_2}} \triangleq \omega_{1,1}, \quad (A.25)
\]

the stability condition \( \dot{L}(t) < 0 \) is satisfied.

Case 2: \( \|s_i\| \geq \delta_{\text{min}} = 1 \) and \( \dot{L}_M \leq 0 \). In this case, the regularizer term is inactive, and the upper-bound for derivative of Lyapunov is reduced to

\[
\dot{L}(t) \leq -\lambda_0 \|\hat{w}_{\text{vd}}\|^4 + \lambda_1 \|\hat{w}_{\text{vd}}\|^2 + \lambda_2^2 - \lambda_3 \|\hat{w}_{\text{vd}}\|^2 + c_M \|\nabla L_s\|^2 \triangleq \omega_{0,3}
\]

Case 3: \( \|s_i\| \geq \delta_{\text{min}} = 1 \) and \( \dot{L}_s \geq 0 \). In this case, we find the upper-bound for the \( \dot{L}(t) \) as

\[
\dot{L}(t) \leq -\lambda_0 \|\hat{w}_{\text{vd}}\|^4 + \lambda_1 \|\hat{w}_{\text{vd}}\|^2 + \lambda_2^2 - \lambda_3 \|\hat{w}_{\text{vd}}\|^2 + c_M \|\nabla L_s\|^2 \triangleq \omega_{1,3}
\]

where \( \lambda_{0,1,2,3} \) are defined as

\[
\lambda_0 = -\frac{3}{4h_1} \lambda_1^2 + \frac{1}{2} h_1^2 \lambda_1^2 - \frac{3}{4h_1^2} \lambda_2^2 + \frac{1}{2} \lambda_2^2,
\]
\[
\lambda_1 = \frac{3}{4h_1} \lambda_2 + \frac{1}{2} h_1^2 \lambda_2^2 + h_1^2 \lambda_2^2 - \lambda_1^2,
\]
\[
\lambda_2 = \frac{3}{4h_1} \lambda_3 + \frac{1}{2} h_1^2 \lambda_3^2 - \frac{1}{2} h_1^2 \lambda_3^2 - \lambda_2^2 + \lambda_4^2,
\]
\[
\lambda_3 = \frac{3}{4h_1} \lambda_4 + \frac{1}{2} h_1^2 \lambda_4^2 + \frac{1}{2} h_1^2 \lambda_4^2 - \lambda_3^2 + \lambda_4^2.
\]

Depending on the state of the UAV, three cases can occur in (A.21) as
In summary, when $\|\hat{w}_t\| \geq \omega_0 = \max\{\omega_{0,1}, \omega_{0,2}, \omega_{0,3}\}$, or $\|\hat{w}_t\| \geq \omega_1 = \max\{\omega_{1,1}, \omega_{1,2}, \omega_{1,3}\}$, or $\|\nabla L(s_i(t))\| \geq \max\{\gamma_2, \gamma_3\}$ occurs, then the Lyapunov stability condition holds, i.e., $\hat{L}(t) < 0$. Considering all the cases 1-3, we can conclude that there exist $s_{\text{dest}}, w_0$, and $w_1$ at time $T$ such that $\|s(t)\| \leq s_{\text{dest}}, \|w_0(t) - \hat{w}_t(0)\| \leq w_0,$ and $\|w_0(t) - \hat{w}_t(t)\| \leq w_1$ for all $t \geq T'$. 

**Appendix B**

**Proof of Proposition 2.**

The candidate Lyapunov function is chosen as

$$L(t) = \frac{1}{2\mu_F} \hat{w}_0^T \hat{w}_0 + \frac{1}{2\mu_F} \hat{w}_1^T \hat{w}_1.$$  \hspace{1cm} (B.1)

Then, the derivative of the Lyapunov function is obtained as

$$\dot{L}(t) = \hat{w}_0^T (\nabla_{\hat{w}_F} \hat{w}_F) \hat{w}_F + \hat{w}_1^T (\nabla_{\hat{w}_F} \hat{w}_F) \hat{w}_F,$$

$$= -\hat{w}_0^T (\nabla_{\hat{w}_F} \hat{w}_F) \hat{w}_F + \hat{w}_1^T \left[ \nabla_{\hat{w}_F} \hat{w}_F \right] + \hat{w}_1^T \|e\| \hat{w}_F + \|e\|,$$

$$= -\hat{w}_0^T (\nabla_{\hat{w}_F} \hat{w}_F) \hat{w}_F + \|e\| [\nabla_{\hat{w}_F} \hat{w}_F] + \|e\|.$$  \hspace{1cm} (B.2)

Each term of the derivative of the Lyapunov function has an upper-bound obtained as

$$-\hat{w}_0^T (\nabla_{\hat{w}_F} \hat{w}_F) \hat{w}_F \leq -\lambda_{\hat{m}}^2 \|\hat{w}_F\|^2,$$

$$-\hat{w}_0^T \left[ \nabla_{\hat{w}_F} \hat{w}_F \right] (\hat{w}_F) \leq \lambda_{\hat{m}} \lambda_{\hat{m}} \|\hat{w}_F\|^2,$$

where it is assumed that

$$\lambda_{\hat{m}} \leq \|\nabla_{\hat{w}_F} \hat{w}_F\| \leq \lambda_{\hat{m}} \lambda_{\hat{m}},$$

$$\|e\| \leq \lambda_{\hat{m}}.$$  \hspace{1cm} (B.3) \hspace{1cm} (B.4) \hspace{1cm} (B.5) \hspace{1cm} (B.6)

Then, the derivative of Lyapunov is upper-bounded as

$$\dot{L}(t) \leq -\lambda_{\hat{m}}^2 \|\hat{w}_F\|^2 + \lambda_{\hat{m}} \lambda_{\hat{m}} \|\hat{w}_F\|^2.$$  \hspace{1cm} (B.7)

Therefore, when the following condition occurs, i.e.,

$$\|\hat{w}_F\| \geq \frac{\lambda_{\hat{m}} \lambda_{\hat{m}}}{\lambda_{\hat{m}}^2},$$

the stability condition holds, i.e., $\dot{L}(t) < 0$. However, \hspace{1cm} (B.8)

means that the model which makes the term $\|e\|$ small, can increase the stability of FPK learning algorithm.

**Appendix C**

**Proof of Proposition 3.**

Here we aim to show the convergence and bias of the FPK learning updates following the proof method of [46]. Let us first define extended vectors as

$$\hat{w}_F(n) = [\hat{w}_f(n)]^T \quad [\hat{w}_f(n)]^T, \quad \nabla_{\hat{w}_F} \hat{w}_F = [\nabla_{\hat{w}_F} \hat{w}_F]^T \quad [\nabla_{\hat{w}_F} \hat{w}_F]^T.$$  \hspace{1cm} (C.1) \hspace{1cm} (C.2) \hspace{1cm} (C.3)

Then, the FPK error vector update is obtained as

$$\hat{w}_F(n+1) = [I - \mu_F [\nabla_{\hat{w}_F} \hat{w}_F] [\nabla_{\hat{w}_F} \hat{w}_F]^T] \hat{w}_F(n) - \mu_F [\nabla_{\hat{w}_F} \hat{w}_F] \hat{w}_F(n).$$  \hspace{1cm} (C.4)

### A. FPK Mean Stability

Taking the expectation of equation (C.4) yields

$$E[\hat{w}_F(n+1)] = [I - \mu_F E[\nabla_{\hat{w}_F} \hat{w}_F][\nabla_{\hat{w}_F} \hat{w}_F]^T]\hat{w}_F(n)$$

$$- \mu_F E[\nabla_{\hat{w}_F} \hat{w}_F]^T \hat{w}_F.$$  \hspace{1cm} (C.5)

We can assume that the vector $[\nabla_{\hat{w}_F} \hat{w}_F]$ which depends on the inputs, and $e_F$ which depends on the neural network design, are independent. Then we can write

$$E[\nabla_{\hat{w}_F} \hat{w}_F]^T \hat{w}_F = 0.$$  \hspace{1cm} (C.6)

Let us define the matrix $R$ as

$$R \triangleq E[\nabla_{\hat{w}_F} \hat{w}_F][\nabla_{\hat{w}_F} \hat{w}_F]^T].$$  \hspace{1cm} (C.7)

By substituting (C.6) and (C.7) in equation (C.5), it can be rewritten as the form

$$E[\hat{w}_F(n+1)] = [I - \mu_F R]E[\hat{w}_F(n)].$$  \hspace{1cm} (C.8)

Then, the necessary condition for the convergence of this equation is

$$0 < \mu_F < \frac{2}{\lambda_{\max}},$$

where $\lambda_{\max}$ is the largest eigenvalue of the matrix $R$.

### B. Biasness

Assuming small step-sizes and also the condition (C.9), the bias of estimation is calculated as

$$\text{bias} = \lim_{n \to \infty} E[\hat{w}_F(n)] = 0,$$

which means that if the step size is small enough such that the convergence condition holds, the parameters of the FPK equation tend to its optimal values.

### C. Mean Square Convergence Analysis

The mean square deviation (MSD) of the estimation algorithm is defined as

$$\text{MSD}_F = \lim_{n \to \infty} E[\|\hat{w}_F(n)\|^2].$$  \hspace{1cm} (C.11)

In order to find the MSD, let us first define the weighted MSD of the algorithm as $E[\|\hat{w}_F(n)\|^2_{\Sigma}]$, which can be obtained by the recursive equation

$$E[\|\hat{w}_F(n+1)\|^2_{\Sigma}] = E[\|\hat{w}_F(n)\|^2_{\Sigma}] + \mu_F^2 E[R_{\Sigma}] \|e\|^2,$$

where $\Sigma$ is a positive definite matrix, and

$$\Sigma = (I - \mu_F R)^T \Sigma(I - \mu_F R),$$

$$R_{\Sigma} = E[\nabla_{\hat{w}_F} \hat{w}_F]^T \Sigma \nabla_{\hat{w}_F} \hat{w}_F].$$  \hspace{1cm} (C.12) \hspace{1cm} (C.13) \hspace{1cm} (C.14)

We know that $\text{tr}(\Sigma X) = \text{vec}(X)^T \Sigma$, and $\text{vec}(UV) \in (V^T \otimes \Sigma)\sigma$, where $\text{vec}(\cdot)$ is a vectorization operator, i.e., $\text{vec}(\Sigma) = \sigma$. Using these equalities, we can obtain

$$\text{tr}(R_{\Sigma}) = \text{vec}(\Sigma)^T \sigma,$$

$$\sigma' = F \sigma,$$

$$F = (I - \mu_F R)^T \otimes (I - \mu_F R)^T.$$  \hspace{1cm} (C.15) \hspace{1cm} (C.16) \hspace{1cm} (C.17)

At the convergence stage, the MSD is written as

$$\lim_{n \to \infty} E[\|\hat{w}_F(n)\|^2_{\Sigma}] = \mu_F^2 \|e\|^2 \text{vec}(R)^T (I - F)^{-1} \text{vec}(\Omega).$$  \hspace{1cm} (C.18)
where $\vec{v}(\Omega) = (I - \mathcal{F}) \sigma$. Therefore the steady state MSD is obtained as
\[
\text{MSD} = \mu \mathbb{E}[\|\vec{e}\|^2 (\text{vec}(R))^\top (I - \mathcal{F})^{-1}\text{vec}(I)], \quad (C.19)
\]

The value of MSD can be very small by choosing a small value for step sizes, i.e., $\mu_F$, and choosing a model which makes $\|\vec{e}\|$ small.

REFERENCES

[1] H. Kim, J. Park, M. Bennis, and S.-L. Kim, “Massive UAV-to-ground communication and its stable movement control: A mean-field approach,” in Proc. IEEE SPAWC, Kalamata, Greece, Jun. 2018.

[2] E. Ackerman and E. Strickland, “Medical delivery drones take flight in east africa,” IEEE Spectrum, vol. 55, no. 1, pp. 34–35, Jan. 2018.

[3] J. Tisdale, Z. Kim, and J. K. Hedrick, “Autonomous UAV path planning and estimation,” IEEE Robot. Autom. Mag., vol. 16, no. 2, pp. 35–42, Jun. 2009.

[4] H. Shiri, J. Park, and M. Bennis, “Massive autonomous UAV path planning: A neural network based mean-field game theoretic approach,” arXiv preprint arXiv:1905.04752, 2019.

[5] M. Huang, R. P. Malhamé, P. E. Caines et al., “Large population stochastic dynamic games: closed-loop mean-field limit and the nash certainty equivalence principle,” Communications in Information & Systems, vol. 6, no. 3, pp. 221–252, 2006.

[6] M. Huang, P. E. Caines, and R. P. Malhamé, “Large-population cost-coupled lag problems with nonuniform agents: individual-mass behavior and decentralized n-nash equilibria,” IEEE transactions on automatic control, vol. 52, no. 9, pp. 1560–1571, 2007.

[7] J.-M. Lasry and P.-L. Lions, “Mean field games,” Japanese journal of mathematics, vol. 2, no. 1, pp. 299–260, 2007.

[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson et al., “Communication-efficiency-learnedcontenido de deep networks from decentralized data,” arXiv preprint arXiv:1602.05629, 2016.

[9] R. Shokri and V. Shmatikov, “Privacy-preserving deep learning,” in Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, 2015, pp. 1310–1321.

[10] D. Liu, D. Wang, F. Wang, H. Li, and X. Yang, “Neural-network-based online hjb solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems,” IEEE Transactions on Cybernetics, vol. 44, no. 12, pp. 2834–2847, Dec 2014.

[11] H. Shiri, J. Park, and M. Bennis, “Remote uav online path planning via neural network based opportunistic control,” arXiv preprint arXiv:1910.04969, 2019.

[12] A. Kavvadias, E. Poomiadis, M. Chanioti, E. Gala, and S. Michas, “Pressure-precision in vineyards using high resolution hyperspectral imagery acquired from an unmanned aerial vehicle (uav),” Agricultural and forest meteorology, vol. 171, pp. 281–294, 2013.

[13] D. Rathiun, S. Kraguljand, A. Pongponrattanawattana, and B. Capozzi, “An evolution based path planning algorithm for autonomous motion of a uav through uncertain environments,” in Proceedings. The 21st Digital Avionics Systems Conference, vol. 2. IEEE, 2002, pp. S2–8D2.

[14] Y. Zeng and R. Zhang, “Energy-efficient uav communication with trajectory optimization,” IEEE Transactions on Wireless Communications, vol. 16, no. 6, pp. 3747–3760, 2017.

[15] H. Inaltekin, M. Gorlatova, and M. Chiang, “Virtualized control over fog: interplay between reliability and latency,” IEEE Internet of Things Journal, vol. 5, no. 6, pp. 5030–5045, 2018.

[16] D. Doering, A. Benenmann, R. Lern, E. P. de Freitas, I. Muller, J. M. Winter, and C. E. Pereira, “Design and optimization of a heterogeneous platform for multiple uav for precision agriculture applications,” IFAC Proceedings Volumes, vol. 47, no. 3, pp. 1272–1277, 2014.

[17] G. Vásárhelyi, C. Virág, G. Somorjai, T. Nepus, A. E. Eiben, and T. Vicsek, “Optimized flocking of autonomous drones in confined environments,” Science Robotics, vol. 3, no. 20, 2018. [Online]. Available, http://robotics. sciencemag.org/content/3/20/eaaq3536

[18] C. Ju and H. Son, “Multiple uav systems for agricultural applications: control, implementation, and evaluation,” Electronics, vol. 7, no. 9, p. 162, 2018.

[19] M. Nourian, P. E. Caines, and R. P. Malhamé, “Mean field analysis of controlled cucker-smale type flocking: Linear analysis and perturbation equations,” IFAC Proceedings Volumes, vol. 44, no. 1, pp. 4471–4476, 2011.

[20] K. W. Morton and D. F. Meyers, Numerical solution of partial differential equations: an introduction. Cambridge university press, 2005.

[21] S. Khardi, “Aircraft flight path optimization. the hamilton-jacobi-bellman considerations,” Applied Mathematical Sciences, vol. 6, no. 25, pp. 1221–1212, 2012.

[22] C. Greif, “Numerical methods for hamilton-jacobi-bellman equations,” 2017.

[23] Y. Tassa and T. Erez, “Least squares solutions of the hjb equation with neural network value-function approximators,” IEEE transactions on neural networks, vol. 18, no. 4, pp. 1031–1041, 2007.

[24] B. Luo, H.-N. Wu, T. Huang, and D. Liu, “Reinforcement learning solution for hjb equation in constrained optimal control problem,” Neural Networks, Vol. 71, pp. 150–158, 2015.

[25] A. Faust, P. Raymgaart, M. Salman, R. Fierro, and L. Tapia, “Continuous action reinforcement learning for control-affine systems with unknown linear system,” IEEE/CAC Journal of Automatica Sinica, vol. 1, no. 3, pp. 323–336, 2014.

[26] J. W. Kim, B. J. Park, H. Yoo, J. H. Lee, and J. M. Lee, “Deep reinforcement learning based finite-horizon optimal tracking control for nonlinear system,” IFAC-PapersOnLine, vol. 51, no. 25, pp. 257–262, 2018.

[27] C. Beck, S. Becker, P. Grohs, N. Jafarafi, and A. Jentzen, “Solving stochastic differential equations and Kolmogorov equations by means of deep learning,” arXiv e-prints, p. arXiv:1806.00421, Jun, 2018.

[28] H. M. Lu, “Multi-robot swarm for cooperative scalar field mapping,” in Handbook of Research on Design, Control, and Modeling of Swarm Robotics. IGI Global, 2016, pp. 383–395.

[29] H. M. Lu, W. Sheng, and J. Chen, “Cooperative and active sensing in mobile sensor networks for scalar field mapping,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 45, no. 1, pp. 1–12, 2015.

[30] P. Sujit, S. Saripalli, and J. B. Sousa, “Reinforcement learning for control-affine systems with unknown characteristics and a uav specific extension to the rice model,” in Proc. IEEE PESGM, Boston, MA, USA, 2016.

[31] F. Cucker and J.-G. Dong, “Avoiding collisions in flocks,” IEEE Transactions on Automatic Control, vol. 55, no. 5, pp. 1238–1243, 2010.
[43] Z. Lin, M. Broucke, and B. Francis, “Local control strategies for groups of mobile autonomous agents,” IEEE Transactions on automatic control, vol. 49, no. 4, pp. 622–629, 2004.

[44] N. Goddemeier and C. Wietfeld, “Investigation of air-to-air channel characteristics and a uav specific extension to the rice model,” in 2015 IEEE Globecom Workshops (GC Wkshps), Dec 2015, pp. 1–5.

[45] D. Liu, D. Wang, F.-Y. Wang, H. Li, and X. Yang, “Neural-network-based online HJB solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems,” IEEE Trans. Cybern., vol. 44, no. 12, pp. 2834–2847, Dec. 2014.

[46] H. Shiri, M. A. Tinati, M. Codreanu, and S. Daneshvar, “Distributed sparse diffusion estimation based on set membership and affine projection algorithm,” Digital Signal Processing, vol. 73, pp. 47–61, 2018.