

Identification of Gaussian Process State-Space Models with Particle Stochastic Approximation EM

Roger Frigola* Fredrik Lindsten** Thomas B. Schön***
Carl E. Rasmussen*

* *Dept. of Engineering, University of Cambridge, United Kingdom.*

** *Div. of Automatic Control, Linköping University, Sweden.*

*** *Dept. of Information Technology, Uppsala University, Sweden.*

Abstract: Gaussian process state-space models (GP-SSMs) are a very flexible family of models of nonlinear dynamical systems. They comprise a Bayesian nonparametric representation of the dynamics of the system and additional (hyper-)parameters governing the properties of this nonparametric representation. The Bayesian formalism enables systematic reasoning about the uncertainty in the system dynamics. We present an approach to maximum likelihood identification of the parameters in GP-SSMs, while retaining the full nonparametric description of the dynamics. The method is based on a stochastic approximation version of the EM algorithm that employs recent developments in particle Markov chain Monte Carlo for efficient identification.

Keywords: System identification, Bayesian, Non-parametric identification, Gaussian processes

1. INTRODUCTION

Inspired by recent developments in robotics and machine learning, we aim at constructing models of nonlinear dynamical systems capable of quantifying the uncertainty in their predictions. To do so, we use the Bayesian system identification formalism whereby degrees of uncertainty and belief are represented using probability distributions [Peterka, 1981]. Our goal is to identify models that provide error-bars to any prediction. If the data is informative and the system identification unambiguous, the model will report high confidence and error-bars will be narrow. On the other hand, if predictions are made in operating regimes that were not present in the data used for system identification, we expect the error-bars to be larger.

Nonlinear state-space models are a very general and widely used class of dynamical system models. They allow for modeling of systems based on observed input-output data through the use of a latent (unobserved) variable, the *state* $\mathbf{x}_t \in \mathcal{X} \triangleq \mathbb{R}^{n_x}$. A discrete-time state-space model (SSM) can be described by

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{v}_t, \quad (1a)$$

$$\mathbf{y}_t = g(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{e}_t, \quad (1b)$$

where \mathbf{y}_t represents the output signal, \mathbf{u}_t is the input signal, and \mathbf{v}_t and \mathbf{e}_t denote i.i.d. noises. The state transition dynamics is described by the nonlinear function f whereas g links the output data at a given time to the latent state and input at that same time. For convenience, in the following we will not explicitly represent the inputs

in our formulation. When available, inputs can be straightforwardly added as additional arguments to the functions f and g .

A common approach to system identification with nonlinear state-space models consists in defining a parametric form for the functions f and g and finding the value of the parameters that minimizes a cost function, e.g. the negative likelihood. Those parametric functions are typically based on detailed prior knowledge about the system, such as the equations of motion of an aircraft, or belong to a class of parameterized generic function approximators, e.g. artificial neural networks (ANNs). In the following, it will be assumed that no detailed prior knowledge of the system is available to create a parametric model that adequately captures the complexity of the dynamical system. As a consequence, we will turn to generic function approximators. Parametrized nonlinear functions such as radial basis functions or other ANNs suffer from both theoretical and practical problems. For instance, a practitioner needs to select a parametric structure for the model, such as the number of layers and the number of neurons per layer in a neural network, which are difficult to choose when little is known about the system at hand. On a theoretical level, fixing the number of parameters effectively bounds the complexity of the functions that can be fitted to the data [Ghahramani, 2012]. In order to palliate those problems, we will use Gaussian processes [Rasmussen and Williams, 2006] which provide a practical framework for Bayesian nonparametric nonlinear system identification.

Gaussian processes (GPs) can be used to identify nonlinear state-space models by placing GP priors on the unknown functions. This gives rise to the Gaussian Process State-Space Model (GP-SSM) [Turner et al., 2010, Frigola et al.,

* Supported by the Engineering and Physical Sciences Research Council (UK) and the project *Probabilistic modeling of dynamical systems* (Contract number: 621-2013-5524) funded by the Swedish Research Council.

2013] which will be introduced in Section 2. The GP-SSM is a nonparametric model, though, the GP is in general governed by a (typically) small number of hyper-parameters, effectively rendering the model semiparametric. In this work, the hyper-parameters of the model will be estimated by maximum likelihood, while retaining the full nonparametric richness of the system model. This is accomplished by analytically marginalizing out the nonparametric part of the model and using the particle stochastic approximation EM (PSAEM) algorithm by Lindsten [2013] for estimating the parameters.

Prior work on GP-SSMs includes [Turner et al., 2010], which presented an approach to maximum likelihood estimation in GP-SSMs based on analytical approximations and the parameterization of GPs with a pseudo data set. Ko and Fox [2011] proposed an algorithm to learn (i.e. identify) GP-SSMs based on observed data which also used weak labels of the unobserved state trajectory. Frigola et al. [2013] proposed the use of particle Markov chain Monte Carlo to provide a fully Bayesian solution to the identification of GP-SSMs that did not need a pseudo data set or weak labels about unobserved states. However, the fully Bayesian solution requires priors on the model parameters which are unnecessary when seeking a maximum likelihood solution. Approaches for filtering and smoothing using already identified GP-SSMs have also been developed [Deisenroth et al., 2012, Deisenroth and Mohamed, 2012].

2. GAUSSIAN PROCESS STATE-SPACE MODELS

2.1 Gaussian Processes

Whenever there is an unknown function, GPs allow us to perform Bayesian inference directly in the space of functions rather than having to define a parameterized family of functions and perform inference in its parameter space. GPs can be used as priors over functions that encode vague assumptions such as smoothness or stationarity. Those assumptions are often less restrictive than postulating a parametric family of functions.

Formally, a GP is defined as a collection of random variables, any finite number of which have a joint Gaussian distribution. A GP $f(\mathbf{x}) \in \mathbb{R}$ can be written as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (2)$$

where the mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ are defined as

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \quad (3a)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]. \quad (3b)$$

A finite number of variables from a Gaussian process follow a jointly Gaussian distribution

$$\begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} m(\mathbf{x}_1) \\ m(\mathbf{x}_2) \\ \vdots \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) \\ \ddots & \ddots & \ddots \end{bmatrix} \right). \quad (4)$$

We refer the reader to [Rasmussen and Williams, 2006] for a thorough exposition of GPs.

2.2 Gaussian Process State-Space Models

In this article we will focus on problems where there is very little information about the nature of the state transition function $f(\mathbf{x}_t)$ and a GP is used to model it. However, we will consider that more information is available about $g(\mathbf{x}_t)$ and hence it will be modeled by a parametric function. This is reasonable in many cases where the mapping from states to observations is known, at least up to some parameters.

The generative probabilistic model for the GP-SSM is fully specified by

$$\mathbf{f}_{t+1} | \mathbf{x}_t \sim \mathcal{GP}(m_{\theta_{\mathbf{x}}}(\mathbf{x}_t), k_{\theta_{\mathbf{x}}}(\mathbf{x}_t, \mathbf{x}'_t)), \quad (5a)$$

$$\mathbf{x}_{t+1} | \mathbf{f}_{t+1} \sim \mathcal{N}(\mathbf{x}_{t+1} | \mathbf{f}_{t+1}, \mathbf{Q}), \quad (5b)$$

$$\mathbf{y}_t | \mathbf{x}_t \sim p(\mathbf{y}_t | \mathbf{x}_t, \theta_{\mathbf{y}}), \quad (5c)$$

where $\mathbf{f}_{t+1} = f(\mathbf{x}_t)$ is the value taken by the state \mathbf{x}_{t+1} after passing through the transition function, but before the application of process noise \mathbf{v}_{t+1} . The Gaussian process in (5a) describes the prior distribution over the transition function. The GP is fully specified by its mean function $m_{\theta_{\mathbf{x}}}(\mathbf{x})$ and its covariance function $k_{\theta_{\mathbf{x}}}(\mathbf{x}_t, \mathbf{x}'_t)$, which are parameterized by the vector of hyper-parameters $\theta_{\mathbf{x}}$. Equation (5b) describes the addition of process noise following a zero-mean Gaussian distribution of covariance \mathbf{Q} . We will place no restrictions on the likelihood distribution (5c) which will be parameterized by a finite-dimensional vector $\theta_{\mathbf{y}}$. For notational convenience we group all the (hyper-)parameters into a single vector $\theta = \{\theta_{\mathbf{x}}, \theta_{\mathbf{y}}, \mathbf{Q}\}$.

3. MAXIMUM LIKELIHOOD IN THE GP-SSM

Maximum likelihood (ML) is a widely used frequentist estimator of the parameters in a statistical model. The ML estimator $\hat{\theta}^{\text{ML}}$ is defined as the value of the parameters that makes the available observations $\mathbf{y}_{0:T}$ as likely as possible according to,

$$\hat{\theta}^{\text{ML}} = \arg \max_{\theta} p(\mathbf{y}_{0:T} | \theta). \quad (6)$$

The GP-SSM has two types of latent variables that need to be marginalized (integrated out) in order to compute the likelihood

$$\begin{aligned} p(\mathbf{y}_{0:T} | \theta) &= \int p(\mathbf{y}_{0:T}, \mathbf{x}_{0:T}, \mathbf{f}_{1:T} | \theta) d\mathbf{x}_{0:T} d\mathbf{f}_{1:T} \\ &= \int p(\mathbf{y}_{0:T} | \mathbf{x}_{0:T}, \theta) \left(\int p(\mathbf{x}_{0:T}, \mathbf{f}_{1:T} | \theta) d\mathbf{f}_{1:T} \right) d\mathbf{x}_{0:T}. \end{aligned} \quad (7)$$

Following results from Frigola et al. [2013], the latent variables $\mathbf{f}_{1:T}$ can be marginalized analytically. This is equivalent to integrating out the uncertainty in the unknown function f and working directly with a prior over the state trajectories $p(\mathbf{x}_{0:T} | \theta)$ that encodes the assumptions (e.g. smoothness) of f specified in (5a). The prior over trajectories can be factorized as

$$p(\mathbf{x}_{0:T} | \theta) = p(\mathbf{x}_0 | \theta) \prod_{t=1}^T p(\mathbf{x}_t | \theta, \mathbf{x}_{0:t-1}). \quad (8)$$

Using standard expressions for GP prediction, the one-step predictive density is given by

$$p(\mathbf{x}_t | \theta, \mathbf{x}_{0:t-1}) = \mathcal{N}(\mathbf{x}_t | \boldsymbol{\mu}_t(\mathbf{x}_{0:t-1}), \boldsymbol{\Sigma}_t(\mathbf{x}_{0:t-1})), \quad (9a)$$

where

$$\boldsymbol{\mu}_t(\mathbf{x}_{0:t-1}) = \mathbf{m}_{t-1} + \mathbf{K}_{t-1,0:t-2} \tilde{\mathbf{K}}_{0:t-2}^{-1} (\mathbf{x}_{1:t-1} - \mathbf{m}_{0:t-2}), \quad (9b)$$

$$\boldsymbol{\Sigma}_t(\mathbf{x}_{0:t-1}) = \tilde{\mathbf{K}}_{t-1} - \mathbf{K}_{t-1,0:t-2} \tilde{\mathbf{K}}_{0:t-2}^{-1} \mathbf{K}_{t-1,0:t-2}^\top, \quad (9c)$$

for $t \geq 2$ and $\boldsymbol{\mu}_1(\mathbf{x}_0) = \mathbf{m}_0$, $\boldsymbol{\Sigma}_1(\mathbf{x}_0) = \tilde{\mathbf{K}}_0$. Here we have defined the mean vector $\mathbf{m}_{0:t-1} \triangleq [m(\mathbf{x}_0)^\top \dots m(\mathbf{x}_{t-1})^\top]^\top$ and the $(n_x t) \times (n_x t)$ positive definite matrix $\mathbf{K}_{0:t-1}$ with block entries $[\mathbf{K}_{0:t-1}]_{i,j} = k(\mathbf{x}_{i-1}, \mathbf{x}_{j-1})$. These matrices use two sets of indices, as in $\mathbf{K}_{t-1,0:t-2}$, to refer to the off-diagonal blocks of $\mathbf{K}_{0:t-1}$. We also define $\tilde{\mathbf{K}}_{0:t-1} = \mathbf{K}_{0:t-1} + \mathbf{I}_t \otimes \mathbf{Q}$, where \otimes denotes the Kronecker product.

Using (8) we can thus write the likelihood (7) as

$$p(\mathbf{y}_{0:T} | \boldsymbol{\theta}) = \int p(\mathbf{y}_{0:T} | \mathbf{x}_{0:T}, \boldsymbol{\theta}) p(\mathbf{x}_{0:T} | \boldsymbol{\theta}) d\mathbf{x}_{0:T}. \quad (10)$$

The integration with respect to $\mathbf{x}_{0:T}$, however, is not analytically tractable. This difficulty will be addressed in the subsequent section.

A GP-SSM can be seen as a hierarchical probabilistic model which describes a prior over the latent state trajectories $p(\mathbf{x}_{0:T} | \boldsymbol{\theta}_x, \mathbf{Q})$ and links this prior with the observed data via the likelihood $p(\mathbf{y}_t | \mathbf{x}_t, \boldsymbol{\theta}_y)$. Direct application of maximum likelihood on $p(\mathbf{y}_t | \mathbf{x}_t, \boldsymbol{\theta}_y)$ to obtain estimates of the state trajectory and likelihood parameters would invariably result in over-fitting. However, by introducing a prior on the state trajectories¹ and marginalizing them as in (10), we obtain the so-called marginal likelihood. Maximization of the marginal likelihood with respect to the parameters results in a procedure known as type II maximum likelihood or empirical Bayes [Bishop, 2006]. Empirical Bayes reduces the risk of over-fitting since it automatically incorporates a trade-off between model fit and model complexity, a property often known as Bayesian Occam's razor [Ghahramani, 2012].

4. PARTICLE STOCHASTIC APPROXIMATION EM

As pointed out above, direct evaluation of the likelihood (10) is not possible for a GP-SSM. However, by viewing the latent states $\mathbf{x}_{0:T}$ as missing data, we are able to evaluate the *complete data* log-likelihood

$$\log p(\mathbf{y}_{0:T}, \mathbf{x}_{0:T} | \boldsymbol{\theta}) = \log p(\mathbf{y}_{0:T} | \mathbf{x}_{0:T}, \boldsymbol{\theta}) + \log p(\mathbf{x}_{0:T} | \boldsymbol{\theta}), \quad (11)$$

by using (5c) and (9). We therefore turn to the Expectation Maximization (EM) algorithm [Dempster et al., 1977]. The EM algorithm uses (11) to construct a surrogate cost function for the ML problem, defined as

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}') = \mathbb{E}_{\boldsymbol{\theta}'}[\log p(\mathbf{y}_{0:T}, \mathbf{x}_{0:T} | \boldsymbol{\theta}) | \mathbf{y}_{0:T}] \\ = \int \log p(\mathbf{y}_{0:T}, \mathbf{x}_{0:T} | \boldsymbol{\theta}) p(\mathbf{x}_{0:T} | \mathbf{y}_{0:T}, \boldsymbol{\theta}') d\mathbf{x}_{0:T}. \quad (12)$$

It is an iterative procedure that maximizes (10) by iterating two steps, expectation (E) and maximization (M),

- (E) Compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{k-1})$.
- (M) Compute $\boldsymbol{\theta}_k = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{k-1})$.

¹ A prior over the state trajectories is not an exclusive feature of GP-SSMs. Linear-Gaussian state-space models, for instance, also describe a prior distribution over state trajectories: $p(\mathbf{x}_{1:T} | \mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{x}_0)$.

The resulting sequence $\{\boldsymbol{\theta}_k\}_{k \geq 0}$ will, under weak assumptions, converge to a stationary point of the likelihood $p(\mathbf{y}_{0:T} | \boldsymbol{\theta})$.

To implement the above procedure we need to compute the integral in (12), which in general is not computationally tractable for a GP-SSM. To deal with this difficulty, we employ a Monte-Carlo-based implementation of the EM algorithm, referred to as PSAEM [Lindsten, 2013]. This procedure is a combination of stochastic approximation EM (SAEM) [Delyon et al., 1999] and particle Markov chain Monte Carlo (PMCMC) [Andrieu et al., 2010, Lindsten et al., 2012]. As illustrated by Lindsten [2013], PSAEM is a competitive alternative to particle-smoothing-based EM algorithms (e.g. [Schön et al., 2011, Olsson et al., 2008]), as it enjoys better convergence properties and has a much lower computational cost. The method maintains a stochastic approximation of the auxiliary quantity (12), $\hat{Q}_k(\boldsymbol{\theta}) \approx Q(\boldsymbol{\theta}, \boldsymbol{\theta}_{k-1})$. This approximation is updated according to

$$\hat{Q}_k(\boldsymbol{\theta}) = (1 - \gamma_k) \hat{Q}_{k-1}(\boldsymbol{\theta}) + \gamma_k \log p(\mathbf{y}_{0:T}, \mathbf{x}_{0:T}[k] | \boldsymbol{\theta}). \quad (13)$$

Here, $\{\gamma_k\}_{k \geq 0}$ is a sequence of step sizes, satisfying the usual stochastic approximation conditions: $\sum_k \gamma_k = \infty$ and $\sum_k \gamma_k^2 < \infty$. A typical choice is to take $\gamma_k = k^{-p}$ with $p \in]0.5, 1]$, where a smaller value of p gives a more rapid convergence at the cost of higher variance. In the vanilla SAEM algorithm, $\mathbf{x}_{0:T}[k]$ is a draw from the smoothing distribution $p(\mathbf{x}_{0:T} | \mathbf{y}_{0:T}, \boldsymbol{\theta}_{k-1})$. In this setting, Delyon et al. [1999] show that using the stochastic approximation (13) instead of (12) in the EM algorithm results in a valid method, i.e. $\{\boldsymbol{\theta}_k\}_{k \geq 0}$ will still converge to a maximizer of $p(\mathbf{y}_{0:T} | \boldsymbol{\theta})$.

The PSAEM algorithm is an extension of SAEM, which is useful when it is not possible to sample directly from the joint smoothing distribution. This is indeed the case in our setting. Instead of sampling from the smoothing distribution, the sample trajectory $\mathbf{x}_{0:T}[k]$ in (13) may be drawn from an ergodic Markov kernel, leaving the smoothing distribution invariant. Under suitable conditions on the kernel, this will not violate the validity of SAEM; see [Andrieu and Vihola, 2011, Andrieu et al., 2005].

In PSAEM, this Markov kernel on the space of trajectories, denoted as $P_{\boldsymbol{\theta}}^N(\mathbf{x}'_{0:T} | \mathbf{x}^*_{0:T})$, is constructed using PMCMC theory. In particular, we use the method by Lindsten et al. [2012], particle Gibbs with ancestor sampling (PGAS). We have previously used PGAS for Bayesian identification of GP-SSMs [Frigola et al., 2013]. PGAS is a sequential Monte Carlo method, akin to a standard particle filter (see e.g. [Doucet and Johansen, 2011, Gustafsson, 2010]), but with the difference that one particle at each time point is specified *a priori*. These reference states, denoted as $\mathbf{x}'_{0:T}$, can be thought of as guiding the particles of the particle filter to the "correct" regions of the state-space. More formally, as shown by Lindsten et al. [2012], PGAS defines a Markov kernel which leaves the joint smoothing distribution invariant, i.e. for any $\boldsymbol{\theta}$,

$$\int P_{\boldsymbol{\theta}}^N(\mathbf{x}'_{0:T} | \mathbf{x}'_{0:T}) p(\mathbf{x}'_{0:T} | \mathbf{y}_{0:T}, \boldsymbol{\theta}) d\mathbf{x}'_{0:T} = p(\mathbf{x}'_{0:T} | \mathbf{y}_{0:T}, \boldsymbol{\theta}). \quad (14)$$

The PGAS kernel is indexed by N , which is the number of particles used in the underlying particle filter. Note in particular that the desired property (14) holds for any $N \geq 1$, i.e. the number of particles only affects the mixing of the Markov kernel. A larger N implies faster mixing, which in turn results in better approximations of the auxiliary quantity (13). However, it has been experienced in practice that the correlation between consecutive trajectories drops of quickly as N increases [Lindsten et al., 2012, Lindsten and Schön, 2013], and for many models a moderate N (e.g. in the range 5–20) is enough to get a rapidly mixing kernel. We refer to [Lindsten, 2013, Lindsten et al., 2012] for details. We conclude by noting that it is possible to generate a sample $\mathbf{x}_{0:T}[k] \sim P_{\boldsymbol{\theta}[k-1]}^N(\cdot | \mathbf{x}_{0:T}[k-1])$ by running a particle-filter-like algorithm. This method is given as Algorithm 1 in [Lindsten et al., 2012] and is described specifically for GP-SSMs in Section 3 of [Frigola et al., 2013].

Next, we address the M-step of the EM algorithm. Maximizing the quantity (13) will typically not be possible in closed form. Instead, we make use of a numerical optimization routine implementing a quasi-Newton method (BFGS). Using (11), the gradient of the complete data log-likelihood can be written as

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y}_{0:T}, \mathbf{x}_{0:T} | \boldsymbol{\theta}) &= \sum_{t=0}^T \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y}_t | \mathbf{x}_t, \boldsymbol{\theta}) \\ &+ \sum_{t=1}^T \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \boldsymbol{\theta}) + \frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{x}_0 | \boldsymbol{\theta}), \end{aligned} \quad (15)$$

where the individual terms can be computed using (5c) and (9), respectively. The resulting PSAEM algorithm for learning of GP-SSMs is summarized in Algorithm 1.

Algorithm 1 PSAEM for GP-SSMs

- (1) Set $\boldsymbol{\theta}_0$ and $\mathbf{x}_{0:T}[0]$ arbitrarily. Set $\widehat{Q}_0(\boldsymbol{\theta}) \equiv 0$.
 - (2) For $k \geq 1$:
 - (a) Simulate $\mathbf{x}_{0:T}[k] \sim P_{\boldsymbol{\theta}[k-1]}^N(\cdot | \mathbf{x}_{0:T}[k-1])$ (run Algorithm 1 in [Lindsten et al., 2012] and set $\mathbf{x}_{0:T}[k]$ to one of the particle trajectories with probabilities given by their importance weights).
 - (b) Update $\widehat{Q}_k(\boldsymbol{\theta})$ according to (13).
 - (c) Compute $\boldsymbol{\theta}_k = \arg \max_{\boldsymbol{\theta}} \widehat{Q}_k(\boldsymbol{\theta})$.
-

A particular feature of the proposed approach is that it performs smoothing even when the state transition function is not yet explicitly defined. Once samples from the smoothing distribution have been obtained it is then possible to analytically describe the state transition probability density (see [Frigola et al., 2013] for details). This contrasts with the standard procedure where the smoothing distribution is found using a given state transition density.

5. EXPERIMENTAL RESULTS

In this section we present the results of applying PSAEM to identify various dynamical systems.

5.1 Identification of a Linear System

Although GP-SSMs are particularly suited to nonlinear system identification, we start by illustrating their behav-

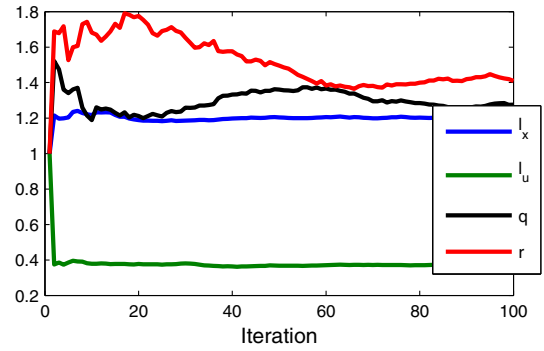


Fig. 1. Convergence of parameters when learning a linear system using a linear covariance function.

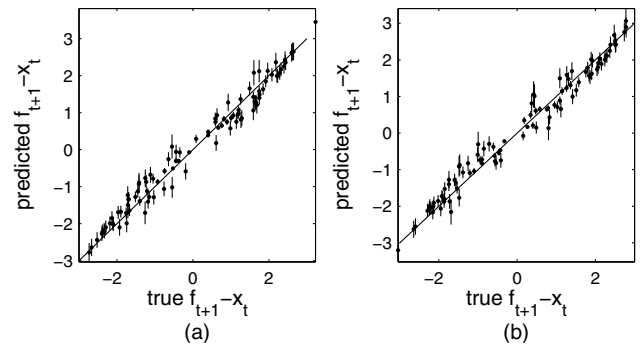


Fig. 2. Linear dynamical system learned using a GP-SSM with linear covariance function. Predictions (a) on training data, and (b) on test data (see text for more details).

ior when identifying the following linear system

$$\mathbf{x}_{t+1} = 0.8 \mathbf{x}_t + 3 \mathbf{u}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, 1.5), \quad (16a)$$

$$\mathbf{y}_t = 2 \mathbf{x}_t + \mathbf{e}_t, \quad \mathbf{e}_t \sim \mathcal{N}(0, 1.5), \quad (16b)$$

excited by a periodic input. The GP-SSM can model this linear system by using a linear covariance function for the GP. This covariance function imposes, in a somehow indirect fashion, that the state-transition function in (16a) must be linear. A GP-SSM with linear covariance function is formally equivalent to a linear state-space model where a Gaussian prior is placed over the, unknown to us, parameters ($A = 0.8$ and $B = 3$) [Rasmussen and Williams, 2006, Section 2.1]. The hyper-parameters of the covariance function are equivalent to the variances of a zero-mean prior over A and B . Therefore, the application of PSAEM to this particular GP-SSM can be interpreted as finding the hyper-parameters of a Gaussian prior over the parameters of the linear model that maximize the likelihood of the observed data whilst marginalizing over A and B . In addition, the likelihood will be simultaneously optimized with respect to the process noise and measurement noise variances (q and r respectively).

Figure 1 shows the convergence of the GP hyper-parameters (l_x and l_u) and noise parameters with respect to the PSAEM iteration. In order to judge the quality of the learned GP-SSM we evaluate its predictive performance on the data set used for learning (training set) and on an independent data set generated from the same dynamical system (test set). The GP-SSM can make prob-

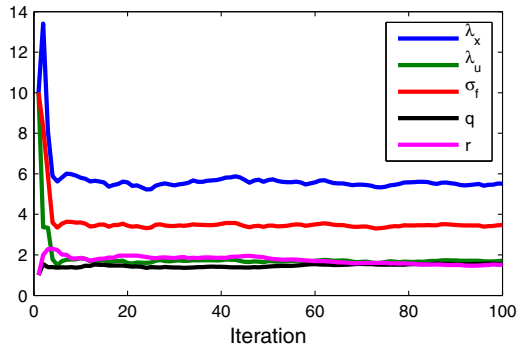


Fig. 3. Convergence of parameters when learning a linear system using a squared exponential covariance function.

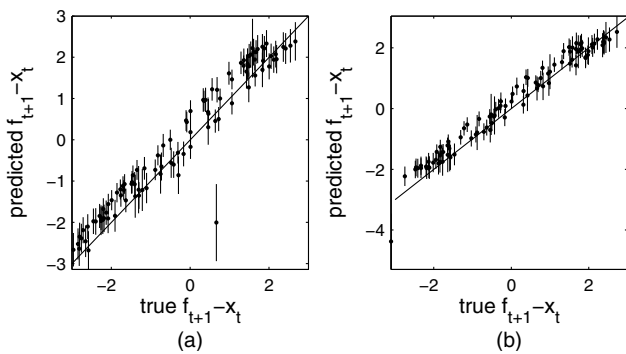


Fig. 4. Linear dynamical system learned using a GP-SSM with squared exponential covariance function. Predictions (a) on training data, and (b) on test data.

abilistic predictions which report the uncertainty arising from the fact that only a finite amount of data is observed.

Figure 2 displays the predicted value of $\mathbf{f}_{t+1} - \mathbf{x}_t$ versus the true value. Recall that $\mathbf{f}_{t+1} - \mathbf{x}_t$ is equivalent to the step taken by the state in one single transition before process noise is added: $f(\mathbf{x}_t, \mathbf{u}_t) - \mathbf{x}_t$. One standard deviation error bars from the predictive distribution have also been plotted. Perfect predictions would lie on the unit slope line. We note that although the predictions are not perfect, error-bars tend to be large in predictions that are far from the true value and narrower for predictions that are closer to the truth. This is the desired outcome since the goal of the GP-SSM is to represent the uncertainty in its predictions.

We now move into a scenario in which the data is still generated by the linear dynamical system in (16) but we pretend that we are not aware of its linearity. In this case, a covariance function able to model nonlinear transition functions is a judicious choice. We use the squared exponential covariance function which imposes the assumption that the state transition function is smooth and infinitely differentiable [Rasmussen and Williams, 2006]. Figure 3 shows, for a PSAEM run, the convergence of the covariance function hyper-parameters (length-scales λ_x and λ_u and signal variance σ_f) and also the convergence of the noise parameters.

The predictive performance on training data and independent test data is presented in Figure 4. Interestingly, in

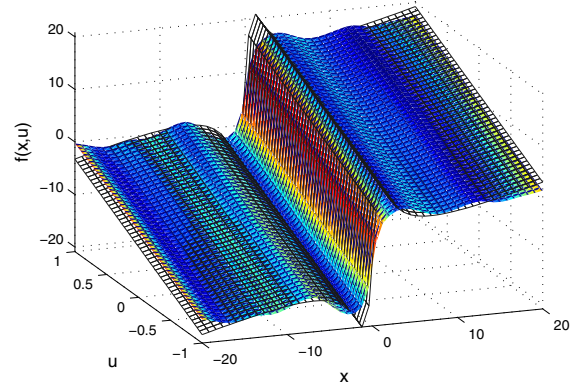


Fig. 5. Nonlinear dynamical system with one state and one input. The black mesh represents the ground truth dynamics function and the colored surface is the mean of the identified function. Color is proportional to the standard deviation of the identified function (red represents high uncertainty and blue low uncertainty).

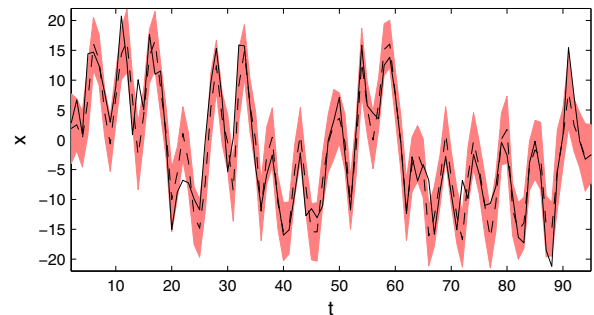


Fig. 6. State trajectory from a test data set (solid black line). One step ahead predictions made with the identified model are depicted by a dashed line (mean) and a colored interval at ± 1 standard deviation (including process noise).

the panel corresponding to training data (a), there is particularly poor prediction that largely underestimates the value of the state transition. However, the variance for this prediction is very high which indicates that the identified model has little confidence in it. In this particular case, the mean of the prediction is 2.5 standard deviations away from the true value of the state transition.

5.2 Identification of a Nonlinear System

GP-SSMs are particularly powerful for nonlinear system identification when it is not possible to create a parametric model of the system based on detailed knowledge about its dynamics. To illustrate this capability of GP-SSMs we consider the nonlinear dynamical system

$$\mathbf{x}_{t+1} = a\mathbf{x}_t + b \frac{\mathbf{x}_t}{1 + \mathbf{x}_t^2} + c\mathbf{u}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(0, q), \quad (17a)$$

$$\mathbf{y}_t = d\mathbf{x}_t^2 + \mathbf{e}_t, \quad \mathbf{e}_t \sim \mathcal{N}(0, r), \quad (17b)$$

with parameters $(a, b, c, d, q, r) = (0.5, 25, 8, 0.05, 10, 1)$ and a known input $\mathbf{u}_t = \cos(1.2(t + 1))$. One of the challenging properties of this system is that the quadratic measurement function (17b) tends to induce a bimodal distribution in the marginal smoothing distribution. For

instance, if we were to consider only one measurement in isolation and $r = 0$ we would have $\mathbf{x}_t = \pm \sqrt{\frac{y_t}{d}}$. Moreover, the state transition function (17a) exhibits a very sharp gradient in the \mathbf{x}_t direction at the origin, but is otherwise parsimonious as $\mathbf{x}_t \rightarrow \pm\infty$.

Again, we pretend that detailed knowledge about the particular form of (17a) is not available to us. We select a covariance function that consists of a Matérn covariance function in the \mathbf{x} direction and a squared exponential in the \mathbf{u} direction. The Matérn covariance function imposes less smoothness constraints than the squared exponential [Rasmussen and Williams, 2006] and is therefore more suited to model functions that can have sharp transitions.

Figure 5 shows the true state transition dynamics function (black mesh) and the identified function as a colored surface. Since the identified function from the GP-SSM comes in the form of a probability distribution over functions, the surface is plotted at $\mathbb{E}[\mathbf{f}^*|\mathbf{x}^*, \mathbf{u}^*, \mathbf{y}_{0:T}]$ where the symbol * denotes test points. The standard deviation of \mathbf{f}^* , which represents our uncertainty about the actual value of the function, is depicted by the color of the surface. Figure 6 shows the one step ahead predictive distributions $p(\mathbf{x}_{t+1}^*|\mathbf{x}_t^*, \mathbf{u}_t^*, \mathbf{y}_{0:T})$ on a test data set.

6. CONCLUSIONS

GP-SSMs allow for a high degree of flexibility when addressing the nonlinear system identification problem by making use of Bayesian nonparametric system models. These models enable the incorporation of high-level assumptions, such as smoothness of the transition function, while still being able to capture a wide range of nonlinear dynamical functions. Furthermore, the GP-SSM is capable of making probabilistic predictions that can be useful in adaptive control and robotics, when the control strategy might depend on the uncertainty in the dynamics. Our particle-filter-based maximum likelihood inference of the model hyper-parameters preserves the full nonparametric richness of the model. In addition, marginalization of the dynamical function effectively averages over all possible dynamics consistent with the GP prior and the data, and hence provides a strong safeguard against overfitting.

REFERENCES

- C. Andrieu and M. Vihola. Markovian stochastic approximation with expanding projections. arXiv.org, arXiv:1111.5421, November 2011.
- C. Andrieu, E. Moulines, and P. Priouret. Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization*, 44(1):283–312, 2005.
- C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010. ISSN 1467-9868.
- C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- M. P. Deisenroth and S. Mohamed. Expectation Propagation in Gaussian process dynamical systems. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 25*, pages 2618–2626. 2012.
- M. P. Deisenroth, R. D. Turner, M. F. Huber, U. D. Hanebeck, and C. E. Rasmussen. Robust filtering and smoothing with Gaussian processes. *IEEE Transactions on Automatic Control*, 57(7):1865–1871, July 2012. ISSN 0018-9286. doi: 10.1109/TAC.2011.2179426.
- B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):pp. 94–128, 1999.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977.
- A. Doucet and A. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan and B. Rozovskii, editors, *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011.
- 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 L. Bottou, C.J.C. Burges, Z. Ghahramani, M. Welling, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 26*. 2013.
- Z. Ghahramani. Bayesian nonparametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society A*, 2012.
- F. Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82, 2010.
- J. Ko and D. Fox. Learning GP-BayesFilters via Gaussian process latent variable models. *Autonomous Robots*, 30(1):3–23, 2011.
- F. Lindsten. An efficient stochastic approximation EM algorithm using conditional particle filters. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6274–6278, 2013.
- F. Lindsten and T. B. Schön. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143, 2013.
- F. Lindsten, M. Jordan, and T. B. Schön. Ancestor sampling for particle Gibbs. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 25*, pages 2600–2608. 2012.
- J. Olsson, R. Douc, O. Cappé, and E. Moulines. Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state-space models. *Bernoulli*, 14(1):155–179, 2008.
- V. Peterka. Bayesian system identification. *Automatica*, 17(1):41–53, 1981.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- T. B. Schön, A. Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.
- R. Turner, M. P. Deisenroth, and C. E. Rasmussen. State-space inference and learning with Gaussian processes. In Yee Whye Teh and Mike Titterton, editors, *13th International Conference on Artificial Intelligence and Statistics*, volume 9 of *W&CP*, pages 868–875, Chia Laguna, Sardinia, Italy, May 13–15 2010.