2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

A Mean Field Game Approach to Swarming Robots Control

Zhiyu Liu, Bo Wu and Hai Lin

Abstract— **Controlling the collective behavior through individuals of swarming robots to accomplish the tasks which are beyond the capability of an individual robot is an important yet challenging problem for both research and industry communities, since the relationship between individual behavior and collective behavior for swarming robots still remain unclear. To bridge this gap, mean field game is applied using two coupled PDEs where the backward equation is used to understand the behavior of individual robots while the forward equation governs the evolution of the robots' distribution. The proposed approach aims to provide an optimal control strategy for each individual such that the desired swarm distribution generated by random forest regression can be tracked. The discrete mean field game is solved numerically to find the optimal control where the running cost is learned instead of directly given. Our proposed approach is illustrated by an example with the data from NYC taxi trips.**

I. INTRODUCTION

With the development of robotics, communication and artificial intelligence, autonomous robots are playing increasingly crucial roles in many applications such as search and rescue, 3D mapping, manufacturing, military operations and transportation systems. The amount of autonomous robots is experiencing a tremendous growth and will keep up with the trend in the foreseeable future [1]. This growth makes it possible to employ a swarm of robots to accomplish the tasks which beyond the capability of an individual robot. Thus, studying and manipulating the emerging behavior of swarming robots via controlling individual robots is an important but challenging problem for both academia and industry. For example, with the major progress on driver-less vehicles, future intelligent transportation systems equipped with autonomous vehicles will be able to serve passengers more efficiently with even fewer vehicles by sharing [2]. Maintaining desired traffic density under time-varying traveling demands by controlling individual vehicles would significantly improve the system efficiency.

Two formal steps need to be considered for controlling swarming robots [3], namely, the generation of a swarm condition which describes the specifications of collective behavior and guides the pattern evolution of a swarm of robots, and the fabrication of a set of behaviors such that the swarm condition can be satisfied. The specifications of collective behavior over space and time are non-trivial to obtain for a large swarm since pattern evolution for large scale systems can be really complicated over time. To tackle this problem, we apply the random forest regression [4] to extract the desired pattern from data directly.

To implement a set of control policies for a large swarm of robots such that the specifications of collective behavior can be satisfied, researchers consider two aspects of this problem, namely macroscopic level and microscopic level. Macroscopic level focuses on the system's global pattern evolution and microscopic level models explicitly each individual agent's dynamic and local interactions. However, modelling and controlling the collective behavior through individuals of the swarm is a challenging problem [5] since the relationship between microscopic level (individual dynamics) and macroscopic levels (collective behavior) for swarming robots still remain unclear due to the huge amount of interactions among agents and the environment [1].

Existing work addressing this problem includes consensus based approach which studies the group behavior where all agents asymptotically reach a certain common agreement through a locally distributed protocol [6]. However, to ensure convergence and connectivity, communication among agents are required which leads to high communication and computational complexity for systems with a large number of agents [7], [8]. A probabilistic finite state machine is used for individual robots to make decisions based on its sensor inputs and internal memory in [9]. Borrowing ideas from physics, virtual physics-based schemes introduce the concept of artificial potential field [10] where robots are driven by virtual forces coming from the environment. Generally, these bottom-up approaches, including evolutionary robotics [11] and reinforcement learning [12], cannot guarantee to achieve the desired global behavior. A top-down design is proposed in [13] based on prescriptive modeling which can guide the desired system to the actual implementation, however, it uses a tedious iterative method based on trial and error. A statistical physics oriented framework is introduced in [14] where a Langevin equation is used to model the motion of robots and a Fokker-Planck equation describes the dynamic of the swarm, however, the method requires careful design and tuning as well.

Mean field game, first independently introduced by [15] and [16], becomes a promising candidate to bridge the gap between macroscopic level and microscopic level for swarming robots. It provides two coupled forward-backward partial differential equations (PDE) to model the changes of distribution of agents and the optimal cost for individuals via forward Kolmogorov equation and backward Hamilton-Jacobi-Bellman (HJB) equation respectively. Intuitively speaking, individuals anticipate the swarm distribution

The partial support of the National Science Foundation (Grant No. CNS-1446288, ECCS-1253488, IIS-1724070) and of the Army Research Laboratory (Grant No. W911NF- 17-1-0072) is gratefully acknowledged.

Zhiyu Liu, Bo Wu and Hai Lin are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 46556 USA. zliu9@nd.edu, bwu3@nd.edu, hlin1@nd.edu

via mean field term and obtain their strategies by solving HJB equation and reaching the Nash equilibrium while the mean field evolves according to these strategies governed by Kolmogorov equation [17]. Unlike other methods using Kolmogorov equation to model swarm distribution [14], the solution of Kolmogorov equation here depends on the solution of HJB equation as well. Most existing work on mean field game use relatively simple model for cost function, however, the cost function of motion planning for large scale systems is more complicated since agents are deeply coupled with its neighbors and environment, for example, vehicles in the dense urban environment. Therefore, it is more reasonable to model the cost function by learning it from the data directly.

Motivated by the challenges in swarming robots, we propose a data-driven approach for swarming robots control using mean field game. The main contributions of this paper are threefold. First, the desired swarm distribution and the running cost function for mean field game are learned from real world data. Second, mean field game is used to model the individual robot's behavior and the connection with the global emerging behavior for swarming robots. Third, the proposed algorithm is tested and verified with the real world data. Even though we use NYC taxi trips data to test our theory where the taxi is not autonomous in practice, our proposed framework can be readily generalized to swarming robots with autonomous agents.

The rest of paper is organized as follows. Section II introduces the related work. Section III formulates the swarm pattern control problem via mean field game. Section IV proposes the mean field game approach to solve the problem. An example and conclusion are presented in section V and section VI respectively.

II. PRELIMINARIES

A. Mean Field Game

Mean field game studies the limiting behavior of large scale systems by modeling the interaction between individual agents and the distribution of agents. It becomes a powerful tool to study the collective behavior of multi-agent systems where the number of agents approaches to infinity by building an interface between the macroscopic level and microscopic level through two coupled PDEs where the HJB equation is used to understand the behavior of individual agents over time which is partially influenced by the distribution of all agents and the Kolmogorov equation governs the evolution of the agents' distribution.

For a multi-agent system consisting of an infinite number of rational and homogeneous agents with unique labels $\mathcal{P} = \{1, 2, \dots, +\infty\}$, the agent dynamics is governed by the following stochastic differential equation (SDE).

$$dX_t^i = \alpha_t dt + \sigma dW_t^i, X_0^i = x_0^i, \forall i \in \mathcal{P}, \tag{1}$$

where $X_t^i \in \mathbb{R}^2$ are the states of agent i which are GPS coordinates in this case, $\alpha_t \in \mathbb{R}^2$ are control inputs and noise W_t^i is standard Brownian motion. The individual cost function J_i for agent i includes two parts, the running cost $f(\alpha_t, X_t^i, m_t)$ and the terminal cost $g(X_T^i, m_T)$.

$$J_i(\alpha_t) = \mathbb{E}[\int_0^T f(\alpha_t, X_t^i, m_t)dt + g(X_T^i, m_T)], \tag{2}$$

where m_t is the probabilistic distribution of agents at time t which is also known as the mean field term and T is the planning horizon. The optimization goal is to find optimal control strategies α_t^* in the sense that a Nash equilibrium can be achieved, where every players cannot improve its award by unilaterally changing its strategy.

Define value function $u(t_0, x_0) = \underset{\alpha_t}{inf}\mathbb{E}[\int_{t_0}^T f(\alpha_t, X_t^i, m_t)dt + g(X_T^i, m_T)]$. Based on the definition of the cost function (2), $u(t_0, x_0)$ is the optimal cost at t_0. It is well known that $u(t_0, x_0)$ is also the solution of the following HJB equation [18].

$$\partial_t u + \frac{\sigma^2}{2}\Delta u + H(t, x, \nabla u, m) = V[m], u|_{t=T} = g(m_T) \tag{3}$$

where the Hamiltonian H is defined as the Legendre transform of the running cost f.

$$H(\nabla u) = \underset{\alpha}{sup}[\alpha \nabla u - f(\alpha)], \tag{4}$$

and operator V maps the probability measures into a bounded set of Lipschitz functions [15]. The optimal control input is given by

$$\alpha_t^* = \partial_p H(t, x, \nabla u, m). \tag{5}$$

As for the macroscopic level, by applying Ito's Lemma, the evolution of the system's probabilistic distribution with the optimal control inputs generated from the HJB equation yields the following Kolmogorov equation [17].

$$\partial_t m - \frac{\sigma^2}{2}\Delta m + div(m\partial_p H(t, x, \nabla u, m)) = 0, m|_{t=0} = m_0, \tag{6}$$

where m_0 denotes the initial agents probabilistic distribution.

The two coupled PDEs demonstrate a forward-backward structure which is the core of mean field game since it serves as an interface between macroscopic level and microscopic level. Agents governed by the SDE (1) in microscopic level anticipate the swarm distribution to evaluate their cost function (2). Then they deduce their control inputs backward by solving the HJB equation (3) (similar to dynamic programming). Finally, the distribution in macroscopic level evolves according to these strategies described by the Kolmogorov equation (6). Notice that the forward-backward structure makes the coupled PDEs difficult to solve which will be addressed in Sec. IV.

B. NYC Taxi Model and Trips Data

We assume taxis can move in free space. Traffic constraints like roadways and density will be considered in the running cost function learned from NYC taxi trips data which is open to the public with over 1.8 billion taxi trips from January 2009 up to now [19]. Each individual trip record contains precise GPS trajectories and time stamps of where and when the trip started and ended, plus a few other variables such as fare amount and distance traveled. This dataset makes it possible to extract specifications on global traffic pattern and predict the running cost in mean field game.

III. Problem Statement

To bridge the gap between macroscopic level and microscopic level for swarming robots, we are interested in finding control policies for each robot using mean field game such that the overall pattern can evolve in the desired way learned from historical data. Given SDE (1) for agent dynamics, cost function J_i (2) for individual robots, where $f(\alpha_t, X_t^i, m_t)$ is the unknown running cost and $g(X_T^i, m_T)$ is the given terminal cost function, and NYC taxi trips data introduced in Section II, we formulate the problem as a mean field game planning problem which learns the running cost function from data and find control strategies α_t^* for each robot, such that the Nash equilibrium can be reached and the robot distribution over time can satisfy the given specifications generated from the regression model.

We divide the environment into $N \times N$ grids uniformly. Denote n, $n \in [0, 1, ..., N_T]$, as the discrete time where $N_T = \frac{T_H}{T}$ and T_H is the overall planning horizon, and $\widetilde{M}_{i,j}^n$ as the robot density in grid (i, j) at time n. Given historical data, we apply random forests regression to predict the robot density $\widetilde{M}_{i,j}^n$ where the inputs for the regression model are agent location $(i, j), i, j \in [1, 2, ..., N]$ and time n.

The running cost function $f(\alpha_t, X_t^i, m_t)$ depends on robots' states X_t^i, robots distribution m_t and control input α_t. Most existing work on mean field game simply treat the running cost as a quadratic function $f = \frac{|\alpha_t|^2}{2}$ which cannot capture the true running cost in a complex large swarm, where the interaction among robots and environment is complicated to model and yet significant enough to not to be ignored. To tackle this problem, we define the running cost function as $f = \frac{|\alpha_t|^2}{2} f_t'$, where $f_t' \in \mathbb{R}^+$ is a constant representing the running cost determined by mean field and agents' location at time t, which can be obtained by learning traveling time between two locations via random forests regression.

Now we can use mean field game to find control strategies such that the desired swarm distribution can be tracked. At each interval $[n, n+1]$, we use mean filed game to find the control strategies for individual robots at the current time interval. The initial distribution m_0^n is the robots' distribution at n and the terminal distribution m_T^n is obtained from the regression model in the previous step.

$$m_T^n = \frac{1}{\sum_{i,j}^N \widetilde{M}_{i,j}^n} \begin{bmatrix} \widetilde{M}_{1,1}^n & \widetilde{M}_{1,2}^n & \cdots & \widetilde{M}_{1,N}^n \\ \widetilde{M}_{2,1}^n & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \widetilde{M}_{N,1}^n & \cdots & \cdots & \widetilde{M}_{N,N}^n \end{bmatrix} \tag{7}$$

Problem 1: Given a swarm of robots where robot dynamics is governed by SDE (1), we aim to find the optimal control strategy α_t^* for individual robots by solving the following mean field game planning problem such that their cost functions in (2) where the running cost functions are learned from data can be minimized by reaching a Nash equilibrium and the desired swarm distribution modeled by random forest regression can be tracked at the same time

$$\begin{aligned} &\forall n \in [0, 1, ..., N_T - 1], \\ &\partial_t u^n + \frac{\sigma^2}{2}\Delta u^n + H(t, x, \nabla u^n, m^n) = V[m^n], \\ &\partial_t m^n - \frac{\sigma^2}{2}\Delta m^n + div(m^n \partial_p H(t, x, \nabla u^n, m^n)) = 0, \\ &m^n|_{t=0} = m_0^n, m^n|_{t=T} = m_T^n, m^n > 0, \int m^n dx = 1, \\ &H(t, x, \nabla u^n, m^n) = \frac{1}{2 f_n'}|\nabla u^n|^2. \end{aligned} \tag{8}$$

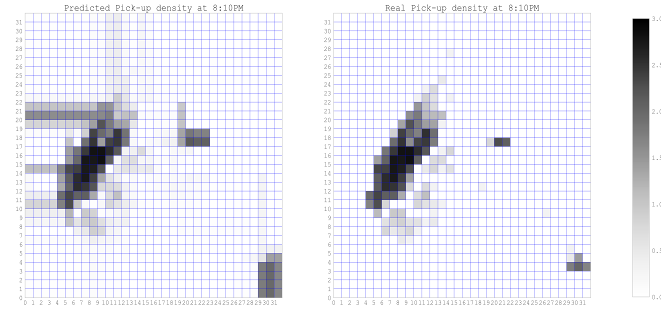

Fig. 1. Predicted and real demand density at 8:10 PM

IV. Pattern Control via Discrete Mean Field Game

A. Desired Pattern Learning

We use NYC taxi trips data set to train the regression model $\widetilde{M}_{i,j}^n$ and predict the demand density every ten minutes in 24 hours by setting $N_T = 24 \times 6$, $T = 10$min and $N = 32$. To test the accuracy of the regression algorithm, we split the dataset into 80% training set and 20% testing set where the training set is employed to train the random forest regression model and the testing set is used to test the accuracy for the model generated by training step. We use the coefficient of determination R^2, ranging from 0 to 1 with 1 being the most accurate, to test the accuracy of the regression algorithm. The R^2 for the training set is 0.9972 and for the testing set is 0.9802 which shows the accuracy of random forests regression. Fig. 1 shows the predicted distribution and the real distribution of demand at 8:10 PM in NYC where a higher value in each grid represents a higher demand. As we can see from these figures (both plotted in logarithmic scale for a better view), Manhattan area has a higher demand for taxi around 8 PM and the learning method is capable of predicting this pattern accurately. The obtained regression model $\widetilde{M}_{i,j}^n$ will be used to generate desired terminal pattern m_T for mean field game.

B. Running Cost Function Learning

With the information of traveling time in the given data set, it is possible to learn a more practical running cost

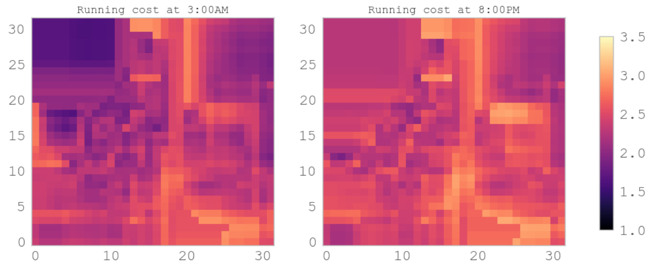

Fig. 2. Running cost at different time

function from data directly, which can be obtained by learning traveling time between two locations via random forests regression and set the predicted traveling time as the constant f_t'. Examples of traveling cost f_t' for the whole environment at different time are given in Fig. 2 where the left figure shows the traveling cost at 3 AM and the right one represents the cost at 8 PM. Comparing the two figures, we can observe that the traveling cost is higher in rush hour.

C. Numerical Solutions for mean field game

The forward-backward structure is the core of mean field game, however, classic methods cannot solve coupled PDEs with forward-backward structure. A numerical method for solving mean field game is presented in [20] using finite difference schemes. To apply finite difference schemes on (8), we uniformly divide the environment into $N \times N$ grids, where the size of each grid is $h \times h$. Agents in grid (i, j) has the same state $x_{i,j}$ which is the GPS coordinate of the center of grid (i, j). Denote $\Delta t = \frac{T}{K}$, where K is the planning horizon for discrete mean field game. The value of u^n and m^n at $x_{i,j}$ at step p, $p \in [0, 1, ..., K]$, are denoted by $U_{i,j}^{n,p}$ and $M_{i,j}^{n,p}$. We also define the following discrete operators.

$$D_x U_{i,j}^{n,p} = \frac{U_{i+1,j}^{n,p} - U_{i,j}^{n,p}}{h}, D_y U_{i,j}^{n,p} = \frac{U_{i,j+1}^{n,p} - U_{i,j}^{n,p}}{h} \quad (9)$$

$$\Delta_h U_{i,j}^{n,p} = -\frac{1}{h^2}(4U_{i,j}^{n,p} - U_{i+1,j}^{n,p} - U_{i-1,j}^{n,p} - U_{i,j+1}^{n,p} - U_{i,j-1}^{n,p}) \quad (10)$$

$$D_h U_{i,j}^{n,p} = (D_x U_{i,j}^{n,p}, D_x U_{i-1,j}^{n,p}, D_y U_{i,j}^{n,p}, D_y U_{i,j-1}^{n,p})^T \quad (11)$$

Numerical Hamiltonian $g(x, q_1, q_2, q_3, q_4)$ is used to approximate the Hamiltonian H. To guarantee the existence and uniqueness properties for the solution of mean field game, $g(x, q_1, q_2, q_3, q_4)$ should satisfy the following properties [20]. 1) Consistency: $g(x, q_1, q_1, q_2, q_2) = H(x, q), \forall x, \forall q = (q_1, q_2)$; 2) Differentiability: g is of class $\mathcal{C}^1$; 3) Monotonicity: g is non-increasing with respect to q_1 and q_3 and non-decreasing with respect to q_2 and q_4; 4) Convexity: $(q_1, q_2, q_3, q_4) \mapsto g(x, q_1, q_2, q_3, q_4)$ is convex.

Proposition 1: $g(x, q_1, q_2, q_3, q_4)$ with the following form [20] satisfies the properties listed above.

$$g(x_{i,j}, D_h U_{i,j}^{n,p}) = \frac{1}{2f_n'}|\widetilde{g}(D_h U_{i,j}^{n,p})|^2 \quad (12)$$

where

$$\begin{aligned}&\widetilde{g}(q_1, q_2, q_3, q_4)\\ &= (max(-q_1, 0), max(q_2, 0), max(-q_3, 0), max(q_4, 0)).\end{aligned}$$

Proof: Consistency can be proved by the equation below with $C = \frac{1}{2f_n'}$.

$$\begin{aligned}&g(x, q_1, q_1, q_2, q_2)\\ &= C(max(-q_1, 0)^2, max(q_2, 0)^2, max(-q_3, 0)^2, max(q_4, 0)^2)\\ &= C(q_1^2 + q_2^2) = H(x, (q_1, q_2)).\end{aligned}$$

g is of class $\mathcal{C}^1$ w.r.t q_1 since the following equation holds.

$$\frac{\partial g}{\partial q_1} = C\frac{\partial}{\partial q_1}max(-q_1, 0)^2 = \begin{cases} 0, q_1 \geqslant 0 \\ 2Cq_1, q1 < 0 \end{cases}$$

The same approach can be used on proving q_2, q_3, q_4.

Monotonicity can be obtained from the proof of differentiability since $\frac{\partial g}{\partial q_1}$ and $\frac{\partial g}{\partial q_3}$ are non-positive and $\frac{\partial g}{\partial q_2}$ and $\frac{\partial g}{\partial q_4}$ are non-negative. Thus, g is non-increasing w.r.t q_1 and q_3 and non-decreasing w.r.t q_2 and q_4.

Since $\frac{\partial^2 g}{\partial q_1^2}$, $\frac{\partial^2 g}{\partial q_2^2}$, $\frac{\partial^2 g}{\partial q_3^2}$ and $\frac{\partial^2 g}{\partial q_4^2}$ are all non-negative, thus, g is convex. ■

Define discrete operators $\mathcal{B}_{i,j}(U, M)$ [21] to substitute the term $div(m\partial_p H(t, x, \nabla u, m))$ in (8).

$$\begin{aligned}&\mathcal{B}_{i,j}(U, M)\\ &= \frac{1}{h}\begin{pmatrix} M_{i,j}\frac{\partial g}{\partial q_1}(x_{i,j}, D_h U_{i,j}) - M_{i-1,j}\frac{\partial g}{\partial q_1}(x_{i-1,j}, D_h U_{i-1,j}) \\ +M_{i+1,j}\frac{\partial g}{\partial q_2}(x_{i+1,j}, D_h U_{i+1,j}) - M_{i,j}\frac{\partial g}{\partial q_2}(x_{i,j}, D_h U_{i,j}) \\ +M_{i,j}\frac{\partial g}{\partial q_3}(x_{i,j}, D_h U_{i,j}) - M_{i,j-1}\frac{\partial g}{\partial q_3}(x_{i,j-1}, D_h U_{i,j-1}) \\ +M_{i,j+1}\frac{\partial g}{\partial q_4}(x_{i,j+1}, D_h U_{i,j+1}) - M_{i,j}\frac{\partial g}{\partial q_4}(x_{i,j}, D_h U_{i,j}) \end{pmatrix}\end{aligned}$$

Now we can formulate the discrete mean field game from the original problem.

$$\frac{U_{i,j}^{n,p+1} - U_{i,j}^{n,p}}{\Delta t} - \frac{\sigma^2}{2}\Delta_h U_{i,j}^{n,p+1} + g(x_{i,j}, D_h U_{i,j}^{n,p+1}) = V[M_{i,j}^{n,p}], \quad (14)$$

$$\frac{M_{i,j}^{n,p+1} - M_{i,j}^{n,p}}{\Delta t} + \frac{\sigma^2}{2}\Delta_h M_{i,j}^{n,p} + \mathcal{B}_{i,j}(U^{n,p+1}, M^{n,p}) = 0, \quad (15)$$

$$M^{n,p} \in \left\{M_{i,j}^{n,p} : h^2 \textstyle\sum_{i,j} M_{i,j}^{n,p} = 1, M_{i,j} \geq 0\right\} \quad (16)$$

$$M_{i,j}^{n,K} = (m_T^n)_{i,j}, M_{i,j}^{n,0} = (m_0^n)_{i,j} \quad (17)$$

Remark 1: The existence and uniqueness are two important factors needed to be addressed in mean field game. The existence of discrete mean field game for planning problem is addressed in [21] using convex duality and the Fenchel-Rockafellar theorem. There exist solutions $M^{n,p}$ and $U^{n,p}$ such that (14)-(17) hold as long as the numerical Hamiltonian g satisfies the assumptions listed above and V is strictly monotone. The uniqueness discussion in [15] holds for discrete mean field game if g and V satisfy the same conditions. So with the function g given in this paper, the existence and uniqueness are guaranteed.

The forward-backward structure makes it impossible to use classic time-marching methods to solve the coupled PDEs. To tackle this problem, we define new vectors $\mathcal{U}^n$ and $\mathcal{M}^n$ with $\mathbb{R}^{(K+1)N^2}$ dimension where $\mathcal{U}_{p(N^2+i)(N+j)}^n = U_{i,j}^{n,p}$ and $\mathcal{M}_{p(N^2+i)(N+j)}^n = M_{i,j}^{n,p}$. $\mathcal{U}^n$ and $\mathcal{M}^n$ contain numerical solutions for $U_{i,j}^{n,p}$ and $M_{i,j}^{n,p}$ at all steps. Equations (14-17) can be reformulated as the following nonlinear equations where the forward-backward structure no longer exists.

$$\begin{cases} F_U(\mathcal{U}^n, \mathcal{M}^n) = 0, \\ F_M(\mathcal{U}^n, \mathcal{M}^n) = 0, \end{cases} \quad (18)$$

where

$$F_U(\mathcal{U}^n, \mathcal{M}^n) = \begin{cases} \frac{U_{i,j}^{n,p+1} - U_{i,j}^{n,p}}{\Delta t} - \frac{\sigma^2}{2}\Delta_h U_{i,j}^{n,p+1} \\ +g(x_{i,j}, D_h U_{i,j}^{n,p+1}) - V[M_{i,j}^{n,p}], \forall p \in [0, K), \end{cases}$$

$$F_M(\mathcal{U}^n, \mathcal{M}^n) = \begin{cases} \frac{M_{i,j}^{n,p+1} - M_{i,j}^{n,p}}{\Delta t} + \frac{\sigma^2}{2}\Delta_h M_{i,j}^{n,p} + \mathcal{B}_{i,j}(U^{n,p+1}, M^{n,p}), \forall p \in [0, K) \\ M^{n,K} = m_T^n, M^{n,0} = m_0^n, \forall i, j. \end{cases}$$

Newton's method is applied to solve the nonlinear equation (18). Notice that it is possible to add extra constraints on the swarm distribution under the proposed framework. For example, constraints like avoiding unsafe zone $\mathcal{X}_{unsafe}$ all the time can be expressed as $M_{i,j}^{n,p} = 0, \forall n, p, \forall (i, j) \in \mathcal{X}_{unsafe}$. These constraints can be included as extra equations in (18).

V. Example

An example of NYC taxi is studied using the proposed method to track the desired distribution under the optimal control inputs generated by mean field game. We aim to design a control strategy for empty taxi so they can find passengers more easily by adjusting the empty taxi pattern using the proposed method to track the predicted demand pattern generated by the regression model.

To describe and predict the complicated demand pattern evolution over time and space, a random forest regression model is trained to predict the demand pattern every 10 minutes in 24 hours. At time n the current empty taxi distribution m_0^n is given by the centralized intelligent transportation system. To make sure the empty taxi pattern tracks the demand pattern, mean field game is solved numerically such that m_0^n can evolve into the predicted demand pattern between $[n, n+1]$ for all $n \in [0, 1, ..., N_T - 1]$ within K steps, where $K = 6$ in our case. The example is programmed with MATLAB and Python where MATLAB is used to solve (18) with Newton method, while the training of the regression model $\widetilde{M}_{i,j}^n$ and learning the running cost function f in mean field game are implemented in Python.

Filtering and pre-processing techniques were employed to prune the data before applying them in our proposed algorithm. The trip duration, trip start time as well as coordinates of demand and empty taxis are of our concern while other fields of data were omitted. To further reduce the data size, around 10% of the trips records in Jan. 2016 were extracted with stratified sampling such that each time slot is equally represented. After deleting outliers and meaningless trips (e.g., trip time less than 20 seconds or larger than 8 hours), the remaining dataset contains 774,681 entries. With the remaining data, we were able to predict demand pattern over time and running cost function for mean field game while still keep the statistical significance of the original dataset. The training process takes about 300 seconds on a PC with Intel core i7-4710MQ 2.50 GHz processor and 8GB RAM.

Remark 2: We assume the pick-up and drop-off coordinates in the original data set represent the coordinates of demand and empty taxis respectively. We are aware that the data set cannot perfectly represent the distribution of empty taxi and traveling demand. But it is sufficient enough (large and rich enough) to test our theory such that future autonomous vehicles can pick up passengers using the proposed method where data can be obtained easily via autonomous vehicles and applications like Uber.

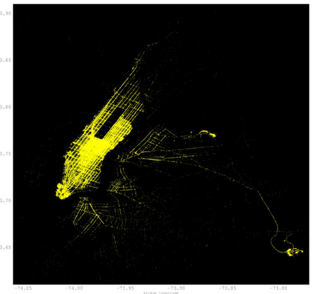

Fig. 3. Demand Pattern

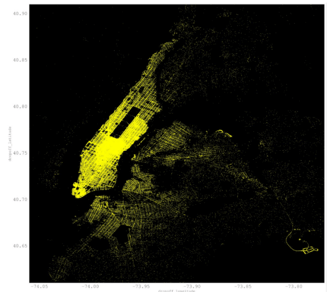

Fig. 4. Empty taxi Pattern

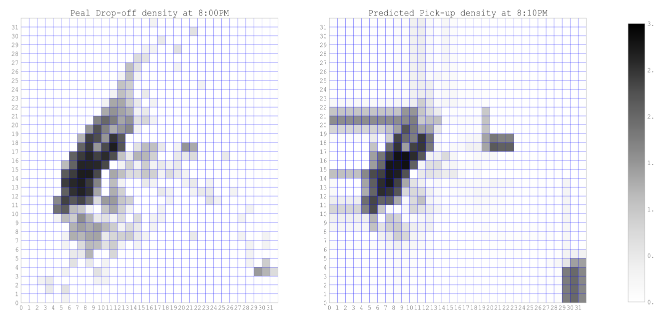

Fig. 5. m_0 & m_T density at 8:00 PM

Fig. 3 and Fig. 4 show the overall distribution of demand and empty taxi without any control design, where a clear difference between two distribution can be observed.

Fig. 5-8 show part of the results. Fig. 5 shows the actual empty taxi density at 8:00 PM on the left and the predicted demand density at 8:10 PM on the right which is the terminal density in mean field game as well. After solving equation (18) for 8:00 PM to 8:10 PM, the distribution evolves from the left to the right as predicted. Fig. 7 demonstrates this process and Fig. 6 shows the corresponding optimal value function u. As mentioned in Section II, the optimal value function $u(t_0, x_0)$ represents the optimal cost from t_0 to T. As we can see from Fig. 6, the optimal value function decreases as the distribution approaches to the terminal distribution. According to equation (5), the optimal control inputs α^* on one direction are shown in Fig. 8, where the amplitude of α^* is relatively large indicating the taxis are expected to move to places where the differences between the demand and the current distribution are big and decreases as the distribution approaches to the terminal distribution.

VI. Conclusion

We studied the control of swarming robots at both macroscopic level and microscopic level using mean field game. The desired swarm distribution is provided by random forest regression. The interaction between individual robots and the collective behavior of the swarm is modeled under mean field game framework where the running cost function is learned from real data. The proposed algorithm provides control strategies for individual robots such that the desired swarm distribution can be achieved by solving mean field game numerically. The example based on NYC taxi trips data illustrates the proposed algorithm.

REFERENCES

[1] J. G. D. S. Durand, "A methodology to achieve microscopic/macroscopic configuration tradeoffs in cooperative multi-robot systems design," Ph.D. dissertation, Georgia Institute of Technology, 2017.

[2] J. Leech, G. Whelan, and M. Bhaiji, "Connected and autonomous vehicles–the uk economic opportunity. kpmg," 2015.

[3] S. T. Kazadi, "Swarm engineering," Ph.D. dissertation, California Institute of Technology, 2000.

[4] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[6] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial informatics*, vol. 9, no. 1, pp. 427–438, 2013.

[7] M. Nourian, P. E. Caines, R. P. Malhame, and M. Huang, "Nash, social and centralized solutions to consensus problems via mean field control theory," *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 639–653, 2013.

[8] Z. Liu, B. Wu, J. Dai, and H. Lin, "Distributed communication-aware motion planning for multi-agent systems from stl and spatel specifications," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 4452–4457.

[9] O. Soysal and E. Sahin, "Probabilistic aggregation strategies in swarm robotic systems," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*. IEEE, 2005, pp. 325–332.

[10] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[11] S. Nolfi, J. C. Bongard, P. Husbands, and D. Floreano, "Evolutionary robotics." 2016.

[12] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous agents and multi-agent systems*, vol. 11, no. 3, pp. 387–434, 2005.

[13] M. Brambilla, A. Brutschy, M. Dorigo, and M. Birattari, "Property-driven design for robot swarms: A design method based on prescriptive modeling and model checking," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 9, no. 4, p. 17, 2015.

[14] H. Hamann and H. Wörn, "A framework of space–time continuous models for algorithm design in swarm robotics," *Swarm Intelligence*, vol. 2, no. 2, pp. 209–239, 2008.

[15] J.-M. Lasry and P.-L. Lions, "Mean field games," *Japanese journal of mathematics*, vol. 2, no. 1, pp. 229–260, 2007.

[16] M. Huang, R. P. Malhamé, P. E. Caines *et al.*, "Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle," *Communications in Information & Systems*, vol. 6, no. 3, pp. 221–252, 2006.

[17] A. Lachapelle and M.-T. Wolfram, "On a mean field game approach modeling congestion and aversion in pedestrian crowds," *Transportation research part B: methodological*, vol. 45, no. 10, pp. 1572–1589, 2011.

[18] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Springer Science & Business Media, 2008.

[19] *TLC Trip Record Data*, 2009. [Online]. Available: http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

[20] Y. Achdou and I. Capuzzo-Dolcetta, "Mean field games: Numerical methods," *SIAM Journal on Numerical Analysis*, vol. 48, no. 3, pp. 1136–1162, 2010.

[21] Y. Achdou, F. Camilli, and I. Capuzzo-Dolcetta, "Mean field games: numerical methods for the planning problem," *SIAM Journal on Control and Optimization*, vol. 50, no. 1, pp. 77–109, 2012.

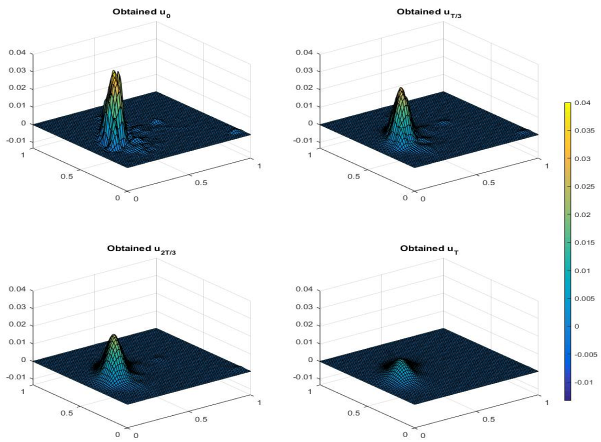

Fig. 6. Optimal cost function u at different time

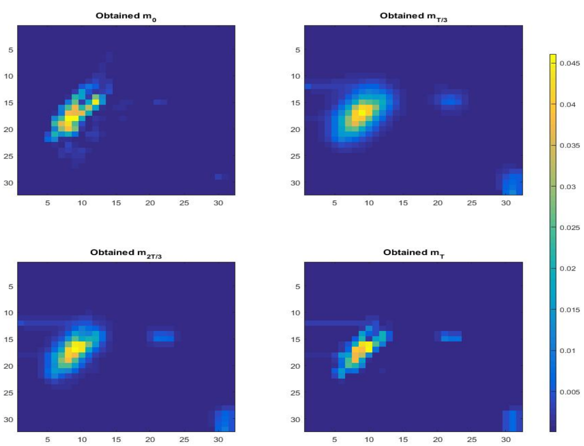

Fig. 7. Swarm distribution m at different time

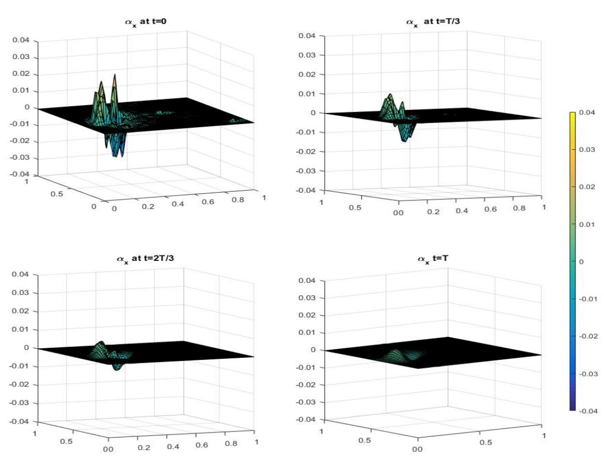

Fig. 8. Optimal control α_x^* at different time