Gaussian Process State-Space Models with Time-Varying Parameters and Inducing Points

Yuhao Liu
Department of Applied Mathematics & Statistics
Stony Brook University
Stony Brook, USA
yuhao.liu.1@stonybrook.edu

Petar M. Djurić
Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, USA
petar.djuric@stonybrook.edu

***Abstract*—We propose time-varying Gaussian process state-space models (TVGPSSM) whose hyper-parameters vary with time. The models have the ability to estimate time-varying functions and thereby increase flexibility to extract information from observed data. The proposed inference approach makes use of time-varying inducing points to adapt to changes of the function, and it exploits hierarchical importance sampling. The experimental results show that the approach has better performance than that of the standard Gaussian process.**

***Index Terms*—System identification, Gaussian processes, state-space model, hierarchical importance sampling**

I. Introduction

Non-linear state-space models (SSMs) are of importance in science and engineering [1], [2], and they are often identified by using data-driven methods [3], [4]. These methods are able to learn patterns from data and based on them predict future observations. The models have the following form:

$$\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_{t-1}) + \boldsymbol{\varepsilon}_{x,t}, \tag{1}$$

$$\boldsymbol{y}_t = \boldsymbol{g}(\boldsymbol{x}_t) + \boldsymbol{\varepsilon}_{y,t}, \tag{2}$$

where t is a discrete time index, $\boldsymbol{x}_t \in \mathbb{R}^D$, $\boldsymbol{y}_t \in \mathbb{R}^O$, $\boldsymbol{u}_t \in \mathbb{R}^P$ denote the latent state, output and input (action) at time t, respectively, and $\boldsymbol{\varepsilon}_{(\cdot,t)} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2_{(\cdot)}\boldsymbol{I})$ are the i.i.d. Gaussian process or measurement noise. The symbols $\boldsymbol{f}(\cdot) : \mathbb{R}^{D+P} \to \mathbb{R}^D$ and $\boldsymbol{g}(\cdot) : \mathbb{R}^D \to \mathbb{R}^O$ represent transition and measurement functions, respectively. Compared with deterministic methods, such as recurrent neural networks [5], probabilistic SSMs [6]–[8] add the notion of uncertainty to ensure safe learning [9]. Probabilistic SSMs have found applications in various control systems [10], [11].

To avoid overfitting by nonlinear SSMs [12], especially for high dimensional systems with modest amount of observations, the Gaussian processes (GPs) have become a useful tool [13]. In comparison to many modern machine learning methods, their ability to process uncertainties can improve their robustness during the learning process. A GP state-space model (GPSSM) [14] models the transition function $\boldsymbol{f}$ and/or $\boldsymbol{g}$ as a GP, using a mean function $\boldsymbol{m}(\cdot)$ and a covariance function (kernel) $K(\cdot, \cdot|\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^H$ denotes the hyper-parameters encoding the prior information or assumptions such as smoothness and flatness of the modeled functions.

The existing literature focuses on models with time-invariant hyper-parameters, i.e., addressing the problem of estimating time-invariant transition functions $\boldsymbol{f}$ and measurement functions $\boldsymbol{g}$. It describes methods based on the MCMC [15], the variational inference [16], [17] or online processing [18], [19]. However, in many practical settings, the assumption of fixed parameters is too restrictive. In various domains like robotics and computational finance, the transition probabilities change slowly, with the cumulative change being significant. To overcome this limitation, in this paper we propose a time-varying GP SSM (TVGPSSM), utilizing both time varying hyper-parameters and inducing points. The varying hyper-parameters are embedded in the transition function, which is hard to optimize analytically. To tackle this challenge, we propose a hierarchical sampling method to estimate the posterior distribution of the hyper-parameters. TVGPSSMs have the ability to estimate time-varying functions while simultaneously stabilizing the learning process. The main contributions of this paper are:

- Proposal of a novel approach to estimating time-varying state-space models based on GPs.
- Adjusting the hyper-parameters of the GPs over time, and thus reflecting the varying correlation between the data points.
- Alleviating model bias by adapting for the posterior mean function by using varying inducing points.
- Allowing the proposed model to be extended to any kernel and any GPSSM using variational inference.

II. Preliminaries

A GP is in essence a distribution over functions [13]. For any set of inputs $\boldsymbol{X} = [\boldsymbol{x}_i]_{i=1}^N$ in the domain of a real-valued GP function $f \sim \mathcal{GP}(m(\cdot), K(\cdot, \cdot|\boldsymbol{\theta}))$, the function values $\boldsymbol{f} = [f(\boldsymbol{x}_i)]_{i=1}^N$ are Gaussian distributed, i.e.,

$$p(\boldsymbol{f}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{m}_{\boldsymbol{X}}, \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}), \tag{3}$$

where $\boldsymbol{m}_{\boldsymbol{X}} = [m(\boldsymbol{x}_i)]_{i=1}^N$ is the mean function and the matrix $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} = K(\boldsymbol{X}, \boldsymbol{X}|\boldsymbol{\theta})$ is the covariance function of all pairs in $\boldsymbol{X}$, and where $\boldsymbol{\theta}$ are the hyper-parameters of the GP.

This work has been supported by NIH under Award 1RO1HD097188-01.

In the framework of GPSSM, independence between the GP priors is a common model assumption [16], [17]. Thus, we apply GP priors on the transition function $\boldsymbol{f}(\cdot)$ for each output dimensions such that $f_d(\cdot) \sim \mathcal{GP}(m_d(\cdot), K(\cdot,\cdot|\boldsymbol{\theta}_d))$ is independent from each other, where $d = 1, 2, ..., D$, and $\boldsymbol{\theta}_d \in \mathbb{R}^H$. Let $\boldsymbol{f}_t = [f_d(\tilde{\boldsymbol{x}}_{t-1})]_{d=1}^D = [f_1(\tilde{\boldsymbol{x}}_{t-1}), ..., f_D(\tilde{\boldsymbol{x}}_{t-1})]^\top$, where $\tilde{\boldsymbol{x}}_t = [\boldsymbol{x}_t, \boldsymbol{u}_t]$ is the state-action input pair at time t. Then for the transition equation we can write $p(\boldsymbol{x}_t|\boldsymbol{f}_t) = \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{f}_t, \sigma_x^2\boldsymbol{I})$, where $\boldsymbol{I}$ denotes the identity matrix.

One benefit of the GP setting is that the conditional distribution of a GP is still a GP, and therefore the predictive distribution of a realisation f^* at a new point $\tilde{\boldsymbol{x}}^*$ is given by

$$p(f^*|\tilde{\boldsymbol{x}}^*, \boldsymbol{f}, \tilde{\boldsymbol{X}}) = \mathcal{N}(f^*|\mu, \sigma^2),$$

with predictive mean and variance:

$$\begin{aligned} \mu &= m_{\tilde{\boldsymbol{x}}^*} + \boldsymbol{k}_{\tilde{\boldsymbol{x}}^*,\tilde{\boldsymbol{X}}}\boldsymbol{K}_{\tilde{\boldsymbol{X}},\tilde{\boldsymbol{X}}}^{-1}(\boldsymbol{f} - \boldsymbol{m}_{\tilde{\boldsymbol{X}}}), \\ \sigma^2 &= k_{\tilde{\boldsymbol{x}}^*,\tilde{\boldsymbol{x}}^*} - \boldsymbol{k}_{\tilde{\boldsymbol{x}}^*,\tilde{\boldsymbol{X}}}\boldsymbol{K}_{\tilde{\boldsymbol{X}},\tilde{\boldsymbol{X}}}^{-1}\boldsymbol{k}_{\tilde{\boldsymbol{X}},\tilde{\boldsymbol{x}}^*}, \end{aligned} \tag{4}$$

where $\boldsymbol{k}_{A,B} = \boldsymbol{k}(A, B)$ denotes the covariance computed for all pairs of A and B and $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_i]_{i=1}^N$. This property makes the sparse GP popular because of the closed-form prediction and low burden of computations.

III. Time Varying Gaussian Process State-Space Model

A. Problem Formulation

Suppose we observe a time series $\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_T$. Let the model of the data be given by a GPSSM whose hyper-parameters and inducing variables follow random walks. Thus, the dynamic system of interest is given by

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{f}_t(\boldsymbol{x}_{t-1}, \boldsymbol{u}_{t-1}) + \boldsymbol{\varepsilon}_{x,t}, & \boldsymbol{x} &\in \mathbb{R}^D, \\ \boldsymbol{y}_t &= \boldsymbol{g}(\boldsymbol{x}_t) + \boldsymbol{\varepsilon}_{y,t}, & \boldsymbol{y} &\in \mathbb{R}^O, \\ \boldsymbol{\theta}_{t,d} &= \boldsymbol{\theta}_{t-1,d} + \varepsilon_{\theta_d,t}, & \boldsymbol{\theta}_d &\in \mathbb{R}^H, \\ \boldsymbol{z}_{t,d} &= \boldsymbol{z}_{t-1,d} + \boldsymbol{\varepsilon}_{z_d,t}, & \boldsymbol{z}_d &\in \mathbb{R}^M, \end{aligned} \tag{5}$$

where the $\boldsymbol{\varepsilon}_{(\cdot,t)}$s represent independent i.i.d. Gaussian noises, $\boldsymbol{f}_t(\cdot) \sim \mathcal{GP}(\boldsymbol{m}_t(\cdot), \boldsymbol{K}(\cdot,\cdot|\boldsymbol{\theta}_t))$, $\boldsymbol{\theta}_t$ is a hyper-parameter of the kernel, and $\boldsymbol{z}_t$ is an inducing variable, both defined at time t. To avoid the unidentifiability problem [12], a known or parametric measurement mapping $\boldsymbol{g}$ is assumed. This is reasonable because the unknown or non-parametric part can be absorbed or explained by the non-parametric GP function and the additional state dimensions [20]. Our goal is to learn the transition function by tracking the time-varying hyper-parameters. In this work, we consider the radial basis function (RBF) kernel, and we set $\boldsymbol{m}_t(\tilde{\boldsymbol{x}}) \equiv \boldsymbol{m}(\tilde{\boldsymbol{x}}) = \boldsymbol{x}$ or $\boldsymbol{0}$.

We make the same assumptions as in Section II, that is, the GP priors for each dimension of the transition function $f_{t,d}(\cdot) \sim \mathcal{GP}(m_{t,d}(\cdot), K(\cdot,\cdot|\boldsymbol{\theta}_{t,d}))$ are independent from each other, where $d = 1, 2, ..., D$, and $\boldsymbol{\theta}_{t,d} \in \mathbb{R}^H$. The realization of the transition function at time t is $\boldsymbol{f}_t = [f_{t,d}(\tilde{\boldsymbol{x}}_{t-1})]_{d=1}^D = [f_{t,1}(\tilde{\boldsymbol{x}}_{t-1}), ..., f_{t,D}(\tilde{\boldsymbol{x}}_{t-1})]^T$.

B. Variational Sparse Gaussian Process

The idea behind sparse GPs is approximating the posterior by using a small set of points, and then, based on it, derive the desired posterior [21], [22]. Let the pseudo inducing inputs for each dimension be fixed points in the same domain as $\tilde{\boldsymbol{x}}$ and denoted by $\boldsymbol{\eta}_d = [\eta_{d,1}, ..., \eta_{d,M}]$. The corresponding pseudo inducing outputs or inducing variables at time 0 have a prior $\boldsymbol{z}_{0,d} = [f_{0,d}(\eta_{d,m})]_{m=1}^M \sim \mathcal{N}(m_d(\boldsymbol{\eta}_d), \boldsymbol{K}_{\boldsymbol{\eta}_d,\boldsymbol{\eta}_d})$, where $m_d(\boldsymbol{\eta}_d) = [m_d(\eta_{d,m})]_{m=1}^M$. It is not difficult to show that the GP prior of $\boldsymbol{z}_t$ is $\mathcal{GP}(\boldsymbol{m}(\boldsymbol{\eta}), K(\boldsymbol{\eta},\boldsymbol{\eta}|\boldsymbol{\theta}_t) + t\sigma_{(z)}^2\boldsymbol{I})$ since $\boldsymbol{z}_t$ follows a random walk. We employ mean-field approximation to the posterior of $\boldsymbol{z}_0$, and find that the distribution of the inducing variable is $q(\boldsymbol{z}_0) = \prod_{d=1}^D \mathcal{N}(\boldsymbol{z}_{0,d}|\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d)$, where $\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d$ are unknown and need to be trained. The posterior of $\boldsymbol{z}_t$ is $q(\boldsymbol{z}_t) = \prod_{d=1}^D \mathcal{N}(\boldsymbol{z}_{t,d}|\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d + t\sigma_{(z)}^2\boldsymbol{I})$. On the basis of equation (4) and $q(\boldsymbol{z}_t)$, the posterior of the transition function at time t has the form of $f_{t,q}(\cdot) = \int f_t(\cdot|\boldsymbol{z}_t, \boldsymbol{\theta}_t)q(\boldsymbol{z}_t)d\boldsymbol{z}_t \sim \mathcal{GP}(m_{t,q}(\cdot), K_{t,q}(\cdot,\cdot|\boldsymbol{\theta}_t))$, where

$$\begin{aligned} m_{t,q,d}(\cdot) &= m_{t,d}(\cdot) + K_t(\cdot,\boldsymbol{\eta}_d)K_{t,\boldsymbol{\eta}_d,\boldsymbol{\eta}_d}^{-1}(\boldsymbol{\mu}_d - \boldsymbol{\eta}_d), \\ K_{t,q,d}(\cdot,\cdot) &= K_t(\cdot,\cdot) - \alpha_t(\cdot)[K_{t,\boldsymbol{\eta}_d,\boldsymbol{\eta}_d} - \boldsymbol{\Sigma}_d - t\sigma_{(z)}^2\boldsymbol{I}]\alpha_t(\cdot)^T, \end{aligned} \tag{6}$$

where $\alpha_t(\cdot) = K_t(\cdot,\boldsymbol{\eta}_d)K_{t,\boldsymbol{\eta}_d,\boldsymbol{\eta}_d}^{-1}$, $K_t(\cdot,\cdot) = K(\cdot,\cdot|\boldsymbol{\theta}_t)$, and $K_{t,\boldsymbol{\eta}_d,\boldsymbol{\eta}_d} = K(\boldsymbol{\eta}_d,\boldsymbol{\eta}_d|\boldsymbol{\theta}_t)$. This leads naturally to our predictive distribution of $\boldsymbol{x}^*$ to be $f_{t,q}(\boldsymbol{x}^*)$.

C. Variational Inference

From (5), we write the full joint distribution of $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}$,

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}) &= p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{\theta})p(\boldsymbol{z})p(\boldsymbol{\theta}) \\ &\quad \times p(\boldsymbol{x}_0)p(\boldsymbol{z}_0)p(\boldsymbol{\theta}_0), \end{aligned} \tag{7}$$

where $p(\boldsymbol{z})$ and $p(\boldsymbol{\theta})$ represent the joint distribution of $\boldsymbol{z}_{1:T}$ and $\boldsymbol{\theta}_{1:T}$, respectively.

The core idea of variational inference [23] is finding a variational posterior distribution $q(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta})$ to approximate the true posterior $p(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{y})$. Then the evidence lower bound (ELBO) is represented by

$$ELBO := \int_{q(\boldsymbol{x},\boldsymbol{f},\boldsymbol{z},\boldsymbol{\theta})} \log \frac{p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta})}{q(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta})} \leq \log p(\boldsymbol{y}). \tag{8}$$

Different choices of variational posterior create different models with different ELBO. Our TVGPSSM can accommodate any choice of $q(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta})$. We select the variational posterior distribution that was used for the probabilistic recursive SSM [17] and extend it to a time-varying version, which with some abuse of notation is given by

$$q(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{z}, \boldsymbol{\theta})p(\boldsymbol{z})p(\boldsymbol{\theta})q(\boldsymbol{x}_0)q(\boldsymbol{z}_0)q(\boldsymbol{\theta}_0), \tag{9}$$

where $q(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0|\boldsymbol{u}_{x_0}, \boldsymbol{\Sigma}_{x_0})$ and $q(\boldsymbol{\theta}_0) = \mathcal{N}(\boldsymbol{\theta}_0|\boldsymbol{u}_{\theta_0}, \boldsymbol{\Sigma}_{\theta_0})$ are to be learnt. Thus, the ELBO is transformed to:

$$\begin{aligned} ELBO &= \int_{q(\boldsymbol{x},\boldsymbol{f},\boldsymbol{z},\boldsymbol{\theta})} \log \frac{p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta})}{q(\boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}, \boldsymbol{\theta})} \\ &= \sum_{t=1}^T \mathbb{E}_{q(\boldsymbol{x}_t)}[\log p(\boldsymbol{y}_t|\boldsymbol{x}_t)] - KL(q(\boldsymbol{x}_0)||p(\boldsymbol{x}_0)) \\ &\quad - KL(q(\boldsymbol{z}_0)||p(\boldsymbol{z}_0)) - KL(q(\boldsymbol{\theta}_0)||p(\boldsymbol{\theta}_0)). \end{aligned} \tag{10}$$

The derivation of ELBO is provided in the appendix. We note that the current parameter space is $\{\sigma_x^2, \sigma_\theta^2, \sigma_z^2, \boldsymbol{\mu}_{\theta_0}, \boldsymbol{\Sigma}_{\theta_0}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\mu}_{x_0}, \boldsymbol{\Sigma}_{x_0}\}$ with size of $O(D(M + H + 1))$, which is independent of T. Once the ELBO is derived, we maximize it to learn the hyper-parameters. The KL divergence terms can easily be solved analytically because the involved distributions are normal. The new term $KL(q(\boldsymbol{\theta}_0)||p(\boldsymbol{\theta}_0))$ in comparison to the standard GPSSM, represents the distance between the posterior and the prior of the hyper-parameters. This enables safe learning of $\boldsymbol{\theta}$. However, the expectation w.r.t $q(\boldsymbol{x}_t)$ needs to be handled by a sampling method due to the intractability of $q(\boldsymbol{x}_t)$. Under the standard GPSSM, a reparameterization trick is applied [24]. The procedure amounts to first drawing samples $\varepsilon \sim \mathcal{N}(0, 1)$ and then using the recursive update of the sample states

$$\overline{x}_{t+1,d} = \mu_d(\tilde{\boldsymbol{x}}_t) + \varepsilon\sqrt{\sigma_d^2(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{x}}_t) + \sigma_{x,d}^2}, \tag{11}$$

where $\tilde{\boldsymbol{x}}_t = (\overline{\boldsymbol{x}}_t, \boldsymbol{u}_t)$ and $\overline{\boldsymbol{x}}_0 \sim q(\boldsymbol{x}_0)$. According to (11), the unbiased estimator for the expectation term is represented by

$$\mathbb{E}_{q(\boldsymbol{x}_t)}[\log(\boldsymbol{y}_t|\boldsymbol{x}_t)] \approx \frac{1}{N}\sum_{i=1}^{N} \log p(\boldsymbol{y}_t|\overline{\boldsymbol{x}}_t^{(i)}). \tag{12}$$

However, under the TVGPSSM setting, $\mu_d(\tilde{\boldsymbol{x}}_t)$ and $\sigma_d^2(\tilde{\boldsymbol{x}}_t, \tilde{\boldsymbol{x}}_t)$ depend on the varying $\boldsymbol{\theta}_t$, and this invalidates the standard reparameterization trick. In order to address this challenge, in the next subsection we propose a hierarchical importance sampling method to generate $\boldsymbol{\theta}_t$ and $\boldsymbol{x}_t$ for estimating their respective means.

D. Hierarchical Importance Sampling Method

Similar to the reparameterization trick, the objective is to sample and propagate $\boldsymbol{x}_t^{(n)}$ forward based on (6) so that $\mathbb{E}_{q(\boldsymbol{x}_t)}[\log p(\boldsymbol{y}_t|\boldsymbol{x}_t)] \approx \frac{1}{N}\sum_{n=1}^{N} \log p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(n)})$. Then, the gradient can be derived backwards using the stochastic-gradient-descent based optimization approach. The proposed sampling processes are shown in Algorithm 1 and Figure 1.

In short, one sample cycle is described as follows:

- sample $\boldsymbol{\theta}_t^{(i)}$ from $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(i)})$ (which is defined by the samples $\boldsymbol{\theta}_{t-1}^{(i)}$ and their associated weights $w(\boldsymbol{\theta}_{t-1}^{(i)})$,
- sample $\boldsymbol{x}_t^{(i,j)}$ from $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{(i,j)}, \boldsymbol{\theta}_t^{(i)})$, where the $\boldsymbol{x}_{t-1}^{(i,j)}$s have weights $w(\boldsymbol{x}_{t-1}^{(i,j)})$,
- assign weights to $\boldsymbol{x}_t^{(i,j)}$ by $w(\boldsymbol{x}_t^{(i,j)}) = p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i,j)})$,
- assign weights to $\boldsymbol{\theta}_t^{(i)}$ by $w(\boldsymbol{\theta}_t^{(i)}) = p(\boldsymbol{y}_t|\boldsymbol{\theta}_t^{(i)})$,

where $i = 1, ..., N_1, j = 1, ..., N_2$ with N_1 being the number of samples of $\boldsymbol{\theta}_t$ and N_2, the number of samples of $\boldsymbol{x}_t$ attached to each $\boldsymbol{\theta}_t^{(i)}$. The proposal $q(\boldsymbol{x}_t^{(i,j)}|\boldsymbol{x}_{t-1}^{(i,j)}, \boldsymbol{\theta}_t^{(i)}) = f_{t,q}(\boldsymbol{x}_{t-1}^{(i,j)}|\boldsymbol{\theta}_t^{(i)})$ is conditioned on the newly sampled $\boldsymbol{\theta}_t^{(i)}$ and is calculated from equation (6) and with weight $p(\boldsymbol{y}_t|\boldsymbol{\theta}_t^{(i)}) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{\theta}_t^{(i)})d\boldsymbol{x}_t \approx \frac{1}{N_2}\sum_{j=1}^{N_2} p(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i,j)})$. In our algorithm, $\boldsymbol{\theta}_t^{(i)}$ and $\boldsymbol{x}_t^{(i,j)}$ are considered as a single sample, i.e., $\boldsymbol{x}_t^{(i,j)}$ and $\boldsymbol{\theta}_t^{(i)}$ are sampled jointly in the first step.

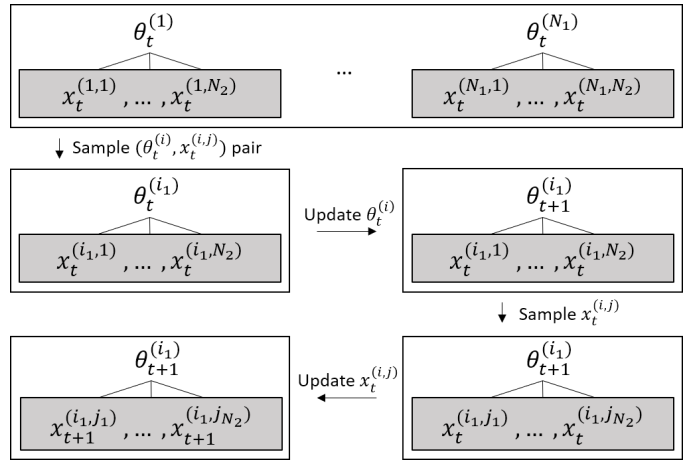

Fig. 1. A graphical model of the hierarchical importance sampling. The white and gray square nodes stand for sampled $(\boldsymbol{\theta}, \boldsymbol{x})$. One value of the hyper-parameters is associated with N_2 different values of the state. All sampling procedures are with replacement. $i_1, ..., i_{N_1}, j_1, ..., j_{N_2}$ are the indices indicating which hyper-parameters and states are sampled from the last step.

Algorithm 1: Hierarchical importance sampling

for $i = 1$ *to* N_1 **do**
  Sample $\boldsymbol{\theta}_0^{(i)} \sim p(\boldsymbol{\theta}_0)$;
  Initialize the weight $w(\boldsymbol{\theta}_0^{(i)}) = 1/N_1$;
for $i = 1$ *to* N_1 **do**
  for $j = 1$ *to* N_2 **do**
    Sample $\boldsymbol{x}_0^{(i,j)} \sim p(\boldsymbol{x}_0)$;
    Initialize the weight $w(\boldsymbol{x}_0^{(i,j)}) = 1/N_2$;
for $t = 0$ *to* $T - 1$ **do**
  for $i = 1$ *to* N_1 **do**
    Sample $\boldsymbol{\theta}_{t+1}^{(i)}$ by importance sampling:
    $$\mathbb{P}(a_t^{(i)} = k) \propto w(\boldsymbol{\theta}_t^{(k)}), k = 1, ..., N_1;$$
    $$\boldsymbol{\theta}_{t+1}^{(i)} \sim \mathcal{N}(\boldsymbol{\theta}_t^{(a_t^{(i)})}, \sigma_\theta^2 I);$$
    where $a_t^{(i)} = k$ means that $\boldsymbol{\theta}_t^{(k)}$ is chosen.
    Sample $\boldsymbol{x}_{t+1}^{(i,j)}$ by importance sampling:
    $$\mathbb{P}(b_t^{(i,j)} = l) \propto w(\boldsymbol{x}_t^{(i,l)}), l = 1, ..., N_2;$$
    $$\boldsymbol{x}_{t+1}^{(i,j)} \sim f_{t,q}(\boldsymbol{x}_t^{(i,b_t^{(i,j)})}) + \mathcal{N}(0, \sigma_x^2 I);$$
    where $b_t^{(i,j)} = l$ means that $\boldsymbol{x}_t^{(i,l)}$ is chosen.
    Update the weight
    $w(\boldsymbol{x}_{t+1}^{(i,j)}) \propto p(\boldsymbol{y}_{t+1}|g(\boldsymbol{x}_{t+1}^{(i,j)}), \sigma_y^2 I)$;
    Update the weight $w(\boldsymbol{\theta}_{t+1}^{(i)}) \propto \Sigma_{j=1}^{N_2} w(\boldsymbol{x}_{t+1}^{(i,j)})$;

TABLE I
TRAINING RMSES UNDER DIFFERENT NUMBER OF INDUCING POINTS.

RMSE (Standard Deviation)	Number of Inducing Points		
	20	*50*	*80*
PRSSM	1.4158 (0.0166)	1.3795 (0.0090)	1.3725 (0.0087)
TVGPSSM	1.2631 (0.0184)	1.1906 (0.0104)	1.1754 (0.0085)

IV. NUMERICAL EVALUATION

A. Example 1

First we consider a system with a time-invariant transition function given by

$$\begin{aligned} x_t &= \frac{\sin(7x_{t-1})+1}{2} + \varepsilon_{x,t}, \quad & \varepsilon_{x,t} \sim \mathcal{N}(0, 0.01), \\ y_t &= x_t + \varepsilon_{y,t}, & \varepsilon_{y,t} \sim \mathcal{N}(0, 0.01), \end{aligned} \tag{13}$$

where we need to learn $\{\sigma_x^2, \sigma_\theta^2, \sigma_z^2, \boldsymbol{\mu}_{\theta_0}, \boldsymbol{\Sigma}_{\theta_0}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\mu}_{x_0}, \boldsymbol{\Sigma}_{x_0}\}$. We used $p(\boldsymbol{\theta}_0) \sim \mathcal{N}((10, 0.5)^\top, 0.1\boldsymbol{I})$ and $p(\boldsymbol{x}_0) \sim \mathcal{N}(0, 1)$, where the first dimension of $\boldsymbol{\theta}_0$ is the length scale and the second is the variance. The number of pseudo inputs was set to $M = 50$ and 100, and $T = 500$. Further, only one particle $N_1 = N_2 = 1$ was used. From Fig. 2, under the standard GPSSM, the root mean square error (RMSE) could not converge as the number of iterations increased irrespectively of the number of used inducing points, while the RMSE under the proposed model was always stable. This is because the standard GPSSM has to find the optimal parameters and inducing points for the global time series. The range of the hidden states and the observations were limited to the interval between 0 to 1, according to (13) and our initialization. However, the first 30 estimated states by the standard GPSSM was up to a highly inflated value of 3,000 as shown in Fig. 3.

B. Example 2

In this example, a system with a time-varying transition function was to be learned. The model is given by

$$\begin{aligned} x_t &= (1 - \frac{t}{2T})x_{t-1} + \varepsilon_{x,t}, \quad & \varepsilon_{x,t} \sim \mathcal{N}(0, 0.1), \\ y_t &= x_t + \varepsilon_{y,t}, & \varepsilon_{y,t} \sim \mathcal{N}(0, 1), \end{aligned} \tag{14}$$

where we need to learn the transition function f. We set $p(\boldsymbol{\theta}_0) \sim \mathcal{N}((10, 5)^\top, 10^{-3}\boldsymbol{I})$, $p(\boldsymbol{x}_0) \sim \mathcal{N}(0, 1)$. We worked with $M = 20, 50, 80$ pseudo inputs and $T = 500$ data points, and set only one particle $N_1 = N_2 = 1$. Table I shows the results which suggest that the training RMSEs of our time-varying model are less than the RMSEs of the standard GPSSM for all chosen number of inducing points. Fig. 4. shows the change of hyper-parameters with time when $M = 80$. From the setting of the transition function, the true correlation of the time series should be decreasing with time, which is in accordance with Fig. 4.

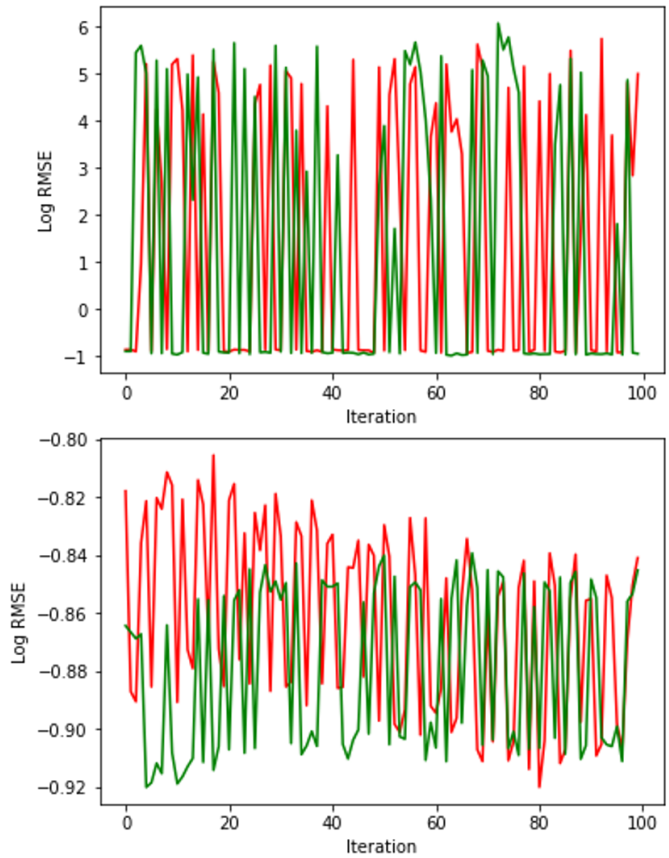

Fig. 2. The change of log RMSE in 100 iterations when $M = 50$ (red line) and $M = 100$ (green line). The upper figure is for the standard GPSSM, while the lower one is for TVGPSSM.

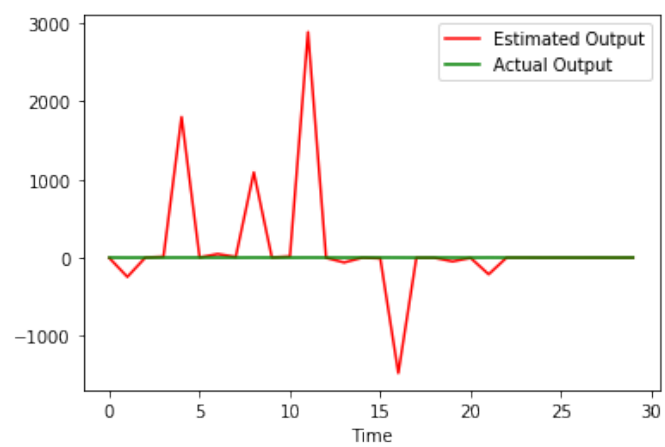

Fig. 3. The first 30 estimated states by the standard model (red line) and actual states (green line).

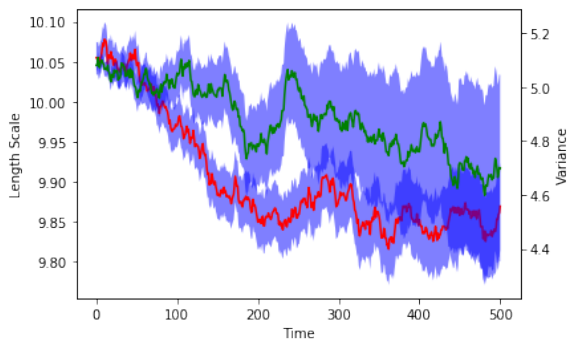

Fig. 4. Trajectory of hyper-parameters varying with time. In RBF kernel, green line represents variance σ_f^2 and red line shows the length scale l.

V. Conclusion

By setting the time-varying hyper-parameters and inducing variables, the GPSSM will not only fit time-varying functions but also offer an alternative approach to learning system. In this paper, we follow a complete Bayesian framework including the estimation of time-varying hyper-parameters of GPs used in state-space models. In combination with variational inference and importance sampling, our proposed method can accommodate any kernel and can enable online GPSSM by assuming time-varying parameters and/or inducing points. Futher, the learned trajectory of the time-varying hyper-parameters can also be treated as a GP.

References

[1] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. John Wiley & Sons, 2013.

[2] J. Kocijan, *Modelling and Control of Dynamic Systems Using Gaussian Process Models*. Springer, 2016.

[3] L. Lennart, "System identification: theory for the user," *PTR Prentice Hall, Upper Saddle River, NJ*, pp. 1–14, 1999.

[4] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, *Nonlinear Black-Box Modeling in System Identification: A Unified Overview*. Linköping University, 1995.

[5] S. Hochreiter and J. Schmidhuber, "LSTM can solve hard long time lag problems," in *Advances in Neural Information Processing Systems*, 1997, pp. 473–479.

[6] J. Ko and D. Fox, "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models," *Autonomous Robots*, vol. 27, no. 1, pp. 75–90, 2009.

[7] R. Turner, M. Deisenroth, and C. Rasmussen, "State-space inference and learning with Gaussian processes," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 868–875.

[8] R. Frigola, F. Lindsten, T. B. Schön, and C. E. Rasmussen, "Bayesian inference and learning in Gaussian process state-space models with particle MCMC," in *Advances in Neural Information Processing Systems*, 2013, pp. 3156–3164.

[9] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 465–472.

[10] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 408–423, 2013.

[11] R. McAllister and C. E. Rasmussen, "Data-efficient reinforcement learning in continuous state-action Gaussian-POMDPs," in *Advances in Neural Information Processing Systems*, 2017, pp. 2040–2049.

[12] R. Frigola, Y. Chen, and C. E. Rasmussen, "Variational Gaussian process state-space models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3680–3688.

[13] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*. MIT Press Cambridge, MA, 2006, vol. 2, no. 3.

[14] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 283–298, 2007.

[15] A. Svensson and T. B. Schön, "A flexible state–space model for learning nonlinear dynamical systems," *Automatica*, vol. 80, pp. 189–199, 2017.

[16] S. Eleftheriadis, T. Nicholson, M. Deisenroth, and J. Hensman, "Identification of Gaussian process state space models," in *Advances in Neural Information Processing Systems*, 2017, pp. 5309–5319.

[17] A. Doerr, C. Daniel, M. Schiegg, D. Nguyen-Tuong, S. Schaal, M. Toussaint, and S. Trimpe, "Probabilistic recurrent state-space models," *arXiv preprint arXiv:1801.10395*, 2018.

[18] Y. Zhao, J. Nassar, I. Jordan, M. Bugallo, and I. M. Park, "Streaming variational monte carlo," *arXiv preprint arXiv:1906.01549*, 2019.

[19] K. Berntorp, "Recursive Bayesian inference and learning of Gaussian-process state-space models," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 1866–1871.

[20] R. Frigola, "Bayesian time series learning with Gaussian processes," Ph.D. dissertation, University of Cambridge, 2015.

[21] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Artificial Intelligence and Statistics*, 2009, pp. 567–574.

[22] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems*, 2006, pp. 1257–1264.

[23] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[24] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

Appendix

A. Derivation of Evidence Lower Bound (ELBO)

1) Full Joint Distribution of TVGPSSM:

$$p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{z})$$
$$= [\prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{f}_t)p(\boldsymbol{f}_t|\boldsymbol{x}_{t-1},\boldsymbol{\theta}_t,\boldsymbol{z}_t))p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})]$$
$$p(\boldsymbol{x}_0)p(\boldsymbol{\theta}_0)p(\boldsymbol{z}_0),$$

where $p(\boldsymbol{x}_0), p(\boldsymbol{\theta}_0)$ are prior distribution we manually set.

2) Posterior Variational Approximation:

$$p(\boldsymbol{x},\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{z}|\boldsymbol{y}) \approx q(\boldsymbol{x},\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{z})$$
$$= [\prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{f}_t)p(\boldsymbol{f}_t|\boldsymbol{x}_{t-1},\boldsymbol{\theta}_t,\boldsymbol{z}_t))p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})]$$
$$q(\boldsymbol{x}_0)q(\boldsymbol{\theta}_0)q(\boldsymbol{z}_0),$$

where $q(\boldsymbol{x}_0), q(\boldsymbol{\theta}_0), q(\boldsymbol{z}_0)$ are posterior distributions.

3) ELBO Derivation:

$$\begin{aligned} ELBO &= \int_{q(\boldsymbol{x},\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{z})} \log \frac{p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{z})}{q(\boldsymbol{x},\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{z})} \\ &= \int_{q(\boldsymbol{x},\boldsymbol{f},\boldsymbol{\theta},\boldsymbol{z})} \log \frac{[\prod_{t=1}^{T} p(\boldsymbol{y}_T|\boldsymbol{x}_T)]p(\boldsymbol{x}_0)p(\boldsymbol{\theta}_0)p(z_0)}{q(\boldsymbol{x}_0)q(\boldsymbol{\theta}_0)q(\boldsymbol{z}_0)} \\ &= \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{x}_t)}[\log p(\boldsymbol{y}_t|\boldsymbol{x}_t)] - KL(q(\boldsymbol{x}_0)||p(\boldsymbol{x}_0)) \\ &\quad - KL(q(\boldsymbol{\theta}_0)||p(\boldsymbol{\theta}_0)) - KL(q(\boldsymbol{z}_0)||p(\boldsymbol{z}_0)). \end{aligned}$$