

Massive Autonomous UAV Path Planning: A Neural Network Based Mean-Field Game Theoretic Approach

Hamid Shiri, Jihong Park, and Mehdi Bennis

Centre for Wireless Communications, University of Oulu, Finland, Email: {hamid.shiri, jihong.park, mehdi.bennis}@oulu.fi

Abstract—This paper investigates the autonomous control of massive unmanned aerial vehicles (UAVs) for mission-critical applications (e.g., dispatching many UAVs from a source to a destination for firefighting). Achieving their fast travel and low motion energy without inter-UAV collision under wind perturbation is a daunting control task, which incurs huge communication energy for exchanging UAV states in real time. We tackle this problem by exploiting a mean-field game (MFG) theoretic control method that requires the UAV state exchanges only once at the initial source. Afterwards, each UAV can control its acceleration by locally solving two partial differential equations (PDEs), known as the Hamilton-Jacobi-Bellman (HJB) and Fokker-Planck-Kolmogorov (FPK) equations. This approach, however, brings about huge computation energy for solving the PDEs, particularly under multi-dimensional UAV states. We address this issue by utilizing a machine learning (ML) method where two separate ML models approximate the solutions of the HJB and FPK equations. These ML models are trained and exploited using an online gradient descent method with low computational complexity. Numerical evaluations validate that the proposed ML aided MFG theoretic algorithm, referred to as *MFG learning control*, is effective in collision avoidance with low communication energy and acceptable computation energy.

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

I. INTRODUCTION

Many unmanned aerial vehicles (UAVs) are essential in mission-critical applications, for covering wide disaster sites in emergency cellular networks [1] and for delivering heavy payload in rescue mission and firefighting scenarios [2], [3]. These applications are real-time, and do not tolerate remote control delays from a central controller. Besides, they necessitate reliable control under uncertainty such as wind perturbations, making pre-programmed offline control algorithms ill-suited. In view of this, in this paper we focus on the problem of controlling a large number of UAVs in a distributed and online way, so as to achieve 1) the *fastest travel* from a source to a destination, while jointly minimizing 2) *motion energy*, and 3) *inter-UAV collision*, under wind dynamics.

This problem is challenging as illustrated in Fig. 1, wherein each UAV is faced with making control decisions with many degrees of freedom, while taking into account energy-saving and collision-avoidance. For collision avoidance, multiple UAVs need to interact with each other, which require inter-UAV communications whose delay and/or energy cost increases exponentially with the number of UAVs. Such communication and control overhead is persistent as the control must be continual under wind perturbations.

To address the aforementioned issues, we leverage mean-field game (MFG) theory [4], [5], a mathematical framework that is effective in reducing the communication and control overhead of distributed control under agent interactions (e.g., collisions) through their states (e.g., locations) [1]. At its core,

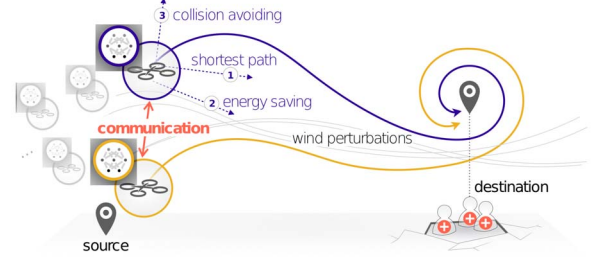


Fig. 1. An illustration of dispatching massive UAVs from a source point to a destination site. Each UAV communicates with neighboring UAVs for achieving: 1) the fastest travel, while jointly minimizing 2) motion energy and 3) inter-UAV collision, under wind perturbations.

MFG considers a large number of agents, each of which approximately views the other agents' states as the *global state* averaged across all agents. The global state is identically given for all agents at any given time, and one can thus focus only on controlling a single agent while incorporating its interactions via the global state distribution, referred to as the *mean-field (MF) distribution*.

The said MFG theoretic control is operated by locally solving two partial differential equations (PDEs) at each agent. Namely, a single agent computes the MF distribution by solving the *Fokker-Planck-Kolmogorov (FPK)* equation, so long as the initial global state is known by exchanging agents' states only once. For the given MF distribution, the optimal control of the agent is determined by solving the other PDE induced by a continuous-time Markov decision problem (MDP), known as the *Hamilton-Jacobi-Bellman (HJB)* equation [5].

While effective, MFG theoretic methods are computationally expensive due to solving HJB and FPK equations, limiting their adoption for real-time multi-dimensional control applications [6]. To circumvent this problem, we propose an *MFG learning control* algorithm in which the HJB and FPK solutions are approximated using two separate machine learning (ML) models (e.g., neural networks), denoted as *HJB model* and *FPK model*, respectively. The HJB and FPK models stored at each UAV are simultaneously trained and exploited for control in an online manner. Simulations validate that the proposed MFG learning control achieves collision-free travel with significant communication energy reduction, at the cost of a slight increase in computation and motion energy.

Related works. The problem of UAV placement for supporting communication systems has been studied in [7]. Under wind perturbations, the real-time placement of massive UAVs without collision has been investigated in [1]. Path planning is a more challenging problem wherein UAVs are controlled to reach a destination. In offline control, a multiple-UAV scenario have been addressed in [7]. In online control, an evolutionary algorithm [8] and a partially observable Markov decision process based method [9] have been proposed. For other communication and control related issues for UAV systems, readers are encouraged to check [10] and [3], respectively.

This research was supported in part by Academy of Finland (Grant Nr. 294128), in part by the 6Genesis Flagship (Grant Nr. 318927), in part by the Kvantum Institute Strategic Project (SAFARI), and in part by the Academy of Finland through the MISSION Project (Grant Nr. 319759).

II. SYSTEM MODEL

We assume a set \mathcal{N} of N UAVs traveling from a common source to a destination in a two-dimensional plane, where the origin is set as the destination. At time $t \geq 0$, the i -th UAV $u_i \in \mathcal{N}$ controls its acceleration $a_i(t) \in \mathbb{R}^2$, so as to minimize its: **1) travel time**, **2) motion energy**, and **3) inter-UAV collision**, during the remaining travel to the destination. The control $a_i(t)$ of u_i is based not only on its local state $s_i(t)$, but also on the states $s_{-i}(t)$ of a set $\mathcal{N}_i(t) \subset \mathcal{N}$ of the other $(N_i - 1)$ UAVs within u_i 's communication range $d > 0$ with $N_i(t) = |\mathcal{N}_i(t)| \leq N$ for collision avoidance, and $s_{-i}(t) = \{s_j(t) \mid \|r_j(t) - r_i(t)\| \leq d, \forall j \neq i\}$. The communication range d is determined by the minimum received signal-to-noise ratio (SNR) $\theta > 0$ required for successful decoding under the standard path loss model, which is given as $d = [P/(\theta\sigma^2)]^{1/\alpha}$ with an identical transmission power P , noise power σ^2 , and path loss exponent $\alpha \geq 2$.

The state $s_i(t) = [r_i(t)^\top, v_i(t)^\top]^\top \in \mathbb{R}^4$ of UAV u_i is comprised of its location $r_i(t) \in \mathbb{R}^2$ and velocity $v_i(t) \in \mathbb{R}^2$ that are dynamically updated by the control $a_i(t)$ under random wind dynamics. Following [11], the wind dynamics are assumed to follow an Ornstein-Uhlenbeck process with an average wind velocity v_o . The temporal state dynamics are thereby given as:

$$dv_i(t) = a_i(t)dt - c_0(v_i(t) - v_o)dt + V_o dW_i(t) \quad (1)$$

$$dr_i(t) = v_i(t)dt, \quad (2)$$

where c_0 is a positive constant, $V_o \in \mathbb{R}^{2 \times 2}$ 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.

To achieve the aforementioned goals **1)**, **2)**, and **3)**, UAV u_i at time $t < T$ aims to minimize its average cost $\psi_i(t)$, where the average is taken with respect to the measure induced by all possible controls for $\tau \in [t, T]$. The cost $\psi_i(t)$ consists of the term $\phi_L(s_i(t))$ depending only on the local state $s_i(t)$ and the term $\phi_G(s_{N_i}(t))$ relying on the global state $s_{N_i}(t) = [s_i(t)^\top, s_{-i}(t)^\top]^\top \in \mathbb{R}^{4N_i}$ observed by u_i , given as:

$$\psi_i(t) = \mathbb{E} \left[\int_t^T \left(\phi_L(s_i(\tau)) + c_3 \phi_G(s(\tau)) \right) d\tau \right] \quad (3)$$

where

$$\begin{aligned} \phi_L(s_i(t)) &= \underbrace{\frac{v_i(t) \cdot r_i(t)}{\|r_i(t)\|}}_{\text{1) travel time minimization}} + \underbrace{c_1 \|v_i(t)\|^2 + c_2 \|a_i(t)\|^2}_{\text{2) motion energy minimization}}, \\ \phi_G(s_{N_i}(t)) &= \underbrace{\frac{1}{N_i(t) - 1} \sum_{u_j \in \mathcal{N}_i(t) \setminus \{u_i\}} \frac{\|v_j(t) - v_i(t)\|^2}{(\varepsilon + \|r_j(t) - r_i(t)\|^2)^\beta}}_{\text{3) collision avoidance \& connectivity guarantee}}, \end{aligned}$$

and the terms c_1 , c_2 , β , and ε are positive constants.

The local term $\phi_L(s_i(t))$ in (3) focuses on the following two objectives. For **1) travel time** minimization, it is intended to minimize the remaining travel distance $\|r_i(t)\|$ by maximizing the velocity towards the destination, i.e., minimizing the projected velocity $v_i(t) \cdot r_i(t)/\|r_i(t)\|$ towards the opposite direction to the destination. For **2) motion energy** minimization, it is planned to minimize the kinetic energy and the acceleration control energy that are proportional to $\|v_i(t)\|^2$ and $\|a_i(t)\|^2$, respectively [12], [13].

The global term $\phi_G(s_{N_i}(t))$ in (3) refers to **3) collision avoidance**, and is intended to form a flock of UAVs moving together [14]. The flocking leads to small relative inter-UAV

velocities for avoiding collision even when their controlled velocities are slightly perturbed by wind dynamics. Furthermore, the flocking yields closer inter-UAV distances without collision. This is beneficial for allowing more UAVs to exchange their states, i.e., larger $N_i(t)$, thereby contributing also to collision avoidance. In view of this, we adopt the Cucker-Smale flocking [1], [14] 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 $\phi_G(s_{N_i}(t))$, respectively.

Incorporating the cost (3) under the temporal dynamics (1) and (2), the control problem of UAV u_i at time t is formulated as:

$$\psi_i^*(t) = \min_{a_i(t)} \psi_i(t) \quad (4)$$

$$\text{s.t. } ds_i(t) = (As_i(t) + B(a_i(t) + c_0 v_o))dt + GdW_i(t), \quad (5)$$

where $A = \begin{pmatrix} 0 & I \\ 0 & -c_0 I \end{pmatrix}$, $B = \begin{pmatrix} 0 \\ I \end{pmatrix}$, $G = \begin{pmatrix} 0 \\ V_o \end{pmatrix}$, and I denotes the two-dimensional identity matrix. The minimum cost $\psi_i^*(t)$ is referred to as the *value function* of the optimal control, and is derived using two different control methods in the next section.

III. HJB CONTROL AND MFG CONTROL

Deriving the UAV u_i 's value function $\psi_i^*(t)$ in (4) is intertwined with other UAVs, through the collision avoidance term $\phi_G(s_{N_i}(t))$ in (3). Therefore, this is an N_i -player non-cooperative game whose well-known solution is the Nash equilibrium (NE), i.e., the control decisions under which no UAV can unilaterally decrease its cost [5]. Its solution complexity exponentially increases with N_i , which is a poor fit for real-time applications. To address this pressing concern, in this section we consider two different control methods: **1) HJB control**, our baseline method in which each UAV's control only takes into account the other UAVs' *states before taking their actions*; and **2) MFG control**, our proposed method that incorporates the intertwined controls via an approximated global state distribution, i.e., the *MF distribution*.

It is noted that HJB control does not always achieve the NE as it intentionally neglects the actual control interactions, i.e., the *states when taking actions*. On the other hand, MFG control relies on the MF approximation, and only achieves the NE asymptotically when $N \rightarrow \infty$ [5]. The operational details of both control schemes are elaborated in the following subsections, and their effectiveness under a large finite number of UAVs will be numerically examined in Sec. V.

A. HJB Control

The UAV u_i 's value function $\psi_i^*(t)$ in (4) is equivalent to the solution of its corresponding HJB $H(\psi_i^*(t); s_{N_i}(t)) = 0$ formulated according to the Markov decision principle. The left-hand side $H(\psi_i^*(t); s_{N_i}(t))$ is given by putting $\psi_i^*(t)$ into $H(\psi_i(t); s_{N_i}(t))$ in (7) at the bottom of the next page (see the derivation details in [5]). Due to the global term $\phi_G(s_{N_i}(t))$ therein for collision avoidance, the HJB solution requires collecting the other UAVs' states. Furthermore, achieving the NE of N_i -UAV controls, necessitates solving N_i -coupled HJBs whose required number of state exchanges exponentially increases with N_i . For example, each HJB is first solved while the other $(N_i - 1)$ UAVs' states are fixed, and this should be iterated for N_i UAVs in a recursive manner until all action changes stop, i.e., convergence to the NE [5]. The said N_i -coupled HJB solutions require $N_i \times (N_i - 1) \times K$ state exchanges per time instant t , where K denotes the number of iterations until convergence to the NE.

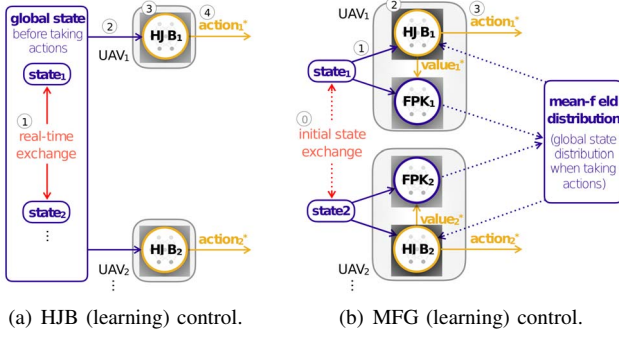


Fig. 2. Operational structures of HJB (learning) control and MFG (learning) control.

Such excessive communication overhead is not bearable for real-time UAV controls. Therefore, while compromising convergence to the NE, as a baseline control scheme we instead consider HJB control of UAVs that exchange $\sum_{i=1}^{N_i} N_i(t)$ number of states before solving the HJBs, i.e., before taking actions, at each time instant t . Afterwards, each HJB is solved independently without recursion, as visualized in Fig. 2-a. At time t , u_i 's HJB control is summarized as below.

Algorithm 1. HJB Control

- 1) Collect the states $s_{-i}(t)$ from $(N_i(t) - 1)$ UAVs.
- 2) Calculate the value $\psi_i^*(t)$ by solving the HJB $H(\psi_i^*(t); s_{N_i}(t)) = 0$ (see (7)).
- 3) Take the optimal action $a_i^*(t) = -\frac{1}{2c_2} B^T \nabla \psi_i^*(t)$.

Here, (7) is derived by applying the optimal control $a_i^*(t)$ to (6), where ∇ denotes the differential operator taken with respect to $s_i(t)$. The optimal control $a_i^*(t) = -\frac{1}{2c_2} B^T \nabla \psi_i^*(t)$ is obtained according to the Karush-Kuhn-Tucker (KKT) conditions, since the HJB's Hamiltonian, i.e., the terms inside the infimum in (6), is convex with respect to $a_i(t)$. The existence of $a_i^*(t)$ is ensured by the fact that the HJB with (6) has a unique solution $\psi_i^*(t)$ according to [5], as long as the drift term $As_i(t) + B(a_i(t) + c_0 v_o)$ in (5) and the instantaneous cost $\phi_L(s_i(t)) + \phi_G(s_{N_i}(t))$ are smooth, i.e., continuous first derivatives.

B. MFG Control

Compared to HJB control with $\sum_{i=1}^{N_i} N_i(t)$ state exchanges *per time instant* t , MFG control requires $N \times (N - 1)$ state exchanges *only at the initial time* $t = 0$, while asymptotically guaranteeing the NE anytime as N goes to infinity. This is viable by locally calculating the MF distribution $m(t)$ that asymptotically converges to the (empirical) global state

distribution when all actions are taken under the NE, i.e., $\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{s_i(t)} = m(t)$. With finite UAVs, it yields an MF approximation that achieves the ϵ -NE [5].

To this end, each UAV under MFG control locally solves a pair of the HJB $H(\psi_i^*(t); s_i(t), m(t)) = 0$ (see (8) with $\psi_i^*(t)$) and its coupled FPK $F(m(t); s_i(t), \psi_i^*(t)) = 0$ (see (10) with $\psi_i^*(t)$) that is derived from the state dynamics (5) with the Itô's lemma [5]. As illustrated in Fig. 2-b, solving the HJB produces the value $\psi_i^*(t)$ (or its corresponding optimal action $a_i^*(t)$), which is fed to the FPK whose solution is the MF distribution $m(t)$. This operation is locally iterated K times until it converges to the NE. At time t , u_i 's MFG control is described as follows.

Algorithm 2. MFG Control

For $k \in [1, K]$:

- 1) Calculate the value $\psi_i^{[k]}(t)$ by solving the HJB $H(\psi_i^{[k]}(t); s_i(t), m^{[k-1]}(t)) = 0$ (see (8)).
- 2) Calculate the MF distribution $m^{[k]}(t)$ by solving the FPK $F(m^{[k]}(t); s_i(t), \psi_i^{[k]}(t)) = 0$.
- 3) Iterate 1) and 2) until $k = K$.
- 4) Take the optimal action $a_i^*(t) = -\frac{1}{2c_2} B^T \nabla \psi_i^{[K]}(t)$.

Initial MF distribution $m^{[0]}(t)$ at $k = 0$:

- If $t = 0$, $m^{[0]}(0) = 1/N \sum_{i=1}^N \mathbb{1}_{s_i(0)}$, computed by collecting the states $s_{-i}(0)$ from N UAVs.
- Otherwise, $m^{[0]}(t) = m^{[K-1]}(t - \Delta t)$, i.e., the converged MF distribution in the previous control where Δt denotes the control interval.

It is noted that the HJB's global term $\phi_G(s_i(t), m(t))$ in (8) approximates $\phi_G(s_{N_i}(t))$ in (7), where

$$\phi_G(s_i(t), m(t)) = \int_s m(t) \frac{\|v(t) - v_i(t)\|^2}{(\epsilon^2 + \|r(t) - r_i(t)\|^2)^\beta} ds. \quad (11)$$

This MF approximation is based on treating each of the UAVs' states as $s = [r(t)^T, v(t)^T]^T$ induced by the MF distribution $m(t)$. The approximation converges to the exact value as $N \rightarrow \infty$, so long as $\phi_G(s_{N_i}(t))$ is bounded and UAV indices are permutable, i.e., the exchangeability of actions for the same states (see the condition details in [5]).

IV. ML AIDED HJB AND MFG CONTROLS

Both HJB and MFG controls are facilitated by the HJB and FPK equations. These PDEs are solved by discretizing the domain in a way that the derivatives therein can be approximated using finite differences. Unfortunately, such a finite difference method requires finer discretization as the domain dimension increases, incurring higher computational

$$\text{(For HJB)} \quad H(\psi_i(t); s_{N_i}(t)) = \partial_t \psi_i(t) + \inf_{a_i(t)} \left\{ [As_i(t) + B(a_i(t) + c_0 v_o)]^T \nabla \psi_i(t) + \frac{1}{2} \text{tr}(GG^T \nabla^2 \psi_i(t)) + \phi_L(s_i(t)) + \phi_G(s_{N_i}(t)) \right\} \quad (6)$$

$$= \partial_t \psi_i(t) + \left[As_i(t) - \frac{1}{2c_2} BB^T \nabla \psi_i(t) + c_0 B v_o \right]^T \nabla \psi_i(t) + \frac{1}{2} \text{tr}(GG^T \nabla^2 \psi_i(t)) + \phi_L(s_i(t)) + \phi_G(s_{N_i}(t)) \quad (7)$$

$$\text{(For MFG)} \quad H(\psi_i(t); s_i(t), m(t)) = \partial_t \psi_i(t) + \left[As_i(t) - \frac{1}{2c_2} BB^T \nabla \psi_i(t) + c_0 B v_o \right]^T \nabla \psi_i(t) + \frac{1}{2} \text{tr}(GG^T \nabla^2 \psi_i(t)) + \phi_L(s_i(t)) + \phi_G(s_i(t), m(t)) \quad (8)$$

$$F(m(t); s_i(t), \psi_i(t)) = \partial_t m(t) + \nabla \left(\left[As_i(t) + B(a_i^*(t) + c_0 v_o) \right] m(t) \right) - \frac{1}{2} \text{tr}(GG^T \nabla^2 m(t)) \quad (9)$$

$$= \partial_t m(t) + \nabla \left(\left[As - \frac{1}{2c_2} BB^T \nabla \psi_i(t) + c_0 B v_o \right] m(t) \right) - \frac{1}{2} \text{tr}(GG^T \nabla^2 m(t)) \quad (10)$$

complexity. For instance, in a two-dimensional x - y domain, the convergence of a numerical PDE solution with the temporal discretization step size Δt is guaranteed by the Courant-Friedrichs-Lewy (CFL) condition $\Delta t \leq (\Delta x^{-1} + \Delta y^{-1})^{-1}$ whose feasible step size is smaller than the required step size in a one-dimensional domain, i.e., $\Delta t \leq \Delta x$ [6].

To enable multi-dimensional control in real time with low computational complexity, we propose *HJB learning control* and *MFG learning control* that approximate both HJB control and FPK control in Sec. III, respectively. Via these methods, ML models learn to solve the HJB and FPK in an online way, as elaborated in the following subsections.

A. HJB Learning Control

HJB learning control exploits ML to enable and represent the baseline method, HJB control in Sec. III-A. The key idea is to approximate the problem of solving the HJB equation $H(\psi_i^*(t); s_{N_i}(t)) = 0$ by minimizing $H(\hat{\psi}_i(t); s_{N_i}(t))$ via a data-driven regression method as proposed in [15]. To this end, a single hidden layer ML model, hereafter referred to as an *HJB model*, is constructed at the UAV u_i . Its input $s_{N_i}(t)$ is fed to M_H hidden nodes with a given activation function $\sigma_H(\cdot)$, which are fully connected to the model output $\hat{\psi}_i(t)$ through a weight vector $w_{i,H}(t)$, i.e.,

$$\hat{\psi}_i(t) = w_{i,H}(t)^\top \sigma_H(s_{N_i}(t)). \quad (12)$$

The model is trained by adjusting $w_{i,H}(t)$ per each observation $s_{N_i}(t)$, so as to minimize its cost function $L_{i,H}(t)$ comprising a loss function $\ell_{i,H}(t)$ and a regularizer $R_i(t)$:

$$L_{i,H}(t) = \underbrace{\frac{1}{2} \left| \hat{H}(\hat{\psi}_i(t); s_{N_i}(t)) \right|^2}_{\ell_{i,H}(t)} + \underbrace{c_H \max \left\{ 0, s_i(t)^\top \frac{ds_i(t)}{dt} \right\}}_{R_i(t)}, \quad (13)$$

where c_H is a positive constant. The loss function is intended to minimize $\hat{H}(\hat{\psi}_i(t); s_{N_i}(t))$ in (7). The regularizer is meant to stop the movement when reaching the destination, i.e., $s_i(T) = [r_i(T)^\top, v_i(T)^\top]^\top = 0$. At time t , u_i 's HJB learning control is given as below.

Algorithm 3. HJB Learning Control

- 1) Collect the states $s_{-i}(t)$ from $(N_i(t) - 1)$ UAVs.
- 2) Update the weight $w_{i,H}(t)$ as:
$$w_{i,H}(t) = w_{i,\psi}(t - \Delta t) - \mu \text{sign}(\nabla_w \ell_{i,H}(t)) - c_H \nabla_w R_i(t).$$
- 3) Calculate the value $\hat{\psi}_i(t) = w_{i,\psi}(t)^\top \sigma_\psi(s_{N_i}(t))$.
- 4) Take the optimal action $a_i^*(t) = -\frac{1}{2c_2} B^\top \nabla \psi_i(t)$.

The weight update rule in 2) of Algorithm 3 is derived by applying a normalized gradient descent algorithm (NGD), modified from the gradient descent algorithm (GD) in order to avoid saddle points under non-convex loss functions [16]. To be specific, the weight update rule under GD with the step size $\mu > 0$ is $w_{i,H}(t) = w_{i,H}(t - \Delta t) - \mu \nabla_w L_{i,H}(t)$, which is modified as $w_{i,H}(t) = w_{i,H}(t - \Delta t) - \mu \text{sign}(\nabla_w L_{i,H}(t))$ under the original NGD [16] with $\text{sign}(x) = x/\|x\|$. As opposed to this, the weight update rule in Algorithm 3 applies the sign operation only to the loss function $\ell_{i,H}(t)$ in $L_{i,H}(t)$, in order not to disturb $R_i(t)$ activations as detailed next.

The regularizer $R_i(t)$ aims to ensure stably reaching the destination without further movement, i.e., the terminal zero-state convergence $s_i(T) = 0$. With this, $R_i(t)$ becomes activated, i.e., $R_i(t) > 0$, for penalizing the loss function $\ell_{i,H}(t)$, when

the state change direction (the sign of $ds_i(t)/dt$) under the current control is the same as the current state direction (the sign of $s_i(t)$), i.e., $s_i(t)^\top ds_i(t)/dt > 0$. Otherwise, the current control is capable of stabilizing the state, and the regularizer is thus inactivated, i.e., $R_i(t) = 0$. The regularizer activations during UAV travels will be discussed in Sec. V.

In the loss function $\ell_{i,H}(t)$, the expression $\hat{H}(\hat{\psi}_i(t); s_{N_i}(t))$ is derived by applying $\hat{\psi}_i(t)$ to (7) with the same procedure as described in Algorithm 1, except for the following detail. The cost function $L_{i,H}(t)$ includes $ds_i(t)/dt$; namely, within $R_i(t)$ in (13) as well as $\ell_{i,H}(t)$ that contains

$$\partial_t \hat{\psi}_i(t) = \left(\frac{ds_i(t)}{dt} \right)^\top \nabla \hat{\psi}_i(t). \quad (14)$$

According to (5), this term introduces $dW_i(t)/dt$ that is computationally intractable. Instead, following [15], we apply the nominal state dynamics without random wind perturbations $ds_i(t) = (As_i(t) + B(a_i(t) + c_0 v_o))dt$ when calculating $ds_i(t)/dt$.

B. MFG Control Learning

In a similar vein to Algorithm 3, MFG learning control exploits ML to approximate the solutions of the HJB $H(\psi_i^*(t); s_i(t), m(t)) = 0$ and the FPK $F(m(t); s_i(t), \psi_i^*(t)) = 0$ induced by MFG control in Sec. III-B as the minima of $H(\hat{\psi}_i(t); s_i(t), \hat{m}(t))$ and $F(\hat{m}(t); s_i(t), \hat{\psi}_i(t))$. To this end, each UAV constructs two separate ML models: the HJB model used in Algorithm 3 and an *FPK model*, minimizing $H(\hat{\psi}_i(t); s_i(t), \hat{m}(t))$ (see (8) with $\hat{\psi}_i(t)$ and $\hat{m}(t)$) and $F(\hat{m}(t); s_i(t), \hat{\psi}_i(t))$ (see (10) with $\hat{\psi}_i(t)$ and $\hat{m}(t)$), respectively. The FPK model has the same structure with M_F hidden nodes, and produces the approximated MF distribution $\hat{m}(t)$ by adjusting its weight vector $w_{i,F}(t)$, i.e.,

$$\hat{m}(t) = w_{i,F}(t)^\top \sigma_F(s_i(t)). \quad (15)$$

Per each observation $s_i(t)$, the FPK model is trained by adjusting $w_{i,F}(t)$ so as to minimize the cost function $L_{i,F}(t)$:

$$L_{i,F}(t) = \frac{1}{2} |F(\hat{m}(t); s_i(t), \hat{\psi}_i(t))|^2. \quad (16)$$

The HJB model's cost function is the same as (13), except for replacing its $H(\hat{\psi}_i(t); s_{N_i}(t))$ with $H(\hat{\psi}_i(t); s_i(t), \hat{m}(t))$. At time t , UAV u_i 's MFG learning control is described as Algorithm 4.

Algorithm 4. MFG Learning Control

For $k \in [1, K]$:

- 1) Update the weight $w_{i,H}^{[k+1]}(t)$ as:
$$w_{i,H}^{[k+1]}(t) = w_{i,H}^{[k]}(t) - \mu \text{sign}(\nabla_w \ell_{i,H}^{[k]}(t)) - c_H \nabla_w R_i^{[k]}(t).$$
- 2) Calculate the value $\hat{\psi}_i^{[k]}(t) = w_{i,\psi}^{[k]}(t)^\top \sigma_\psi(s_i(t))$.
- 3) Update the weight $w_{i,F}^{[k+1]}(t)$ as:
$$w_{i,F}^{[k+1]}(t) = w_{i,F}^{[k]}(t) - \mu \text{sign}(\nabla_w L_{i,F}^{[k]}(t)).$$
- 4) Obtain the MF distribution $\hat{m}^{[k]}(t) = w_{i,F}^{[k]}(t)^\top \sigma_F(s_i(t))$.
- 5) Iterate 1-4) until $k = K$.
- 6) Take the optimal action $a_i^*(t) = -\frac{1}{2c_2} B^\top \nabla \psi_i^{[K]}(t)$.

Initial MF distribution $\hat{m}^{[0]}(t)$ at $k = 0$:

- If $t = 0$, $\hat{m}^{[0]}(0) = 1/N \sum_{i=1}^N \mathbf{1}_{s_i(t)}$, computed by collecting the states $\bar{s}_i(0)$ from N UAVs.
- Otherwise, $\hat{m}^{[0]}(t) = \hat{m}^{[K-1]}(t - \Delta t)$.

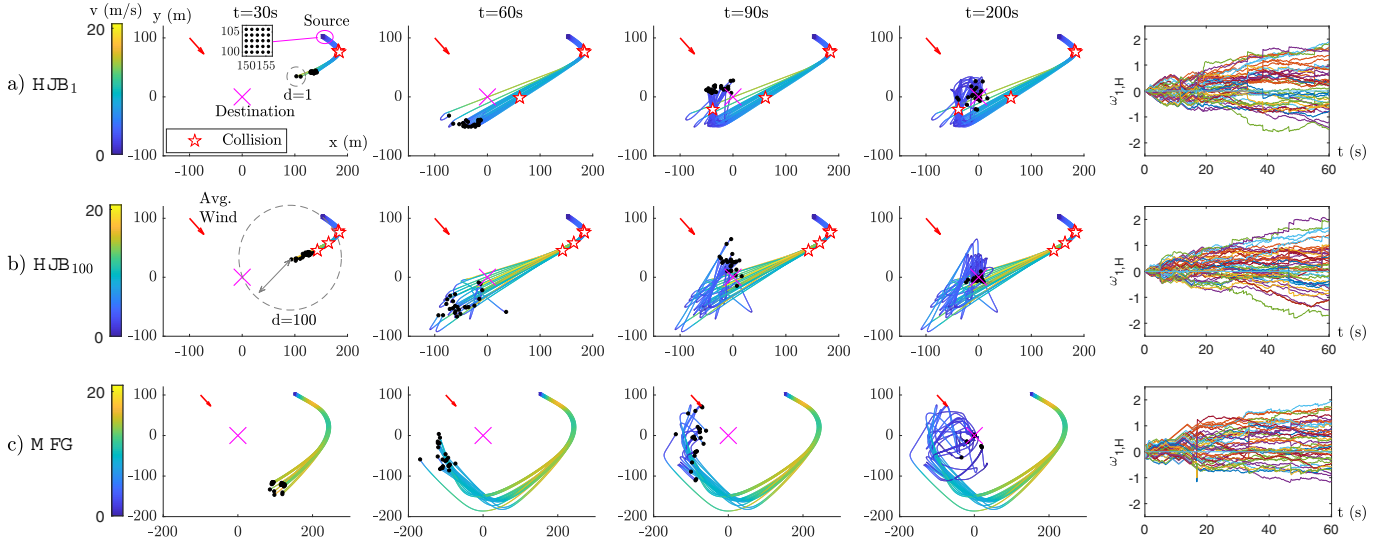


Fig. 3. Trajectory snapshots (left, 4 subplots for each control method) of 25 UAVs under (a) HJB_1 : HJB learning control with the communication range $d = 1\text{m}$ (i.e., $P = 10^{-3}\text{mW}$), (b) HJB_{100} : HJB learning control with $d = 100\text{m}$, (i.e., $P = 10\text{mW}$), and (c) MFG: MFG learning control. During the travel time $t = 0 \sim 200\text{s}$, MFG shows the best flocking behavior and the most stable HJB model parameters $w_{1,H}$ (rightmost subplot for each control method) of a randomly selected reference UAV u_1 . Consequently, MFG yields no collision during its entire travel, in sharp contrast to HJB_1 and HJB_{100} .

V. NUMERICAL RESULTS

In this section, we numerically compare the performances of HJB and MFG learning controls, in terms of travel time, energy consumption, and collision avoidance. For each travel, N UAVs are dispatched to the origin from the source that is a square centered at $(150, 100)$ in meters. At the source, each UAV is separated $\sqrt{2}\text{m}$ away from each other (see Fig. 3-a), and its velocity is solely determined by the wind dynamics with $V_o = 0.1I$ and $v_o = (1, -1)$ in m/s. Under MFG learning control, hereafter denoted as MFG, all UAVs are assumed to exchange their states at the source. Under HJB learning control, before every control, each UAV exchanges its state with the UAVs within the communication range d meter, henceforth referred to as HJB_d , without incurring interference via frequency division multiple access (FDMA).

For an HJB or MFG model, following [15], a single hidden layer model is constructed, wherein each hidden node's activation function corresponds to each non-scalar term in a polynomial expansion. The polynomial is heuristically chosen as: $(1 + x_i(t) + v_{x,i}(t))^6 + (1 + y_i(t) + v_{y,i}(t))^6$ for $\sigma_H(s_i(t))$ and $(1 + x_i(t) + v_{x,i}(t) + y_i(t) + v_{y,i}(t))^4$ for $\sigma_F(s_i(t))$, where $r_i(t) = [x_i(t), y_i(t)]^T$ and $v_i(t) = [v_{x,i}(t), v_{y,i}(t)]^T$. Compared to sigmoidal activations, polynomial activations enables smaller model sizes (i.e., $M_H = 54$, $M_F = 69$), yet the models are known to be less robust against unseen state observations. Optimizing the model architecture is an interesting topic for future research. Other simulation parameters are summarized as follows: $\Delta t = 1\text{s}$, $\alpha = 2$, $\sigma^2 = 10^{-2}\text{mW}$, $\theta = -10\text{dB}$, $c_0 = 0.1$, $c_1 = c_2 = 0.15$, $c_3 = 0.5$, $c_H = 0.5$, $\varepsilon = 0.001$, $\mu = 0.01$, and $w_{i,H}(0) = w_{i,F}(0) = 0$.

Fig. 3 visualizes the trajectories of 25 UAVs under HJB_1 , HJB_{100} , and MFG. During the entire travel, UAVs under HJB_1 hardly communicate with each other. This makes their trajectories almost identical, causing frequent collision, where a collision is counted for an inter-UAV distance less than 0.1m . Focusing on HJB_{100} , and MFG, at the beginning, all UAVs tend to follow the average wind direction to save motion energy, and then turn towards the destination. At this north-

eastern turning point, HJB_{100} fails to avoid collision due to its less trained HJB model. By contrast, MFG incurs no collision thanks to the locally iterated training operations between the HJB and FPK models (see K iterations in Algorithm 4), yielding its more trained HJB (i.e., less variance in weight parameters), as observed in the rightmost subplot of Fig. 3-c. After the turning point, there is a long-distance flight of a UAV fleet. MFG shows the highest flight velocity owing to its better flocking, which partly compensates the longer travel distance for guaranteeing collision avoidance. Finally, at the last part of the travel, UAVs tend to hover around the destination in order to stop their movement while reaching the destination (i.e., $v_i(T) = r_i(T) = 0$), which is detailed next.

Fig. 4 illustrates the accumulated number of regularizer $R_i(t)$ activations (see the details in Sec. IV-A) in the HJB models of HJB_1 , HJB_{100} , and MFG as time elapses. For all controls, $R_i(t)$ is more frequently activated near the destination (i.e., $t \geq 100\text{s}$) so as to reduce the velocity, thereby avoiding excessive hovering around and/or passing by the destination. Note that a better flocking behavior (i.e., lower inter-UAV velocities without collision) enables a more stable control without the regularization. For this reason, MFG achieving the best flocking behavior shows the least number of $R_i(t)$ activations. With more UAVs, MFG yields less frequent $R_i(t)$ activations. This is because the MF approximation (see Sec. III-B) becomes more accurate as the number of UAVs increases, providing its better flocking behavior earlier.

Lastly, Fig. 5 compares the communication, computation, and motion energy consumptions of HJB_{100} and MFG during the entire travel. Each energy is averaged over UAVs, and is normalized by the energy of HJB_1 . We consider that communication, computation, and motion energy consumptions are proportional to the number of state exchanges, the number of gradient calculations, and $\|v_i(t)\|^2 + \|a_i(t)\|^2$, respectively. Focusing on communication energy, MFG exchanges UAV states only once at the source, whereas HJB_{100} does it for every observation. Therefore, MFG consumes significantly less energy, irrespective of the number of UAVs, as opposed to HJB_{100} whose energy increases with the number of communi-

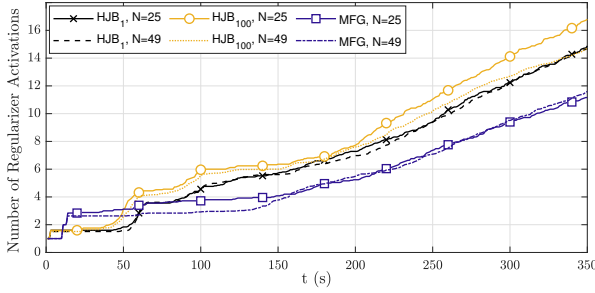


Fig. 4. Accumulated number of regularizer $R_i(t)$ activations over time under HJB_1 , HJB_{100} , and MFG ($N = \{25, 49\}$).

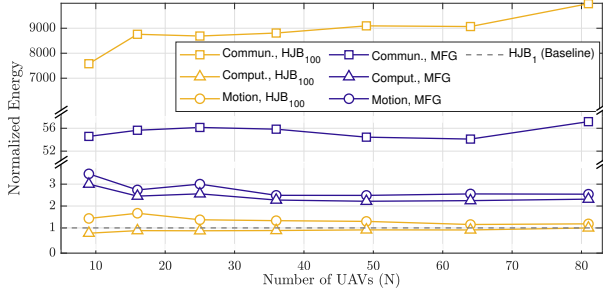


Fig. 5. Comparison of communication, computation, and motion energy between HJB_{100} and MFG, where each energy is normalized by the energy of HJB_1 ($N = \{9, 16, 25, 36, 64, 81\}$).

ating UAVs. Next, motion energy is proportional to the travel distance. As MFG yields its longer travel distance for avoiding collision, it consumes more motion energy. For computation energy, it is also proportional to the travel distance under online learning. Besides, in contrast to HJB_{100} having only an HJB model, MFG performs gradient calculations for both HJB and FPK models, which makes MFG consume more computation energy.

VI. CONCLUSION

To control massive autonomous UAVs, in this work we proposed MFG learning control algorithm that enables each UAV's real-time acceleration control in a distributed manner, by training and exploiting HJB and FPK ML models in an online way. Our simulation validated that MFG learning control guarantees collision avoidance with low communication energy, at the cost of a slight increase in computation and motion energy, compared to a baseline scheme, HJB learning control. The effectiveness of MFG learning control hinges on the level of the HJB and FPK model training. Collaborative

HJB and FPK model training across UAVs via federated learning frameworks [17] could thus be an interesting topic for future work.

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] M. Huang, P. E. Caines, and R. P. Malhamé, "Large-population cost-coupled LQG problems with nonuniform agents: individual-mass behavior and decentralized ϵ -Nash equilibria," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1560–1571, Sep. 2007.
- [5] J.-M. Lasry and P.-L. Lions, "Mean field games," *Japan. J. Math.*, vol. 2, no. 1, pp. 229–260, Mar. 2007.
- [6] R. Courant, K. Friedrichs, and H. Lewy, "On the partial difference equations of mathematical physics," *IBM J. Res. Dev.*, vol. 11, no. 2, pp. 215–234, Mar. 1967.
- [7] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1647–1650, Aug. 2016.
- [8] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 898–912, Dec. 2003.
- [9] S. Ragi and E. K. P. Chong, "UAV path planning in a dynamic environment via partially observable Markov decision process," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2397–2412, Oct. 2013.
- [10] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," to appear in *IEEE Commun. Surveys Tuts*.
- [11] R. Zárate-Minano, F. M. Mele, and F. Milano, "SDE-based wind speed models with Weibull distribution and exponential autocorrelation," in *Proc. IEEE PESGM*, Boston, MA, USA, 2016.
- [12] H. Inaltekin, M. Gorlatova, and M. Chiang, "Virtualized control over fog: Interplay between reliability and latency," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5030–5045, Dec. 2018.
- [13] M. Nourian, P. E. Caines, and R. P. Malhamé, "Mean field analysis of controlled Cucker-Smale type flocking: Linear analysis and perturbation equations," in *Proc. IFAC WC*, Milan, Italy, Aug. 2011.
- [14] F. Cucker and J.-G. Dong, "Avoiding collisions in flocks," *IEEE Transac. Automat. Contr.*, vol. 55, no. 5, pp. 1238–1243, May 2010.
- [15] 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.
- [16] R. Murray, B. Swenson, and S. Kar, "Revisiting normalized gradient descent: Fast evasion of saddle points," to appear in *IEEE Trans. Automat. Contr.*
- [17] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE (under minor revision)* [Online]. ArXiv preprint: <https://arxiv.org/abs/1812.02858>.