# Analytic Moment-based Gaussian Process Filtering

**Marc Peter Deisenroth** MPD37@CAM.AC.UK
Computational and Biological Learning Lab (CBL), University of Cambridge, UK
Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Universität Karlsruhe (TH), Germany

**Marco F. Huber** MARCO.HUBER@IEEE.ORG
**Uwe D. Hanebeck** UWE.HANEBECK@IEEE.ORG
Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Universität Karlsruhe (TH), Germany

## Abstract

We propose an analytic moment-based filter for nonlinear stochastic dynamic systems modeled by Gaussian processes. Exact expressions for the expected value and the covariance matrix are provided for both the prediction step and the filter step, where an additional Gaussian assumption is exploited in the latter case. Our filter does not require further approximations. In particular, it avoids finite-sample approximations. We compare the filter to a variety of Gaussian filters, that is, the EKF, the UKF, and the recent GP-UKF proposed by Ko et al. (2007).

## 1. Introduction

Recursively estimating the internal state of a nonlinear dynamic system from noisy observations is a common problem in many technical applications, for instance, in sensor networks, robotics, or signal processing. Exact Bayesian solutions in closed form, however, can be found only in a few special cases. For example, for linear Gaussian systems, the Kalman filter (1960) is exact.

For most nonlinear cases, approximate methods are required to obtain efficient analytic/closed-form solutions. A variety of approximate Gaussian filters has been proposed in the past. For example, the *Extended Kalman Filter (EKF)* linearizes the transition and measurement functions by means of a Taylor series expansion and applies the Kalman filter to propagate full densities through them (Simon, 2006). Instead of approximating functions, the *Unscented Kalman Filter (UKF)* by Julier and Uhlmann (2004) uses a deterministic sampling approach to approximate distributions, while using the original nonlinear functions to propagate them. This approach is considered equivalent to stochastic linearization (Lefebvre et al., 2005).

Both the EKF and the UKF employ a known parametric model of the transition dynamics and the measurement function. However, lack of modeling accuracy as well as difficulties in the identification of the noise and the model parameters are typically ignored. Instead of a parametric description, Ko et al. (2007) and Ko and Fox (2008) derive the GP-EKF and the GP-UKF by incorporating probabilistic non-parametric Gaussian process (GP) models of the transition dynamics and the measurement function into the EKF and UKF. Model uncertainty can explicitly be incorporated into the prediction and the filtering processes, which is usually not the case for filtering approaches based on a parametric model. Moreover, they train the GP models offline using ground truth of the hidden states.

In this paper, we derive a Gaussian filter algorithm for nonlinear dynamic systems, where the transition dynamics and the observation map are described by GP models. In contrast to finite-sample approximations (UKF, GP-UKF) of the prior and the predictive distribution, we propagate full densities by exploiting specific properties of GP models. Furthermore, we approximate the predictive distribution by a Gaussian with the exact mean and the exact covariance matrix, which can be computed analytically using results from (Quiñonero-Candela et al., 2003). This approximation, on which our filter is based, is known as *moment matching*. Hence, the proposed filter, which we call GP-ADF, is an efficient form of an *Assumed Density Filter* (ADF) (Maybeck, 1979).

The paper is organized as follows: In Section 2, the

Appearing in *Proceedings of the 26$^{th}$ International Conference on Machine Learning*, Montreal, Canada, 2009. 

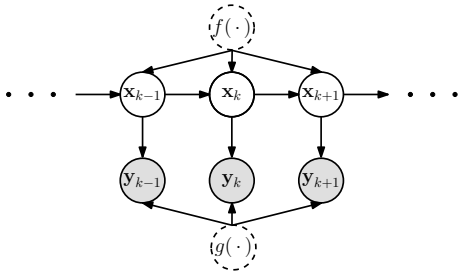



*Figure 1.* Graphical model of a nonlinear dynamic system. The shaded nodes $\mathbf{y}_i$ are observed variables, the other nodes are latent variables. The dependencies between variables are given by the arrows. The dashed nodes represent functions $f$ and $g$, which can either be observed or latent depending on the model used.

models under consideration are reviewed and the prediction and filtering problems are stated. A survey of related work is given in Section 3. In Section 4, we provide background on prediction with GP models. The GP-ADF itself is derived in Section 5. Simulation results are presented in Section 6. In Section 7, we discuss properties of the filter algorithm. Section 8 summarizes the paper and gives a survey of future work.

## 2. Model and Problem Statement

We consider discrete-time dynamic systems with transition dynamics given by

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}\,, \tag{1}$$

where $f$ is a possibly nonlinear function and $\mathbf{w} \sim \mathcal{N}\big(\mathbf{0}, \boldsymbol{\Sigma}_w\big)$ is white, additive Gaussian system noise with uncorrelated dimensions. The $D$-dimensional continuous-valued state is denoted by $\mathbf{x}$, and $k$ is a discrete time index. Furthermore, we consider observations/measurements

$$\mathbf{y}_k = g(\mathbf{x}_k) + \mathbf{v}\,, \tag{2}$$

where $g$ is a (non)linear function, $\mathbf{y}_k$ is the $E$-dimensional observation, and $\mathbf{v} \sim \mathcal{N}\big(\mathbf{0}, \boldsymbol{\Sigma}_v\big)$ is white, additive Gaussian measurement noise with uncorrelated dimensions.

Figure 1 is a graphical model of the considered nonlinear dynamic system. We included dashed "function nodes" for $f$ and $g$. The function node is shaded if and only if the function is explicitly known.

We assume a prior on $\mathbf{x}_0$ and aim to determine probability distributions of the hidden state $\mathbf{x}_k$ based on all observations $\mathbf{y}_{1:k}$. We distinguish between prediction (moving from $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$) and filtering (going from $\mathbf{y}_k$ to $\mathbf{x}_k$). Typically, prediction and filtering alternate.

*Table 1.* Classification of Gaussian filter methods.

| | SAMPLES | FULL DENSITY |
|---|---|---|
| $f, g$ : KNOWN | UKF | EKF |
| $f, g$ : UNKNOWN | GP-UKF | GP-ADF |

**Prediction Step** When we predict, we determine the distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ of the hidden state $\mathbf{x}_k$, where the result of the previous filter result $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ serves as the prior. Bayes' law yields

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})\, \mathrm{d}\mathbf{x}_{k-1} \tag{3}$$

by averaging over $\mathbf{x}_{k-1}$. Often, the involved integral and the multiplication cannot be solved analytically and require approximate methods.

**Filter Update** The filter update determines the distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ of the hidden state $\mathbf{x}_k$ based on collected observations from all previous and the current time steps. Bayes' law yields the *filter update*

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}\,. \tag{4}$$

The likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$ is defined through the measurement equation (2), the prior $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ is the result of the preceding prediction step (3). Often, the filter update (4) does not admit a closed-form solution since the integral in the normalization constant

$$p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k-1})\, \mathrm{d}\mathbf{x}_k$$

and the density multiplication in the numerator in equation (4) cannot be computed exactly.

## 3. Related Work

Table 1 classifies the Gaussian filter methods discussed in this paper. We present density representation against knowledge of the parameterization of the transition dynamics $f$ and the observation function $g$.

The UKF by Julier and Uhlmann (2004) deterministically chooses *sigma points* that capture the moments of the state distribution and maps them using a known parameterization of the original nonlinear functions $f$ and $g$, respectively. The transformed sigma points provide a *finite-sample approximation* of the true predictive distribution. The UKF is not moment preserving.

Ko et al. (2007) and Ko and Fox (2008) propose GPs to model the transition and observation functions $f$ and $g$. GPs are incorporated into standard filters, such as the UKF. The resulting GP-UKF maps the

UKF sigma points through the GP models instead of the parametric functions $f$ and $g$. Like in the UKF, all considered distributions are described by a finite number of samples and the GP-UKF is not moment preserving. In the limit of perfect GP models, that is, the posterior mean functions match the latent functions $f$ and $g$ and the posterior uncertainty is zero, both the UKF and the GP-UKF are equivalent.

Like Ko et al. (2007), we utilize GPs to model $f$ and $g$. In contrast to both the UKF and the GP-UKF, our proposed GP-ADF does not propagate samples from a Gaussian, but the full Gaussian *density*. Our GP-ADF heavily exploits the fact that the true moments of the GP predictive distribution can be computed in closed form. The predictive distribution is approximated by a Gaussian with the exact predictive mean and the exact predictive covariance (moment matching). Therefore, GP-ADF is a form of Assumed Density Filtering (ADF), which has previously been introduced by Maybeck (1979), Boyen and Koller (1998), and Opper (1998). Furthermore, to compute the first two predictive moments, GP-ADF takes the uncertainty about the latent functions $f, g$ into account. GP-ADF is moment preserving.

The UKF propagates samples through known or directly accessible functions, that is, the nodes for $f$ and $g$ in Figure 1 are shaded. A classical ADF and the EKF propagate entire densities, but they also require known functions $f$ and $g$. GP-UKF and GP-ADF are based on probabilistic models of the latent functions. Hence, the nodes $f, g$ in Figure 1 are unshaded. The filters differ in the propagation method: GP-UKF propagates a finite-sample approximation of a Gaussian, whereas GP-ADF propagates the full Gaussian.

Ghahramani and Roweis (1999) discuss the EKF for nonlinear dynamic systems, where the transition dynamics and the measurement function are modeled by a radial basis function network, a parametric approximation with limited expressiveness.

## 4. Gaussian Processes

Following the book by Rasmussen and Williams (2006), we briefly introduce the notation and standard prediction models for Gaussian processes, which are used to infer a latent function $h$ from (noisy) observations $y_i = h(\mathbf{x}_i) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$. A GP is completely specified by a mean function $m(\cdot)$ and a positive semidefinite covariance function $k(\cdot, \cdot)$, also called a *kernel*. We write $h \sim \mathcal{GP}$ if the latent function $h$ is GP distributed. Throughout this paper, we consider the squared exponential (SE) kernel

$$k(\mathbf{x}, \mathbf{x}') = \alpha^2 \exp\big(-\tfrac{1}{2}(\mathbf{x}-\mathbf{x}')^\top \boldsymbol{\Lambda}^{-1}(\mathbf{x}-\mathbf{x}')\big)\,, \quad (5)$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of the characteristic length-scales of the SE kernel, and $\alpha^2$ is the variance of the latent function $h$. The posterior predictive distribution of the function value $h_* = h(\mathbf{x}_*)$ for an arbitrary test input $\mathbf{x}_*$ is Gaussian with mean and variance

$$m_h(\mathbf{x}_*) = \mathrm{E}_h[h_*] = \mathbf{k}_*^\top (\mathbf{K} + \sigma_\varepsilon^2 \mathbf{I})^{-1}\mathbf{y} = \mathbf{k}_*^\top \boldsymbol{\beta}\,, \quad (6)$$

$$\sigma_h^2(\mathbf{x}_*) = \mathrm{var}_h[h_*] = k_{**} - \mathbf{k}_*^\top (\mathbf{K} + \sigma_\varepsilon^2 \mathbf{I})^{-1}\mathbf{k}_*\,, \quad (7)$$

respectively, with $\mathbf{k}_* \coloneqq k(\mathbf{X}, \mathbf{x}_*)$, $k_{**} \coloneqq k(\mathbf{x}_*, \mathbf{x}_*)$, $\boldsymbol{\beta} \coloneqq (\mathbf{K} + \sigma_\varepsilon^2 \mathbf{I})^{-1}\mathbf{y}$, and where $\mathbf{K}$ is the kernel matrix with $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Moreover, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ are the training inputs, and $\mathbf{y} = [y_1, \ldots, y_n]^\top$ are the corresponding training targets (observations).

### 4.1. Predictions for Uncertain Inputs

We review results by Rasmussen and Ghahramani (2003), Quiñonero-Candela et al. (2003), and Kuss (2006) of how to predict with GPs when the test input $\mathbf{x}_*$ is uncertain, which means that it has a probability distribution.

Consider the problem of predicting a function value $h(\mathbf{x}_*)$ for an *uncertain* test input $\mathbf{x}_* \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $h \sim \mathcal{GP}$ with an SE kernel $k_h$. The prediction problem corresponds to seeking the distribution

$$p(h(\mathbf{x}_*)|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \int p(h(\mathbf{x}_*)|\mathbf{x}_*) p(\mathbf{x}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma})\, \mathrm{d}\mathbf{x}_*\,. \quad (8)$$

The mean and variance of the GP predictive distribution for $p(h(\mathbf{x}_*)|\mathbf{x}_*)$ are given in equations (6) and (7), respectively. For the SE kernel, we can compute the mean $\mu_*$ and the variance $\sigma_*^2$ of equation (8) in closed form. The mean $\mu_*$ is

$$\begin{aligned}\mu_* &= \mathrm{E}_{\mathbf{x}_*}[\mathrm{E}_h[h(\mathbf{x}_*)]|\boldsymbol{\mu}, \boldsymbol{\Sigma}] \overset{(6)}{=} \mathrm{E}_{\mathbf{x}_*}[m_h(\mathbf{x}_*)|\boldsymbol{\mu}, \boldsymbol{\Sigma}] \\ &= \int m_h(\mathbf{x}_*)\mathcal{N}\big(\mathbf{x}_* \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}\big)\, \mathrm{d}\mathbf{x}_* = \boldsymbol{\beta}^\top \mathbf{l} \end{aligned} \quad (9)$$

with $\mathbf{l} = [l_1, \ldots, l_n]^\top$, where

$$\begin{aligned} l_i = \int k_h(\mathbf{x}_i, \mathbf{x}_*) p(\mathbf{x}_*)\, \mathrm{d}\mathbf{x}_* &= \alpha^2 |\boldsymbol{\Sigma}\boldsymbol{\Lambda}^{-1} + \mathbf{I}|^{-\frac{1}{2}} \\ &\times \exp\big(-\tfrac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu})^\top (\boldsymbol{\Sigma} + \boldsymbol{\Lambda})^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\big) \end{aligned}$$

is an expectation of $k_h(\mathbf{x}_i, \mathbf{x}_*)$ with respect to $\mathbf{x}_*$. Note that the predictive mean explicitly depends on the mean and covariance of the distribution of the input $\mathbf{x}_*$. The variance $\sigma_*^2$ of $p(h(\mathbf{x}_*)|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is

$$\begin{aligned}\sigma_*^2 &= \mathrm{E}_{\mathbf{x}_*}[m_h(\mathbf{x}_*)^2|\boldsymbol{\mu}, \boldsymbol{\Sigma}] + \mathrm{E}_{\mathbf{x}_*}[\sigma_h^2(\mathbf{x}_*)|\boldsymbol{\mu}, \boldsymbol{\Sigma}] \\ &\quad - \mathrm{E}_{\mathbf{x}_*}[m_h(\mathbf{x}_*)|\boldsymbol{\mu}, \boldsymbol{\Sigma}]^2 \\ &= \boldsymbol{\beta}^\top \tilde{\mathbf{L}} \boldsymbol{\beta} + \alpha^2 - \mathrm{tr}\big((\mathbf{K} + \sigma_\varepsilon^2 \mathbf{I})^{-1}\tilde{\mathbf{L}}\big) - \mu_*^2\,, \end{aligned} \quad (10)$$

where $\mathrm{tr}(\,\cdot\,)$ is the trace and

$$\tilde{L}_{ij} = \frac{k_h(\mathbf{x}_i, \boldsymbol{\mu}) k_h(\mathbf{x}_j, \boldsymbol{\mu})}{|2\boldsymbol{\Sigma}\boldsymbol{\Lambda}^{-1} + \mathbf{I}|^{\frac{1}{2}}} \times \exp\left((\tilde{\mathbf{z}}_{ij} - \boldsymbol{\mu})^\top (\boldsymbol{\Sigma} + \tfrac{1}{2}\boldsymbol{\Lambda})^{-1} \boldsymbol{\Sigma} \boldsymbol{\Lambda}^{-1} (\tilde{\mathbf{z}}_{ij} - \boldsymbol{\mu})\right) \tag{11}$$

with $\tilde{\mathbf{z}}_{ij} \coloneqq \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$. Like the predicted mean in equation (9), the predictive variance explicitly depends on the mean and the covariance matrix of the input distribution. We approximate the predictive distribution $p(h(\mathbf{x}_*)|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by a Gaussian $\mathcal{N}(\mu_*, \sigma_*^2)$ that exactly matches the predictive mean and variance.

### 4.2. Multivariate Predictions

We extend the previous results to the case of a latent function $h : \mathbb{R}^D \to \mathbb{R}^E$, $h \sim \mathcal{GP}$ with an SE kernel $k_h$. We train $E$ GP models independently using the same training inputs $\mathbf{X}$, but different training targets $\mathbf{y}_a = [y_1^a, \ldots, y_n^a]^\top$, $a = 1, \ldots, E$. This model implies that any two target dimensions are conditionally independent given the input. Intuitively, different target dimensions can only "communicate" via the input.

For a *deterministically* given input $\mathbf{x}_*$, the mean and the variance of a predicted function value for each target dimension are given by equations (6) and (7), respectively. The predicted covariance matrix is diagonal since we assume that the predicted target dimensions are conditionally independent given the input.

For an *uncertain* input $\mathbf{x}_* \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the predictive mean vector $\boldsymbol{\mu}_*$ of $p(h(\mathbf{x}_*)|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the collection of all $E$ individual predicted means $\mu_*^a$ given by equation (9). The target dimensions, however, co-vary and the corresponding predictive covariance matrix

$$\boldsymbol{\Sigma}_*|\boldsymbol{\mu}, \boldsymbol{\Sigma} = \begin{bmatrix} \mathrm{var}[h_1^*|\boldsymbol{\mu}, \boldsymbol{\Sigma}] & \ldots & \mathrm{cov}[h_1^*, h_E^*|\boldsymbol{\mu}, \boldsymbol{\Sigma}] \\ \vdots & \ddots & \vdots \\ \mathrm{cov}[h_E^*, h_1^*|\boldsymbol{\mu}, \boldsymbol{\Sigma}] & \ldots & \mathrm{var}[h_E^*|\boldsymbol{\mu}, \boldsymbol{\Sigma}] \end{bmatrix}$$

is no longer diagonal. The variances on the diagonal are the predictive variances of the individual target dimensions given by equation (10). The cross-covariances are given by

$$\mathrm{cov}[h_a^*, h_b^*|\boldsymbol{\mu}, \boldsymbol{\Sigma}] = \mathrm{E}_{h,\mathbf{x}_*}[h_a^* h_b^*|\boldsymbol{\mu}, \boldsymbol{\Sigma}] - \mu_*^a \mu_*^b \,,$$

where $a, b \in \{1, \ldots, E\}$ and $h_a^* \coloneqq h_a(\mathbf{x}_*)$. We rewrite

$$\begin{aligned} \mathrm{E}_{h,\mathbf{x}_*}[h_a^* h_b^*|\boldsymbol{\mu}, \boldsymbol{\Sigma}] &= \iint h_a^* h_b^* p(h_a, h_b|\mathbf{x}_*) p(\mathbf{x}_*) \,\mathrm{d}h \,\mathrm{d}\mathbf{x}_* \\ &\overset{(9)}{=} \int m_h^a(\mathbf{x}_*) m_h^b(\mathbf{x}_*) p(\mathbf{x}_*) \,\mathrm{d}\mathbf{x}_* \,. \end{aligned}$$

With $\boldsymbol{\beta}_a \coloneqq (\mathbf{K}_a + \sigma_{\varepsilon_a}^2 \mathbf{I})^{-1} \mathbf{y}_a$ in equation (6), we obtain

$$\begin{aligned} &\mathrm{E}_{h,\mathbf{x}_*}[h_a^* h_b^*|\boldsymbol{\mu}, \boldsymbol{\Sigma}] \\ &= \int m_h^a(\mathbf{x}_*) m_h^b(\mathbf{x}_*) p(\mathbf{x}_*) \,\mathrm{d}\mathbf{x}_* \\ &\overset{(6)}{=} \int k_h^a(\mathbf{x}_*, \mathbf{X}) \boldsymbol{\beta}_a \, k_h^b(\mathbf{x}_*, \mathbf{X}) \boldsymbol{\beta}_b p(\mathbf{x}_*) \,\mathrm{d}\mathbf{x}_* \\ &= \boldsymbol{\beta}_a^\top \underbrace{\int k_h^a(\mathbf{X}, \mathbf{x}_*) \, k_h^b(\mathbf{x}_*, \mathbf{X}) p(\mathbf{x}_*) \,\mathrm{d}\mathbf{x}_*}_{=: \mathbf{L}} \boldsymbol{\beta}_b \,. \end{aligned}$$

Furthermore, with $\mathbf{R} \coloneqq (\boldsymbol{\Lambda}_a^{-1} + \boldsymbol{\Lambda}_b^{-1})^{-1} + \boldsymbol{\Sigma}$,

$$\begin{aligned} L_{ij} = \alpha_a^2 \alpha_b^2 &|(\boldsymbol{\Lambda}_a^{-1} + \boldsymbol{\Lambda}_b^{-1})\boldsymbol{\Sigma} + \mathbf{I}|^{-\frac{1}{2}} \\ &\times \exp\left(-\tfrac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top (\boldsymbol{\Lambda}_a + \boldsymbol{\Lambda}_b)^{-1} (\mathbf{x}_i - \mathbf{x}_j)\right) \\ &\times \exp\left(-\tfrac{1}{2}(\mathbf{z}_{ij} - \boldsymbol{\mu})^\top \mathbf{R}^{-1} (\mathbf{z}_{ij} - \boldsymbol{\mu})\right), \\ \mathbf{z}_{ij} \coloneqq \boldsymbol{\Lambda}_b (\boldsymbol{\Lambda}_a + \boldsymbol{\Lambda}_b)^{-1} &\mathbf{x}_i + \boldsymbol{\Lambda}_a (\boldsymbol{\Lambda}_a + \boldsymbol{\Lambda}_b)^{-1} \mathbf{x}_j \,. \end{aligned} \tag{12}$$

Note that $\mathbf{L}$ equals $\tilde{\mathbf{L}}$ in equation (11) if $a = b$.

With these results, the first two moments $\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*$ of $p(h(\mathbf{x}_*)|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be exactly determined.

## 5. GP-ADF: Assumed Density Filtering with Gaussian Processes

We assume that the transition dynamics $f$ and the measurement function $g$ in equations (1) and (2) are either not known or no longer accessible. Thus, we use models of the latent functions. We will model both functions by the GPs $\mathcal{GP}_f$ and $\mathcal{GP}_g$ with SE kernels $k_f$ and $k_g$, respectively. We assume that we have access to ground truth observations of the hidden state during training.[1] In the following, we show how to exploit these GP models for assumed density filtering and derive the GP-ADF. We closely follow the steps in Section 2.

### 5.1. Prediction Step ($\mathbf{x}_{k-1} \to \mathbf{x}_k$)

We compute the predictive distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ in equation (3). Using $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$, the result of the preceding filter step, as a Gaussian prior on $\mathbf{x}_{k-1}$, we predict the outcome of $f$ for uncertain inputs according to Section 4.2 by treating $\mathbf{x}_{k-1}$ as $\mathbf{x}_*$ and $f$ as $h$. Note that the transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is exactly Gaussian due to $\mathcal{GP}_f$. By integrating out $\mathbf{x}_{k-1}$ using equation (3), we determine the first two moments $\boldsymbol{\mu}_k^p$ and $\mathbf{C}_k^p$ of the predictive distribution exactly and approximate $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ by $\mathcal{N}(\boldsymbol{\mu}_k^p, \mathbf{C}_k^p)$.[2]

[1] This can be described by the graphical model in Figure 1, where the states $\mathbf{x}_\tau$ are observed (shaded), and the index $\tau$ runs from $-n$ to $-1$.

[2] We write $\boldsymbol{\mu}_k^p$ and $\mathbf{C}_k^p$ to indicate a one-step ahead prediction from time step $k-1$ to $k$ given $\mathbf{y}_{1:k-1}$.

### 5.2. Filter Update ($\mathbf{y}_k \rightarrow \mathbf{x}_k$)

Now, let us consider the actual filter update at time step $k$. The goal is to determine $p(\mathbf{x}_k|\mathbf{y}_{1:k})$. The preceding prediction result $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \approx \mathcal{N}(\boldsymbol{\mu}_k^p, \mathbf{C}_k^p)$ serves as the prior on $\mathbf{x}_k$ and will be combined with the recent observation $\mathbf{y}_k$ to determine the filter update (4) of the hidden state $\mathbf{x}_k$.

First, we determine the joint distribution

$$p(\mathbf{x}_k, \mathbf{y}_k|\mathbf{y}_{1:k-1}) = p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})\,. \qquad (13)$$

The GP measurement model $\mathcal{GP}_g$ yields an exact Gaussian likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$, which is combined with the Gaussian prior $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$, to obtain an approximate Gaussian predictive distribution $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) \approx \mathcal{N}(\boldsymbol{\mu}_k^y, \mathbf{C}_k^y)$. Note that $\boldsymbol{\mu}_k^y$ and $\mathbf{C}_k^y$ are the exact moments of the predictive distribution, which can be computed analytically using the results from Section 4.2 by treating $\mathbf{x}_k$ as $\mathbf{x}_*$ and $g$ as $h$.[3]

To approximate the joint distribution $p(\mathbf{x}, \mathbf{y})$ by a Gaussian distribution[4], we compute the cross terms $\mathbf{C}_{xy} = \mathrm{E}_{\mathbf{x},g}[\mathbf{x}\,\mathbf{y}^\top] - \boldsymbol{\mu}_k^p(\boldsymbol{\mu}_k^y)^\top$ of the joint covariance

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_k^p & \mathbf{C}_{xy} \\ \mathbf{C}_{xy}^\top & \mathbf{C}_k^y \end{bmatrix}.$$

For the unknown values $\mathrm{E}_{\mathbf{x},g}[\mathbf{x}\,y^a]$, we obtain

$$\begin{aligned}
&\mathrm{E}_{\mathbf{x},g_a}[\mathbf{x}\,y^a] = \mathrm{E}_{\mathbf{x},g_a}[\mathbf{x}\,(g_a(\mathbf{x}) + \mathbf{v})] = \mathrm{E}_{\mathbf{x},g_a}[\mathbf{x}\,g_a(\mathbf{x})] \\
&= \int \mathbf{x} \underbrace{\left( \int g_a(\mathbf{x}) p(g_a|\mathbf{x})\,\mathrm{d}g_a \right)}_{=\mathrm{E}_{g_a}[g_a(\mathbf{x})|\mathbf{x}] = m_g^a(\mathbf{x})} p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&\overset{(6)}{=} \int \mathbf{x} \left( \sum_{i=1}^n \beta_i^a\, k_g^a(\mathbf{x}, \mathbf{x}_i) \right) p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \sum_{i=1}^n \beta_i^a \int \mathbf{x}\, c_1\, \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \boldsymbol{\Lambda}_a) \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^p, \mathbf{C}_k^p)\,\mathrm{d}\mathbf{x}
\end{aligned}$$

for each target dimension $a = 1, \ldots, E$. Here, $c_1^{-1}$ is the normalization constant of the unnormalized SE kernel $k_g^a$. Note that $\mathbf{x}_i$, $i = 1, \ldots, n$, are the training inputs of $\mathcal{GP}_g$. The product of the two Gaussians results in a new (unnormalized) Gaussian, the normalization constant of which is denoted by $c_2^{-1}$. The mean of this new Gaussian is a function of $\mathbf{x}_i$ and $\boldsymbol{\mu}_k^p$ and denoted by $\boldsymbol{\psi}(\mathbf{x}_i, \boldsymbol{\mu}_k^p)$. Hence, we finally obtain

$$\mathrm{E}_{\mathbf{x},g}[\mathbf{x}\,y^a] = c_1 c_2^{-1} \sum_{i=1}^n \beta_i^a \boldsymbol{\psi}(\mathbf{x}_i, \boldsymbol{\mu}_k^p)\,,\ a = 1, \ldots, E\,,$$

and the covariance matrix $\mathbf{C}$ is completely determined.

Second and finally, the joint Gaussian distribution $p(\mathbf{x}_k, \mathbf{y}_k|\mathbf{y}_{1:k-1}) = \mathcal{N}([(\boldsymbol{\mu}_k^p)^\top, (\boldsymbol{\mu}_k^y)^\top]^\top, \mathbf{C})$ leads to the actual filter update

$$\begin{aligned}
p(\mathbf{x}_k|\mathbf{y}_{1:k}) &= \mathcal{N}\big(\mathbf{x}_k \,|\, \boldsymbol{\mu}_k^e, \mathbf{C}_k^e\big)\,, \qquad (14) \\
\boldsymbol{\mu}_k^e &= \boldsymbol{\mu}_k^p + \mathbf{C}_{xy}(\mathbf{C}_k^y)^{-1}(\mathbf{y}_k - \boldsymbol{\mu}_k^y)\,, \\
\mathbf{C}_k^e &= \mathbf{C}_k^p - \mathbf{C}_{xy}(\mathbf{C}_k^y)^{-1}\mathbf{C}_{xy}^\top\,.
\end{aligned}$$

### 5.3. Assumptions and Computational Complexity

For performing prediction and filtering in closed form, we employ two approximations: First, if the input $\mathbf{x}_*$ is Gaussian distributed, we approximate the true predictive distributions $f(\mathbf{x}_*)$ and $g(\mathbf{x}_*)$ by a Gaussian with the exact mean and covariance. Second, the assumption that the joint distribution (13) is Gaussian, is only true if there is a linear relationship between $\mathbf{x}$ and $\mathbf{y}$. Otherwise, it is an approximation.

No sampling or finite-sample approximations are required in GP-ADF. In contrast to the UKF or the GP-UKF, the GP-ADF propagates *densities* instead of samples from them, which will allow for gradient-based parameter learning in nonlinear dynamic systems.

The computational complexity of predicting and filtering (after training the GPs) is $\mathcal{O}(E^3) + \mathcal{O}(DE^2n^2)$ due to the inversion of the predicted covariance matrices in equation (14), and the computation of the $\mathbf{L}$-matrix (12) for the predictive covariance matrix. Here, $D$ and $E$ are the dimensionalities of the training inputs and the training targets, respectively, and $n$ is the size of the GP training set. Classical filters, such as the EKF or the UKF, scale in $\mathcal{O}(E^3)$ computations.

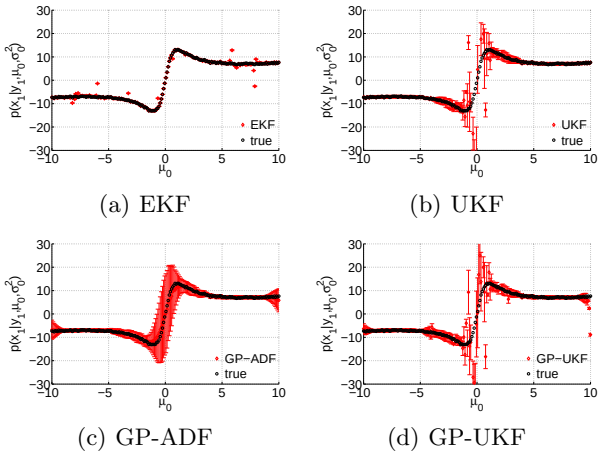
## 6. Results

We assess filter performances for a 1D example with a single filter step and a time-series in a 2D example. The GP-UKF and the GP-ADF use the same models for the transition and observation functions. The UKF and EKF always have access to the true underlying functions and noise models. We implemented the UKF and the GP-UKF as described by (Ko et al., 2007).[5]

[3] In the following paragraph, we will implicitly assume that all variables are conditioned on the previous observations $\mathbf{y}_{1:k-1}$. Moreover, we will omit the time index $k$ for brevity and clarity reasons. For example, $p(\mathbf{x}_k, \mathbf{y}_k|\mathbf{y}_{1:k-1})$ will be denoted by $p(\mathbf{x}, \mathbf{y})$.

[4] This approximation also appears in standard Gaussian filters, such as the UKF by (Julier & Uhlmann, 2004).

[5] The UKF and GP-UKF implementations are based on Nando de Freitas' UPF software available at `http://www.cs.ubc.ca/~nando/software`. The GP-ADF code will be publicly available at `http://mlg.eng.cam.ac.uk/marc`.




(a) EKF (b) UKF

(c) GP-ADF (d) GP-UKF

*Figure 2.* True hidden states (black) and filter distributions (red) for EKF, UKF, GP-ADF, and GP-UKF. The $x$-axis shows $\mu_0$, the mean value of $p(x_0^{(i)})$, the $y$-axis is the filtered distribution $p(x_1^{(i)}|y_1^{(i)}, \mu_0^{(i)}, \sigma_0^2)$ of the hidden state. The error bars show twice the standard deviations of the filtered state distributions. The filtered state distributions of the EKF, UKF, and the GP-UKF suffer from occasional inconsistencies that do not explain the true state at all. In contrast, GP-ADF is always consistent.

### 6.1. 1D Example

We consider the one-dimensional nonlinear problem

$$x_{k+1} = \tfrac{1}{2}x_k + \tfrac{25\,x_k}{1+x_k^2} + w\,, \quad w \sim \mathcal{N}\big(0, 0.2^2\big)\,,$$
$$y_k = 5\,\sin(2\,x_k) + v\,, \quad v \sim \mathcal{N}\big(0, 0.01^2\big)\,,$$

which is similar to the growth model by Kitagawa (1996). We randomly distributed 100 points in $[-10, 10]$ to train $\mathcal{GP}_f$ and $\mathcal{GP}_g$. The prior on $x_0$ is Gaussian with mean $\mu_0 \in [-10, 10]$ and variance $\sigma_0^2 = 0.5^2$. For 200 independent pairs $(x_0^{(i)}, y_1^{(i)})$ of states and observations of the successor states, we assess the performance of a single filter step of four filters, the EKF, the UKF, the GP-UKF, and the GP-ADF. Figure 2 shows a typical realization of the filtered state distributions. We evaluate the performance of the filters using two performance measures, the Mahalanobis distance

$$M_x = \sqrt{(\mathbf{x}_{\text{true}} - \boldsymbol{\mu}_k^e)^\top (\mathbf{C}_k^e)^{-1} (\mathbf{x}_{\text{true}} - \boldsymbol{\mu}_k^e)} \qquad (15)$$

between the ground truth and the filtered mean and the negative log-likelihood $NL_x$ of the hidden states. The filtered state distribution is an approximate Gaussian $\mathcal{N}\big(\boldsymbol{\mu}_k^e, \mathbf{C}_k^e\big)$. The units of $M_x$ are standard deviations of $\mathbf{x}_{\text{true}}$ from the mean of the filter distribution. For both $NL_x$ and $M_x$, lower values indicate better performance. $NL_x$ penalizes both uncertainty and inconsistency, while $M_x$ solely penalizes inconsistency.

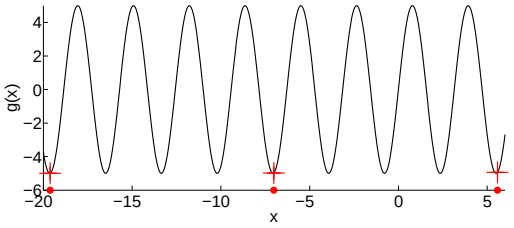



*Figure 3.* Typical failing of unscented filters. Although the function highly varies, the sigma points (red dots) are mapped to almost the same function value (red crosses). The sample predictive distribution is overconfident.

*Table 2.* Average filter performances (1D example).

| | $NL_x^{0.25}$ | $NL_x^{0.5}$ | $NL_x^{0.75}$ | $M_x$ |
|---|---|---|---|---|
| EKF | $2.4 \times 10^5$ | $2.9 \times 10^5$ | $3.5 \times 10^5$ | $30.2 \pm 3.2$ |
| UKF | $4.7 \times 10^4$ | $6.5 \times 10^4$ | $1.1 \times 10^5$ | $3.9 \pm 0.9$ |
| GP-UKF | 319 | $1.1 \times 10^3$ | $1.3 \times 10^4$ | $1.5 \pm 1.0$ |
| GP-ADF | **90** | **98** | **106** | $\mathbf{0.46 \pm 0.04}$ |

A distribution is *inconsistent* if the true underlying value is an outlier under the distribution.

Figure 3 shows that finite-sample approximations of densities can lead to overconfident predictions. The predictive distribution $p(y) = \mathcal{N}\big(-4.9, 0.0003\big)$ based upon finite samples claims full confidence. The actual measurement $y = -2.6$ cannot be explained.

Table 2 shows the average performance of the filters after 100 independent runs of the filter experiment. We report the upper and lower quantiles $NL_x^{0.75}, NL_x^{0.25}$ and the median of $NL_x$ as well as the mean and the standard deviation of $M_x$. According to $NL_x$, EKF is outperformed by all other filters. The EKF and the UKF heavily suffer from inconsistencies. The GP-UKF performs better than the UKF since particularly $\mathcal{GP}_g$ does not have training data in all relevant regions, which alleviates the overconfidence problem in Figure 3. According to the error measure $M_x$, GP-ADF yields substantially better results than all other filters. Moreover, the performance of GP-ADF is stable, which is expressed by the quantiles.

### 6.2. Recursive Filtering: Time-Series

We consider the problem of recursively filtering a time-series of a two-dimensional pendulum, where

$$\mathbf{x}_k = \begin{bmatrix} \varphi_{k-1} + \Delta_t \dot{\varphi}_{k-1} + \frac{\Delta_t^2}{2} \frac{mgl\sin(\varphi_{k-1}) + u_{k-1}}{ml^2} \\ \dot{\varphi}_{k-1} + \Delta_t \frac{mgl\sin(\varphi_{k-1}) + u_{k-1}}{ml^2} \end{bmatrix} + \mathbf{w}\,,$$
$$\mathbf{y}_k = \begin{bmatrix} \arctan\left(\frac{p_1 - l\sin(\varphi_k)}{p_1 - l\cos(\varphi_k)}\right) \\ \arctan\left(\frac{p_2 - l\sin(\varphi_k)}{p_2 - l\cos(\varphi_k)}\right) \end{bmatrix} + \mathbf{v}\,, \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \qquad (16)$$

are the time-discretized dynamics and observation models. We choose $\mathbf{\Sigma}_w = \text{diag}(0.1^2, 0.3^2)$, $\mathbf{\Sigma}_v = \text{diag}(0.2^2, 0.2^2)$. Here, $\mathbf{x} = [\varphi, \dot{\varphi}]^\top$ with $\varphi, \dot{\varphi}$ are the angle and the angular velocity, respectively. The applied torque is denoted by $u \in [-5, 5]\,\text{Nm}$, the acceleration of gravity is $\text{g} = 9.81\,\text{m/s}^2$, the length of the pendulum is $l = 1\,\text{m}$, the mass of the pendulum is $m = 1\,\text{kg}$. The discretization constant is $\Delta_t = 400\,\text{ms}$. The measurement equation (16) describes *bearings-only measurements* of the Cartesian coordinates of the pendulum tip and solely depends on the angle. Thus, the filter distribution of the angular velocity has to be reconstructed by using the cross-correlation information between angle and angular velocity in the transition dynamics model. We used 200 data points to train $\mathcal{GP}_f$ and $\mathcal{GP}_g$.

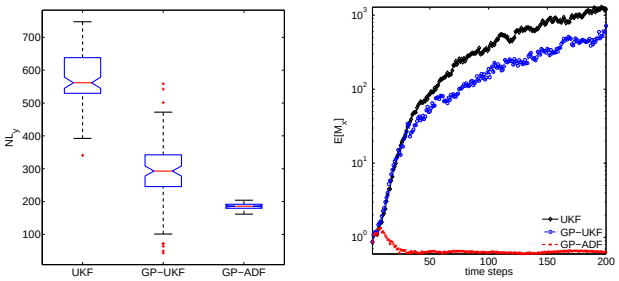
We start 100 independent trajectories from the initial state distribution $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0)$ with $\boldsymbol{\mu}_0 = [-\pi, 0]^\top$ and $\mathbf{\Sigma}_0 = \text{diag}([0.1^2, 0.2^2])$. This corresponds to the still pendulum hanging downward. We fuse information of a state prediction and a corresponding observation at each time step $k$. This filtered state distribution serves as prior for the subsequent state prediction. We iterate this procedure for 200 time steps.

In Figure 4, we compare the performances of the UKF, the GP-ADF, and the GP-UKF by considering $NL_y$, the negative log predictive likelihood of a full trajectory. $NL_y$ assesses whether the observations $\mathbf{y}_k$ can be explained by the predicted measurement distributions $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \mathcal{N}(\boldsymbol{\mu}_k^y, \mathbf{C}_k^y)$. Note that in contrast to $NL_x$, $NL_y$ solely depends on observations $\mathbf{y}$, and no longer on the hidden variables $\mathbf{x}$. Additionally, we consider the $M_x$-measure. A major observation is that the UKF and the GP-UKF are unaware of losing track of the state since the final covariances are tiny. Therefore, they often yield inconsistent solutions after 200 time steps, whereas the GP-ADF determines tight, but consistent distributions.

In general, we observed that the performance of GP-ADF is particularly good for non-negligible noise levels and fairly nonlinear mappings $f$ and $g$. If the state uncertainty is small or the functions $f$ and $g$ are nearly linear, the UKF and the GP-UKF perform well.

## 7. Discussion

Non-parametric probabilistic GP models describe distributions over all functions that plausibly explain the data. In the context of our work, this property matters if a parametric model cannot easily be determined or the real system does not closely follow idealized models.




(a) The median (notch), the lower and upper quantile (blue box), and the spread of the negative log predictive likelihood. The crosses are outliers.

(b) The $x$-axis shows the time steps, the $y$-axis displays the averaged Mahalanobis distance $M_x$ on a logarithmic scale.

*Figure 4.* Recursive filter performances of UKF, GP-UKF, and GP-ADF for the 2D pendulum. Panel (a) shows the negative log predictive likelihood $NL_y$. While the performances of the UKF and the GP-UKF vary strongly and depend on the particular noise realizations, the GP-ADF reliably provides a good solution. Panel (b) shows the averaged Mahalanobis distances of the filters. In contrast to the GP-ADF, the UKF and the GP-UKF quickly become inconsistent.

We observe that the uncertainty in the GP-ADF is often larger than the uncertainty in the UKF and the GP-UKF, which depends on two factors. First, in contrast to the GP-UKF, the GP-ADF explicitly incorporates the uncertainty about the underlying function. Second, the predictive uncertainty is computed using the entire prior. Due to the appropriate treatment of uncertainties, we observe that the predictions of the GP-ADF are rarely inconsistent.

Both UKF-based algorithms can easily fail when the functions, which are used for mapping the sigma points, are highly nonlinear and the input distribution is wide (see Figure 3). The UKF and EKF are solely applicable when the functions are known or directly accessible. If only samples of the underlying function are available, models have to be employed. Ko and Fox (2008) replace transition and measurement functions by GP models in standard filters, such as the EKF and the UKF. However, they do not exploit the GP structure that allows for an exact computation of the first two predictive moments given a Gaussian prior. Since Ko et al. (2007) and Ko and Fox (2008) do not exploit these properties, the GP-UKF is not moment preserving.

The GP-ADF can be considered the limit of the GP-UKF propagating infinitely many samples from a Gaussian input distribution if additionally the corresponding function values are sampled from the GP

predictive distribution.

Like Ko et al. (2007) and Ko and Fox (2008), we assume that the transition function and the measurement function can be learned by having access to ground truth observations of the hidden states. The measurement function could be learned independent of the transition function, but (measurement) noise-free observations of the hidden states in Figure 1 can be difficult to obtain.

For highly uncertain models for the latent functions $f, g$ GP-ADF is still consistent and shows the same stable performance as described in Figure 4(a).

## 8. Summary and Future Work

In this paper, we propose the GP-ADF, a fully Bayesian approach to assumed density filtering for nonlinear dynamics and observation models. Similar to the papers by Ko et al. (2007) and Ko and Fox (2008), we model the transition dynamics and the measurement function by GPs. However, we propagate full densities and approximate the predictive distribution by a Gaussian with the exact moments. In contrast to the EKF, the UKF, and the recent GP-UKF, our filter is consistent and moment preserving.

We will complete the forward-backward algorithm and learn the GP models for the transition dynamics and the measurements without the need of direct access to the hidden states. We will utilize Expectation Maximization for this purpose since GP-ADF allows for gradient-based parameter optimization.

## Acknowledgements

We thank Ryan Turner, Carl Edward Rasmussen, and the reviewers for very helpful comments and suggestions. MPD acknowledges support by the German Research Foundation (DFG) through grant RA 1030/1-3.

## References

Boyen, X., & Koller, D. (1998). Tractable Inference for Complex Stochastic Processes. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 33–42.

Ghahramani, Z., & Roweis, S. T. (1999). Learning Nonlinear Dynamical Systems using an EM Algorithm. In *Advances in Neural Information Processing Systems 11*, 599–605.

Julier, S. J., & Uhlmann, J. K. (2004). Unscented Filtering and Nonlinear Estimation. *IEEE Review*, *92*, 401–422.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME—Journal of Basic Engineering*, *82*, 35–45.

Kitagawa, G. (1996). Monte Carlo Filter and Smoother for non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, *5*, 1–25.

Ko, J., & Fox, D. (2008). GP-BayesFilters: Bayesian Filtering Using Gaussian Process Prediction and Observation Models. *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3471–3476.

Ko, J., Klein, D. J., Fox, D., & Haehnel, D. (2007). Gaussian Processes and Reinforcement Learning for Identification and Control of an Autonomous Blimp. *Proceedings of the International Conference on Robotics and Automation*, 742–747.

Kuss, M. (2006). *Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning*. Doctoral dissertation, Technische Universität Darmstadt, Germany.

Lefebvre, T., Bruyninckx, H., & Schutter, J. D. (2005). *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*. Springer Berlin.

Maybeck, P. S. (1979). *Stochastic Models, Estimation, and Control*, vol. 141 of *Mathematics in Science and Engineering*. Academic Press, Inc.

Opper, M. (1998). A Bayesian Approach to Online Learning. *Online Learning in Neural Networks*, 363–378. Cambridge University Press.

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

Rasmussen, C. E., & Ghahramani, Z. (2003). Bayesian Monte Carlo. In *Advances in Neural Information Processing Systems 15*, 489–496.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Simon, D. (2006). *Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches*. Wiley & Sons. First edition.