Adaptive Optimal Decision in Multi-Agent Random Switching Systems

Mushuang Liu, Yan Wan, and Frank L. Lewis

***Abstract*—Random switching models have been widely used in areas of communication, physics and aerospace, to capture the random movement patterns of mobile agents. In this letter, we study the optimal decision-making problem for multi-agent systems governed by random switching dynamics. In particular, we develop a novel online optimal control solution that integrates the reinforcement learning (RL) with an effective uncertainty sampling method, called multivariate probabilistic collocation method (MPCM), to adaptively find the optimal policies for agents of randomly switching mobility. We also develop a novel estimator that integrates the unscented Kalman filter (UKF) and MPCM to provide online estimation solutions for these agents. Efficiency and accuracy of the proposed solutions are analyzed. A concrete communication and antenna control co-design problem for a multi-UAV network is studied in the end to illustrate and validate the results.**

***Index Terms*—Random switching systems, learning control, nonlinear estimation.**

I. Introduction

RANDOM mobility models (RMMs) [1], [2], [3], including Random Walk, Random Direction, Gauss Markov and Smooth Turn (ST), have been widely used in diverse areas to capture the random movement patterns of mobile agents. Examples include ad hoc networks in wireless communication, random motion of particles in physics, and random unmanned aircraft vehicle (UAV) mobility in aerospace. These RMMs fall under the general random switching modeling framework: at each randomly selected time point, an agent randomly selects its maneuver of certain statistical properties, and moves with the selected maneuvers until the next selected time point. Driven by the emergence of Internet-of-Things (IOT) applications, mobile agents play increasingly important roles in optimal decision processes. In this letter, we study optimal control and effective estimation for such multi-agent random switching systems.

Optimal controller design for stochastic systems has been studied in, e.g., [4]. For linear systems corrupted with additive noise, optimal controls solution that minimize the expected quadratic cost functions can be found analytically. For general stochastic systems with multi-dimensional uncertainties, simulation-based uncertainty evaluation methods need to be utilized. The Monte Carlo (MC) method and its variants including the Markov chain MC and Sequential MC have been widely used to explore the uncertainty space. However, they require a large amount of sample points, and hence, are too time-consuming to be used for online decisions. To address this challenge, paper [5] developed an effective uncertainty evaluation method, called multivariate probabilistic collocation method (MPCM), and paper [6] integrated it with reinforcement learning (RL) to effectively solve the stochastic optimal control problem online. However, the uncertainties considered in [6] do not capture complex random switching behaviors. In this letter, we integrate RL and MPCM to provide an online learning-based adaptive optimal control solution for random switching systems of highly flexible, random, and uncertain agent mobility patterns.

In practice, agents' states are not always available for controller design, and thus, effective state estimators are needed. For linear systems with additive noise, Kalman filter (KF) is the optimal estimator. For nonlinear systems, the sampling-based unscented KF (UKF) [7], [8] have been used practically. In this letter, we also describe a practical estimator for multi-agent random switching systems by integrating UKF and MPCM. A communication and control co-design problem for a multi-UAV network governed by ST mobility is studied in the end to illustrate and validate the results.

II. Modeling and Problem Formulation

A. System Model

Consider a group of N agents, each of which moves independently with a general random switching dynamics as follows. At randomly selected time points T_0^i, T_1^i, T_2^i, $\ldots$, where $0 = T_0^i < T_1^i < \cdots$, agent i randomly selects its maneuver $\mathbf{a}_i[T_l^i]$ (e.g., velocity, heading direction, or turning center, etc.), and maintains the selected maneuver until the next selected time point. The time duration for agent i to maintain

Manuscript received March 1, 2019; revised May 21, 2019; accepted June 7, 2019. Date of publication June 26, 2019; date of current version July 9, 2019. This work was supported in part by the Office of Naval Research under Grant N00014-18-1-2221, and in part by the National Science Foundation under Grant 1714519 and Grant 1839804. Recommended by Senior Editor G. Cherubini. *(Corresponding author: Yan Wan.)*

M. Liu and Y. Wan are with the Department of Electrical Engineering, University of Texas at Arlington, Arlington, TX 76019 USA (e-mail: mushuang.liu@mavs.uta.edu; yan.wan@uta.edu).

F. L. Lewis is with the UTA Research Institute, University of Texas at Arlington, Fort Worth, TX 75052 USA, and also with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: lewis@uta.edu).

Digital Object Identifier 10.1109/LCSYS.2019.2923915

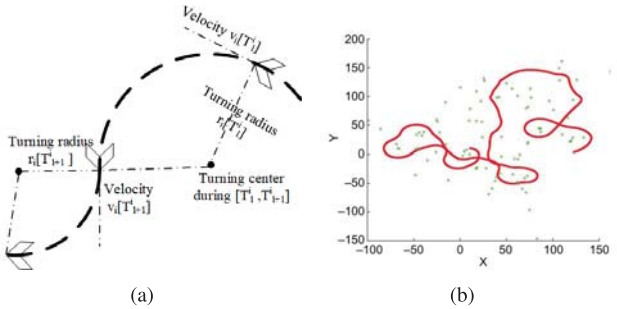

Fig. 1. Illustration of the ST RMM. (a) Maneuver selection and switching behavior. (b) A sample trajectory (red curve). Green spots are randomly chosen turning centers [2].

its current maneuver is $\tau_i[T_l^i]$, i.e., $\tau_i[T_l^i] = T_{l+1}^i - T_l^i$. Note that such a general random switching dynamics is constructed using two types of random variables. Type 1 random variable, $\mathbf{a}_i[T_l^i]$, describes the characteristics for each maneuver, and type 2 random variable $\tau_i[T_l^i]$ describes how often the switching of type 1 random variable occurs. The agent dynamics is described as

$$\mathbf{x}_i[k] = f(\mathbf{x}_i[k-1], \mathbf{a}_i[k], \tau_i[T_l^i]), \tag{1}$$

where $\mathbf{x}_i[k] \in \mathbf{R}^n$ is the system state vector of agent i at time instant k, and $f(.)$ captures the general agent dynamics. $\mathbf{a}_i[k] \in \mathbf{R}^m$ is the agent's maneuver at time k, and m is the number of uncertain parameters that describe the statistic properties of the maneuver. Each element of $\mathbf{a}_i[k]$, $a_{i,p}[k]$, where $p \in \{1, \ldots, m\}$, follows the random switching rule,

$$a_{i,p}[k] = \begin{cases} a_{i,p}[T_l^i], & \text{if } \exists l \in [0, 1, 2, ..), k = T_l^i; \\ a_{i,p}[k-1], & \text{if } \forall l = 0, 1, 2, .., k \neq T_l^i, \end{cases} \tag{2}$$

where $a_{i,p}[T_l^i](p = 1, \ldots, m)$ is the element of the type 1 random variable $\mathbf{a}_i[T_l^i]$, and $a_{i,p}[T_l^i](p = 1, \ldots, m)$ changes independently over time with pdf $f_{A_p}(a_{i,p}[T_l^i])$. The random variables $(\mathbf{a}_i[T_l^i], \tau_i[T_l^i])$ are independent for each agent i to capture their independent movement patterns.

We use a simple but widely-used UAV RMM, smooth turn RMM [1], [2], to illustrate the random switching dynamics. In the ST RMM, each agent selects a velocity $v_i[T_l^i]$ and a turning center with a turning radius $r_i[T_l^i]$ along the line perpendicular to its current heading direction, and then circles around it until the next selected time point. The type 1 random variables $\mathbf{a}_i[T_l^i] = [r_i[T_l^i], v_i[T_l^i]]$ are inversely Gaussian and uniformly distributed respectively, and the type 2 random variable $\tau_i[T_l^i] = T_{l+1}^i - T_l^i$ is exponentially distributed. The switching behavior and sample trajectory are shown in Figs. 1(a) and 1(b) respectively.

The communication topology among the agents is fixed and represented using an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of agents $\mathcal{V} = 1, 2, \ldots, N$, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of communication links. A link $(i, j)(i \neq j)$ means that agents i and j can directly communicate with each other, where j is one of the neighbors of agent i.

Each agent has a local measurement model of a general nonlinear form

$$\mathbf{z}_{i|j}[k] = g(\mathbf{x}_i[k]) + \varpi_{z,i}[k], \tag{3}$$

where $\mathbf{z}_{i|j}[k]$ is the measured output of agent i by its neighbor j, $g(.)$ is a general nonlinear function, and $\varpi_{z,i}[k]$ is the white Gaussian noise.

B. Problem Formulation

We consider the following stochastic optimal control problem defined on a network of agents subject to the random switching dynamics. Denote the number of agent i's neighbors as n_i. Each agent i seeks its control policies $\mathbf{u}_{i,j}[k]$, $j \in [1, \ldots, n_i]$, to optimize a performance cost with its neighbor j according to the measurement $\mathbf{z}_{j|i}[k]$. Each agent i has at least n_i controllers to optimize the cost with the n_i neighbors respectively. This formulation has practical use in a wide range of new mobile networking applications, where the co-design of communication and control components becomes essential. An example is illustrated in Section IV.

In general, the expected cost to optimize is

$$J_{i,j} = E[\sum_{k=0}^{\infty} r_{i,j}(\mathbf{x}_i[k], \mathbf{x}_j[k], \mathbf{u}_{i,j}[k], \mathbf{u}_{j,i}[k])]. \tag{4}$$

where $r_{i,j}[k]$, $(j = 1, \ldots, n_i)$ is the cost between agent i and its neighbor j at time k. $\mathbf{u}_{i,j}[k]$ is the control vector of agent i, which seeks to minimize the communication cost with its neighbor j, $J_{i,j}$. The value function $V_{i,j}(\mathbf{x})$ corresponding to the performance index is defined as

$$V_{i,j}[k] = E[\sum_{k'=k}^{\infty} r_{i,j}(\mathbf{x}_i[k'], \mathbf{x}_j[k'], \mathbf{u}_{i,j}[k'], \mathbf{u}_{j,i}[k'])]. \tag{5}$$

Consider the problem of finding the optimal control policies $\mathbf{u}_{i,j}^*[k]$ that satisfies

$$\mathbf{u}_{i,j}^*[k] = \underset{\mathbf{u}_{i,j}}{\operatorname{argmin}} J_{i,j}[k](\mathbf{x}_i[k], \mathbf{x}_j[k], \mathbf{u}_{i,j}[k], \mathbf{u}_{j,i}[k]). \tag{6}$$

III. Main Results

A. Optimal Control in Random Switching Systems

In this subsection, we assume the state information, i.e., $\mathbf{x}_i[k]$ and $\mathbf{x}_j[k]$, is available, and design an adaptive optimal controller to find the optimal policies for agents moving with random switching dynamics.

Consider the value function described in (5). Because the uncertain parameters are independent from the system state, the following Bellman equation holds,

$$\begin{aligned} V_{i,j}[k] &= E(\sum_{k'=k}^{\infty} r_{i,j}[k']) = E(r_{i,j}[k] + \sum_{k'=k+1}^{\infty} r_{i,j}[k']) \\ &= E(r_{i,j}[k] + V_{i,j}[k+1]). \end{aligned} \tag{7}$$

This Bellman equation can be solved online using RL [9]. In particular, assume that a neural network weight $\mathbf{W}_{i,j}$ exists such that the value function can be approximated as

$$V_{i,j}(\mathbf{x}_i[k], \mathbf{x}_j[k]) = \mathbf{W}_{i,j}^T \phi(\mathbf{x}_i[k], \mathbf{x}_j[k]). \tag{8}$$

Using the value function approximation (VFA) method, the optimal control policy can be found from the policy iteration (PI) algorithm [10]. Two main steps are involved in the PI algorithm: 1) policy evaluation, which evaluates the value

function at each time step, and 2) policy improvement, which finds the optimal policy based on the current value function.

For random switching systems, the policy evaluation step involves uncertainty evaluation to calculate the expectation of a function as shown in (7). This uncertainty evaluation is typically obtained using the Monte Carlo method and its variants, too slow to be used for on-line decision algorithms. Here we use a multivariate probabilistic collocation method (MPCM) [5] to effectively evaluate the uncertainty. To map to the MPCM framework, we here denote the generic function whose expectation to be evaluated as $G(a_1, \ldots, a_m)$, which is modulated by uncertain parameters $a_1, a_2, \ldots, a_m$ with the degree of each parameter up to a certain number. The MPCM accurately evaluates the output mean of G, by smartly selecting a limited number of sample points according to the Gaussian Quadrature rules, evaluating these sample points, and producing the output mean from a reduced-order mapping G'. The main property of MPCM is described in the following lemma. Please refer to [5] for the proof and detailed MPCM design procedures.

Lemma 1 [5, Th. 2]: Consider a generic system mapping modulated by m independent uncertain parameters:

$$G(a_1, \ldots, a_m) = \sum_{q_1=0}^{2n_1-1} \sum_{q_2=0}^{2n_2-1} \cdots \sum_{q_m=0}^{2n_m-1} \psi_{q_1,\ldots,q_m} \prod_{p=1}^{m} a_p^{q_p},$$

where a_p ($p \in 1, 2, \ldots, m$) is an uncertain parameter with degree up to $2n_p - 1$. n_p is a positive integer for any $p \in 1, 2, \ldots, m$, and $\psi_{q_1,\ldots,q_m} \in \mathbb{R}$ are the coefficients. Each uncertain parameter a_p follows an independent pdf $f_{A_p}(a_p)$. The MPCM approximates $G(a_1, \ldots, a_m)$ with the following low-order mapping

$$G'(a_1, \ldots, a_m) = \sum_{q_1=0}^{n_1-1} \sum_{q_2=0}^{n_2-1} \cdots \sum_{q_m=0}^{n_m-1} \Omega_{q_1,\ldots,q_m} \prod_{p=1}^{m} a_p^{q_p},$$

with $E[G(a_1, \ldots, a_m)] = E[G'(a_1, \ldots, a_m)]$, where $\Omega_{q_1,\ldots,q_m} \in \mathbb{R}$ are coefficients.

Remark 1: Lemma 1 shows that the MPCM reduces the number of simulations from $2^m \prod_{p=1}^{m} n_p$ to $\prod_{p=1}^{m} n_p$, where m is the number of uncertain parameters. Despite the significant reduction of computation by 2^m, MPCM accurately predicts the output mean [5]. We note that the degree of an uncertain parameter in G is dependent on specific applications. For a nonlinear system, G is a polynomial approximation with properly selected maximal degree for each parameter, $2n_p - 1$. With the increase of maximal degree, the approximation accuracy can be improved, but at the cost of additional computational load and the chance of overfitting.

Here we integrate RL and MPCM to provide an effective online learning-based optimal control algorithm for random switching systems.

To evaluate the value function $V_{i,j}[k]$ at each time instant, one needs to calculate $E(V_{i,j}[k+1])$ according to the Bellman equation (7). The value function $V_{i,j}[k+1]$ is determined uniquely by the system states $\mathbf{x}_i[k+1]$ and $\mathbf{x}_j[k+1]$, which can be found from the current states $\mathbf{x}_i[k]$ and $\mathbf{x}_j[k]$, system dynamics $f(.)$, and the random switching behaviors. In particular, given the current states $\mathbf{x}_i[k]$ and $\mathbf{x}_j[k]$, agent i can predict its future state $\mathbf{x}_i[k+1]$ according to its current maneuver $\mathbf{a}_i[k+1]$ using the system dynamics $f(.)$. However, agent i does not know agent j's current maneuver $\mathbf{a}_j[k+1]$, and as such, $\mathbf{x}_j[k+1]$ needs to be estimated by agent i considering its switching behaviors. Denote the switching behavior of agent j at time k as $s_j[k]$. $s_j[k] = 1$ or 0 indicates if the current maneuver switches at time k or not. Denote the value function $V_{i,j}[k]$ when $s_j[k] = 1$ (or $s_j[k] = 0$) as $V_{i,j}^1[k]$ (or $V_{i,j}^0[k]$ correspondingly). When $s_j[k] = 0$, agent j keeps its previous maneuver $\mathbf{a}_j[k]$, and the system state $\mathbf{x}_j[k+1]$ is obtained using $\mathbf{a}_j[k]$, i.e.,

$$V_{i,j}^0[k] = r_{i,j}[k] + V_{i,j}[k+1](\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[k]). \quad (9)$$

When $s_j[k] = 1$, agent j chooses a new random maneuver from $\mathbf{a}_j[T_l^j]$ at time k, and in this case, $E(V_{i,j}[k+1])$ needs to be estimated from the characteristics of the random variable $\mathbf{a}_j[T_l^j]$. Define a system mapping subject to uncertain input parameters $\mathbf{a}_j[T_l^j]$: $G_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j]) = r_{i,j}[k] + V_{i,j}[k+1](\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])$, then the value function $V_{i,j}^1[k]$ can be estimated from the mean output of the system mapping $G_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])$ using MPCM, i.e., $V_{i,j}^1[k] = E[G_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])]$. In particular, a set of samples are selected based on the pdfs of uncertain parameters, and simulations are run at these samples to estimate $E[G_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])]$. Under the assumption that each uncertain parameter $a_{j,p}$ has a degree up to $2n_p - 1$, $G_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])$ has the following form,

$$G_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j]) = \sum_{q_1=0}^{2n_1-1} \sum_{q_2=0}^{2n_2-1} \cdots \sum_{q_m=0}^{2n_m-1} \psi_{q_1,\ldots,q_m}^V(\mathbf{x}_i[k+1], \mathbf{x}_j[k]) \prod_{p=1}^{m} a_{j,p}^{q_p}.$$

According to Lemma 1, the output mean of this system mapping can be estimated from the output of a reduced-order mapping $G'_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])$ derived from the MPCM procedure [5, Sec. II],

$$V_{i,j}^1[k] = E[G_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])] = E[G'_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])], \quad (10)$$

$$G'_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j]) = \sum_{q_1=0}^{n_1-1} \sum_{q_2=0}^{n_2-1} \cdots \sum_{q_m=0}^{n_m-1} \Omega_{q_1,\ldots,q_m}^V(\mathbf{x}_i[k+1], \mathbf{x}_j[k]) \prod_{p=1}^{m} a_{j,p}^{q_p}. \quad (11)$$

The coefficients $\Omega_{q_1,\ldots,q_m}^V(\mathbf{x}_i[k+1], \mathbf{x}_j[k])$ and output mean can be obtained using the evaluated outputs $G'_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])$ at each selected simulation point, according to the procedures in [5, Sec. II-B].

Theorem 1: The value function described in (7) can be estimated as

$$V_{i,j}[k] = PV_{i,j}^0[k] + (1-P)V_{i,j}^1[k], \quad (12)$$

where $V_{i,j}^0[k]$ and $V_{i,j}^1[k]$ are described in (9) and (10) respectively. P is the probability that agent j switches its maneuver at time k. This probability can be derived from the distribution of $\tau_j[T_l^j]$, $f_\tau(\tau_j[T_l^j])$.

Algorithm 1 Policy Iteration Algorithm for Random Switching Systems

1: Initialize the system with initial states $\mathbf{x}_i[0], \mathbf{x}_j[0]$, and admissible control policies $\mathbf{u}_{i,j}[0]$ and $\mathbf{u}_{j,i}[0]$.
2: Select $\prod_{p=1}^{m} n_p$ MPCM sample points according to the pdfs $f_{A_p}(a_{j,p}[T_l^j])$ and the MPCM procedure [5, Sec. II]. Denote each selected MPCM sample as $\mathcal{A}^l$, where $l = 1, ..., \prod_{p=1}^{m} n_p$.
3: For each iteration s, find the value function when $s_j[k] = 0$, $V_{i,j}^{0,(s)}$, using (9).
4: Find the value function $\mathcal{V}_{i,j}^{l,(s)}(\mathbf{x}_i[k], \mathbf{x}_j[k])$ at each MPCM sample $\mathcal{A}^l$, using the Bellman equation: $\mathcal{V}_{i,j}^{l,(s)}[k] = r_{i,j}[k] + \mathbf{W}_{i,j}^{(s-1)T}\phi(\mathbf{x}_i[k+1], \mathbf{x}_j[k+1])$.
5: Find the reduced polynomial mapping from $a_{j,p}$ to $G'_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])$ described in (11) according to Lemma 1. $a_{j,p}$ and $G'_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])$ take the value of $\mathcal{A}^l$ and $\mathcal{V}_j^l[k]$ respectively.
6: Find the value function when $s_j[k] = 1$, i.e., $V_{i,j}^{1,(s)}[k]$, by combining (10) and the derived system mapping $G'_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])$.
7: Find the value function $V_{i,j}^{(s)}[k]$ by combining Theorem 1, $V_{i,j}^{0,(s)}[k]$ and $V_{i,j}^{1,(s)}[k]$.
8: Update the value function coefficients $W_{i,j}^{(s)}$ according to the estimated $V_{i,j}^{(s)}[k]$: $\mathbf{W}_{i,j}^{(s)T}\phi(\mathbf{x}_i[k], \mathbf{x}_j[k]) = V_{i,j}^{(s)}[k]$.
9: Update the control policy $\mathbf{u}_{i,j}^{(s)}$ as $\mathbf{u}_{i,j}^{(s)} = \text{argmin}_{\mathbf{u}_{i,j}} V_{i,j}^{(s)}[k]$.
10: Repeat procedures $3-9$.

Proof: The value function for an agent of random switching dynamics can be derived as

$$\begin{aligned} V_{i,j}[k] &= E(V_{i,j}[k]|s_j[k]=0)P(s_j[k]=0) \\ &\quad + E(V_{i,j}[k]|s_j[k]=1)P(s_j[k]=1) \\ &= PV_{i,j}^0[k] + (1-P)V_{i,j}^1[k]. \end{aligned} \tag{13}$$

Equations (9), (10) and (13) naturally lead to Theorem 1. ■

The detailed algorithm that integrates the PI learning algorithm and MPCM is described in Algorithm 1. After initialization in Step 1, Step 2 samples $\mathbf{a}_j[T_l^j]$ to prepare for the uncertainty evaluation procedures in Steps $4-6$. Steps $3-7$ are value function evaluation. In particular, Step 3 evaluates $V_{i,j}^0$, Steps $4-6$ evaluate $V_{i,j}^1$, and Step 7 combines $V_{i,j}^0$ and $V_{i,j}^1$ according to Theorem 1 to find $V_{i,j}[k]$. After value function evaluation, the approximation weights $\mathbf{W}_{i,j}$ and optimal control polices $\mathbf{u}_{i,j}$ are updated respectively in Steps 8 and 9. The detailed procedures for MPCM and PI algorithm can be found in [5], [10] respectively.

Theorem 2: Consider the random switching system shown in (1) with the value function described in (5). Assume there exists a unique optimal solution and Algorithm 1 converges. Given the current system states $\mathbf{x}_i[k]$ and $\mathbf{x}_j[k]$, the solution found by Algorithm 1 is the optimal control policy.

Proof: The control policy derived by evaluating the value function $V_{i,j}[k] = PV_{i,j}^0[k] + (1-P)V_{i,j}^1[k]$ is optimal according to (6), Theorem 1, and the policy iteration properties [10]. As such, to prove this theorem, we are left to show that the two optimal solutions, which are found by evaluating the reduced-order mapping $PV_{i,j}^0[k] + (1-P)G'_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])$ and the original value function mapping $PV_{i,j}^0[k] + (1-P)G_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])$ are the same. Lemma 1 proves that $E[G'_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])] = E[G_{V_{i,j}}(\mathbf{x}_i[k+1], \mathbf{x}_j[k], \mathbf{a}_j[T_l^j])]$. Therefore, the equivalence of the two optimal solutions can be proved from a contradiction argument described in [6, Th. 1]. ■

Remark 2: The convergence of Algorithm 1 depends on three numerical solutions: the policy iteration method, the value function approximation, and the MPCM approximation. The policy iteration method has been widely used in dynamic programming and reinforcement learning fields [6], [10], with its convergence conditions provided in [10]. The value function approximation uses neural networks to approximate the smooth value function. The assumptions that make this approximation hold are listed in [11]. MPCM works well for both polynomial and non-polynomial system mappings as guaranteed by Lemma 1, with properly selected degrees for the polynomials (see [5] for the details).

B. State Estimation in Random Switching Systems

In many practical applications, state information $\mathbf{x}_i[k]$ and $\mathbf{x}_j[k]$ may not be available for controller design. In this subsection, we provide a practical online state estimation solution for agents of random switching systems.

Given the previous state $\mathbf{x}_i[k-1]$, the expected current state $E(\mathbf{x}_i[k]|\mathbf{x}_i[k-1])$ can be estimated considering the two possible switching behaviors: $s_i[k-1] = 1$ or 0. When $s_i[k-1] = 1$, agent i chooses a new random maneuver from $\mathbf{a}_i[T_l^i]$ at time $k-1$. As such, the estimation of the expected conditional system state $E(\mathbf{x}_i[k]|\mathbf{x}_i[k-1], s_i[k-1] = 1)$ involves uncertainty evaluation, which we solve using MPCM, instead of the Monte Carlo methods which are computationally ineffective. In particular, we define $f(\mathbf{x}_i[k-1], \mathbf{a}_i[T_l^i])$ as a system mapping subject to uncertain input parameters $\mathbf{a}_i[T_l^i]$, i.e., $G_i(\mathbf{x}_i[k-1], \mathbf{a}_i[T_l^i])$. The expected system state when $s_i[k-1] = 1$ is then approximated from the mean output of the system mapping $G_i(\mathbf{x}_i[k-1], \mathbf{a}_i[T_l^i])$ using MPCM, i.e.,

$$E(\mathbf{x}_i[k]|\mathbf{x}_i[k-1], s_i[k-1]=1) = E[G_i(\mathbf{x}_i[k-1], \mathbf{a}_i[T_l^i])]. \tag{14}$$

Theorem 3: Given the previous system state $\mathbf{x}_i[k-1]$, the expected current state for agent i is estimated as

$$\begin{aligned} E(\mathbf{x}_i[k]|\mathbf{x}_i[k-1]) &= PE[G'_i(\mathbf{x}_i[k-1], \mathbf{a}_i[T_l^i])] \\ &\quad + (1-P)f(\mathbf{x}_i[k-1], \mathbf{a}_i[k-1]), \end{aligned} \tag{15}$$

where P is the probability that agent i switches its maneuver at time $k-1$. $G'_i(\mathbf{x}_i[k-1], \mathbf{a}_i[T_l^i])$ is a reduced order mapping of $G_i(\mathbf{x}_i[k-1], \mathbf{a}_i[T_l^i])$ derived from MPCM.

Proof: The expected system state at time k for an agent of random switching dynamics can be derived as

$$\begin{aligned} &E(\mathbf{x}_i[k]|\mathbf{x}_i[k-1]) \\ &\quad = E(\mathbf{x}_i[k]|\mathbf{x}_i[k-1], s_i[k-1]=0)P(s_i[k-1]=0) \\ &\quad + E(\mathbf{x}_i[k]|\mathbf{x}_i[k-1], s_i[k-1]=1)P(s_i[k-1]=1). \end{aligned} \tag{16}$$

In the case of $s_i[k-1]=0$, agent i keeps its previous maneuver. The conditional expected system state $E(\mathbf{x}_i[k]|\mathbf{x}_i[k-1], s_i[k-1]=0)$ can be found from the previous system state $\mathbf{x}_i[k-1]$ and maneuver $\mathbf{a}_i[k-1]$ as

$$E(\mathbf{x}_i[k]|\mathbf{x}_i[k-1], s[k-1]=0) = f(\mathbf{x}_i[k-1], \mathbf{a}_i[k-1]). \quad (17)$$

In the case of $s_i[k-1]=1$, agent i chooses a new maneuver from $\mathbf{a}_i[T_l^i]$ at time $k-1$. The expected conditional system state $E(\mathbf{x}_i[k]|\mathbf{x}_i[k-1], s_i[k-1]=1)$ can be estimated from the mean output of the reduced-order system mapping $G_i'(\mathbf{x}_i[k-1], \mathbf{a}_i[T_l^i])$ using MPCM according to (14) and Lemma 1.

Equations (14), (16) and (17) naturally lead to Theorem 3. ■

Theorem 3 provides an accurate and computationally-efficient approach to estimate the expected system state for random switching systems, given the previous state. Next we integrate it with UKF to provide the state estimation solution from the measurement signals $\mathbf{z}_{i|j}[k]$. The system is assumed to be observable. In particular, MPCM and UKF are integrated for a 5-step state estimation procedure. Steps 1 and 2 select initial conditions and MPCM points to initialize Steps 3–5. Steps 3 and 4 find the state estimator for the switching behaviors $s_i[k-1]=0$ and 1 respectively. Step 5 finds the expected state by integrating the two estimators in Steps 3 and 4.

Step 1 (Initialize): Initialize $\hat{\mathbf{x}}_i[0]$ and $\mathbf{P}_i[0]$.

Step 2 (Select MPCM Points): $\prod_{p=1}^m n_p$ MPCM sample points are selected according to the pdfs $f_{A_p}(a_{i,p}[T_l^i])$ and the MPCM procedure [5, Sec. II]. Denote each selected MPCM sample as $\mathcal{A}^l$, where $l=1,\ldots,\prod_{p=1}^m n_p$.

Step 3 (Estimate the System State When $s_i[k-1]=0$): The expected system state $E(\mathbf{x}_i[k]|\hat{\mathbf{x}}_i[k-1], \mathbf{z}_{i|j}[k], s_i[k-1]=0)$ can be estimated using UKF through the following four sub-steps [7]: (a) select sigma points from $\hat{\mathbf{x}}_i[k-1]$; (b) predict system state by instantiating the sigma points through the system dynamics $f(.)$; (c) select new sigma points from the predicted state, and predict measurement by instantiating the sigma points through the measurement model $g(.)$; (d) update the Kalman gain and find the expected state $E(\mathbf{x}_i[k]|\hat{\mathbf{x}}_i[k-1], \mathbf{z}_{i|j}[k], s_i[k-1]=0)$ and covariance $E(\mathbf{P}_i[k]|\mathbf{P}_i[k-1], \mathbf{z}_{i|j}[k], s_i[k-1]=0)$. Please refer to [7], [8] for the detailed UKF procedure.

Step 4 (Estimate the System State When $s_i[k-1]=1$): Uncertainty evaluation is necessary in this step, and the expected system state is derived by integrating MPCM and UKF using the following three sub-steps (a)-(c).

(a) Estimate system state at each selected MPCM point: At each selected MPCM point $\mathcal{A}^l$ ($l=1,\ldots,\prod_{p=1}^m n_p$), the system state can be estimated from the UKF procedure shown in ***Step 3***, *(a)-(d)*. Denote the estimated state from UKF at each sample point as $\hat{\mathbf{x}}_i^l[k]$ with the covariance $\mathbf{P}_i^l[k]$ ($l=1,\ldots,\prod_{p=1}^m n_p$).

(b) Find the reduced polynomial mappings: Define system mappings $G'_{\mathbf{x}_i}(\hat{\mathbf{x}}_i[k-1], \mathbf{a}_i[T_l^i])$ and $G'_{\mathbf{P}_i}(\hat{\mathbf{x}}_i[k-1], \mathbf{a}_i[T_l^i])$ as the relationships between the expected system state and covariance with the random variable $\mathbf{a}_i[T_l^i]$. According to Lemma 1, the reduced-order mappings can be found respectively as $G'_{\mathbf{x}_i}(\hat{\mathbf{x}}_i[k-1], \mathbf{a}_i[T_l^i]) = \sum_{q_1=0}^{n_1-1}\sum_{q_2=0}^{n_2-1}\cdots\sum_{q_m=0}^{n_m-1}\Omega^{\mathbf{x}}_{q_1,\ldots,q_m}(\hat{\mathbf{x}}_i[k-1])\prod_{p=1}^m a_{i,p}^{q_p}$, and $G'_{\mathbf{P}_i}(\hat{\mathbf{x}}_i[k-1], \mathbf{a}_i[T_l^i]) = \sum_{q_1=0}^{n_1-1}\sum_{q_2=0}^{n_2-1}\cdots\sum_{q_m=0}^{n_m-1}\Omega^{\mathbf{P}}_{q_1,\ldots,q_m}(\hat{\mathbf{x}}_i[k-1])\prod_{p=1}^m a_{i,p}^{q_p}$ respectively.

(c) Find the expected system state and covariance: The expected system state and covariance are then found from the mean output of the system mapping according to Lemma 1 and the MPCM design procedure [5], $E(\mathbf{x}_i[k]|\hat{\mathbf{x}}_i[k-1], \mathbf{z}_{i|j}[k], s_i[k-1]=1) = E[G'_{\mathbf{x}_i}(\hat{\mathbf{x}}_i[k-1], \mathbf{a}_i[T_l^i])]$, $E(\mathbf{P}_i[k]|\mathbf{P}_i[k-1], \mathbf{z}_{i|j}[k], s_i[k-1]=1) = E[G'_{\mathbf{P}_i}(\hat{\mathbf{x}}_i[k-1], \mathbf{a}_i[T_l^i])]$.

Step 5 (Estimate the Expected System State): The estimated state and covariance are then derived from *Steps 3* and *4* according to Theorem 3 as

$$\begin{aligned}
&E(\mathbf{x}_i[k]|\hat{\mathbf{x}}_i[k-1], \mathbf{z}_{i|j}[k])\\
&\quad = PE(\mathbf{x}_i[k]|\hat{\mathbf{x}}_i[k-1], \mathbf{z}_{i|j}[k], s_i[k-1]=1])\\
&\qquad + (1-P)E(\mathbf{x}_i[k]|\hat{\mathbf{x}}_i[k-1], \mathbf{z}_{i|j}[k], s_i[k-1]=0]),\\
&E(\mathbf{P}_i[k]|\mathbf{P}_i[k-1], \mathbf{z}_{i|j}[k])\\
&\quad = PE(\mathbf{P}_i[k]|\mathbf{P}_i[k-1], \mathbf{z}_{i|j}[k], s_i[k-1]=1])\\
&\qquad + (1-P)E(\mathbf{P}_i[k]|\mathbf{P}_i[k-1], \mathbf{z}_{i|j}[k], s_i[k-1]=0]).
\end{aligned}$$

As such, the estimator of $\mathbf{x}_i[k]$ is $\hat{\mathbf{x}}_i[k] = E(\mathbf{x}_i[k]|\hat{\mathbf{x}}_i[k-1], \mathbf{z}_{i|j}[k])$, and the expected error covariance is $\mathbf{P}_i[k] = E(\mathbf{P}_i[k]|\mathbf{P}_i[k-1], \mathbf{z}_{i|j}[k])$.

Remark 3: The performance of the state estimation algorithm is jointly determined by UKF and MPCM. UKF addresses the nonlinear system dynamics and measurement models. MPCM effectively samples the random switching behavior. The accuracy of MPCM is guaranteed by Lemma 1. Note that UKF is not an optimal estimator. It has been practically used to provide approximations to optimal solutions with certain accuracy. Performance analysis on UKF for general systems is limited in the literature. When the measurement model is linear, the estimation error of UKF is bounded when an extra positive definite matrix is added in the calculated covariance matrix [8]. Here we use the UKF method to address random switching system dynamics. The use of MPCM does not deteriorate the convergence or the optimality of state estimation as guaranteed by Lemma 1.

IV. Illustrative Examples

Consider a five-UAV network to support a surveillance-like mission. UAVs move independently according to the ST RMM described in Section II-A. The randomly-generated trajectories of the UAVs are shown in Fig. 2(a).

A three-sector directional antenna is mounted on each UAV to communicate with its neighbor UAVs over long distances upon an ID-based fixed communication topology (Fig. 2(b)). To maximize the communication performance, each UAV controls the heading directions of its antennas to maximize the received signal strength indicators (RSSI), which measure the communication channel performance. The cost function in this example is $J_{i,j} = -E[\sum_{k=0}^N R_{i,j}[k]]$, where $R_{i,j}$, the RSSI that UAV i receives from its neighbor j is $R_{i,j}[k] = P_{t|dBm} + 20\log_{10}(\frac{\lambda}{4\pi d[k]}) + G_{l|dBi}[k]$ (see [12] for the details), and N is the experimental time. Here $P_{t|dBm}$ is

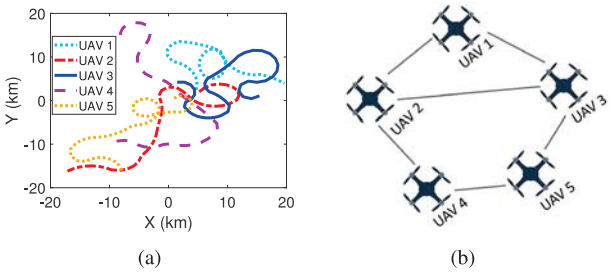

Fig. 2. (a) Sample trajectories of the UAVs, (b) Communication topology of the five-UAV network.

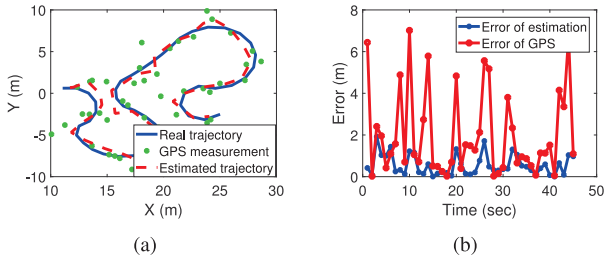

Fig. 3. Estimation performance. (a) Trajectory of UAV 3. (b) Estimation errors.

the transmitted signal power, λ is the wavelength, $d[k]$ is the distance between neighboring UAVs, and $G_{l|dBi}[k]$ is the sum of gains of the two antennas, which depends on their heading angles. Measurements are GPS corrupted with Gaussian white noise. The performances of the designed estimators and controllers are simulated for all five UAVs and communication links. We here only show the performance of UAV 3 and its communication link to UAV 2.

We first investigate the computation load reduction of the MPCM through a comparative study. Because two type 1 random variables are involved, the number of uncertain parameters in the system mapping $G(a_1, a_2)$ is $m = 2$. We select $n_1 = 2$ for the degree of $a_1 = v_i[T_l^i]$ and $n_2 = 3$ for $a_2 = r_i[T_l^i]$. With this parameter setting, $\prod_{p=1}^{m} n_p = 6$ MPCM points are selected according to the MPCM procedure [5]. For the optimal control solution developed in III-A, the Monte Carlo method requires about 8000 sample points to converge to the output mean, while the MPCM method only needs 6 points to converge to the correct result. The significant reduction of computation load shows the value of MPCM to facilitate online uncertainty evaluation.

We then analyze performance of the state estimator designed in Section III-B (see Fig. 4) . The estimated trajectory matches well with the real UAV trajectory, and the estimation errors are much smaller than the errors of GPS signals, which validates the effectiveness of the estimation solution.

Finally, we simulate the optimal controller designed based on the estimated states. Fig. 4(a) shows the controlled heading directions of the directional antenna mounted on UAV 3 to communicate with UAV 2, and Fig. 4(b) shows the errors between the controlled and real optimal heading directions. The controlled directional antenna heading direction is very close to the optimal solution, and the errors are within

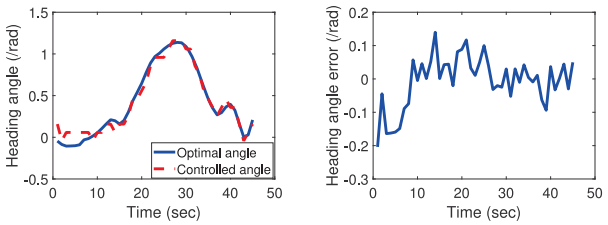

Fig. 4. Control performance. (a) Optimal headings of directional antenna on UAV 3 to communicate with UAV 2. (b) Errors between the optimal and the controlled heading angles of the directional antenna.

$(-0.2, 0.15)$ rad, which validates effectiveness of the proposed adaptive optimal control solution.

V. Conclusion

This letter studies the design of adaptive optimal decision solutions for multi-agent random switching systems. An optimal controller and a practical state estimator, developed based on RMM, UKF, MPCM and RL constructs, provide fast online decision solutions for multiple agents moving with general highly flexible and uncertain movement patterns. Efficiency and accuracy of the proposed solutions are analyzed. In the future work, we will further investigate properties of the proposed solutions for random switching systems, including convergence, robustness, and optimality.

References

[1] J. Xie, Y. Wan, J. H. Kim, S. Fu, and K. Namuduri, "A survey and analysis of mobility models for airborne networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1221–1238, 3rd Quart., 2014.

[2] Y. Wan, K. Namuduri, Y. Zhou, and S. Fu, "A smooth-turn mobility model for airborne networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 3359–3370, Sep. 2013.

[3] M. Liu and Y. Wan, "Analysis of random mobility model with sense and avoid protocols for UAV traffic management," in *Proc. Inf. Syst. AIAA Infotech Aerosp.*, Kissimmee, FL, USA, 2018, pp. 1–14.

[4] H. J. Kappen, "An introduction to stochastic control theory, path integrals and reinforcement learning," *Cooperat. Behav. Neural Syst.*, vol. 887, no. 1, pp. 149–181, Feb. 2007.

[5] Y. Zhou *et al.*, "Multivariate probabilistic collocation method for effective uncertainty evaluation with application to air traffic flow management," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 10, pp. 1347–1363, Oct. 2014.

[6] J. Xie, Y. Wan, K. Mills, J. J. Filliben, and F. L. Lewis, "A scalable sampling method to high-dimensional uncertainties for optimal and reinforcement learning-based controls," *IEEE Control Syst. Lett.*, vol. 1, no. 1, pp. 98–103, Jul. 2017.

[7] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proc. IEEE Amer. Control Conf.*, Seattle, WA, USA, 1995, pp. 1628–1632.

[8] K. Xiong, H. Zhang, and C. Chan, "Performance evaluation of UKF-based nonlinear filtering," *Automatica*, vol. 42, no. 2, pp. 261–270, Feb. 2006.

[9] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, 3rd Quart., 2009.

[10] D. P. Bertsekas, "Value and policy iterations in optimal control and adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 500–509, Mar. 2017.

[11] F. Tatari, K. Vamvoudakis, and M. Mazouchi, "Optimal distributed learning for disturbance rejection in networked nonlinear games under unknown dynamics," *IET Control Theory Appl.*, to be published.

[12] S. Li *et al.*, "The design and implementation of aerial communication using directional antennas: Learning control in unknown communication environments," *IET Control Theory Appl.*, to be published.