Expectation Propagation in Gaussian Process Dynamical Systems

Marc Peter Deisenroth*
Department of Computer Science
Technische Universität Darmstadt, Germany

Shakir Mohamed*
Department of Computer Science
University of British Columbia, Canada

Abstract

Rich and complex time-series data, such as those generated from engineering systems, financial markets, videos, or neural recordings are now a common feature of modern data analysis. Explaining the phenomena underlying these diverse data sets requires flexible and accurate models. In this paper, we promote Gaussian process dynamical systems as a rich model class that is appropriate for such an analysis. We present a new approximate message-passing algorithm for Bayesian state estimation and inference in Gaussian process dynamical systems, a non-parametric probabilistic generalization of commonly used state-space models. We derive our message-passing algorithm using Expectation Propagation and provide a unifying perspective on message passing in general state-space models. We show that existing Gaussian filters and smoothers appear as special cases within our inference framework, and that these existing approaches can be improved upon using iterated message passing. Using both synthetic and real-world data, we demonstrate that iterated message passing can improve inference in a wide range of tasks in Bayesian state estimation, thus leading to improved predictions and more effective decision making.

1 Introduction

The Kalman filter and its extensions [1], such as the extended and unscented Kalman filters [7], are principled statistical models that have been widely used for some of the most challenging and mission-critical applications in automatic control, robotics, machine learning, and economics. Indeed, wherever complex time-series are found, Kalman filters have been successfully applied for Bayesian state estimation. However, in practice, time series often have an unknown dynamical structure, and they are high dimensional and noisy, violating many of the assumptions made in established approaches for state estimation. In this paper, we look beyond traditional linear dynamical systems and advance the state-of the-art in state estimation by developing novel inference algorithms for the class of nonlinear *Gaussian process dynamical systems* (GPDS).

GPDSs are non-parametric generalizations of state-space models that allow for inference in time series, using Gaussian process (GP) probability distributions over nonlinear transition and measurement dynamics. GPDSs are thus able to capture complex dynamical structure with few assumptions, making them of broad interest. This interest has sparked the development of general approaches for filtering and smoothing in GPDSs, such as [8, 3, 5]. In this paper, we further develop inference algorithms for GPDSs and make the following contributions: (1) We develop an iterative local message passing framework for GPDSs based on Expectation Propagation (EP) [11, 10], which allows for refinement of the posterior distribution and, hence, improved inference. (2) We show that the general message-passing framework recovers the EP updates for existing dynamical systems as a special case and expose the implicit modeling assumptions made in these models. We show that EP in GPDSs encapsulates all GPDS forward-backward smoothers [5] as a special case and transforms them into iterative algorithms yielding more accurate inference.

* Authors contributed equally.

2 Gaussian Process Dynamical Systems

Gaussian process dynamical systems are a general class of discrete-time state-space models with

$$\boldsymbol{x}_t = h(\boldsymbol{x}_{t-1}) + \boldsymbol{w}_t\,, \quad \boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})\,, \quad h \sim \mathcal{GP}_h\,, \tag{1}$$

$$\boldsymbol{z}_t = g(\boldsymbol{x}_t) + \boldsymbol{v}_t\,, \quad \boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})\,, \quad g \sim \mathcal{GP}_g\,, \tag{2}$$

where $t = 1, \dots, T$. Here, $\boldsymbol{x} \in \mathbb{R}^D$ is a latent state that evolves over time, and $\boldsymbol{z} \in \mathbb{R}^E$, $E \geq D$, are measurements. We assume i.i.d. additive Gaussian system noise $\boldsymbol{w}$ and measurement noise $\boldsymbol{v}$. The central feature of this model class is that both the measurement function g and the transition function h are not explicitly known or parametrically specified, but instead described by probability distributions over these functions. The function distributions are non-parametric Gaussian processes (GPs), and we write $h \sim \mathcal{GP}_h$ and $g \sim \mathcal{GP}_g$, respectively.

A GP is a probability distribution $p(f)$ over functions f that is specified by a mean function μ_f and a covariance function k_f [15]. Consider a set of training inputs $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n]^\top$ and corresponding training targets $\boldsymbol{y} = [y_1, \dots y_n]^\top$, $y_i = f(\boldsymbol{x}_i) + w$, $w \sim \mathcal{N}(0, \sigma_w^2)$. The posterior predictive distribution at a test input $\boldsymbol{x}_*$ is Gaussian distributed $\mathcal{N}(y_* \mid \mu_f(\boldsymbol{x}_*), \sigma_f^2(\boldsymbol{x}_*))$ with mean $\mu_f(\boldsymbol{x}_*) = \boldsymbol{k}_*^\top \boldsymbol{K}^{-1}\boldsymbol{y}$ and variance $\sigma_f^2(\boldsymbol{x}_*) = k_{**} - \boldsymbol{k}_*^\top \boldsymbol{K}^{-1}\boldsymbol{k}_*$, where $\boldsymbol{k}_* = k_f(\boldsymbol{X}, \boldsymbol{x}_*)$, $k_{**} = k_f(\boldsymbol{x}_*, \boldsymbol{x}_*)$, and $\boldsymbol{K}$ is the kernel matrix.

Since the GP is a non-parametric model, its use in GPDSs is desirable as it results in fewer restrictive model assumptions, compared to dynamical systems based on parametric function approximators for the transition and measurement functions (1)–(2). In this paper, we assume that the GP models are trained, i.e., the training inputs and corresponding targets as well as the GP hyperparameters are known. For both $\mathcal{GP}_h$ and $\mathcal{GP}_g$ in the GPDS, we used zero prior mean functions. As covariance functions k_h and k_g we use squared- exponential covariance functions with automatic relevance determination plus a noise covariance function to account for the noise in (1)–(2).

Existing work for *learning* GPDSs includes the Gaussian process dynamical model (GPDM) [20], which tackles the challenging task of analyzing human motion in (high-dimensional) video sequences. More recently, variational [2] and EM-based [19] approaches for learning GPDS were proposed. Exact Bayesian *inference*, i.e., filtering and smoothing, in GPDSs is analytically intractable because of the dependency of the states and measurements on previous states through the nonlinearity of the GP. We thus make use of approximations to infer the posterior distributions $p(\boldsymbol{x}_t|\boldsymbol{Z})$ over latent states $\boldsymbol{x}_t$, $t = 1, \dots, T$, given a set of observations $\boldsymbol{Z} = \boldsymbol{z}_{1:T}$. Existing approximate inference approaches for filtering and forward-backward smoothing are based on either linearization, particle representations, or moment matching as approximation strategies [8, 3, 5].

A principled incorporation of the posterior GP model uncertainty into inference in GPDSs is necessary, but introduces additional uncertainty. In tracking problems where the location of an object is not directly observed, this additional source of uncertainty can eventually lead to losing track of the latent state. In this paper, we address this problem and propose approximate message passing based on EP for more accurate inference. We will show that forward-backward smoothing in GPDSs [5] benefits from the iterative refinement scheme of EP, leading to more accurate posterior distributions over the latent state and, hence, to more informative predictions and improved decision making.

3 Bayesian State Estimation using Expectation Propagation

Expectation Propagation [10, 11] is a widely-used deterministic algorithm for approximate Bayesian inference that has been shown to be highly accurate in many problems, including sparse regression models [17], GP classification [9], and inference in dynamical systems [13, 6, 18]. EP is derived using a factor-graph, in which the distribution over the latent state $p(\boldsymbol{x}_t|\boldsymbol{Z})$ is represented as the product of factors $f_i(\boldsymbol{x}_t)$, i.e., $p(\boldsymbol{x}_t|\boldsymbol{Z}) = \prod_i f_i(\boldsymbol{x}_t)$. EP then specifies an iterative message passing algorithm in which $p(\boldsymbol{x}_t|\boldsymbol{Z})$ is approximated by a distribution $q(\boldsymbol{x}_t) = \prod_i q_i(\boldsymbol{x}_t)$, using approximate messages $q_i(\boldsymbol{x}_t)$. In EP, q and the messages q_i are members of the exponential family, and q is determined such that the the KL-divergence KL($p||q$) is minimized. EP is provably robust for log-concave messages [17] and invariant under invertible variable transformations [16]. In practice, EP has been shown to be more accurate than competing approximate inference methods [9, 17].

In the context of the dynamical system (1)–(2), we consider factor graphs of the form of Fig. 1 with three types of messages: forward, backward, and measurement messages, denoted by the symbols

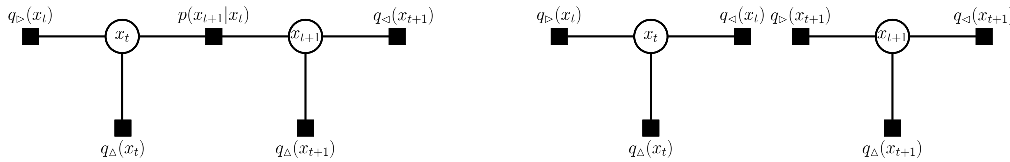


Figure 1: Factor graph (left) and fully factored graph (right) of a general dynamical system.

Algorithm 1 Gaussian EP for Dynamical Systems

1: **Init:** Set all factors q_i to $\mathcal{N}(\mathbf{0}, \infty \boldsymbol{I})$; Set $q(\boldsymbol{x}_1) = p(\boldsymbol{x}_1)$ and marginals $q(\boldsymbol{x}_{t \neq 1}) = \mathcal{N}(\mathbf{0}, 10^{10}\boldsymbol{I})$
2: **repeat**
3:     **for** $t = 1$ to T **do**
4:         **for** all factors $q_i(\boldsymbol{x}_t)$, where $i = \triangleright, \triangle, \triangleleft$ **do**
5:             Compute cavity distribution $q^{\setminus i}(\boldsymbol{x}_t) = q(\boldsymbol{x}_t)/q_i(\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_t \,|\, \boldsymbol{\mu}^{\setminus i}, \boldsymbol{\Sigma}^{\setminus i})$ with

$$\boldsymbol{\Sigma}^{\setminus i} = (\boldsymbol{\Sigma}_t^{-1} - \boldsymbol{\Sigma}_i^{-1})^{-1}\,, \qquad \boldsymbol{\mu}^{\setminus i} = \boldsymbol{\Sigma}^{\setminus i}(\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\mu}_t - \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i) \tag{3}$$

6:             Determine moments of $f_i(\boldsymbol{x}_t)q^{\setminus i}(\boldsymbol{x}_t)$, e.g., via the derivatives of

$$\log Z_i(\boldsymbol{\mu}^{\setminus i}, \boldsymbol{\Sigma}^{\setminus i}) = \log \textstyle\int f_i(\boldsymbol{x}_t)q^{\setminus i}(\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_t \tag{4}$$

7:             Update the posterior $q(\boldsymbol{x}_t) \propto \mathcal{N}(\boldsymbol{x}_t \,|\, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ and the approximate factor $q_i(\boldsymbol{x}_t)$:

$$\boldsymbol{\mu}_t = \boldsymbol{\mu}^{\setminus i} + \boldsymbol{\Sigma}^{\setminus i}\boldsymbol{\nabla}_m^\top\,, \qquad \boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}^{\setminus i} - \boldsymbol{\Sigma}^{\setminus i}(\boldsymbol{\nabla}_m^\top\boldsymbol{\nabla}_m - 2\boldsymbol{\nabla}_s)\boldsymbol{\Sigma}^{\setminus i} \tag{5}$$

$$\boldsymbol{\nabla}_m \coloneqq \mathrm{d}\log Z_i/\mathrm{d}\boldsymbol{\mu}^{\setminus i}\,, \qquad \boldsymbol{\nabla}_s \coloneqq \mathrm{d}\log Z_i/\mathrm{d}\boldsymbol{\Sigma}^{\setminus i} \tag{6}$$

$$q_i(\boldsymbol{x}_t) = q(\boldsymbol{x}_t)/q^{\setminus i}(\boldsymbol{x}_t) \tag{7}$$

8:         **end for**
9:     **end for**
10: **until** Convergence or maximum number of iterations exceeded

---

$\triangleright, \triangleleft, \triangle$, respectively. For EP inference, we assume a fully-factored graph, using which we compute the marginal posterior distributions $p(\boldsymbol{x}_1|\boldsymbol{Z}), \ldots, p(\boldsymbol{x}_T|\boldsymbol{Z})$, rather than the full joint distribution $p(\boldsymbol{X}|\boldsymbol{Z}) = p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T|\boldsymbol{Z})$. Both the states $\boldsymbol{x}_t$ and measurements $\boldsymbol{z}_t$ are continuous variables and the messages q_i are unnormalized Gaussians, i.e., $q_i(\boldsymbol{x}_t) = s_i\mathcal{N}(\boldsymbol{x}_t \,|\, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

3.1 Implicit Linearizations Require Explicit Consideration

Alg. 1 describes the main steps of Gaussian EP for dynamical systems. For each node $\boldsymbol{x}_t$ in the fully-factored factor graph in Fig. 1, EP computes three messages: a forward, backward, and measurement message, denoted by $q_\triangleright(\boldsymbol{x}_t)$, $q_\triangleleft(\boldsymbol{x}_t)$, and $q_\triangle(\boldsymbol{x}_t)$, respectively. The EP algorithm updates the marginal $q(\boldsymbol{x}_t)$ and the messages $q_i(\boldsymbol{x}_t)$ in three steps. First, the *cavity distribution* $q^{\setminus i}(\boldsymbol{x}_t)$ is computed (step 5 in Alg. 1) by removing $q_i(\boldsymbol{x}_t)$ from the marginal $q(\boldsymbol{x}_t)$. Second, in the *projection* step, the moments of $f_i(\boldsymbol{x}_t)q^{\setminus i}(\boldsymbol{x}_t)$ are computed (step 6), where f_i is the true factor. In the exponential family, the required moments can be computed using the derivatives of the log-partition function (normalizing constant) $\log Z_i$ of $f_i(\boldsymbol{x}_t)q^{\setminus i}(\boldsymbol{x}_t)$ [10, 11, 12]. Third, the moments of the marginal $q(\boldsymbol{x}_t)$ are set to the moments of $f_i(\boldsymbol{x}_t)q^{\setminus i}(\boldsymbol{x}_t)$, and the message $q_i(\boldsymbol{x}_t)$ is updated (step 7). We apply this procedure repeatedly to all latent states $\boldsymbol{x}_t$, $t = 1, \ldots, T$, until convergence.

EP does not directly fit a Gaussian approximation q_i to the non-Gaussian factor f_i. Instead, EP determines the moments of q_i in the context of the cavity distribution such that $q_i = \mathrm{proj}[f_i q^{\setminus i}]/q^{\setminus i}$, where $\mathrm{proj}[\cdot]$ is the projection operator, returning the moments of its argument.

To update the posterior $q(\boldsymbol{x}_t)$ and the messages $q_i(\boldsymbol{x}_t)$, EP computes the log-partition function $\log Z_i$ in (4) to complete the projection step. However, for nonlinear transition and measurement models in (1)–(2), computing Z_i involves solving integrals of the form

$$p(\boldsymbol{a}) = \int p(\boldsymbol{a}|\boldsymbol{x}_t)p(\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_t = \int \mathcal{N}(\boldsymbol{a}\,|\,\boldsymbol{m}(\boldsymbol{x}_t), \boldsymbol{S}(\boldsymbol{x}_t))\mathcal{N}(\boldsymbol{x}_t\,|\,\boldsymbol{b}, \boldsymbol{B})\mathrm{d}\boldsymbol{x}_t\,, \tag{8}$$

where $\boldsymbol{a} = \boldsymbol{z}_t$ for the measurement message, or $\boldsymbol{a} = \boldsymbol{x}_{t+1}$ for the forward and backward messages. In nonlinear dynamical systems $\boldsymbol{m}(\boldsymbol{x}_t)$ is a nonlinear measurement or transition function. In GPDSs, $\boldsymbol{m}(\boldsymbol{x}_t)$ and $\boldsymbol{S}(\boldsymbol{x}_t)$ are the corresponding predictive GP means and covariances, respectively, which are nonlinearly related to $\boldsymbol{x}_t$. Because of the nonlinear dependencies between $\boldsymbol{a}$ and $\boldsymbol{x}_t$, solving (8) is analytically intractable. We propose to approximate $p(\boldsymbol{a})$ by a Gaussian distribution $\mathcal{N}(\boldsymbol{a}\,|\,\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})$. This Gaussian approximation is only correct for a linear relationship $\boldsymbol{a} = \boldsymbol{J}\boldsymbol{x}_t$, where $\boldsymbol{J}$ is independent of $\boldsymbol{x}_t$. Hence, the Gaussian approximation is an *implicit linearization* of the functional relationship between $\boldsymbol{a}$ and $\boldsymbol{x}_t$, effectively linearizing either the transition or the measurement models.

When computing EP updates using the derivatives $\boldsymbol{\nabla}_m$ and $\boldsymbol{\nabla}_s$ according to (5), it is crucial to explicitly account for the implicit linearization assumption in the derivatives—otherwise, the EP updates are inconsistent. For example, in the measurement and the backward message, we directly approximate the partition functions Z_i, $i \in \{\triangle, \triangleleft\}$ by Gaussians $\tilde{Z}_i(\boldsymbol{a}) = \mathcal{N}(\tilde{\boldsymbol{\mu}}^i, \tilde{\boldsymbol{\Sigma}}^i)$. The consistent derivatives $\mathrm{d}(\log \tilde{Z}_i)/\mathrm{d}\boldsymbol{\mu}^{\setminus i}$ and $\mathrm{d}(\log \tilde{Z}_i)/\mathrm{d}\boldsymbol{\Sigma}^{\setminus i}$ of $\tilde{Z}_i$ with respect to the mean and covariance of the cavity distribution q are obtained by applying the chain rule, such that

$$\boldsymbol{\nabla}_m = \tfrac{\mathrm{d}\log \tilde{Z}_i}{\mathrm{d}\boldsymbol{\mu}^{\setminus i}} = \tfrac{\partial \log \tilde{Z}_i}{\partial \tilde{\boldsymbol{\mu}}^i}\tfrac{\partial \tilde{\boldsymbol{\mu}}^i}{\partial \boldsymbol{\mu}^{\setminus i}} = (\boldsymbol{a} - \tilde{\boldsymbol{\mu}}^i)^\top (\tilde{\boldsymbol{\Sigma}}^i)^{-1}\boldsymbol{J}^\top \in \mathbb{R}^{1\times D}\,, \tag{9}$$

$$\boldsymbol{\nabla}_s = \tfrac{\mathrm{d}\log \tilde{Z}_i}{\mathrm{d}\boldsymbol{\Sigma}^{\setminus i}} = \tfrac{\partial \log \tilde{Z}_i}{\partial \tilde{\boldsymbol{\Sigma}}^i}\tfrac{\partial \tilde{\boldsymbol{\Sigma}}^i}{\partial \boldsymbol{\Sigma}^{\setminus i}} = \tfrac{1}{2}\left(\tfrac{\partial \log \tilde{Z}_i}{\partial (\tilde{\boldsymbol{\mu}}^i)^\top}\tfrac{\partial \log \tilde{Z}_i}{\partial \tilde{\boldsymbol{\mu}}^i} - \tilde{\boldsymbol{\Sigma}}^i\right)\tfrac{\partial \tilde{\boldsymbol{\Sigma}}^i}{\partial \boldsymbol{\Sigma}^{\setminus i}} \in \mathbb{R}^{D\times D}\,, \tag{10}$$

$$\tfrac{\partial \tilde{\boldsymbol{\mu}}^i}{\partial \boldsymbol{\mu}^{\setminus i}} = \boldsymbol{J}^\top \in \mathbb{R}^{E\times D}\,, \quad \tfrac{\partial \tilde{\boldsymbol{\Sigma}}^i}{\partial \boldsymbol{\Sigma}^{\setminus i}} = \boldsymbol{J}\mathbb{I}_4\boldsymbol{J}^\top \in \mathbb{R}^{E\times E\times D\times D}\,, \tag{11}$$

where $\mathbb{I}_4 \in \mathbb{R}^{D\times D\times D\times D}$ is an identity tensor. Note that with the implicit linear model $\boldsymbol{a} = \boldsymbol{J}\boldsymbol{x}_t$, the derivatives $\partial\tilde{\boldsymbol{\mu}}^i/\partial\boldsymbol{\Sigma}^{\setminus i}$ and $\partial\tilde{\boldsymbol{\Sigma}}^i/\partial\boldsymbol{\mu}^{\setminus i}$ vanish. Although we approximate Z_i by a Gaussian $\tilde{Z}_i$, we are still free to choose a method of computing its mean $\tilde{\boldsymbol{\mu}}^i$ and covariance matrix $\tilde{\boldsymbol{\Sigma}}^i$, which also influences the computation of $\boldsymbol{J} = \partial(\tilde{\boldsymbol{\mu}}^i)/\partial\boldsymbol{\mu}^{\setminus i}$. However, even if $\tilde{\boldsymbol{\mu}}^i$ and $\tilde{\boldsymbol{\Sigma}}^i$ are general functions of $\boldsymbol{\mu}^{\setminus i}$ and $\boldsymbol{\Sigma}^{\setminus i}$, the derivatives $\partial\tilde{\boldsymbol{\mu}}^i/\partial\boldsymbol{\mu}^{\setminus i}$ and $\partial\tilde{\boldsymbol{\Sigma}}^i/\partial\boldsymbol{\Sigma}^{\setminus i}$ must equal the corresponding partial derivatives in (11), and $\partial\tilde{\boldsymbol{\mu}}^i/\partial\boldsymbol{\Sigma}^{\setminus i}$ and $\partial\tilde{\boldsymbol{\Sigma}}^i/\partial\boldsymbol{\mu}^{\setminus i}$ must be set to $\mathbf{0}$. Hence, the implicit linearization expressed by the Gaussian approximation $\tilde{Z}_i$ must be explicitly taken into account in the derivatives to guarantee consistent EP updates.

3.2 Messages in Gaussian Process Dynamical Systems

We now describe each of the messages needed for inference in GPDSs, and outline the approximations required to compute the partition function in (4). Updating a message requires a *projection* to compute the moments of the new posterior marginal $q(\boldsymbol{x}_t)$, followed by a Gaussian division to update the message itself. For the projection step, we compute approximate partition functions $\tilde{Z}_i$, where $i \in \{\triangle, \triangleright, \triangleleft\}$. Using the derivatives $\mathrm{d}\log\tilde{Z}_i/\mathrm{d}\boldsymbol{\mu}_t^{\setminus i}$ and $\mathrm{d}\log\tilde{Z}_i/\mathrm{d}\boldsymbol{\Sigma}_t^{\setminus i}$, we update the marginal $q(\boldsymbol{x}_t)$, see (5).

Measurement Message For the measurement message in a GPDS, the partition function is

$$Z_\triangle(\boldsymbol{\mu}_t^{\setminus\triangle}, \boldsymbol{\Sigma}_t^{\setminus\triangle}) = \int f_\triangle(\boldsymbol{x}_t)q_{\setminus\triangle}(\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_t \propto \int f_\triangle(\boldsymbol{x}_t)\mathcal{N}(\boldsymbol{x}_t\,|\,\boldsymbol{\mu}_t^{\setminus\triangle}, \boldsymbol{\Sigma}_t^{\setminus\triangle})\mathrm{d}\boldsymbol{x}_t\,, \tag{12}$$

$$f_\triangle(\boldsymbol{x}_t) = p(\boldsymbol{z}_t|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{z}_t\,|\,\boldsymbol{\mu}_g(\boldsymbol{x}_t), \boldsymbol{\Sigma}_g(\boldsymbol{x}_t)), \tag{13}$$

where $f_\triangle$ is the true measurement factor, and $\boldsymbol{\mu}_g(\boldsymbol{x}_t)$ and $\boldsymbol{\Sigma}_g(\boldsymbol{x}_t)$ are the predictive mean and covariance of the measurement GP $\mathcal{GP}_g$. In (12), we made it explicit that $Z_\triangle$ depends on the moments $\boldsymbol{\mu}_t^{\setminus\triangle}$ and $\boldsymbol{\Sigma}_t^{\setminus\triangle}$ of the cavity distribution $q_{\setminus\triangle}(\boldsymbol{x}_t)$. The integral in (12) is of the form (8), but is intractable since solving it corresponds to a GP prediction at uncertain inputs [14], resulting in non-Gaussian predictive distributions. However, the mean and covariance of a Gaussian approximation $\tilde{Z}_\triangle$ to $Z_\triangle$ can be computed analytically: either using exact moment matching [14, 3], or approximately by expected linearization of the posterior GP [8]; details are given in [4]. The moments of

$\tilde{Z}_\Delta$ are also functions of the mean $\boldsymbol{\mu}_t^{\backslash\Delta}$ and variance $\boldsymbol{\Sigma}_t^{\backslash\Delta}$ of the cavity distribution. By taking the linearization assumption of the Gaussian approximation into account explicitly (here, we implicitly linearize $\mathcal{GP}_g$) when computing the derivatives, the EP updates remain consistent, see Sec. 3.1.

Backward Message To update the backward message $q_\triangleleft(\boldsymbol{x}_t)$, we require the partition function

$$Z_\triangleleft(\boldsymbol{\mu}_t^{\backslash\triangleleft}, \boldsymbol{\Sigma}_t^{\backslash\triangleleft}) = \int f_\triangleleft(\boldsymbol{x}_t) q_{\backslash\triangleleft}(\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_t \propto \int f_\triangleleft(\boldsymbol{x}_t)\mathcal{N}(\boldsymbol{x}_t \,|\, \boldsymbol{\mu}_t^{\backslash\triangleleft}, \boldsymbol{\Sigma}_t^{\backslash\triangleleft})\mathrm{d}\boldsymbol{x}_t \,, \tag{14}$$

$$f_\triangleleft(\boldsymbol{x}_t) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) q_{\backslash\triangleright}(\boldsymbol{x}_{t+1})\mathrm{d}\boldsymbol{x}_{t+1} = \int \mathcal{N}(\boldsymbol{x}_{t+1} \,|\, \boldsymbol{\mu}_h(\boldsymbol{x}_t), \boldsymbol{\Sigma}_h(\boldsymbol{x}_t)) q_{\backslash\triangleright}(\boldsymbol{x}_{t+1})\mathrm{d}\boldsymbol{x}_{t+1} \,. \tag{15}$$

Here, the true factor $f_\triangleleft(\boldsymbol{x}_t)$ in (15) takes into account the coupling between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$, which was lost in assuming the full factorization in Fig. 1. The predictive mean and covariance of $\mathcal{GP}_h$ are denoted $\boldsymbol{\mu}_h(\boldsymbol{x}_t)$ and $\boldsymbol{\Sigma}_h(\boldsymbol{x}_t)$, respectively. Using (15) in (14) and reordering the integration yields

$$Z_\triangleleft(\boldsymbol{\mu}_t^{\backslash\triangleleft}, \boldsymbol{\Sigma}_t^{\backslash\triangleleft}) \propto \int q_{\backslash\triangleright}(\boldsymbol{x}_{t+1}) \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) q_{\backslash\triangleleft}(\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_t \mathrm{d}\boldsymbol{x}_{t+1} \,. \tag{16}$$

We approximate the inner integral in (16), which is of the form (8), by $\mathcal{N}(\boldsymbol{x}_{t+1} \,|\, \tilde{\boldsymbol{\mu}}^{\backslash\triangleleft}, \tilde{\boldsymbol{\Sigma}}^{\backslash\triangleleft})$ by moment matching [14], for instance. Note that $\tilde{\boldsymbol{\mu}}^{\backslash\triangleleft}$ and $\tilde{\boldsymbol{\Sigma}}^{\backslash\triangleleft}$ are functions of $\boldsymbol{\mu}_t^{\backslash\triangleleft}$ and $\boldsymbol{\Sigma}_t^{\backslash\triangleleft}$. This Gaussian approximation implicitly linearizes $\mathcal{GP}_h$. Now, (16) can be computed analytically, and we obtain a Gaussian approximation $\tilde{Z}_\triangleleft = \mathcal{N}(\boldsymbol{\mu}_{t+1}^{\backslash\triangleright} \,|\, \tilde{\boldsymbol{\mu}}^{\backslash\triangleleft}, \tilde{\boldsymbol{\Sigma}}^{\backslash\triangleleft} + \boldsymbol{\Sigma}_{t+1}^{\backslash\triangleright})$ of $Z_\triangleleft$ that allows us to update the moments of $q(\boldsymbol{x}_t)$ and the message $q_\triangleleft(\boldsymbol{x}_t)$.

Forward Message Similarly, for the forward message, the projection step involves computing the partition function

$$Z_\triangleright(\boldsymbol{\mu}_t^{\backslash\triangleright}, \boldsymbol{\Sigma}_t^{\backslash\triangleright}) = \int f_\triangleright(\boldsymbol{x}_t) q_{\backslash\triangleright}(\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_t = \int f_\triangleright(\boldsymbol{x}_t)\mathcal{N}(\boldsymbol{x}_t \,|\, \boldsymbol{\mu}_t^{\backslash\triangleright}, \boldsymbol{\Sigma}_t^{\backslash\triangleright})\mathrm{d}\boldsymbol{x}_t, \tag{17}$$

$$f_\triangleright(\boldsymbol{x}_t) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) q_{\backslash\triangleleft}(\boldsymbol{x}_{t-1})\mathrm{d}\boldsymbol{x}_{t-1} = \int \mathcal{N}(\boldsymbol{x}_t \,|\, \boldsymbol{\mu}_f(\boldsymbol{x}_{t-1}), \boldsymbol{\Sigma}_f(\boldsymbol{x}_{t-1})) q_{\backslash\triangleleft}(\boldsymbol{x}_{t-1})\mathrm{d}\boldsymbol{x}_{t-1} \,,$$

where the true factor $f_\triangleright(\boldsymbol{x}_t)$ takes into account the coupling between $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$, see Fig. 1. Here, the true factor $f_\triangleright(\boldsymbol{x}_t)$ is of the form (8). We propose to approximate $f_\triangleright(\boldsymbol{x}_t)$ directly by a Gaussian $q_\triangleright(\boldsymbol{x}_t) \propto \mathcal{N}(\tilde{\boldsymbol{\mu}}^\triangleright, \tilde{\boldsymbol{\Sigma}}^\triangleright)$. This approximation implicitly linearizes $\mathcal{GP}_h$. We obtain the updated posterior $q(\boldsymbol{x}_t)$ by Gaussian multiplication, i.e., $q(\boldsymbol{x}_t) \propto q_\triangleright(\boldsymbol{x}_t) q_{\backslash\triangleright}(\boldsymbol{x}_t)$. With this approximation we do not update the forward message in context, i.e., the true factor $f_\triangleright(\boldsymbol{x}_t)$ is directly approximated instead of the product $f_\triangleright(\boldsymbol{x}_t) q_{\backslash\triangleright}(\boldsymbol{x}_t)$, which can result in suboptimal approximation.

3.3 EP Updates for General Gaussian Smoothers

We can interpret the EP computations in the context of classical Gaussian filtering and smoothing [1]. During the *forward sweep*, the marginal $q(\boldsymbol{x}_t) = q_{\backslash\triangleleft}(\boldsymbol{x}_t)$ corresponds to the filter distribution $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$. Moreover, the cavity distribution $q_{\backslash\Delta}(\boldsymbol{x}_t)$ corresponds to the time update $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t-1})$. In the *backward sweep*, the marginal $q(\boldsymbol{x}_t)$ is the smoothing distribution $p(\boldsymbol{x}_t|\boldsymbol{Z})$, incorporating the measurements of the entire time series. The mean and covariance of $\tilde{Z}_\triangleleft$ can be interpreted as the mean and covariance of the time update $p(\boldsymbol{x}_{t+1}|\boldsymbol{z}_{1:t})$.

Updating the moments of the posterior $q(\boldsymbol{x}_t)$ via the derivatives of the log-partition function recovers exactly the standard Gaussian EP updates in dynamical systems described by Qi and Minka [13]. For example, when incorporating an updated *measurement message*, the moments in (5) can also be written as $\boldsymbol{\mu}_t = \boldsymbol{\mu}_t^{\backslash\Delta} + \boldsymbol{K}(\boldsymbol{z}_t - \boldsymbol{\mu}_z^{\backslash\Delta})$ and $\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_t^{\backslash\Delta} - \boldsymbol{K}\boldsymbol{\Sigma}_t^{zx\backslash\Delta}$, respectively, where $\boldsymbol{\Sigma}_t^{xz\backslash\Delta} = \mathrm{cov}[\boldsymbol{x}_t^{\backslash\Delta}, \boldsymbol{z}_t^{\backslash\Delta}]$ and $\boldsymbol{K} = \boldsymbol{\Sigma}_t^{xz\backslash\Delta}(\boldsymbol{\Sigma}_z^{\backslash\Delta})^{-1}$. Here, $\boldsymbol{\mu}_z^{\backslash\Delta} = \mathbb{E}[g(\boldsymbol{x}_t)]$ and $\boldsymbol{\Sigma}_z^{\backslash\Delta} = \mathrm{cov}[g(\boldsymbol{x}_t)]$, where $\boldsymbol{x}_t \sim q_{\backslash\Delta}(\boldsymbol{x}_t)$. Similarly, the updated moments of $q(\boldsymbol{x}_t)$ with a new *backward message* via (5) correspond to the updates [13] $\boldsymbol{\mu}_t = \boldsymbol{\mu}_t^{\backslash\triangleleft} + \boldsymbol{L}(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_{t+1}^{\backslash\triangleleft})$ and $\boldsymbol{\Sigma}_t = \boldsymbol{\Sigma}_t^{\triangleleft} + \boldsymbol{L}(\boldsymbol{\Sigma}_{t+1} - \boldsymbol{\Sigma}_{t+1}^{\backslash\triangleleft})\boldsymbol{L}^\top$, where $\boldsymbol{L} = \mathrm{cov}[\boldsymbol{x}_t^{\backslash\triangleleft}, \boldsymbol{x}_{t+1}^{\backslash\triangleleft}](\boldsymbol{\Sigma}_{t+1}^{\backslash\triangleleft})^{-1}$. Here, we defined $\boldsymbol{\mu}_{t+1}^{\backslash\triangleleft} = \mathbb{E}[h(\boldsymbol{x}_t)]$ and $\boldsymbol{\Sigma}_{t+1}^{\backslash\triangleleft} = \mathrm{cov}[h(\boldsymbol{x}_t)]$, where $\boldsymbol{x}_t \sim q_{\backslash\triangleleft}(\boldsymbol{x}_t)$.

Table 1: Performance comparison on the synthetic data set. Lower values are better.

	EKS	EP-EKS	GPEKS	EP-GPEKS	GPADS	EP-GPADS
NLL_x	-2.04 ± 0.07	-2.17 ± 0.04	-1.67 ± 0.22	-1.87 ± 0.14	$+\mathbf{1.67 \pm 0.37}$	$\mathbf{-1.91 \pm 0.10}$
MAE_x	$0.03 \pm 2.0 \times 10^{-3}$	$0.03 \pm 2.0 \times 10^{-3}$	$0.04 \pm 4.6 \times 10^{-2}$	$0.04 \pm 4.6 \times 10^{-2}$	$\mathbf{1.79 \pm 0.21}$	$\mathbf{0.04 \pm 4 \times 10^{-3}}$
NLL_z	-0.69 ± 0.11	-0.73 ± 0.11	-0.75 ± 0.08	-0.81 ± 0.07	1.93 ± 0.28	-0.77 ± 0.07

The iterative message-passing algorithm in Alg. 1 provides an EP-based generalization and a unifying view of existing approaches for smoothing in dynamical systems, e.g., (Extended/Unscented/Cubature) Kalman smoothing and the corresponding GPDS smoothers [5]. Computing the messages via the derivatives of the approximate log-partition functions $\log \tilde{Z}_i$ recovers not only standard EP updates in dynamical systems [13], but also the standard Kalman smoothing updates [1].

Using any prediction method (e.g., unscented transformation, linearization), we can compute Gaussian approximations of (8). This influences the computation of $\log \tilde{Z}_i$ and its derivatives with respect to the moments of the cavity distribution, see (9)–(10). Hence, our message-passing formulation is also general as it includes all conceivable Gaussian filters/smoothers in (GP)DSs, solely depending on the prediction technique used.

4 Experimental Results

We evaluated our proposed EP-based message passing algorithm on three data sets: a synthetic data set, a low-dimensional simulated mechanical system with control inputs, and a high-dimensional motion-capture data set. We compared to existing state-of-the-art forward-backward smoothers in GPDSs, specifically the GPEKS [8], which is based on the expected linearization of the GP models, and the GPADS [5], which uses moment-matching. We refer to our EP generalizations of these methods as EP-GPEKS and EP-GPADS.

In all our experiments, we evaluated the inference methods using test sequences of measurements $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_T]$. We report the negative log-likelihood of predicted measurements using the observed test sequence (NLL_z). Whenever available, we also compared the inferred posterior distribution $q(\boldsymbol{X}) \approx p(\boldsymbol{X}|\boldsymbol{Z})$ of the latent states with the underlying ground truth using the average negative log-likelihood (NLL_x) and Mean Absolute Errors (MAE_x). We terminated EP after 100 iterations or when the average norms of the differences of the means and covariances of $q(\boldsymbol{X})$ in two subsequent EP iterations were smaller than 10^{-6}.

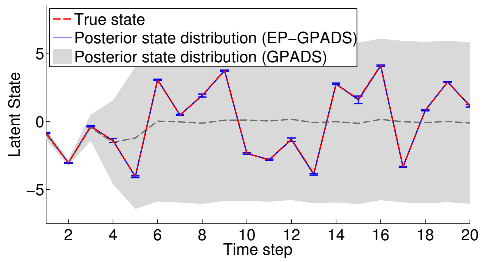
4.1 Synthetic Data

We considered the nonlinear dynamical system

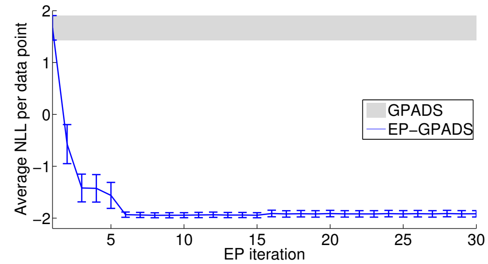
$$x_{t+1} = 4\sin(x_t) + w\,, \quad w \sim \mathcal{N}(0, 0.1^2)\,, \qquad z_t = 4\sin(x_t) + v\,, \quad v \sim \mathcal{N}(0, 0.1^2)\,.$$

We used $p(x_1) = \mathcal{N}(0, 1)$ as a prior on the initial latent state. We assumed access to the latent state and trained the dynamics and measurement GPs using 30 randomly generated points, resulting in a model with a substantial amount of posterior model uncertainty. The length of the test trajectory used was $T = 20$ time steps.

Tab. 1 reports the quality of the inferred posterior distributions of the latent state trajectories using the average NLL_x, MAE_x, and NLL_z (with standard errors), averaged over 10 independent scenarios. For this dataset, we also compared to the Extended Kalman Smoother (EKS) and an EP-iterated EKS (EP-EKS). Both inference methods make use of the known transition and measurement mappings h and g, respectively. Iterated forward-backward smoothing with EP (EP-EKS, EP-GPEKS, EP-GPADS) improved the smoothing posteriors using a single sweep only (EKS, GPEKS, GPADS). The GPADS performed poorly across all our evaluation criteria for two reasons: First, the GPs were trained using few data points, resulting in posterior distributions with a high degree of uncertainty. Second, predictive variances using moment-matching are generally conservative and increased the uncertainty even further. This uncertainty caused the GPADS to quickly lose track of the period of the state, as shown in Fig. 2(a). By iterating forward-backward smoothing using EP (EP-GPADS), the posteriors $p(\boldsymbol{x}_t|\boldsymbol{Z})$ were iteratively refined, and the latent state could be followed closely as indicated by both the small blue error bars in Fig. 2(a) and all performance measures in Tab. 1. EP smoothing typically required a small number of iterations for the inferred posterior distribution to closely track the true state, Fig. 2(b). On average, EP required fewer than 10 iterations to converge to a good solution in which the mean of the latent-state posterior closely matched the ground truth.



(a) Example trajectory distributions with 95% confidence bounds.



(b) Average NLL_x as a function of the EP iteration with twice the standard error.

Figure 2: (a) Posterior latent state distributions using EP-GPADS (blue) and the GPADS (gray). The ground truth is shown in red (dashed). The GPADS quickly loses track of the period of the state revealed by the large posterior uncertainty. EP with moment matching (EP-GPADS) in the GPDS iteratively refines the GPADS posterior and can closely follow the true latent state trajectory. (b) Average NLL_x per data point in latent space with standard errors of the posterior state distributions computed by the GPADS and the EP-GPADS as a function of EP iterations.

4.2 Pendulum Tracking

We considered a pendulum tracking problem to demonstrate GPDS inference in multidimensional settings, as well as the ability to handle control inputs. The state $\boldsymbol{x}$ of the system is given by the angle ϕ measured from being upright and the angular velocity $\dot{\phi}$. The pendulum used has a mass of $1\,\text{kg}$ and a length of $1\,\text{m}$, and random torques $u \in [-2, 2]\,\text{Nm}$ were applied for a duration $200\,\text{ms}$ (zero-order-hold control). The system noise covariance was set to $\boldsymbol{\Sigma}_w = \text{diag}(0.3^2, 0.1^2)$. The state was measured indirectly by two bearings sensors with coordinates $(x_1, y_1) = (-2, 0)$ and $(x_2, y_2) = (-0.5, -0.5)$, respectively, according to $\boldsymbol{z} = [z_1, z_2]^\top + \boldsymbol{v}\,,\ \boldsymbol{v} \sim \mathcal{N}\left(\boldsymbol{0}, \text{diag}(0.1^2, 0.05^2)\right)$ with $z_i = \arctan\left(\frac{\sin\phi - y_i}{\cos\phi - x_i}\right)$, $i = 1, 2$. We trained the GP models using 4 randomly generated trajectories of length $T = 20$ time steps, starting from an initial state distribution $p(\boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{0}, \text{diag}(\pi^2/16^2, 0.5^2))$ around the upright position. For testing, we generated 12 random trajectories starting from $p(\boldsymbol{x}_1)$.

Table 2: Performance comparison on the pendulum-swing data. Lower values are better.

	NLL_x	MAE_x	NLL_z
GPEKS	$\mathbf{-0.35 \pm 0.39}$	0.30 ± 0.02	-2.41 ± 0.047
EP-GPEKS	$\mathbf{-0.33 \pm 0.44}$	0.31 ± 0.02	-2.39 ± 0.038
GPADS	$\mathbf{-0.80 \pm 0.06}$	0.30 ± 0.02	-2.37 ± 0.042
EP-GPADS	$\mathbf{-0.85 \pm 0.05}$	0.29 ± 0.02	-2.40 ± 0.037

Tab. 2 summarizes the performance of the various inference methods. Generally, the (EP-)GPADS performed better than the (EP-)GPEKS across all performance measures. This indicates that the (EP-)GPEKS suffered from overconfident posteriors compared to (EP-)GPADS, which is especially pronounced in the degrading NLL_x values with increasing EP iterations and the relatively high standard errors. In about 20% of the test cases, the inference methods based on explicit linearization of the posterior mean function (GPEKS and EP-GPEKS) ran into numerical problems typical of linearizations [5], i.e., overconfident posterior distributions that caused numerical problems. We excluded these runs from the results in Tab. 2. The inference algorithms based on moment matching (GPADS and EP-GPADS) were numerically stable as their predictions are typically more coherent due to conservative approximations of moment matching.

4.3 Motion Capture Data

We considered motion capture data (from `http://mocap.cs.cmu.edu/`, subject 64) containing 10 trials of golf swings recorded at $120\,\text{Hz}$, which we subsampled to $20\,\text{Hz}$. After removing observation dimensions with no variability we were left with observations $\boldsymbol{z}_t \in \mathbb{R}^{56}$, which were then whitened as a pre-processing step. For trials 1–7 (403 data points), we used the GPDM [20] to learn MAP estimates of the latent states $\boldsymbol{x}_t \in \mathbb{R}^3$. These estimated latent states and their corresponding observations are used to train the GP models $\mathcal{GP}_f$ and $\mathcal{GP}_g$. Trials 8–10 were used as test

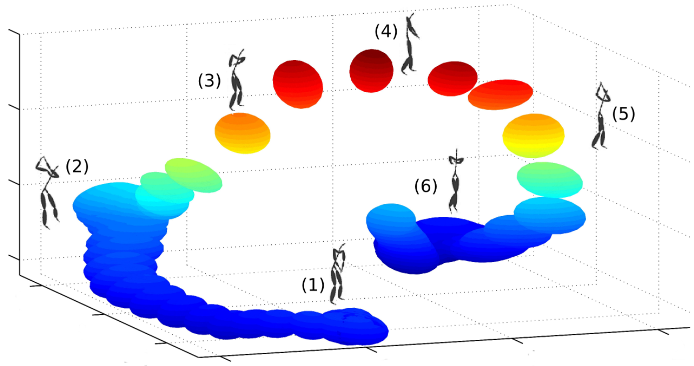


Figure 3: Latent space posterior distribution (95% confidence ellipsoids) of a test trajectory of the golf-swing motion capture data. The further the ellipsoids are separated the faster the movement.

data without ground truth labels. The GPDM [20] focuses on learning a GPDS; we are interested in good approximate inference in these models.

Fig. 3 shows the latent-state posterior distribution of a single test sequence (trial 10) obtained from the EP-GPADS. The most significant prediction errors in observed space occurred in the region corresponding to the yellow/red ellipsoids, which is a low-dimensional embedding of the motion when the golf player hits the ball, i.e., the periods of high acceleration (poses 3–5).

Tab. 3 summarizes the results of inference on the golf data set in all test trials: Iterating forward-backward smoothing by means of EP improved the inferred posterior distributions over the latent states. The posterior distributions in latent space inferred by the EP-GPEKS were tighter than the ones inferred by the EP-GPADS. The NLL_z-values suffered a bit from this overconfidence, but the predictive performance of the EP-GPADS and EP-GPEKS were similar. Generally, inference was more difficult in areas with fast movements (poses 3–5 in Fig. 3) where training data were sparse.

Table 3: Average inference performance (NLL_z, motion capture data set). Lower values are better.

Test trial	GPEKS	EP-GPEKS	GPADS	EP-GPADS
Trial 8	14.20	13.82	14.28	14.09
Trial 9	15.63	14.71	15.19	14.84
Trial 10	26.68	25.73	25.64	25.42

The computational demand the two inference methods for GPDSs we presented is vastly different. High-dimensional approximate inference in the motion capture example using moment matching (EP-GPADS) was about two orders of magnitude slower than approximate inference based on linearization of the posterior GP mean (EP-GPEKS): For updating the posterior and the messages for a single time slice, the EP-GPEKS required less than 0.5 s, the EP-GPADS took about 20 s. Hence, numerical stability and more coherent posterior inference with the EP-GPADS trade off against computational demands.

5 Conclusion

We have presented an approximate message passing algorithm based on EP for improved inference and Bayesian state estimation in GP dynamical systems. Our message-passing formulation generalizes current inference methods in GPDSs to iterative forward-backward smoothing. This generalization allows for improved predictions and comprises existing methods for inference in the wider theory for dynamical systems as a special case. Our new inference approach makes the full power of the GPDS model available for the study of complex time-series data. Future work includes investigating alternatives to linearization and moment matching when computing messages, and the more general problem of learning in Gaussian process dynamical systems.

Acknowledgements

We thank Zhikun Wang for helping with the motion capture experiment and Jan Peters for valuable discussions. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement #270327 (ComPLACS) and the Canadian Institute for Advanced Research (CIFAR).

References

[1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Dover Publications, 2005.

[2] A. Damianou, M. K. Titsias, and N. D. Lawrence. Variational Gaussian Process Dynamical Systems. In *Advances in Neural Information Processing Systems*. 2011.

[3] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck. Analytic Moment-based Gaussian Process Filtering. In *Proceedings of the 26th International Conference on Machine Learning*, pages 225–232. Omnipress, 2009.

[4] M. P. Deisenroth and S. Mohamed. Expectation Propagation in Gaussian Process Dynamical Systems: Extended Version, 2012. `http://arxiv.org/abs/1207.2940`.

[5] M. P. Deisenroth, R. Turner, M. Huber, U. D. Hanebeck, and C. E. Rasmussen. Robust Filtering and Smoothing with Gaussian Processes. *IEEE Transactions on Automatic Control*, 2012.

[6] T. Heskes and O. Zoeter. Expectation Propagation for Approximate Inference in Dynamic Bayesian Networks. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 216–233, 2002.

[7] S. J. Julier and J. K. Uhlmann. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, 92(3):401–422, March 2004.

[8] J. Ko and D. Fox. GP-BayesFilters: Bayesian Filtering using Gaussian Process Prediction and Observation Models. *Autonomous Robots*, 27(1):75–90, 2009.

[9] M. Kuss and C. E. Rasmussen. Assessing Approximate Inference for Binary Gaussian Process Classification. *Journal of Machine Learning Research*, 6:1679–1704, 2005.

[10] T. P. Minka. Expectation Propagation for Approximate Bayesian Inference. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufman Publishers, 2001.

[11] T. P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[12] T. P. Minka. EP: A Quick Reference. 2008.

[13] Y. Qi and T. Minka. Expectation Propagation for Signal Detection in Flat-Fading Channels. In *Proceedings of the IEEE International Symposium on Information Theory*, 2003.

[14] J. Quiñonero-Candela, A. Girard, J. Larsen, and C. E. Rasmussen. Propagation of Uncertainty in Bayesian Kernel Models—Application to Multiple-Step Ahead Forecasting. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 701–704, 2003.

[15] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[16] M. W. Seeger. Expectation Propagation for Exponential Families. Technical report, University of California Berkeley, 2005.

[17] M. W. Seeger. Bayesian Inference and Optimal Design for the Sparse Linear Model. *Journal of Machine Learning Research*, 9:759–813, 2008.

[18] M. Toussaint and C. Goerick. *From Motor Learning to Interaction Learning in Robotics*, chapter A Bayesian View on Motor Control and Planning, pages 227–252. Springer-Verlag, 2010.

[19] R. Turner, M. P. Deisenroth, and C. E. Rasmussen. State-Space Inference and Learning with Gaussian Processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume JMLR: W&CP 9, pages 868–875, 2010.

[20] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian Process Dynamical Models for Human Motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008.