

Differentiable Bootstrap Particle Filters for Regime-Switching Models

Wenhan Li

Department of Computer Science
University of Surrey
Guildford, UK
wenhan.li@surrey.ac.uk

Xiongjie Chen

Department of Computer Science
University of Surrey
Guildford, UK
xiongjie.chen@surrey.ac.uk

Wenwu Wang

Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford, UK
w.wang@surrey.ac.uk

Víctor Elvira

School of Mathematics
University of Edinburgh
Edinburgh, UK
victor.elvira@ed.ac.uk

Yunpeng Li

Department of Computer Science
University of Surrey
Guildford, UK
yunpeng.li@surrey.ac.uk

Abstract—Differentiable particle filters are an emerging class of particle filtering methods that use neural networks to construct and learn parametric state-space models. In real-world applications, both the state dynamics and measurements can switch between a set of candidate models. For instance, in target tracking, vehicles can idle, move through traffic, or cruise on motorways, and measurements are collected in different geographical or weather conditions. This paper proposes a new differentiable particle filter for regime-switching state-space models. The method can learn a set of unknown candidate dynamic and measurement models and track the state posteriors. We evaluate the performance of the novel algorithm in relevant models, showing its great performance compared to other competitive algorithms.

Index Terms—Sequential Monte Carlo, differentiable particle filters, regime switching systems.

I. INTRODUCTION

Inferring unknown quantities based on sequential observations is an important task in many real-world data analysis problems. One common example is Bayesian filtering, which aims to sequentially estimate posterior distributions of hidden states given observations in a state-space model [1]. Sequential Monte Carlo methods [2], [3], a.k.a. particle filters (PFs), are a class of Monte Carlo algorithms where the posteriors are recursively updated and approximated by a set of particles, i.e. weighted Monte Carlo samples. Since the seminal work on the bootstrap particle filter (BPF) [4], many variants of particle filters have been proposed, such as the auxiliary particle filter (APF) [5]–[8], the Gaussian sum particle filter (GSPF) [9], [10], and the Rao-Blackwellised particle filter (RBPf) [11], [12]. They are designed for non-linear non-Gaussian filtering tasks where the posteriors are analytically intractable and have been widely used in various real-world applications including geoscience [13], robotics [14], control systems [15], and machine learning [16].

Particle filters require the knowledge of state evolution (described by a dynamic model) and the link between the hidden

state and an observation (via a measurement model). It is often non-trivial to specify these models in real-world applications where complex dynamic patterns and high-dimensional observations exist [17]. An added layer of complexity is that both the state dynamics and observations can switch between a set of candidate models [18]–[24]. For example, a manoeuvring vehicle can exhibit a mixture of dynamic patterns ranging from moving through city traffic to cruising on motorways. Camera observations in autonomous vehicles are affected by light and weather conditions. This poses an interesting question on how to construct state-space models and perform particle filtering that account for a mixture of switching sub-models.

One class of solutions is to employ a bank of particle filters, one for each candidate model, before fusing the results of each filter [20]–[23]. They can incur high computational complexity when the number of candidate models is high. Another direction is to construct regime-switching particle filters [24] that augment the state space with the regime index while allowing for a flexible regime index proposal distribution¹. Both classes of methods commonly assume that candidate models either are pre-defined [20]–[24] or follow specific model structures so that model parameters can be estimated analytically [25], [26]. This restricts their applicability and effectiveness in real-world filtering tasks.

Differentiable particle filters (DPFs) are a family of recently emerging particle filtering approaches characterised by building and learning components of particle filters with neural networks through automatic differentiation [27]–[32]. Several variants [27], [28] adopt Gaussian dynamic models due to the simplicity of their differentiable implementations via the reparameterisation trick [33]. Normalising flows [34] have been adopted to construct more complicated dynamic models [31]. For measurement models, the conditional like-

¹Note that we use the terms “regime”, “pattern”, “candidate model”, and “sub-model” interchangeably throughout the paper.

likelihood of an observation can be obtained as a direct neural network output [27], [28], feature similarity [35], or derived using a conditional normalising flow [36]. To the best of our knowledge, existing differentiable particle filters have not considered dynamic and measurement models with switching regimes. While generative models such as normalising flows are expressive in theory, it is unclear whether they are effective in practice when coupled with differentiable particle filters in filtering tasks with a set of candidate models.

In this paper, we propose a new differentiable particle filter algorithm able to learn the models that govern the state dynamics and the observations in regime-switching state-space models. The resulted *regime-switching differentiable bootstrap particle filter* combines the best of both worlds – it inherits the desired properties of regime-switching particle filters including the flexibility to switch between candidate models without running separate filters, with the added benefit to learn unknown candidate models via the optimisation of neural networks. We demonstrate its effectiveness in non-linear filtering simulations with switching regimes.

The rest of the paper is organised as follows. Section II formulates the problem. Related work including regime switching particle filters and differentiable particle filters is introduced in Section III. We present the regime-switching differentiable bootstrap particle filters in Section IV. Section V provides simulation results and we conclude the paper in Section VI.

II. PROBLEM FORMULATION

We consider nonlinear filtering tasks with switching dynamic and measurement models defined as follows [24]:

$$m_0 \sim \pi(m_0), \quad (1)$$

$$m_t \sim \phi(m_t|m_{1:t-1}), \quad (2)$$

$$s_0 \sim \mu(s_0), \quad (3)$$

$$s_t \sim f_{\theta_{m_t}}(s_t|s_{t-1}), \quad (4)$$

$$o_t \sim g_{\theta_{m_t}}(o_t|s_t), \quad (5)$$

where $t \in \mathbb{N}^+$ represents the time index, $\theta_{m_t} \in \Theta$ is the parameter set of the m_t -th regime of the dynamic system. The regime indices $\{m_t\}_{t \geq 1}$ take values from a discrete space $\mathcal{M} := \{1, \dots, N_m\}$, and are distributed according to a categorical distribution $m_t \sim \mathcal{C}(p(m_t = 1|m_{1:t-1}), \dots, p(m_t = N_m|m_{1:t-1}))$. In a time-varying latent Markov process $\{s_t\}_{t \geq 1}$, the d_s -dimensional hidden state of interest s_t is generated by a mixture of N_m or less dynamic patterns. The d_o -dimensional observation o_t is generated by a measurement model defined by $g_{\theta_{m_t}}(o_t|s_t)$. The observations $\{o_t\}_{t \geq 1}$ are conditionally independent given the latent process $\{s_t\}_{t \geq 1}$. We follow the convention that vectors and matrices are denoted using bold fonts while scalars are denoted in normal font letters. The system diagram is shown in Fig. 1.

Our goal is to jointly learn the parameter set $\theta = \cup_{j=1}^{N_m} \theta_j$ and track the posterior distributions $p(s_{0:t}, m_{0:t} | o_{1:t})$ of hidden states $s_{0:t} \triangleq \{s_0, \dots, s_t\}$ and model indices $m_{0:t} \triangleq \{m_0, \dots, m_t\}$, given a collection of observations $o_{1:t} \triangleq \{o_1, \dots, o_t\}$.

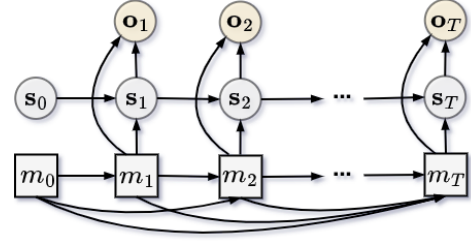


Fig. 1. Diagram of a regime switching state-space model with a period of T .

III. RELATED WORK

A. Regime-switching particle filters

The regime switching particle filter (RS-PF) was proposed in [24] for general regime switching systems (Equations (1)-(5)). The joint posterior can be factorised as [24]:

$$p(s_{0:t}, m_{0:t} | o_{1:t}) \propto p(s_{0:t-1}, m_{0:t-1} | o_{1:t-1}) p(o_t | s_t, m_t) \times p(s_t | s_{t-1}, m_t) p(m_t | m_{0:t-1}). \quad (6)$$

The unnormalised importance weight of the i -th particle $\{s_{0:t}^{(i)}, m_{0:t}^{(i)}\}$ is computed as:

$$\bar{w}_t^{(i)} = \frac{p(s_{0:t}^{(i)}, m_{0:t}^{(i)} | o_{1:t})}{q(s_{0:t}^{(i)}, m_{0:t}^{(i)} | o_{1:t})}, \quad (7)$$

where the joint proposal distribution is factorised by:

$$q(s_{0:t}, m_{0:t} | o_{1:t}) = q(s_{0:t-1}, m_{0:t-1} | o_{1:t-1}) \times q(s_t | s_{t-1}, o_t, m_t) q(m_t | m_{0:t-1}). \quad (8)$$

If bootstrap particle filters are adopted for each candidate model, i.e., $q(s_t | s_{t-1}, m_t, o_t) = p(s_t | s_{t-1}, m_t)$, the computation of the importance weight is simplified to:

$$w_t^{(i)} \propto \frac{p(m_t^{(i)} | m_{0:t-1}^{(i)}) p(o_t | s_t^{(i)}, m_t^{(i)})}{q(m_t^{(i)} | m_{0:t-1}^{(i)})} w_{t-1}^{(i)}. \quad (9)$$

Three methods were proposed to construct the regime index proposal distribution $q(m_t | m_{0:t-1})$ [24]: (i) a discrete uniform distribution, i.e. $q(m_t = j | m_{1:t-1}) = \frac{1}{N_m}$ for $j = \{1, \dots, N_m\}$; (ii) a bootstrap method, i.e. $q(m_t = j | m_{0:t-1}) = p(m_t | m_{0:t-1})$ for each j ; (iii) a deterministic method that assigns an equal number of particles to each candidate model.

B. Differentiable particle filters

Differentiable particle filters [27]–[32] apply neural networks to construct dynamic and measurement models of particle filters in a data-adaptive way, i.e., the dynamic and measurement models are learned from data using machine learning models, e.g. neural networks. The forward propagation of a differentiable bootstrap particle filter [28] at a single time step is illustrated in Algorithm 1. Functions $k_\theta(\cdot)$ and $l_\theta(\cdot)$, which are the particle proposer and the observation likelihood estimator, respectively, are parameterised by neural networks. The dynamic model constructed with $k_\theta(\cdot)$ takes an auxiliary noise vector $\epsilon^{(i)}$ as part of its input for optimisation

Algorithm 1: Forward propagation of a differentiable bootstrap particle filter at time step t

Input: $\{\mathbf{s}_{t-1}^{(i)}, w_{t-1}^{(i)}, \boldsymbol{\epsilon}^{(i)}\}_{i=1}^{N_p}, \mathbf{e}_t, k_{\boldsymbol{\theta}}, l_{\boldsymbol{\theta}}$
Output: $\{\mathbf{s}_t^{(i)}, \bar{w}_t^{(i)}\}_{i=1}^{N_p}$

- 1 Sample $\mathbf{s}_t^{(i)}$ according to the dynamic model
 $\{\mathbf{s}_t^{(i)} \sim f_{\boldsymbol{\theta}}(\mathbf{s}_{t-1}^{(i)}) = k_{\boldsymbol{\theta}}(\mathbf{s}_{t-1}^{(i)}, \boldsymbol{\epsilon}^{(i)})\}_{i=1}^{N_p}$;
 - 2 Compute observation likelihood
 $\{g_{\boldsymbol{\theta}}(\mathbf{e}_t | \mathbf{s}_t^{(i)}) = l_{\boldsymbol{\theta}}(\mathbf{e}_t, \mathbf{s}_t^{(i)})\}_{i=1}^{N_p}$;
 - 3 Evaluate importance weights $\{\bar{w}_t^{(i)} = w_{t-1}^{(i)} g_{\boldsymbol{\theta}}(\mathbf{e}_t | \mathbf{s}_t^{(i)})\}_{i=1}^{N_p}$;
-

with the reparameterisation trick. The observation \mathbf{o}_t can be encoded by a neural network $h_{\boldsymbol{\theta}}(\cdot)$ to generate a feature vector \mathbf{e}_t , i.e., $\mathbf{e}_t = h_{\boldsymbol{\theta}}(\mathbf{o}_t)$. Objective functions $\mathcal{L}(\boldsymbol{\theta})$ employed by differentiable particle filters can be classified mainly as supervised losses [27], [28], [31], [36], where the ground truth state information is available for training, and semi-supervised losses to leverage observations with unknown ground truth state information [35].

IV. REGIME-SWITCHING DIFFERENTIABLE BOOTSTRAP PARTICLE FILTERS

We now introduce the proposed regime-switching differentiable bootstrap particle filter (RS-DBPF). We show in Algorithm 2 how to integrate the regime-switching system into the design of a differentiable bootstrap particle filter. Its key steps are clarified as follows.

At the beginning of each time step, the regime index $m_t^{(i)}$ of the i -th particle is sampled from the model proposal distribution $q(m_t | m_{0:t-1})$. Three options of $q(\cdot)$ were described in the end of Section III-A. Note that the filter is run with a constant number of particles N_p although adaptive mechanisms could be readily used [37], [38]. A forward propagation of the differentiable bootstrap particle filter (Algorithm 1) is performed to sample the state component $\mathbf{s}_t^{(i)}$ with a neural network-based particle proposer $k_{\boldsymbol{\theta}_{m_t^{(i)}}}(\cdot)$ [27], [28], where $\boldsymbol{\theta}_{m_t^{(i)}}$ denotes the parameter set of the $m_t^{(i)}$ -th candidate model:

$$\mathbf{s}_t^{(i)} \sim f_{\boldsymbol{\theta}_{m_t^{(i)}}}(\mathbf{s}_{t-1}^{(i)} | \mathbf{s}_{t-1}^{(i)}) = k_{\boldsymbol{\theta}_{m_t^{(i)}}}(\mathbf{s}_{t-1}^{(i)}, \boldsymbol{\epsilon}^{(i)}). \quad (10)$$

The auxiliary noise term $\boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\mathbf{0}_{d_s}, \mathbf{I}_{d_s})$ where $\mathbf{0}_{d_s}$ denotes a d_s -dimensional zero vector and \mathbf{I}_{d_s} is a $d_s \times d_s$ identity matrix. The likelihood of the i -th particle is computed as [27], [28]:

$$p(\mathbf{o}_t | \mathbf{s}_t^{(i)}, m_t^{(i)}) = g_{\boldsymbol{\theta}_{m_t^{(i)}}}(\mathbf{o}_t | \mathbf{s}_t^{(i)}) = l_{\boldsymbol{\theta}_{m_t^{(i)}}}(\mathbf{o}_t, \mathbf{s}_t^{(i)}). \quad (11)$$

Assuming knowledge of the model switching distribution $p(m_t^{(i)} | m_{1:t-1}^{(i)})$, the unnormalised importance weight $\bar{w}_t^{(i)}$ is updated following Equation (9):

$$\bar{w}_t^{(i)} = w_{t-1}^{(i)} \frac{p(m_t^{(i)} | m_{1:t-1}^{(i)}) g_{\boldsymbol{\theta}_{m_t^{(i)}}}(\mathbf{o}_t | \mathbf{s}_t^{(i)})}{q(m_t^{(i)} | m_{1:t-1}^{(i)})}. \quad (12)$$

Finally, a resampling step is performed if the effective sample size (ESS) is smaller than a threshold [39].

Algorithm 2: Regime switching differentiable bootstrap particle filters (RS-DBPFs) framework

Input: $\eta, \pi(m_0), \mu(\mathbf{s}_0), N_p, T, \mathbf{o}_{1:T}, k_{\boldsymbol{\theta}}, l_{\boldsymbol{\theta}}, \text{ESS}_{\text{thres}}, \mathcal{L}$
1 Initialise parameter sets $\boldsymbol{\theta}_j \subseteq \boldsymbol{\theta}$ for $j = \{1, \dots, N_m\}$ and set learning rate η ;

- 2 **while** $\boldsymbol{\theta}$ has not converged **do**
 - 3 Draw regime index $\{m_0^{(i)}\}_{i=1}^{N_p} \sim \pi(m_0)$;
 - 4 Draw samples $\{\mathbf{s}_0^{(i)}\}_{i=1}^{N_p} \sim \mu(\mathbf{s}_0)$;
 - 5 Set importance weights $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$;
 - 6 **for** $t = 1, 2, \dots, T$ **do**
 - 7 Draw regime index from the model proposal distribution $\{m_t^{(i)} \sim q(m_t | m_{1:t-1}^{(i)})\}_{i=1}^{N_p}$;
 - 8 Sample $\mathbf{s}_t^{(i)}$ as in Eq. (10);
 - 9 Estimate observation likelihood $g_{\boldsymbol{\theta}_{m_t^{(i)}}}(\mathbf{o}_t | \mathbf{s}_t^{(i)})$ according to Equation (11);
 - 10 Calculate importance weights
 $\{\bar{w}_t^{(i)} = w_{t-1}^{(i)} \frac{p(m_t^{(i)} | m_{1:t-1}^{(i)}) g_{\boldsymbol{\theta}_{m_t^{(i)}}}(\mathbf{o}_t | \mathbf{s}_t^{(i)})}{q(m_t^{(i)} | m_{1:t-1}^{(i)})}\}_{i=1}^{N_p}$;
 - 11 Normalise weights $\{w_t^{(i)} = \frac{\bar{w}_t^{(i)}}{\sum_{k=1}^{N_p} \bar{w}_t^{(k)}}\}_{i=1}^{N_p}$;
 - 12 Compute the effective sample size
 $\text{ESS}_t = \frac{1}{\sum_{i=1}^{N_p} (w_t^{(i)})^2}$;
 - 13 **if** $\text{ESS}_t < \text{ESS}_{\text{thres}}$ **then**
 - 14 Resample and update $\{m_{1:t}^{(i)}, \mathbf{s}_{1:t}^{(i)}\}_{i=1}^{N_p}$ according to importance weights $w_t^{(i)}$;
 - 15 Update $\{w_t^{(i)} = \frac{1}{N_p}\}_{i=1}^{N_p}$;
 - 16 **end**
 - 17 Calculate the estimate $\hat{\mathbf{s}}_t = \sum_{i=1}^{N_p} w_t^{(i)} \mathbf{s}_t^{(i)}$;
 - 18 **end**
 - 19 Calculate the total loss $\mathcal{L}(\boldsymbol{\theta})$;
 - 20 Update parameters by gradient descent
 $\boldsymbol{\theta}_j = \boldsymbol{\theta}_j - \eta \nabla_{\boldsymbol{\theta}_j} \mathcal{L}$ for $j = \{1, \dots, N_m\}$;
 - 21 **end**
-

V. SIMULATIONS AND RESULTS

We adopt the synthetic data experiment explored in [24]. It includes a mixture of eight candidate dynamic and measurement models with small variances for each candidate model. This leads to multi-modal posterior distributions that are challenging for filtering algorithms to explore all modes².

A. Experiment setting

The j -th candidate model ($j \in \{1, \dots, 8\}$) is as follows:

$$\mathcal{M}_j : \begin{cases} s_t = a_j s_{t-1} + b_j + u_t \\ o_t = c_j \sqrt{|s_t|} + d_j + v_t \end{cases} \quad (13)$$

The number of time steps for one trajectory is $T = 50$. The initial state $s_0 \in \mathbb{R}$ is sampled from a continuous uniform distribution $\mathcal{U}[-0.5, 0.5]$ whereas the index of the initial sub-model m_0 is sampled from a discrete uniform distribution $\mathcal{U}\{1, 8\}$. Coefficients $[a_1, \dots, a_8] = [-0.1, -0.3, -0.5, -0.9, 0.1, 0.3, 0.5, 0.9]$, $[b_1, \dots, b_8] = [0, -2, 2, -4, 0, 2, -2, 4]$, $[c_1, \dots, c_8] = [a_1, \dots, a_8]$, and

²Code to reproduce experiment results is available at <https://github.com/WickhamLi/RS-DBPF>

$[d_1, \dots, d_8] = [b_1, \dots, b_8]$. The noise terms $u_t \sim \mathcal{N}(0, 0.1)$ and $v_t \sim \mathcal{N}(0, 0.1)$. The overall dataset includes 2000 trajectories (1000 for training, 500 for validation, 500 for testing).

The regime switching dynamic follows either a Markovian dynamic or a Pólya urn dynamic. In the Markovian switching system, $p(m_t|m_{0:t-1}) = p(m_t|m_{t-1})$. Following the example in [24], we set the transition probability matrix \mathbf{P} as:

$$\mathbf{P} = \begin{bmatrix} 0.80 & 0.15 & \rho & \cdots & \rho \\ \rho & 0.80 & 0.15 & \cdots & \rho \\ \vdots & & \ddots & & \vdots \\ \rho & \cdots & \rho & 0.80 & 0.15 \\ 0.15 & \rho & \cdots & \rho & 0.80 \end{bmatrix}, \quad (14)$$

where $\rho = \frac{1}{120}$. $\mathbf{P}_{j,k} \triangleq p(m_t = k|m_{t-1} = j)$.

The Pólya switching model is a more general dynamic process to describe long-term time dependencies between candidate models:

$$p(m_t = k|m_{0:t-1}) = \frac{\sum_{\tau=0}^{t-1} \mathbb{1}_{k,\tau} + \beta_k}{\sum_{j=1}^{N_m} (\sum_{\tau=0}^{t-1} \mathbb{1}_{j,\tau} + \beta_j)}, \quad (15)$$

where $k = \{1, \dots, N_m\}$ denotes the regime index, $\mathbb{1}_{k,\tau} = \mathbb{1}(m_\tau = k)$ is an indicator function on whether the system is switched to the k -th model at time step τ . β_k is set to 1.

B. Parameter values for the filtering algorithms

We compare the proposed regime-switching differentiable bootstrap particle filter (RS-DBPF) with a multi-model particle filter (MM-PF) [20], a differentiable bootstrap particle filter (DBPF) [28], and a regime-switching particle filter (RS-PF) [24]. 200 particles are employed for training and validation to reduce computational costs. 2000 particles are used to perform filtering with testing trajectories for smooth estimated trajectories. Particles are initialised from a uniform distribution $\mathcal{U}[-0.5, 0.5]$. The RS-DBPF and the RS-PF adopt a uniform distribution for the regime index proposal $q(m_t|m_{1:t-1})$ to evaluate the robustness of the filters when the regime index proposal deviates from the regime switching dynamic.

For the DBPF and the RS-DBPF, the particle proposer k_θ adopts a 2-layer neural network with 8 neurons in the hidden layer. A Gaussian kernel with a learnable kernel parameter is used to generate conditional likelihoods l_θ , by comparing the observation with an embedding generated from the state through a 2-layer neural network with 8 neurons in the hidden layer with a tanh-activation function. A supervised loss based on mean squared errors between the ground truth and predicted states is adopted. Stochastic gradient descent with a momentum factor of 0.9 is used as optimiser. We choose learning rates $\eta \in \{0.01, 0.02, 0.05, 0.1\}$ with a step-wise decay that halves the learning rate every 10 epochs for the DBPF and the RS-DBPF based on best validation performance. The epoch number is set to 60. The mini-batch size is set to 100.

C. Tracking performance

We compute root mean squared errors (RMSEs) of predicted states for each test trajectory and report error statistics among

TABLE I
AVERAGE, BEST, AND WORST RMSEs WITH A MARKOVIAN DYNAMIC.

	Average	Best	Worst
MM-PF (baseline)	1.9016	0.5601	9.4422
DBPF (baseline)	1.5176	0.5085	9.5790
RS-DBPF (proposed)	0.8325	0.3779	8.6401
RS-PF (oracle)	0.4627	0.2570	2.2972

TABLE II
AVERAGE, BEST, AND WORST RMSEs WITH A PÓLYA DYNAMIC.

	Average	Best	Worst
MM-PF (baseline)	2.1334	0.6409	5.3060
DBPF (baseline)	1.6144	0.4754	5.4350
RS-DBPF (proposed)	0.8394	0.3817	2.5627
RS-PF (oracle)	0.6399	0.3171	2.0383

500 test trajectories in TABLES I and II. The proposed RS-DBPF leads to significantly smaller average RMSEs compared with the baselines including the DBPF and the MM-PF. Fig. 2 plots the absolute errors along each time step averaged over all the test trajectories for the Pólya urn switching model. Note that both the RS-DBPF and the DBPF assume no knowledge of the candidate models, while the MM-PF and the RS-PF have access to the ground truth candidate models. The RS-PF can further utilise the regime switching dynamic so it is served as the oracle model to generate optimal filtering performance as a benchmark. The poor performance of the MM-PF is due to its algorithmic assumption that there is no regime switching.

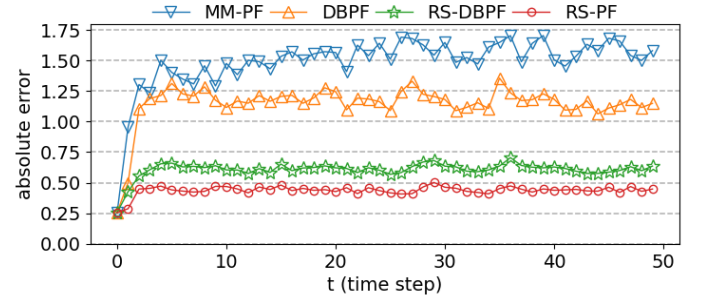


Fig. 2. Mean absolute errors at each time step (Pólya urn dynamics).

VI. CONCLUSION

In this paper, we address filtering tasks where a mixture of unknown candidate dynamic and measurement models exist. The proposed RS-DBPF can flexibly switch between candidate models, i.e. regimes, while simultaneously learn the candidate models without prior knowledge of their functional forms. Numerical simulations show that the RS-DBPF outperforms both a vanilla DBPF and a MM-PF in two simulation setups with different regime switching dynamics. Future work includes the estimation of the regime-switching dynamic, the incorporation of more expressive neural networks to construct candidate models, and more extensive experimental evaluation with high-dimensional numerical and real-world experiments.

REFERENCES

- [1] A. Doucet and A. M. Johansen, “A tutorial on particle filtering and smoothing: Fifteen years later,” *Handb. of Nonlinear Filtering*, vol. 12, no. 656-704, p. 3, 2009.
- [2] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, “Particle filtering,” *IEEE Signal Process. Mag.*, vol. 20, no. 5, pp. 19–38, 2003.
- [3] A. Doucet, N. De Freitas, N. J. Gordon *et al.*, *Sequential Monte Carlo methods in practice*. Springer, 2001, vol. 1, no. 2.
- [4] N. J. Gordon, D. J. Salmond, and A. F. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” in *IEE Proc. F (Radar Signal Process.)*, Apr. 1993, pp. 107–113.
- [5] M. K. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filters,” *J. Amer. Statist. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.
- [6] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djurić, “In search for improved auxiliary particle filters,” in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Rome, Italy, Sep. 2018, pp. 1637–1641.
- [7] V. Elvira, L. Martino, M. F. Bugallo, and P. M. Djuric, “Elucidating the auxiliary particle filter via multiple importance sampling,” *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 145–152, 2019.
- [8] N. Branchini and V. Elvira, “Optimized auxiliary particle filters: adapting mixture proposals via convex optimization,” in *Proc. Conf. Uncertain. Artif. Intell. (UAI)*, Jul. 2021, pp. 1289–1299.
- [9] J. H. Kotecha and P. M. Djuric, “Gaussian sum particle filtering,” *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2602–2612, 2003.
- [10] —, “Gaussian sum particle filtering for dynamic state-space models,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Salt Lake City, USA, May 2001.
- [11] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, “Rao-Blackwellised particle filtering for dynamic Bayesian networks,” in *Proc. Conf. Uncertain. Artif. Intell. (UAI)*, Stanford, USA, Jun. 2000, pp. 176–183.
- [12] N. De Freitas, “Rao-Blackwellised particle filtering for fault diagnosis,” in *Proc. IEEE Aerosp. Conf.*, vol. 4, Mar. 2002, pp. 4–4.
- [13] P. J. Van Leeuwen, H. R. Künsch, L. Nerger, R. Potthast, and S. Reich, “Particle filters for high-dimensional geoscience applications: A review,” *Quart. Journal of the R. Meteorological Soc.*, vol. 145, no. 723, pp. 2335–2365, 2019.
- [14] A. Gunatilake, S. Kodagoda, and K. Thiagarajan, “A novel UHF-RFID dual antenna signals combined with Gaussian process and particle filter for in-pipe robot localization,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6005–6011, 2022.
- [15] C. Pozna, R.-E. Precup, E. Horváth, and E. M. Petriu, “Hybrid particle filter–particle swarm optimization algorithm and application to fuzzy controlled servo systems,” *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 10, pp. 4286–4297, 2022.
- [16] M. H. Dupty, Y. Dong, and W. S. Lee, “PF-GNN: Differentiable particle filtering based approximation of universal graph representations,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2021.
- [17] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin, “On particle methods for parameter estimation in state-space models,” *Stat. Sci.*, vol. 30, no. 3, pp. 328–351, 2015.
- [18] S. McGinnity and G. W. Irwin, “Multiple model bootstrap filter for maneuvering target tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 3, pp. 1006–1012, 2000.
- [19] L.-Q. Li, W.-X. Xie, J.-X. Huang, and J.-J. Huang, “Multiple model rao-blackwellized particle filter for manoeuvring target tracking,” *Defence Sci. Journal*, vol. 59, no. 3, p. 197, 2009.
- [20] B. Liu, “Instantaneous frequency tracking under model uncertainty via dynamic model averaging and particle filtering,” *IEEE Trans. Wirel. Commun.*, vol. 10, no. 6, pp. 1810–1819, 2011.
- [21] I. Urteaga, M. F. Bugallo, and P. M. Djurić, “Sequential Monte Carlo methods under model uncertainty,” in *Proc. IEEE Conf. Stat. Signal Process. Workshop (SSP)*, Palma de Mallorca, Spain, Feb. 2016, pp. 1–5.
- [22] L. Martino, J. Read, V. Elvira, and F. Louzada, “Cooperative parallel particle filters for online model selection and applications to urban mobility,” *Digit. Signal Process.*, vol. 60, pp. 172–185, 2017.
- [23] H. Wang, X. Zhao, and M. F. Bugallo, “Indoor UAV height estimation with multiple model-detecting particle filters,” in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Belgrade, Serbia, Aug. 2022, pp. 977–981.
- [24] Y. El-Laham, L. Yang, P. M. Djurić, and M. F. Bugallo, “Particle filtering under general regime switching,” in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Online, May 2021, pp. 2378–2382.
- [25] P. Fearnhead, “Particle filters for mixture models with an unknown number of components,” *Stat. Comput.*, vol. 14, no. 1, pp. 11–21, 2004.
- [26] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe, “Bayesian inference for linear dynamic models with dirichlet process mixtures,” *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 71–84, 2007.
- [27] P. Karkus, D. Hsu, and W. S. Lee, “Particle filter networks with application to visual localization,” in *Proc. Conf. Robot Learn. (CoRL)*, Zürich, Switzerland, Oct. 2018, pp. 169–178.
- [28] R. Jonschkowski, D. Rastogi, and O. Brock, “Differentiable particle filters: End-to-end learning with algorithmic priors,” in *Proc. Robot. Sci. Syst.*, Pittsburgh, USA, Jun. 2018.
- [29] X. Ma, P. Karkus, D. Hsu, and W. S. Lee, “Particle filter recurrent neural networks,” in *Proc. AAAI Conf. AI (AAAI)*, New York, USA, Feb. 2020.
- [30] A. Kloss, G. Martius, and J. Bohg, “How to train your differentiable filter,” *Auton. Robots*, vol. 45, no. 4, pp. 561–578, 2021.
- [31] X. Chen, H. Wen, and Y. Li, “Differentiable particle filters through conditional normalizing flow,” in *Proc. IEEE Int. Conf. Inf. Fusion (FUSION)*, Sun City, South Africa, Nov. 2021, pp. 1–6.
- [32] A. Corenflos, J. Thornton, G. Deligiannidis, and A. Doucet, “Differentiable particle filtering via entropy-regularized optimal transport,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, July 2021.
- [33] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Banff, Canada, Apr. 2014.
- [34] D. Rezende and S. Mohamed, “Variational inference with normalizing flows,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, Lille, France, Jul. 2015, pp. 1530–1538.
- [35] H. Wen, X. Chen, G. Papagiannis, C. Hu, and Y. Li, “End-to-end semi-supervised learning for differentiable particle filters,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Xi’an, China, May 2021, pp. 5825–5831.
- [36] X. Chen and Y. Li, “Conditional measurement density estimation in sequential Monte Carlo via normalizing flow,” in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Belgrade, Serbia, Aug. 2022, pp. 782–786.
- [37] V. Elvira, J. Míguez, and P. M. Djurić, “Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment,” *IEEE Trans. Signal Process.*, vol. 65, no. 7, pp. 1781–1794, 2016.
- [38] V. Elvira, J. Míguez, and P. M. Djurić, “On the performance of particle filters with adaptive number of particles,” *Stat. Comput.*, vol. 31, pp. 1–18, 2021.
- [39] V. Elvira, L. Martino, and C. P. Robert, “Rethinking the effective sample size,” *Int. Stat. Rev.*, vol. 90, no. 3, pp. 525–550, 2022.