Unmanned Systems, Vol. 0, No. 0 (2013) 1–16

Estimation of Random Mobility Models with Application to Unmanned Aerial Vehicles

Tao Li[a], Yan Wan[b], Mushuang Liu[c], Songwei Li[b], Murilo Augusto Pinheiro[b], Chenyuan He[d], and Frank L. Lewis[b]

[a] *Texas A&M Transportation Institute.*
Email: t-li@tti.tamu.edu.

[b] *Department of Electrical Engineering, University of Texas at Arlington.*
Email: yan.wan@uta.edu (contact author), songwei.li@mavs.uta.edu, muriloaugusto.pinheiro@mavs.uta.edu and lewis@uta.edu

[c] *Department of Mechanical and Aerospace Engineering, University of Missouri.*
Email: ml529@missouri.edu.

[d] *School of Automotive and Traffic Engineering, Jiangsu University.*
Email: chenyuan_he@ujs.edu.cn.

Random mobility models (RMMs) capture the statistical movement characteristics of mobile agents and play an important role in the evaluation and design of mobile wireless networks. Particularly, RMMs are used to model the movement of Unmanned Aerial Vehicles (UAV) as the platforms for airborne communication networks. In many RMMs, the movement characteristics are captured as stochastic processes constructed using two types of independent random variables. The first type describes the movement characteristics for each maneuver, and the second type describes how often the maneuvers are switched. We develop a generic method to estimate RMMs that are composed of these two types of random variables. Specifically, we formulate the dynamics of movement characteristics generated by the two types of random variables as a special Jump Markov System and develop an estimation method based on the Expectation-Maximization principle. Both off-line and on-line variants of the method are developed. We apply the estimation method to the Smooth-Turn RMM developed for fixed-wing UAVs. The simulation study validates the performance of the proposed estimation method. We further conduct a UAV experimental study and apply the estimation methods to real UAV trajectories.

Keywords: Random Mobility Model; Jump Markov System; Expectation-Maximization.

1. Introduction

Random mobility models (RMMs) capture the movement characteristics of mobile agents, and have been widely used to evaluate the performance of mobile wireless networks [1]. Many RMMs have been developed, ranging from the basic ones (e.g., Random Direction and Random Walk) to more sophisticated ones designed for specific vehicle types (e.g., ground vehicles and airborne vehicles [2]) and vehicle movement patterns with specific constraints (e.g., [3]). Additional examples of RMMs can be found in several survey papers [1,4]. With the increasing use of UAVs as platforms for airborne wireless networks [5–7], RMMs are playing a more important role in modeling UAV's movements [8,9]. More broadly, autonomous systems are equipped with more intelligence, and RMMs naturally provide a mathematical framework to capture the uncertain intentions of autonomous systems [10].

Despite the wide variability, many RMMs capture the movement characteristics as stochastic processes constructed using two types of independent random variables. Type 1 describes the movement characteristics for each maneuver. Type 2 describes how often the maneuvers are switched. For example, paper [2] developed the smooth-turn mobility (ST) RMM to capture the smooth movement of fixed-wing unmanned aerial vehicles (UAVs). The ST RMM is composed of a sequence of switching turning maneuvers, in which the turning radius in each maneuver is captured as a type 1 random variable, and the duration of each maneuver is captured as a type 2 random variable. Similarly, the semi-random circular movement RMM in [11] describes an airborne vehicle's hovering movement over a specific location. In this model, an airborne vehicle randomly chooses a new radius (type 1 random variable) and a randomly chosen speed to circle the same location after it completes a circling movement, and therefore the time to complete one circling maneuver is also random (described by a type 2 random variable).

For the RMMs to generate realistic movement characteristics, parameters in the two types of random variables need to be properly specified using real trajectory data. Most of the existing estimation methods (see, e.g., [12,13]) were developed only for specific RMMs and lack the flexibility to be applied to other RMMs. In most of these studies, the movement characteristics are captured by only one random variable and hence the parameters for the random variable can be simply estimated through a direct fitting of the associated distribution with the observed trajectory data statistics. However, for the RMMs that we consider, estimating the parameters in one type by simply fitting the distribution with the trajectory data would result in inaccurate estimates because the trajectory is determined by the two types of random variables jointly.

To the best of our knowledge, only a few studies have been devoted to estimating parameters in the two types of random variables using trajectory data. Papers [14,15] developed a heuristic method to estimate both types of random variables in a 2-dimensional (2D) ST RMM. Specifically, the turning radius and switching points are first estimated from trajectory data by heuristically balancing among multiple criteria (e.g., degree of correlation and estimation error statistics). Then, the parameters in each type of random variable are estimated using the corresponding statistics. Paper [16] adopted a similar approach to estimate the random variable that determines the travel pause time of cell phone users. One drawback of this approach is that parameters are estimated based on one possible (heuristic) separation of the trajectory into maneuver sessions out of a large number of possibilities. Large errors can arise when the noise level in the trajectory data is high.

This study is focused on estimating parameters in the two types of random variables using trajectory data. A short conference version of this paper can be found at [17]. In contrast to [14, 16] which estimates the parameters in two types of random variables separately for specific movement characteristics of specific RMMs, the originality of our work is two-fold. First, we estimate parameters simultaneously by considering all possible separations of trajectory data into maneuver sessions. In other words, our method avoids the drawback of the methods in [14,16], and provides more reliable and accurate estimates. Second, our estimation method is general in that it is not restricted to specific movement characteristics of a specific RMM. In addition, this study presents both batch (offline) and recursive (online) applications of the estimation method.

We show that the parameter estimation problem falls into the category of estimating a Jump Markov Linear System (JMLS), which is NP-hard. We adopt the Expectation-Maximization (EM) algorithm to estimate the parameters. The algorithm is an iterative procedure, which generates maximum likelihood parameter estimates. It has been widely used in estimating JMLS (see, for example, [18–21]). However, the existing work on the estimation of the JMLS (e.g., [22–25]) focuses on the direct estimation of states, model coefficients, or transition matrix of the modulating Markov chain. In our study, the model coefficients and transition matrix are functions of the parameters that describe the two types of random variables. We apply the EM algorithm to estimate these parameters.

The rest of the paper is organized as follows. In Section 2, we formulate the parameter estimation problem as a special JMLS estimation problem. In Section 3, we present the estimation methodology. In Section 4.1, we apply the estimation methods to the 2-D ST RMM for validation. In Section 4.2, we collect data from a UAV experimental study and apply the methods to real UAV trajectories. Section 5 concludes the paper.

2. Problem Formulation

In this section, we formulate the parameter estimation problem as a special JMLS estimation problem.

As shown in Figure 1, the 2-D ST RMM is characterized by a sequence of turning maneuvers. Each turning maneuver has a turning radius of M (fixed for a maneuver) and turning duration of T. The reciprocal of the turning radius (i.e., $\frac{1}{M}$) is a normal random variable with a mean μ and standard variance σ, and duration T is an exponential random variable with mean λ. The turning center of each maneuver is on the line perpendicular to its current heading to ensure the smoothness of trajectories.

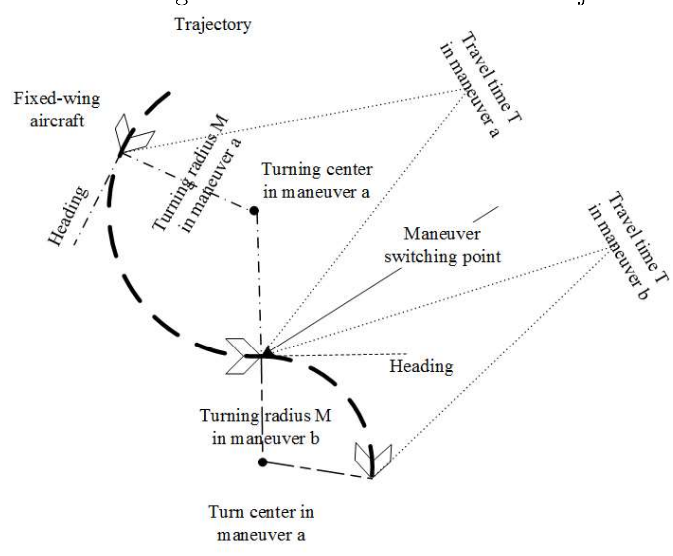


Fig. 1. Movement characteristics of a 2-D ST RMM.

The key dynamics of the ST RMM can be simply captured by the following random process M_t:

$$M_t = \begin{cases} \hat{M}_t, & \text{if } M_t \text{ switches at } t, \\ M_{t-1}, & \text{otherwise,} \end{cases} \tag{1}$$

where $\hat{M}_t$ is a random variable with the probabilistic density function (pdf) $f(\frac{1}{M_t}=\frac{1}{\hat{M}_t})=\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(\frac{1}{\hat{M}_t}-\mu)^2}{2\sigma^2}}$, and the probability of switching at each time step t is given by 1- exp(- $\lambda^{-1}\delta t$), where δt is the sampling time interval.

Next, we formulate the above dynamics as a Jump Markov System (JMS) for which we will provide the estimation method:

$$x_t{=}A(z_t,r_t)x_{t\text{-}1} + F(z_t,r_t)u_t + H(z_t,r_t),\\ y_t{=}C(z_t,r_t)x_t + D(z_t,r_t)w_t + G(z_t,r_t)u_t, \tag{2}$$

where x_t is the system state, y_t is the measurement of x_t, and

$$z_t{=}\begin{cases} z_{t\text{-}1}, \text{ if } r_t{=}s_0,\\ \hat{Z}_t, \ \ \text{if } r_t{=}s_1. \end{cases} \tag{3}$$

where $\hat{Z}_t$ is a random variable with the pdf $f(\hat{Z}_t = \hat{z}_t)$ denoted as $f(\hat{z}_t;\theta_1)$ with parameter $\theta_1.\theta_1$ determines the type 1 random variable for the movement characteristics of each maneuver. $r_t \in \{s_0, s_1\}$ is a discrete Markov chain of two states with transition matrix P given as follows:

$$P = \begin{bmatrix} p_{s_0,s_0}(t),\, p_{s_0,s_1}(t) \\ p_{s_1,s_0}(t),\, p_{s_1,s_1}(t) \end{bmatrix}, \tag{4}$$

where

$$p_{s_1,s_1}(t){=}P(r_t{=}s_1|r_{t\text{-}1}{=}s_1){=}g_1(\theta_2), \tag{5}$$
$$p_{s_0,s_1}(t){=}P(r_t{=}s_1|r_{t\text{-}1}{=}s_0){=}g_0(\theta_2), \tag{6}$$
$$p_{s_1,s_0}(t){=}P(r_t{=}s_0|r_{t\text{-}1}{=}s_1){=}1 - p_{s_1,s_1}(t), \tag{7}$$
$$p_{s_0,s_0}(t){=}P(r_t{=}s_0|r_{t\text{-}1}{=}s_0){=}1 - p_{s_0,s_1}(t), \tag{8}$$

θ_2 is a parameter that determines the type 2 random variable for the maneuver-switching behavior. g_0 and g_1 are functions of θ_2. θ_1 and θ_2 both can be vectors. w_t is zero-mean white Gaussian noise with known variance Σ_w and we further assume that $D(z_t,r_t)\Sigma_w D(z_t,r_t)' > 0$, where $'$ represents the matrix transpose operator. u_t is a known deterministic input. The matrices A, H, C, D, F, G are called model coefficients in this study and are all known functions of z_t and r_t. Furthermore, we assume that $A(\cdot, s_1)$ is a zero matrix to guarantee that the movement characteristics before and after a switch are independent.

The dynamics of the ST RMM in (1) can be described by the general modeling framework in (2) with the following matrix settings:

$$F(z_t,\cdot){=}H(z_t,s_0){=}G(z_t,\cdot){=}u_t{=}0,\\ C(z_t,r_t){=}D(z_t,r_t){=}A(z_t,s_0){=}1, H(z_t,s_1){=}z_t,\\ g_1(\lambda){=}g_0(\lambda){=}1\text{- exp(-}\lambda^{-1}\delta t),\\ f(z){=}N(\mu,\sigma), \theta_1{=}(\mu,\sigma), \theta_2{=}\lambda. \tag{9}$$

That is, when there is no switching in the turning maneuvers at t (i.e., $r_t = s_0$), (2) becomes

$$x_t = x_{t-1},\\ y_t = x_t + w_t \tag{10}$$

When there is a switching in the turning maneuvers at t (i.e., $r_t = s_1$), (2) becomes

$$x_t = z_t,\\ y_t = x_t + w_t, \tag{11}$$

where x_{t-1} and x_t are the reciprocal of the turning radius at t-1 and t, respectively. s_0 represents maintaining the current turning maneuver, s_1 represents switches to a new turning maneuver. z_t is the reciprocal of a new turning radius.

The focus of this study is to estimate the parameters $\theta{=}[\theta_1, \theta_2]'$ for both type 1 random variables for the movement characteristics of each maneuver, and the type 2 random variables that describe the maneuver switching behavior.

Estimating parameters of the system described by (2)-(6) is related to estimating a JMLS. In the literature, a significant amount of work has been focused on estimating the latter. The most used JMLS is of the following form:

$$x_t{=}A(\overline{r}_t)x_{t\text{-}1} + B(\overline{r}_t)v_t + F(\overline{r}_t)u_t,\\ y_t{=}C(\overline{r}_t)x_t + D(\overline{r}_t)w_t + G(\overline{r}_t)u_t, \tag{12}$$

where v_t is a zero-mean white Gaussian noise, $\overline{r}_t \in \{1, 2, \ldots U\}$ is the state of a discrete Markov chain with the transition matrix $P_{\overline{r}}$. One important difference between the two systems is as follows. The model coefficients and transition matrix (both of which are usually called model parameters in the literature) in (12) are functions of $\overline{r}_t$ and their values are typically in finite sets. In contrast, the model coefficients and transition matrix in (2) are functions of both r_t and z_t, which correspond to the two types of random variables. Another difference is that we introduce a new term $H(z_t, r_t)$ in (2) to create a switch of the movement characteristic that is independent of previous ones (i.e., $x_{1:t\text{-}1}$).

Estimating a JMLS can be NP-hard as the number of possible realizations of $\overline{r}_t$ grows exponentially with the size of states. While most of the existing estimation studies focus on estimating the states (i.e., $x_t, \overline{r}_t$), model coefficients (i.e., matrices $A(\overline{r}_t), B(\overline{r}_t), C(\overline{r}_t), D(\overline{r}_t), F(\overline{r}_t), G(\overline{r}_t)$) and transition matrix (i.e., $P_{\overline{r}_t}$), this study focuses on estimating the parameter θ, which determines the states, model coefficients, and transition matrix in (2).

3. Estimation methodology

In this section, we present the methodology to generate a maximum likelihood estimate of the parameter θ.

Let $\overline{Y}{=}[Y^1, Y^2, \ldots, Y^N]$ be the set of N mutually independent measurement experiments. For each $s \in \{1, ..., N\}$, $Y^s{=}[y_0^s, y_1^s \ldots, y_{L_s}^s, y_{L_s}^s]'$ is composed of $L_s + 1$ measurements of y_t, where $t{=}0, ..., L_s$. $\overline{R}{=}[R^1, R^2, \ldots, R^N]$ is the set of actual Markov chain states corresponding to $\overline{Y}$, where $R^s{=}[r_0^s, r_1^s \ldots, r_{L_s}^s]'$. In this study, we assume that $r_0^s{=}s_1,\ s{=}1, ..., N$.

The maximum likelihood estimate of θ is given by

$$\arg_\theta \max P(\overline{Y}|\theta). \tag{13}$$

Directly calculating $P(\overline{Y}|\theta)$ is challenging as the computations of high-dimensional integrals are not always tractable analytically. In the following, we apply the EM to estimate θ. The main idea of the EM is to treat $\overline{Y}$ as incomplete data and introduce a latent variable R (i.e., the Markov chain states) for which the joint likelihood $P(\overline{Y}, R|\theta)$ is available and easier to evaluate. The EM solves for θ that maximizes the expected log-likelihood of the complete data. Different from the method in [14, 16], the EM method estimates the parameter θ by considering all the possible realizations of R for $\overline{Y}$ (via the expected value). In addition, the prior distribution of parameter θ can also be considered by the EM method.

EM formulation: By introducing the latent variable R representing the Markov chain states, we solve the following estimation problem for θ.

$$\arg_\theta \max \sum_R P(\overline{Y}, R|\theta). \tag{14}$$

Overview of the EM algorithm:

The EM algorithm involves an iterative procedure that alternates between computing the expected log-likelihood and maximizing this computed expected log-likelihood. More specifically, the algorithm estimates θ using two iterative steps: E-step and M-step. It first calculates the expected value of the log-likelihood of the latent variables for a given parameter estimate (E-step) and then updates the parameter estimate by maximizing the expected value from E-step (M-step). The process is repeated until the convergence is reached. The EM algorithm is summarized as follows:

E-Step: By taking the log of the summation of the likelihood function in (14) and applying Jensen's inequality, the EM objective function $\phi(\theta, \theta^l)$ can be derived as:

$$\phi(\theta, \theta^l)=E_{R|\overline{Y},\theta^l}(\log P(\overline{Y}, R|\theta)), \tag{15}$$

where θ^l is the parameter estimated from previous step l. The right-hand side of (15) represents taking the expected value of the log-likelihood function with respect to the conditional distribution of R given $\overline{Y}$ under the current estimate of the parameter θ^l.

M-Step: The computed expected log-likelihood is then maximized to update θ:

$$\begin{aligned} &\theta^{l+1}=\arg_\theta \max(\phi(\theta, \theta^l)) \\ &s.t. \;\; \theta_L \leq \theta \leq \theta_U, \end{aligned} \tag{16}$$

where θ_L and θ_U denote the lower bound and upper bound of θ respectively. They can be obtained from prior information on θ.

Given that EM addresses an optimization problem, the selection of initial values for the parameters can adhere to established practices for initializing values in optimization problems. For instance, one common approach involves assigning random values to the variables within a specified range. Alternatively, domain-specific knowledge or heuristics may be employed to inform the selection of initial values.

3.1. *E-step:computing the expected log-likelihood*

In the E-step, we calculate the function $\phi(\theta, \theta^l)$.

$$\begin{aligned} \phi(\theta, \theta^l)=&E_{R|\overline{Y},\theta^l}(\log(P(\overline{Y}, R|\theta))) \\ =&\Sigma_s\Sigma_i p_i^s \log P(Y^s, R_i^s|\theta), \end{aligned} \tag{17}$$

where $p_i^s=P(R_i^s|Y^s, \theta^l)$, R_i^s is the ith possible realization of the discrete Markov chain $\{r_t\}_{t=0}^{L_s}$, and $(R_i^s)_t$ represents the value of the Markov chain at t in R_i^s. Applying the Bayes' rule, (17) can be rewritten as

$$\begin{aligned} \Sigma_s\Sigma_i p_i^s \log(P(Y^s, R_i^s|\theta))=&\Sigma_s\Sigma_i p_i^s \log(P(Y^s|R_i^s, \theta)) \\ &+ \Sigma_s\Sigma_i p_i^s \log(P(R_i^s|\theta)). \end{aligned} \tag{18}$$

For the first term in (18), enumerating all the possible state transition cases, we have

$$\begin{aligned} &\Sigma_s\Sigma_i p_i^s \log(P(Y^s|R_i^s, \theta)) \\ &=\Sigma_s\Sigma_{t=0}^{L_s}\Sigma_i p_i^s \log(P(y_t^s|y_{0:t\text{-}1}^s, R_i^s, \theta)) \\ &=\Sigma_s\Sigma_{t=0}^{L_s}[\Sigma_{(i,r_t^s=s_1)} p_i^s \log(P(y_t^s|r_t^s=s_1, \theta)) \\ &\quad + \Sigma_{n=0}^{t\text{-}1}\Sigma_{(i,r_{t\text{-}(n+1):t}^s=S_{n+2})} p_i^s \\ &\qquad \times \log(P(y_t^s|y_{0:t\text{-}1}^s, r_{t\text{-}(n+1):t}^s=S_{n+2}, \theta))] \\ &=\Sigma_{s,t}[\log(P(y_t^s|r_t^s=s_1, \theta))(\Sigma_{(i,r_t^s=s_1)} p_i^s) \\ &\quad + \Sigma_{n=0}^{t\text{-}1} \log(P(y_t^s|y_{0:t\text{-}1}^s, r_{t\text{-}(n+1):t}^s=S_{n+2}, \theta)) \\ &\quad \times (\Sigma_{(i,r_{t\text{-}(n+1):t}^s=S_{n+2})} p_i^s)] \\ &=\Sigma_{s,t}[\log(P(y_t^s|r_t^s=s_1, \theta))P(r_t^s=s_1|Y^s, \theta^l) \\ &\quad + \Sigma_{n=0}^{t\text{-}1} \log(P(y_t^s|y_{0:t\text{-}1}^s, r_{t\text{-}(n+1):t}^s=S_{n+2}, \theta)) \\ &\qquad \times P(r_{t\text{-}(n+1):t}^s=S_{n+2}|Y^s, \theta^l)], \end{aligned} \tag{19}$$

where

$$\begin{aligned} r_{t\text{-}(n+1):t}^s=&[r_{t\text{-}(n+1)}, r_{t\text{-}n}, \ldots, r_t]_{1\times(n+2)}, \\ S_{n+2}=&[s_1, s_0, \ldots, s_0]_{1\times(n+2)}. \end{aligned}$$

The second row of (19) is based on Bayes' rule. For the third row, if the current state $r_t = s_1$, the current observation only depends on r_t. However, when the state $r_t = s_0$, all the possible state transition cases in past time should be taken into consideration.

The second term in (18) can be expressed as

$$\begin{aligned} &\Sigma_s\Sigma_i p_i^s \log(P(R_i^s|\theta)) \\ &=\Sigma_s\Sigma_i p_i^s \log[P(r_0^s=s_1|\theta)\Pi_{t=1}^{L_s} P(r_t^s|r_{t\text{-}1}^s, \theta)] \\ &=\Sigma_s\Sigma_{t=1}^{L_s}\Sigma_i p_i^s \log(P(r_t^s|r_{t\text{-}1}^s, \theta)) \\ &\quad +\Sigma_s\Sigma_i p_i^s \log(P(r_0^s=s_1|\theta)). \end{aligned} \tag{20}$$

The second term in (20) is zero as the initial $r_0^s=s_1$ is independent of θ. Then, we have

$$
\begin{aligned}
&\Sigma_s \Sigma_i p_i^s \log(P(R_i^s|\theta)) \\
&= \Sigma_s \Sigma_{t=1}^{L_s} [\Sigma_{r^s_{t\text{-}1:t}} \log(P(r_t^s|r_{t\text{-}1}^s, \theta)) \\
&\quad \times (\Sigma_i p_i^s 1_{\{(R_i^s)_{t\text{-}1:t} = r^s_{t\text{-}1:t}\}})] \\
&= \Sigma_s \Sigma_{t=1}^{L_s} \Sigma_{r^s_{t\text{-}1:t}} \log(P(r_t^s|r_{t\text{-}1}^s, \theta)) \\
&\quad \times P(r^s_{t\text{-}1:t}|Y^s, \theta^l),
\end{aligned} \tag{21}
$$

where

$$
1_{\{(R_i^s)_{t\text{-}1:t} = r^s_{t\text{-}1:t}\}} = \begin{cases} 1, \text{ if } (R_i^s)_{t\text{-}1:t} = r^s_{t\text{-}1:t}, \\ 0, \text{otherwise.} \end{cases} \tag{22}
$$

3.1.1. *Offline coefficient calculation in E-step*

The probabilities $P(r_t^s|Y^s, \theta^l)$ and $P(r^s_{t\text{-}(n+1):t} = S_{n+2}|Y^s, \theta^l)$ in (19) and $P(r^s_{t\text{-}1}, r^s_t|Y^s, \theta^l)$ in (21) can be calculated using the backward-forward algorithm. All the following probabilities in this section are conditioned on θ^l. For notation simplicity, we drop the notation for the conditional probability on θ^l as long as it does not cause confusion. To facilitate the backward-forward operation, we define

$$
a_t^s(r_t^s) = P(y^s_{0:t}, r_t^s), \tag{23}
$$
$$
b_t^s(r_t^s) = P(y^s_{t+1:L_s}|y^s_{0:t}, r_t^s). \tag{24}
$$

We first calculate $a_t^s(r_t^s)$ and $b_t^s(r_t^s)$ before we calculate the probabilities in (19) and (21).

Recursive calculation of a_t^s:

$a_t^s, t \geq 1$ can be expressed using the following recursive relationship based on the Bayes' rule:

$$
\begin{aligned}
a_t^s &= P(y_t^s|y^s_{0:t\text{-}1}, r_t^s) P(y^s_{0:t\text{-}1}, r_t^s) \\
&= P(y_t^s|y^s_{0:t\text{-}1}, r_t^s) [\Sigma_{r^s_{t\text{-}1}} P(r_t^s|y^s_{0:t\text{-}1}, r^s_{t\text{-}1}) P(y^s_{0:t\text{-}1}, r^s_{t\text{-}1}))] \\
&= P(y_t^s|y^s_{0:t\text{-}1}, r_t^s) [\Sigma_{r^s_{t\text{-}1}} P(r_t^s|r^s_{t\text{-}1}) a^s_{t\text{-}1}].
\end{aligned} \tag{25}
$$

Based on (5) and (6) and the fact that the state of the Markov chain at t does not depend on previous measurements, in (25), we have $P(r_t^s|y^s_{0:t\text{-}1}, r^s_{t\text{-}1}) = P(r_t^s|r^s_{t\text{-}1})$. $P(r_t^s|r^s_{t\text{-}1})$ in (25) is given by (5) and (6).

To calculate $P(y_t^s|y^s_{0:t\text{-}1}, r_t^s)$ in (25), we consider the following two cases based on the value of r_t^s:

Case 1: $r_t^s = s_1$. Since a new $\hat{z}_t$ will be generated by (3) independently from previous states and parameters at time t, it is straightforward that

$$
P(y_t^s|y^s_{0:t\text{-}1}, r_t^s = s_1) = P(y_t^s|r_t^s = s_1). \tag{26}
$$

The probability can be calculated using the following general result with n=-1.

$$
\begin{aligned}
W^s_{t\text{-}(n+1):t}(\theta_1^l) &= P(y^s_{t\text{-}(n+1):t}|r^s_{t\text{-}(n+1):t} = S_{n+2}) \\
&= \int P(y^s_{t\text{-}(n+1):t}|\hat{z}_{t\text{-}(n+1)}, r^s_{t\text{-}(n+1):t} = S_{n+2}) \\
&\quad \times f(\hat{z}_{t\text{-}(n+1)}; \theta_1^l) d\hat{z}_{t\text{-}(n+1)} \\
&= \int \Pi_{j=t\text{-}(n+1)}^t P(y_j^s|\hat{z}_{t\text{-}(n+1)}, r^s_{t\text{-}(n+1):t} = S_{n+2}) \\
&\quad \times f(\hat{z}_{t\text{-}(n+1)}; \theta_1^l) d\hat{z}_{t\text{-}(n+1)},
\end{aligned} \tag{27}
$$

where

$$
\begin{aligned}
&y_j^s|(\hat{z}_{t\text{-}(n+1)}, r^s_{t\text{-}(n+1):t} = S_{n+2}) \\
&\quad \sim N(\mu_j^s, D(\hat{z}_{t\text{-}(n+1)}, r_j) \Sigma_w D'(\hat{z}_{t\text{-}(n+1)}, r_j)),
\end{aligned} \tag{28}
$$

and

$$
\begin{aligned}
\mu_j^s = &G(\hat{z}_{t\text{-}(n+1)}, s_0) u_j^s + C(\hat{z}_{t\text{-}(n+1)}, s_0) \\
&\times [\Sigma_{l=0}^{j\text{-}(t\text{-}n)} A^l(\hat{z}_{t\text{-}(n+1)}, s_0) F(\hat{z}_{t\text{-}(n+1)}, s_0) u^s_{j\text{-}l} \\
&\quad + A^{j\text{-}(t\text{-}n\text{-}1)}(\hat{z}_{t\text{-}(n+1)}, s_0)[H(\hat{z}_{t\text{-}(n+1)}, s_1) \\
&\quad\quad + F(\hat{z}_{t\text{-}(n+1)}, s_1) u^s_{t\text{-}(n+1)}]].
\end{aligned} \tag{29}
$$

Here $W^s_{t\text{-}(n+1):t}(\theta_1^l)$ calculates the conditional probability of measurements from time $t-(n+1)$ to t under the condition that the closest state of s_1 to time t occurs at time $t-(n+1)$. This completes the calculations for Case 1.

Case 2: $r_t^s = s_0$. In this case, all the possible state switch cases in the past time should be included when calculating $P(y_t^s|y^s_{0:t\text{-}1}, r_t^s = s_0)$.

$$
\begin{aligned}
&P(y_t^s|y^s_{0:t\text{-}1}, r_t^s = s_0) \\
&= \Sigma_{n=0}^{t\text{-}1} P(y_t^s, r^s_{t\text{-}(n+1):t\text{-}1} = S_{n+1}|y^s_{0:t\text{-}1}, r_t^s = s_0) \\
&= [\Sigma_{n=0}^{t\text{-}1} P(y_t^s|y^s_{0:t\text{-}1}, r^s_{t\text{-}(n+1):t\text{-}1} = S_{n+1}, r_t^s = s_0) \\
&\quad \times P(r^s_{t\text{-}(n+1):t\text{-}1} = S_{n+1}|y^s_{0:t\text{-}1}, r_t^s = s_0)].
\end{aligned} \tag{30}
$$

To calculate $P(y_t^s|y^s_{0:t\text{-}1} in (30), r^s_{t\text{-}(n+1):t\text{-}1} = S_{n+1}, r_t^s = s_0)$, we have

$$
\begin{aligned}
&P(y_t^s|y^s_{0:t\text{-}1}, r^s_{t\text{-}(n+1):t\text{-}1} = S_{n+1}, r_t^s = s_0) \\
&= P(y_t^s|y^s_{0:t\text{-}1}, r^s_{t\text{-}(n+1):t} = S_{n+2}) \\
&= P(y_t^s|y^s_{t-(n+1):t\text{-}1}, r^s_{t\text{-}(n+1):t} = S_{n+2}) \\
&= \frac{P(y^s_{t\text{-}(n+1):t}|r^s_{t\text{-}(n+1):t} = S_{n+2})}{P(y^s_{t\text{-}(n+1):t\text{-}1}|r^s_{t\text{-}(n+1):t} = S_{n+2})} \\
&= \frac{P(y^s_{t\text{-}(n+1):t}|r^s_{t\text{-}(n+1):t} = S_{n+2})}{P(y^s_{t\text{-}(n+1):t\text{-}1}|r^s_{t\text{-}(n+1):t\text{-}1} = S_{n+1})}.
\end{aligned} \tag{31}
$$

The closest state of s_1 occurs at time $t-(n+1)$, which means $r^s_{t\text{-}(n+1)} = s_1$. $y^s_{0:t-(n+2)}$ can be discarded since y_t^s is independent of previous measurements before time $t-(n+1)$. Therefore, the third equation in (31) holds. The probabilities in (31) can be calculated using (27).

For $P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|y^s_{0:t\text{-}1}, r^s_t{=}s_0)$ in (30), we have

$$P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|y^s_{0:t\text{-}1}, r^s_t{=}s_0) \\ = \frac{P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t}{=}S_{n+2})}{P(y^s_{0:t\text{-}1}|r^s_t{=}s_0)} \\ \times P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|r^s_t{=}s_0). \quad (32)$$

where

$$P(y^s_{0:t\text{-}1}|r^s_t{=}s_0){=}\Sigma^{t\text{-}1}_{n=0}P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t}{=}S_{n+2}) \\ \times P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|r^s_t{=}s_0). \quad (33)$$

Now we calculate the probability $P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t}{=}S_{n+2})$. As r^s_t is independent of $y^s_{0:t\text{-}1}$, we have

$$P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t}{=}S_{n+2}) \\ {=}P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}), \\ {=}P(y^s_{0:t\text{-}(n+2)}|r^s_{t\text{-}(n+1)}{=}s_1) \\ \times P(y^s_{t\text{-}(n+1):t\text{-}1}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}). \quad (34)$$

For the first term in (34), we have

$$P(y^s_{0:t\text{-}(n+2)}|r^s_{t\text{-}(n+1)}{=}s_1) \\ {=}\Sigma_{r^s_{t\text{-}(n+2)}}P(y^s_{0:t\text{-}(n+2)}|r^s_{t\text{-}(n+2)}, r^s_{t\text{-}(n+1)}{=}s_1) \\ \times P(r^s_{t\text{-}(n+2)}|r^s_{t\text{-}(n+1)}{=}s_1) \\ {=}\Sigma_{r^s_{t\text{-}(n+2)}}P(y^s_{0:t\text{-}(n+2)}|r^s_{t\text{-}(n+2)})P(r^s_{t\text{-}(n+2)}|r^s_{t\text{-}(n+1)}{=}s_1) \\ {=}\Sigma_{r^s_{t\text{-}(n+2)}}P(y^s_{0:t\text{-}(n+2)}, r^s_{t\text{-}(n+2)})\frac{P(r^s_{t\text{-}(n+1)}{=}s_1|r^s_{t\text{-}(n+2)})}{P(r^s_{t\text{-}(n+1)}{=}s_1)} \\ {=}\Sigma_{r^s_{t\text{-}(n+2)}}a^s_{t\text{-}(n+2)}\frac{P(r^s_{t\text{-}(n+1)}{=}s_1|r^s_{t\text{-}(n+2)})}{P(r^s_{t\text{-}(n+1)}{=}s_1)}. \quad (35)$$

Therefore, (32) becomes

$$P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|y^s_{0:t\text{-}1}, r^s_t{=}s_0) \\ = \left(\Sigma_{r^s_{t\text{-}(n+2)}}a^s_{t\text{-}(n+2)}\frac{P(r^s_{t\text{-}(n+1)}{=}s_1|r^s_{t\text{-}(n+2)})}{P(r^s_{t\text{-}(n+1)}{=}s_1)}\right) \\ \times P(y^s_{t\text{-}(n+1):t\text{-}1}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}) \\ \times \frac{P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|r^s_t{=}s_0)}{P(y^s_{0:t\text{-}1}|r^s_t{=}s_0)}. \quad (36)$$

$P(y^s_{t\text{-}(n+1):t\text{-}1}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1})$ in (36) can be calculated using (27). The probability $P(r^s_{t\text{-}(n+1)}{=}s_1)$ can be calculated using the following recursive relationship.

$$P(r^s_t){=}\Sigma_{r^s_{t\text{-}1}}P(r^s_t, r^s_{t\text{-}1}){=}\Sigma_{r^s_{t\text{-}1}}P(r^s_t|r^s_{t\text{-}1})P(r^s_{t\text{-}1}), \quad (37)$$

where $P(r^s_t|r^s_{t\text{-}1})$ is given by (5) and (6). $P(r^s_0{=}s_1){=}1$ and $P(r^s_0{=}s_0){=}0$ as we assume $r^s_0{=}s_1$. The probability $P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|r^s_t{=}s_0)$ can be calculated based on the Bayes' rule and Markov properties as follows:

$$P(r^s_{t\text{-}(n+1):t\text{-}1}|r^s_t){=}P(r^s_t|r^s_{t\text{-}(n+1):t\text{-}1})\frac{P(r^s_{t\text{-}(n+1):t\text{-}1})}{P(r^s_t)} \\ {=}P(r^s_t|r^s_{t\text{-}1})\frac{P(r^s_{t\text{-}(n+1)})}{P(r^s_t)}\left(\prod_{l=1}^{n}P(r^s_{t\text{-}l}|r^s_{t\text{-}l\text{-}1})\right). \quad (38)$$

This completes the calculations for Case 2.

If $t{=}0$, we have $r^s_0{=}s_1$ and

$$a^s_0{=}P(y^s_0, r^s_0{=}s_1){=}P(y^s_0|r^s_0{=}s_1), \quad (39)$$

which can be calculated by (27). This completes the calculation of a^s_t.

Calculation of b^s_t:

We assign 1 to b^s_{Ls}. When $t \leq Ls$-1, b^s_t in (24) can be calculated as follows:

$$b^s_t{=}\Sigma^{t\text{-}1}_{n=0}P(y^s_{t+1:L_s}, r_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|y^s_{0:t}, r^s_t) \\ {=}\Sigma^{t\text{-}1}_{n=0}P(y^s_{t+1:L_s}|y^s_{0:t}, r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t) \\ \times P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|y^s_{0:t}, r^s_t) \\ {=}\Sigma^{t\text{-}1}_{n=0}b^s_{t,n}P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|y^s_{0:t}, r^s_t), \quad (40)$$

where $b^s_{t,n}(r^s_t){=}P(y^s_{t+1:L_s}|y^s_{0:t}, r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t)$. In the following, we describe the calculation of $P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|y^s_{0:t}, r^s_t)$ and $b^s_{t,n}(r^s_t)$.

$P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|y^s_{0:t}, r^s_t)$ can be calculated as follows,

$$P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|y^s_{0:t}, r^s_t) \\ {=}\frac{P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|r^s_t)}{P(y^s_{0:t}|r^s_t)}P(y^s_{0:t}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t). \quad (41)$$

Similar to the calculation of (32), the probability $P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|y^s_{0:t}, r^s_t)$ can be calculated based on the two probabilities $P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|r^s_t)$ and $P(y^s_{0:t}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t)$. $P(r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}|r^s_t)$ can be calculated using (38). Now we calculate $P(y^s_{0:t}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t)$. We consider two cases. If $r^s_t{=}s_0$, the probability $P(y^s_{0:t}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t)$ can be calculated using (27) by setting $n{=}t$-1. If $r^s_t{=}s_1$, we have

$$P(y^s_{0:t}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t) \\ {=}P(y^s_{0:t\text{-}1}|y^s_t, r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t{=}s_1) \\ \times P(y^s_t|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t{=}s_1) \\ {=}P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1})P(y^s_t|r^s_t{=}s_1) \quad (42)$$

$P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1})$ can be calculated by (34) and $P(y^s_t|r^s_t{=}s_1)$ can be calculated using (27).

$b^s_{t,n}(r^s_t)$ can be expressed as:

$$b^s_{t,n}(r^s_t){=}\Sigma_{r^s_{t+1}}P(y^s_{t+1:L_s}, r^s_{t+1}|y^s_{0:t}, r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t) \quad (43)$$

If $t \leq L_s$-2, (43) can be calculated using the following recursion:

$$\begin{aligned} &b^s_{t,n}(r^s_t) \\ &=\Sigma_{r^s_{t+1}} P(y^s_{t+2:L_s}|y^s_{0:t+1}, r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_{t:t+1}) \\ &\quad \times P(r^s_{t+1}|r^s_t) P(y^s_{t+1}|y^s_{0:t}, r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_{t:t+1}) \\ &= \begin{cases} \Sigma_{r^s_{t+1}} P(y^s_{t+1}|y^s_t, r^s_t{=}s_1, r^s_{t+1}) \\ \quad \times b^s_{t+1,0} P(r^s_{t+1}|r^s_t), \text{if } r^s_t{=}s_1. \\ \Sigma_{r^s_{t+1}} P(y^s_{t+1}|y^s_{0:t}, r^s_{t\text{-}(n+1):t}{=}S_{n+2}, r^s_{t+1}) \\ \quad \times b^s_{t+1,n+1} P(r^s_{t+1}|r^s_t), \text{if } r^s_t{=}s_0. \end{cases} \end{aligned} \tag{44}$$

If $r^s_{t+1}{=}s_1$, then

$P(y^s_{t+1}|y^s_t, r^s_t{=}s_1, r^s_{t+1})$ and $P(y^s_{t+1}|y^s_{0:t}, r^s_{t\text{-}(n+1):t}{=}S_{n+2}, r^s_{t+1})$

can be calculated using (27). If $r^s_{t+1}{=}s_0$, both probabilities can be calculated using (31). If $t{=}L_s$-1, (43) can be expressed as:

$$\begin{aligned} b^s_{t,n}(r^s_t) &= \Sigma_{r^s_{t+1}} P(y^s_{t+1}|y^s_{0:t}, r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_{t:t+1}) \\ &\quad \times P(r^s_{t+1}|r^s_t) \\ &= \begin{cases} \Sigma_{r^s_{t+1}} P(y^s_{t+1}|y^s_t, r^s_t{=}s_1, r^s_{t+1}) \\ \quad \times P(r^s_{t+1}|r^s_t), \text{if } r^s_t{=}s_1. \\ \Sigma_{r^s_{t+1}} P(y^s_{t+1}|y^s_{0:t}, r^s_{t\text{-}(n+1):t}{=}S_{n+2}, r^s_{t+1}) \\ \quad \times P(r^s_{t+1}|r^s_t), \text{if } r^s_t{=}s_0. \end{cases} \end{aligned} \tag{45}$$

The probabilities in (45) can be calculated in the same way as those in (44). This completes the calculation of b^s_t.

Calculation of $P(r^s_t|Y^s)$ in (19):

$$P(r^s_t|Y^s) = \frac{P(y^s_{t+1:L_s}|y^s_{0:t}, r^s_t) P(y^s_{0:t}, r^s_t)}{P(Y^s)} = \frac{a^s_t(r^s_t) b^s_t(r^s_t)}{P(Y^s)} \tag{46}$$

where

$$P(Y^s) = \Sigma_{r^s_t} a^s_t(r^s_t) b^s_t(r^s_t).$$

Similar to the calculation of (32), the probability $P(r^s_t|Y^s)$ is calculated using a^s_t, b^s_t.

Calculation of $P(r^s_{t\text{-}1}, r^s_t|Y^s)$ in (21):

$$\begin{aligned} &P(r^s_{t\text{-}1}, r^s_t|Y^s) \\ &= \frac{P(y^s_t, r^s_t, y^s_{t+1:L_s}|y^s_{0:t\text{-}1}, r^s_{t\text{-}1}) P(y^s_{0:t\text{-}1}, r^s_{t\text{-}1})}{P(Y^s)} \\ &= \frac{P(y^s_{t+1:L_s}|y^s_{0:t}, r^s_{t\text{-}1}, r^s_t) P(y^s_t|y^s_{0:t\text{-}1}, r^s_{t\text{-}1}, r^s_t)}{P(Y^s)} \\ &\quad \times P(r^s_t|r^s_{t\text{-}1}) a^s_{t\text{-}1}. \end{aligned} \tag{47}$$

Similar to the calculation of probability in (46), we only need to calculate the probabilities $P(y^s_{t+1:L_s}|y^s_{0:t}, r^s_{t\text{-}1}, r^s_t), P(y^s_t|y^s_{0:t\text{-}1}, r^s_{t\text{-}1}, r^s_t), P(r^s_t|r^s_{t\text{-}1})$ and $a^s_{t\text{-}1}$.

To calculate $P(y^s_{t+1:L_s}|y^s_{0:t}, r^s_{t\text{-}1}, r^s_t)$, we consider the following two cases:

Case 1: $r^s_{t\text{-}1}{=}s_1$,

$$P(y^s_{t+1:L_s}|y^s_{0:t}, r^s_{t\text{-}1}{=}s_1, r^s_t) = b^s_{t,0}. \tag{48}$$

Case 2: $r^s_{t\text{-}1}{=}s_0$,

$$\begin{aligned} &P(y^s_{t+1:L_s}|y^s_{0:t}, r^s_{t\text{-}1}{=}s_0, r^s_t) \\ &= \Sigma^{t\text{-}1}_{n=1} P(y^s_{t+1:L_s}, r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|y^s_{0:t}, r^s_{t\text{-}1}{=}s_0, r^s_t) \\ &= \Sigma^{t\text{-}1}_{n=1} P(y^s_{t+1:L_s}|y^s_{0:t}, r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t) \\ &\quad \times P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|y^s_{0:t}, r^s_{t\text{-}1}{=}s_0, r^s_t) \\ &= \Sigma^{t\text{-}1}_{n=1} b^s_{t,n} P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|y^s_{0:t}, r^s_{t\text{-}1}{=}s_0, r^s_t), \end{aligned} \tag{49}$$

where

$$\begin{aligned} &P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|y^s_{0:t}, r^s_{t\text{-}1}{=}s_0, r^s_t) \\ &= \frac{P(y^s_{0:t}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t)}{P(y^s_{0:t}|r^s_{t\text{-}1}{=}s_0, r^s_t)} \\ &\quad \times P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|r^s_{t\text{-}1}{=}s_0). \end{aligned} \tag{50}$$

Similar to the calculation of (32), the probability $P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|y^s_{0:t}, r^s_{t\text{-}1}{=}s_0, r^s_t)$ is calculated from the probabilities $P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|r^s_{t\text{-}1}{=}s_0)$ and $P(y^s_{0:t}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t)$. $P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|r^s_{t\text{-}1}{=}s_0)$ is calculated using (38). Based on the value of r^s_t, we have

$$\begin{aligned} &P(y^s_{0:t}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t) \\ &= \begin{cases} P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}) \\ \quad \times P(y^s_t|r^s_t{=}s_1), \text{if } r^s_t{=}s_1. \\ P(y^s_{0:t}|r^s_{t\text{-}(n+1):t}{=}S_{n+2}), \text{if } r^s_t{=}s_0. \end{cases} \end{aligned} \tag{51}$$

If $r^s_t{=}s_1$,

$$\begin{aligned} &P(y^s_{0:t}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}, r^s_t) \\ &= P(y^s_t|r^s_t{=}s_1) P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1}), \end{aligned} \tag{52}$$

where $P(y^s_t|r^s_t{=}s_1)$ is calculated using (26), and $P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t\text{-}1}{=}S_{n+1})$ is calculated using (34). If $r^s_t{=}s_0$, the probability is calculated using (34). This completes the calculation of $P(y^s_{t+1:L_s}|y^s_{0:t}, r^s_{t\text{-}1}, r^s_t)$.

For the probability $P(y^s_t|y^s_{0:t\text{-}1}, r^s_{t\text{-}1}, r^s_t)$ in (47), it can be calculated using (26) if $r^s_t{=}s_1$. The probability can be calculated using (31) if $r^s_{t\text{-}1}{=}s_1$ and $r^s_t{=}s_0$. When $r^s_{t\text{-}1}{=}s_0$ and $r^s_t{=}s_0$, we have

$$\begin{aligned} &P(y^s_t|y^s_{0:t\text{-}1}, r^s_{t\text{-}1}{=}s_0, r^s_t{=}s_0) \\ &= \Sigma^{t\text{-}1}_{n=1} P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|y^s_{0:t\text{-}1}, r^s_{t\text{-}1}{=}s_0, r^s_t{=}s_0) \\ &\quad \times P(y^s_t|y^s_{0:t\text{-}1}, r^s_{t\text{-}(n+1):t}{=}S_{n+2}), \end{aligned} \tag{53}$$

where

$$\begin{aligned} &P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|y^s_{0:t\text{-}1}, r^s_{t\text{-}1}{=}s_0, r^s_t{=}s_0) \\ &= \frac{P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t}{=}S_{n+2})}{P(y^s_{0:t\text{-}1}|r^s_{t\text{-}1}{=}s_0, r^s_t{=}s_0)} \\ &\quad \times P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|r^s_{t\text{-}1}{=}s_0). \end{aligned} \tag{54}$$

Similar to the calculation of (32), the probability $P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|y^s_{0:t\text{-}1}, r^s_{t\text{-}1}{=}s_0, r^s_t{=}s_0)$ can be computed from the probabilities $P(y^s_{0:t\text{-}1}|r^s_{t\text{-}(n+1):t}{=}S_{n+2})$ and $P(r^s_{t\text{-}(n+1):t\text{-}2}{=}S_n|r^s_{t\text{-}1}{=}s_0)$. The probability

$P(y^s_t|y^s_{0:t-1}, r^s_{t-(n+1):t}{=}S_{n+2})$ in (53) and (54) can be calculated using (31). $P(r^s_{t-(n+1):t-2}{=}S_n|r^s_{t-1}{=}s_0)$ can be calculated with (38).

Calculation of $P(r^s_{t-(n+1):t}{=}S_{n+2}|Y^s)$ in (19):

$$\begin{aligned}P(r^s_t{=}s_0|Y^s)&=\Sigma^{t-1}_{n=0}P(r^s_{t-(n+1):t}{=}S_{n+2}|Y^s)\\&=\Sigma^{t-1}_{n=0}\frac{P(r^s_{t-(n+1):t}{=}S_{n+2}, Y^s)}{P(Y^s)}.\end{aligned} \tag{55}$$

That is,

$$P(Y^s)=\Sigma^{t-1}_{n=0}\frac{P(r^s_{t-(n+1):t}{=}S_{n+2}, Y^s)}{P(r^s_t{=}s_0|Y^s)}. \tag{56}$$

Similar to the calculation of the probability in (32), the probability $P(r^s_{t-(n+1):t}{=}S_{n+2}|Y^s)$ is calculated as

$$\frac{P(r^s_{t-(n+1):t}{=}S_{n+2}, Y^s)}{P(r^s_t{=}s_0|Y^s)}, \tag{57}$$

where $P(r^s_t{=}s_0|Y^s)$ is given by (46), and

$$\begin{aligned}&P(r^s_{t-(n+1):t}{=}S_{n+2}, Y^s)\\&=P(r^s_{t-(n+1):t}{=}S_{n+2}, y^s_{0:t-(n+2)}, y^s_{t-(n+1):t}, y^s_{t+1:L_s})\\&=P(y^s_{t+1:L_s}|r^s_{t-(n+1):t}{=}S_{n+2}, y^s_{0:t})\\&\quad\times P(r^s_{t-(n+1):t}{=}S_{t-(n+1):t}, y^s_{0:t-(n+2)}, y^s_{t-(n+1):t})\\&=b^s_{t,n}(s_0)P(y^s_{0:t-(n+2)}|r^s_{t-(n+1)}{=}s_1)P(r^s_{t-(n+1):t}{=}S_{n+2})\\&\quad\times P(y^s_{t-(n+1):t}|r^s_{t-(n+1):t}{=}S_{n+2}).\end{aligned} \tag{58}$$

$P(y^s_{0:t-n-2}|r^s_{t-n-1}{=}s_1)$ is calculated using (35). $P(y^s_{t-n-1:t}|r^s_{t-(n+1):t}{=}S_{n+2})$ is calculated using (27). $P(r^s_{t-(n+1):t}{=}S_{n+2})$ can be calculated with (37).

The probability $P(r^s_t|r^s_{t-1}, \theta)$ in (21) can be expressed as a function of θ using (5) and (6). The probabilities $P(y^s_t|r^s_t{=}s_1, \theta)$ and $P(y^s_t|y^s_{0:t-1}, r^s_{t-(n+1):t}{=}S_{n+2}, \theta)$ in (19) can be expressed as a function of θ using (27). The closed-form functions for the two probabilities based on the integration of $f(z; \theta_1)$ are needed for the M-step.

In summary, we have

$$\begin{aligned}\phi(\theta, \theta^l)=&\Sigma_{s,t}[\log(W^s_{t:t}(\theta_1))P(r^s_t{=}s_1|Y^s)\\&+\Sigma^{t-1}_{n=0}\log(\frac{W^s_{t-(n+1):t}(\theta_1)}{W^s_{t-(n+1):t-1}(\theta_1)})P(r^s_{t-(n+1):t}{=}S_{n+2}|Y^s)]\\&+\Sigma_s\Sigma^{L_s}_{t=1}\Sigma_{s_i,s_j}[\log(p_{r_{t-1}=s_i,r_t=s_j}(t))\\&\times P(r^s_{t-1}{=}s_i, r^s_t{=}s_j|Y^s)].\end{aligned} \tag{59}$$

3.1.2. *Recursive coefficient calculation in E-Step*

The coefficient calculation in section 3.1.1 is not computationally efficient to be applied recursively as it requires a backward operation (i.e., for (24)) on the entire observations each time a new observation is available. On-line EM algorithm also has been developed and used in recursive parameter identification (see, for example, [24–27]). An important step in applying the estimation method recursively is to recursively update the coefficients in (59). In this section, we provide an algorithm that recursively updates the coefficients using the particle filter introduced in [28].

For simplicity, we assume that there is only one set of observation Y (i.e., $s{=}1$). The observation set at time $t{=}N$ is given as $Y_N{=}\{y_0, y_1, \dots, y_N\}$. (59) can be simplified as

$$\begin{aligned}\phi(\theta, \theta^l)=&\Sigma_t[\log(W_{t:t}(\theta_1))P(r_t{=}s_1|Y)\\&+\Sigma^{t-1}_{n=0}\log(\frac{W_{t-(n+1):t}(\theta_1)}{W_{t-(n+1):t-1}(\theta_1)})P(r_{t-(n+1):t}{=}S_{n+2}|Y)]\\&+\Sigma_{s_i,s_j}\Sigma^N_{t=1}[\log(p_{r_{t-1}=s_i,r_t=s_j}(t))\\&\times P(r_{t-1}{=}s_i, r_t{=}s_j|Y)].\end{aligned} \tag{60}$$

We further assume that $\log(W_{t:t}(\theta_1))$, $\log(\frac{W_{t-(n+1):t}(\theta_1)}{W_{t-(n+1):t-1}(\theta_1)})$, $\log(p_{r_{t-1},r_t}(t))$ can be decomposed into two parts. One is independent of t and the other one is a function of t. Without loss of generality, we assume the following relationship:

$$\begin{aligned}&\log(W_{t:t}(\theta_1)){=}F(\theta_1)\overline{F}(t),\\&\log(\frac{W_{t-(n+1):t}(\theta_1)}{W_{t-(n+1):t-1}(\theta_1)}){=}H(n, \theta_1)\overline{H}(t, n),\\&\log(p_{r_{t-1}=s_i,r_t=s_j}(t)){=}J(s_i, s_j, \theta_2)\overline{J}(t, r_{t-1}{=}s_i, r_t{=}s_j).\end{aligned}$$

Note that

$$E(1_{r_t=s_1}|Y_N){=}P(r_t{=}s_1|Y_N), \tag{61}$$

$$E(1_{r_{t-(n+1):t}=S_{n+2}}|Y_N){=}P(r_{t-(n+1):t}{=}S_{n+2}|Y_N), \tag{62}$$

$$E(1_{r_{t-1}=s_i,r_t=s_j}|Y_N){=}P(r_{t-1}{=}s_i, r_t{=}s_j|Y_N). \tag{63}$$

where

$$1_{r_t=s_1}=\begin{cases}1, & \text{if } r_t{=}s_1,\\0, & \text{otherwise.}\end{cases} \tag{64}$$

$1_{r_{t-(n+1):t}=S_{n+2}}$ and $1_{r_{t-1}=s_i,r_t=s_j}$ are analogous to $1_{r_t=s_1}$. Then, (60) could be written as

$$\begin{aligned}&\phi(\theta, \theta^l)\\&=[F(\theta_1)(\Sigma^N_t(\overline{F}(t)E(1_{r_t=s_1}|Y_N)))]\\&\quad+\Sigma^{N-1}_{n=0}[H(n, \theta_1)\Sigma^N_{t=n+1}\overline{H}(t, n)E(1_{r_{t-(n+1):t}=S_{n+2}}|Y_N)]\\&\quad+\Sigma_{s_i,s_j}[J(s_i, s_j, \theta_2)(\Sigma^N_{t=1}\overline{J}(t, r_{t-1}{=}s_i, r_t{=}s_j)\\&\quad\times E(1_{r_{t-1}=s_i,r_t=s_j}|Y_N)))]\\&=F(\theta_1)E[\Sigma^N_t(\overline{F}(t)1_{r_t=s_1}|Y_N)]\\&\quad+\Sigma^{N-1}_{n=0}(H(n, \theta_1)E[\Sigma^N_{t=n+1}\overline{H}(t, n)1_{r_{t-(n+1):t}=S_{n+2}}|Y_N])\\&\quad+\Sigma_{s_i,s_j}(J(s_i, s_j, \theta_2)\\&\quad\times E[\Sigma^N_{t=1}\overline{J}(t, r_{t-1}{=}s_i, r_t{=}s_j)1_{r_{t-1}=s_i,r_t=s_j}|Y_N]).\end{aligned} \tag{65}$$

Let

$$\overline{S}_N = E[\Sigma_t^N(\overline{F}(t)1_{r_t=s_1}|Y_N)], \quad (66)$$

$$S_{N,n} = E[\Sigma_{t=n+1}^N \overline{H}(t,n)1_{r_{t\text{-}(n+1):t}=S_{n+2}}|Y_N], n=0, ..., N\text{-}1. \quad (67)$$

$$\widehat{S}_N = E[\Sigma_{t=1}^N \overline{J}(t, r_{t\text{-}1}=s_i, r_t=s_j)1_{r_{t\text{-}1}=s_i, r_t=s_j}|Y_N]. \quad (68)$$

A recursive update of coefficients in (59) is essentially the same as a recursive update of $\overline{S}_N$, $S_{N,n}$, and $\widehat{S}_N$. In the following, we apply a particle filter to recursively update $S_{N,n}$. The other two statistics could be recursively updated in a similar way and for conciseness they are not shown in this paper.

Let

$$T_{N,n}(r_N, x_N) = E[\Sigma_{t=n+1}^N \overline{H}(t,n)1_{r_{t\text{-}(n+1):t}=S_{n+2}}|r_N, x_N, Y_N].$$

Note that

$$S_{N,n} = E[T_{N,n}|Y_N] = \int T_{N,n}P(r_N, x_N|Y_N)d_{r_N}d_{x_N} = \int T_{N,n}P(r_{0:N}, x_{0:N}|Y_N)d_{r_{0:N}}d_{x_{0:N}}. \quad (69)$$

Further, $T_{N,n}(r_N, x_N)$ and $T_{N\text{-}1,n}(r_{N\text{-}1}, x_{N\text{-}1})$ have the following recursive relationship:

$$\begin{aligned} T_{N,n}(r_N, x_N) &= \int T_{N,n}(r_N, x_N, r_{N\text{-}1}, x_{N\text{-}1}) \times P(r_{N\text{-}1}, x_{N\text{-}1}|r_N, x_N, Y_N)dr_{N\text{-}1}dx_{N\text{-}1} \\ &= \int T_{N,n}(r_N, x_N, r_{N\text{-}1}, x_{N\text{-}1}) \times P(r_{0:N\text{-}1}, x_{0:N\text{-}1}|r_N, x_N, Y_N)d_{r_{0:N\text{-}1}}d_{x_{0:N\text{-}1}} \\ &= \int T_{N,n}(r_N, x_N, r_{N\text{-}1}, x_{N\text{-}1}) \times \frac{P(y_N, r_N, x_N|r_{0:N\text{-}1}, x_{0:N\text{-}1}, Y_{N\text{-}1})}{P(y_N, r_N, x_N|Y_{N\text{-}1})} \\ &\quad \times P(r_{0:N\text{-}1}, x_{0:N\text{-}1}|Y_{N\text{-}1}) \times d_{r_{0:N\text{-}1}}d_{x_{0:N\text{-}1}}, \end{aligned} \quad (70)$$

where

$$\begin{aligned} &T_{N,n}(r_N, x_N, r_{N\text{-}1}, x_{N\text{-}1}) \\ &= E[\Sigma_{t=n+1}^{N\text{-}1}\overline{H}(t,n)1_{r_{t\text{-}(n+1):t}=S_{n+2}}|r_N, x_N, r_{N\text{-}1}, x_{N\text{-}1}, Y_N] \\ &\quad + E[\overline{H}(N,n)1_{r_{N\text{-}(n+1):N}=S_{n+2}}|r_N, x_N, r_{N\text{-}1}, x_{N\text{-}1}, Y_N] \\ &= E[\Sigma_{t=n+1}^{N\text{-}1}\overline{H}(t,n)1_{r_{t\text{-}(n+1):t}=S_{n+2}}|r_{N\text{-}1}, x_{N\text{-}1}, Y_{N\text{-}1}] \\ &\quad + \overline{H}(N,n)P(r_{N\text{-}(n+1):N}=S_{n+2}|r_N, x_N, r_{N\text{-}1}, x_{N\text{-}1}, Y_N) \\ &= \overline{H}(N,n)P(r_{N\text{-}(n+1):N}=S_{n+2}|r_N, x_N, r_{N\text{-}1}, x_{N\text{-}1}, Y_N) \\ &\quad + T_{N\text{-}1,n}(r_{N\text{-}1}, x_{N\text{-}1}), \end{aligned} \quad (71)$$

with $T_{N\text{-}1,n}(r_{N\text{-}1}, x_{N\text{-}1})=0,$ for $n=N\text{-}1$.

Using (69-71), $S_{N,n}$ can be updated recursively. However, the integration in the equations is generally not easy to evaluate. In this case, a particle filter can be used to approximate the integration.

Approximation of the integrations in (69-70) with a particle filer:

Assuming that there are M particles, at $t=N\text{-}1$ they are given as $\{r^i_{0:N\text{-}1}, x^i_{0:N\text{-}1}, \omega^i_{N\text{-}1}\}^M_i$. $\omega^i_{N\text{-}1}$ is the weight associated with particle i and is defined as follows:

$$\omega^i_{N\text{-}1} = \frac{P(r^i_{0:N\text{-}1}, x^i_{0:N\text{-}1}|Y_{N\text{-}1})}{\pi(r^i_{0:N\text{-}1}, x^i_{0:N\text{-}1}|Y_{N\text{-}1})}\frac{1}{M}, \quad (72)$$

where π is the importance density function and $\pi(r^i_{0:N\text{-}1}, x^i_{0:N\text{-}1}|Y_{N\text{-}1}) = \frac{1}{M}$. In this study, π is assumed to have the following form:

$$\pi(r_{0:N}, x_{0:N}|Y_N) = \pi(r_N, x_N|r_{0:N\text{-}1}, x_{0:N\text{-}1}, Y_N)\pi(r_{0:N\text{-}1}, x_{0:N\text{-}1}|Y_{N\text{-}1}). \quad (73)$$

We define that

$$\pi(r_N, x_N|r_{0:N\text{-}1}, x_{0:N\text{-}1}, Y_N) = P(r_N, x_N|r_{0:N\text{-}1}, x_{0:N\text{-}1}, Y_{N\text{-}1}). \quad (74)$$

When y_N is available at $t=N$, for particle i, sample $\{r^i_N, x^i_N\}^M_{i=1}$ according to $\pi(r_N, x_N|r^i_{0:N\text{-}1}, x^i_{0:N\text{-}1}, Y_N)$. Using (72 -74), the weight associated with each particle is given as:

$$\begin{aligned} \omega^i_N &= \frac{P(r^i_{0:N}, x^i_{0:N}|Y_N)}{\pi(r^i_{0:N}, x^i_{0:N}|Y_N)}\frac{1}{M} \\ &= \frac{P(y_N, r^i_N, x^i_N|r^i_{0:N\text{-}1}, x^i_{0:N\text{-}1}, Y_{N\text{-}1})}{P(r^i_N, x^i_N|r^i_{0:N\text{-}1}, x^i_{0:N\text{-}1}, Y_{N\text{-}1})} \times \frac{P(r^i_{0:N\text{-}1}, x^i_{0:N\text{-}1}|Y_{N\text{-}1})}{P(y_N|Y_{N\text{-}1})\pi(r^i_{0:N\text{-}1}, x^i_{0:N\text{-}1}|Y_{N\text{-}1})M} \\ &\propto P(y_N|r^i_N, x^i_N)\omega^i_{N\text{-}1}. \end{aligned} \quad (75)$$

Therefore, ω^i_N can be updated as follows:

$$\omega^i_N = \frac{\rho^i_N\omega^i_{N\text{-}1}}{\Sigma_j\rho^j_N\omega^j_{N\text{-}1}}, \quad (76)$$

where $\rho^j_N=P(y_N|r^j_N, x^j_N)$ and can be calculated by using (28) and (29). Then, $S_{N,n}$ can be approximated as follows:

$$S_{N,n} \approx \Sigma^M_i T_{N,n}(r^i_N, x^i_N)\omega^i_N. \quad (77)$$

Let $\alpha^j_{i,N}=P(y_N, r^i_N, x^i_N|r^j_{0:N\text{-}1}, x^j_{0:N\text{-}1}, Y_{N\text{-}1})$, by (70), we have

$$\begin{aligned} T_{N,n}(r^i_N, x^i_N) &\approx \Sigma_j[T_{N,n}(r^i_N, x^i_N, r^j_{N\text{-}1}, x^j_{N\text{-}1}) \times \frac{\alpha^j_{i,N}}{P(y_N, r^i_N, x^i_N|Y_{N\text{-}1})}\frac{P(r^j_{0:N\text{-}1}, x^j_{0:N\text{-}1}|Y_{N\text{-}1})}{\pi(r^j_{0:N\text{-}1}, x^j_{0:N\text{-}1}|Y_{N\text{-}1})}\frac{1}{M}] \\ &= \Sigma^M_{j=1}[T_{N,n}(r^i_N, x^i_N, r^j_{N\text{-}1}, x^j_{N\text{-}1})\frac{\alpha^j_{i,N}\omega^j_{N\text{-}1}}{P(y_N, r^i_N, x^i_N|Y_{N\text{-}1})}]. \end{aligned} \quad (78)$$

Note that

$$\alpha^j_{i,N} \propto \frac{P(y_N, r^i_N, x^i_N|r^j_{0:N\text{-}1}, x^j_{0:N\text{-}1}, Y_{N\text{-}1})}{P(y_N, r^i_N, x^i_N|Y_{N\text{-}1})}\omega^j_{N\text{-}1}. \quad (79)$$

Therefore,

$$T_{N,n}(r_N^i, x_N^i) \approx \Sigma_j^M [T_{N,n}(r_N^i, x_N^i, r_{N\text{-}1}^j, x_{N\text{-}1}^j) \frac{\alpha_{i,N}^j \omega_{N\text{-}1}^j}{\Sigma_l \alpha_{i,N}^l \omega_{N\text{-}1}^l}]. \quad (80)$$

By (71), we have

$$T_{N,n}(r_N^i, x_N^i, r_{N\text{-}1}^j, x_{N\text{-}1}^j) = T_{N\text{-}1,n}(r_{N\text{-}1}^j, x_{N\text{-}1}^j) + \overline{H}(N,n) P(r_{N\text{-}(n+1):N} = S_{n+2} | r_N^i, x_N^i, r_{N\text{-}1}^j, x_{N\text{-}1}^j, Y_N), \quad (81)$$

where

$$P(r_{N\text{-}(n+1):N} = S_{n+2} | r_N^i, x_N^i, r_{N\text{-}1}^j, x_{N\text{-}1}^j, Y_N) = \int P(r_{N\text{-}(n+1):N} = S_{n+2}, \overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2} | r_N^i, x_N^i, r_{N\text{-}1}^j, x_{N\text{-}1}^j, Y_N) d_{\overline{r}_{0:N\text{-}2}} d_{\overline{x}_{0:N\text{-}2}}. \quad (82)$$

Note that we have

$$P(r_{N\text{-}(n+1):N} = S_{n+2}, \overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2} | r_N^i, x_N^i, r_{N\text{-}1}^j, x_{N\text{-}1}^j, Y_N) = \frac{P_1 P_2 P_3 P_4 P_5 P_6 P_7}{P_8} P(\overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2} | Y_{N\text{-}2}), \quad (83)$$

where

$P_1 = P(y_N | x_N^i, x_{N\text{-}1}^j, r_N^i, r_{N\text{-}1}^j, \overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2}, r_{N\text{-}(n+1):N} = S_{n+2}, Y_{N\text{-}1})$,
$P_2 = P(x_N^i | x_{N\text{-}1}^j, r_N^i, r_{N\text{-}1}^j, \overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2}, r_{N\text{-}(n+1):N} = S_{n+2}, Y_{N\text{-}1})$,
$P_3 = P(r_N^i | x_{N\text{-}1}^j, r_{N\text{-}1}^j, \overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2}, r_{N\text{-}(n+1):N} = S_{n+2}, Y_{N\text{-}1})$,
$P_4 = P(y_{N\text{-}1} | x_{N\text{-}1}^j, r_{N\text{-}1}^j, \overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2}, r_{N\text{-}(n+1):N} = S_{n+2}, Y_{N\text{-}2})$,
$P_5 = P(x_{N\text{-}1}^j | r_{N\text{-}1}^j, \overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2}, r_{N\text{-}(n+1):N} = S_{n+2}, Y_{N\text{-}2})$,
$P_6 = P(r_{N\text{-}1}^j | \overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2}, r_{N\text{-}(n+1):N} = S_{n+2}, Y_{N\text{-}2})$,
$P_7 = P(r_{N\text{-}(n+1):N} = S_{n+2} | \overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2}, Y_{N\text{-}2})$,
$P_8 = P(y_{N\text{-}1:N}, x_N^i, r_N^i, x_{N\text{-}1}^j, r_{N\text{-}1}^j | Y_{N\text{-}2})$.

(82) becomes

$$P(r_{N\text{-}(n+1):N} = S_{n+2} | r_N^i, x_N^i, r_{N\text{-}1}^j, x_{N\text{-}1}^j, Y_N) = \int \frac{P_1 P_2 P_3 P_4 P_5 P_6 P_7}{P_8} \frac{P(\overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2} | Y_{N\text{-}2})}{\pi(\overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2} | Y_{N\text{-}2})} \times \pi(\overline{r}_{0:N\text{-}2}, \overline{x}_{0:N\text{-}2} | Y_{N\text{-}2}) d_{\overline{r}_{0:N\text{-}2}} d_{\overline{x}_{0:N\text{-}2}} \approx \Sigma_l^M \frac{P_1 P_2 P_3 P_4 P_5 P_6 P_7}{P_8} \omega_{N\text{-}2}^l, \quad (84)$$

where P_1 and P_4 can be calculated using (27) and (31), P_2 and P_5 can be calculated using the relationship and probability density function specified in (3), and P_3, P_6 and P_7 can be calculated using (5) and (6). We also have

$$\Sigma_{n=-1}^{N\text{-}1} P(r_{N\text{-}(n+1):N} = S_{n+2} | r_N^i, x_N^i, r_{N\text{-}1}^j, x_{N\text{-}1}^j, Y_N) = 1. \quad (85)$$

P_8 is independent of index n in (85) and the particles $\{r_{0:N\text{-}2}^l, x_{0:N\text{-}2}^l, \omega_{N\text{-}2}^l\}_{l=1}^M$. Therefore, we have

$$P(r_{N\text{-}(n+1):N} = S_{n+2} | r_N^i, x_N^i, r_{N\text{-}1}^j, x_{N\text{-}1}^j, Y_N) \propto P_1 P_2 P_3 P_4 P_5 P_6 P_7 \omega_{N\text{-}2}^l. \quad (86)$$

The recursive update of $S_{N,n}$ is summarized as follows:

At $t=N$, we have $\{r_{0:N\text{-}1}^i, x_{0:N\text{-}1}^i, \omega_{N\text{-}1}^i\}_i^M$, $\theta^{N\text{-}1}$ from the previous M-step (i.e., at t=N-1) and y_N.

Step 1: Calculate $T_{N,n}(r_N^i, x_N^i)$, for , $i = 1, ..., M$.

1. for a given i, calculate $T_{N,n}(r_N^i, x_N^i, r_{N\text{-}1}^j, x_{N\text{-}1}^j)$ using (81-86) and $\alpha_{i,N}^j$ for j=1, ..., M.
2. calculate $T_{N,n}(r_N^i, x_N^i)$ using Eq (80).

Step 2: for particle i, sample $\{r_N^i, x_N^i\}_{i=1}^M$ according to $\pi(r_N, x_N | r_{0:N\text{-}1}^i, x_{0:N\text{-}1}^i, Y_N)$ to obtain $\{r_{0:N}^i, x_{0:N}^i\}$. Update ω_N^i using (76).

Step 3: Calculate $S_{N,n}$, n=, 0, ..., N-1, using (77).

$\overline{S}_N$ and $\widehat{S}_N$ can be recursively updated in a similar way. After calculating the three statistics, a resampling can be performed to avoid degeneracy if necessary. Then, M-step is performed to obtain θ^N.

3.2. *M-step: maximizing this computed expected log-likelihood*

In this step, the estimate of θ is updated by solving the optimization model in (16), where $\phi(\theta, \theta^l)$ is given in (59).

If a solution for the optimization problem can be expressed in a closed-form (e.g., by using the first order optimality conditions), the update of θ^l can be easily obtained. If a closed-form solution cannot be obtained, numerical methods for nonlinear optimization problems (e.g., generalized reduced gradient method, sequential quadratic method, and interior point method) can be used to find an optimal solution. For example, a natural gradient iterative method is utilized in [21,29] to solve the optimization problem in M-step with linear constraints for the estimates of transition matrix. Since the objective function of the optimization model is not necessarily convex, local optimal solutions may exist. Multiple initial solutions and different solution algorithms can be applied to avoid local optimal solutions [19].

4. Validation and Application

In this section, we provide 1) a validation of the estimation method using an airborne random mobility model (in section 4.1), and 2) an application of the estimation method for the trajectories of real fixed-wing unmanned aircraft in a field test (in section 4.2).

4.1. *Validation using the ST RMM*

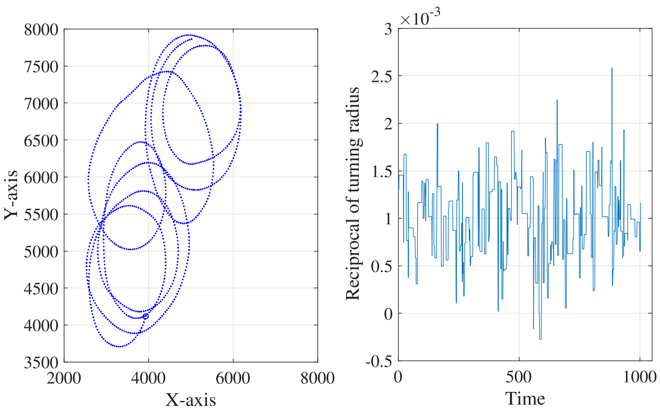
We validate the estimation method using the ST RMM introduced in section 2. For given parameters (μ, σ, and λ), we first create trajectories according to the ST RMM. We then apply the estimation methods to the trajectories to estimate these parameters and compare the parameter estimates with the given ones. The observation $(\overline{Y})$ here is the reciprocal of the turning radius calculated from the trajectories with measurement error following $N(0, \sigma_w)$. With the above assumptions and matrix settings in (9), (27) becomes:

$$W^s_{t\text{-}(n+1):t}(\theta^l_1)=[(2\pi)^{\frac{-(n+2)}{2}}[\sigma_w^{2(n+2)}+(n+2)\sigma^2\sigma_w^{2(n+1)}]^{\frac{-1}{2}}] \times \exp\{\frac{1}{2\sigma_w^2}[\Sigma^t_{l=t\text{-}(n+1)}(y^s_l\text{-}\mu)^2 - \frac{\sigma^2}{\sigma_w^2+(n+2)\sigma^2}(\Sigma^t_{l=t\text{-}(n+1)}(y^s_l\text{-}\mu))^2]\}. \quad (87)$$

We assume that

$$\mu=0.001, \sigma=0.5\times 10^{-3}, \lambda=5. \quad (88)$$

For validation, we generated 100 trajectories by simulating the 2-D ST RMM with the parameters given in (88).



Positive radius denotes right turn. A negative radius denotes left turn.

Fig. 2. Trajectory simulated by the 2-D ST RMM with parameters given in (88)

.

Each trajectory contains between 1,000 to 2,000 data points (randomly chosen). Figure 2 shows an example of the simulated trajectory and the reciprocal of the turning radius at each time point.

We assume that the prior distributions of the three parameters are independent and follow uniform distributions. The initial values for the parameters are: $\mu=0, \sigma=0.5\times 10^{-2}, \lambda=20$. The constraints $\sigma > 0, \lambda > 0$ are imposed in (16). The optimization problem in M-step is solved using the Matlab optimization toolbox with an interior-point algorithm. The estimation process is terminated if the changes in the parameter estimates are less than 0.5% for all the parameters.

For σ_w, we consider five measurement error levels in the observations: error level (EL) 1: σ_w=0.1 $\times 10^{-3}$, EL 2: σ_w=0.2$\times 10^{-3}$, EL 3: σ_w=0.3$\times 10^{-3}$, EL 4: σ_w=0.4$\times 10^{-3}$, and EL 5: σ_w=0.5 $\times 10^{-3}$.

4.1.1. *Off-line estimation*

Table 1. Parameter Estimates Different Measurement Errors

Actual Parameters	μ=0.001	σ=0.0005	λ=5
Estimated (EL 1)	0.001009	0.000493	5.098846
Estimated (EL 2)	0.000987	0.000505	4.921386
Estimated (EL 3)	0.001015	0.000511	4.904359
Estimated (EL 4)	0.001008	0.000484	4.769724
Estimated (EL 5)	0.001014	0.000499	5.139847

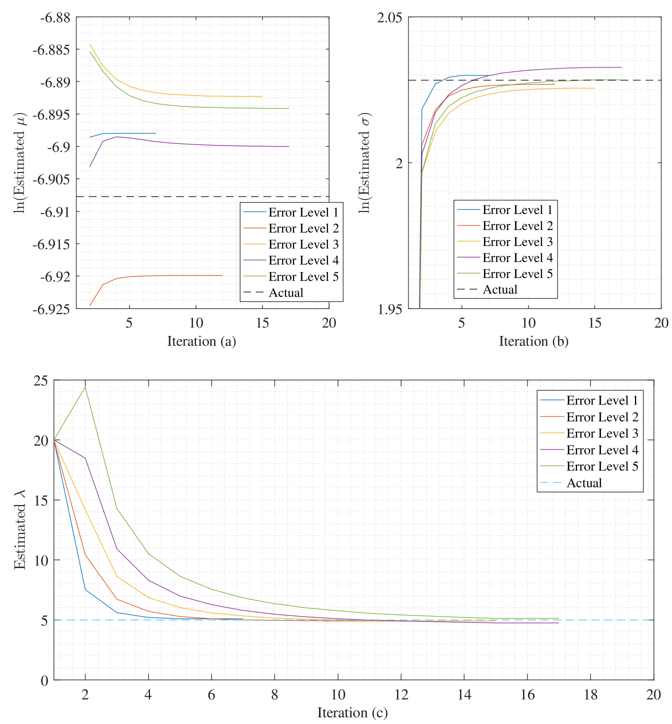


Fig. 3. Convergence of the estimation process with different error levels

The convergence of the estimation process is shown in Figure 3. The estimation processes converge with all measurement error levels. In addition, the convergence is faster if a smaller measurement error is involved in the dynamics. The parameter estimates are shown in Table 1. For all measurement errors, the estimates are close to the actual ones.

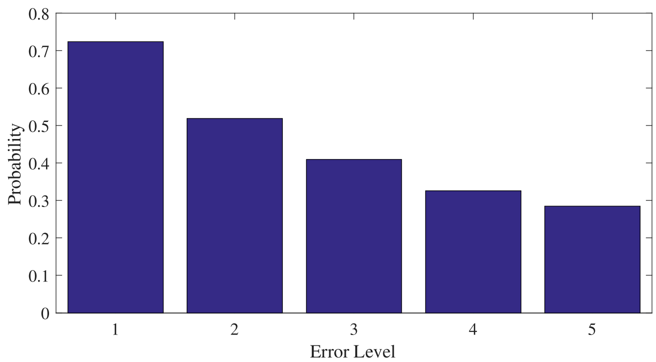


Fig. 4. Average $P(r_t^s = s_1|Y^s)$ with different measurement error levels

A slightly increasing trend can be observed in the estimation error for λ while a similar trend does not exist for the other parameter estimates. We further evaluate the probabilities of detecting the event $r_t{=}s_1$ with different measurement errors. To do so, we calculate $P(r_t^s{=}s_1|Y^s)$ using the actual parameters with different measurement errors. The results are shown in Figure 4. As measurement error increases the probability of detecting $r_t{=}s_1$ decreases. Since in the ST model the Markov chain r_t (i.e., switches) is mainly driven by parameter λ, misidentification of switches tends to have more impact on the estimation of λ than the other parameters.

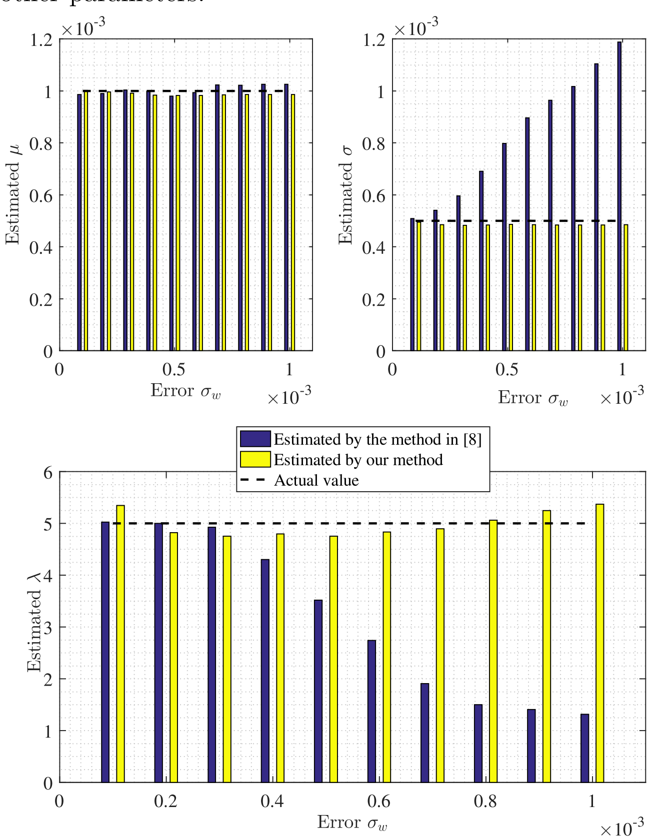


Fig. 5. Comparison of estimation results

We further compare the off-line estimation method with the method presented in [14]. We apply the two methods to one of the 100 trajectories and compare the estimation results with different error levels. For our method, we use the same settings as those used in the validation. To maintain consistency, the method in [14] is applied to the reciprocal of the turning radius. The comparison is shown in Figure 5. Both methods can generate parameter estimates close to the actual one when σ_w is small. However, as σ_w increases, our method generates parameter estimates closer to the actual ones.

4.1.2. *Recursive estimation*

For the validation of recursive estimation, we generate a flight trajectory of 30,000 points by simulating the 2-D ST mobility model with the parameters given in (88). The initial values and constraints of the parameters are the same as those in the off-line estimation setup. In the particle filter, we used 1,000 particles and implemented a resampling every 5 time steps. The recursive estimation is also applied with the five measurement error levels used in the off-line estimation.

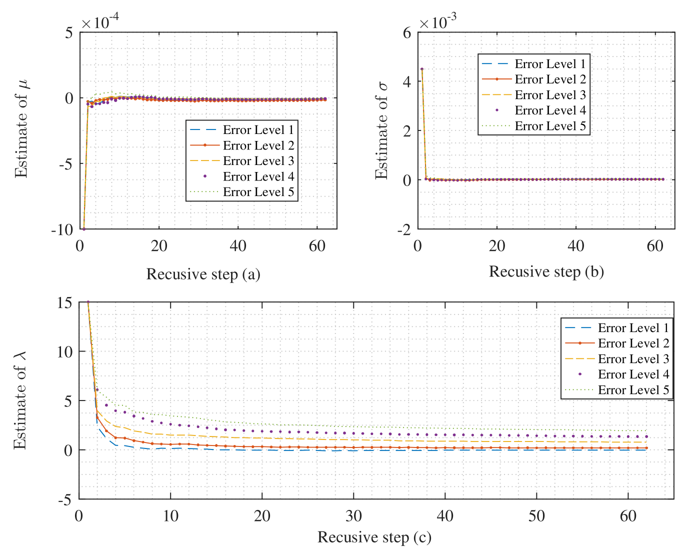


Fig. 6. Difference between actual parameters and estimated ones at every 500 recursive steps

The differences between the estimated parameters and the parameters given in (88) at every 500 recursive steps are shown in Figure 6. It can be observed the estimated values of μ and σ converge to those in (88) relatively fast with all measurement error levels. The convergence of the estimated values of λ is slower for larger measurement errors.

Table 2. Parameter Estimates with Different Measurement Errors by Recursive Estimation

Actual Parameters	$\mu = 0.001$	$\sigma = 0.0005$	$\lambda = 5$
Estimated (EL 1)	0.000994	0.000525	4.967397
Estimated (EL 2)	0.000983	0.000514	5.1990
Estimated (EL 3)	0.000993	0.000519	5.765693
Estimated (EL 4)	0.000993	0.000524	6.344167
Estimated (EL 5)	0.000998	0.000534	6.950382

Table 2 presents the final parameter estimates. The parameter estimates with error levels 1 and 2 are close to those in (88). However, the parameter estimates with error levels 3-5 are still converging and significantly different from those in (88). This is consistent with the observations in the off-line application. Particularly, Figure 3 (c) also shows a slower convergence for the estimated value of λ for larger measurement errors.

4.2. *Application using the Trajectories of Unmanned Fixed-wing Aircraft*

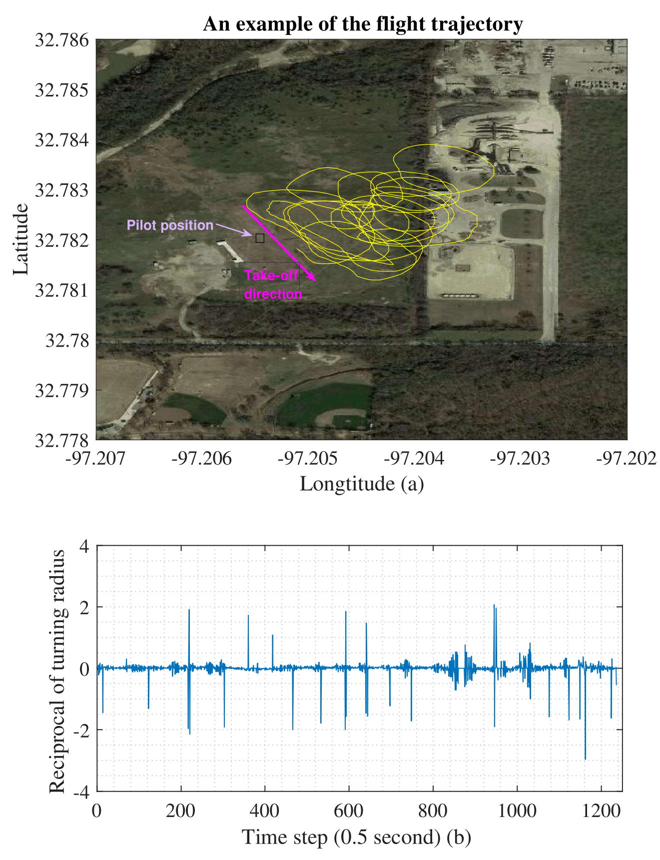


Fig. 7. Trajectory (part a) and reciprocal of turning radius (part b) of the fixed-wing aircraft with respect to time

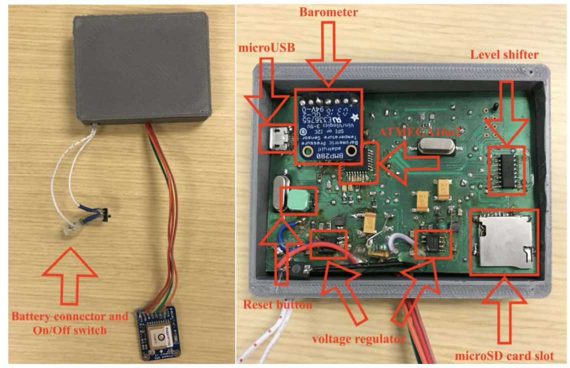


Fig. 8. UAV Attitude Acquisition System

We develop a UAV attitude acquisition system (Figure 8) and work with pilots from the Greater Southwest Aero Modelers flight club to conduct test flights using their unmanned aircraft and record the flight trajectories. This UAV attitude acquisition system consists of an IMU, GPS, barometer, and a microSD card, and is able to measure UAV angular velocity, acceleration, orientation, magnetic field, position, and altitude.

In this paper, we select nine flight trajectories flown by fixed-wing aircraft and each of the flights lasts about five minutes. It is necessary to point out that there were other aircraft flying within the same airspace when some of these flights were flown. The flight trajectories of the flights consist of position reports every 0.5 seconds. Figure 7 (a) shows one of the flight trajectories used in this study. An interview with the pilots indicates that they intended to keep their aircraft flying around the landing facility (i.e., within relatively small airspace) and avoid colliding with other aircraft based on their visual judgments. By observing their flight trajectories, we conclude that they did so by using turning maneuvers. Therefore, mobility patterns in the flight trajectories can be modeled by the ST model. We apply the estimation method to estimate the parameters μ, σ, and λ. The turning radius at time t on the trajectory is estimated as

$$R_t = \frac{v}{w},$$

where w is the angular velocity and v is the speed at t, which are estimated by using the information collected by on-board avionics. Figure 7 (b) shows the reciprocal of the turning radius at each time point. Several small turning radii can be observed, which indicates that the pilot made several sudden and sharp turns. This is consistent with the fact that the pilot was keeping the fixed-wing aircraft flying within a relatively small area and avoiding collision with other aircraft using visual judgments.

4.2.1. *Off-line estimation results*

The initial values for the parameters are as follows:

$$\mu = 0.5, \sigma = 0.1, \lambda = 5.$$

The constraints $\lambda > 0, \tau > 0$ are imposed in (16). We tentatively assume that the measurement error for the reciprocal of the turning radius is $\sigma_w = 0.05$. The estimation is terminated if the changes in the parameter estimate are less than 0.5% for all the parameters. The iterative process is terminated after five iterations. The parameter estimates are

$$\mu = -0.299, \tau = 0.655, \lambda = 20.877.$$

μ is negative, which indicates that the pilots tend to make left turns. Considering the pilots' position and the take-off direction which are similar for all nine test flights (as shown in Figure 7), pilots tend to make left turns to keep the aircraft in the vicinity. The estimated λ shows that the average time that the pilots maintain the same turning radius is about 10 seconds.

4.2.2. *Recursive estimation results*

As the nine flight trajectories are from separate flights, we apply the recursive estimation to each flight trajectory according to the time order by which the trajectories are created. The parameter estimates from one trajectory are used as the initial values for the parameter estimates of the next trajectory. The initial values for the first trajectory are the same as those in the off-line case. We use the same measurement error as the one used in the off-line case. The number of particles in the particle filter is $1,000$ and a resampling is performed at every time step. For each trajectory, the recursive estimation starts when the 100th observation becomes available.

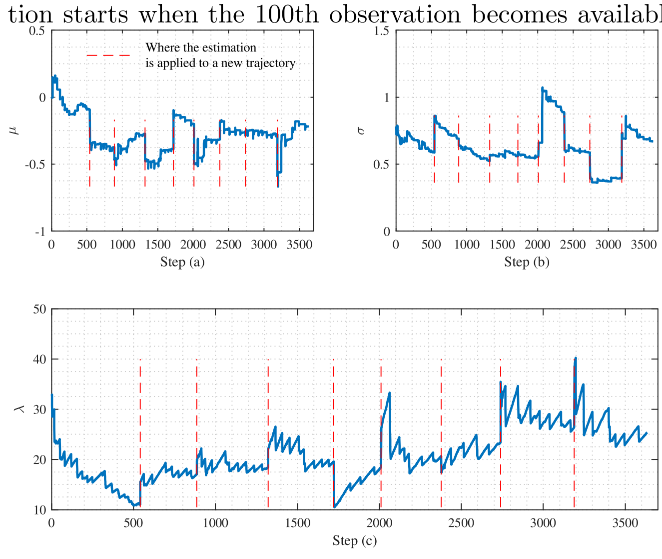


Fig. 9. Parameter estimates at each step

Figure 9 shows the parameter estimates at each step. Significant changes in the parameter estimates can be observed when the recursive estimation is applied to a new trajectory. This can be due to the fact that 1) these flights were flown by different pilots, and 2) pilots may control their aircraft differently when there were other aircraft within the same airspace. The parameter estimates after applying the method to all the trajectories are:

$$\mu = -0.208, \tau = 0.658, \lambda = 24.546.$$

They are relatively close to those estimated by the off-line estimation.

5. Summary and Conclusion

RMMs play an important role in the evaluation of mobile wireless networks in that they capture the movement characteristics of mobile agents. Particularly, with the use of UAVs as platforms for airborne communication networks, RRMs have been adopted to simulate the movement of UAVs. This study focused on the RMMs that can be formulated as random switching systems of two types of random variables. Formulating them into the special JMS, we developed a generic method based on EM to estimate the parameters in the random variables. Compared with existing estimation methods, the EM-based estimation method has the following advantages. First, the parameters in the two types of random variables are estimated simultaneously by considering all possible separations of trajectory data into maneuver sessions. Second, our estimation method is general in that it is not restricted to specific movement characteristics of a specific RMM. We verified the estimation method using simulation and showed that it outperformed an existing method. We also applied the estimation methods to real UAV trajectories to identify the parameters in the ST RMM.

Acknowledgments

We would like to thank the Army Research Office (ARO) under Grant W911NF-20-1-0132 and NSF under Grant 2235160 for the support of this work. We also thank the Greater Southwest Aero Modelers flight club for conducting the UAS test flights.

References

[1] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[2] Y. Wan, K. Namuduri, Y. Zhou, and S. Fu, "A smooth-turn mobility model for airborne networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3359–3370, Sept 2013.

[3] J. G. Markoulidakis, G. L. Lyberopoulos, D. F. Tsirkas, and E. D. Sykas, "Mobility modeling in third-generation mobile telecommunications systems," *IEEE Personal Communications*, vol. 4, no. 4, pp. 41–56, Aug 1997.

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

[5] G. Cai, J. Dias, and L. Seneviratne, "A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends," *Unmanned Systems*, vol. 02, no. 02, pp. 175–199, 2014. [Online]. Available: *https://doi.org/10.1142/S2301385014300017*

[6] S. Huang, R. S. H. Teo, W. W. L. Leong, N. Martinel, G. L. Forest, and C. Micheloni, "Coverage control of multiple unmanned aerial vehicles: A short review," Unmanned Systems*, vol. 06, no. 02, pp. 131–144, 2018. [Online]. Available: https://doi.org/10.1142/S2301385018400046*

[7] G. Skorobogatov, C. Barrado, and E. Salam, "Multiple uav systems: A survey," Unmanned Systems*, vol. 08, no. 02, pp. 149–169, 2020. [Online]. Available: https://doi.org/10.1142/S2301385020500090*

[8] P. K. Sharma and D. I. Kim, "Random 3d mobile uav networks: Mobility modeling and coverage probability," IEEE Transactions on Wireless Communications*, vol. 18, no. 5, pp. 2527–2538, 2019.*

[9] M. Banagar and H. S. Dhillon, "Fundamentals of drone cellular network analysis under random waypoint mobility model," in 2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.*

[10] M. Liu, Y. Wan, S. Li, F. L. Lewis, and S. Fu, "Learning and uncertainty-exploited directional antenna control for robust long-distance and broad-band aerial communication," IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 593–606, 2020.*

[11] W. Wang, X. Guan, B. Wang, and Y. Wang, "A novel mobility model based on semi-random circular movement in mobile ad hoc networks," Information Sciences*, vol. 180, no. 3, pp. 399 – 413, 2010.*

[12] M. Kim, D. Kotz, and S. Kim, "Extracting a mobility model from real user traces," in Proceedings of IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, April 2006, pp. 1–13.*

[13] Y. Li, D. Jin, Z. Wang, P. Hui, L. Zeng, and S. Chen, "A markov jump process model for urban vehicular mobility: Modeling and applications," IEEE Transactions on Mobile Computing*, vol. 13, no. 9, pp. 1911–1926, Sept 2014.*

[14] J. Xie, Y. Wan, B. Wang, S. Fu, K. Lu, and J. H. Kim, "A comprehensive 3-dimensional random mobility modeling framework for airborne networks," IEEE Access*, pp. 1 – 15, 2018.*

[15] M. Liu, Y. Wan, S. Li, F. L. Lewis, and S. Fu, "Learning and uncertainty-exploited directional antenna control for robust long-distance and broad-band aerial communication," IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 593–606, 2019.*

[16] M. Kim, D. Kotz, and S. Kim, "Extracting a mobility model from real user traces," in Proceedings of IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, April 2006, pp. 1–13.*

[17] T. Li, Y. Wan, M. Liu, and F. L. Lewis, "Estimation of random mobility models using the expectation-maximization method," in Proceedings of IEEE 14th International Conference on Control and Automation (ICCA)*, Anchorage, AK, 2018.*

[18] A. Logothetis and V. Krishnamurthy, "Map state sequence estimation for jump markov linear systems via the expectation-maximization algorithm," in Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 2, Dec 1997, pp. 1700–1705 vol.2.*

[19] Andrew Logothetis and V. Krishnamurthy, "Expectation maximization algorithms for map estimation of jump markov linear systems," IEEE Transactions on Signal Processing*, vol. 47, no. 8, pp. 2139–2156, Aug 1999.*

[20] F. Zheng, S. Derrode, and W. Pieczynski, "Parameter estimation in conditionally gaussian pairwise markov switching models and unsupervised smoothing," in Proceeding of 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sept 2016, pp. 1–6.*

[21] J. S. M. M. Baro, "Offline bayesian identification of jump markov nonlinear systems," in Proceedings of the 18th IFAC World Congress,Milan*, Aug 2011.*

[22] A. Doucet, A. Logothetis, and V. Krishnamurthy, "Stochastic sampling algorithms for state estimation of jump markov linear systems," IEEE Transactions on Automatic Control*, vol. 45, no. 2, pp. 188–202, Feb 2000.*

[23] U. Orguner and M. Demirekler, "Maximum likelihood estimation of transition probabilities of jump markov linear systems," IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5093–5108, Oct 2008.*

[24] C. Fritsche, E. zkan, and F. Gustafsson, "Online em algorithm for jump markov systems," in Proceedings of 2012 15th International Conference on Information Fusion*, July 2012, pp. 1941–1946.*

[25] E. zkan, F. Lindsten, C. Fritsche, and F. Gustafsson, "Recursive maximum likelihood identification of jump markov nonlinear systems," IEEE Transactions on Signal Processing*, vol. 63, no. 3, pp. 754–765, Feb 2015.*

[26] O. Cappe, "Online sequential monte carlo em algorithm," in Proceedings of 2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, Aug 2009, pp. 37–40.*

[27] S. Yildirim, S. S. Singh, and A. Doucet, "An online expectationmaximization algorithm for changepoint models," Journal of Computational and Graphical Statistics*, vol. 22, no. 4, pp. 906–926, 2013.*

[28] S. Särkkä, Bayesian Filtering and Smoothing*. Cambridge University Press, 1993.*

[29] M. Baro, J. S. Marques, and J. M. Lemos, "An improved em-method for the estimation of transition probabilities in multiple model switching systems," in Proceedings of 8th IFAC Symposium on Nonlinear

Control Systems*, vol. 43, no. 14, July 2010, pp. 374 – 378.*